HOW TO RUN ESSENTIAL ANALYSES IN R

EDITION TWO

Correlations, Exploratory Factor Analysis, Confirmatory Factor Analysis, & Basics of Item Response Theory

NAREN SELVARATNAM

“At times overwhelming, but deeply satisfying programming language for statistical computing”

DOI: 10.61495/ejcr.0002.24

How to Run Essential Analyses in R Studio: For Undergraduate Students with No Background in Programming

Correlations, Exploratory Factor Analysis, Confirmatory Factor Analysis, & Basics of Item Response Theory

Second Edition

Dr. Naren D. Selvaratnam

B.A. (*Psych*)., M.Sc. (*Applied Psych*)., M.S. (*Ed LDRS*)., Ph.D. (*Ed LDRS*)

Eagan Journal of Contemporary Research (EJCR) is a peer-reviewed open-access e-journal introduced in the year 2021 to disseminate scientific knowledge in Sri Lanka with a specific focus on Psychology, Education, and Public Health disciplines. To further enhance the interest in scholarly work, the following short practical guide to R is written. This guidebook is the second issue of many to come.

Website: www.eaganjournal.com, Email: eaganjournal@gmail.com

EJCR is published by Deep Haven Counseling (DHC) (Pvt) Ltd.
Address: 117/2/39, Horahena, Kottawa, Pannipitiya

This e-book is distributed free of charge.

To

My parents,
Siblings,
Students,
Dr. Robert 'Bob' Howman,
The Land of Ten Thousand Lakes,
Uncle Sam,
&
The Pearl of Indian Ocean

Contents

Part A

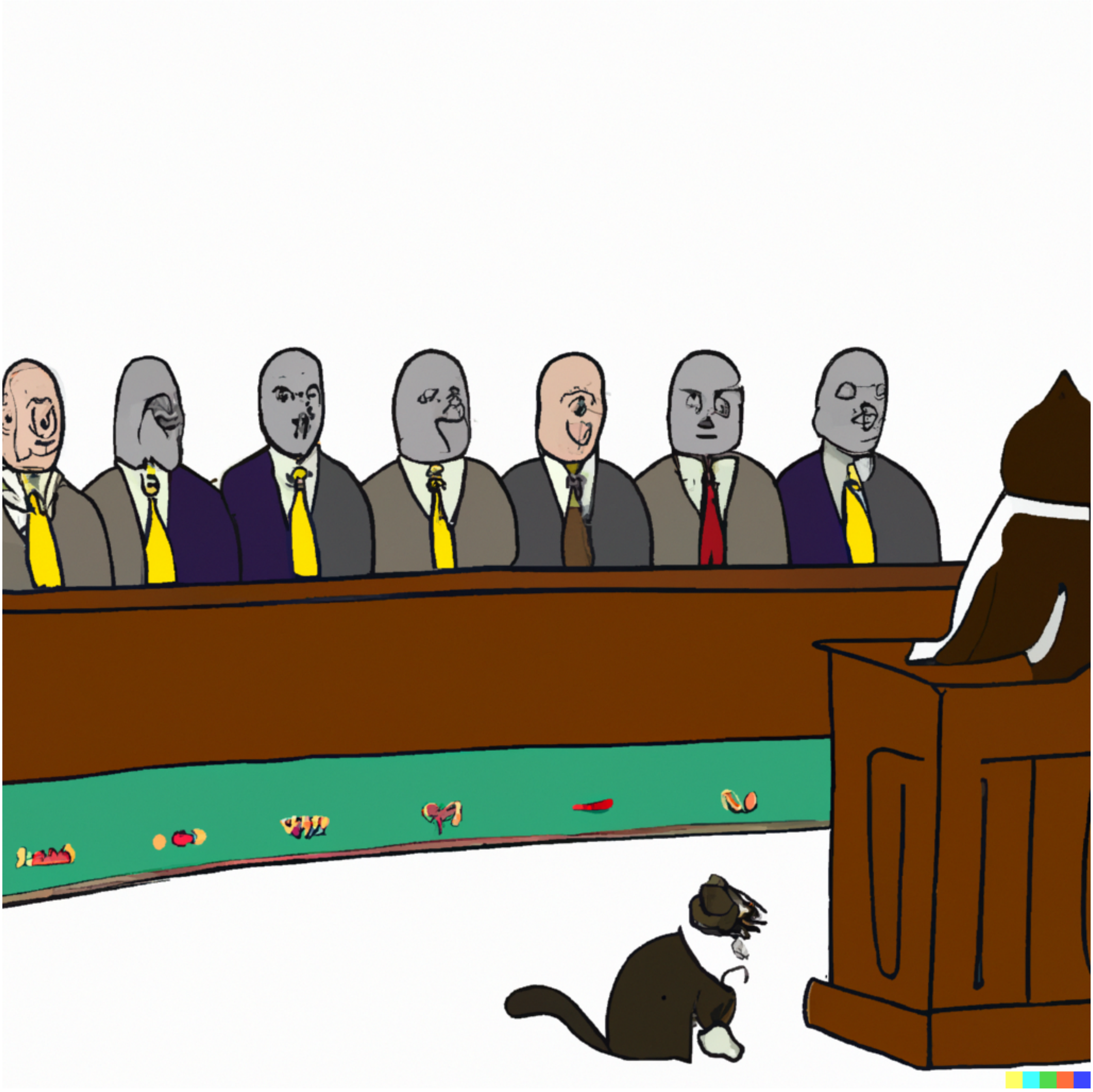
Part B

Part C

Additional lessons

Figures

Tables



To explore the unknown frontiers of Science

Foreword

Preface

This book is a culmination of years of work. During my first master's degree, I struggled to come up with a researchable problem. After multiple attempts with no luck, I was told by my supervisor to culturally adapt and statistically validate a scale. Out of desperation, I agreed. During my postgraduate years, I had no idea that this step into psychometrics was going to change my life forever. Although I was weak in high school mathematics, I learned to be good at statistics. As a result, some little voice in the back of my head kept on asking me to explore more about concepts such as *factor analysis (FA)* and *item response theory (IRT).* In the next few years up to 2017, I dedicated most of my time to self-learn exploratory factor analysis *(EFA).*

In my mind, *EFA* is always a straightforward set of statistics lessons. As a result, my true passion was to make sense of *IRT*. Now, *IRT* is not easy by any means, and it is not the animal you would want to tame alone. It did not matter how much I tried, I failed to make sense of the tough equations and the concepts. During this time, my dear colleague and statistician Dananjaya Hettiarachchi helped me to develop a research proposal for us to investigate the psychometric properties of our translated '*Sinhala generalized self-efficacy scale*' (*SGSES*). Similarly, Prof. Silas Bergen of Statistics Consulting at *Winona State University (WSU)* helped me to develop the required codes on *R Studio* to run the first few sets of analyses. Even after all of this, it took me a substantial number of years to have the confidence to talk about *IRT* and hone my programming skills to write this book. So, I take this time to state both my colleague and the professor how much gratitude I have for both of you for helping me out.

In the previous edition of this book "*How to Run Essential Analyses in R: For First Year Undergraduate Students with Limited Background in Programming*" the objective was to get students to understand the basics of *R* Programming. As a result, I have only included basic descriptive analyses, visualizations, and correlations in the first edition. In this second edition, you will find the same introductory concepts and in-depth elaborations on reliability, validity, *FA*, *EFA*, and *IRT*. I have also included a few sample scenarios to elaborate more on the application of psychometrics in psychological research and two more bonus lessons on simple linear regression and *confirmatory factor analysis (CFA).*

To get the maximum out of this book, I have divided this book into 3 noticeable sections. **Part A** introduces you to basic descriptive statistics and correlation (this entails basics for you to fully make sense of *FA*). In **Part B**, I show you *EFA* and how to achieve the required analyses using *R Studio*. In **Part C**, you learn the basics of the *IRT* and some comprehensive set of analyses using an *IRT* model. There are many *IRT* models and covering all of them is not the objective of this book. Taking that into consideration, I have included a practical example of running a graded response model *(GRM)* with narrative introductions to other basic and slightly complex *IRT* models.

In this edition of the book, you will also notice that I have included the full set of codes I have used to do all the analyses as a separate section. I did this for two reasons. Firstly, I have included the basics of *R* programming and its workings in **Part A** only. In **Parts B** and **C**, I have given codes with elaborations on the statistical concepts, and they lack an explanation of how the codes are constructed. One of the main reasons for this exclusion is to eliminate redundant explanations. However, newer forms of codes are supported with appropriate descriptions. It is my understanding that **Part A** helps students in learning the basics of the codes. Also, having a separate section on codes helps students observe patterns in the programming language and identify common methods in which *R* programming is used for statistical analyses. Secondly, if you are someone who knows all the concepts I have mentioned in the book and knows *R* programming a little bit and requires only the codes, then you can skip all the instructions and directly go to the *R Script* section. I also have included video lessons where applicable. Students who practice lessons with this book can also view the videos to further improve their understanding of the concepts and the process by watching short videos. I believe including more than one method of learning is ideal for a university textbook of this nature. Lastly, you will also find some sample results reports in this edition of the book. Since, most students need guidance in creating a good results section in a research report, I hope this new addition would help you in doing your coursework.

This book may not have everything you would expect, but, writing a book of this nature has always been a dream for me. I will continue to add newer sections and further explanations in the subsequent editions of this book. As a result, your feedback is very important for me to succeed in this endeavor. I would welcome both good and bad criticisms equally. So feel free to write to me at naren.d@sliit.lk if you have any feedback.

Thank you very much for considering this book for your studies.

Naren D. Selvaratnam

Abbreviations

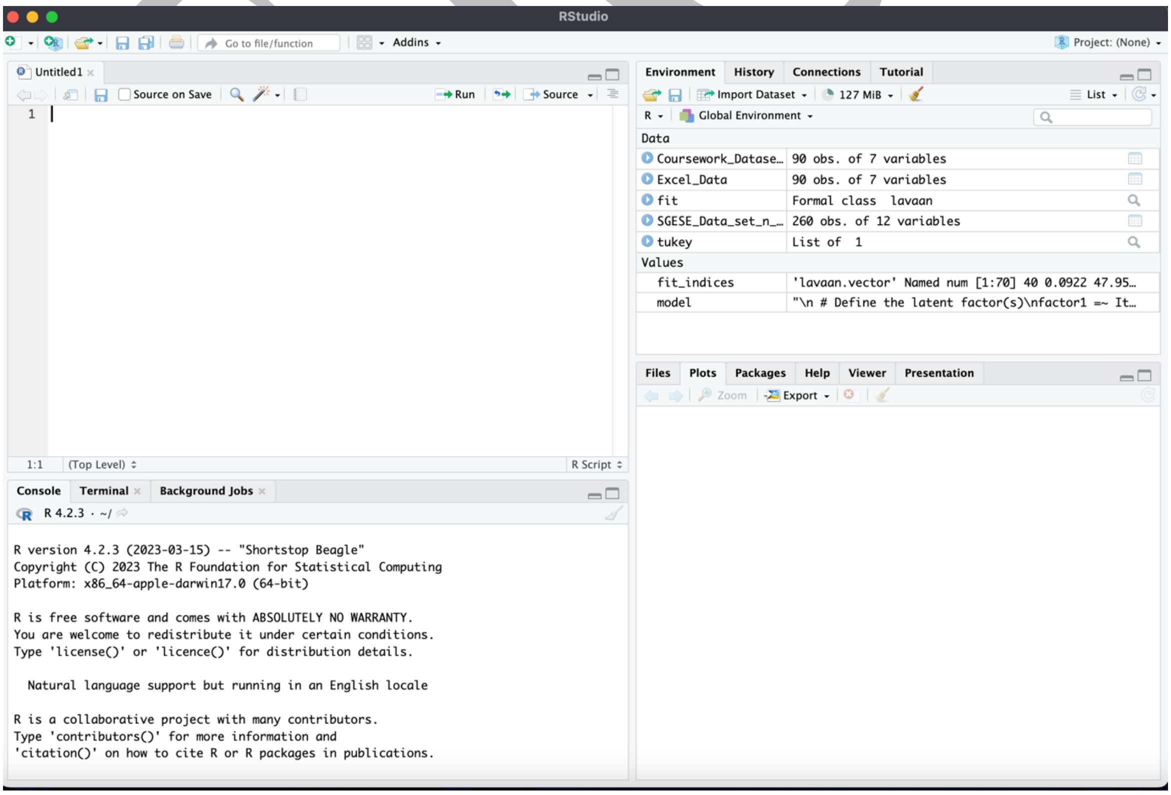
1. AIC – Akaike Information Criterion
2. ANOVA – Analysis of Variance
3. BDI – Beck's Depression Inventory
4. BIC – Bayesian Information Curve
5. BTS – Bartlett's Test of Sphericity
6. CFA – Confirmatory Factor Analysis
7. CFI – Comparative Fit Index
8. CTT – Classical Test Theory
9. DF – Degree of freedom
10. EFA – Exploratory Factor Analysis
11. FA – Factor Analysis
12. G6 – Guttman Lambda (λ) 6
13. GAD – General Anxiety Disorder
14. GPCM – Generalized Partial Credit Model
15. GRM – Graded Response Model
16. GSES – Generalized Self-Efficacy Scale
17. HAM-D – Hamilton Depression Rating Scale
18. ICC – Item Characteristic Curve
19. IIC – Item Information Curve
20. IRT – Item Response Theory
21. KMO – Kaiser-Meyer-Olkin
22. LRT – Likelihood Ratio Test
23. LTM – Latent Trait Modeling
24. MANOVA – Multivariate Analysis of Variance
25. MIRT – Multidimensional Item Response Theory
26. MR – Minimum Residuals
27. MSA – Measure of Sampling Adequacy
28. NNFI – Non-Normed Fit Index
29. NRM – Nominal Response Model
30. PCA – Principal Component Analysis
31. PCM – Partial Credit Model
32. RMSEA – Root Mean Square Error of Approximation
33. SGSES – Sinhala Generalized Self-Efficacy Scale
34. SLIIT – Sri Lanka Institute of Information Technology
35. SLR – Simple Linear Regression
36. SME – Subject Matter Expert
37. SPSS – Statistical Package for Social Sciences
38. SRMR – Standardized Root Mean Square Residual
39. TIF – Test Information Function
40. TLI – Tucker Lewis Index
41. WSU – Winona State University

Part A
1.0 Introduction

R is one of my go-to software for data analysis. Since *SPSS* does not offer some high-end analysis for us on a *Macintosh* operating system, I was always on the lookout for a better alternative. I have used *JASP* and *JAMOVI* as well. However, the degree to which it could assist me with various forms of statistical modeling is minimal. Unlike *SPSS*, *R* is more flexible and offers quite a lot of customization for researchers. At first, I was intimidated by *R*, but, slowly, I managed to learn the basics to help me run most of the analysis required to teach my undergraduate students. I am still learning *R* and the different packages it offers (more on packages later).

Essentially, *R* is an open-source software. First, you should install *R* and then *R Studio*. *R* will offer you a unique experience if you, like most of us, are used to running statistical analyses with a few clicks and interpreting ready-made outputs on *SPSS*. I personally love running analyses in *R Studio*. But, if you are new to *R*, you should learn the *R* language. However, there is no reason to worry as this guide will help you learn the basics of *R language* with an engaging psychological study and a simple set of variables to play around with.

I am by no means a programming expert. I am more or less like you with this software. I just have been using this for an extended period of time. As a result, I thought I should share what I have learned with you so that you can also effectively master *R*. In that manner, consider this book as a concise narration of a researcher who has no background in programming towards a similar group of researchers.



```
Data
Coursework_Datase…   90 obs. of 7 variables
Excel_Data           90 obs. of 7 variables
fit                  Formal class  lavaan
SGESE_Data_set_n_…   260 obs. of 12 variables
tukey                List of  1
Values
fit_indices          'lavaan.vector' Named num [1:70] 40 0.0922 47.95…
model                "\n # Define the latent factor(s)\nfactor1 =~ It…
```

```
R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

Figure 1: R Console

Now, before continuing this discussion further, let's get an idea about how *R* the software looks like when it is fully installed. Take a moment to carefully look at *Figure 1*. Unlike *SPSS*, here you will notice 4 windows. On the left, you notice two separate spaces. The upper gives your data view and also a place to develop your commands (scripts), and the lower is the place where you get the output of your commands. Similarly, you have two windows to the right in which the upper displays your variables and the lower shows the visual output of any analysis you conduct. Since you have just now learned some information about *R* console, you must be wondering how to install this software on your computer or laptop. I have a MacBook Air, and I am now explaining how I installed it on my laptop. First, you should go to cran.r-project.org. Once you are on this website, you can download *R*. Make sure to select the *R* that best fits your *Mac* or *PC*. Upon installing *R*, you should then install *R Studio*. What you see in *Figure 1* is *R Studio* when it is installed and opened.

1.1 Sample Study

To make the best of this study guide, I encourage you to download the Excel File named *'Cleaned_Data'* to your laptop. This is a dataset I developed with my first year first semester psychology students at Sri Lanka Institute of Information Technology *(SLIIT)*. We collected data from 59 students to examine the relationship between 'generalized self-efficacy' (Schwarzer and Jerusalem, 1995) and 'general anxiety' (Spitzer *et al.,*2006). Self-efficacy was measured by the generalized self-efficacy scale *(GSES)* and anxiety was measured by general anxiety disorder 7 *(GAD-7)*. We entered all our data into *Microsoft Excel* and did some data cleaning. This final dataset is after all the cleaning and we have retained some demographic variables as well to further assist you with the analyses to come.

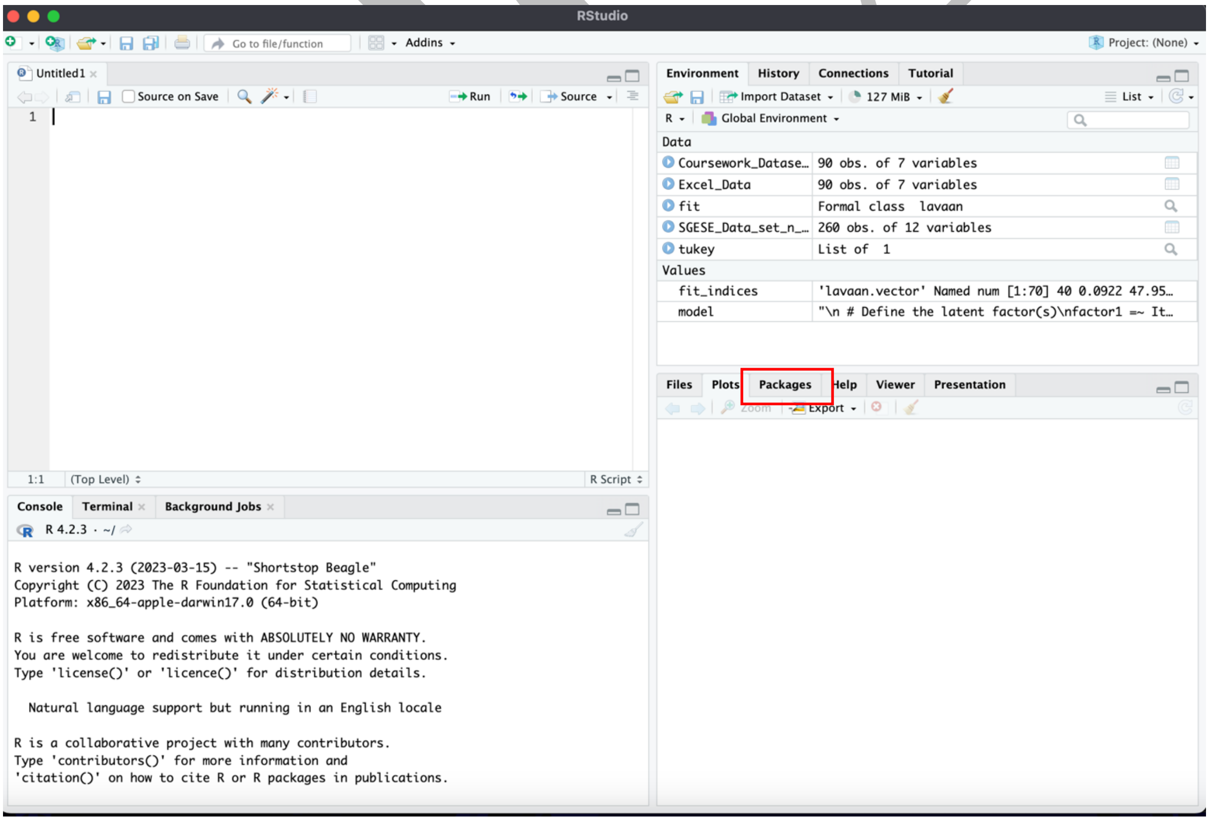
Variable	**Level of measurement**
Gender	Nominal
Age	Ratio
Year of study	Ordinal
GSES Q1	Ordinal
GSES Q2	Ordinal
GSES Q3	Ordinal
GSES Q4	Ordinal
GSES Q5	Ordinal
GSES Q6	Ordinal
GSES Q7	Ordinal
GSES Q8	Ordinal
GSES Q9	Ordinal
GSES Q10	Ordinal
GSES Total	Ratio
GAD7 Q1	Ordinal
GAD7 Q2	Ordinal
GAD7 Q3	Ordinal
GAD7 Q4	Ordinal
GAD7 Q5	Ordinal
GAD7 Q6	Ordinal
GAD7 Q7	Ordinal
GAD7 Total	Ratio

Table 1: Sample study variables

2.0 Understanding & Installing Packages in R Studio

Okay, since we now have a good understanding of how to install *R*. Let's take a moment to further understand the nature in which *R Studio* works. *R* is an open-source software and it enables researchers to modify and develop their own packages to run analyses specific to certain academic fields. For instance, although many in our field of psychology lack a preference for statistics, some of us feel excited working with larger datasets, especially to understand latent structures of psychological constructs. A latent structure can only be measured using a set of complicated processes of statistics (explained in-depth in **Part B** and **Part C**). Most psychology post-graduate programs cover the 'classical test theory' *(CTT)* procedures. However, 'item response theory' *(IRT)* and other 'latent trait modeling' *(LTM)* methods are hardly discussed. Most of the time, they are limited to individuals who are enrolled in psychometrics programs. Most of my colleagues hate all of these statistical concepts. I understand their feelings. But, once you start understanding research methods and statistics, a whole new dimension of reality opens up for students. I know it did for me. Now, despite the benefits statistics bring to our lives, the biggest problem we have as of now is the difficulty of obtaining affordable software for data analysis. Most software is expensive; but, it turns out that *R* is an affordable alternative.

In the *R Studio*, to run anything from basic descriptive statistics to complicated *LTM*, we need to install different packages. These packages are available to be downloaded from the 'packages' tab on the *R Studio*. The image given below shows where to locate this tab.



Data	
Coursework_Datase…	90 obs. of 7 variables
Excel_Data	90 obs. of 7 variables
fit	Formal class lavaan
SGESE_Data_set_n_…	260 obs. of 12 variables
tukey	List of 1
Values	
fit_indices	'lavaan.vector' Named num [1:70] 40 0.0922 47.95…
model	"\n # Define the latent factor(s)\nfactor1 =~ It…

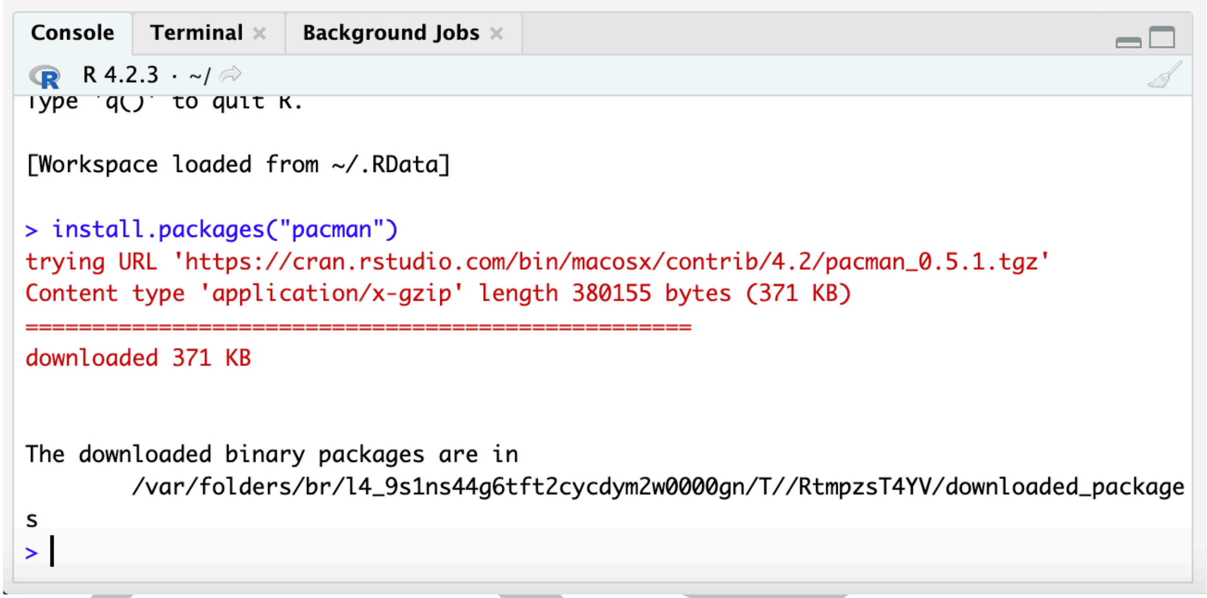
```
R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

Figure 2: Locating packages in R

Alternatively, I can enter a script to load a specific *R package* as well. In the proceeding lines, I am going to show you how to do this. Please note that *R scripts* are always colored in blue in this book while *R output* is colored in green. Now, to run some descriptive statistics, I am going to first install a package named ‘pacman.’ So, I am going to enter the below given *R* script.

install.packages (“pacman”)

It might not work if you just copy and paste. So, it would be better if you could write this yourself on the lower window to your left. When you enter *R* commands, take good note of the inverted commas. If you miss it here, it will not run. Once you type and press enter, this package will be installed. If it runs smoothly, you should get an output like this on your lower left window.



```
Type `q()` to quit R.

[Workspace loaded from ~/.RData]

> install.packages("pacman")
trying URL 'https://cran.rstudio.com/bin/macosx/contrib/4.2/pacman_0.5.1.tgz'
Content type 'application/x-gzip' length 380155 bytes (371 KB)
==================================================
downloaded 371 KB


The downloaded binary packages are in
        /var/folders/br/l4_9s1ns44g6tft2cycdym2w0000gn/T//RtmpzsT4YV/downloaded_package
s
> 
```

Figure 3: Running scripts in R

Once this process is completed, you should load the installed package to use it for analyses. This process has to be done each time you open *R* for analyses. To load, add the below-given lines to the same lower left box.

library(pacman)

Upon writing the above, hit enter, add the below-given script as well, and hit enter. Sometimes, you might have to manually write them if it does not seem to work.

pacman::p_load(pacman, dplyr, GGally, ggplot2, ggthemes, ggvis, httr, lubridate, plotly, rio, rmarkdown, shiny, stringer, tidyr)

Given below is the breakdown of the above code.

```
pacman::p_load(
  pacman,     # The 'pacman' package itself (required to use 'p_load')
  dplyr,      # Data manipulation package
  GGally,     # Extension of 'ggplot2' for exploratory data analysis
  ggplot2,    # Data visualization package
  ggthemes,   # Additional themes for 'ggplot2' graphics
  ggvis,      # Interactive data visualization package
  httr,       # Package for working with HTTP requests
  lubridate,  # Package for working with dates and times
  plotly,     # Interactive web-based plotting package
  rio,        # Data import/export package
  rmarkdown,  # Package for creating dynamic documents
  shiny,      # Web application framework
  stringer,   # String manipulation package
  tidyr       # Data tidying package
)
```

The above code entails multiple packages commonly used for psychology research. A combination of packages is usually used for statistical analyses. In this book, we will be using some of the packages and their functions, but not all of these. However, once you have completed this book, you will have the confidence to run more analyses using other packages as well.

Learning Check

1. Is R a programming language?
2. List all the variables given in the excel data file (*Cleaned_Data).*
3. Compared to *SPSS* and *Jamovi*, what are the difference you noticed in *R Studio?*

Answers available at the end of* **Part A

3.0 Useful R Packages for Psychology Undergraduates

Here is a list of useful *R packages* for psychology students. Here I have done a brief introduction for each package. Based on your preference, you may go ahead and explore others as well. Always remember, the first step is to get familiarized with the general ways and means of *R*. So, let's take a moment to review the listed packages.

a. **base:** Provides fundamental functions for basic statistical procedures
b. **dplyr:** This is an essential package for data manipulation. dplyr provides a variety of functions including filter(), select(), and group_by() for efficiently analyze large datasets. Data scientists call this process *data wrangling.*
c. **effsize:** Helps calculate and interpret effect sizes of statistical tests.
d. **ez:** Makes the analysis of factorial arrangements simpler.
e. **foreign:** Interprets datasets form *SPSS*, *SAS*, etc.
f. **ggplot2:** Helps visualize data and offers researchers a greater degree of customization.
g. **ggstatsplot:** An extension to ggplot2
h. **jtools:** Assists in summarizing regression models.
i. **lavaan:** Used in structural equation modeling and latent trait modeling
j. **lme4:** For fitting linear and nonlinear mixed-effects models.
k. **ltm:** Helps run item response theory models such as Rasch model, graded response model, etc. that is commonly used in psychometrics. Helps in investigating latent traits.
l. **MBESS:** Useful for effect size calculations.
m. **nortest:** Helps conduct normality tests
n. **psych:** Offers descriptive statistics, factor analysis, reliability analysis, etc. required for psychological testing and measurement.
o. **psychTools:** Provides various functions for psychological testing, including item analysis, test scaling, and scoring.
p. readxl: Provides the ability to import and read Microsoft Excel files.
q. **tidyr:** Helps in *tidying* data.
r. **pwr:** Helps with power and sample size calculations.

These packages cover a wide range of statistical analyses commonly used in psychology research, from data manipulation and visualization to hypothesis testing and basic forms of statistical modeling. I did not include some of the complicated packages as it does not fit the scope of this book. That being said, the best way to utilize this book is to have good research methods textbooks to go through as you learn the *R* programming language. I have listed two books you could find in both Sri Lanka and internationally for your reference. Both are excellent books.

Azam, S. M. F., Yajid, M. S. A., Tham, J., Hamid, J. A., Khatibi, A., Johar, M. G. M. Arrifin,I. A. (2021). *Research methodology: Building research skills*, McGraw Hill Education (Malaysia) Sdn Bhd, Kuala Lumpur: Malaysia

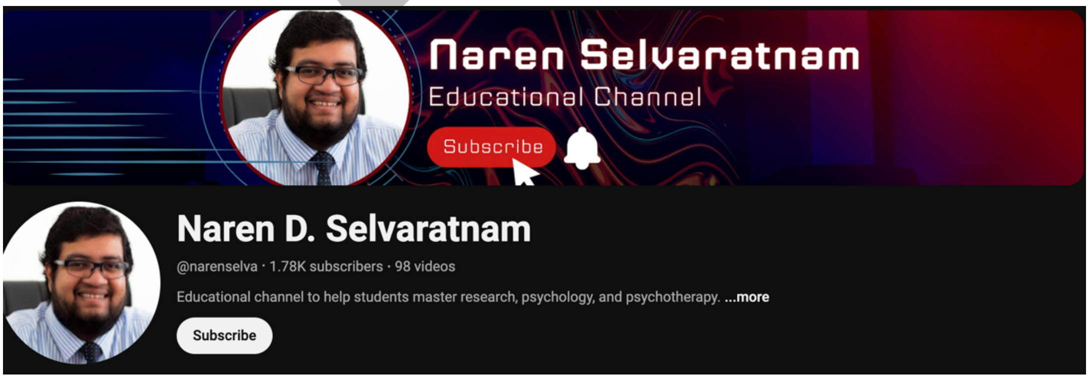
Gravetter, F. J., & Wallnau, L. B. (2017). *Statistics for the behavioral sciences, Tenth Edition*. Boston, USA: Cengage Learning.

My YouTube Research Playlist

To further help students learn the fundamental concepts of quantitative research methodology, I strongly advise you to review these videos so that you can begin the proceeding statistical analyses with confidence.

1. Qualitative vs. quantitative research
2. Research for beginners: The scientific method
3. Research questions & first impressions
4. Measurements in quantitative research
5. How to write a simple ‘draft’ research proposal (for absolute beginners)?
6. Introduction to Jamovi: Data entry and descriptive statistics
7. Correlations (a step-by-step guide)
8. Correlations Part II & Chi-Square test
9. Understanding validity and reliability in psychometric scales (basics)
10. How to write a results section?
11. Introduction to qualitative research
12. Qualitative Research Part II
13. Shapiro-Wilk, ANOVA, & Kruskal Wallis on R Studio (simple instructional video)
14. Introduction to Jamovi (descriptive statistics, scatter plots, & correlations)
15. T Tests, ANOVAs, and Factorial Designs (for psychology undergraduates)
16. T-Test & ANOVA manual computation
17. One-Way ANOVA (Jamovi): Data entry, analysis, interpretation, and reporting
18. Two-Way ANOVA (Jamovi): Data entry, analysis, interpretation, and reporting
19. 15 lessons I learned about research paper publications
20. My favorite research methods books
21. Making sense of review papers, introspective reports, and qualitative research reports
22. Factor analysis (exploratory & confirmatory) and reliability testing via Jamovi.
23. Mixed factorial design: Data entry, analysis, interpretation, & reporting
24. Two factor ANOVA & Eta Squared (manual computation)
25. Repeated measures T-Test & ANOVA (parametric and non-parametric)

All of the above videos are available on my YouTube channel, Naren D. Selvaratnam



Naren D. Selvaratnam

@narenselva · 1.78K subscribers · 98 videos

Educational channel to help students master research, psychology, and psychotherapy. ...more

Figure 4: YouTube channel information

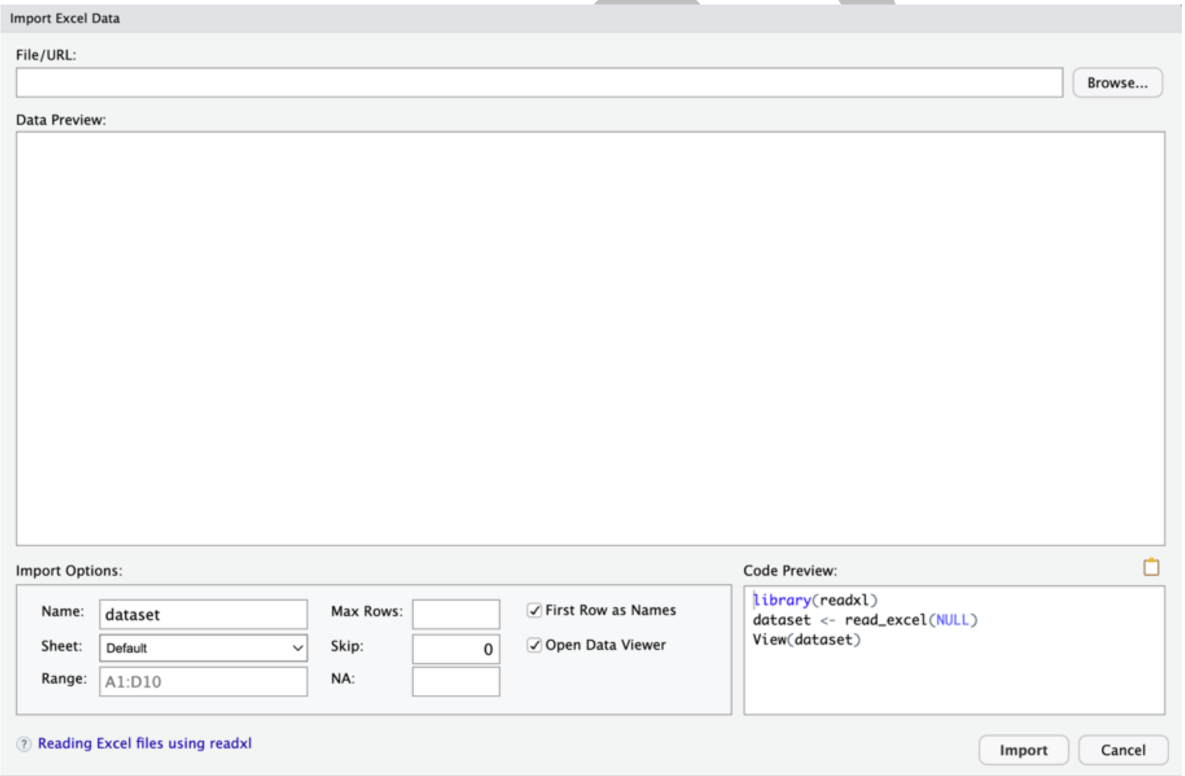
4.0 Loading Excel Data to R Studio

Now that you have loaded some of the packages. Let's learn how to load Excel data to *R*. In one of the previous sections, I introduced the variables we have. If you cannot recall, please take a moment to review them again. The collected data was cleaned and saved to a file named 'Cleaned_Data.' Let's import this dataset to *R Studio* now. Please follow the following steps.

First you should install the package **readxl**. Simply add the code given below. Even if you do not run it, *R Studio* will prompt you to install it.

```
install.packages (“readxl”)
```

Once this package is installed, click '**File'** and then go to **'Import Dataset'** and then select **'From Excel.'** You will get a pop-up window as shown in *figure 4*.

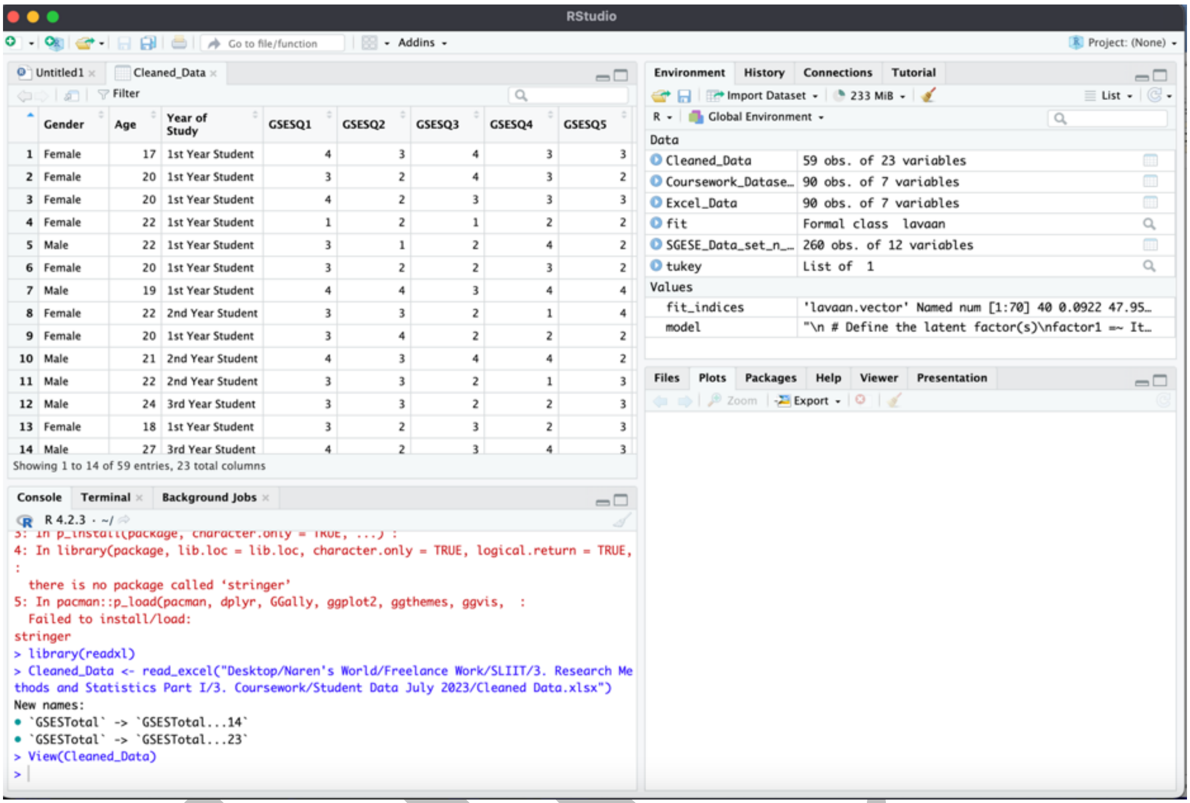


Name: dataset
Sheet: Default
Range: A1:D10
Max Rows:
Skip: 0
NA:
☑ First Row as Names
☑ Open Data Viewer

```
library(readxl)
dataset <- read_excel(NULL)
View(dataset)
```

Figure 5: Loading data

Click on the **'Browse'** and select your Excel file from your *PC* or *Mac*. The Excel sheet should contain data added appropriately. Remove unnecessary variables and calculations from your *Excel* to properly load it on *R Studio*. We call this process data cleaning. Let's say you collected data through a Google form. It is a good practice to see whether you have participants who have missed questions, completed the survey in an extremely short period of time, those who have unusual answer combinations, etc. This is a slow process; but, we should spend some time to understand the kind of data we have in our dataset. For instance, in the present dataset, there were participants with extremely high efficacy and high anxiety. This is an unusual combination since anxiety precipitates in the absence of self-beliefs. In that sense, quantitative researchers should have an in-depth understanding of the theoretical basis of constructs utilized in the study. I will explain more about constructs and their measurements in a later edition under latent trait modeling. For now, we will focus on fundamentals. So, load only the Excel data that you have cleaned. Name your dataset as well. For this example, I am going to name my data set “Cleaned_Data.” Once you import your data, your console should look like this *(Figure 6)*. You will notice that my data now appears on the left upper window.



	Gender	Age	Year of Study	GSESQ1	GSESQ2	GSESQ3	GSESQ4	GSESQ5
1	Female	17	1st Year Student	4	3	4	3	3
2	Female	20	1st Year Student	3	2	4	3	2
3	Female	20	1st Year Student	4	2	3	3	3
4	Female	22	1st Year Student	1	2	1	2	2
5	Male	22	1st Year Student	3	1	2	4	2
6	Female	20	1st Year Student	3	2	2	3	2
7	Male	19	1st Year Student	4	4	3	4	4
8	Female	22	2nd Year Student	3	3	2	1	4
9	Female	20	1st Year Student	3	4	2	2	2
10	Male	21	2nd Year Student	4	3	4	4	2
11	Male	22	2nd Year Student	3	3	2	1	3
12	Male	24	3rd Year Student	3	3	2	2	3
13	Female	18	1st Year Student	3	2	3	2	3
14	Male	27	3rd Year Student	4	2	3	4	3

Showing 1 to 14 of 59 entries, 23 total columns

```
3: In p_install(package, character.only = TRUE, ...) :
4: In library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE,
:
  there is no package called ‘stringer’
5: In pacman::p_load(pacman, dplyr, GGally, ggplot2, ggthemes, ggvis,  :
  Failed to install/load:
stringer
> library(readxl)
> Cleaned_Data <- read_excel("Desktop/Naren's World/Freelance Work/SLIIT/3. Research Me
thods and Statistics Part I/3. Coursework/Student Data July 2023/Cleaned Data.xlsx")
New names:
• `GSESTotal` -> `GSESTotal...14`
• `GSESTotal` -> `GSESTotal...23`
> View(Cleaned_Data)
> 
```

```
Data
Cleaned_Data          59 obs. of 23 variables
Coursework_Datase…    90 obs. of 7 variables
Excel_Data            90 obs. of 7 variables
fit                   Formal class  lavaan
SGESE_Data_set_n_…    260 obs. of 12 variables
tukey                 List of  1
Values
  fit_indices         'lavaan.vector' Named num [1:70] 40 0.0922 47.95…
  model               "\n # Define the latent factor(s)\nfactor1 =~ It…
```

Figure 6: Excel data loaded

Take a moment to browse through your excel data. Scan whether you have all your variables. If you have a trackpad, simply by swiping it to left and right, you can observe the full dataset. Once the dataset is loaded, you are all set to commence using *R* commands.

Learning Check

4. What is an *R* package?
5. Why should researchers spend time cleaning data before transferring them to *R Studio?*
6. As per your understanding what would be a good package to analyze large datasets?

Answers available at the end of* **Part A

5.0 Defining Variables in R Studio

Upon loading data to *R Studio*, the next step for us is to define our variables. Usually, this process is done for categorical variables. Once you define these categorical variables, it is easy for us to proceed with any other analyses to come. But, before we go any further, let's try to comprehend the logic behind defining variables in *R studio*. Similar to many other programming languages variables have to be defined due to several crucial reasons. I have listed some of them down below.

- **Memory Allocation:** When you define your variables, the system remembers them. Usually, when we define, we allocate a new name and this newly defined variable can be accessed at any time during your analysis session. Such allocation of memory is important for efficient data manipulation and analysis.

- **Data Integrity:** Defining variables helps researchers to further specify their data type (e.g., numeric, character, logical) and initial values of the variable identified for defining. This process boosts the integrity of our data.

- **Debugging and Error Handling:** Having properly defined variables makes it easier for the researcher to debug if there are any errors in the codes written.

- **Readability and Documentation:** Properly defined variables are readable and help researchers efficiently handle analyses. Being able to read a code clearly is a bigger advantage than having defined variables. You may also consider this as a process of documentation. In this book, even I am doing some form of documentation for you. In that sense, my codes will be mostly self-explanatory once you go through the first few of them.

- **Scope:** Variables in R have a scope that helps researchers determine how to access and modify these variables. In this edition of the book, I have not included this aspect. To understand the concept of variable scope, it would be better to know the very basics of R. Once you finish this book, you will be more confident to understand the rest.

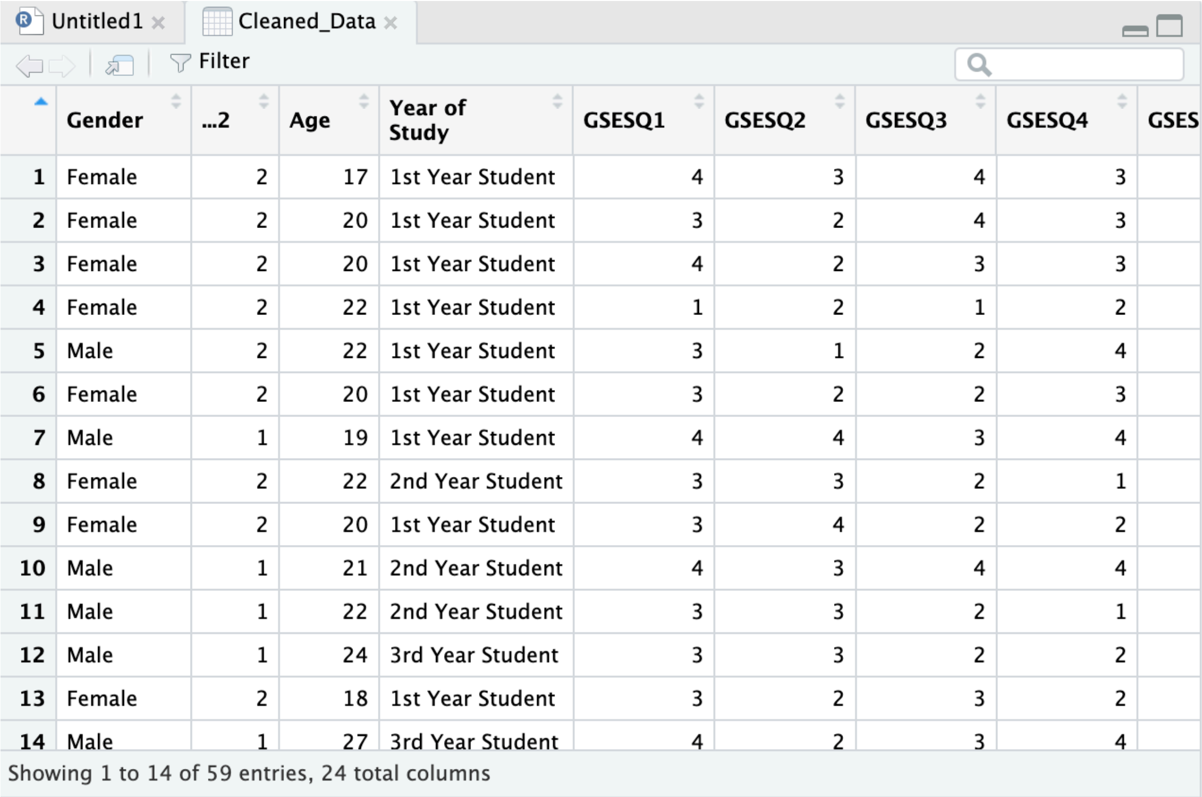
Now that we know the benefits of defining variables in *R Studio*, see an example in *R* of defining variables from the dataset we have loaded already. Since I have 2 categorical demographic variables, I am going to first define them. Let's start with 'Gender.' The below-given code has to be entered into the *R console*.

```
> RGender <- table (Cleaned_Data$Gender)
```

In this above code, 'RGender' is the new variable I created using the existing variable 'Gender' in the dataset 'Cleaned_Data' which is already loaded to *R Studio*. Hereafter, for most of my analyses, instead of 'Gender', I will be using 'RGender.' I will point out any exceptions to this rule. If you run the above code, it commands *R* to create a table of frequencies for values of the 'Gender' variable in a data frame called 'Cleaned_Data.' Let's try to further break down this code for us to further understand its elements.

- **Cleaned_Data$Gender:** In this extracted portion of the above code, the data frame is 'Cleaned_Data.' '$Gender' is used to extract the values from the 'Gender' column of the dataset named ‘Cleaned_Data.’ The ‘Gender’ column should contain categorical (nominal) data such as “Male,” “Female,” etc. In the loaded dataset, data of the “Gender” variable are entered using words instead of numbers.

- ‘$Gender’ uses the ‘$’ operator to extract a specific column from the data frame identified. In this case, the ‘Gender’ column is extracted from the data frame.



	Gender	...2	Age	Year of Study	GSESQ1	GSESQ2	GSESQ3	GSESQ4	GSES
1	Female	2	17	1st Year Student	4	3	4	3	
2	Female	2	20	1st Year Student	3	2	4	3	
3	Female	2	20	1st Year Student	4	2	3	3	
4	Female	2	22	1st Year Student	1	2	1	2	
5	Male	2	22	1st Year Student	3	1	2	4	
6	Female	2	20	1st Year Student	3	2	2	3	
7	Male	1	19	1st Year Student	4	4	3	4	
8	Female	2	22	2nd Year Student	3	3	2	1	
9	Female	2	20	1st Year Student	3	4	2	2	
10	Male	1	21	2nd Year Student	4	3	4	4	
11	Male	1	22	2nd Year Student	3	3	2	1	
12	Male	1	24	3rd Year Student	3	3	2	2	
13	Female	2	18	1st Year Student	3	2	3	2	
14	Male	1	27	3rd Year Student	4	2	3	4	

Showing 1 to 14 of 59 entries, 24 total columns

Figure 7: Loaded Excel data

- **table():** The ‘table()’ function in *R* is used to create the frequency table. It takes a vector as input and gives us back a table that displays the frequency (count) of each unique value in that vector. In this case, the vector is the values in the ‘Gender’ column of ‘Cleaned_Data.’

    - **What is a vector?** A vector is a ‘one-dimensional data structure.’ In *R*, a vector is a fundamental data structure that can hold a sequence of values of the same data type. In our example, the vector holds categorical values that represent two types of ‘Gender’, such as ‘Male’ and ‘Female.’ This is a one-dimensional data structure (aka *1D arrays*). Refer to the section on terms for a concise explanation of *1D arrays*.

- The ‘Gender’ column contains data as displayed in *Figure 7*. Thus, the resulting ‘RGender’ table might look like this.

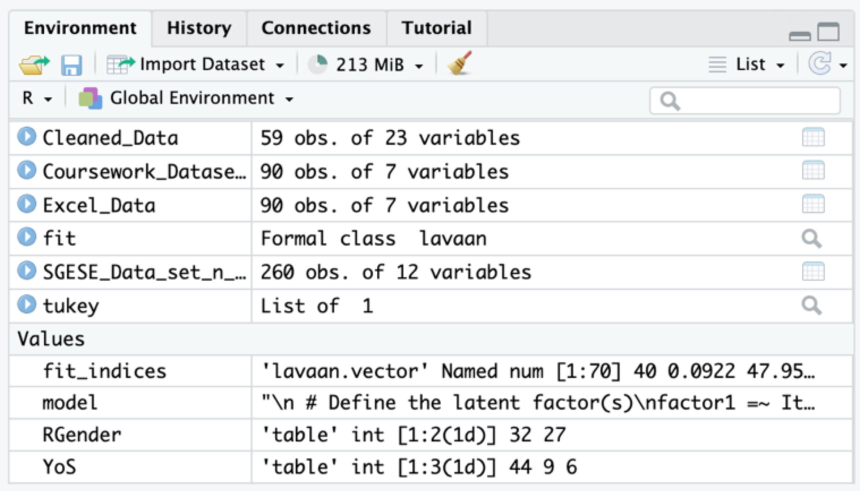
```
Gender
Female    Male   Other
   2        3       1
```

- In the above example, the count for **Male** is **3**, **Female** is **2**, and for **Other** is **1**. These kinds of frequency tables are excellent for summarizing categorical variables and to understand the nature of the collected data.

Since our ‘Year of Study’ variable is also categorical, we can repeat the same process for ‘Year of Study’ as well. The *R* command you should enter in the console is given below.

```
YoS <- table(Cleaned_Data$`Year of Study`)
```

Once these are entered, you will notice, your right topmost square will now show new variables entered.



Cleaned_Data	59 obs. of 23 variables
Coursework_Datase…	90 obs. of 7 variables
Excel_Data	90 obs. of 7 variables
fit	Formal class lavaan
SGESE_Data_set_n_…	260 obs. of 12 variables
tukey	List of 1
Values	
fit_indices	'lavaan.vector' Named num [1:70] 40 0.0922 47.95…
model	"\n # Define the latent factor(s)\nfactor1 =~ It…
RGender	'table' int [1:2(1d)] 32 27
YoS	'table' int [1:3(1d)] 44 9 6

Figure 8: Newly created variables

With this lesson, we have completed most of the basic lessons required to run simple statistical procedures. In the proceeding sections, I will take you through descriptive statistics, normality testing, and correlations. There are a lot of statistical procedures that could be introduced in this book, but, to make this a feasible and entertaining read, I included the most essential ones for a first-year student.

Learning Check

7. The name of the dataset is **SGESData**, and the categorical variable that requires a frequency calculation is **BirthOrder**. Write a code to create a new variable named **BO** that could help us generate frequencies for **BirthOrder** later.

Answers available at the end of* **Part A

6.0 Descriptive Statistics in R

Something I have noticed often in student coursework is the limited emphasis they place on descriptive statistics. Oftentimes, students attempt to directly test hypotheses the moment they get hold of the datasets. Accordingly, their research reports mostly contain inferential statistical procedures as opposed to a critical analysis of data. Although inferential statistical measures are essential to evaluate hypotheses, it is important students spend substantial time examining other trends within their data. This objective of unraveling trends within data can be achieved by running effective descriptive analyses. As a result, in this section, I am introducing some useful *R* commands to facilitate descriptive analyses.

6.1 Working with Categorical Data

Now let's do some descriptive analysis. I have colored codes in blue and output in green. Let's first obtain some summaries of the categorical variables we have already defined. This can be easily obtained by typing the code given below in the *R console*. You may have noticed that I use *R Studio* and *R Console* interchangeably. Make sure to check the difference between these two before proceeding further. Check **Page 109** for definitions. Here I have given both the input and the output.

```
> summary(RGender)
```

```
Number of cases in table: 59
Number of factors: 1
```

summary() is a generic function in *R* that provides summary statistics and information about the structure of an object. You can obtain detailed information by giving the *R* command, **str()** displayed below. Here also, I have given both the input (code) and the output for your reference. Please make sure to try these right now so that you know how to enter these codes and obtain output. If you get error messages, you should try to type in the code. Usually, when you type the name of your defined variable, *R Studio* gives a pop-up list of variables, and you can choose the appropriate variable from the list.

```
> str(RGender)
```

```
Female   Male
   32     27
```

Now, we have some understanding of the nature of our categorical variable (RGender). Our total sample is n = 59. We have 32 female participants and 27 male participants. We can visualize this categorical variable and its frequencies by typing in the below-given *R* command. This command helps us to generate a bar chart. Since our data is categorical, for visualization purposes, a bar chart would be ideal.

```
> barplot(RGender)
```

When you enter this command, the below-presented image will appear on the bottom right window of R Studio.

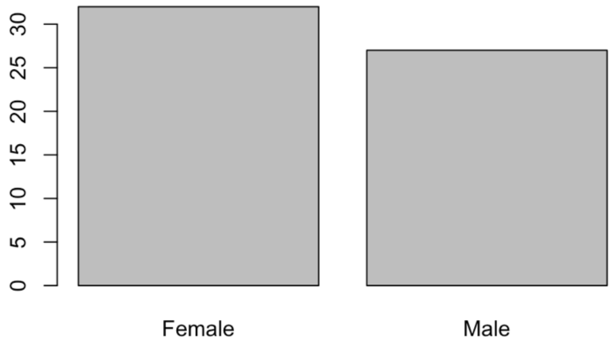


Figure 9: Gender visualization

Unlike SPSS, R Studio provides a greater degree of flexibility, so, I am going to add a new title and change the color of the above bar plot. Now, instead of just '> barplot(RGender)' I am going to add a couple more components to the above code. Given below is my new code.

```
> barplot(RGender, col = "white", main = "Gender Frequencies")
```

In the above, after typing RGender, I have included a comma, and have then typed '**col**' which is a function we can enter to add a color to our figure. Here, I have added "**white**" within inverted commas, and I have written the full word in lowercase. I have then added another comma and have written "**main**" which stands for the main title. After adding this, I ran my new code, and here is my new output. What do you think?

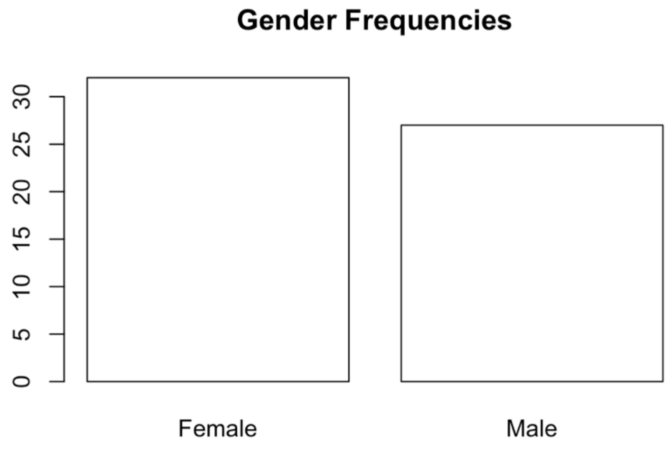


Figure 10: Gender visualization, new elements added to the visualization

R Studio is a great platform for developing high-quality visualizations. You may simply right-click the image and select 'save image as' to go ahead and save your generated bar plot in a format (i.e., jpeg, png, etc.) that you think fits. So, if you think research is your calling, it is not a bad decision to master these skills right away to be good at descriptive analysis.

Now that we know how to do some descriptive statistics, let's check our second categorical variable 'Year of Study' as well. I have presented the codes and the findings below from our dataset.

```
> summary(YoS)
Number of cases in table: 59
Number of factors: 1
```

```
> str(YoS)
 'table' int [1:3(1d)] 44 9 6
 - attr(*, "dimnames")=List of 1
  ..$ : chr [1:3] "1st Year Student" "2nd Year Student" "3rd Year Student"
```

Unlike the output of RGender, the **str()** this time has given some more details about our data. Let's decode this information one by one so that we can make sense of our descriptive data.

- **'table':** This indicates the researcher the data type of the object. In this example, it is a table that is commonly used in R for storing frequency or contingency tables.

- **int:** This lets the researcher know that he/she is working with integers (whole numbers).

- **[1:3(1d)]:** In this component, '1d' tells the researcher that this data has one dimension **(1d)** and consists of 3 distinct values **(1:3)**. This means that the table is a one-dimensional vector with three values.

- **44 9 6:** These are the frequencies associated with the categories or levels. Although a vector displays the same type of data, in our example there are 3 distinct values. Each value has a specific frequency. In this analysis, 44 students are first-year students, 9 students are second-year students, and 6 are 3rd year students.

- **attr(*, "dimnames")=List of 1:** In this bit of the output, "**dimanames**" refers to dimension names which are often used to label the levels or categories in the table. In some sense, this information provides metadata about our object. The "**List of 1**" indicates the '**dimnames**' attribute contains a list *(vector)* with one element. In R, a list is a data structure that can hold multiple values of different types. However, since we have previously defined our variable (YoS), in this case, the list, which now is a vector contains one element which is used to store the names of the dimensions.

- **..$: chr [1:3]** "1st Year Student" "2nd Year Student" "3rd Year Student": These are the dimension names, and they are of character data type **(chr).** They correspond to the labels for each count in the table. In this case, they represent the categories or levels: "1st Year Student," "2nd Year Student," and "3rd Year Student."

As described above, if one intends, one can obtain an in-depth insight into a specific variable in a dataset. This is useful to determine further analyses that are required to do for future analyses.

Very similar to what we did earlier, we can also generate a bar chart for this categorical variable (YoS) by the following *R* command.

```
> barplot(YoS)
```

The resulting bar chart for the above command is presented below.

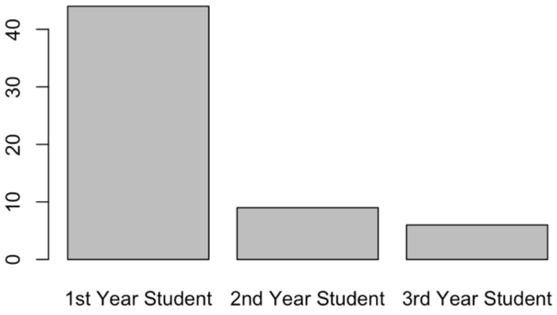


Figure 11: YoS visualization

Similar to what we did with the RGender variable, this barplot can also be made more meaningful by adding a couple more components to the code. Here is an example with the output. Just to make things interesting, this time, I changed the color of the barplot.

```
barplot(YoS, col = "lightblue", main = "Year of Study")
```

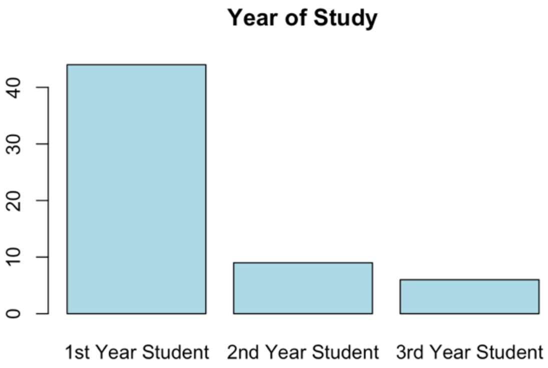


Figure 12: YoS visualization with new colors

With what we have done in the above analyses, I hope now you have some understanding of analyzing categorical variables and making sense of them. Now, from this, let's move to discuss further analysis to do with our continuous data.

6.2 Working with Continuous Data

Considering what we have examined in our categorical data, I think we have a world to explore here with our continuous data. To enable my work with continuous data, we can load our packages.

```
> pacman::p_load(pacman, dplyr, psych)
```

If you want to install a specific package, let's say **'psych'** then you can add this code.

```
install.packages ("psych)
load(psych)
```

Now, the first thing I am going to do is to obtain some summary statistics about my continuous variable in the 'Cleaned_Data' named 'Age.' For that, I am going to add this code.

```
summary_stats <- describe(Cleaned_Data$Age)
> print(summary_stats)
```

Before proceeding further, let's try to decipher the above code. I have listed the main components of the code and their explanations.

- **describe(Cleaned_Data$Age):** Here the describe() is used with the argument "Cleaned_Data$Age." To understand this first start with "Cleaned_Data$Age." The $ says the variable 'Age' of our data file named "Cleaned_Data." This code essentially describes the "Age" variable of the dataset named "Cleaned_Data" simply put.

- **summary_stats <- describe(...):** Once the results are obtained for **describe(Cleaned_Data$Age),** store the results in a variable called "summary_stats." In R Studio **<-** operator is used to assign the results of a calculation or function to a variable. If you recall, we used the same operator when we were defining variables as well.

- Accordingly, we enter the first code, summary_stats <- describe(Cleaned_Data$Age)

- **print(summary_stats):** After the first code is entered, print() provides the results stored in "summary_stats." I have given the output below. Let's see whether we can make sense of the obtained results.

```
   vars  n  mean  sd  median  trimmed  mad  min  max  range  skew  kurtosis  se
X1    1  59   21  1.64   21     20.96  1.48   17   27     10    0.6     1.86  0.21
>
```

Unlike the output of categorical variables, here for this continuous variable, we have quite a lot of information. Let's try to understand what each of these columns tells us about our variable, 'Age.'

- **vars:** variable name, X1.

- **n:** sample size, or the number of observations. The variable X1 has 59 observations. So, one could say there are 59 individual informants associated with this variable.

- **mean:** Here, the mean is the average of the variable. In this case, the mean age is 21 years for the sample of 59.

- **sd:** Standard deviation, which is the variability of data, given at 1.48.

- **median:** The middle value of the data, which is also 21.

- **trimmed:** 'Trimmed' here stands for the Trimmed mean which is the average calculated upon removing a specific percentage of extreme values from both ends of the data.

- **mad:** The "mad" column represents the median absolute deviation, a measure of data variability that is less sensitive to extreme values than the standard deviation.

- **min:** Minimum value of the variable. In this analysis, it is 17.

- **max:** Similar to the minimum of 17, the maximum is 27. This means the age varies between 17 and 27 years.

- **range:** Max – Min (27 – 17) = 10.

- **skew:** Symmetry of the data distribution. Usually, if these values range between -2 and +2, and close to zero, the distribution is mostly likely normal. This can visually be observed using histograms (discussed in later sections). In this analysis, skewness is 0.6 which is most likely to be normal, but there is a slight right skew.

- **kurtosis:** Kurtosis measures the "tailedness" of the data distribution. A positive kurtosis indicates heavy tails (narrower than a usual normal distribution making more extreme values on tails likely), while a negative kurtosis indicates light tails. For "X1," the kurtosis is 1.86, suggesting slightly heavier tails than a normal distribution. This also means the distribution is **leptokurtic** (narrower, thus, extreme values in tails are likely).

- **se:** Standard error of the mean which estimates a variability of 0.21 for the sample mean.

If you look closely at our output, it gives a wide range of descriptive information about **central tendency** (mean, median, mode), **variability** (standard deviation), and the nature of our **distribution** (skewness, kurtosis). Information about variability and the shape of the distribution is extremely useful when selecting most types of inferential statistical procedures as demonstrating normality is an assumption in the parametric strain of inferential statistics.

In my original Excel dataset, I have created a total variable for both **GSES** and **GAD7**. I did this in Excel itself to avoid unnecessary hassle. I performed the same commands I ran for **Age** to understand the nature of the **GSES** variable. This is also a continuous variable.

```
GSES_Summary <- describe(Cleaned_Data$GSESTotal...15)
> print(GSES_Summary)

    vars  n  mean   sd  median  trimmed  mad  min  max  range  skew  kurtosis  se
X1     1  59 29.12 3.61    29      29.1  2.97   20   38     18  0.01      0.08  0.47
>
```

Can you try to interpret the above data? Now, unlike the Age variable, if you look at skewness and kurtosis values, both of them indicate a greater degree of normality (values are closer to zero). So, this data is most likely to be normally distributed. We can visually observe normality and also test using a normality test like *Kolmogorov-Smirnov* or *Shapiro-Wilk* test.

Similar to running analyses for one continuous variable at a time, we can further do statistical analyses by dividing the **GSES** total by a categorical variable like **Gender**. This will help you notice how the **GSES** scores tends to vary between males and females. In *SPSS* you can perform the same in *Custom Tables*, or *Descriptive-Explore* option whereas in *Jamovi* you can obtain the same by obtaining descriptives for a continuous variable by splitting it using another categorical variable such as Gender of Year of Study. In *R*, the same can be achieved by writing a simple script. Take a moment to go through the script I have produced. Here, I have used my original '**Gender**' variable instead of the newly defined '**RGender**.'

```
> mean_by_gender <- Cleaned_Data %>%
+    group_by(Gender) %>%
+    summarise(Mean_GSESTotal = mean(GSESTotal...15, na.rm = TRUE))
```

Once you enter the above, then enter the final line.

```
> print(mean_by_gender)
```

Here is your output. Take a moment to observe the output. It provides the mean of **GSES** based on the levels of our categorical variable, **Gender**.

```
# A tibble: 2 × 2
  Gender    Mean_GSESTotal
  <chr>              <dbl>
1 Female              28.4
2 Male                30.0
>
```

Now that we have completed this analysis, let's try to further decode this *R* command.

- **mean_by_gender <-** ...: Here, we are creating a new variable called "mean_by_gender" to store the results of our calculation. <- indicates the results are assigned to this variable.

- **Cleaned_Data %>%..:** Here, %>% is a pipe operator. Usually, we use this in R to chain together a series of statistical operations. What this code says is to start with the 'Cleaned_Data' and then perform the next set of operations.

- **group_by(Gender) %>%** ...: Here, we're grouping the data by the "Gender" variable. So, we are separating the dataset into two groups because Gender has two distinct types of values one representing Males and the other representing Females.

- **summarise(Mean_GSESTotal = mean(GSESTotal...15, na.rm = TRUE)):** Within each Gender group, now we want to calculate a summary statistic which is GSES mean. Please note the variable that contains the composite score of GSES in my uploaded dataset is named GSESTotal…15. The function summarise() is to create this summary. Essentially, this says "For each group, calculate the average of the 'GSESTotal...15' variable and call it 'Mean_GSESTotal'."

- **mean(GSESTotal...15, na.rm = TRUE):** This part calculates the mean (average) of the "GSESTotal...15" variable within each group. The na.rm = TRUE part tells R to ignore any missing values (NA) when calculating the mean.

So, in simple terms, this code is grouping the "**Cleaned_Data**" by gender, and for each group (e.g., males and females), it calculates the average of the "**GSESTotal...15**" variable. The results are stored in a new variable called "**mean_by_gender**," which will contain the average values for each gender group. This can help us understand how the "**GSESTotal...15**" variable differs between different genders in the dataset. You can add further functions to the same code. For instance, for the above analysis, I added another command to generate standard deviation as well. Given below is the input and the output.

```
summary_by_gender <- Cleaned_Data %>%
+     group_by(Gender) %>%
+     summarise(
+         Mean_GSESTotal = mean(GSESTotal...15, na.rm = TRUE),
+         SD_GSESTotal = sd(GSESTotal...15, na.rm = TRUE)
+     )
> print(summary_by_gender)

# A tibble: 2 × 3
  Gender Mean_GSESTotal    SD_GSESTotal
  <chr>           <dbl>           <dbl>
1 Female           28.4            3.45
2 Male             30.0            3.67
>
```

I think now with these commands we can carry out quite a lot of descriptive analysis. For instance, we can see whether the **GSES** score varies based on YoS and we can perform similar operations to **GAD7** as well. It is practically not possible to come up with brand-new commands for newer analyses if we are new to the *R* language. But, by following the above steps, we can utilize our understanding to generate newer codes through **Chat GPT**. Although we discourage students from using *Chat GPT* to write essays, it is a powerful tool when used for learning programming languages. However, make sure you give appropriate commands. Sometimes, the scripts of *Chat GPT* do not work on *R*. You should always indicate the package you are using, and at times you need to have a conversation with it to obtain the correct code. However, it is still a beneficial tool. Since I am new to programming, I generated all the above commands from *Chat GPT*. However, I have amended them according to the nature of my dataset and also my needs. Hence, I was able to compile a comprehensive set of codes that you could use directly to analyse your data. Further, when you generate codes, you should have sound understanding of statistical procedures to get the maximum benefit of it. Lack of understanding may hinder your chances of obtaining accurate codes to run analyses efficiently.

6.3 Histograms

Now that we have done some descriptive analyses on our continuous variables, let's draw a histogram for our **GSESTotal** variable. You may enter the below-given command. Please note that direct copying and pasting might not work, especially since I have a unique name for my GSESTotal variable. Instead of GSESTotal, it has named the same variable GSESTotal…15 when loading the *Excel* sheet to *R*. If you manually type them, *R* will prompt the variable list for you to select to complete the command.

```
> hist(Cleaned_Data$GSESTotal...15)
```

When you enter the above code, you should get a histogram like the one displayed in figure 10.

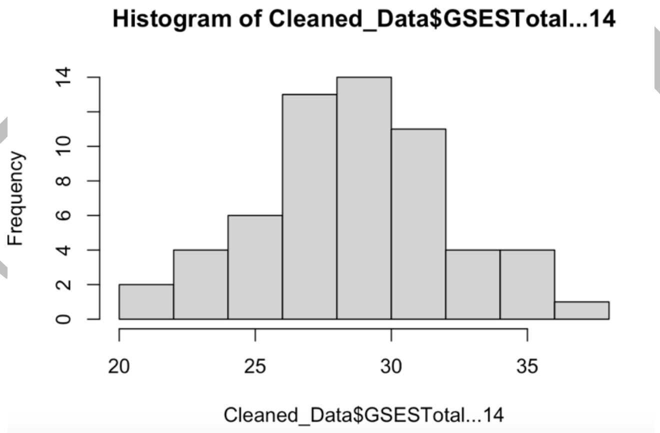


Figure 13: Histogram

A histogram is a good way to observe the distribution of data for a continuous variable. Further, this is a great method to observe the normality of the distribution. As per the descriptive statistics we obtained earlier, both skewness scores and kurtosis scores are closer to zero. So, we know our data is normally distributed. We can observe the same visually here. Just like how we added new commands to our bar charts, in this example also, we can add new labels. Take a moment to observe the below-given code and the resulting output.

```
> hist(Cleaned_Data$GSESTotal...15, col = "lightblue", main = "Histogram with Normality Curve")
```

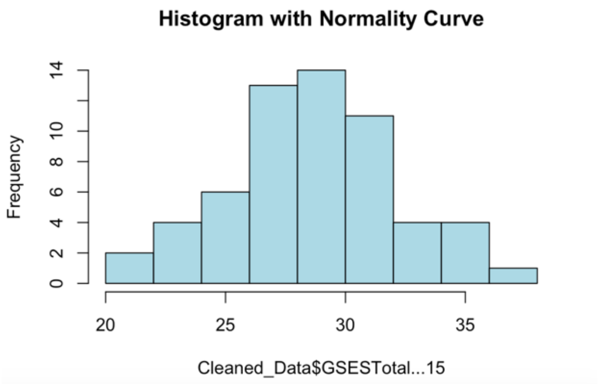


Figure 14: Histogram with new colors added

Unlike the previous histogram, in this new histogram, we have changed the title and also the color as well. If you want to add a new **x label**, you can simply add **xlab = “GSES**” to the last bit of the code. Thus, your new code will look like this. Similarly, the title is represented by **main = “Histogram with Normal Curve.”**

```
hist(Cleaned_Data$GSESTotal...15, col = "lightblue", main = "Histogram with Normal
Curve", xlab = “GSES”)
```

This code will make sure you have a much better histogram with a proper title, color, and labels. Let’s make this histogram white to make this *APA*-friendly.

```
hist(Cleaned_Data$GSESTotal...15, col = "white", main = "Histogram with Normality
Curve", xlab = "General Self Efficacy")
```

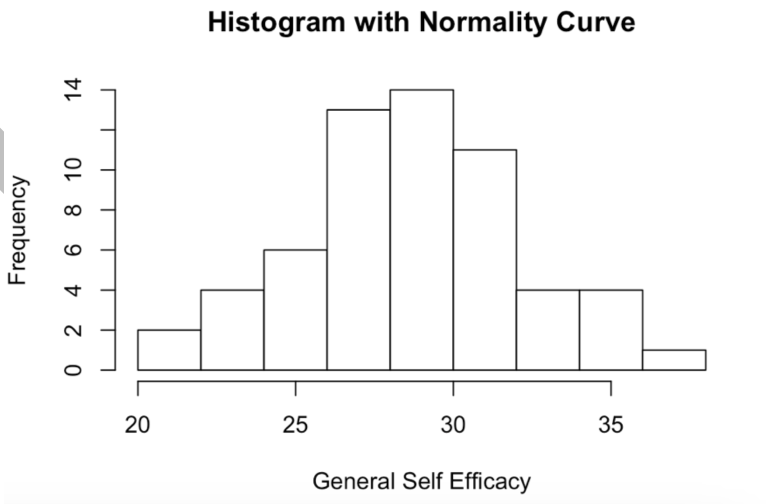


Figure 15: APA formatted histogram

In this example, I have not included the normality curve although I have named the label so. In the proceeding section, we will be discussing how to inspect the normality of a variable visually using a histogram and a normality curve. Before proceeding to the subsequent section, now would an excellent time to do a learning check.

Learning Check

8. Develop a code to obtain a bar chart for **RGender** in which bars are colored in light blue and the figure header states ‘Gender Comparison.’
9. Develop a code to obtain a bar chart for **YoS** in which bars are colored in light green and the figure header states ‘Type of Student.’
10. Explain the purpose of these operators: **$, <-** , **%>%** .
11. Obtain descriptive statistics for **GAD7** variable.
12. Generate a code to analyze data to obtain answer to question 5 from *ChatGPT* and compare the *ChatGPT* code and the one that you found from this book. Comment about similarities and differences you have noticed.

Answers available at the end of* **Part A

7.0 Normality Testing in R

You will notice, similar to running a bar chart, I just now ran a histogram for the GSES Total variable. Here, again, I have given the code and the output.

```
> hist(Cleaned_Data$GSESTotal...15, main = "Normality Testing for GSES", xlab =
"GSES")
```

Look at my output now.

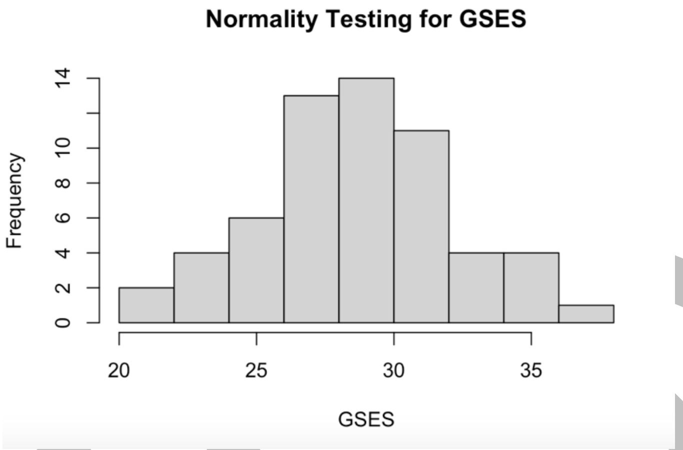


Figure 16: Histogram header and x labels included

Although generating a histogram is an easier job, developing the normality curve is a bit of trouble. This has 7 steps. Even if you use *ChatGPT* to generate your commands, it still takes multiple attempts to get them right. Here I have demonstrated how I did it. You can follow this step by step. If by any chance, you get error messages in one of the steps, a different system has to be used. However, with the dataset I have used, assuming you followed all the steps in this book as it is, this, R script should run well. Let's take a moment to review the steps associated with obtaining a histogram with a normality curve.

7.1 Histogram with normality curve

Step 1: Create the histogram

```
hist(Cleaned_Data$GSESTotal...15, col = "lightgray", main = "Histogram with Normal
Curve", xlab = “General Self Efficacy”)
```

Step 2: Calculate mean and standard deviation of the data. The operator **<-** should immediately indicate what we are doing here. We are assigning results to new variables.

```
mean_value <- mean(Cleaned_Data$GSESTotal...15)
sd_value <- sd(Cleaned_Data$GSESTotal...15)
```

Step 3: Create a sequence of **x** values for the curve

```
x <- seq(min(Cleaned_Data$GSESTotal...15), max(Cleaned_Data$GSESTotal...15), length =
100)
```

Step 4: Calculate the corresponding y values using **dnorm()** for the normal distribution. Previously defined variables of step 2 and 3 are included here in this code. Take note.

```
y <- dnorm(x, mean = mean_value, sd = sd_value)
```

Step 5: Determine the maximum count in the histogram

```
max_count <- max(hist(Cleaned_Data$GSESTotal...15, plot = FALSE)$counts)
```

Step 6: Scale the y values to fit the histogram, considering the maximum count

```
y <- y * max_count / max(y)
```

Step 7: Add the normal distribution curve to the histogram

```
lines(x, y, col = "red", lwd = 2)
```

If all your commands run smooth, this is your output.

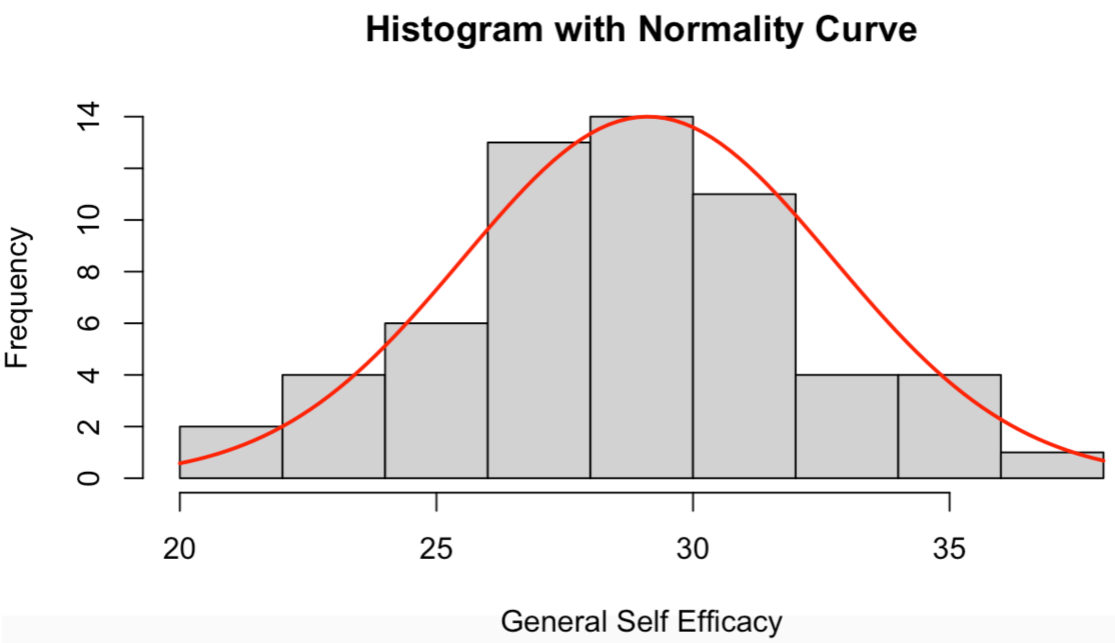


Figure 17: Histogram with normality curve

Now, you probably must already have felt that this script is way too much work to get the normality curve. So, let’s try to understand the logic here.

Let’s try to understand the logic of the earlier code.

Step 1: Create the histogram

```
hist(Cleaned_Data$GSESTotal...15, col = "lightgray", main = "Histogram with Normal
Curve", xlab = “General Self Efficacy”)
```

This is an easier step. We just created a histogram like we did in the previous section of this book. To make our histogram look better, we have given a color, title, and x label as well.

Step 2: Calculate the mean and standard deviation of the data

```
mean_value <- mean(Cleaned_Data$GSESTotal...15)
sd_value <- sd(Cleaned_Data$GSESTotal...15)
```

This step is also self-explanatory. We have found the mean and standard deviation of GSESTotal…15 variable and have assigned the results to two new variables named mean_value and sd_value.

Step 3: Create a sequence of **x** values for the curve

```
x <- seq(min(Cleaned_Data$GSESTotal...15), max(Cleaned_Data$GSESTotal...15), length = 100)
```

This step is very new to us. To draw a smooth curve, we need a bunch of x values that cover the range of our data. Think of this like creating a line of points on a graph. Now, you might wonder whether this length can be changed. Usually, a larger length offers a finely spaced sequence as opposed to a shorter length.

Step 4: Calculate the corresponding y values using **dnorm()** for the normal distribution

```
y <- dnorm(x, mean = mean_value, sd = sd_value)
```

In this step, we use **dnorm()** function to calculate y values for a normal curve. This curve helps us to see if our data looks like a bell-shaped curve.

Step 5: Determine the maximum count in the histogram

```
max_count <- max(hist(Cleaned_Data$GSESTotal...15, plot = FALSE)$counts)
```

In this step, we figure out how tall the tallest bar in our histogram is. This helps us make sure our curve fits nicely on top of the histogram.

Step 6: Scale the y values to fit the histogram, considering the maximum count

```
y <- y * max_count / max(y)
```

In this step, we adjust the height of our curve so it matches the histogram. It's like stretching or squeezing the curve to make it fit nicely on top of the bars.

Step 7: Add the normal distribution curve to the histogram

```
lines(x, y, col = "red", lwd = 2)
```

In this final step, we draw a red colored curve on top of the histogram. This is the curve that helps us to see if our data looks like a bell-shaped curve. The col = ‘red’ gives the color and 1wd = 2 gives the thickness of it.

Phoof!! I do not know about you guys, I already feel like a programmer.

7.2 Shapiro-Wilk for Normality Testing

Now that we know how to visually observe normality using histograms, let's see whether we could do the same using a test dedicated to investigating normality. There are two commonly used tests for this. One is the *Kolmogorov-Smirnov* test and the other is the *Shapiro-Wilk* test. Here, I have chosen *Shapiro Wilk* as this is a test ideal to be used when you have smaller samples (Mishra *et al.,* 2019). Also, *Shapiro Wilk* is a very sensitive test making it detect even slight deviations from normality (Mishra *et al.,* 2019). Finding normality is a simple process compared to what we did to generate normality curves. I have given both the code and the output for **GSES**.

```
> shapiro.test(Cleaned_Data$GSESTotal...14)

        Shapiro-Wilk normality test

data:  Cleaned_Data$GSESTotal...14
W = 0.98463, p-value = 0.6611
```

Well, can you interpret? Our variable is normally distributed as per the *Shapiro-Wilk* test (*p value less than 0.05 indicates violations of normality*). The same was confirmed during our visual inspection of normality using histograms. Usually, we encourage students to observe normality using multiple methods. You can observe normality descriptively through skewness/kurtosis scores, visually through histograms, *q-q plots*, and *p-p plots*, and lastly through statistical tests such as *Shapiro-Wilk*. In this study, **GSES** is normally distributed as per all of these three (Read more about testing for normality in **Section 7.3** Commentary on normality testing).

Let's do the same for our **GAD7** Total variable as well. I first decided to go for my descriptive analysis. I added the following code, and here is my output.

```
GAD_Score <- describe(Cleaned_Data$GAD7Total)
> print(GAD_Score)

  vars   n   mean   sd   median  trimmed  mad   min  max   range  skew  kurtosis   se
X1    1  59    7.53  3.61      7         7.37    2.97    0     23      23      1.1      3.89.    0.47
>
```

Based on the above findings, our data indicates both a noticeable skew and heavy tails indicating extreme values. Now, my next option is to go for a visual inspection to further investigate this. So, I decided to generate a histogram. Given below is my *R* command.

```
> hist(Cleaned_Data$GAD7Total, main = "Normality Testing for GAD", xlab = "GAD")
```

The output is given on the next page (Figure 18). Needless to say, the histogram looks very skewed. With the information I already have obtained using descriptive measures and visual inspection methods, I am pretty sure Shapiro-Wilk will say this variable is not normally distributed. But, let's check nevertheless.

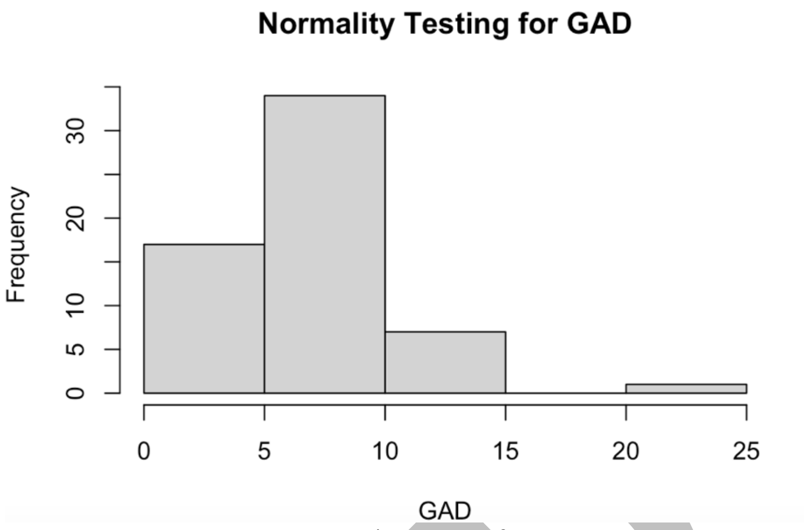


Figure 18: Histogram for GAD

So, this is the code for Shapiro Wilk and the output. Can you interpret it this time?

```
> shapiro.test(Cleaned_Data$GAD7Total)

        Shapiro-Wilk normality test

data:  Cleaned_Data$GAD7Total
W = 0.91862, p-value = 0.0007543
```

Similar to the previous two methods, Shapiro-Wilk also confirms GAD7 to be not normally distributed (*the resulting p value of the test is less than 0.05*). Thus, in our study, one of our variables is normally distributed, and one is not. Now, I have more decision-making to do before proceeding with inferential statistical measures.

I further decided to go ahead and inspect the histogram based on gender categories. This requires a different code. I have given it below with the output. Can you see whether you could try to interpret the code?

```
hist(Cleaned_Data$GAD7Total[Cleaned_Data$Gender=="Male"], main = "Histogram for
Males", col = "lightblue", xlab = "GAD7")
```

I performed the same for females by slightly changing my code. I have given my code below.

```
hist(Cleaned_Data$GAD7Total[Cleaned_Data$Gender=="Female"], main = "Histogram for
Females", col = "lightblue", xlab = "GAD7")
```

Both the outputs of the above codes are demonstrated on the next page. Based on the obtained results, either men have no anxiety, or women tend to be more truthful in their answers. There is a clear gender-related difference here. Such differences can further be observed through inferential statistical measures such as T-Tests. However, that is not the objective of this book. I will make sure to include them in a later version of this book. For now, I am satisfied with these descriptive analyses.

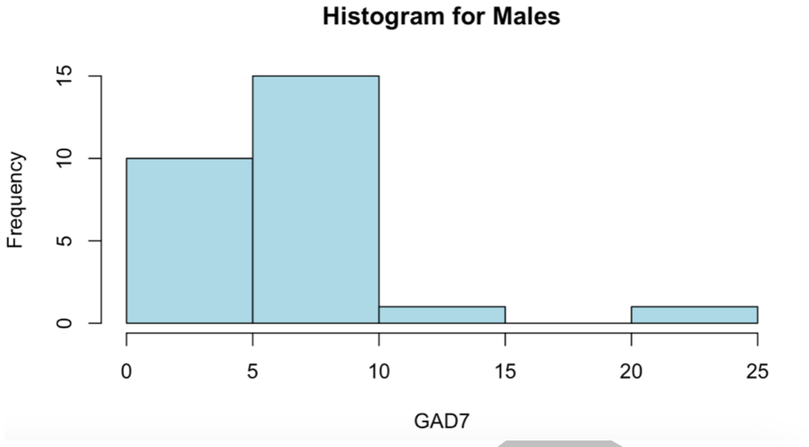


Figure 19: Histogram for GAD7 (Males)

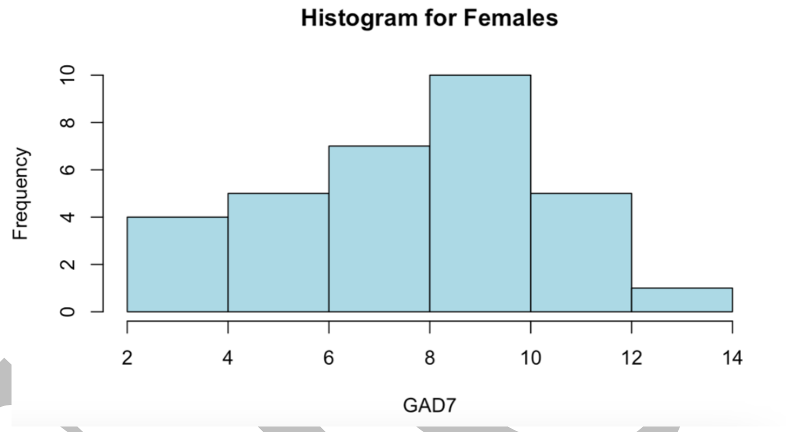


Figure 20: Histogram for GAD7 (Females)

I think the lesson on normality is the biggest lesson in **Part A** of this book. However, I think now all of you have a good understanding of normality testing. Now, take a moment to recall the study I have included in section 1.1 (page 2). Our intention was to investigate the relationship between general self-efficacy and anxiety. Let's start with that, shall we?

7.3 Commentary on normality testing

"*Sir, why do we need to meet assumptions like normality for statistical analyses?*" This is a question I commonly get in all my first year undergraduate research methods classes. I often reply saying, "*Well, I want ensure my sample accurately represents the population of interest*." Although I say this answer, my real intentions are a bit complicated. In statistics, we use both parametric tests (i.e., *T-Tests*, *ANOVA*, etc.) and non-parametric tests (i.e., *Wilcoxon Test*, *Kruskal Wallis Test*, etc.). Normality is a parametric test assumption in which the validity of the test is based upon. There are more reasons to consider normality beyond just enhancing the validity of a test; but, those explanations are a little too much right now. We shall re-visit them in later editions of this book when we discuss about regression analyses, *T-Tests*, and *ANOVA*s. However, in this edition, I will comment about normality and ways to evaluate it in a general sense.

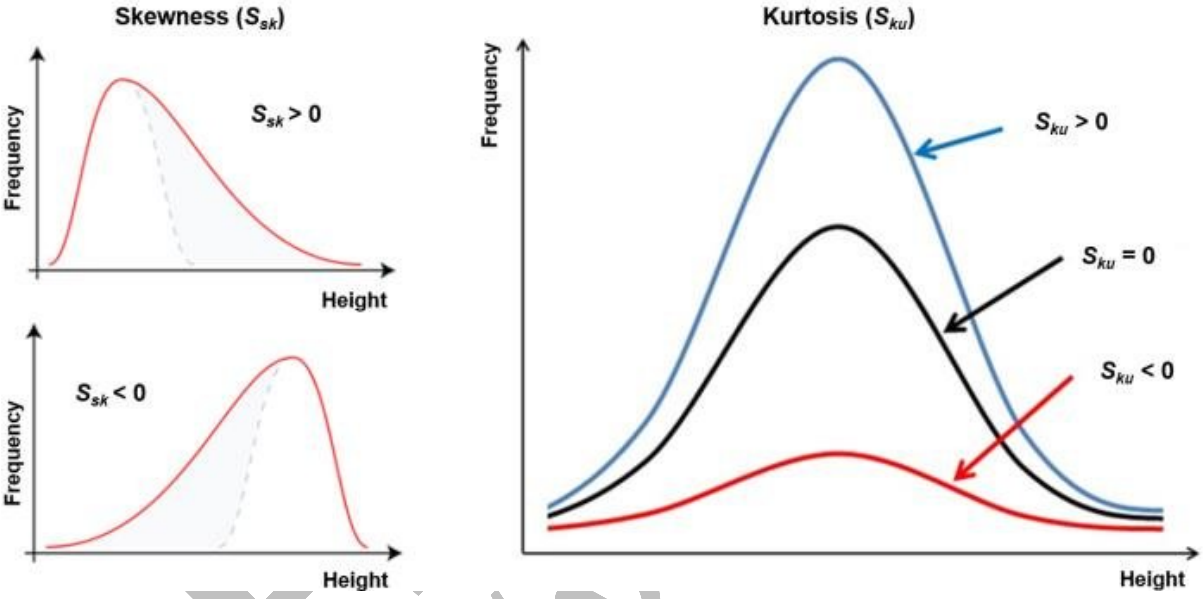


Figure 21: Skewness and kurtosis

Normality can be tested using 3 major methods: descriptively, visually, and using specialized tests (Mishra *et al.,* 2019; Singh & Masuku, 2014). When we test normality descriptively and visually, generally, we look for skewness and kurtosis. Skewness helps us to notice whether our data is inclined towards one side leaving a tail to the opposite side. Kurtosis checks how narrow or flat the data is. Usually values closer to zero are preferred for both skewness and kurtosis when we check for normality. Negative skewness scores indicate a tail to the left while positive scores indicate a tail to the right (DeCarlo, 1997). Similarly, positive values for kurtosis indicate distributions to be tall and narrow (**leptokurtic**) while flatter, more spread distributions (**platykurtic**) are associated with negative values (DeCarlo, 1997). Normal distributions are called **mesokurtic** (values positive or negative closer to zero). Skewness and kurtosis can also be visually observed by obtaining histograms with normality curves. Lastly, we can also check normality using a statistical test like *Shapiro-Wilk.* If the resulting *p* value is not less than 0.05, then normality assumption is met. We usually advice students to utilize all these 3 methods (descriptive, visual, and tests) to evaluate normality before deciding on a inferential statistical test. This should also be decided along with appropriate justifications from pertinent literature sources.

I think this commentary helps you to scaffold your previous learning to enhance your chances of application to analyse real-datasets in future. Now, if you are ready, spend some more time to do the activities in the learning check, so that we could move to the final section of the **Part A** of this book.

Learning Check

13. Obtain 3 histograms for **GSESTotal** based on the categories of **YoS**.
14. Obtain Shapiro-Wilk output for **GSESTotal** score based on the categories of **YoS**.

Answers available at the end of* **Part A

8.0 Running a Correlational Analysis in R

We have slowly approached the final lesson in the **Part A** of this book. If you recall our study from **Section 1.1**, we are trying to find a relationship between self-efficacy and general anxiety. First, let’s develop our hypothesis.

H_1: Self-efficacy and anxiety share a statistically significant relationship

Since this book is intended for undergraduate students, I decided to first investigate this hypothesis through a correlational analysis. In later editions of this book, I will analyse this dataset through regression analyses. I will also further introduce you to the application of *T-Tests* and *ANOVA*s to test similar hypotheses using the same dataset. However, for now, we are focusing on correlations as we are still at the earlier stages of learning *R* and suddenly approaching complex analyses will derail us from our purpose. However, you will notice that I have included some difficult lessons in **Part B** and **C**. You will be quite ready to proceed with more inferential statistical testing once this edition of the book is completed.

About correlations, there are both parametric versions (*Pearson correlation*) and non-parametric versions (*Spearman correlation*). Pearson requires the assumption of normality, and two continuous variables to proceed with analysis. In the event of normality violation, Spearman correlation can be considered. Based on the previous analyses we have performed, we know one of our variables violate the assumption of normality. As a result, I am thinking of using Spearman correlation to investigate this relationship to asses **H_1**. Surprisingly running a correlation test is far easier compared to generating histograms and normality curves. Here is the code and the output.

```
> cor.test(Cleaned_Data$GSESTotal...15, Cleaned_Data$GAD7Total, method = "spearman")

        Spearman's rank correlation rho

data:  Cleaned_Data$GSESTotal...15 and Cleaned_Data$GAD7Total
S = 49420, p-value = 0.000425
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.4441752
```

I am not going to explain the code as it is self-explanatory. By now we have learned multiple functions and operators. So, by now you should be able to dissect these codes and understand them. Based on the output, self-efficacy, and anxiety share a statistically significant negative relationship, r_s = -.444, p = 0.000425. You can further proceed to obtain the effect size (coefficient of determination, r^2) by simply squaring the correlation coefficient. In the present example, r^2 = .197.

To further make things interesting, I also ran a correlational analysis separately for men and women. Take note of how I have changed my R commands. First I have given the analysis for men, and then women.

Analysis for men

```
> cor.test(Cleaned_Data$GSESTotal...15[Cleaned_Data$Gender == "Male"],
Cleaned_Data$GAD7Total[Cleaned_Data$Gender == "Male"], method = "spearman")

        Spearman's rank correlation rho

data:  Cleaned_Data$GSESTotal...15[Cleaned_Data$Gender == "Male"] and
Cleaned_Data$GAD7Total[Cleaned_Data$Gender == "Male"]
S = 4863.5, p-value = 0.01042
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.4845797
```

Based on the output, self-efficacy, and anxiety for males share a statistically significant negative relationship, r_s = -.4845797, p = 0.01042.

Analysis for women

```
> cor.test(Cleaned_Data$GSESTotal...15[Cleaned_Data$Gender == "Female"],
Cleaned_Data$GAD7Total[Cleaned_Data$Gender == "Female"], method = "spearman")

        Spearman's rank correlation rho

data:  Cleaned_Data$GSESTotal...15[Cleaned_Data$Gender == "Female"] and
Cleaned_Data$GAD7Total[Cleaned_Data$Gender == "Female"]
S = 7528.9, p-value = 0.03197
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.3799251
```

Based on the output, self-efficacy, and anxiety for males share a statistically significant negative relationship, r_s = -.3799251, p = 0.03197. It is interesting to note that females have a lower correlational coefficient compared to men. This could be due to the fact that women have more score spread for anxiety as opposed to men (*Figure 19 and Figure 20*).

8.1 Drawing the Scatter Plot and the Best-Fit Line

To further complete the analysis, I also drew the scatter plot and the best-fit line. I have given the input and output below for your reference. A negative correlation is clear.

```
> plot(Cleaned_Data$GSESTotal...15, Cleaned_Data$GAD7Total,
+      xlab = "GSESTotal...15", ylab = "GAD7Total",
+      main = "Scatter Plot")
> fit <- lm(Cleaned_Data$GAD7Total ~ Cleaned_Data$GSESTotal...15)
> abline(fit, col = "blue")
>
```

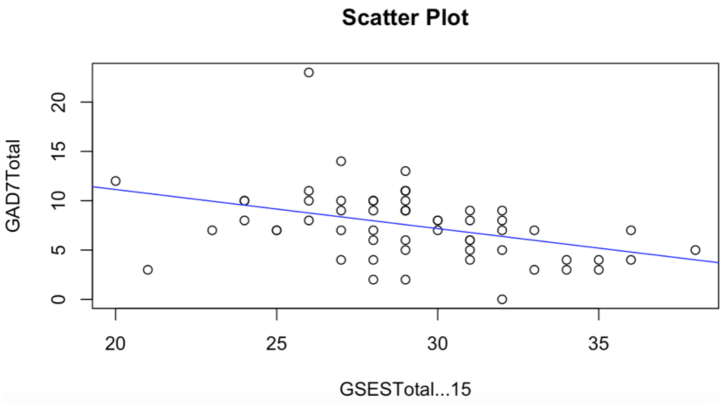


Figure 22: Scatter plot and best-fit line for GSES and GAD7

Spend some time to understand the overall code I have produced above. Please note that '+' is an operator added to indicate the argument provided for **plot()** continued in the subsequent lines. Here, the **lm()** stands for linear model. Essentially, what we display is a linear relationship.

8.2 Commentary on correlations

In the previous outputs, I mentioned normality as one of the assumptions to consider between Pearson and Spearman. However, there are a few other assumptions to consider when deciding the ideal correlation to analyze your data. I have summarized key assumptions in the table given below.

Table 2: Types of correlations and assumptions (Gravetter & Wallnau, 2017)

Type of Correlation	Assumptions
Pearson correlation (r)	Two linearly related, normally distributed, continuous variables.
Spearman correlation (r_s)	Two variables of which one could be ordinal. Normality is not expected.
Point biserial correlation (r_{pb})	Two variables of which one is nominal.
Phi-coefficient (ϕ)	Two binary variables (refer definitions)

One of the easiest methods to observe linearity is to obtain a scatter plot. The plotted data typically demonstrate an upward trend from or a downward trend from left to write in the instance of the existence of a linear relationship (refer to the example in Figure 23). Similarly, the type of measurement should be decided based on the type of data that you have collected. In the event of psychometric scales, if you obtain a numeric output, it should be a ratio scale. However, it is always best to download key papers of the respective scales to further understand the nature of its output. Although many scales provide a final score, some scales provide categorical output (i.e., low depression, moderate depression, severe depression). In such instances, you end up having a categorical variable. If it is ordinal, you may proceed with Spearman correlation as long as you have another continuous variable. But, if one of your variables is binary (i.e., depression, no depression) with a continuous variable, you will have to consider a point-biserial correlation. In the rare instance of having a two-binary variable (depression, no depression vs. urban, rural), you will have to use the phi-coefficient.

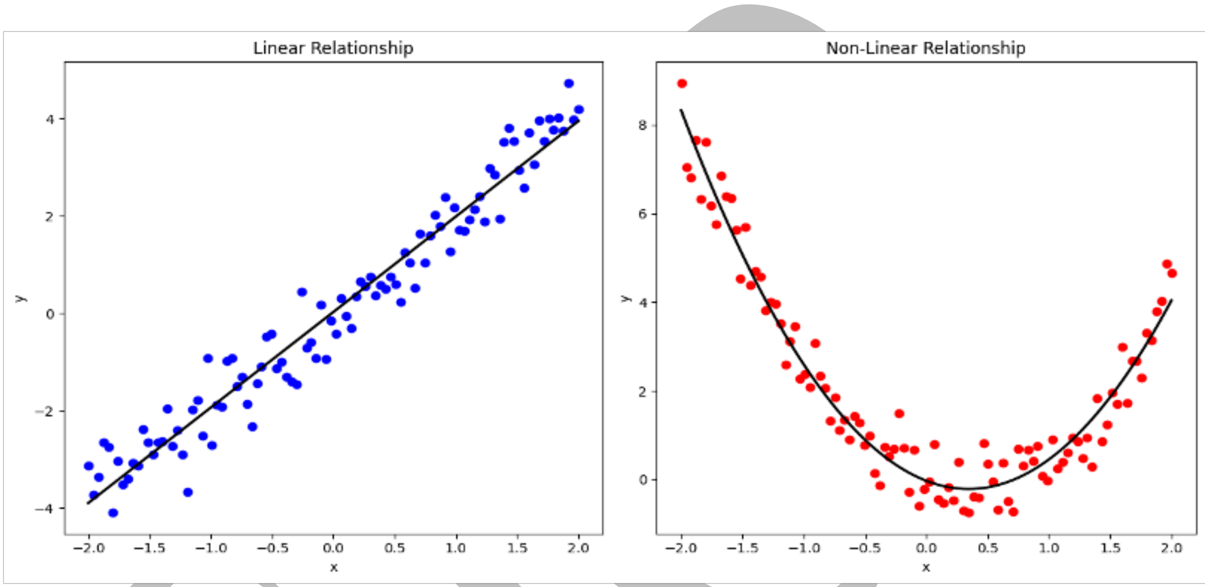


Figure 23: Linear and non-linear relationships

Because of the considerable variation in the types of correlations, it is advised for students to spend considerable time to evaluate assumptions prior to selecting the right type of correlational analysis for a specific dataset.

Table 3: Types of correlations and assumptions (Gravetter & Wallnau, 2017)

Assumption	**(*r*)**	**(*r_s*)**	**(*r_{pb}*)**	**(*ϕ*)**
Normality	✓			
Linearity	✓			
Binary variable (X)			✓	✓
Ordinal variable (X)		✓		
Continuous variables (X & Y)	✓	✓		
Continuous variable (Y)		✓	✓	
Binary variable (Y)				✓

The table 3 given above will help you further decide the type of correlation you would require when making subsequent considerations for your studies.

Now take a moment to check the YouTube video that summarizes our learnings of **Part A**.

Learning Check

15. What is the correlation coefficient (Spearman & Pearson) for **GSESTotal** & **GAD7Total** for all categories of **YoS**?
16. Draw a scatterplot for each output of the previous question.

Answers available at the end of* **Part A

Answers

The questions posted in each of the previous sections of **Part A** are addressed below as succinctly as possible.

1. Is R a programming language?

   Yes. *R* is a programming language designed for statistical computing and graphics.

2. List all the variables given in the Excel data file (*Cleaned_data*).

   Gender, Age, Year of Study, GSES, GAD7

3. Compared to *SPSS* and *JAMOVI*, what are the differences you noticed in *R* Studio?

   R Studio is more flexible, and cost-effective, but often entails a steeper learning curve. *SPSS* tends to be expensive, but relatively easy to use. *Jamovi* is further easier to use and can be further programmed through *R*.

4. What is an *R* package?

   A package is a collection of functions, datasets, documentation, and sometimes compiled code in a well-defined format that enhances the functionality of *R*.

5. Why should researchers spend time cleaning data before transferring them to *R* Studio?

   Clean data improves the overall quality of data and subsequent analyses. Researchers clean the data to identify variables which could improve debugging if the codes entered are faulty.

6. As per your understanding, what would be a good package to analyze large datasets?

   dplyr

7. The name of the dataset is **SGSESData**, and the categorical variable that requires a frequency calculation is **BirthOrder**. Write a code to create a new variable named **BO** that could help us generate frequencies for **BirthOrder** later.

   ```
   BO <- table(BirthOrder)
   ```

8. Develop a code to obtain a bar chart for **RGender** in which bars are colored in light blue and the figure header states ‘Gender Comparison.’

   ```
   RGender <- table(Cleaned_Data$Gender)
   barplot(RGender, col = "lightblue", main = "Gender Comparison")
   ```

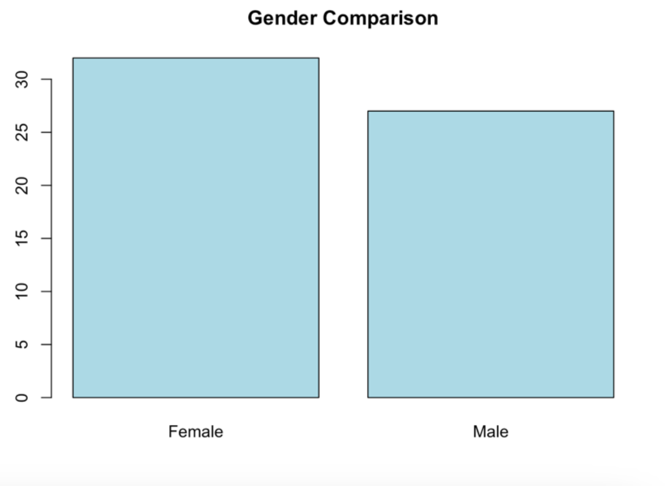


Figure A: Gender comparison of the study

9. Develop a code to obtain a bar chart for **YoS** in which bars are colored in light green and the figure header states 'Type of Student.'

```
YoS <- table(Cleaned_Data$`Year of Study`)
barplot(YoS, col = "lightgreen", main = "Type of Student")
```

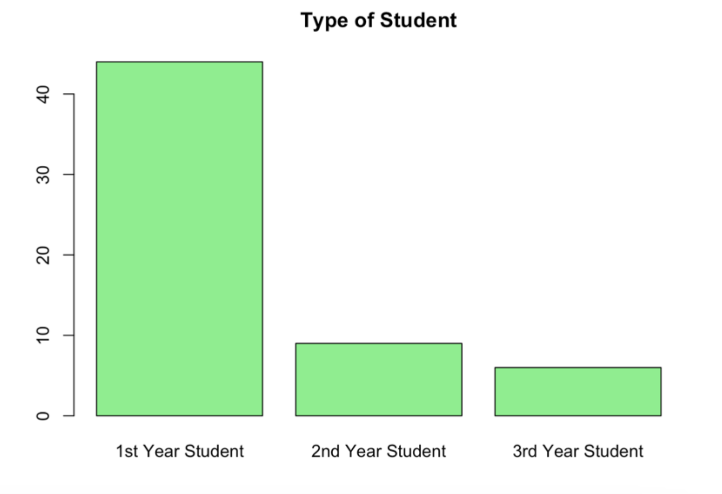


Figure B: Distribution of students based on the year of study

10. Explain the purpose of these operators: **$, <- , %>%** .

$** is used to indicate a variable belonging to a specific dataset (e.g., Cleaned_Data$GAD7Total), **<- refers to results assigned to a variable, **%>%** is a pipe operator.

11. Obtain descriptive statistics for the **GAD7** variable.

```
pacman::p_load(pacman, dplyr, psych)
summary_stats <- describe(Cleaned_Data$GAD7Total)
print(summary_stats)
```

```
   vars  n  mean  sd  median  trimmed  mad  min  max  range  skew  kurtosis  se
X1    1  59  7.53  3.61    7      7.37  2.97   0   23     23   1.1      3.89  0.47
```

12. Generate a code to analyze data to answer question 7 from ChatGPT and compare the ChatGPT code and the one you found in this book. Comment about similarities and differences you have noticed.

My Code	ChatGPT Code
BO <- table(BirthOrder)	

13. Obtain 3 histograms for **GAD7Total** based on the categories of **YoS**.

```
hist(Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "1st Year
Student"], main = "Histogram for 1st Year Students", col = "lightblue", xlab =
"GAD7")
```

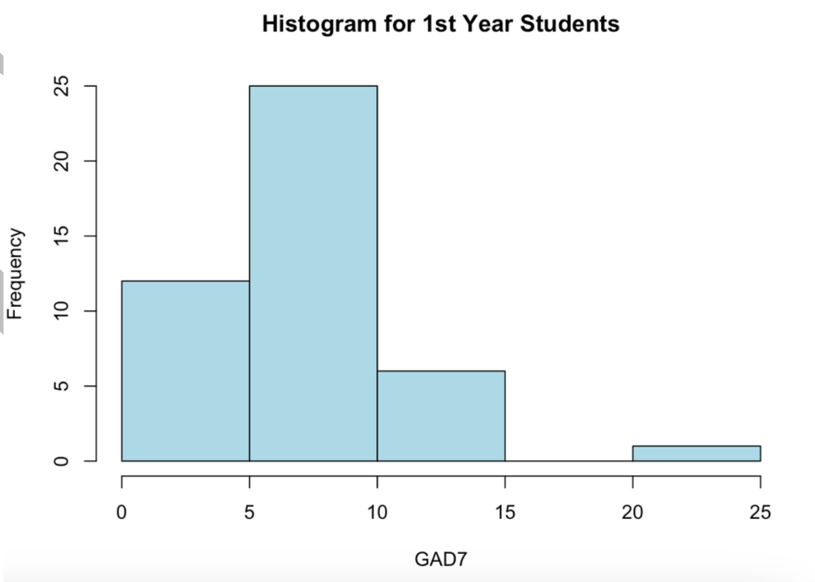


Figure C: Anxiety scores distribution for 1st year students

```
hist(Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "2nd Year
Student"], main = "Histogram for 2nd Year Students", col = "lightblue", xlab =
"GAD7")
```

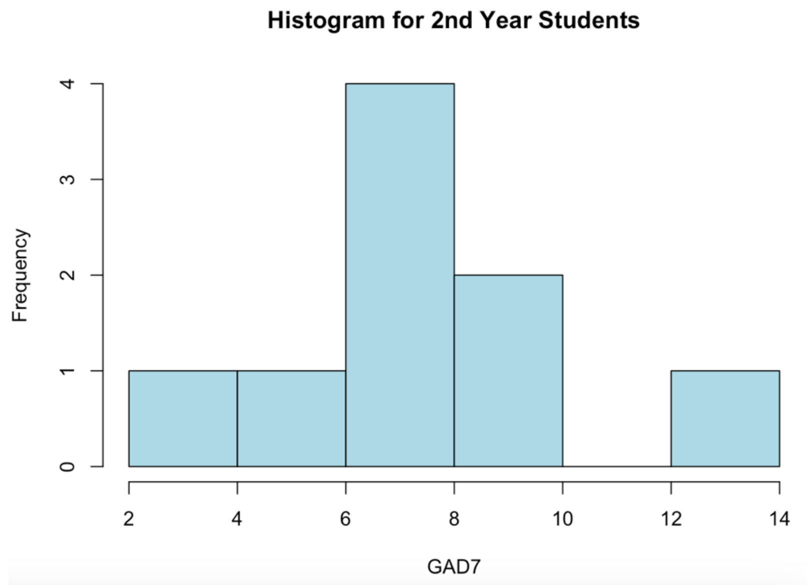


Figure D: Anxiety scores distribution for 2nd year students

```
hist(Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "3rd Year
Student"], main = "Histogram for 3rd Year Students", col = "lightblue", xlab =
"GAD7")
```

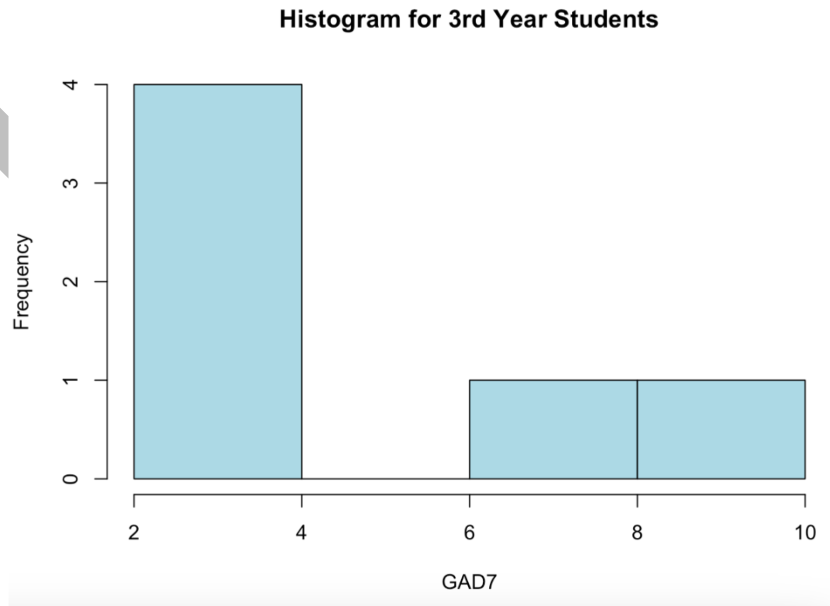


Figure E: Anxiety scores distribution for 3rd year students

14. Obtain Shapiro-Wilk output for **GAD7Total** score based on the categories of **YoS**.

Shapiro-Wilk for GAD7 (1st Year Students)

```
shapiro.test(Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "1st Year Student"])
```

```
	Shapiro-Wilk normality test

data:  Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "1st Year Student"]
W = 0.89969, p-value = 0.001065
```

Shapiro-Wilk for GAD7 (2nd Year Students)

```
shapiro.test(Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "2nd Year Student"])
```

```
	Shapiro-Wilk normality test

data:  Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "2nd Year Student"]
W = 0.924, p-value = 0.4264
```

Shapiro-Wilk for GAD7 (3rd Year Students)

```
shapiro.test(Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "3rd Year Student"])
```

```
	Shapiro-Wilk normality test

data:  Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "3rd Year Student"]
W = 0.89251, p-value = 0.3316
```

15. What is the correlation coefficient (Spearman & Pearson) for **GSESTotal** & **GAD7Total** for all categories of **YoS**?

Pearson Correlation between GSES & GAD7 (1st Year Students)

```
cor.test(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "1st Year Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "1st Year Student"], method = "pearson")
```

```
Pearson's product-moment correlation
```

```
data:  Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study` == "1st Year
Student"] and Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "1st
Year Student"]
t = -2.773, df = 42, p-value = 0.008248
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6180783 -0.1092616
sample estimates:
       cor
-0.3933816
```

Spearman Correlation between GSES & GAD7 (1st Year Students)

```
cor.test(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "1st Year
Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "1st Year
Student"], method = "spearman")

        Spearman's rank correlation rho

data:  Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study` == "1st Year
Student"] and Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "1st
Year Student"]
S = 20579, p-value = 0.002165
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.4502374
```

Spearman Correlation between GSES & GAD7 (2nd Year Students)

```
cor.test(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "2nd Year
Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "2nd Year
Student"], method = "spearman")

Spearman's rank correlation rho

data:  Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study` == "2nd Year
Student"] and Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "2nd
Year Student"]
S = 117.93, p-value = 0.9649
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
0.01724202
```

Pearson Correlation between GSES & GAD7 (2nd Year Students)

```
cor.test(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "2nd Year
Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "2nd Year
Student"], method = "pearson")
```

```
Pearson's product-moment correlation

data:  Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study` == "2nd Year
Student"] and Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "2nd
Year Student"]
t = 0.26508, df = 7, p-value = 0.7986
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6044490  0.7163835
sample estimates:
       cor
0.09969179
```

Pearson Correlation between GSES & GAD7 (3rd Year Students)

```
cor.test(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "3rd Year
Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "3rd Year
Student"], method = "pearson")
```

```
        Pearson's product-moment correlation

data:  Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study` == "3rd Year
Student"] and Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "3rd
Year Student"]
t = -1.5077, df = 4, p-value = 0.2061
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.9496103  0.4098010
sample estimates:
       cor
-0.6019589
```

Spearman Correlation between GSES & GAD7 (3rd Year Students)

```
cor.test(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "3rd Year
Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "3rd Year
Student"], method = "spearman")
```

```
        Spearman's rank correlation rho
```

```
data:  Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study` == "3rd Year
Student"] and Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study` == "3rd
Year Student"]
S = 53.529, p-value = 0.2801
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.5294118
```

16. Draw a scatterplot for each output of the previous question.

```
plot(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "1st Year
Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "1st Year
Student"], xlab = "GSES", ylab = "GAD7", main = "Scatter Plot")

fit <- lm(Cleaned_Data$GAD7Total ~ Cleaned_Data$GSESTotal)
abline(fit, col = "blue")
```

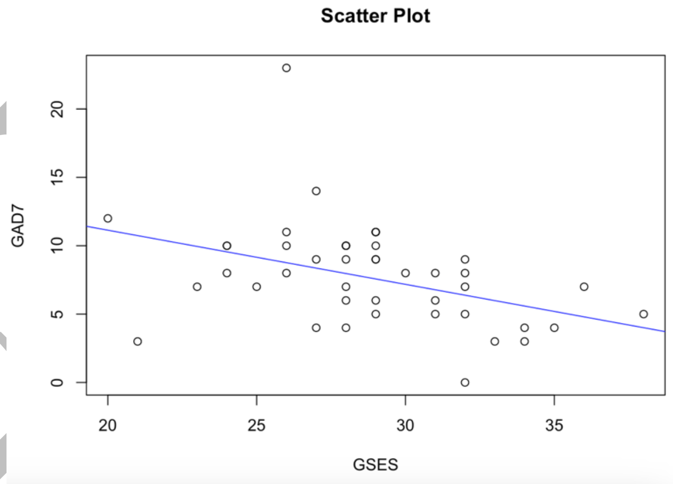


Figure F: Scatter plot and best fit line for GSES and GAD of 1st year students

```
plot(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "2nd Year
Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "2nd Year
Student"], xlab = "GSES", ylab = "GAD7", main = "Scatter Plot")
fit <- lm(Cleaned_Data$GAD7Total ~ Cleaned_Data$GSESTotal)
abline(fit, col = "blue")
```

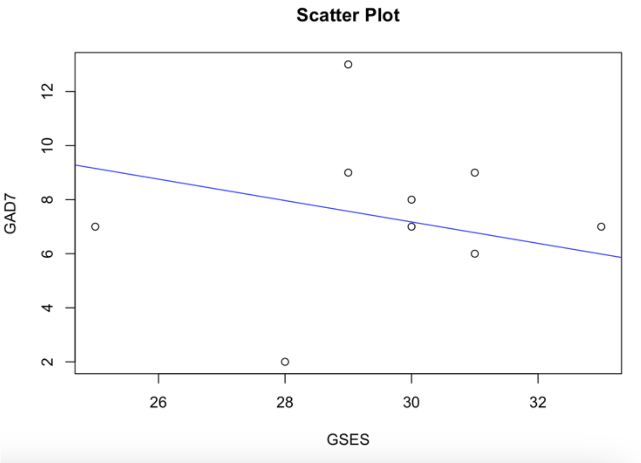


Figure G: Scatter plot and best fit line for GSES and GAD of 2nd year students

```
plot(Cleaned_Data$GSESTotal[Cleaned_Data$`Year of Study`== "3rd Year
Student"], Cleaned_Data$GAD7Total[Cleaned_Data$`Year of Study`== "3rd Year
Student"], xlab = "GSES", ylab = "GAD7", main = "Scatter Plot")
fit <- lm(Cleaned_Data$GAD7Total ~ Cleaned_Data$GSESTotal)
abline(fit, col = "blue")
```

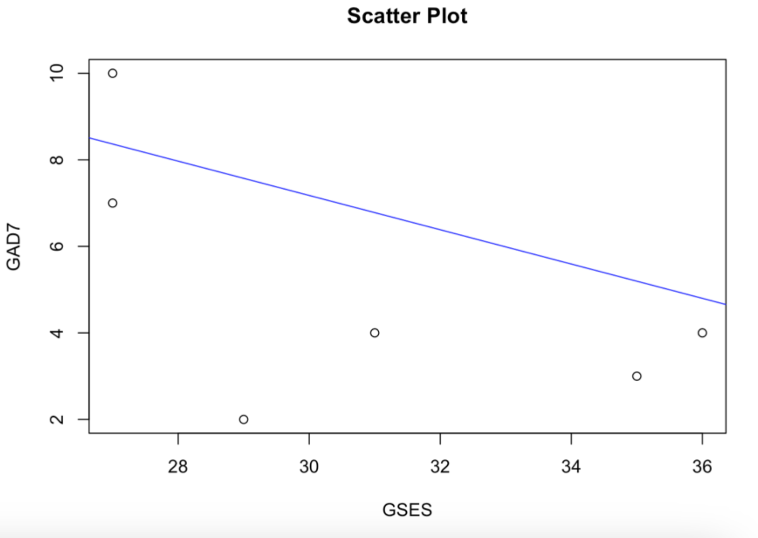


Figure H: Scatter plot and best fit line for GSES and GAD of 3rd year students

End of Answers

9.0 Concluding Remarks

Wow, what a relief!!!!

We must all be proud of ourselves. We have now mastered the basics of a new programming language. It is my understanding that this book helps you see *R* as a valuable software for statistical analyses. Originally, I started to write this book to help my first-year students develop key skills in *R* Studio to help them run simple studies that require nothing more than a correlation. One might argue that all of these can easily be done using *SPSS*. That is correct, however, *SPSS*'s functionality is far simpler compared to the greater benefits *R* offers. In that sense, learning this will help students, especially during their graduate programs. Also, in this edition of the book, I have included **Part B** and **C** to further help my second year students to be benefitted from this book.

Another bigger problem I always felt using *SPSS* was the readymade nature of *SPSS* output. Compared to that, *R* offers me great flexibility and customization. In that sense, *R* is an excellent alternative everyone should master. I also believe that psychology graduates should possess marketable skills when they enter the job market. I think knowing a little bit of programming for statistical computing opens up a new set of avenues to succeed in life.

Apart from testing your skills using the learning check questions, you can also view the video tutorials to further fine-tune your programming skills. Towards the end of this book, instructions are provided for students to revisit all the lessons in this book in the video format. In my opinion combining multiple methods could certainly augment the overall learning experience.

Now, with all the content we have learned, it is now time for us to go to **Part B** of this book. This subsequent section will be slightly harder compared to **Part A**. The next couple of sections contain a step by step approach to reliability and validity testing. Most of the analyses in reliability and validity testing are powered through correlations. As a result, it fits well if we can focus on learning the fundamentals of assessing reliability and validity of scales in the next two section. Often students say learning the technique involved in statistical validation of a scale is a fun activity. So, I think, you will also enjoy the lessons to come in the next sections.

This page is intentionally left blank

Part B
10.0 Classical Test Theory (CTT)

Now that we have mastered the skills of descriptive analysis and correlational analysis in *R Studio*, we are all set to learn about CTT. In my humble opinion, it will be a futile effort to write this book if I do not help readers understand the key concepts of CTT. In some sense, learning CTT is the first step you should take to venture into the world of psychometrics. In this section, I will explain the basics of CTT with a simple 10-item scale. Browsing through these will help you master the *R* programming skills we are about to discuss. Similarly, the basics of correlations you already have learned are of great importance to fully grasp the concepts in CTT.

10.1 Reliability

There are two major concepts you should learn to make sense of CTT: reliability and validity. The former helps researchers establish consistency while the latter assists in ensuring the accuracy of psychometric scales (Masuwai *et al.,* 2024). Before we move forward, let's understand some of the basic types of reliability. These are all powered by correlations.

Table 4: Types of reliability

Form of reliability	Type	Description
Internal consistency	Split half reliability	A correlation between two halves or parts of a scale or a test. For example, for a scale with 10 items, split half can be calculated by obtaining the correlation coefficient between two halves of a test. This could be odd numbered items and even numbered items in a test. You may consider either the average score or the sum of the considered item for this correlation. This could also be two different parts of the test (Pronk *et al.,* 2022). Here, both halves considered should have items that measures the same construct (check definition). Generally, this method of correlation is ideal when there are larger number of items. Higher correlations indicate, greater reliability.
	Cronbach's alpha (α)	Since all the items within a specific unidimensional (check definition) scale measure the same construct, they should indicate a greater deal of inter-relatedness. Internal consistency refers to this inter-relatedness (Tavakol & Dennick, 2011). Alpha tends to be higher with increasing items. α larger than 0.6 or 0.7 are generally considered a good indication of internal consistency reliability.

		α is also a form of correlation, thus, the outcome can be squared and subtracting from 1 would produce the index of measurement error. For example a scale that has α = .80, has 1 - (.80*.80) = 0.36 error variance (check definition) or random error in the scores. Thus, for this error to go down, the reliability coefficient should go higher (Tavakol & Dennick, 2011).
Stability	Test re-test reliability	A correlation between scores of a group of participants who have attempted a scale in two different times (i.e., correlation of IQ scores of last month with last week for 30 students). This is an ideal method for stable constructs. Correlation output between 0.4 to 0.74 are considered good and 0.75 and above are considered excellent (Matheson, 2019).
Equivalence	Parallel forms reliability	A correlation between two forms of a scale administered to a group of people (Babu & Kohli, 2023). Two forms would mean the original scale and a scale similar to the original scale (i.e., with slightly different wordings). Higher correlation between them suggests high reliability.

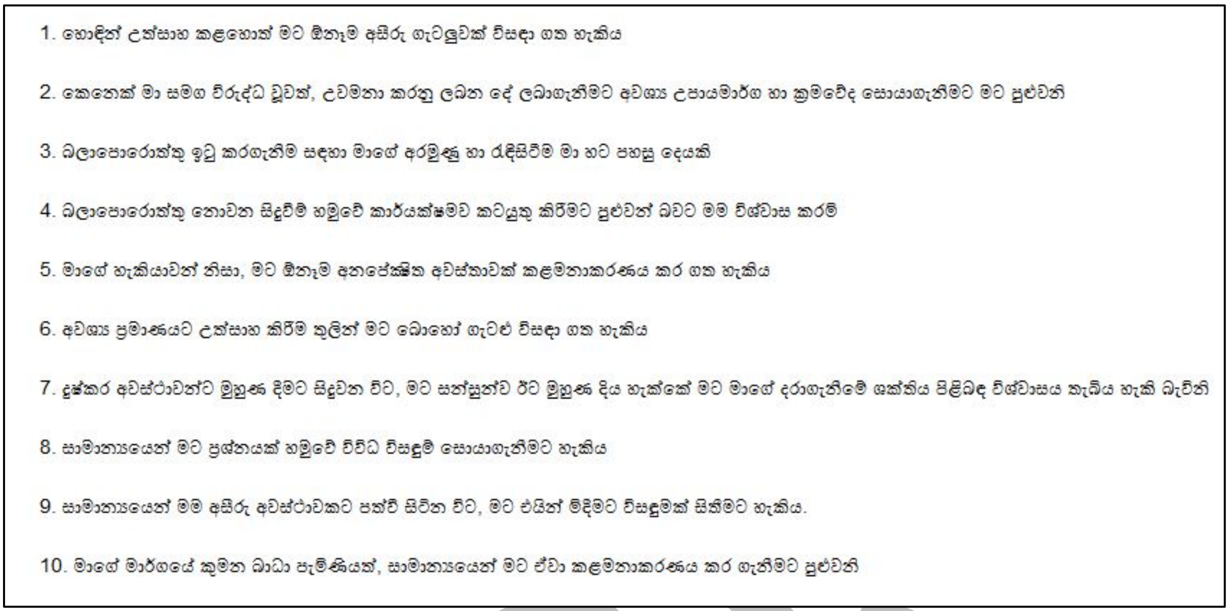
Since we know key types of reliability, let's perform reliability analyses for a chosen scale. The scale I have chosen is given below (the generalized self-efficacy scale). Throughout this section from this point onwards consider the below-given scale for analysis. This is a scale I have translated and adapted to Sri Lanka. This is the same self-efficacy scale we have used in **Part A** of this book as well. This scale has 10 items and the translated scale is given below for your reference. Each item in this scale has 4 answer options, 1 = *Not at all true*, 2 = *Hardly true*, 3 = *Moderately true*, and 4 = *Exactly true*. In a study I carried out with my colleagues in 2018, we found the scale to be **unidimensional** (see definitions) and highly reliable (α = 0.86).

Generalized self-efficacy scale (GSES) – Scholz *et al.,* 2002; Schwarzer *et al.* 1995

1. I can always manage to solve difficult problems if I try hard enough.
2. If someone opposes me, I can find the means and ways to get what I want.
3. I am certain that I can accomplish my goals.
4. I am confident that I could deal efficiently with unexpected events.
5. Thanks to my resourcefulness, I can handle unforeseen situations.
6. I can solve most problems if I invest the necessary effort.
7. I can remain calm when facing difficulties because I can rely on my coping abilities.
8. When I am confronted with a problem, I can find several solutions.
9. If I am in trouble, I can think of a good solution.
10. I can handle whatever comes my way.

Response format: 1 = Not at all true, 2 = Hardly true, 3 = Moderately true, 4 = Exactly true

Sinhala Generalized Self-Efficacy Scale (SGSES) – Selvaratnam et al. 2018



1. හොඳින් උත්සාහ කළහොත් මට ඕනෑම අසීරු ගැටලුවක් විසඳා ගත හැකිය

2. කෙනෙක් මා සමග විරුද්ධ වූවත්, උවමනා කරනු ලබන දේ ලබාගැනීමට අවශ්‍ය උපායමාර්ග හා ක්‍රමවේද සොයාගැනීමට මට පුළුවනි

3. බලාපොරොත්තු ඉටු කරගැනීම සඳහා මාගේ අරමුණු හා රැඳිසිටීම මා හට පහසු දෙයකි

4. බලාපොරොත්තු නොවන සිදුවීම් හමුවේ කාර්යක්ෂමව කටයුතු කිරීමට පුළුවන් බවට මම විශ්වාස කරමි

5. මාගේ හැකියාවන් නිසා, මට ඕනෑම අනපේක්ෂිත අවස්තාවක් කළමනාකරණය කර ගත හැකිය

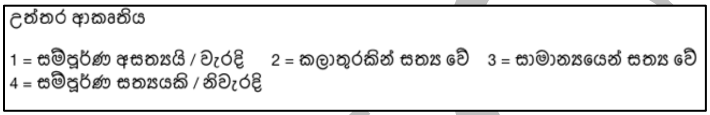
6. අවශ්‍ය ප්‍රමාණයට උත්සාහ කිරීම තුලින් මට බොහෝ ගැටළු විසඳා ගත හැකිය

7. දුෂ්කර අවස්ථාවන්ට මුහුණ දීමට සිදුවන විට, මට සන්සුන්ව ඊට මුහුණ දිය හැක්කේ මට මාගේ දරාගැනීමේ ශක්තිය පිළිබඳ විශ්වාසය තැබිය හැකි බැවිනි

8. සාමාන්‍යයෙන් මට ප්‍රශ්නයක් හමුවේ විවිධ විසඳුම් සොයාගැනීමට හැකිය

9. සාමාන්‍යයෙන් මම අසීරු අවස්ථාවකට පත්වී සිටින විට, මට එයින් මිදීමට විසඳුමක් සිතීමට හැකිය.

10. මාගේ මාර්ගයේ කුමන බාධා පැමිණියත්, සාමාන්‍යයෙන් මට ඒවා කළමනාකරණය කර ගැනීමට පුළුවනි



උත්තර ආකෘතිය

1 = සම්පූර්ණ අසත්‍යයි / වැරදි 2 = කලාතුරකින් සත්‍ය වේ 3 = සාමාන්‍යයෙන් සත්‍ය වේ
4 = සම්පූර්ණ සත්‍යයකි / නිවැරදි

Recommended reading:

Selvaratnam, N. C., Selvaratnam, N. D., Nanayakkara, A. M. N. A. D. J. S., Tennakoon, S. (2024). Household Water Insecurity Experiences (HWISE) Scale: The protocol of cultural adaptation and statistical validation, *European Journal of Public Health Studies. 7*(1), 33-53. http://dx.doi.org/10.46827/ejphs.v7i1.166

Selvaratnam, N. D., Selvaratnam, N. C. (2024). Assessment of Judgmental Validity of the Sinhala Household Water Insecurity Experiences (HWISE) Scale. Presented at *SLIIT International Conference of Advancements in Sciences and Humanities* on 4th December 2024. 413-429.

Concepts in reliability and validity sometimes tend to get complicated if we fail to see how the concepts are applied in the real-world. As a result, I encourage you to take time to read the above cited papers. In these papers, we have provided a concisely written strategies to culturally adapt and statistically validate the ‘Household Water Insecurity Experiences’ **(HWISE)** Scale.

11.0 Cronbach's alpha (α)

Since we know how to run correlations, in this section, I focus only on Cronbach's α. While other methods of reliability also use correlations, Cronbach's α is a widely used method by most psychologists to form an understanding of the reliability of a measuring instrument (i.e., a psychometric scale). So, let's take a moment to calculate Cronbach's alpha for our GSES scale.

This is a two-step process. First, let's create a new variable that contains only the 10 items of the scale. My original dataset has quite some other variables (i.e., age, final score, etc.) as well. As a result, when I upload my Excel data to *R Studio*, I have a dataset with the 10 items of the scale and a few other demographic variables as well. Because of this, I am asking *R Studio* to consider only my scale's 10 items and I am commanding to create a new variable for those 10 items. This new variable will be named 'selected_items.' Once you enter the below-given code to your console, you should be able to see the new variable on your top right window (*Figure 23*).

```
selected_items <- GSESData[, c("Item_1", "Item_2", "Item_3", "Item_4", "Item_5",
"Item_6", "Item_7", "Item_8", "Item_9", "Item_10")]
```

Sometimes, this code might not work due to unnecessary spaces the column headers of your data set may have. In such an instance follow the troubleshooting steps can be implemented.

```
selected_items <- GSESData[, 1:10]
```

In the above code, items are selected based on the position in which each item is listed in the dataset. For example, if "Item_1" appears in the 3rd column of the dataset, in that case, the code should be amended as follows.

```
selected_items <- GSESData[, 3:13]
```

This code indicates which items to be selected based on the column positions. If this code runs and your 'selected_items' variable is created, our original dataset might have spaces in the column headers which hinders the previous code from running. As a result, we can clean-up column names by adding the code below.

```
colnames(GSESData) <- trimws(colnames(GSESData))
```

Once this process is completed, you can rerun the code,

```
selected_items <- GSESData[, c("Item_1", "Item_2", "Item_3", "Item_4", "Item_5",
"Item_6", "Item_7", "Item_8", "Item_9", "Item_10")]
```

Now, before we go through the construction of this code, take a moment to review the new variable you have constructed (*Figure 24*). If the new variable is created, let's try to decode the code we just entered. I have listed the key components of the code with a brief explanation.

- **GSESData**: In **Part A** of this book, when we entered our Excel data into *R Studio*, I named my file, 'Cleaned_Data.' Similar to that, in this section, I have named my imported data file GSESData. In fact, you can use any name you like for your data file. I used GSESData as it enables me to quickly understand my dataset. Naming your dataset appropriately is important when you work with multiple datasets.

- **[,]**: This is the indexing or subsetting operation in R. It allows you to select specific elements, rows, or columns from a data frame.

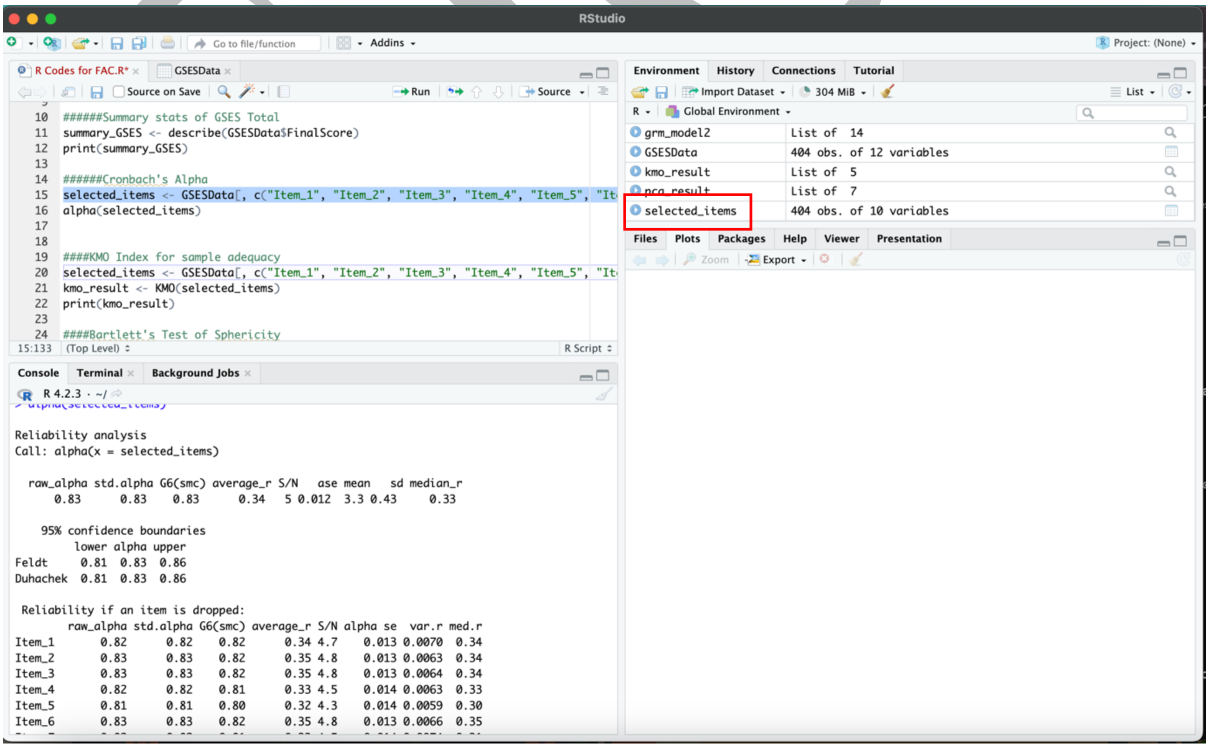
- **c("Item_1", "Item_2", ..., "Item_10")**: This part creates a vector containing the names of columns that you want to select from the GSESData data frame. The columns are named "Item_1" through "Item_10". Each column that corresponds with my scale is named "Item_1", "Item_2" and so on.

Sex	Age	Item_1	Item_2	Item_3	Item_4	Item_5	Item_6	Item_7	Item_8	Item_9	Item_10

Let's say my dataset looks like the table I have given above. You will notice that it has 12 columns. By including the columns I need for the vector c(), we can easily program our new variable.

- **selected_items <-** : The final set of columns selected will be assigned to a new variable named selected_items.

In summary, the code is extracting a subset of columns from the GSESData data frame. The columns being selected are "Item_1" through "Item_10", and they are stored in a new variable called selected_items. This could be useful if you only want to work with or analyze a specific subset of columns from the original dataset.



```
######Summary stats of GSES Total
summary_GSES <- describe(GSESData$FinalScore)
print(summary_GSES)

######Cronbach's Alpha
selected_items <- GSESData[, c("Item_1", "Item_2", "Item_3", "Item_4", "Item_5", "It
alpha(selected_items)


####KMO Index for sample adequacy
selected_items <- GSESData[, c("Item_1", "Item_2", "Item_3", "Item_4", "Item_5", "It
kmo_result <- KMO(selected_items)
print(kmo_result)

####Bartlett's Test of Sphericity
```

```
> alpha(selected_items)

Reliability analysis
Call: alpha(x = selected_items)

  raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
      0.83      0.83    0.83      0.34   5 0.012  3.3 0.43     0.33

    95% confidence boundaries
         lower alpha upper
Feldt     0.81  0.83  0.86
Duhachek  0.81  0.83  0.86

 Reliability if an item is dropped:
        raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
Item_1       0.82      0.82    0.82      0.34 4.7    0.013 0.0070  0.34
Item_2       0.83      0.83    0.82      0.35 4.8    0.013 0.0063  0.34
Item_3       0.83      0.83    0.82      0.35 4.8    0.013 0.0064  0.34
Item_4       0.82      0.82    0.81      0.33 4.5    0.014 0.0063  0.33
Item_5       0.81      0.81    0.80      0.32 4.3    0.014 0.0059  0.30
Item_6       0.83      0.83    0.82      0.35 4.8    0.013 0.0066  0.35
```

```
grm_model2       List of  14
GSESData         404 obs. of 12 variables
kmo_result       List of  5
pca_result       List of  7
selected_items   404 obs. of 10 variables
```

Figure 24: Newly created variable, 'selected_items.'

Once you create the new variable (*Figure 23*), the next step to compute Cronbach's α is simple. You just have to add the below-written code.

```
alpha(selected_items)
```

Here is your resulting output.

```
Reliability analysis
Call: alpha(x = selected_items)

 raw_alpha std.alpha G6(smc)  average_r S/N   ase    mean    sd   median_r
    0.83        0.83      0.83       0.34       5   0.012   3.3   0.43   0.33

   95% confidence boundaries
              lower   alpha.  upper
Feldt          0.81    0.83     0.86
Duhachek   0.81    0.83     0.86
```

Let's try to make sense of the output we received by entering the previous code.

- **raw_alpha:** This is the raw value of Cronbach's α, which is a measure of internal consistency reliability. It ranges from 0 to 1, with higher values indicating better reliability. In my scale, the raw_alpha is 0.83 which is indicative of good reliability.

- **std.alpha:** This is the standardized alpha, which is adjusted for the number of items in the scale. It gives you a reliability estimate that can be compared across different scales. In my scale, std.alpha is also 0.83, indicating that the reliability is consistent across different scales.

- **G6(smc):** This is an alternative measure of reliability known as Guttman's Lambda (λ) 6 (Isernia *et al.,* 2023). It also ranges from 0 to 1, and higher values suggest better reliability. In my scale, G6 λ is 0.83.

- **average_r:** This is the average inter-item correlation. It represents the average correlation between all pairs of items in your scale.

- **S/N (Signal-to-Noise):** This is a measure of the ratio of the "signal" (true score variance) to the "noise" (error variance). A higher S/N ratio is generally desirable, indicating a more reliable scale. Usually, an S/N ratio of 3 – 4 is acceptable. However, depending on the nature of the scale, this value may vary. A score of 5 is generally considered 'high' and indicates that a substantial portion of the variance in our scale is because of the true score variance as opposed to measurement error (noise). However, interpretation of the S/N ratio should be done always in consultation with other reliability measures (Cronbach & Gleser, 1964).

- **ase (Average Standard Error):** This is the average standard error of measurement. It provides an estimate of the precision of the reliability estimate. Lower values are better (Duhachek *et al.,* 2005; Duhachek & Iacobucci, 2004).

- **mean, sd, median_r:** These are statistics related to the distribution of inter-item correlations.

As per the data we have obtained, the scale is highly reliable. However, the real value of reliability could vary depending on the context. As per Feldt and Duhacheck's reliability estimates, the true reliability of the scale could fall between the lower bound of .81 and the upper bound of .86 with 95% confidence (Iacobucci & Duhachek, 2003).

11.1 Reliability if an item is dropped

Now that we know the overall reliability of our scale, let's take this one more step further by getting into item level statistics. This is not IRT analysis. This is another step we do along with reliability analysis in CTT.

Just like in SPSS, *R* also provides a variety of item-level analyses when you compute Cronbach's α. When you run the code alpha(selected_items), apart from the main reliability analysis given previously, another table with 'reliability if an item is dropped' along with a table with 'item statistics' is also given. Let's take a moment to dissect the output we have received.

```
Reliability if an item is dropped:
        raw_alpha  std.alpha  G6(smc)  average_r   S/N  alpha se   var.r    med.r
Item_1     0.82       0.82      0.82      0.34      4.7   0.013   0.0070   0.34
Item_2     0.83       0.83      0.82      0.35      4.8   0.013   0.0063   0.34
Item_3     0.83       0.83      0.82      0.35      4.8   0.013   0.0064   0.34
Item_4     0.82       0.82      0.81      0.33      4.5   0.014   0.0063   0.33
Item_5     0.81       0.81      0.80      0.32      4.3   0.014   0.0059   0.30
Item_6     0.83       0.83      0.82      0.35      4.8   0.013   0.0066   0.35
Item_7     0.82       0.82      0.81      0.33      4.5   0.014   0.0074   0.31
Item_8     0.81       0.81      0.80      0.32      4.2   0.014   0.0049   0.31
Item_9     0.81       0.82      0.81      0.33      4.4   0.014   0.0062   0.33
Item_10     0.81      0.81      0.81      0.33      4.4   0.014   0.0059   0.32
```

In the above table, the scale's overall reliability if a specific item is dropped is displayed. For instance, if 'Item_1' is removed, the α of the scale will drop down from .83 to .82. If by any chance, an item contains some error, you will see the value under 'raw alpha' goes beyond the α of the overall scale. For instance, if the α of 'Item_1' goes beyond 0.83 when it is dropped, that indicates some fault in 'Item_1.'

However, in instances where the reliability goes noticeably up as items get dropped, it is important to check the status of the specific item further through item-level statistics (next section). If reliability rises with an item dropped, you need to check the same item's item-total correlation, and its overall association with other items in the correlation matrix (*see Item Statistics*). If both of these suggest some problem (i.e., low correlations), it would be wise to revisit the scale's translation process or expert review documentation to see whether the item was poorly rated by experts or loosely translated.

If you are unsure of the translation process and expert review process, now would be a good time to read the recommended papers. In those paper, we have elaborated the process of culturally adapting and statistically validating a scale in-depth.

11.2 Item statistics

Now that we understand the purpose of the table '*reliability if an item is dropped,*' let's proceed to further discuss the next table *R Studio* computes for us when we run the code alpha(selected_items). This next table **(Page 55)** gives some key item-level statistics for us to assess the reliability of each item. Before we start deciphering the table, it will be beneficial if we learn its contents. I have listed them with simple explanations below.

- **n:** The number of cases (participants or responses) included in the analysis for each item.

- **raw.r:** The raw correlation between each item and the total score. It indicates the strength and direction of the linear relationship between each item and the overall scale. Some researchers call this '*item total correlation*.'

- **std.r:** The standardized correlation between each item and the total score. It is adjusted for the scale's reliability and indicates the contribution of each item to the overall reliability of the scale.
- **r.cor:** The inter-item correlation, representing the correlation between each item and the average of all other items in the scale.

- **r.drop:** The change in inter-item correlation if the particular item is dropped. It indicates how each item influences the overall inter-item correlation.

- **mean:** The mean score on each item, representing the average response.

- **sd:** The standard deviation of scores on each item, indicating the variability of responses.

Let's check the table now.

```
Item statistics
          n    raw.r   std.r   r.cor   r.drop   mean    sd
Item_1   404   0.58    0.60    0.53    0.47     3.5    0.60
Item_2   404   0.56    0.56    0.48    0.43     3.2.   0.71
Item_3   404   0.56    0.55    0.47    0.43     3.4.   0.72
Item_4   404   0.64    0.64    0.58    0.53     3.2.   0.72
Item_5   404   0.71    0.71    0.68    0.62     3.2.   0.68
Item_6   404   0.55    0.56    0.49    0.44     3.5.   0.63
Item_7   404   0.65    0.65    0.59    0.54     3.4    0.72
Item_8   404   0.72    0.72    0.70    0.63     3.4    0.67
Item_9   404   0.67    0.67    0.62    0.56     3.3    0.71
Item_10  404   0.67    0.68    0.64    0.58     3.4    0.62
```

As per the output given above, if we consider 'Item_1' for our analysis, this item has a raw correlation of 0.58, a standardized correlation of 0.60, and contributes to an inter-item correlation of 0.53. If this item is dropped, the inter-item correlation decreases to 0.47. You must learn how to compare both these tables to understand more about your items' reliability.

Usually, lower correlations here indicate problematic items. In some cases, you will notice the problem nature of the items is reflected both in item statistics (i.e., low correlations) and '*reliability if an item is dropped'* (i.e., reliability of the scale goes up when the item is dropped).

With the item-level statistics, we now have come to the end of reliability analysis.

Phoof!! What a relief huh?

Learning Check

1. Explain the purpose of the operator **[,]:**
2. Do a full reliability analysis for GAD7 scale.

Answers available at the end of* **Part B

Rough seas ahead captain!! I repeat rough seas ahead!!

11.3 Validity

We have completed our lesson on reliability. In my experience, a lot of students say understanding the content on reliability is the easiest. However, I should tell you that the process I have mentioned is not always this smooth and straight. When you have multidimensional (check definition) scales, you will have to go through a few more additional steps. In this book, I will focus only on unidimensional scales. Multidimensional factor analysis is beyond the scope of this book as this is dedicated mainly to undergraduate students.

Let's say our scale is highly reliable. The previous analyses indicate that clearly. Now, with that information with us, we should proceed to ensure the validity of our scale. Before we go forward, please take a moment to learn the basic types of validity we commonly use in psychological research.

Table 5: Types of validity

Form of validity	Type of validity	Description
Judgmental validity	Face validity	The degree to which a scale looks like it actually could measure what it claims to measure. This is more or less a subjective impression one forms about a scale (Musawai *et al.,* 2024; Sireci, 1998). In this step of validity testing, aspects of a scale such as grammar, sentence structure, word choices, etc. are determined by two subject matter experts (SMEs). The degree of agreement between the two raters (also known as inter-rater reliability) can be calculated via Cohen's Kappa Statistic (read the commentary on content validity in p. 74) (Desai & Patel, 2020).
	Content validity	A qualitative assessment of the content of a scale often achieved through expert reviews. Delphi method is a commonly used method to assess content validity (refer the cited paper Selvaratnam & Selvaratnam, 2024).
Criterion related validity	Predictive validity	The degree to which a scale can predict a related construct (Maisto *et al.,* 2011). For example, a positive correlation of self-efficacy and optimism suggests predictive validity of the self-efficacy scale.
	Concurrent validity	The degree to which a scale correlate with an existing scale that measures the same construct (Maisto *et al.,* 2011). For example, a positive correlation between Back's depression inventory (BDI) and Hamilton Depression Rating Scale (HAM-D) indicates concurrent validity of both scales.
Construct validity	Convergent validity	The degree to which scores of a scale correlate with measures that assesses the same construct (Holten *et al.,* 2007). For instance, items that fall under the same construct should display good correlations.

	Divergent validity	The degree to which scores of a scale does not correlate with scales that measure unrelated constructs. For example, remorse and appetite are two construct that generally do not demonstrate a strong theoretical connection (Holten *et al.,* 2007).
Internal Structural validity	Exploratory factor analysis	EFA is a method used to understand the internal structure of a scale without a priori (i.e., without knowing the number of factors) assumptions.
	Confirmatory factor analysis	CFA is used to understand the internal structure of a scale based on a conceptual model.

I have kept the descriptions of reliability and validity as succinct as possible. Providing in-depth explanations is beyond the scope of this book as my objective is to help you master *R* programming. However, if you go through the descriptions it becomes clear that judgmental validity is a qualitative process, and both criterion-related validity and construct validity are correlation powered mechanisms. Since we already have learned about correlations, you should be able to run those analyses without any issues. So, let's dive directly into internal structural validity.

Assessing the internal structure of a scale is a multi-step process. In this section, I explain how to do this in CTT. First, let's go through the steps, shall we? The steps are given based on the protocol given in Scholz *et al.,* 2002 and Selvaratnam *et al.,* 2024.

1. KMO index
2. Correlation matrix
3. Bartlett's test of sphericity
4. Exploratory factor analysis (EFA)
5. Factor diagram and scree plot

Let's go through each of the above in depth.

11.4 Kaiser-Meyer-Olkin (KMO) index in R

Based on the reliability statistics, it should be clear to you that we have administered the GSES scale to *n* = 404. Although I think this sample is more than enough to go for further analysis, it is a prerequisite to check the KMO index before proceeding toward exploring the factor structure of a scale. The KMO index checks whether the current sample is adequate for us to subject our data to undergo factor analysis. In simple terms, the KMO index is a way of assessing sampling adequacy (Field, 2000; Shrestha, 2021; Williams *et al.,* 2010).

I already have assigned all my 10 items from the dataset to a new variable named 'selected_items.' If you have not defined the variable yet, you may do so now by adding the below-given code.

```
> selected_items <- GSESData[, c("Item_1", "Item_2", "Item_3", "Item_4", "Item_5",
"Item_6", "Item_7", "Item_8", "Item_9", "Item_10")]
```

Followed by entering the above code, simply adding the code given below could generate you the KMO index.

```
> kmo_result <- KMO(selected_items)
> print(kmo_result)
```

By now you should be able to understand the above code without any issues. **KMO()** is a function and KMO is conducted for our ‘selected_items’ and we have assigned it to kmo_result. In the subsequent line, we have commanded R to provide us with the results of the KMO analysis using the **print()** function. Given below is our KMO output.

```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = selected_items)
Overall MSA =  0.89
MSA for each item =
 Item_1  Item_2  Item_3  Item_4  Item_5  Item_6  Item_7  Item_8  Item_9 Item_10
   0.87    0.91    0.90    0.90    0.89    0.87    0.92    0.89    0.86    0.87
>
```

Now, let’s interpret this output. Overall MSA (Measure of Sampling Adequacy) is 0.89 for the selected items. 0.89 is a higher score indicating a high degree of sampling adequacy for factor analysis. The closer the value is to 1, the better it is to proceed with factor analysis. Generally, a KMO value above 0.6 is considered acceptable. Individual MSA values are also provided for each item. Each item's MSA indicates how well that specific variable correlates with other variables in the dataset. Higher MSA values (closer to 1) suggest that the variable is suitable for inclusion in factor analysis. In our scale, individual MSA values also range from 0.86 to 0.92 which suggests that each item contributes well to the overall factor structure.

With this we have completed our first step in validity testing. The next step for us is to proceed towards Bartlett’s test of sphericity. But, for Bartlett’s test, checking the correlation matrix is a must.

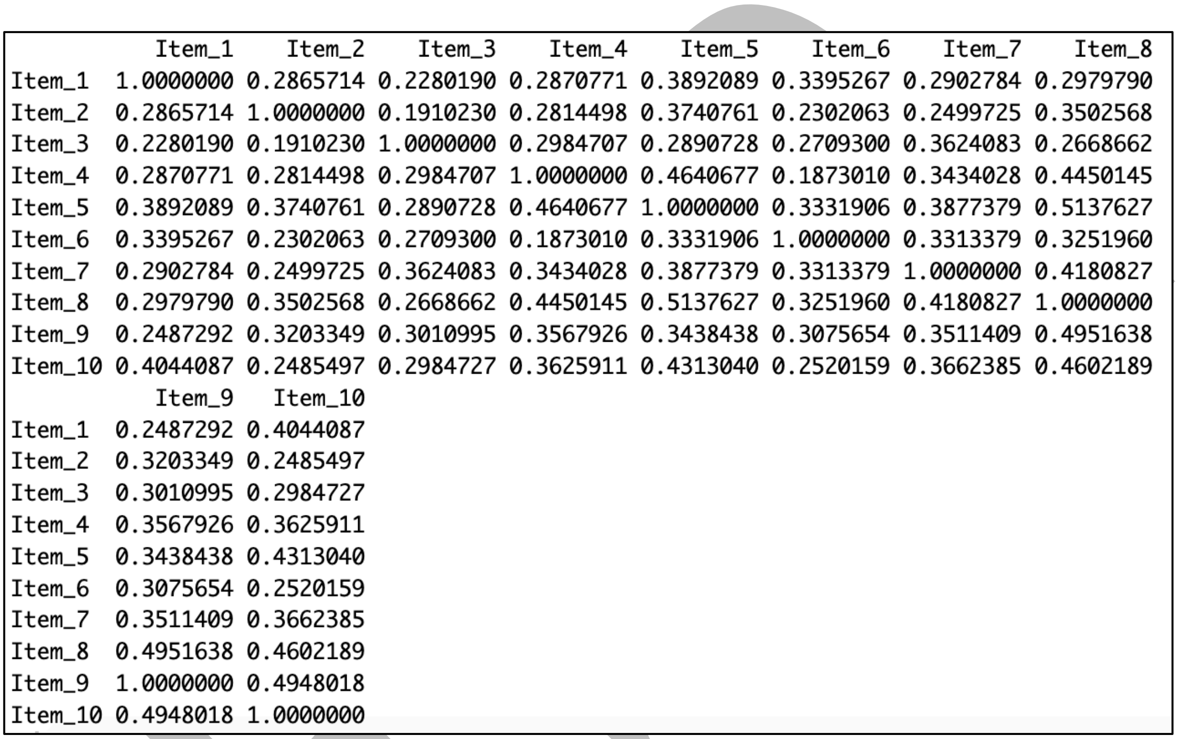
11.5 Correlation matrix

As the name suggests here we generate a correlation matrix for all our selected_items. A correlation matrix assists us in doing at least a couple of things. First, you should enter the code given below to run the correlation matrix. Since we have already entered similar-looking codes, by now you should be able to understand the workings of the code in the proceeding lines.

```
> correlation_matrix <- cor(selected_items)
> print(correlation_matrix)
```

A correlation matrix helps us to observe the linear relationship between each item in the scale. Usually, items in a unidimensional scale should have positive correlations with each other. If there are negative correlations, it could be because of a translational error and it should be corrected before continuing further analyses. Translations are generally addressed in judgmental validity testing. Secondly, a correlation matrix helps us look for any evidence of multicollinearity and singularity. If items in a correlation matrix have extremely strong intercorrelations (i.e., 0.8 and above), that could be an indication of multicollinearity. If by any chance, a pair of items have a perfect correlation, then we have a singularity. Similar to both multicollinearity and singularity, low correlations (i.e., 0.1 and below) could also be a problem as that item might not help in understanding the construction under investigation (Selvaratnam *et al.,* 2024).

Now, before we go forward, let's go through our R output for the code given above. Although I have colored all analysis output in green, here I could not do it since the table is slightly wider than usual. So, I have included a direct screenshot of the results obtained.



```
           Item_1    Item_2    Item_3    Item_4    Item_5    Item_6    Item_7    Item_8
Item_1  1.0000000 0.2865714 0.2280190 0.2870771 0.3892089 0.3395267 0.2902784 0.2979790
Item_2  0.2865714 1.0000000 0.1910230 0.2814498 0.3740761 0.2302063 0.2499725 0.3502568
Item_3  0.2280190 0.1910230 1.0000000 0.2984707 0.2890728 0.2709300 0.3624083 0.2668662
Item_4  0.2870771 0.2814498 0.2984707 1.0000000 0.4640677 0.1873010 0.3434028 0.4450145
Item_5  0.3892089 0.3740761 0.2890728 0.4640677 1.0000000 0.3331906 0.3877379 0.5137627
Item_6  0.3395267 0.2302063 0.2709300 0.1873010 0.3331906 1.0000000 0.3313379 0.3251960
Item_7  0.2902784 0.2499725 0.3624083 0.3434028 0.3877379 0.3313379 1.0000000 0.4180827
Item_8  0.2979790 0.3502568 0.2668662 0.4450145 0.5137627 0.3251960 0.4180827 1.0000000
Item_9  0.2487292 0.3203349 0.3010995 0.3567926 0.3438438 0.3075654 0.3511409 0.4951638
Item_10 0.4044087 0.2485497 0.2984727 0.3625911 0.4313040 0.2520159 0.3662385 0.4602189
           Item_9   Item_10
Item_1  0.2487292 0.4044087
Item_2  0.3203349 0.2485497
Item_3  0.3010995 0.2984727
Item_4  0.3567926 0.3625911
Item_5  0.3438438 0.4313040
Item_6  0.3075654 0.2520159
Item_7  0.3511409 0.3662385
Item_8  0.4951638 0.4602189
Item_9  1.0000000 0.4948018
Item_10 0.4948018 1.0000000
```

As per our correlation matrix, most items have good correlations with each other. However, some items such as 'Item_3' seem to have slightly lower correlations compared to others. However, as per our reliability analysis, all the items seem to be alright. Since our correlation matrix is good, we can now proceed to Bartlett's test of sphericity.

11.6 Bartlett's test of sphericity

Bartlett's test of sphericity is used to check the suitability of our correlation matrix to go through factor analysis (Field, 2000). In this test, our correlation matrix is compared against an identity matrix. An identity matrix assumes that all items are not correlated with each other. As a result, to proceed with EFA, our correlation matrix should be significantly different from an identity matrix. Since we already have observed good correlations among our items, the current correlation matrix should be significantly different from the identity matrix. Bartlett's test gives us an output with a p-value which helps us to determine the mentioned significance.

First, let’s run the below-given command on R Studio.

```
> sample_size <- 404
> bartlett_test_result <- cortest.bartlett(correlation_matrix, n = sample_size)
> print(bartlett_test_result)
```

Based on the code given above, you should now see that this analysis also considers the sample size and the code has assigned n as the first step before Bartlett’s test. Since the rest of the code is self-explanatory, let’s proceed with the output.

```
$chisq
[1] 1060.777

$p.value
[1] 7.496197e-193

$df
[1] 45
```

As per the output given above, the current correlation matrix is highly significant. The p-value here is given in scientific notations. 7.496197e-193 can be read as 7.496197 multiplied by 10 raised to the power of -193. In other words, it's an extremely small positive number, close to zero.

11.7 Factor analysis in R

Since our KMO index and Bartlett’s test of sphericity meet the required thresholds and requirements, we can now proceed with factor analysis. First, enter the code given below to run the factor analysis.

```
fac(selected_items, nfactors = 9, rotate = F)
```

In the above code, we have instructed to extract 9 factors. Technically it is not possible to have 9 factors in this data. I will explain this logic a bit later. However, since I am still exploring my factor structure, there could be more than 1 factor within this data as well. As a result, I have asked to extract 9 factors. In the above given code, ‘**fac**’ stands for the function of ‘factor analysis.’ Also, I have instructed not to rotate my factors. When you work with multidimensional scales, even though you extract multiple factors through factor analysis, the individual items might not correctly get loaded under the desired factors. As a result, we have to rotate the factors. In some sense, rotations make our factors more interpretable.

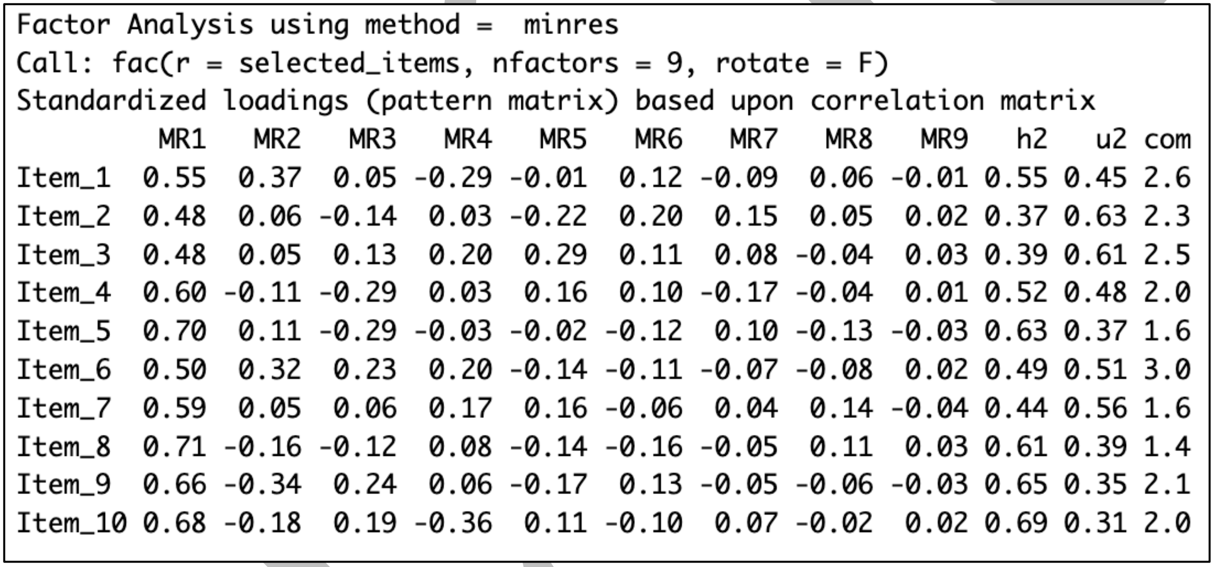
There are two major types of rotation: orthogonal and oblique. I have briefly explained them below.

Table 6: Types of EFA rotations (Osborne, 2015)

Orthogonal rotation	Oblique rotation
This rotation technique is used for uncorrelated factors. This considers each factor independent. Commonly used types orthogonal rotations include *Varimax*, *Quartimax*, and *Equamax*.	This rotation technique allows factors to correlate with each other. If there are no theoretical reasons to assume independence of factors, one could proceed with this. Commonly used types of oblique rotation include *Promax*, *Oblimin*, and *Direct Oblimin*.

In this book, I have used a well-known unidimensional scale. Since it has already been tested a few times, I do not focus on the rotation techniques in this book. However, you should know if you have multifactor solutions, you have two major types of rotations to use (Table 4).

With this knowledge, let's analyze the output of our code fac(selected_items, nfactors = 9, rotate = F). The output given below is a screenshot and as a result, it is not colored in green. But, let's take a moment to analyze our output.



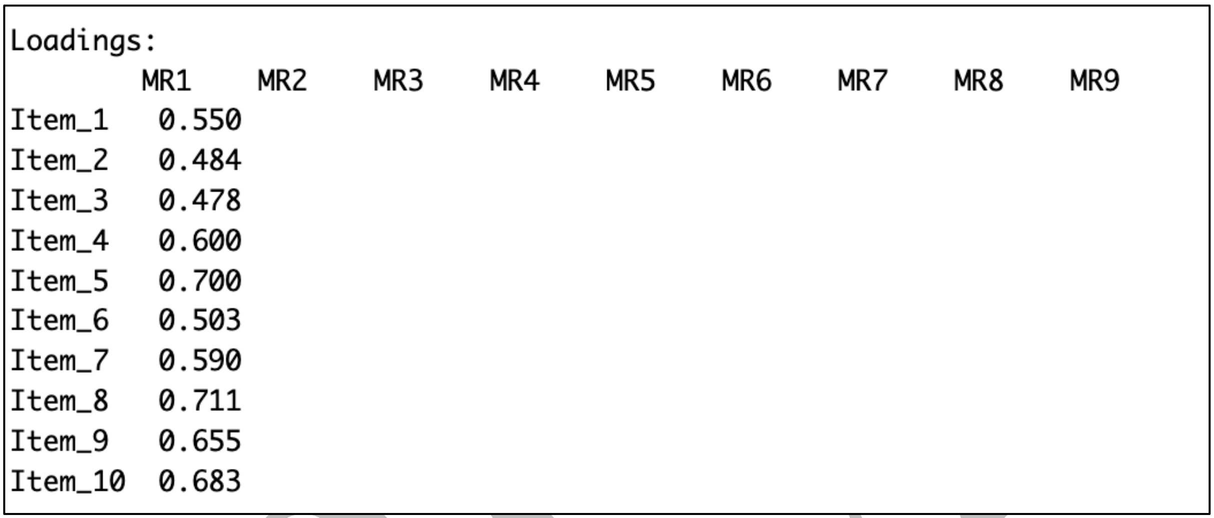
```
Factor Analysis using method =  minres
Call: fac(r = selected_items, nfactors = 9, rotate = F)
Standardized loadings (pattern matrix) based upon correlation matrix
         MR1   MR2   MR3   MR4   MR5   MR6   MR7   MR8   MR9   h2   u2 com
Item_1  0.55  0.37  0.05 -0.29 -0.01  0.12 -0.09  0.06 -0.01 0.55 0.45 2.6
Item_2  0.48  0.06 -0.14  0.03 -0.22  0.20  0.15  0.05  0.02 0.37 0.63 2.3
Item_3  0.48  0.05  0.13  0.20  0.29  0.11  0.08 -0.04  0.03 0.39 0.61 2.5
Item_4  0.60 -0.11 -0.29  0.03  0.16  0.10 -0.17 -0.04  0.01 0.52 0.48 2.0
Item_5  0.70  0.11 -0.29 -0.03 -0.02 -0.12  0.10 -0.13 -0.03 0.63 0.37 1.6
Item_6  0.50  0.32  0.23  0.20 -0.14 -0.11 -0.07 -0.08  0.02 0.49 0.51 3.0
Item_7  0.59  0.05  0.06  0.17  0.16 -0.06  0.04  0.14 -0.04 0.44 0.56 1.6
Item_8  0.71 -0.16 -0.12  0.08 -0.14 -0.16 -0.05  0.11  0.03 0.61 0.39 1.4
Item_9  0.66 -0.34  0.24  0.06 -0.17  0.13 -0.05 -0.06 -0.03 0.65 0.35 2.1
Item_10 0.68 -0.18  0.19 -0.36  0.11 -0.10  0.07 -0.02  0.02 0.69 0.31 2.0
```

In our code fac(selected_items, nfactors = 9, rotate = F), I have not specified a type of factor analysis to use. Even under EFA, there are multiple methods of factor extraction. I commonly use a method called 'principal component analysis' **(PCA).** In the above example, since I have not specified the type of FA I require, *R Studio* has run it using minimum residuals technique. Minimum residuals ('*minres*') is a FA technique whereas PCA is a dimension reduction technique. The differences between '*minres*' and PCA are far more complicated than what I have explained here. But, to keep this book simple, understanding this key difference is adequate. Now, take a moment to analyze the previous output. Columns named 'MR' identify factors. We need to see which items get loaded under each of these factors. For instance, if you get 'Item_1' it has some values given for each column. Of all the values generated for 'Item_1' the column that contains the highest value of 'Item_1' is considered an item of that specific factor. You are not required to do this factor loadings checking manually.

You can add the code given below.

```
> F1 <- fac(selected_items, nfactors = 9, rotate = F)
> print(F1$loadings, cutoff = 0.4)
```

If you notice, in this code, we have assigned our previous output to 'F1' and then we have instructed *R Studio* to print the loadings that are above the cut-off point of 0.4. By adding this cut-off score, all factor loadings below 0.4 will be removed. The output you see next has removed all loadings below 0.4.



```
Loadings:
        MR1    MR2    MR3    MR4    MR5    MR6    MR7    MR8    MR9
Item_1   0.550
Item_2   0.484
Item_3   0.478
Item_4   0.600
Item_5   0.700
Item_6   0.503
Item_7   0.590
Item_8   0.711
Item_9   0.655
Item_10  0.683
```

Since the GSES scale is rigorously tested, all item loadings fall under the same factor. However, in some instances, you will notice that some items might load under other factors. In that case, you need to have a minimum of 3 loadings under a factor to consider it as significant and that factor should explain at least 5% of the total variance of the scale. I hope you understand what I mean when I say retaining at least 3 items under a factor. In our example, all have loaded under the same factor. So, my job is easier. But, you might probably be wondering what I meant by this minimum 5% variance. Let's understand this concept a bit more in-depth.

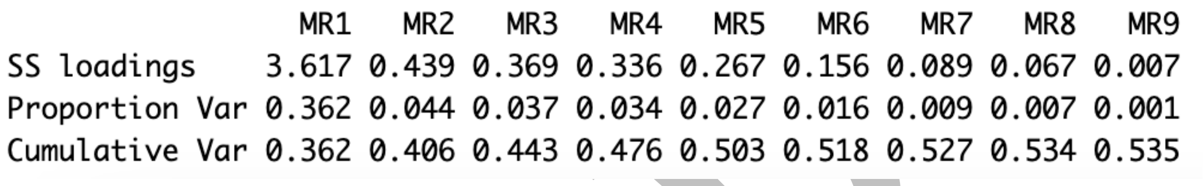
Table 7: Factor loadings, squared factor loadings and the eigenvalue

Item	Factor loadings	Squared factor loadings	Eigenvalue
Item 1	0.550	0.3025	Σ(squared factor loadings)= 3.617384 Factor 1's eigenvalue is 3.617 and it explains approximately a total variance of 36%.
Item 2	0.484	0.234256	
Item 3	0.478	0.228484	
Item 4	0.600	0.36	
Item 5	0.700	0.49	
Item 6	0.503	0.253009	
Item 7	0.590	0.3481	
Item 8	0.711	0.505521	
Item 9	0.655	0.429025	
Item 10	0.683	0.466489	

Table 6 is not an R output. I developed it to help you understand how to make sense of variance and eigenvalues. Each item has a factor loading and this is something you can observe on *R*

FA output. I simply squared each value and added them all up. The resulting value of the sum of squared factor loadings is 3.617384. This is also called an eigenvalue of a factor. If a factor has an eigenvalue above 1.0, we consider it a significant factor. Considering a factor based on its eigenvalue cut-off is called Kaiser's rule. Accordingly, the factor also explains 36% of the total variability of the observed variables. *R Studio* also gives a similar output when you run the code we previously entered.

```
> F1 <- fac(selected_items, nfactors = 9, rotate = F)
> print(F1$loadings, cutoff = 0.4)
```



```
                 MR1   MR2   MR3   MR4   MR5   MR6   MR7   MR8   MR9
SS loadings    3.617 0.439 0.369 0.336 0.267 0.156 0.089 0.067 0.007
Proportion Var 0.362 0.044 0.037 0.034 0.027 0.016 0.009 0.007 0.001
Cumulative Var 0.362 0.406 0.443 0.476 0.503 0.518 0.527 0.534 0.535
```

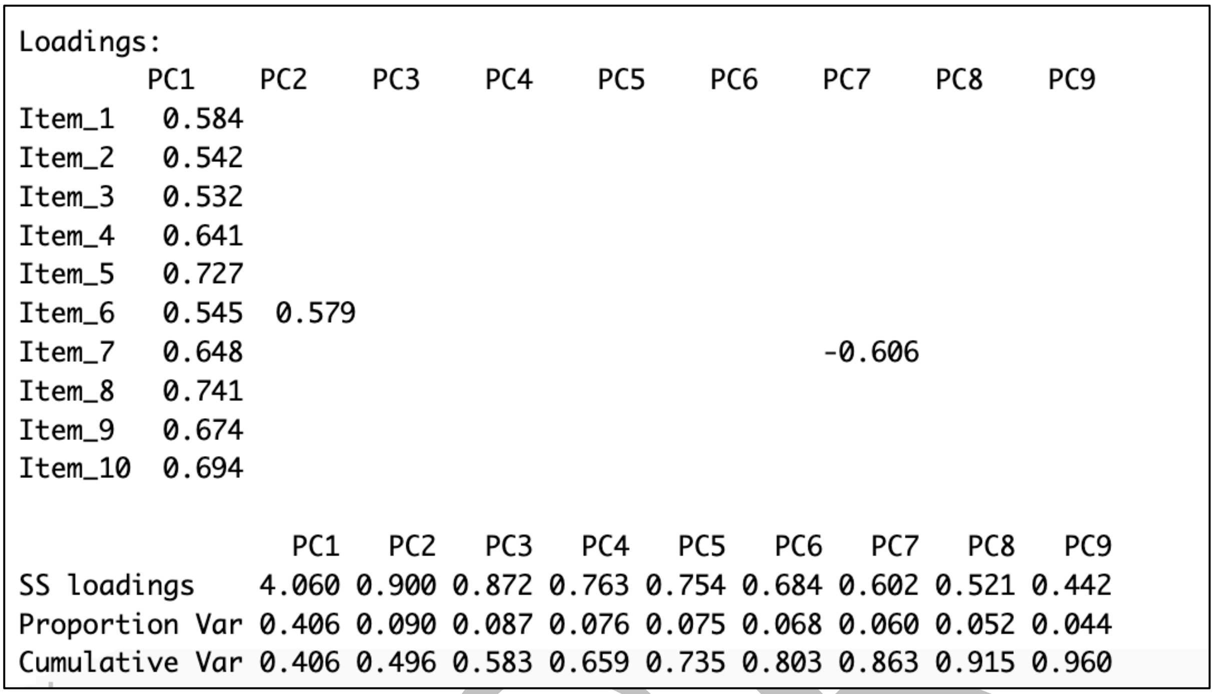
In the above *R* output, the total of SS loadings for factor 1 (MR1) is 3.671 (which is also the eigenvalue). As you see, all other factors have eigenvalues lesser than 1. So, only factor 1 (MR1) has more than 3 items loaded under its factor (in our case, all items), has an eigenvalue more than 1, and explains more than 5% of variability in observed variables. As a result, we can confirm that GSES is a unidimensional scale because it has only 1 significant factor. Simply put, all 10 items in the scale measure only self-efficacy and nothing else.

11.8 Principal Component Analysis (PCA) in R

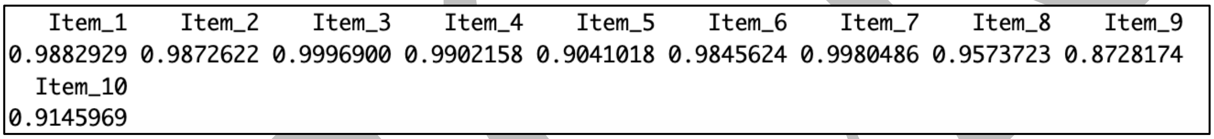
Let's run PCA now to learn dimension reduction as well. In terms of interpreting the data, since we have covered factor analysis already, the proceeding output is comprehensive. To run PCA add the below given code. Similar to the previous code in FA, we have put a cut-off value, and this time we have commanded to run PCA instead of FA.

```
pca(selected_items, nfactors = 9, rotate = F)
F1 <- pca(selected_items, nfactors = 9, rotate = F)
print(F1$loadings, cutoff = 0.5)
F1$communality
```

As per the output given below, all items except for item 6 load under the principle component 1 (PC1). Since PC2 has only 1 item and does not exceed the eigenvalue threshold of 1, I am not going to consider it a significant component. However, you will notice that as per PCA output, our scale is unidimensional and explains 40% of the variability in observable variables. Similarly, we have instructed a commonalities output from *R Studio* as well. Given below immediately after the factor loadings output is the communalities output.



```
Loadings:
        PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9
Item_1   0.584
Item_2   0.542
Item_3   0.532
Item_4   0.641
Item_5   0.727
Item_6   0.545  0.579
Item_7   0.648                                    -0.606
Item_8   0.741
Item_9   0.674
Item_10  0.694

                 PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9
SS loadings    4.060 0.900 0.872 0.763 0.754 0.684 0.602 0.521 0.442
Proportion Var 0.406 0.090 0.087 0.076 0.075 0.068 0.060 0.052 0.044
Cumulative Var 0.406 0.496 0.583 0.659 0.735 0.803 0.863 0.915 0.960
```



```
   Item_1    Item_2    Item_3    Item_4    Item_5    Item_6    Item_7    Item_8    Item_9 
0.9882929 0.9872622 0.9996900 0.9902158 0.9041018 0.9845624 0.9980486 0.9573723 0.8728174 
  Item_10 
0.9145969 
```

In PCA, every item contributes to multiple components. In our case, we have only one component, and communalities provide us with an indication of how well each item is represented by the selected components. A communality of 0.98 suggests that almost 98% of the variability in that specific item is captured by the extracted component of the PCA analysis. Since almost all items have values above 0.9, it is clear that our extracted component is successful in explaining the variance of all the listed observable items. The closer the communality is to 1, the better the variable is represented by the principal components. High communality values suggest that the variable contributes significantly to the common factors extracted by PCA, in this case, self-efficacy.

11.9 Scree Plot in R

After we form a good understanding of the internal structure of the scale, we can visually display our principal components using a scree plot by adding the below given code.

```
fa_scores <- F1$scores
pca_result <- princomp(fa_scores, scores = TRUE, cor = TRUE)
screeplot(pca_result, type = "line", col = "blue", main = "Scree Plot")
```

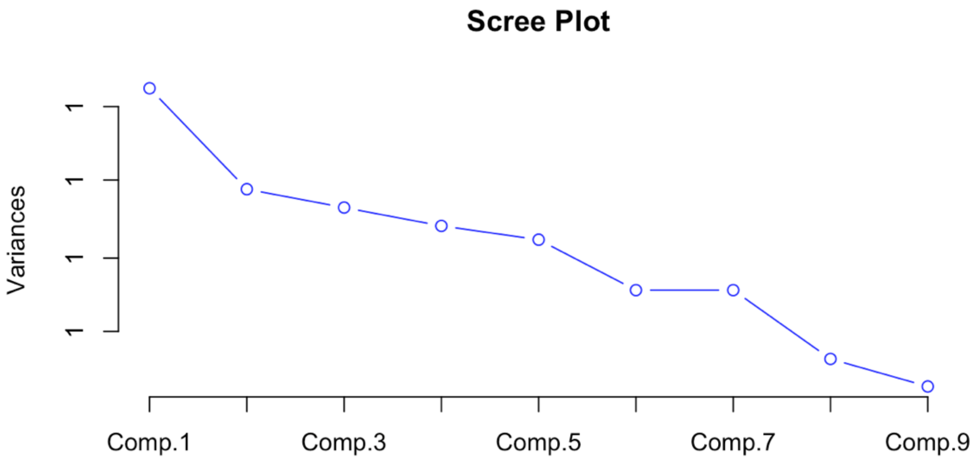


Figure 25: Scree plot

Essentially, the scree plot is a plot that plots the eigenvalues of all components extracted. In the above visualization, the values of the **Y axis** are not clear. Therefore, I added a new set of codes to correct this. I included both codes for you to compare as you learn more about R programming. So, here is an alternative method with a new package installed named nFactors.

```
install.packages("nFactors")
library(nFactors)

ev <- eigen(cor(selected_items))
nS <- nScree(x=ev$values)
plotnScree(nS, legend = F, main = "Scree Plot")
```

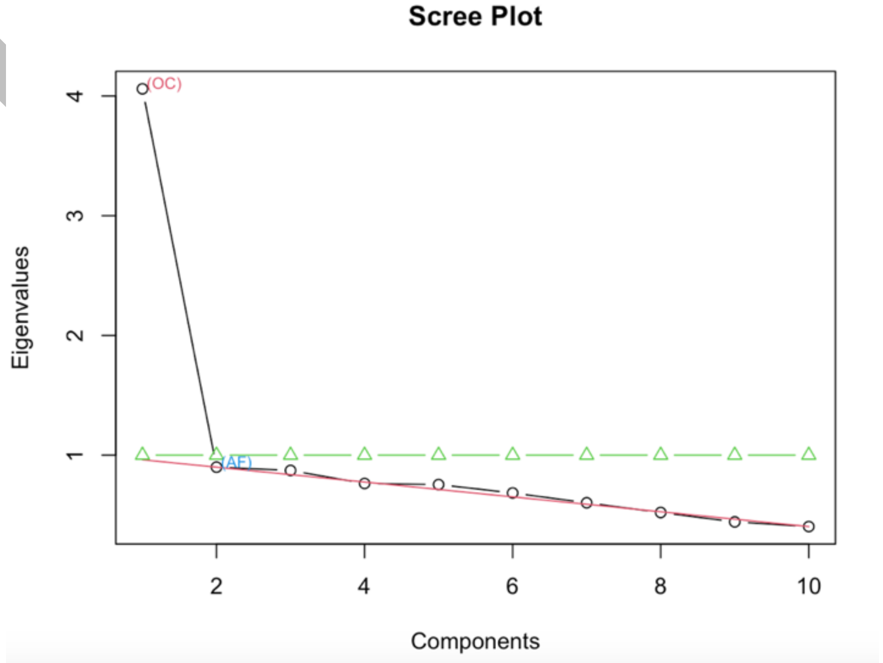


Figure 26: Scree plot created with nfactors

In the above scree plot, we can see one component at eigenvalue 4 range and the rest of the components below eigenvalue 1. So, this is an excellent way of visualizing your PCA results for readers to quickly grasp the content.

11.10 Factor diagram

Similarly, you can also draw a factor diagram for your EFA by adding the below given set of codes.

```
> fac(selected_items, nfactors = 1, rotate = F)
> F1 <- fac(selected_items, nfactors = 1, rotate = F)
> fa.diagram(F1, main = "Factor Analysis")
```

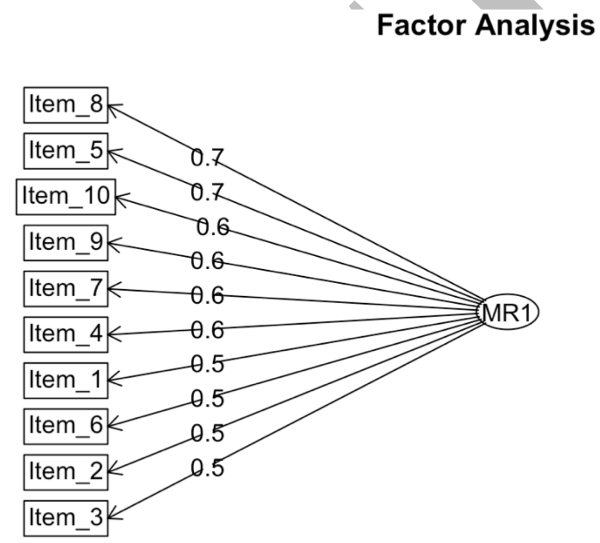


Figure 27: Factor diagram created with nfactors

The figure drawn above shows you the factor structure of GSES. In this figure, you can notice the following information.

- **Arrows:** Arrows represent the relationships between the latent factor(s) and observed variables. The direction and length of the arrows indicate the strength and direction of the relationship.
- **Loadings:** The length of the arrows often corresponds to the factor loadings, indicating how much each observed variable contributes to the underlying factor.
- **Variable Labels:** The observed variables are labeled near the arrows, helping you identify which variables are associated with the factor.
- **Factor Labels**: MR1 is the factor label.

With all of these, it is safe to say that we have completed basic EFA for a unidimensional scale. Now, if you are statistically validating a psychometric scale, to assess reliability and validity, you may take your scale through this process we have discussed in **Part B** of this book.

I don't know about you guys, but I genuinely feel awesome about running all of this in R.

Learning Check

3. Use the data of GAD-7 scale and generate the below listed output.
    a. KMO index
    b. Correlation matrix
    c. Bartlett's test of sphericity
    d. Principal component analysis
    e. Scree plot
    f. Factor diagram

Answers available at the end of* **Part B

Now take a moment to check the YouTube video that summarizes our learnings of **Part B**.

Answers

The questions posted in each of the previous sections of **Part B** are addressed below as succinctly as possible.

1. Explain the purpose of the operator **[,]:**

   This is an indexing or subsetting operation which helps a researcher select a few variables out of many variables in the dataset.

2. Do a full reliability analysis for GAD7 scale.

```
selected_items <-Cleaned_Data[, c("GADQ1", "GADQ2", "GADQ3", "GADQ4",
"GADQ5", "GADQ6", "GADQ7")]
alpha(selected_items)
```

```
Reliability analysis
Call: alpha(x = selected_items)

  raw_alpha  std.alpha  G6(smc)   average_r   S/N   ase    mean   sd   median_r
     0.62        0.62      0.65        0.19     1.7  0.077   1.1  0.52     0.15

   95% confidence boundaries
              lower   alpha   upper
Feldt          0.45    0.62    0.75
Duhachek       0.47    0.62    0.77

 Reliability if an item is dropped:
          raw_alpha  std.alpha   G6(smc)   average_r   S/N   alpha  se   var.r   med.r
GADQ1      0.55        0.55        0.57       0.17       1.2     0.091     0.024   0.15
GADQ2      0.52        0.53        0.56       0.16       1.1     0.097     0.024   0.15
GADQ3      0.60        0.60        0.59       0.20       1.5     0.082     0.015   0.15
GADQ4      0.64        0.65        0.65       0.23       1.8     0.073     0.018   0.17
GADQ5      0.58        0.59        0.62       0.19       1.4     0.086     0.030   0.17
GADQ6      0.60        0.60        0.63       0.20       1.5     0.082     0.032   0.19
GADQ7      0.56        0.57        0.58       0.18       1.3     0.090     0.023   0.15

 Item statistics
           n    raw.r    std.r   r.cor   r.drop   mean    sd
GADQ1  59    0.60     0.63    0.56     0.43     1.12   0.79
GADQ2  59    0.67     0.68    0.63     0.49     1.15   0.93
GADQ3  59    0.55     0.52    0.43     0.29     1.36   1.06
GADQ4  59    0.37     0.39    0.22     0.13     1.07   0.91
GADQ5  59    0.56     0.56    0.42     0.33     0.86   0.97
GADQ6  59    0.53     0.51    0.35     0.28     1.07   1.00
GADQ7  59    0.59     0.59    0.51     0.40     0.90   0.88
```

```
Non missing response frequency for each item
            0     1     2     3     4  miss
GADQ1    0.19  0.58  0.17  0.07  0.00    0
GADQ2    0.27  0.39  0.25  0.08  0.00    0
GADQ3    0.25  0.31  0.29  0.14  0.02    0
GADQ4    0.29  0.44  0.19  0.08  0.00    0
GADQ5    0.42. 0.39  0.10  0.07  0.02    0
GADQ6    0.32  0.42  0.12  0.14  0.00    0
GADQ7    0.39  0.36  0.24  0.00  0.02    0
>
```

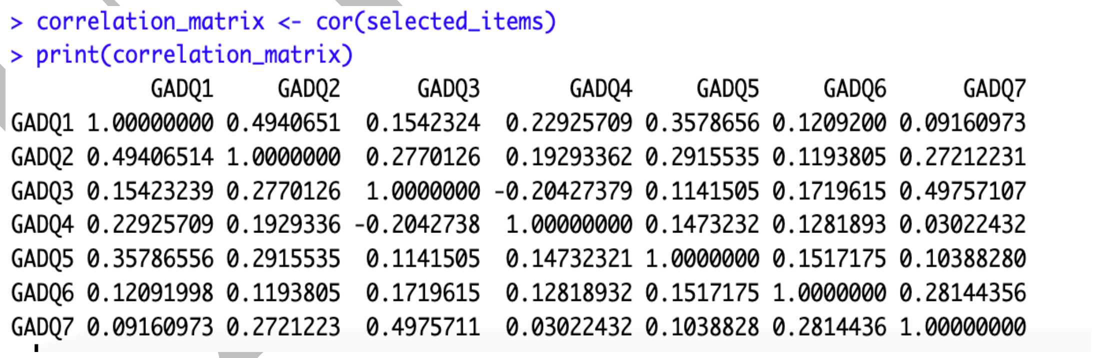
3. Use the data of GAD-7 scale and generate the below listed output.

   1. **KMO index**

```
kmo_result <- KMO(selected_items)
print(kmo_result)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = selected_items)
Overall MSA =  0.63
MSA for each item =
GADQ1  GADQ2  GADQ3  GADQ4  GADQ5  GADQ6  GADQ7
  0.65   0.69   0.56   0.51   0.77   0.71   0.59
```

   2. **Correlation matrix**

```
correlation_matrix <- cor(selected_items)
print(correlation_matrix)
```



```
> correlation_matrix <- cor(selected_items)
> print(correlation_matrix)
           GADQ1     GADQ2      GADQ3       GADQ4     GADQ5     GADQ6      GADQ7
GADQ1 1.00000000 0.4940651  0.1542324  0.22925709 0.3578656 0.1209200 0.09160973
GADQ2 0.49406514 1.0000000  0.2770126  0.19293362 0.2915535 0.1193805 0.27212231
GADQ3 0.15423239 0.2770126  1.0000000 -0.20427379 0.1141505 0.1719615 0.49757107
GADQ4 0.22925709 0.1929336 -0.2042738  1.00000000 0.1473232 0.1281893 0.03022432
GADQ5 0.35786556 0.2915535  0.1141505  0.14732321 1.0000000 0.1517175 0.10388280
GADQ6 0.12091998 0.1193805  0.1719615  0.12818932 0.1517175 1.0000000 0.28144356
GADQ7 0.09160973 0.2721223  0.4975711  0.03022432 0.1038828 0.2814436 1.00000000
```

   3. Bartlett’s test of sphericity

```
sample_size <- 59
bartletts_test_result <- cortest.bartlett(correlation_matrix, n = sample_size)
print(bartletts_test_result)

$chisq
[1] 61.30978
```

```
$p.value
[1] 8.074863e-06

$df
[1] 21
```

4. Principal component analysis

```
pca(selected_items, nfactors = 5, rotate = "varimax")
```

```
Principal Components Analysis
Call: principal(r = r, nfactors = nfactors, residuals = residuals,
    rotate = rotate, n.obs = n.obs, covar = covar, scores = scores,
    missing = missing, impute = impute, oblique.scores = oblique.scores,
    method = method, use = use, cor = cor, correct = 0.5, weight = NULL)

Standardized loadings (pattern matrix) based upon correlation matrix

        RC2    RC1    RC5    RC3    RC4    h2     u2      com
GADQ1  -0.05   0.87   0.06   0.10   0.20   0.81   0.1943   1.2
GADQ2   0.30.  0.80   0.13  -0.03   0.07   0.75   0.2457   1.3
GADQ3   0.77   0.22  -0.35   0.06   0.05   0.77   0.2261   1.6
GADQ4  -0.05   0.16   0.95   0.07   0.06   0.94   0.0637   1.1
GADQ5   0.06   0.21   0.06   0.06   0.97   1.00   0.0041   1.1
GADQ6   0.16   0.06   0.06   0.98   0.06   0.99   0.0072   1.1
GADQ7   0.90   0.02   0.15   0.15   0.03   0.85   0.1516   1.1

                       RC2    RC1    RC5    RC3    RC4
SS loadings            1.52   1.52   1.07   1.00   1.00
Proportion Var         0.22   0.22   0.15   0.14   0.14
Cumulative Var         0.22   0.43   0.59   0.73   0.87
Proportion Explained   0.25   0.25   0.18   0.16   0.16
Cumulative Proportion  0.25   0.50   0.67   0.84   1.00

Mean item complexity =  1.2
Test of the hypothesis that 5 components are sufficient.

The root mean square of the residuals (RMSR) is  0.07
 with the empirical chi square  13.19  with prob <  NA

Fit based upon off diagonal values = 0.91
```

End of Answers

12.0 Cohen's Kappa Statistic (CKS) & Fleiss Kappa Statistic (FKS)

While introducing the concept of judgmental validity, I mentioned *Cohen's Kappa* and *Fleiss Kappa*. Essentially, these are two methods of finding inter-rater agreement (reliability). Cohen's Kappa is used to find agreement between two raters while Fleiss Kappa is used to find agreement in more than two raters. Although we rely heavily on the subjective opinions of SMEs in judgmental validity processes, we need to ensure the opinions are well aligned. The Kappa coefficients help us observe the degree of this alignment. Higher scores indicate higher agreement between SMEs opinion, thus implying good inter-rater reliability. Such good reliability is a quality indicator of good judgement validity.

I have explained both of these in-depth in the paper Selvaratnam & Selvaratnam (2024) cited in the recommended readings (p. 50). However, in this book, I will introduce you to the formulae involved in Kappa statistic computations.

Table 8: Cohen's Kappa (Yusoff, 2019)

K_{CKS} =	**I-CVI**	**Pc**
$\frac{(\text{I} - \text{CVI} - \text{Pc})}{1 - \text{Pc}}$	$= \frac{\text{Number of YES for an item}}{\text{Total number of criteria}}$	$\textbf{Pc} = \left[\frac{\text{N}!}{\text{A}!\,(\text{N} - \text{A})!}\right] X\, 0.5^{\text{N}}$

Note. N refers to total number of items and *A* refers to the number of agreement.

The formula in the Table 7 should be computed to find the agreement between two raters. This value should be higher than 0.74 to indicate good face validity (good agreement between the two raters) (Shrotryia & Dhanda, 2019). K_{CKS} = 1.00 indicate perfect agreement. To calculate *Cohen's Kappa*, Content Validity Index (*I-CVI*) for a single item of a scale as rated by the SMEs should be calculated as displayed in Table 8. Along with this, formula for Pc (proportion of observed agreement) should also be calculated.

Table 9: Fleiss Kappa (Landis & Koch, 1977)

K_{FKS} =	**Expected agreement**	**Observed agreement**
$\frac{Po - Pe}{1 - pe}$	$\text{Pe} = \Sigma p_j^2$	$\text{Po} = \frac{1}{N \cdot n \cdot (n-1)} \left(\sum_{i=1}^{N} \sum_{j=1}^{K} n_{ij}^2 - N \cdot n\right)$

Compared to *Cohen's Kappa*, *Fleiss Kappa* is slightly complicated. Fleiss Kappa formula involves computation of expected agreement (*Pe*) and observed agreement (*Po*). Pe and Po has their own formulas as displayed in Table 8. *Fleiss Kappa* is used in instances where inter-rater reliability is required for when more than two SMEs opinions are considered. Fleiss Kappa can be easily calculated in DataTab statistical software. If not, it can manually be calculated as well. The video uploaded to YouTube by the channel DATAtab is highly useful in the manual calculations. In this edition of the book, I have not included the step by step manual calculation as that is beyond the scope of a psychology undergraduate student. Here, K_{FKS} = 0.8 is considered good agreement between raters and K_{FKS} = 0.2 and above indicate fair agreement.

This page is intentionally left blank

13.0 Concluding Remarks

Compared to what we learned in **Part A**, I think we definitely have mastered a significant chunk of psychometrics in **Part B**. Further, I hope after the completion of **Part A** and **Part B**, you are now more familiar with *R* language. You must have noticed that my explanations for every code is reduced in **Part B**. Similarly, in **Part C**, you will not encounter redundant elaborations of codes. Instead you will notice brief commentary in places where some support to understand novel codes are absolutely required.

I hope you took time to do analyses given in the learning check and the video links I have posted (end of the book). By repeatedly practicing these concepts on sample datasets, you can further master both programming skills and also statistical analysis. After the completion of **Part B** you are now capable of successfully adapting a psychometric scale using the classical test theory (CTT). I would also encourage you to read the papers I have suggested. This includes a complete methodology involved in a scale adaptation and an in-depth critique of judgmental validity. These papers will further help you to realize some of the finer details in scale adaptation and statistical validation.

Now, with this information in our hands, let's take our enthusiasm for psychometrics one more step further with **Part C** of this book which is dedicated for Item Response Theory (IRT). In my opinion, IRT is not everyone's cup of tea. One of the biggest regrets in my life as an academic is that I never got the chance to study a PhD program in 'psychometrics.' I always felt that psychometrics is my biggest calling. However, most programs in USA require a very competitive GRE score to get into such programs. So, instead of the brutal selection process, I decided to take a swift turn to education and subsequently obtain a master and a PhD for educational leadership and management. As a result, my understanding on IRT is not perfect. Consider this **Part C** as a section written by a young student to another young student. It took me 7 years to make sense of most of the IRT theories and I did it all alone. There was no single resource person in Sri Lanka to receive the required assistance. So, please, read this with some caution; but, I still hope this will enlighten you about the modern advances we have made in the field of psychology to make it a true science.

Part C
14.0 Item Response Theory (IRT)

We have a solid understanding of CTT, and now is the time to learn about IRT. Essentially, IRT is made to assess the accuracy of the scoring of test items on a scale. The idea of IRT was first introduced by D. N. Lawley of Edinburgh University in the 1940s (Baker, 2001). Lawley was a British statistician. Later, more work in this area was pioneered by George Rasch in the 1960s (Baker, 2001). Rasch was a Danish mathematician and developed the Rasch model which is a form of IRT. Since this time, multiple IRT models have been developed by various other scientists. Essentially, IRT is a collection of models and unlike CTT, IRT is not impacted by the sample and deals with items as opposed to overall scales or questionnaires.

Rasch model

One of the best methods to learn IRT is to start with the Rasch model as it is generally considered easier to understand. This model is used to test the scoring accuracy of scales that has items with dichotomous options (i.e., Y / N). So, without further ado, let’s go through a basic Rasch output for two items in a test (Figure 26).

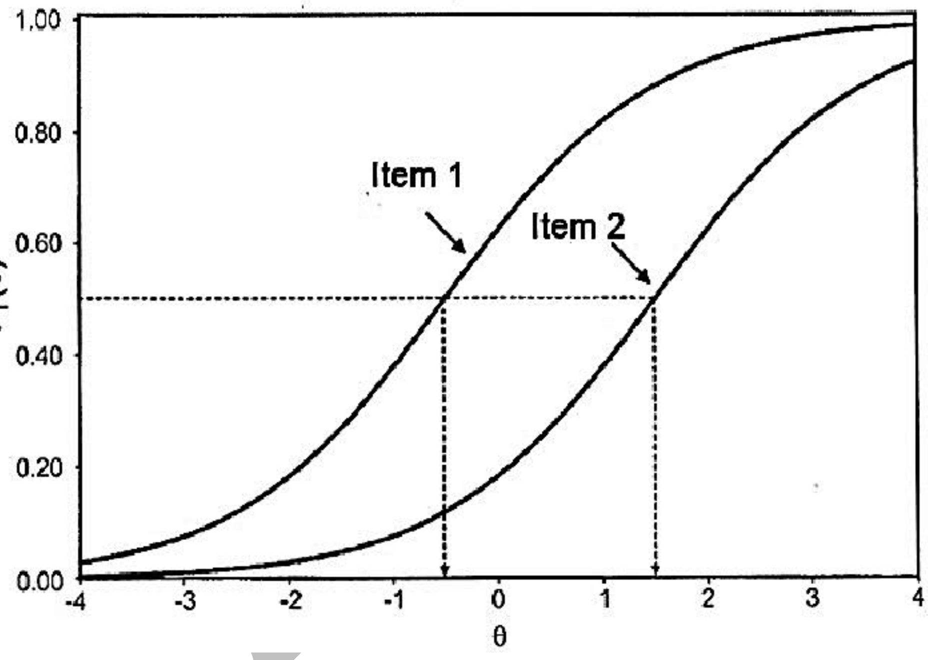


Figure 28: Rasch model

As per Figure 26, the X-axis shows theta (**θ**), which is also known as the ability of an individual. The word ability might not resonate well when you use IRT models in social sciences. So, you may consider **θ** as the latent variable (i.e., math ability, depression, etc.). Here, the Y axis shows the probability of getting an item correct. As per the example given above, item 1 can be considered easier compared to item 2. A person has a 0.5 probability to get item 1 right at **θ** less than 0 (or in other words, at lower levels of ability). A person has a 0.5 probability of getting item 2 right at **θ** 1.5. If both items test the math ability of students, to get item 2 accurate, the student has to have more math ability compared to getting item 1 correct. In that sense, **θ** is the latent variable in a continuum. As per this information item 2 can be used to identify students who have more math ability as opposed to item 1. In this manner, the Rasch model can assess item difficulty. Here item 2 is more difficult compared to item 1. Accordingly, this is also called a 1 parameter model as it assesses only item difficulty.

2PL (parameter) model and graded response model (GRM)

Since the Rasch model only evaluates 'item difficulty', another parameter, 'discriminability' can also be introduced. The 2PL model can assess both item difficulty and discrimination (Matteucci & Stracqualursi, 2006).

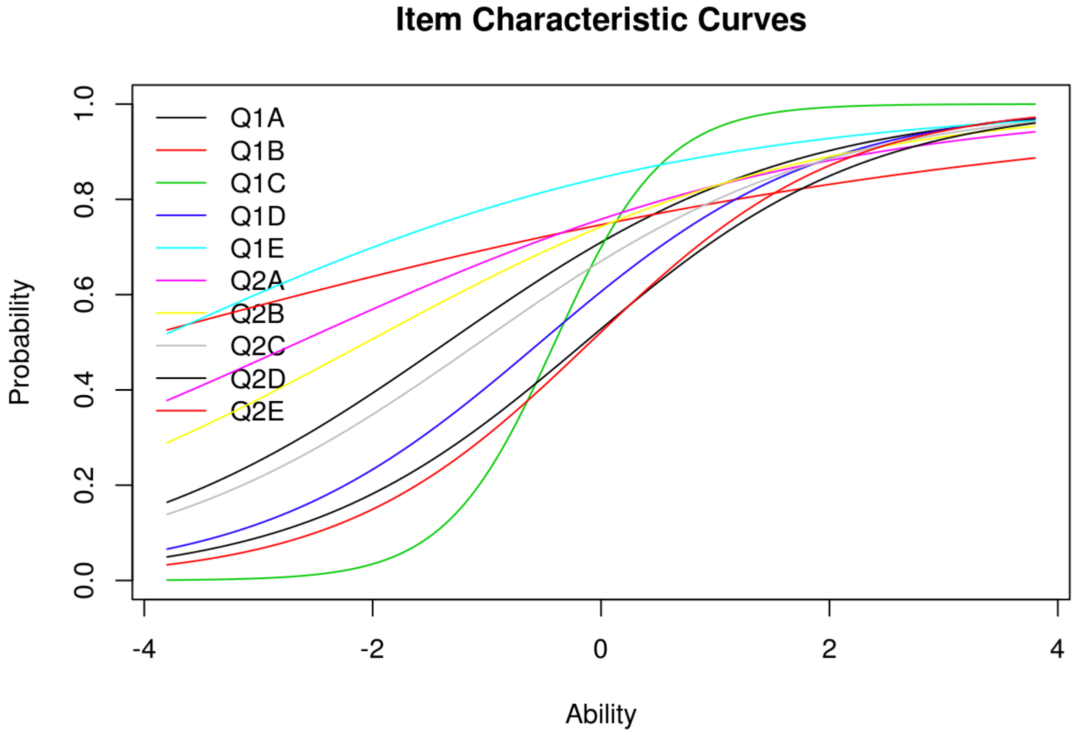


Figure 29: 2PL Model

The 2PL model can discriminate which items are better at identifying the latent trait in addition to the item difficulty. In Figure 27, Q1C (green line) and Q1D (blue line) have a 0.5 probability of obtaining the correct answer at **θ** less than 0. So, both items are not difficult. So, for instance, if the latent variable we are interested in measuring is math ability, individuals who take this test do not require a lot of ability to get Q1C and Q1D accurate. But, unlike the Rasch model, here you will notice that each item has a different 'steepness' to its curve. This steepness helps us observe the discriminability of an item. Higher steepness indicates a higher ability of an item to discriminate test takers based on the latent trait under investigation. As per this logic, Q1C (green line) is steeper than Q1D (blue line) indicating Q1C to be better discriminating individuals for math ability as opposed to Q1D. As per this figure, the lowest discrimination comes from Q1E (light blue line). Although the 2PL provides two parameters, it still can accommodate items with dichotomous responses (i.e., Y / N). To address this problem, we could use a graded response model (GRM) which is a 2PL model that can accommodate items with polytomous answer options. To keep this book simple and comprehensive, I am considering only a GRM analysis. In the subsequent sections, you will learn how to run a GRM using R Studio for the same dataset we used previously for our **Part B**. In that manner, now we use IRT on the Sinhala generalized self-efficacy scale (SGSES).

If you have used logistic regression, you might notice that Item Characteristic Curves (ICCs) look similar to what you obtain when you run a logistic regression for a study with categorical outcomes. GRMs are also specifically designed for ordered categorical outcomes (i.e., Likert scale items). Now, with this understanding in our minds, let's first run a graded response model to our dataset.

14.1 How to run a graded response model (GRM)

Similar to our previous analyses, to run IRT, we need to install a new R package named LTM (Rizopoulos, 2006). So, let's first install our package by adding the below code.

```
> install.packages("ltm")
```

Upon adding the package, now, let's load it, so that we can start using it.

```
> library(ltm)
```

Now, we are all set to run our GRM. Now, when you want to run GRM, you should first decide what type of GRM is required for your data. For instance, running an unconstrained version assumes that your model will consider items to have different levels of discriminability. This unconstrained version is what we usually consider when running a GRM. Similarly, the constrained version keeps discriminability constant for all items. Constrained versions assume that all items are equally good at discriminating ability (Smyth, n.d.). In fact, we could run both constrained and unconstrained versions of GRM for SGSES data and later go for a model-fit to see which model (unconstrained vs. constrained) is the best for our scale. This model fit is tested through a likelihood ratio test and other fit indexes. However, one should always keep in their mind that in LRT analyses, there is an implicit assumption that the constrained version is nested (check definition) within the unconstrained model (Williams, 2020). A nested model can be seen as a special case of the unconstrained model, typically achieved by imposing restrictions (constraints) on the parameters of the unconstrained model.

Anyways, to see which model fits the best, first, we should be able to run the GRM. So, without, further a-do, let's run the unconstrained GRM first. Later, we can run the constrained version and perform a model fit analysis as well.

Unconstrained GRM

The code required for GRM is simple. The below-given code can run the GRM analysis, and the code is given in blue color.

```
> grm_model <- grm(selected_items)
> summary(grm_model)
```

Once you run this code, it will generate a list of statistical output. The output seems to be categorized based on the items of our scale. Take a moment to go through the output colored in green. At first the output gives a model summary interpreted through **log.Lik**, **AIC**, and **BIC**. Similarly, you will notice there are 3 extremity values given for each item with a value to determine the degree of discriminability (Dscrmn).

```
Call:
grm(data = selected_items)

Model Summary:
   log.Lik      AIC      BIC
 -3404.391 6888.782 7048.839

Coefficients:
$Item_1
        value
Extrmt1 -3.948
Extrmt2 -3.104
Extrmt3 -0.121
Dscrmn   1.321

$Item_2
        value
Extrmt1 -3.912
Extrmt2 -1.891
Extrmt3  0.574
Dscrmn   1.211

$Item_3
        value
Extrmt1 -4.071
Extrmt2 -2.405
Extrmt3  0.063
Dscrmn   1.063

$Item_4
        value
Extrmt1 -3.195
Extrmt2 -1.672
Extrmt3  0.512
Dscrmn   1.514

$Item_5
        value
Extrmt1 -2.871
Extrmt2 -1.557
Extrmt3  0.419
Dscrmn   2.203
```

```
$Item_6
        value
Extrmt1 -4.468
Extrmt2 -2.907
Extrmt3 -0.174
Dscrmn   1.193

$Item_7
        value
Extrmt1 -3.603
Extrmt2 -1.770
Extrmt3  0.018
Dscrmn   1.566

$Item_8
        value
Extrmt1 -2.975
Extrmt2 -1.630
Extrmt3  0.076
Dscrmn   2.522

$Item_9
        value
Extrmt1 -3.040
Extrmt2 -1.693
Extrmt3  0.193
Dscrmn   1.806

$Item_10
        value
Extrmt1 -3.754
Extrmt2 -2.018
Extrmt3  0.037
Dscrmn   2.065


Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.032
quasi-Newton: BFGS
```

Let’s take the output of item 1 for further analysis.

```
Coefficients:
$Item_1
       value
Extrmt1 -3.948
Extrmt2 -3.104
Extrmt3 -0.121
Dscrmn   1.321
```

As you now notice here, item 1 has produced 3 extremity parameters. In Likert type scales, the output is usually given for ***k* – 1** category. These parameters help us estimate the probability in which a respondent chooses a specific response based on their ability (latent variable). In the present context, extremity parameters will help us understand how likely respondents or test-takers are to choose the most extreme options such as ‘strongly disagree’ or ‘strongly agree’ in the SGSES scale. The extremity values generally tend to be negative, indicating a higher likelihood of the respondents towards selecting extreme categories. A higher negative extremity parameter for a response category suggest that individuals with lower levels of the latent trait are more likely to choose that response. For instance, in the present context, a respondent should not have higher ability (self-efficacy) to select the first response (not at all agree) which is represented by Extremity 1. You will also notice that from the first Extremity parameter to the second, and from second to third, ability required increases. In the LTM package, lower categories are estimated first so that highest category can be derived from others. However, in this book, we do not dive deep into those calculations. Here, we try to use IRT mostly to complement CTT analyses.

```
Coefficients:
$Item_1
       value
Extrmt1 -3.948
Extrmt2 -3.104
Extrmt3 -0.121
Dscrmn   1.321
```

Now, again, let’s take a moment to understand what is colored RED in the above output. This is the discrimination parameter that is usually denoted by *a*. The thresholds for an item’s capacity to discriminate can be interpreted as follow.

Table 10: Discrimination cut-offs (Mueller, 2023)

a above 1	High discriminability
a above 0.5 to 1	Moderate discriminability
a below 0.5	Low discriminability

A larger *a* (in this case *a* = 1.321) indicates that the item has relatively high discrimination. This means that item 1 is effective in identifying respondents with varying levels of the latent trait (or ability). As per the above analysis, item 2 and item 3 has the lowest capacity to discriminate when compared with rest of the items. This low capacity to discriminate is reflected in CTT as well. For instance, item 2 and 3 has the lowest **r.cor** values and the lowest factor loadings scores in factor analysis. Similarly, item 5, item 8, and item 10 display the highest discriminability and the CTT output also indicate those items to have the highest factor loadings. In that sense, we can use GRM analysis to complement what we have already learned about this scale via CTT.

Constrained GRM

Similar to the unconstrained model, we can also run a constrained model to re-evaluate our scale using GRM. To run a constrained version, the previous command has to be amended.

```
grm_model2 <- grm(selected_items, constrained = TRUE)
summary(grm_model2)
```

In the above command, you will notice a change. Here, we have asked the GRM to consider the model to be constrained by keeping constrained as TRUE. Please note that unconstrained is defined as **grm_model** and constrained is defined as **grm_model2**. We can later compare these two models to see which one fits the data the best.

Let's take a moment to review the output. First it has given the model summary (**Log.Lik**, **AIC**, and **BIC**) and then extremity parameters and discrimination for each item.

```
Call:
grm(data = selected_items, constrained = TRUE)

Model Summary:
   log.Lik      AIC      BIC
 -3432.976 6927.953 7051.997

Coefficients:
$Item_1
        value
Extrmt1 -3.506
Extrmt2 -2.756
Extrmt3 -0.109
Dscrmn   1.572

$Item_2
        value
Extrmt1 -3.244
Extrmt2 -1.594
Extrmt3  0.480
Dscrmn   1.572

$Item_3
        value
Extrmt1 -3.077
Extrmt2 -1.840
Extrmt3  0.034
Dscrmn   1.572
```

```
$Item_4
        value
Extrmt1 -3.107
Extrmt2 -1.626
Extrmt3  0.504
Dscrmn   1.572

$Item_5
        value
Extrmt1 -3.456
Extrmt2 -1.838
Extrmt3  0.514
Dscrmn   1.572

$Item_6
        value
Extrmt1 -3.676
Extrmt2 -2.408
Extrmt3 -0.154
Dscrmn   1.572

$Item_7
        value
Extrmt1 -3.589
Extrmt2 -1.758
Extrmt3  0.025
Dscrmn   1.572

$Item_8
        value
Extrmt1 -3.772
Extrmt2 -2.040
Extrmt3  0.124
Dscrmn   1.572

$Item_9
        value
Extrmt1 -3.301
Extrmt2 -1.814
Extrmt3  0.223
Dscrmn   1.572

$Item_10
        value
Extrmt1 -4.301
Extrmt2 -2.334
Extrmt3  0.062
Dscrmn   1.572
```

```
Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.11
quasi-Newton: BFGS
```

You will notice that in the above output, all items have the same discrimination, *a* = 1.572. As per the discrimination thresholds we discussed earlier, it is safe to say that all items are good at identifying individuals at varying levels of ability. So, let's take a moment to further review which model actually fits the data the best.

Constrained vs. unconstrained model fit

To check the model fit, the following command has to be run.

```
anova(grm_model2,grm_model)
```

Please take note that, in the above command, constrained version should be included first within parenthesis for the model testing to run correctly. The resulting output is given below.

```
Likelihood Ratio Table
              AIC       BIC      log.Lik    LRT   df    p.value
grm_model2   6927.95   7052.00   -3432.98
grm_model    6888.78   7048.84   -3404.39   57.17   9    <0.001
```

As per the results, the unconstrained version (**grm_model**) is a better fit for the data. Here, the model fit is assessed through log-likelihood (**log.Lik**), Akaike Information Criterion (**AIC**), and Bayesian Information Criterion (**BIC**). Usually, in GRMs, there are no specific agreed cut-off points to determine which model fits the best for the given data. As a result, these values should be used for model comparison as opposed to absolute assessment of model fit.

Table 11: Model fit indexes

Log-likelihood	Higher (less negative) values indicate a better fit of the model to the data.
Akaike Information Criterion	Lower values indicate better model fit (Williams, 2020). In the present output, ΔAIC = 6888.78 – 6927.95 = 39.17 The negative ΔAIC indicates that **grm_model** is a better fit compared **grm_model2**.

	Here **Δ** (*delta*) stands for the difference of the AIC values. A **Δ** of 2 or above indicates evidence for one model over the other.
Bayesian Information Criterion	Lower values suggest evidence for one model over the other and the interpretation is done via **Δ**BIC of the two models similar to **Δ**AIC (Williams, 2020).

Further, in the above output a likelihood ratio test (LRT) further suggest the two models are different in a statistically significant manner further confirming the **grm_model** a better fit compared to **grm_model2**.

14.2 Item Response Category Characteristic Curves

Similar to the output we obtained earlier, each item can also be visualized to further obtain an understanding about how each item’s responses help identify ability of the test taker. For instance, here, the command to obtain the mentioned curves is given below.

```
Constrained <- grm(selected_items, constrained = TRUE, start.val = 'random')
par(mfrow = c(2,3))
plot(Constrained, legend=TRUE)
summary(Constrained)
```

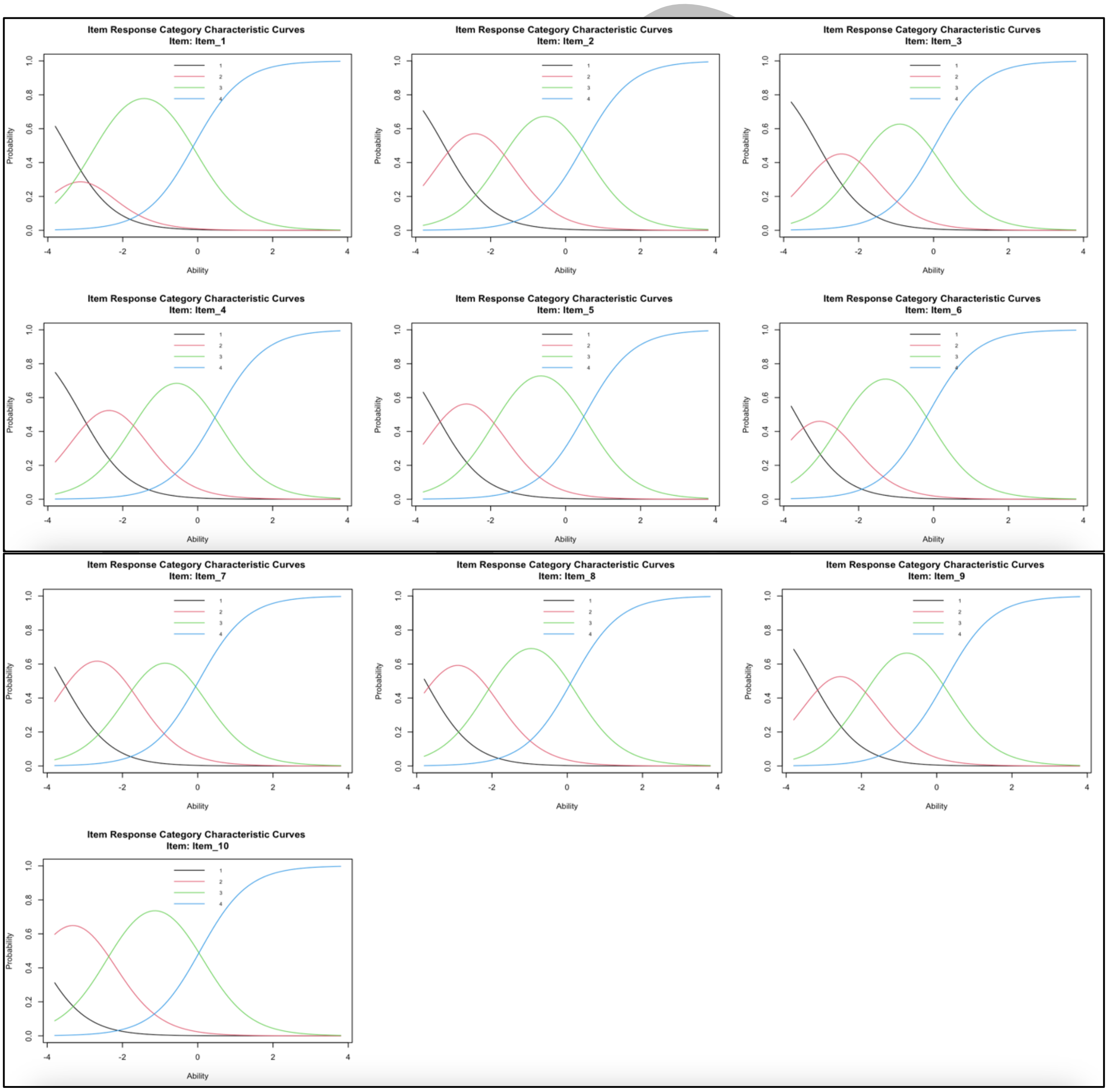
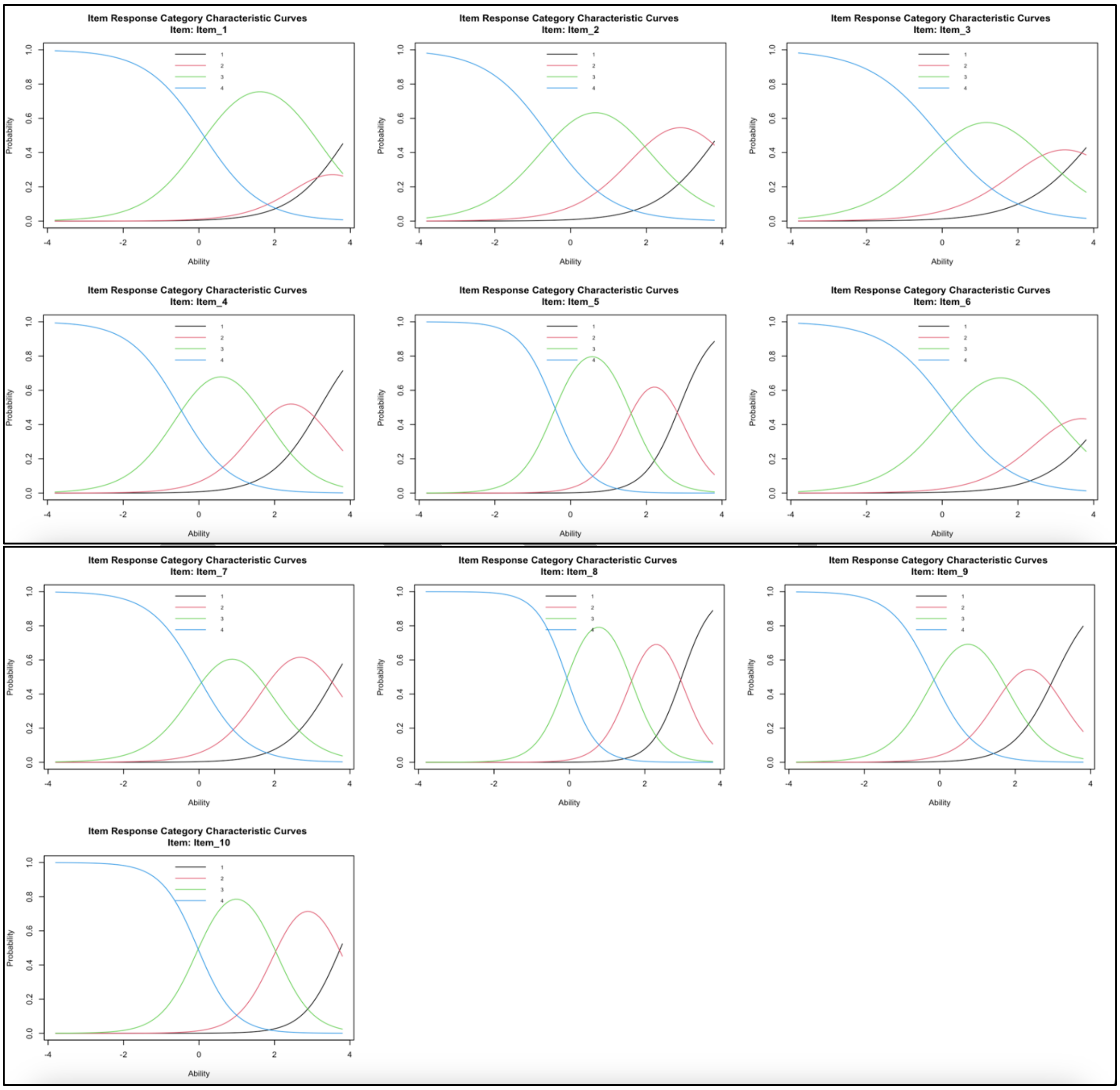


Figure 30: Item Response Category Characteristic Curves (Constrained)

Each curve can be assessed in relation to the explanations given in the previous sections of the book (page 85). We can obtain the curves for the unconstrained version too by slightly amending our command.

```
Unconstrained <- grm(selected_items, constrained = FALSE, start.val = 'random')
par(mfrow = c(2,3))
plot(Unconstrained, legend=TRUE)
summary(Unconstrained)
```


Item Response Category Characteristic Curves Item: Item_1

Item Response Category Characteristic Curves Item: Item_2

Item Response Category Characteristic Curves Item: Item_3

Item Response Category Characteristic Curves Item: Item_4

Item Response Category Characteristic Curves Item: Item_5

Item Response Category Characteristic Curves Item: Item_6

Item Response Category Characteristic Curves Item: Item_7

Item Response Category Characteristic Curves Item: Item_8

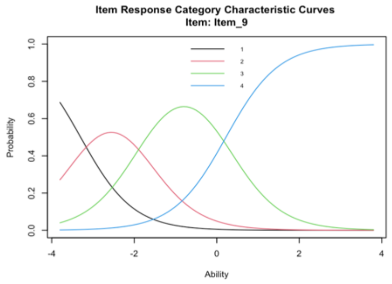
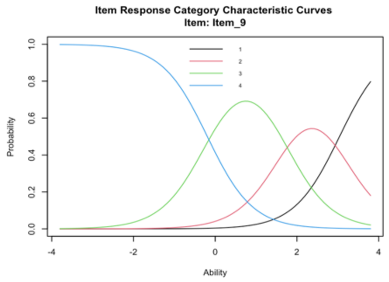
Item Response Category Characteristic Curves Item: Item_9

Item Response Category Characteristic Curves Item: Item_10

Figure 31: Item Response Category Characteristic Curves (Unconstrained)

Through this unconstrained version, we can observe which responses are more difficult and require more ability to be endorsed by a test taker. To have a better understanding about what we could see visually in the constrained and unconstrained versions, let’s keep the curves of item 9 side by side.

Table 12: Constrained Vs. Unconstrained GRM

Constrained	**Unconstrained**
 In this, you will notice, for item 9, of the 4 responses, individuals have 0.5 probability of selecting 1st response (not at all true) at negative **θ**. For, option 2, this occurs in -2.5 **θ**, 3rd response at -1.8 **θ** and 4th (exactly true) at 0.2 **θ**. I gave the **θ** values approximately based on what I could gather from this figure. Thus, individuals with higher ability (in this context, efficacy) will have a higher likelihood of endorsing option 4 as opposed to the other 3 in item 9).	 Compared to the constrained version, you will notice that 1st response (not at all true) requires the highest **θ** while 4th response (exactly true) has the lowest **θ** of -0.2. In this context, individuals with the highest latent trait tend to endorse the response options that tends to negate with the idea of efficacy, indicating a counter-intuitive way of these curves work.

Why is there a contradiction in how responses are endorsed in the two versions?

There is no easy answer for this. The data and model fit indexes already support the unconstrained version. Perhaps, our understanding of the latent trait should be amended to make the most sense of the obtained results. If we measure 'self-efficacy' through our latent trait, then the constrained version makes the most sense. However, I tend to have a slightly different style of thinking here. Self-efficacy is a universal construct. In almost all research studies we did, generally, people tend to have higher scores for self-efficacy. It is not unusual even for most people to have scores closer to 40, which is the highest achievable in the scale. However, it is extremely unusual for someone to score 10, which is the lowest achievable in the scale.

If the 10-item scale measures a person's '*understanding or perception of self-efficacy'*, then, a greater comprehension of the concept of self-efficacy is required to say no to it. For example, the person with greater **θ** (perception of self-efficacy) tends to endorse the item responses 1 and 2. In that sense, the latent trait requires more explanations about its true meaning.

However, keep in mind that this is simply my understanding of what is happening here. This book is not a book that carries theoretical justification. This mainly carries method to run analyses in R and some explanations to help you interpret the output. When you interpret these outputs, you have to consult relevant theory and research to make the best decisions.

14.3 Item Information Curves (IIC)

After comparing the item response category characteristic curves, we also obtain item information curves (IICs). These IIC curves helps us see how much of information each item is capable of providing us about the latent trait. This IIC curves can be obtained by simply adding the command below to your console.

```
plot(Unconstrained, type = "IIC", lwd = 2, cex = 0.8, legend = TRUE,
    cx = "topleft", xlab = "SGSES", cex.main = 1, cex.lab = 1, cex.axis = 1)
```

The resulting output helps us to see the items that has the highest and the lowest amount of information about the latent trait. As per the image given below, item 8 provides the highest amount of information while item 2 and 3 provide the least. The same patterns were noticed during the observation of extremity parameters and discrimination in the IRT, and also even during reliability analysis and factor analysis of CTT. Like we discussed earlier, in that sense, IRT can be further used to complement our understanding about a scale's psychometric properties obtained through CTT.

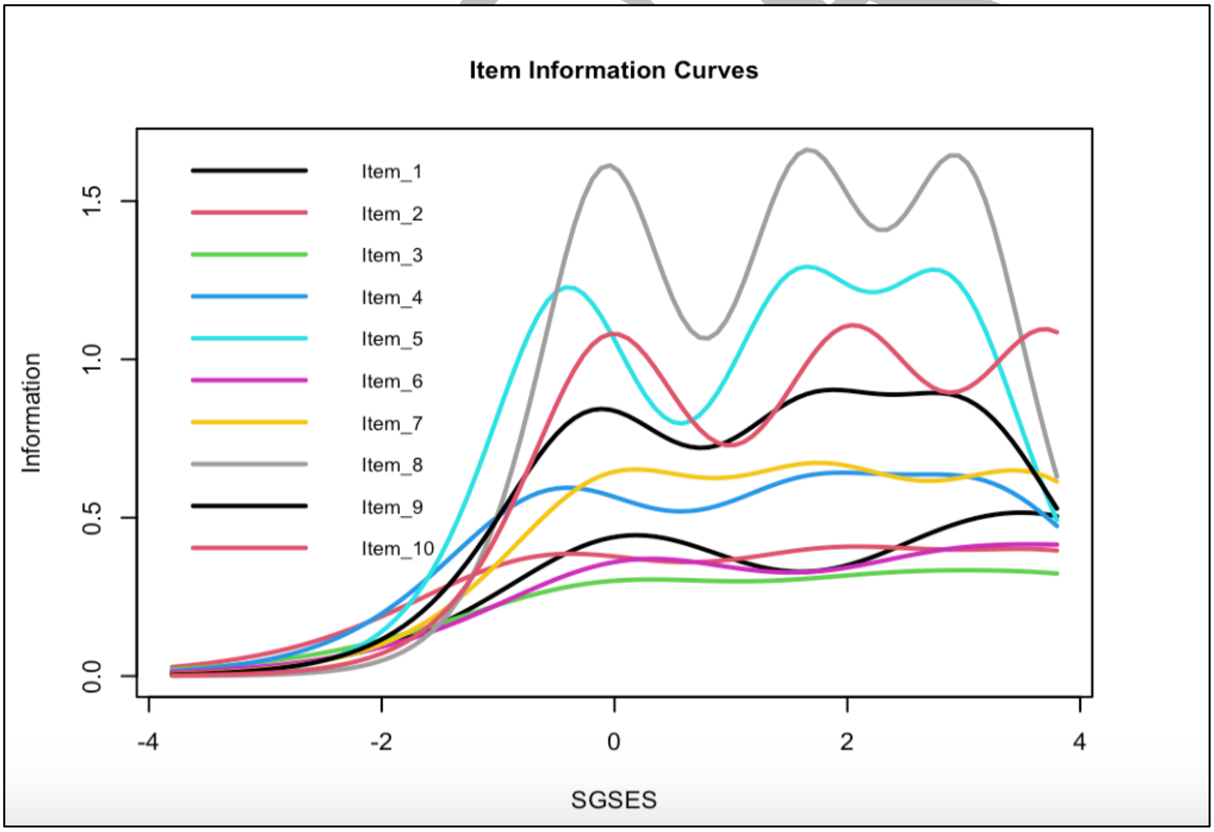


Figure 32: Item Information Curves (Constrained)

As per the figure given above, most items are good at providing information at higher **θ**.

14.4 Test Information

Similar to understanding information each item covers in the scale, total information provided by the scale can be understood by adding the below given commands. I have given the command and output together.

```
> information(Constrained, c(-4, 4))

Call:
grm(data = selected_items, constrained = TRUE, start.val = "random")

Total Information = 38.96
Information in (-4, 4) = 33.8 (86.76%)
Based on all the items

> information(Unconstrained, c(-4, 4))

Call:
grm(data = selected_items, constrained = FALSE, start.val = "random")

Total Information = 41.77
Information in (-4, 4) = 36.81 (88.14%)
Based on all the items
```

The unconstrained version provides **41.77** information, which is the overall amount of information available from the selected items. Within the latent trait from -4 to 4, 36.81 units of information is provided. This 36.81 units is **88.14%** of 41.77 of total information the scale provides. Using this same method, test information can be interpreted for the constrained version as well. However, since the unconstrained is considered the ideal model for the data we have collected, I decided to interpret just that.

This forms the end of our IRT lesson indicating the completion of the second edition of this book. I am pretty sure that this book will help you effectively adapt and statistically validate psychometric scales.

This page is intentionally left blank

Additional Lessons
15.0 Application of Psychometrics

What I have given in this book's **Part C** is how a simple demonstration of IRT is applied and that too using a simple model. There are plenty of IRT models for researchers to select from. However, it is of great importance to note that the reason to learn correlations, CTT, and IRT is to make sense of the construction process of scales. When you decide a scale to be statistically tested, there are quite a lot of things to consider than what I have given above. I have listed some scenarios with some possible solutions to consider when you work on adapting and statistically validating scales. Regardless of the software one would use, constructing a scale or statistical adaptation of a scale always consist of a qualitative evaluation of a series of outcomes. With that in our minds, let's go through some sample scenarios.

Scenario 1:
My scale is unidimensional, but, the correlation matrix has low correlations or minimal correlations. Why?

Minimal correlations (0.2 and less) or no correlations between items of a unidimensional scale is unlikely. Some possible reasons could include sampling error. For instance, not having an appropriate sampling strategy could seriously impact the correlation matrix. A good sampling strategy from Selvaratnam *et al.* (2024) is given below for your reference.

Table 13: Sampling strategy example (Selvaratnam *et al.* 2024)

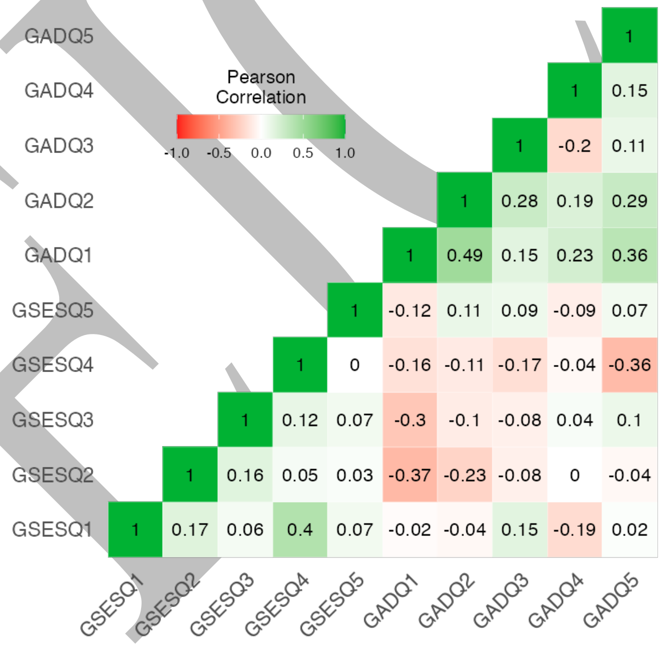
Component	Explanation
Target population	Sri Lankans 18 years and above who live with their families and share substantial responsibilities with them in their respective households will be considered for the study. Individuals who may live in hostels, residence halls, etc. will not be considered. Both mother and father of a nuclear family, and adults of an extended family who share key responsibilities in the household also qualify to take part in this study.
Sampling unit	Households (not more than 1 qualifying person from a single household)
Sampling frame	A list of registered households in Sri Lanka from government records that are accessible to researchers.
Sampling technique	Systematic sampling will be used to identify a few locations representing the three major climatic zones of Sri Lanka. Within each zone, households will be randomly selected to meet individuals who may qualify to take part in the present study.
Sample size calculation	Sample to item ratio (20:1) as suggested by Costello & Osborne (2005), Gorsuch (1983), and Memon *et al.* (2020).

In the above example, to administer the 'household water insecurity experiences' (HWISE) scale, authors have identified the target population, sampling unit, sampling frame, sampling technique, and the sample size calculation. All researchers should focus on developing a good sampling strategy similar to what is given above to avoid sampling error. Sampling error in return can impact the overall scores obtained from participants which may impact the correlation matrix.

In addition to this, if the scale underwent translations, language errors could have resulted in poor correlations. If the judgmental validity obtained for the scale's translations indicated deficits, in that case, the scale has to be amended and take through an additional round of judgmental validity testing. Most Sri Lankans scholars use the Delphi method to evaluate the content of a scale. If the correlational coefficients are low in the matrix, a good step is to revisit the Delphi process. Usually, such problems are noted by the SMEs during the Delphi rounds. A good explanation of the usage of the Delphi process and how to further quantify SME input is available in the cited paper Selvaratnam & Selvaratnam (2024) in p. 50.

Scenario 2:
My scale is multidimensional, so, how can I get a correlation matrix for my scale?

The process of obtaining a correlational matrix for a multidimensional scale is the same process as obtaining it for a unidimensional scale. However, when you observe correlational coefficients, observe them for items within a specific dimension. This will help you identify items that have poor or negative correlations.



	GSESQ1	GSESQ2	GSESQ3	GSESQ4	GSESQ5	GADQ1	GADQ2	GADQ3	GADQ4	GADQ5
GADQ5										1
GADQ4									1	0.15
GADQ3								1	-0.2	0.11
GADQ2							1	0.28	0.19	0.29
GADQ1						1	0.49	0.15	0.23	0.36
GSESQ5					1	-0.12	0.11	0.09	-0.09	0.07
GSESQ4				1	0	-0.16	-0.11	-0.17	-0.04	-0.36
GSESQ3			1	0.12	0.07	-0.3	-0.1	-0.08	0.04	0.1
GSESQ2		1	0.16	0.05	0.03	-0.37	-0.23	-0.08	0	-0.04
GSESQ1	1	0.17	0.06	0.4	0.07	-0.02	-0.04	0.15	-0.19	0.02

Figure 33: Correlation heatmap (Jamovi output)

In the output given above, although it is a *Jamovi* output, it still shows how a correlation matrix functions. Here I have included a correlation matrix between 5 items of GSES and 5 items of GAD7. As you can see, GSES 5 items show reasonable correlations among each other and the green colour indicates that these are all positive correlations. However, if you notice the items of GAD7, the 3rd item has a negative correlation with its 4th item. In such a situation, you need to do what I have mentioned in scenario I to figure out what has gone wrong.

Scenario 3:
What if my Bartlett's Test of Sphericity (BTS) is not significant?

This would be problematic as BTS is a requirement for factor analysis. It is a method researchers use to assess the factorability of a scale along with KMO index. Such a result is unlikely and if that is the case, it could be due to serious methodological errors. I have seen that once in my entire life when a student added fake data to a scale adaptation process and the correlation matrix had no correlations whatsoever. In such a scenario, a factor analysis, of course, is not possible. If the scale was already constructed in a foreign language with outstanding psychometric properties displayed, the mentioned issue of non-significance in BTS is unlikely. However, if you are constructing a scale for a new construct from the scratch, this could be a possibility especially if relevant psychological theories are not properly utilized to identify the required items. However, the existence of judgmental validity is to avoid issues in the correlation matrix. So, if one creates a scale from the scratch, but decided to stay away from judgmental validity, scenario 3 is a definite possibility.

Scenario 4:
What if my overall reliability for a chosen unidimensional scale is very low?

Low reliability does not necessarily mean your scale is bad. This dip in reliability may have resulted due to some items that are not functioning well. First check the correlation matrix to see what specific items have the lowest correlations with other items. Then, go to reliability statistics and check whether the reliability improves if some of those items are removed. If the reliability goes up when these items are removed, go back to the judgmental validity outcome and see whether experts flagged these items to have errors. Finally, get your factor analysis outcome (EFA) and see whether the items have good factor loadings. If the loadings are poor, and everything else I have mentioned also ticks, then, you can remove the faulty items. The remaining items should still belong to a specific factor (dimension) and should explain noticeable variance.

In some instances the causes could be too obvious. For example, not identifying the right sample could bring low reliability. If the scale has not reached the intended audience, participants may make errors when providing answers. If the scale is filled by a clinician or a research assistant, distortions in reporting patient, client, or participant answers is also a possibility. Underreporting some data by clinicians, researchers, etc. could also be another possible issue. In that sense, you would have to spend substantial time to figure out the causes.

In my opinion the best option is to focus all of these during the conceptualization of the research project than to correct errors when data does not add up. We have to be proactive in these kinds of situations.

Scenario 5:
What if my overall multidimensional scale has poor reliability?

Repeat the scenario 4 separately for each dimension. If that does not work, you will have to rotate your factors. Orthogonal rotations for uncorrelated factors and oblique rotations for correlated factors should be used. Upon rotating, for each factor or dimension, repeat scenario 4.

Scenario 6:
How could I evaluate my scale using both reliability and exploratory factor analysis?

Cronbach's alpha (α) above 0.7 indicates good reliability, but make sure to check item level statistics as well. Validity has to be established first judgmentally, then via correlation matrix, EFA, and CFA. For this, you should follow the instructions given in pages 53-55.

Scenario 7:
What are some possible reasons for poor internal structural validity?

Lack of good supporting theory could be a major reason for a scale to lose fit to a specific set of latent variables. If not, poor judgmental validity testing could have been a reason for a poorly translated scale to receive the green light for more statistical testing. Cultural differences could also impact the nature of items making them less valid for certain cultures. I remember once one of my students wanted to translate and statistically validate the PTSD checklist scale. The translations went smooth, but one item of this scale never had good correlations with the rest of the scale. The scale identified reactions of people when they hear loud thudding or blast sounds. Usually, people with PTSD will have a panicked kind of a reaction to such sounds. My student administered the scale in Jaffna, Sri Lanka where most people are used to loud sounds from the civil war our country experienced. None of his participants had PTSD, but, all his participants had selected extreme options in the scale for this particular item making it share unusual correlations with the rest of the items. This impacted his factor structure. This is a fine example to show how cultural context could impact the internal factor structure of a scale.

Scenario 8:
What is the ideal sample size for my scale?

For most of the statistical tests such as T-tests and ANOVAs we usually perform a power analysis through software such as G*Power to find the optimal sample size. As far as I know such software does not exist for factor analysis. So, what we usually do is following some rules. For example, one of the most commonly used rules is sample to item ratio. A frequently used ratio is 20:1 which means 20 participants per item. So, if a scale has 10 items, as per this ratio, the total should be a minimum of 200 participants. Some researchers use 5:1 and 10: 1 ratios as well. However, 20:1 ratio has the capacity to provide the appropriate factor structures 70% of the times while 10:1 at 60% and 5:1 at 40% (Costello & Osborne, 2005). Further guidance for sample size calculation can be obtained through Costello & Osborne (2005), Gorsuch (1983), and Memon *et al*. (2020).

Scenario 9:
Is it always necessary to have a confirmatory factor analysis?

Not necessarily. EFA is usually performed to explore the factor structure. When we know for certain the number of factors (especially in instances of scale adaptation to new cultures), some researchers directly proceed with CFA. This is if the previous versions of the scale has already been tested psychometrically and the researchers expect a particular factor structure. Some perform both EFA and CFA in the same research. The added benefit of the CFA is the capacity to obtain model fit indexes which helps to fix the model. More information on obtaining the right model fit and how to interpret model fit indexes will be explained in-depth in the next edition of the book.

Scenario 10:
How can I find the ideal IRT model for my scale?

Finding the right IRT model for a scale depends largely on the nature of items that the scale contains and the number of factors. If the items of the scale has binary (Yes or No) options, in that case, a Rasch model is a good fit. If the items of the scale has polytomous (Likert) options, then usually a GRM is ideal. However, the number of dimensions in the scale also is another determinant. For example, most IRT models are made for unidimensional scales. For multidimensional scales, researchers should use multidimensional IRT models. Given below is a summary of key IRT models.

Table 14: A summary of IRT models

IRT Model	**# Dimensions**	**Item Type**	**Parameters**	**Usage**
1 PL Rasch Model	Unidimensional	Dichotomous	Item difficulty	Basic model for measuring abilities
2 PL Model	Unidimensional	Dichotomous	Item difficulty, discrimination	More flexibility than 1 PL by accounting for item discrimination
3 PL Model	Unidimensional	Dichotomous	Item difficulty, discrimination, guessing	Accounts for guessing, used in high-stakes testing
4 PL Model	Unidimensional	Dichotomous	Item difficulty, discrimination, guessing, slipping or careless mistakes/	Robust against aberrant answers. Can be used in high-stakes testing
Graded Response Model (GRM)	Unidimensional	Polytomous	Item difficulty, discrimination	Used for ordinal items (e.g., Likert scales)
Generalized Partial Credit Model (GPCM)	Unidimensional	Polytomous	Item difficulty, discrimination	Used for ordered responses and discrimination varies across items. Could be used as an alternative to GRM.

Multidimensional IRT (MIRT)	Multidimensional	Dichotomous / Polytomous	Item difficulty, discrimination, guessing (optional)	Measures multiple traits or abilities simultaneously
Partial Credit Model (PCM)	Unidimensional	Polytomous	Item difficulty	Extension of Rasch for items with partial credit
Nominal Response Model (NRM)	Unidimensional	Nominal	Item category parameters	Used for nominal items (no natural order). Discrimination allowed to vary across items.

Note. This table is prepared based on the insights provided in Cohen *et al*., 1993, Gyamfi & Acquaye, 2023, Kean *et al*., 2017 & Liao, 2012.

Bonus Lesson
16.0 Simple Linear Regression (SLR)

We have now completed all the intended lessons of this book. But, after learning all of this, at least in my opinion, it would be futile if we do not learn the most basic regression analysis. One could consider simple linear regression a slight extension of correlations. However, one should always keep in the mind that the manual process of doing a regression is a painstaking process that involves multiple steps. We will not be discussing the manual steps involved in this book. However, we will focus on the key steps involved in a simple linear regression when performed via *R* studio. Some of these steps will look very familiar to you. As a result, I have included explanation only when we encounter newer codes and outputs.

Step 1: Install the required packages

The first step of running a simple linear regression is to install the relevant packages and load them to be used. I have displayed the necessary commands for your reference. These codes are the same that I have given in the **Part A** of this book.

```
install.packages("pacman")
```

```
library(pacman)
```

```
pacman::p_load(pacman, dplyr, GGally, ggplot2, ggthemes, ggyis, httr, lubridate, plotly, rio,
rmarkdown, shiny, stringer, tidyr)
```

Step 2: Attach the data

Once the relevant data packages are all loaded, we can then proceed to attach the data. Similar to **Part A** of the book, in this as well, I will be using the same dataset.

```
attach(Cleaned_Data)
```

Step 3: Check names of the variables

Once the two steps are completed, the names of the variables can also be checked by adding the below given command. I have not included the output since the output mentions the same variables we have discussed in the previous parts of this book.

```
names(Cleaned_Data)
```

Step 4: Check variable type

We can also proceed to check the variable type before proceeding towards the analysis by running the below given command. This is not a must, but a recommended step to ensure that both variables are indeed alright to be considered for a simple linear regression. Similar to step 3, I have not given the output since we already know that both these variables are 'numeric.'

```
class(GAD7Total)
class(GSESTotal...15)
```

Step 5: Scatter plot

The next step is to obtain the scatter plot. This step is similar to what we did in **Part A** and as a result I hope you will be able to interpret the command without any problem.

```
plot(GAD7Total, GSESTotal...15, main = "General Anxiety and Self-Efficacy", col =
"darkblue", xlab = "General Anxiety", ylab = "Self-Efficacy")
```

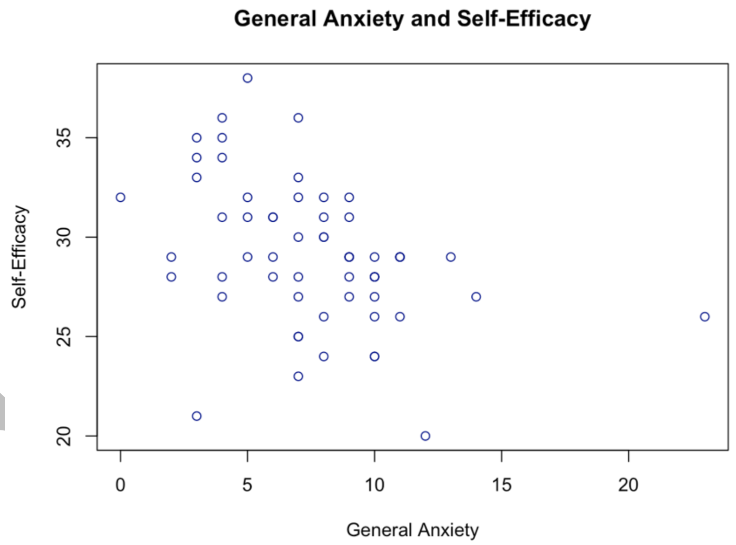


Figure 34: Scatter plot of general anxiety and self-efficacy

Step 6: Quick Correlation

We can then perform a quick correlation test by running the code given below. Please take note that in the below given code, **X** variable is Anxiety while **Y** is Efficacy. This order will be swapped when we run the linear model (Step 7).

```
cor(GAD7Total, GSESTotal...15)
```

```
[1] -0.3961685
```

As per the output given, general anxiety and self-efficacy shares a moderate correlation.

Step 7: Linear model

The next step is a new step. We are going to run a linear model, hence we run the function **lm()** and feed it to a new variable named ‘mod’ (This is the first command). After this, we can obtain a summary of the mod. The summary is given below for interpretation. Please note, in the below given code variable **Y** (Efficacy) is added first.

```
mod <- lm(GSESTotal...15 ~ GAD7Total)
summary(mod)
```

```
Call:
lm(formula = GSESTotal...15 ~ GAD7Total)

Residuals:
   Min         1Q       Median      3Q        Max
-9.9108.   -1.9406      0.4653     2.0594     7.8812

Coefficients:
                   Estimate     Std. Error      t value      Pr(>|t|)
(Intercept)         32.0989      1.0132         31.681      <2e-16 ***
GAD7Total          -0.3960       0.1216         -3.258        0.0019 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.344 on 57 degrees of freedom
Multiple R-squared:  0.1569,  Adjusted R-squared:  0.1422
F-statistic: 10.61 on 1 and 57 DF,  p-value: 0.001896
```

As per the output given above, the model is significant with a notable effect size as denoted by $r^2 = .1569$. What this means is that Anxiety can explain total variance of efficacy approximately 15% of the time.

Step 8: Scatter plot and regression line

Now that we know this model is significant, the next step is to include the regression line into the scatter plot we already have developed. The first step is to repeat step 5 and then include the **abline** (best-fit line). Commands for both are given below along with the output.

```
plot(GAD7Total, GSESTotal...15, main = "General Anxiety and Self-Efficacy", col =
"darkblue", xlab = "General Anxiety", ylab = "Self-Efficacy")

abline(mod)
```

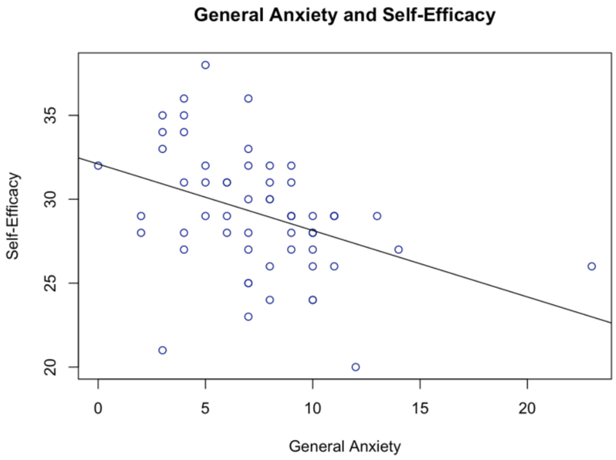


Figure 35: Scatter plot of general anxiety and self-efficacy with regression line

Step 8: ANOVA table for Linear Regression

The final step remaining is to general the conventional ANOVA table. Which can be obtained by adding the command given below. As per this output, you can easily interpret the output of the linear model.

```
anova(mod)
```

```
Analysis of Variance Table

Response: GSESTotal...15
             Df    Sum Sq    Mean Sq    F value    Pr(>F)
GAD7Total     1    118.68    118.680    10.612     0.001896 **
Residuals    57    637.49     11.184
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Voila! Now you know how to run a simple linear regression as well.

17.0 Confirmatory Factor Analysis (CFA)

I feel this book is incomplete if I do not discuss CFA. So, in this section I would explain how to run a CFA as briefly as possible. When we culturally adapt and statistically validate a psychometric scale, many researchers first run EFA which is more suitable in early stages of research to explore the internal structure of a scale. However, if the adapted scale was already established as a valid psychometric tool in other languages or countries, researchers can go ahead with a CFA as well without necessarily going for an EFA. If I simply put it, if the researcher already hold a hypothesis about the factor structure of the scale, CFA is the ideal method to proceed.

Now, to run CFA, I am going to install **Lavaan** package. Unlike other package installation, here the code is slightly different. The function, **install.packages()** remain the same, but an additional component is added. After running the command, you may check whether it is appropriately installed by the **library()** function.

```
install.packages("lavaan", dependencies = TRUE)
library(lavaan)
```

In case you are starting to go through this book from this chapter, you may have to install the below listed packages as well. The package **semPlot** is required to visualize the path diagram of the CFA.

```
install.packages("lavaanplots")
install.packages("tidySEM")
install.packages("ggplot2")
install.packages("dplyr")

install.packages("semPlot")
library(semPlot)
```

Once these packages are installed, we need to add the below given codes to obtain the model fit indexes to evaluate our CFA. Here, I have selected the GAD7 data from the same dataset I have used for EFA and IRT in the previous sections. To get the CFA for GAD7, first we need to state what our model is. In the below given command, I have stated that my factor 1, which is the only factor I have contains all the 7 items and they are all fed to a new variable named 'GADModel.'

```
GADModel <- 'factor1 =~ GADQ1 + GADQ2 + GADQ3 + GADQ4 + GADQ5 + GADQ6 +
GADQ7'
fit <- cfa(GADModel, data = Cleaned_Data)
summary(fit, fit.measure = TRUE, standardized = TRUE)
```

Once the first line of the command is entered, then we can enter the command to test the model fit, and then finally obtain the summary of fit conducted. The resulting output is given in the next page.

First, let's go through the output, and then let's try to interpret this.

```
Estimator                                         ML
Optimization method                           NLMINB
Number of model parameters                        12

Number of observations                            59

Model Test User Model:
    Test statistic                            24.841
    Degrees of freedom                             9
    P-value (Chi-square)                       0.003

Model Test Baseline Model:
    Test statistic                            55.786
    Degrees of freedom                            15
    P-value                                    0.000

User Model versus Baseline Model:
    Comparative Fit Index (CFI)                0.612
    Tucker-Lewis Index (TLI)                   0.353

Loglikelihood and Information Criteria:
    Loglikelihood user model (H0)           -455.755
    Loglikelihood unrestricted model (H1)   -443.335
    Akaike (AIC)                             935.510
    Bayesian (BIC).                          960.441
    Sample-size adjusted Bayesian (SABIC)    922.704

Root Mean Square Error of Approximation:
     RMSEA                                     0.173
     90 Percent confidence interval - lower   0 .094
     90 Percent confidence interval - upper    0.255
     P-value H_0: RMSEA <= 0.050               0.009
     P-value H_0: RMSEA >= 0.080               0.970

Standardized Root Mean Square Residual:
     SRMR                                      0.118

Parameter Estimates:
     Standard errors                        Standard
     Information                            Expected
     Information saturated (h1) model     Structured
```

To understand whether this CFA model is a good model, we have to interpret some of the statistics given above. First, we should check the chi-square test. The *p* value of chi-square should not be significant. This *p* value is sensitive to sample sizes. Larger sample sizes generally tend to make the chi-square significant. In such situations chi-square statistic (χ^2) is divided by degrees of freedom (*df*) and if that does not exceed 3, then the model is acceptable.

That would be, χ^2/df = 24.841/9 = **2.76**

In our output, *p* value is significant which indicates the model to be a bad fit. However, when the above mentioned calculation was performed, the data seem to fit the proposed unidimensional model. However, assessing the chi-square output alone is not adequate. We usually should proceed toward alternative indexes such as Comparative Fit Index (CFI), Tucker Lewis Index (TLI), Standardized Root Mean Square Residual (SRMR), and Root Mean Square Error of Approximation (RMSEA).

CFI and TLI values should be close to or higher than .95. However, values from .90 are also acceptable. In our output, CFI is much lesser compared to the accepted range and TLI also significantly falls short. Similarly, SRMR and RMSEA test the degree to which our model does not fit the data. SRMR is based on factor loadings. The maximum one could obtain here is 1, and higher the values worse the fit. Here, values that are below .08 are acceptable. RMSEA is also similar to SRMR. In RMSEA, .06 is preferred and below .08 is considered acceptable. Considering the output we have obtained, SRMR and RMSEA also indicate our model to be a poor fit. So, as per the results we have obtained, this is not a good model.

The interpretation guidelines briefed above are elaborated in depth by Kyndt & Onghena (2014) in the book *Discourse on professional learning: On the boundary between learning and working*.

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)  Std.lv  Std.all
 factor1 =~
  GADQ1              1.000                               0.457   0.584
  GADQ2              1.543    0.557    2.769    0.006    0.705   0.769
  GADQ3              0.908    0.399    2.277    0.023    0.415   0.394
  GADQ4              0.425    0.315    1.347    0.178    0.194   0.216
  GADQ6              0.521    0.350    1.490    0.136    0.238   0.241
  GADQ7              0.768    0.333    2.305    0.021    0.351   0.400

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)  Std.lv  Std.all
 .GADQ1              0.404    0.102    3.949    0.000    0.404   0.659
 .GADQ2              0.344    0.170    2.017    0.044    0.344   0.409
 .GADQ3              0.938    0.188    4.987    0.000    0.938   0.845
 .GADQ4              0.771    0.145    5.321    0.000    0.771   0.953
 .GADQ6              0.922    0.174    5.291    0.000    0.922   0.942
 .GADQ7              0.646    0.130    4.969    0.000    0.646   0.840
  factor1            0.209    0.110    1.897     0.058   1.000   1.000
```

Once the initial model is interpreted we have two more tables to interpret. This include how each item relates to the given latent variable and their residual variances (check definition). As per the first table (measures each item's relationship to the latent construct), some items are strongly related to the latent variable. For example, GADQ2 has a higher unstandardized loading of 1.543, and a standardized loading of .769. GADQ2 also has a higher z-value and it is statistically significant (denoted by P(>|z|)). GADQ3, although has a weaker loading, is still significantly associated to the latent factor identified (factor1). GADQ7 is also highly significant. You would notice the relationships are weak for GADQ4 and GADQ6. You will also notice that I have not include GADQ5. This was a mistake from my end (not that I deliberately excluded it). When you run the analysis, you can include GADQ5 and see whether that has an overall effect on the model fit indexes. In some sense, my mistake can create a different type of a learning experience.

Of the two tables given above, in the first table, you will notice that GADQ1's factor loading is fixed at 1.00. This is a standard practice in CFA in which one item if fixed so that other items can be interpreted relative to that. I have given a better explanation of this standard practice in the next section. So, the interpretation we already have done about all GAD7 items are done in relation to GADQ1 which is fixed at 1.00.

Now, let's move to the second table that gives us information about residual variances. Here, GADQ4 has the highest residual variance at .771. What this means is that GADQ4 has quite a lot of error and explains only a little of its intended variance. Similarly, if you notice, for all items the residual variances are relatively high. This might have caused the poor model fit.

I hope with the above description you have some clarity about how we interpret the CFA. After this process, we can also visualize the CFA using a path diagram (Figure 35). For this you should have installed **semPaths** and run the below given code.

```
semPaths(fit, what = "std", layout = "tree", style = "lisrel", edge.label.cex = 1, edge.color =
"black", sizeMan = 5, sizeLat = 7, rotation = 2)
```

Given below is the out.

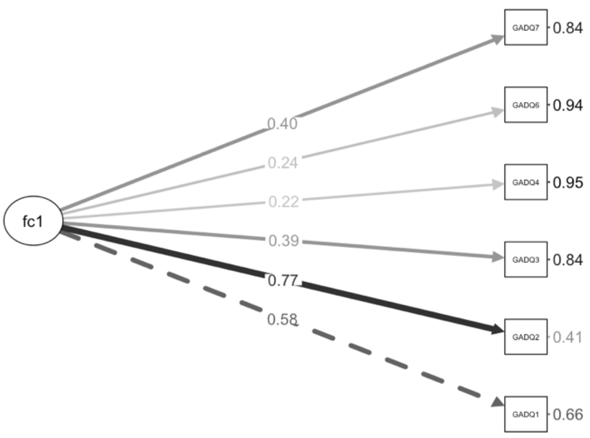


Figure 36: Path diagram of confirmatory factor analysis for general anxiety

In the above figure, you can clearly see how each item is loaded under factor 1. The fixed item is indicated by the dotted line. Similar to GAD7, I also ran a CFA for the Sinhala version of the generalized self-efficacy scale (SGSES). I have given the commands and the output here for your reference. Please note that this is not based on the data we have used in the previous parts of the book. The below given CFA is based on the data collected by one of my students named *Lahiru Pothmulla* during his BSc dissertation. I reanalysed his data to create this CFA.

```
SGSES <- 'factor1 =~ Q1 + Q2 + Q3 + Q4 + Q5 + Q6 + Q7 + Q8 + Q9 + Q10'
modelfit <- cfa(SGSES, data = SGSES_Data)
summary(modelfit, fit.measure = TRUE, standardized = TRUE)
```

```
semPaths(modelfit, what = "std", layout = "tree", style = "lisrel", edge.label.cex = 1,
edge.color = "black", sizeMan = 5, sizeLat = 7, rotation = 2)
```

```
Estimator                                         ML
  Optimization method                         NLMINB
  Number of model parameters                      20

                                        Used   Total
Number of observations                   120     121

Model Test User Model:

Test statistic                                89.276
  Degrees of freedom                              35
  P-value (Chi-square)                         0.000

Model Test Baseline Model:
  Test statistic                             295.204
  Degrees of freedom                              45
  P-value                                      0.000

User Model versus Baseline Model:
  Comparative Fit Index (CFI)                  0.783
  Tucker-Lewis Index (TLI)                     0.721

Loglikelihood and Information Criteria:
  Loglikelihood user model (H0)            -1104.858
  Loglikelihood unrestricted model (H1)           NA
  Akaike (AIC)                              2249.716
  Bayesian (BIC)                            2305.466
  Sample-size adjusted Bayesian (SABIC)     2242.235

Root Mean Square Error of Approximation:
  RMSEA                                        0.114
  90 Percent confidence interval - lower       0.085
  90 Percent confidence interval - upper       0.143
  P-value H_0: RMSEA <= 0.050                  0.000
  P-value H_0: RMSEA >= 0.080                  0.971

Standardized Root Mean Square Residual:
  SRMR                                         0.086

Parameter Estimates:
  Standard errors                           Standard
  Information                               Expected
  Information saturated (h1) model        Structured
```

In the above output, chi-square test is significant. CFI and TLI scores fall short of the identified threshold of .90. SRMR is acceptable, but RMSEA is unacceptable. Overall, it seems the model is not a good fit. As per the two tables given below, all questions are significantly related to the latent variable. Here, Q1 is constrained and fixed and as a result *p* value cannot be directly retrieved as it is not estimated. Similar to GAD7, all items of SGSES have higher residual variances indicating higher error component in all items. The highest residual variance is noticed in Q4.

```
Latent Variables:
              Estimate  Std.Err  z-value   P(>|z|)   Std.lv    Std.all
 factor1 =~
   Q1           1.000                       0.266     0.480
   Q2           1.075    0.303    3.553     0.000     0.286     0.445
   Q3           1.591    0.387    4.115     0.000     0.423     0.569
   Q4           1.207    0.341    3.534     0.000     0.321     0.442
   Q5           1.008    0.297    3.394     0.001     0.268     0.416
   Q6           1.348    0.323    4.175     0.000     0.359     0.585
   Q7           1.809    0.417    4.333     0.000     0.481     0.630
   Q8           1.324    0.324    4.089     0.000     0.352     0.562
   Q9           1.785    0.417    4.277     0.000     0.475     0.613
   Q10          1.055    0.279    3.784     0.000     0.281      0.491

Variances:
             Estimate   Std.Err   z-value   P(>|z|)   Std.lv    Std.all
  .Q1         0.237     0.033     7.170     0.000     0.237     0.770
  .Q2         0.331     0.046     7.270     0.000     0.331     0.802
  .Q3         0.375     0.055     6.821     0.000     0.375     0.676
  .Q4         0.426     0.058     7.279     0.000     0.426     0.805
  .Q5         0.344     0.047     7.343     0.000     0.344     0.827
  .Q6         0.248     0.037     6.741     0.000     0.248     0.658
  .Q7         0.352     0.054     6.469     0.000     0.352     0.603
  .Q8         0.269     0.039     6.853     0.000     0.269     0.684
  .Q9         0.374     0.057     6.578     0.000     0.374     0.624
  .Q10        0.248     0.035     7.132     0.000     0.248     0.759
  factor1     0.071     0.028      2.505    0.012     1.000      1.000
```

Similar to the previous instance, once again the factor structure was visualized using the below given code with the semPath package.

```
semPaths(fit, what = "std", layout = "tree", style = "lisrel", edge.label.cex = 1, edge.color =
"black", sizeMan = 5, sizeLat = 7, rotation = 2)
```

The output is displayed in the next page.

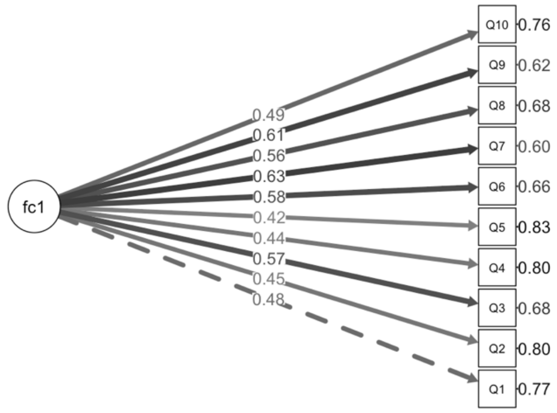


Figure 37: Path diagram of confirmatory factor analysis for Sinhala general self-efficacy

We have now a comprehensive idea about how to run CFA. However, I think I still owe you a good explanation on fixing one of the items in a CFA model and the logic behind this process. Fixing one item's factor loading (λ) to a 1.000 is a standard practice. There are a few reasons for fixing one item at 1.000.

1. In CFA, if at least one item is not fixed at 1.000 then the model is said to be under-defined and it would be impossible to uniquely estimate the parameters.
2. A latent variable is an abstract concept that cannot be measured directly. As a result, by fixing the loading of one item to 1.000, we set a reference point which helps us define the latent variable by measuring all the items relative to that.

The fixed loading means that for every 1-unit increase in the latent variable, the observed item is expected to increase by exactly 1 unit, assuming no error. Similar to factor loadings, the residual variance in CFA for each item (θ) is also estimated. This residual variance represents the part of the item's variance that is not explained by the latent factor. The sum of the factor loading squared times the latent variance and the residual variance equals the total variance of the item. The equation for this can be written as follows.

Total variance = λ^2 * var(Factor) + θ

Generally, var(Factor) is kept at 1.000. If the resulting total variance is greater than 1 for an item, it indicates the item to have greater variability either due to the influence of latent factors or due to error variance. For instance, Q2 of SGSES has a bigger factor loading. This makes the total variance of the item larger than 1 indicating higher variance. This could also be true for the rest of the items considering the larger nature of the factor loadings.

With the above explanations, we reach the end of this book. I am pretty sure now you have good understanding about the basic concepts in psychometrics. If you still have not watched the lesson videos, you can take this time to view them as well. Watching videos will further help you to master these lessons effectively.

This page is intentionally left blank

Definitions

1D array:
A 1D array is a linear collection of data elements, such as a list of numbers or characters in a given row.

2D array:
A 2D array is a two-dimensional structure that stores data in rows and columns, forming a grid or table, commonly used for tables of data.

ANOVA:
Analysis of variance is a parametric statistical test that helps researchers assess the significance of difference between 2 or more groups (Gravetter & Wallnau, 2017).

A priori:
Reasoning independent of sensory experiences.

A posteriori:
Reasoning dependent of sensory experiences.

Binary variables:
Variables with two options (i.e., Yes or No, A or B, etc.).

Bivariate:
An analysis that involves 2 variables. This is commonly observed in correlations and simple linear regression.

Categorical variables:
Categorical variables represent data that can be divided into distinct categories or groups, variables like gender, colors, type of vehicle brand, and yes/no responses are all categorical data. We also call yes/no type of answers dichotomous responses (Azam *et al.,* 2021; Gravetter & Wallnau, 2017).

Cohen's *d*:
A form of assessing effect size. It considers difference of means of two groups and present it in standard deviation units. Values above 0.5 indicate substantial effect size while 0.8 and above indicate large effect size.

Coefficient of determination (R^2):
A form of effect size that helps researchers assess how much variance is accounted by the predictor variables. In a correlation, this says the degree to which **X** explains **Y**, and vice versa. This is a form of effect size calculation that you will find in other statistical analyses including linear regression.

Constructs:
Constructs refer to theoretical concepts or ideas that researchers aim to measure or study. They are often abstract and may involve multiple variables. One has to develop an **operational definition** (a method of making a construct measurable) to convert it to a variable so that it can be measured.

Contingency table:
A contingency table is used to display the frequency of two or more categorical variables' combinations, helping to analyze relationships between them.

Continuous variables:
Continuous variables are those that can take any real value within a range, such as height or weight, and can have an infinite number of possible values. The opposite of continuous variables is categorical or discreet variables.

Data wrangling:
Data wrangling involves the process of cleaning, transforming, and organizing data to make it suitable for analysis.

Data tidying:
Data tidying refers to the practice of structuring datasets to make them more organized, consistent, and ready for analysis using specific conventions.

Debugging:
Debugging is the process of identifying and fixing errors or issues in computer code or scripts, ensuring they run correctly.

Delphi process:
A process used to assess consensus of Subject Matter Experts (SMEs) in culturally adapting a psychometric tool. Delphi process is often used to evaluate content and consensual validity of a psychometric tool (Selvaratnam *et al.,* 2024).

Dichotomous:
Two options (binary). An item that belongs to a scale with two answer options such as *Yes* or *No* is called an item with dichotomous options.

Effect size:
Effect size measures the strength of a relationship or the magnitude of an effect in statistical analysis, often used in hypothesis testing. Commonly used effect sizes include R^2, Cohen's *d,* and η^2.

Error variance:
Error variance is the portion of total variance in observed scores that is attributed to measurement error rather than the true variance in the construct being measured. Error variance include random fluctuations and inconsistencies in a tool that can arise due to a myriad of reasons (i.e., biased sample, issues in scale administration, respondent factors, etc.). Error variance could also result due to problems in the scale itself (this should be corrected by rigorously testing for validity). Error variance is derived from the CTT equation, **X = T + E**. Here, **X** is the observed score, **T** is the true score, and **E** is the error. Error variance therefore is the variability in **E** indicating how much **X** deviate from **T** due to measurement error.

Eta squared:
A form of effect size represented by η^2.

Expert review:
A chosen panel of experts helping obtain content validity in culturally adapting a psychometric scale. This expert review could also be the Delphi process.

Factor:
In the context of psychometrics, a factor refers to an underlying variable (latent variable) that could explain the correlations among a set of observed variables. In the context of parametric testing, a factor is an independent variable with multiple groups or levels.

General Linear Model:
The General Linear Model (GLM) is a statistical framework used to analyze relationships between variables, including linear regression, ANOVA, and ANCOVA.

Generalized Linear Model (GLM):
The Generalized Linear Model (GLM) extends the GLM to handle a broader range of data types and distributions, such as binomial or Poisson.

Item Response Theory (IRT):
Item Response Theory (IRT) is a statistical model used in educational and psychological testing to evaluate the difficulty and discrimination of test items. Rasch model, graded response model (GRM), etc. are commonly used statistical methods in IRT.

Jamovi:
A software designed for statistical analysis. Additional modules can be downloaded and can be integrated with *R* commands. Offers a convenient and efficient user interface.

Kruskal Wallis Test:
A non-parametric equivalent for one-factor analysis of variance.

Latent traits:
Latent traits are unobservable characteristics or attributes that can influence or underlie observed behaviours or responses. Researchers may identify these as constructs. For example, the 10-item generalized self-efficacy scale measures the latent variable, self-efficacy.

Latent Trait Modeling (LTM):
Latent Trait Modeling (LTM) is a statistical approach used to estimate latent traits and their relationships with observed variables.

Leptokurtic:
Positive kurtosis with fatter tails and sharper peak.

Mesokurtic:
Zero kurtosis with tails and peaks similar to a normal distribution.

Metadata:
Metadata is data that describes other data, providing information about the content, structure, and context of datasets.

Multicollinearity:
When items of a scale display high intercorrelations (0.8 and above) it is considered as multicollinearity.

Multidimensional:
Psychometric scales that has more than one dimension (i.e., DASS-21 has 3 dimensions, namely, depression, anxiety, and stress).

Multivariate:
In the context of group difference testing, when more than one dependent variable is used it is commonly identified as a multivariate analysis. However, some scholars refer to any analysis with more than 1 variable as multivariate as well. So, based on the context, the usage of this word may vary slightly.

Nested model:
In the context of structural equation modeling (SEM), a nested model is a model that is more constrained or simplified version of a more complex model. This means that the nested model can be obtained by imposing additional restrictions. For example, if Model A includes all possible paths and relationships among variables, and Model B is obtained by removing or constraining some of those paths, the Model B is nested within Model A.

Normal distribution:
A bell-shaped curve in which the left and the right sides of the curve are mirror images (thus symmetrical) and contains mean, median, mode at the centre of the distribution. In terms of the spread, about 68% of data falls within 1SD, 95% within 2 SD, and 99.7 within 3 SD. A normal distribution is asymptotic (the tails of the curve approach horizontal axis but never touch it, extending infinitely in both directions.

Omega squared (ω^2):
A measure of effect size for ANOVA designs.

Partial Eta Squared:
In factorial ANOVA, you will observe both η^2 and partial η^2. In such context, η^2 demonstrated total variance attributable for all independent variables, while the partial η^2 displays variance that is attributable to a single independent variable.

Path analysis:
A form of statistical analysis to display causal relationships between variables.

Platykurtic:
Negative kurtosis with thinner tails and flatter peak compared to a normal distribution.

Polytomous:
More than two. If an item that belongs to a scale has more than two answer options (i.e., Likert scale), such items are polytomous.

R Console:
The *R* Console is an interface in *R* Studio where you can enter *R* commands and see their output, making it a hub for working with R.

R Studio:
R Studio is an integrated development environment (IDE) for *R* that provides tools and features for data analysis, visualization, and code development.

Residual variance:
In the context of CFA, residual variance is the portion of total variance for an item that is not accounted for the latent variable. Residual variance can also be considered as error variance. In the context of regression, residual variance is the difference between observed values and predicted values in a regression model (Gravetter & Wallnau, 2017).

Singularity:
When items of a scale display perfect correlations (r = +/- 1.00) it is considered a singularity.

Sphericity:
In the context of factor analysis, sphericity ensures the data has enough shared variance for factor analysis to be appropriate. Sphericity is an essential element that determines the factorability of a scale.

SPSS:
Statistical Package for Social Sciences, the go-to statistical software for psychology students. I guess I do not even need to give a description.

SPSS Amos:
An extension to SPSS for researchers to conduct confirmatory factor analysis (CFA) and structural equation modeling (SEM). This helps researchers understand complex relationships between multiple variables.

Standard deviation:
Measures the amount of dispersion in a set of values. It quantifies how much the values in a data set differ from the mean (average) of the data set. Low SD indicates that values are close to the mean, while a high SD indicates that values are spread out over a wider range. SD is the square root of variance.

Statistical significance:
Statistical significance indicates whether an observed effect in data is likely to be real or just due to chance, often determined through hypothesis testing (this gives you the *p.* value). This *p* value should be less than 0.05 to affirm statistical significance.

Structural Equation Modeling (SEM):
Structural Equation Modeling (SEM) is a statistical technique used to analyze complex relationships among variables, often involving multiple latent constructs.

Subject Matter Expert (SME):
Experts in any field. For example, to conduct a Delphi process, one should seek a group of experts (i.e., psychologists and other health professionals).

T-Test:
A parametric statistical test that can identify the significance of the difference between two means that belongs to two groups.

Unidimensional:
A scale in which all items account for a single dimension or factor is called a unidimensional scale.

Univariate:
In the context of ANOVAs, having one dependent variable is what is meant as univariate analyses. However, the term bivariate does not mean using two dependent variables in a MANOVA. Bivariate is a term commonly considered in correlations and regressions. MANOVAs by default are multivariate since there is more than 1 dependent variable in the study.

Wilcoxon Test:
A non-parametric version of the T-Test.

This page is intentionally left blank

References

1. Azam, S. M. F., Yajid, M. S. A., Tham, J., Hamid, J. A., Khatibi, A., Johar, M. G. M. Arrifin,I. A. (2021). *Research methodology: Building research skills*, McGraw Hill Education (Malaysia) Sdn Bhd, Kuala Lumpur: Malaysia

2. Babu, N. & Kohli, P. (2023). Commentary: Reliability in research, *Indian Journal of Ophthalmology, 71*(2), 400-401. http://doi.org/10.4103/ijo.IJO_2016_22

3. Baker, F. B. (2001). *The Basics of Item Response Theory.* ERIC Clearinghouse on Assessment and Evaluation.

4. Cohen, A. S., Kim, S., & Baker, F. K. (1993). Detection of differential item functioning in the Graded Response Model. *Applied Psychological Measurement, 17*(4), 335-350.

5. Costello, A., & Osborne, J. (2005). Best Practices in Exploratory Factor analysis: Four Best Practices in Exploratory Factor analysis: Four Recommendations for Getting the Most from Your Analysis Recommendations for Getting the Most from Your Analysis. *Research, and Evaluation Practical Assessment, Research, and Evaluation*, *10*(7), 7. https://doi.org/10.7275/jyj1-4868

6. Cronbach, L. J. (1964). The signal/noise ratio in the comparison of reliability coefficients, *Educational and Psychological Measurement, XXIV*(3), 467-480.

7. DeCarlo, L. T. (1997). On the meaning and use of kurtosis, *Psychological Methods, 2*(3), 292-307.

8. Duhachek, A. & Iacobucci, D. (2004). Alpha's standard error (ASE): An accurate and precise confidence interval estimate, *Journal of Applied Psychology, 89*(5), 792-808. https://doi.org/10.1037/0021-9010.89.5.792

9. Duhachek, A., Coughlan, A. T., Iacobucci, D. (2005). Results on the standard error of the coefficient alpha index of reliability, *Marketing Science, 24*(2), 294-301. https://doi.org/10.1287/mksc.1040.0097

10. Field, A. (2000). *Discovering statistics using IBM SPSS Statistics.* SAGE Publications Inc.

11. Gorsuch, R. L. (1983). *Factor analysis* (2nd ed.). Psychology Press. https://doi.org/10.4324/9780203781098

12. Gravetter, F. J., & Wallnau, L. B. (2017). *Statistics for the behavioral sciences, Tenth Edition*. Boston, USA: Cengage Learning.

13. Gyamfi, A. & Acquaye, R. (2023). Parameters and models of Item Response Theory (IRT): A review of literature, *Acta Educationis Generalis, 13*(3), 68-78. https://doi.org/10.2478/atd-2023-0022

14. Iacobucci, D. & Duhachek, A. (2003). Advancing alpha: Measuring reliability with confidence, *Journal of Consumer Psychology, 13*(4), 478-487.

15. Isernia, S., Rossetto, F., Shamay-Tsoory, S., Marchetti, A., Bagilo, F. (2023). Standardization and normative data of the 48-item Yoni short version for the assessment of theory of mind in typical and atypical conditions, *Frontiers in Aging Neuroscience.* https://doi.org/10.3389/fnagi.2022.1048599

16. Kean, J., Brodke, D. S., Biber, J., Gross, P. (2018). An introduction to Item Response Theory and Rasch analysis of the Eating Assessment Tool (EAT-10), *Brain Impairment, 19*(1), 91-102. https://doi.org/10.1017/Brlmp.2018.31

17. Kyndt, E. & Onghena, P. (2014). The integration of work and learning: Tackling the complexity by means of structural equation modeling, In *Discourses on professional learning: On the boundary between learning and working*, eds by Harteis C., Rausch A., and Seifried J. Springer.

18. Landis, J. R. & Koch, G. G. (1977). The measurement of observer agreement for categorical data, *Biometrics, 33,* 159-174.

19. Liao, W., Ho, R., Yen, Y. (2012). The four-parameter logistic item response theory model as a robust method of estimating ability despite aberrant responses, *Social Behavior and Personality, 40*(10), 1679-1694. http://dx.doi.org/10.2224/sbp.2012.40.10.1679

20. Masuwai, A., Zulkifli, H., Hamzah, M. I. (2024). Evaluation of content validity and face validity of secondary school Islamic education teacher self-assessment instrument, *Cogent Education, 11*(1). https://doi.org/10.1080/2331186X.2024.2308410

21. Matheson, G. J. (2019). We need to talk about reliability: Making better use of test-retest studies for design and interpretation, *PeerJ,* http://doi.org/10.7717/peerj.6918

22. Memon, A. M., Ting, H., Cheah, J.-H., Thurasamy, R., Chuah, F., & Cham, T. (2020). Sample size for survey research: Review and recommendations. *Journal of Applied Structural Equation Modeling*, *4*(2), i–xx. https://jasemjournal.com/wp-content/uploads/2020/08/Memon-et-al_JASEM_-Editorial_V4_Iss2_June2020.pdf

23. Mishra, P., Pandey, C. M., Singh, U., Gupta, A., Sahu, C., Keshri, A. (2019). Descriptive statistics and normality tests for statistical data, *Annals of Cardiac Anaesthesia, 22,* 67-72. http://dx.doi.org/10.4103/aca.ACA_157_18

24. Mohsen, T. & Dennick, R. (2011). Making sense of Cronbach's alpha, *International Journal of Medical Education, 2,* 53-55. http://doi.org/10.5116/ijme.4dfb.8dfd

25. Mueller, S. T. (2023). *Item Response Theory: Advanced Psychometrics Using Item Response Theory, the Rasch Model, and Related Concepts.* Michigan Tech https://pages.mtu.edu/~shanem/psy5220/daily/Day10-11/irt.html

26. Osborne, J. W. (2015). What is rotating in exploratory factor analysis? *Practical Assessment, Research and Evaluation, 20*(2).

27. Pronk, T., Molenaar, D., Wiers, R. W., Murre, J. (2022). Methods to split cognitive task data for estimating reliability: A comprehensive review and systematic assessment, *Psychonomic Bulletin & Review, 29,* 44-54. https://doi.org/10.3758/s13423-021-01948-3

28. Scholz, U., Dona, B. G., Sud, S., Schwarzer, R. (2002). Is general self-efficacy a universal construct?, *European Journal of Psychological Assessment, 18*(3), 242-251. https://doi.org/10.1027/1015-5759.18.3.242

29. Schwarzer, R., Jerusalem, M., Johnston, M. (1995). Generalized self-efficacy scale, *Measures in Health Psychology: A user's portfolio.*

30. Selvaratnam, N. C., Selvaratnam, N. D., Nanayakkara, A. M. N. A. D. J. S., Tennakoon, S. (2024). Household Water Insecurity Experiences (HWISE) Scale: The protocol of cultural adaptation and statistical validation, *European Journal of Public Health Studies. 7*(1), 33-53. http://dx.doi.org/10.46827/ejphs.v7i1.166

31. Selvaratnam, N. D., & Selvaratnam, N. C. (2024, December 4). Assessment of Judgmental Validity of the Sinhala Household Water Insecurity Experiences (HWISE) Scale. *SLIIT International Conference on Advancements in Sciences and Humanities*.

32. Shrestha, N. (2021). Factor analysis as a tool for survey analysis, *American Journal of Applied Mathematics and Statistics, 9*(1), 4-11. https://doi.org/10.12691/ajams-9-1-2

33. Shrotryia, V. K. & Dhanda, U. (2019). Content validity of assessment instrument for employee engagement, *SAGE Open,* 1-7.

34. Singh, A. S. & Masuku, M. B. (2014). Assumption and testing of normality for statistical analysis, *American Journal of Mathematics and Mathematical Sciences, 3*(1), 169-175.

35. Smythe, R. (n.d.). *Item Response Theory for Polytomous Items.* Western University. https://www.uwo.ca/fhs/tc/labs/12.PolytomousIRT.pdf

36. Spitzer, R. L., Kroenke, K., Williams, J. B. W., Lowe, B. (2006). A brief measure for assessing general anxiety disorder: the GAD-7. https://doi.org/10.1001/archinte.166.10.1092

37. UCLA (2024). Principal components (PCA) and exploratory factor analysis (EFA) with SPSS, *UCLA Advanced Research Computing: Statistical Methods and Data Analytics.* Online: https://stats.oarc.ucla.edu/spss/seminars/efa-spss/#:~:text=The%20eigenvalue%20represents%20the%20communality,components%20is%20the%20total%20variance.

38. Williams, R. (2023). Scalar measures of fit: Pseudo R^2 and Information Measures (AIC and BIC). https://www3.nd.edu/~rwilliam/stats3/L05.pdf

39. Yusoff, M. S. B. (2019). ABC of content validation and content validity index calculation, *Education in Medicine Journal, 11*(2), 49-54. https://doi.org/10.21315/eimj2019.11.2.6

This page is intentionally left blank

R Script for EFA

```
#loading all required packages
library(pacman)
pacman::p_load(pacman, dplyr, GGally, ggplot2, ggthemes, ggvis, httr, lubridate, plotly, rio,
rmarkdown, shiny, stringer, tidyr)
pacman::p_load(pacman, dplyr, psych)

#Summary stats of the Age
summary_stats <- describe(GSESData$Age)
print(summary_stats)

#Summary stats of GSES Total
summary_GSES <- describe(GSESData$FinalScore)
print(summary_GSES)

#Cronbach's Alpha
selected_items <- GSESData[, c("Item_1", "Item_2", "Item_3", "Item_4", "Item_5", "Item_6",
"Item_7", "Item_8", "Item_9", "Item_10")]
alpha(selected_items)

#KMO Index for sample adequacy
selected_items <- GSESData[, c("Item_1", "Item_2", "Item_3", "Item_4", "Item_5", "Item_6",
"Item_7", "Item_8", "Item_9", "Item_10")]
kmo_result <- KMO(selected_items)
print(kmo_result)

#Bartlett's Test of Sphericity
correlation_matrix <- cor(selected_items)
print(correlation_matrix)
sample_size <- 404
bartlett_test_result <- cortest.bartlett(correlation_matrix, n = sample_size)
print(bartlett_test_result)

#Factor Analysis
fac(selected_items, nfactors = 9, rotate = F)
F1 <- fac(selected_items, nfactors = 9, rotate = F)
print(F1$loadings, cutoff = 0.4)

#PCA#Alternative Method
pca(selected_items, nfactors = 9, rotate = F)
F1 <- pca(selected_items, nfactors = 9, rotate = F)
print(F1$loadings, cutoff = 0.5)
F1$communality

#Scree plot method I
fa_scores <- F1$scores
pca_result <- princomp(fa_scores, scores = TRUE, cor = TRUE)
screeplot(pca_result, type = "line", col = "blue", main = "Scree Plot")
```

```
#Scree plot method I
install.packages("nFactors")
library(nFactors)

ev <- eigen(cor(selected_items))
nS <- nScree(x=ev$values)
plotnScree(nS, legend = F, main = "Scree Plot")

#Factor Diagram
fac(selected_items, nfactors = 1, rotate = F)
F1 <- fac(selected_items, nfactors = 1, rotate = F)
fa.diagram(F1, main = "Factor Analysis")
#####Item Response Theory
install.packages("ltm")
library(ltm)
```

R Script for IRT GRM

#Item Response Theory Unconstrained

```
grm_model <- grm(selected_items)
summary(grm_model)
```

#Item Response Theory Constrained

```
grm_model2 <- grm(selected_items, constrained = TRUE)
summary(grm_model2)
```

#Item Response Theory Constrained

```
#constrained version should be included first in the brackets
anova(grm_model2,grm_model)
```

#Plot Item Response Characteristic Curves (IRCCs)

```
plot(grm_model)
```

#Plot Test Information Functions (TIFs)

```
plot(grm_model, type = "IIC", items = 0)

plot(grm_model, type = "IIC")
```

#Estimates of theta by examinee

```
est <- factor.scores(grm_model)
```

#First 10 examinees

```
est$score.dat$z1[1:10]
```

#Plot TIFs all together

```
par(mfrow = c(2, 5))
for (i in 1:10) {  # Assuming you have 10 items
    plot(grm_model, item = i, type = "IIC", xlab = "Theta",
            ylab = "Information", main = paste("Item_", i, sep = ""))
  }

for (i in 1:10) {  # Assuming you have 10 items
  plot(grm_model, item = i, type = "ICC", xlab = "Theta",
       ylab = "Information", main = paste("Item_", i, sep = ""))
}
```

#ConstrainedVsUnconstrained#CCCs

```
Constrained <- grm(selected_items, constrained = TRUE, start.val = 'random')
par(mfrow = c(2,3))
plot(Constrained, legend=TRUE)
summary(Constrained)

Unconstrained <- grm(selected_items, constrained = FALSE, start.val = 'random')
par(mfrow = c(2,3))
plot(Unconstrained, legend=TRUE)
summary(Unconstrained)
```

```
#ConstrainedVsUnconstrained#IICs
plot(Unconstrained, type = "IIC", lwd = 2, cex = 0.8, legend = TRUE,
    cx = "topleft", xlab = "SGSES", cex.main = 1, cex.lab = 1, cex.axis = 1)

plot(Constrained, type = "IIC", items = 0,
    lwd = 2, xlab = "SGSES", cex.main = 1, cex.lab = 1, cex.axis = 1)


#ModelFit#
anova(Constrained,Unconstrained)

#TestInformation##
information(Constrained, c(-4, 4))
information(Unconstrained, c(-4, 4))
```

R Script for Simple Linear Regression

```
##ScatterPlot##
plot(GAD7Total, GSESTotal...15, main = "General Anxiety and Self-Efficacy", col = "darkblue",
xlab = "General Anxiety", ylab = "Self-Efficacy")

##CorrelationTest##
cor(GAD7Total, GSESTotal...15)

##ModelSummary##
mod <- lm(GSESTotal...15 ~ GAD7Total)
summary(mod)

##RegressionModelPlot##
plot(GAD7Total, GSESTotal...15, main = "General Anxiety and Self-Efficacy", col = "darkblue",
xlab = "General Anxiety", ylab = "Self-Efficacy")

abline(mod)

##AnovaTest##
anova(mod)
```

R Script for Confirmatory Factor Analysis

```
##FactorIdentification##
SGSES <- 'factor1 =~ Q1 + Q2 + Q3 + Q4 + Q5 + Q6 + Q7 + Q8 + Q9 + Q10'

##ModelFit##
modelfit <- cfa(SGSES, data = SGSES_Data)
summary(modelfit, fit.measure = TRUE, standardized = TRUE)

##PathDiagram##
semPaths(modelfit, what = "std", layout = "tree", style = "lisrel", edge.label.cex = 1, edge.color =
"black", sizeMan = 5, sizeLat = 7, rotation = 2)
```

Results Reporting (Samples)

In the next set of pages, I have provided some sample reports for your reference. These reports include multiple statistical analyses and their correct method of reporting which includes correlations, reliability analysis, factor analysis, structural equation modeling, etc. For the most part, I have followed APA style results reporting. However, I have not kept line spacing doubled since it would drastically impact the aesthetics of my book. But, these reports should help you understand majority of the fundamentals in results reporting.

The reports will begin from the next page.

The samples are not necessarily from the datasets discussed in this book.

Sample 1: Descriptive report

The present study included *N* = 59 participants. The group indicate a relatively higher average for self-efficacy (*M* = 29.12, *SD* = 3.61). Further descriptive data of the study is reported in the table given below.

Table 15

Descriptive statistics of the self-efficacy variable

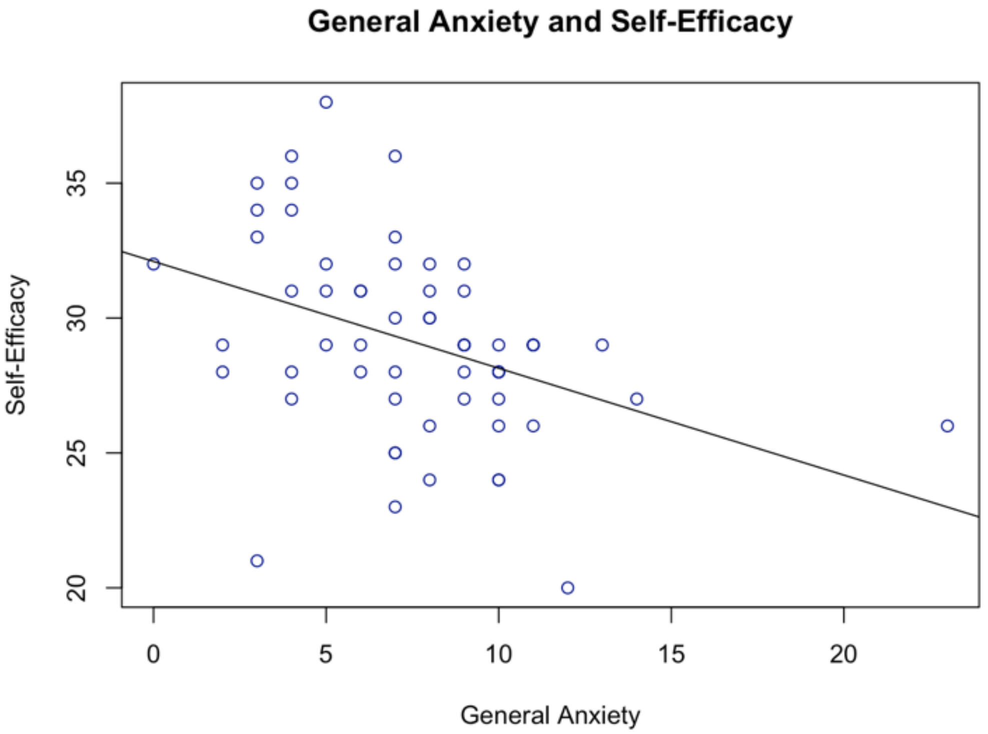
Type of information	Output
N	59
Mean	29.12
Standard deviation	3.61
Median	29
Trimmed mean	29.1
Median absolute deviation	2.97
Minimum	20
Maximum	38
Range	18
Skewness	0.01
Kurtosis	0.08
Standard error	0.47

The lowest reported score for self-efficacy in the present study is 20 while the highest recorded was 38. As per the skewness and kurtosis values obtained, the variable is normally distributed.

Sample 2 & 3: APA formatted figure and associated results reporting

Figure 38

A linear model demonstrating the relationship of self-efficacy and general anxiety



General anxiety and self-efficacy share a negative, yet a moderate correlation, *r* = -.3961685, R^2 = .1569. A simple linear regression between the two variables indicated the proposed relationship to be statistically significant, *F* (1, 57) = 10.612, *p* = .001896, R^2 = .1569. As per the results obtained, anxiety explains approximately 15% of the observed variance in self-efficacy.

Sample 4: Descriptive report

Upon completing basic frequency calculations, descriptive statistics were conducted. As per the provided conceptual framework, descriptive statistics were calculated for each variable: internationalization, business network, innovation, and balanced scorecard.

Table 16

Descriptive Statistics of Independent and Dependent Variables

	N	Mean	Std. Deviation	Variance	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
Internationalization Scores	404	44.3342	6.69726	44.853	-.161	.121	-.035	.242
Business Network	404	55.3837	6.70201	44.917	.606	.121	.752	.242
Innovation	404	20.9257	3.87387	15.007	.936	.121	3.064	.242
Balanced Scorecard	404	75.0792	7.64366	58.425	.234	.121	.982	.242

As per this table, all variables are normally distributed. However, “innovation” has a bit of a higher value for kurtosis, but the researcher continued with the rest of the analysis assuming the entire dataset is normally distributed.

Sample 5: Exploratory factor analysis

To observe the internal structural validity of all the items, exploratory factor analysis was considered. In doing that, first the ‘internationalization variable was chosen.

Table 17

KMO-Index and Bartlett’s Test of Sphericity for Internationalization

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.722
Bartlett's Test of Sphericity	Approx. Chi-Square	2442.407
	df	55
	Sig.	.000

Using the data provided, to check the internal structural validity of the variable ‘internationalization,’ a principal component analysis was run. Principal component analysis (PCA) is a form of dimension reduction. Prior to this, adequacy of sample must be evaluated through KMO index and suitability of the correlation matrix to undergo a PCA must be assessed through Bartlett’s Test of Sphericity. As per the findings, the sample is adequate for PCA, KMO = .772, and the current correlation matrix significantly diverges from an identity matrix, $\chi^2 = 2442.407$, *df* =55, *p* = .00.

Table 18

Total Variance Explained by the Significant Factors

Total Variance Explained

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	3.931	35.733	35.733	3.931	35.733	35.733	3.002	27.295	27.295
2	2.709	24.627	60.360	2.709	24.627	60.360	2.862	26.021	53.316
3	1.112	10.110	70.470	1.112	10.110	70.470	1.887	17.154	70.470
4	.796	7.239	77.709						
5	.653	5.937	83.646						
6	.530	4.822	88.468						
7	.343	3.121	91.589						
8	.316	2.877	94.466						
9	.300	2.729	97.195						
10	.212	1.929	99.124						
11	.096	.876	100.000						

Extraction Method: Principal Component Analysis.

The *Table 17* given above demonstrates the conducted PCA. Kaiser criterion was applied where eigenvalues above 1.0 were considered as significant factors. Further, each factor was considered based on the number of factor loadings it contains and based on the degree of variance each factor explains. As per Scholz et al. (2002), each factor above eigenvalue 1.0 must also have at least 3 factor loadings, and should explain 5% of variance. As per the findings of the above table, there are 3 significant factors as per the applied Kaiser criterion, and each explains more than 5%. The biggest factor explains 35.733% of total variance of the latent construct.

Since there are 3 major factors, varimax rotation was applied, and based on the findings (*Table 18*), three components were retained. Thus, 'internationalization' variable has 3 dimensions named: KN, TR, and CM. The rotated factors are displayed in the *table XYZA*.

Table 19

Rotated Component Matrix: Varimax Rotation

	Component		
	1	2	3
KN1	.869		
KN2	.864		
KN3	.859		
KN4	.843		
TR1		.858	
TR2		.861	
TR3		.897	
TR4		.530	
CM2			.646
CM4			.770
CM5			.780

Sample 6: Reliability analysis

After conducting the exploratory factor analysis, reliability was measured for each of the identified variables and its dimensions. Reliability was measured using Cronbach's alpha. The *Table 19* given below summarizes the findings.

Table 20

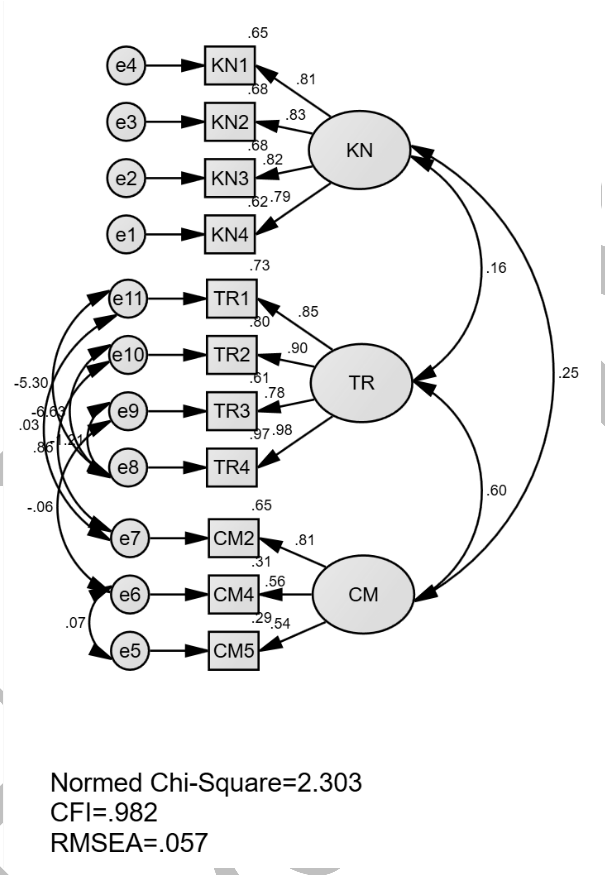
Reliability Analysis

Variable	Dimension	KMO Value	Cronbach's Alpha	Number of Items
Internationalization	KN	.722	.882	4
	TR		.848	4
	CM		.685	3
	Overall		**.815**	**11**
Business Network	STB	.822	.905	5
	CMP		.847	5
	EFF		.865	4
	Overall		**.778**	**14**
Innovation	Unidimensional	.864	.864	5
Balanced Scorecard	IBP	.813	.874	5
	LGP		.827	5
	FP		.885	4
	CP		.774	5
	Overall		**.781**	**19**
Full Scale	**4 Dimensions**	**N/A**	**.869**	**49**

Sample 7: Confirmatory factor analysis

After conducting EFA and reliability analysis, confirmatory factor analysis was conducted. CFA is also a form of factor analysis one could do once the structure of a scale or questionnaire is explored through EFA or identified prior. Thus, variables of the scale used in this study is used in a confirmatory factor analysis, and the results are provided below.

To accept a specific CFA model as an absolute fit, RMSEA must be less than 0.08. Incremental fit, CFI must be higher than 0.9. Further, parsimonious fit must be demonstrated by a Chi-square value less than 5.0.



Normed Chi-Square=2.303
CFI=.982
RMSEA=.057

Figure 39: *Confirmatory Factor Analysis of "Internationalization" Scale*

As per the findings of *figure 1*, the variable 'internationalization' demonstrates absolute fit, RMSEA = .057, incremental fit, CFI = .982, and parsimonious fit, Chi-square = 2.303. Based on these, CFA of 'internationalization' confirms a good model fit, and can be utilized in future to measure the specific latent variable in multiple individuals.

This page is intentionally left blank

Sample MCQ Paper (20 questions)

1. Which one of these software is **NOT** used for statistical analysis?
    a. ULead
    b. Jamovi
    c. SPSS
    d. JASP

2. Which one of these R Packages are used in finding Cronbach's alpha?
    a. GGally
    b. Psych
    c. tidyr
    d. semPaths

3. Which one of these variables is a discreet variable?
    a. Age
    b. Gender
    c. Anxiety
    d. Self-efficacy

4. Which one of these is **NOT** a method of assessing normality of a distribution?
    a. Kolmogorov-Smirnov test
    b. Histogram with normality curve
    c. Guttman Lambda (λ) 6
    d. Skewness and kurtosis

5. In the context of kurtosis, normal distributions are also considered as
    a. Leptokurtic
    b. Mesokurtic
    c. Platykurtic
    d. Regykurtic

6. Which one of these is bivariate?
    a. Simple linear regression
    b. Factor analysis
    c. Multiple regression
    d. Graded Response Model

7. A correlation between two binary variables is called
    a. Phi-coefficient
    b. Spearman correlation
    c. Polyphoric correlation
    d. Tetrachoric correlation

8. Which one of these is an alternative to Cronbach's alpha?
    a. Guttman Lambda (λ) 6 (G6)
    b. Fleiss Kappa
    c. Wilk's Lambda (λ)

   d. Eta squared η^2

9. Which one of these is **NOT** a measure of inter-rater reliability?
   a. Cohen's Kappa Statistic
   b. Fleiss Kappa Statistic
   c. Kuder-Richardson (KR) 20
   d. All of the above

10. A positive correlation between Beck's Depression Inventory (BDI) and Hamilton Depression Rating Scale (HAM-D) is an example of
    a. Content validity
    b. Consensual validity
    c. Internal structural validity
    d. Concurrent validity

11. Which one of these is a process used in factor analysis?
    a. KMO index
    b. Bartlett's test of sphericity
    c. Correlation matrix
    d. All of the above

12. In the context of factor analysis, sample adequacy is assessed through
    a. KMO index
    b. Bartlett's test of sphericity
    c. Correlation matrix
    d. All of the above

13. Which one of these is **NOT** a form of orthogonal rotation in exploratory factor analysis?
    a. Varimax
    b. Quatimax
    c. Equamax
    d. Direct Oblimin

14. Which one of these IRT models can accommodate polytomous data?
    a. Rasch model
    b. 2 PL model
    c. Graded Response Model (GRM)
    d. None of the above

15. In IRT, ability is denoted by
    a. θ
    b. λ
    c. Σ
    d. ϕ

16. In the context of IRT, discrimination parameter above 0.5 but below 1.0 indicate
    a. Low discriminability
    b. Moderate discriminability
    c. High discriminability

    d. Tough to say

17. What is **TRUE** about constrained GRM
    a. Multiple researchers are required to run code for GRM
    b. GRM cannot accommodate polytomous data
    c. Discrimination parameter is kept constant in constrained GRM
    d. There is nothing called constrained GRM

18. log.Lik, AIC, and BIC are all methods of understanding
    a. Model summary
    b. Factorability
    c. Dimensionality
    d. Multicollinearity

19. Which of these R packages could help you run CFA?
    a. Lavaan
    b. ggplot
    c. dplyr
    d. None of the above

20. Which one of these parameters are **NOT** used in the interpretation of CFA output?
    a. TLI
    b. CFI
    c. SRMR
    d. S/N ratio

Answer key:

1. A
2. B
3. B
4. C
5. B
6. A
7. A
8. A
9. D
10. D
11. D
12. A
13. D
14. C
15. A
16. B
17. C
18. A
19. A
20. D

This page is intentionally left blank

Conflict of Interest Statement
I declare no conflict of interest. Cover designs are developed through CANVA. Other images included in the script are developed through DALL-E of Open-AI.

Editing and proofreading
DHC Publishers

About the author

Naren D. Selvaratnam, PhD is an academic with interests primarily in *research methodology* and *psychometrics*. He has authored 6+ peer-reviewed papers in international journals and 1 book chapter. He completed his bachelor's degree in psychology from Winona State University (WSU), MN, USA in 2013. Upon completing his bachelor's degree, he completed his master's degree in applied psychology from Coventry University, UK in 2016. Soon after, he started his career as a young academic at the Colombo Institute of Research and Psychology as a trainee lecturer. In 2019, he earned a post-master degree in K-12 Educational Leadership and Administration from Winona State University, MN, USA. He also earned a Doctor of Philosophy degree in Educational Leadership from Management and Science University (MSU), Shah Alam, Malaysia. The focus of the PhD was leadership self-efficacy of minority students in state campuses. Currently, Naren instructs at Sri Lanka Institute of Information Technology (SLIIT) in the capacity of a senior lecturer. He has supervised more than 100 BSc students and 30 MSc students during their dissertations. In addition to being a lecturer, Naren also spends substantial time learning religion, spirituality, and philosophy. He is an avid enthusiast of learning languages and currently speaks Sinhala, Tamil, English, and also a little bit of Chinese. He also has developed 10+ hypnotherapy tracks.

doi: 10.61495/ejcr.0002.24

ISBN: XXXXXXXX