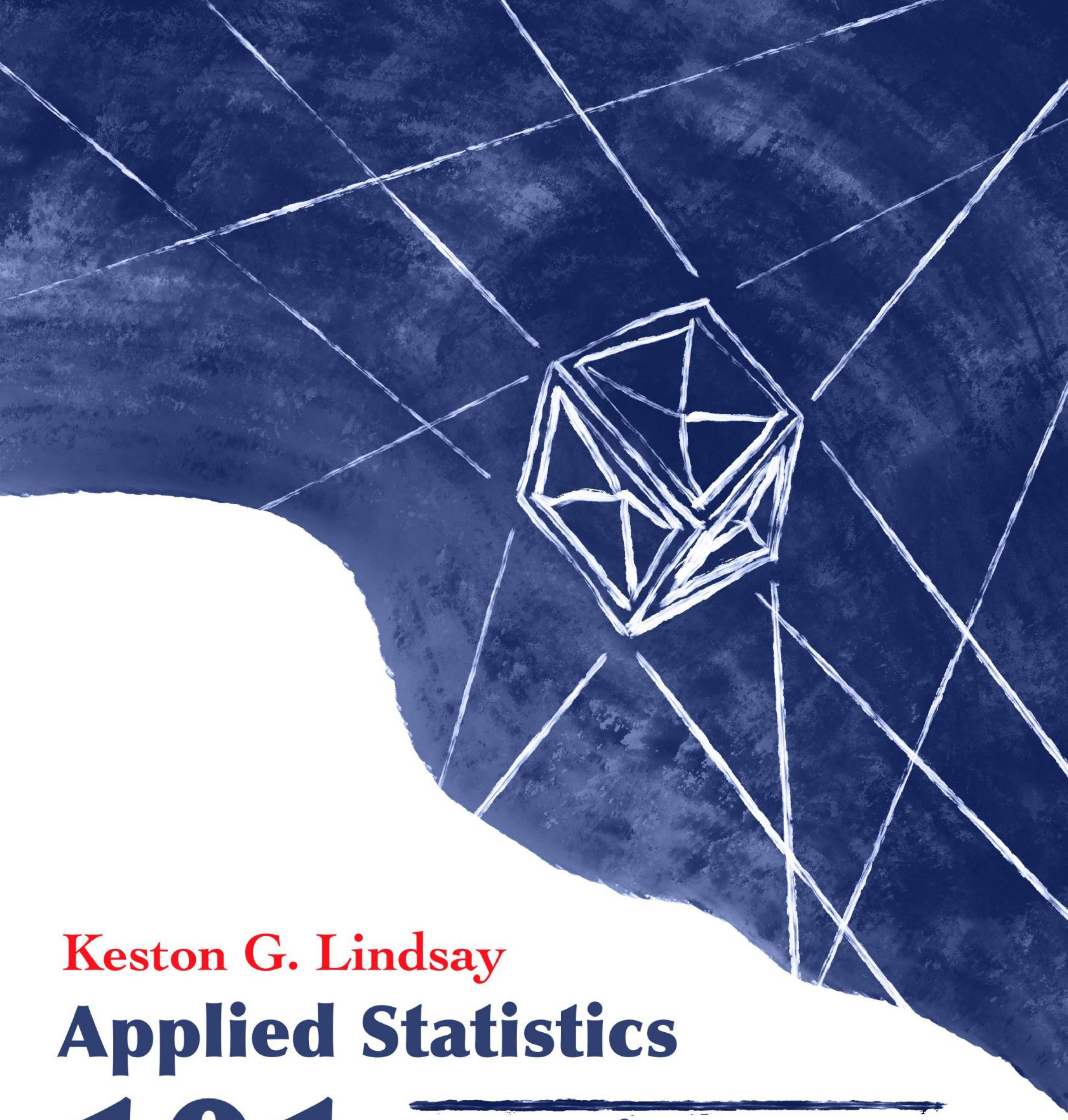
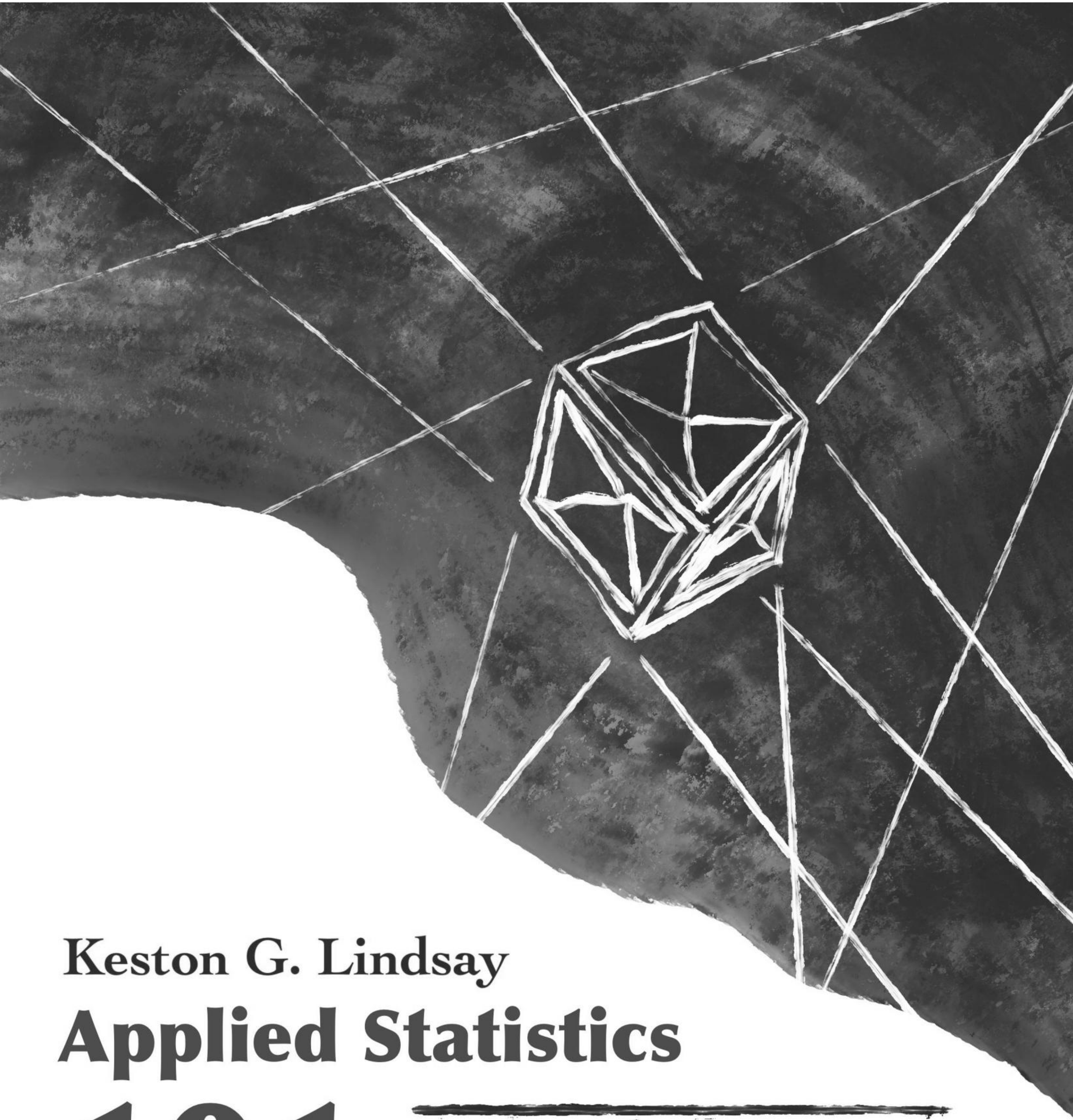


Keston G. Lindsay

Applied Statistics 101 using R:

A primer for an Introductory Class in Statistical Programming

iSTES



Keston G. Lindsay

Applied Statistics 101 using R:

A primer for an Introductory Class in Statistical Programming

iSTES

Applied Statistics 101 using R: A Primer for an Introductory Class in Statistical Programming

Author
Keston G. Lindsay, University of Colorado-Colorado Springs, USA

Cover Designed by
Mevlut Unal, Necmettin Erbakan University, Türkiye

ISBN: 978-1-952092-93-0

Date of Publication
April 2026

Publisher
ISTES
Monument, CO, USA

Contact
International Society for Technology, Education and Science (ISTES)
www.istes.org
istesoffice@gmail.com

Citation

Lindsay, K. G. (2026). *Applied Statistics 101 using R: A Primer for an Introductory Class in Statistical Programming*. ISTES.

Table of Contents

Preface

The purpose of this book is to serve as a guide for students concurrently learning statistics and R. Students new to both programming and statistics may experience increased cognitive load as they have to learn both concepts for the first time. This book is meant to reduce that barrier. That said, students who are familiar with statistical content and not coding, or computer science students learning statistics may also benefit. The chapters focus on pragmatic workflows in R, applied statistical reasoning, as well as interpretation of results. As R is free and open source, proprietary tools are not required. This also makes the book suitable for institutions with limited resources, and for anyone wishing to practice outside of formal coursework.

The methods introduced include descriptive statistics, t-tests, z-tests, chi-square tests, correlation, regression, and one-way analyses of variance. Assumption checks, non-parametric alternatives, and effect sizes are also included, though these are not strictly required for an introductory course. However, they may help to develop habits that transfer to additional coursework, research, hands-on data analysis, and interpretation. Where possible, several packages are discussed for each task in order to demonstrate R's flexibility as an ecosystem. Users of this manual may use these examples as starting points, and adapt other packages as their confidence grows and as they see fit. The datasets featured in this manual may be obtained using this link: https://osf.io/mrw5v/overview.

Although the order of the chapters is intentional, the book is intended to be modular. Readers are not obligated to read through the book sequentially. Chapters are written to stand alone, and the book may be read in any order. All examples use synthetic instructional datasets.

Above all, this book is meant to be practical and easy to use. Learning R and statistics should be challenging in productive ways, but not discouraging. I hope that this book helps students to build both competence and confidence and helps instructors to build a replicable bridge between statistical concepts and computing.

Dr. Keston G. Lindsay
https://orcid.org/0000-0002-2039-7477
University of Colorado, Colorado Springs
Helen and Arthur Johnson Beth-El College of Nursing and Health Sciences
Department of Human Physiology & Nutrition
4925 N Nevada Avenue
Colorado Springs, CO 80918, USA
Contact e-mail: *klindsay@uccs.edu*

Citation

Lindsay, K. G. (2026). Preface. In *Applied Statistics 101 using R: A primer for an introductory class in statistical programming* (pp. 1-2). ISTES.

Chapter 1 - Descriptive Statistics and Visualization

Chapter Highlights

- Learn to compute measures of central tendency and dispersion using R and add-on packages (e.g., *DescTools, psych, skimr*).
- Explore and interpret methods of data visualization such as frequency tables, histograms, and boxplots.
- Practice creation and management of objects in order to facilitate efficient and reproducible workflows
- Learn to check data assumptions and check for outliers
- Compare outputs from multiple R packages for descriptive statistics and visualization.

Introduction

Descriptive statistics are used to summarize data. They describe measures of central tendency (mean, median, mode), measures of dispersion (standard deviation, interquartile range, etc.), z-scores, and methods of visualizing these parameters. Classical methods of visualization include frequency tables, boxplots, and histograms. These parameters, charts and graphs are used to check assumptions for parametric analyses. They are used to ensure distributions are approximately Gaussian (bell-shaped and symmetrical), and to screen for outliers. After completing this tutorial, readers should be able to:

- Compute measures of central tendency using base *R* and add-on packages.
- Create and interpret basic visualization such as boxplots, frequency tables, and histograms.
- Compare descriptive statistics outputs from multiple *R* packages.
- Manage and reference *R* objects to organize analyses.
- Build confidence in applying *R* to data analysis.

Definitions

Virtually all descriptive statistics tables should include the sample size in each data column (henceforth referred to as *n*), or a similar term such as “count”, or “number of rows”. Some explicitly mention “total” and “missing” values as well. The mean is defined as the average of all valid scores in a data column. The median is the middle value of the scores in ascending or descending order. The mode is the most common score and is unique in that several modes may exist within a data column. The range is a measure of dispersion that represents the difference between the maximum and minimum values. Variance is calculated by squaring deviation scores (i.e., difference between the mean and each score), summing them, and dividing by degrees of freedom (i.e., n - 1). Simply, variance indicates the spread of data. If scores are consistent, variance is small. If scores are very different, variance is large. Standard deviation (SD) is the square root of variance and describes how much each score differs from the mean on average. Like deviation scores, each score has its own unique z-score, which may be computed by dividing the deviation score by the standard deviation. The coefficient of variation is the standard deviation divided by the mean, and the standard error of the mean is calculated by dividing the standard deviation by the square root of the sample size.

Other measures of dispersion include the interquartile range (IQR), which is the difference between the 25th and 75th percentiles within a data column. The median absolute deviation (MAD) is computed by finding deviation scores from the median, then finding the median of those scores, and should be used for skewed data (Rindskopf & Shiyko, 2010). Skewness describes the lack of symmetry in a Gaussian distribution (i.e., is the bell-shaped curve symmetrical?), while kurtosis describes how peaked or flat a distribution is.

Introduction to *R*

R (R Core Team 2025) is an open-source statistical programming language and environment. It is supported by a comprehensive worldwide community of scholars from several disciplines. Members of this community contribute add-on packages for virtually any task in statistics, data science, and related areas. While other commercial statistical packages exist, they may be costly; one advantage of *R* is that it is free. However, as *R* involves coding, it may deter potential users due to coding anxiety (Forrester et al., 2022). Users of *R* may choose to use an integrated development environment (IDE) for *R,* such as *RStudio* (Posit, 2025). Although *RStudio* is not a “point-and-click” graphical user interface, it provides benefits such as the import and management of data and packages, and visualization.

Objects

Management of objects is also a benefit of *RStudio*. An object may be used to easily abbreviate, manage and store datasets, analyses, functions, or even methods of visualization. Depending on the nature of objects, they may be analyzed and modified to create new objects. For example, a subset of an object assigned as a complete dataset may be assigned as another object. The object may be created/abbreviated by creating the abbreviation followed by “<-“ or “=”.

For example, the object below:

```
Points<-c(12, 15, 376)
```

contains three scores. By typing the following command:

```
ave<-mean(points)
```

We may create an object for the mean, or average of the scores. Other examples of creating objects will follow throughout this manuscript. Creating and tracking objects are strongly recommended as methods to simplify and document analyses.

Base R and Other Packages

The base package performs basic statistical descriptive statistics and visualization. Additionally, packages may be added to perform specific tasks. There are several comprehensive packages for descriptive statistics and/or visualization. These include the *psych* package, *pastecs* (Grosjean & Ibanez, 2024), *summarytools* (Comtois, 2025), *skimr* (Waring et al., 2025), and *DescTools* (Signorell, 2025). One objective of this paper is to comparatively review some of these add-on packages. It may be used as a tutorial for descriptive statistics for a basic applied statistics class, or a statistical programming course, using *R*. It is also a free supplement for students who wish to familiarize themselves with the *R* environment, to practice coding, and practice basic descriptive statistics using *R*. Users are encouraged to replicate the syntaxes, and to practice creating and naming objects as shown in the tables below.

Premise and Dataset

The dataset contains variables related to the achievement of 50 fictitious students in an introductory statistics course. It may be used for practice and teaching, as the user sees fit. A statistics professor/researcher wants to see whether anxiety, statistical self-efficacy, time spent studying, math background score, and class attendance affect final grade in a statistics class. Their abbreviations are listed and described in Table 1. The data featured in this chapter may be obtained using this link: https://osf.io/mrw5v/overview.

Table 1. Variable Key

Variable	Meaning
fg	Final course grade (overall achievement in course)
hs	Number of hours per week spent studying (self-reported)
mbs	Mathematics background score (composite score from math placement test)

Variable	Meaning
sse	Statistical self-efficacy score (validated psychometric scale)
car	Class attendance rate (%)
anx	Test anxiety score (validated psychometric scale)

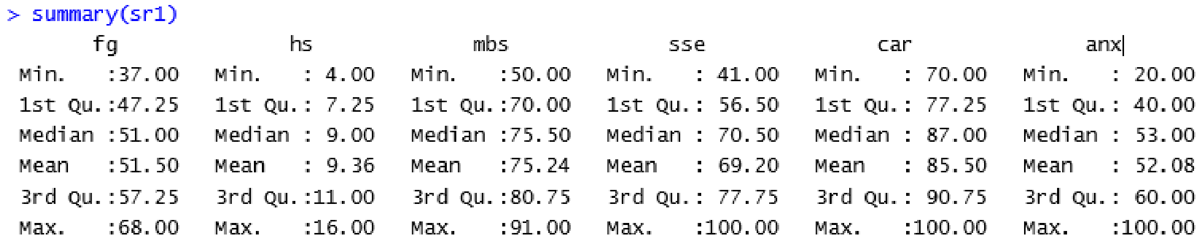
Method

Descriptive Statistics using Base R

Base *R* can provide an abbreviated list of descriptive statistics. The *summary* command provides an abbreviated descriptives table for all variables. The table may also be obtained for one variable at a time. While the output does not include measures of dispersion, these may be obtained via the *sd*, *var*, and *IQR* syntaxes, for standard deviation, variance, and the interquartile range, respectively. Range may be obtained by subtracting minimum from maximum values as shown in row 5 of Table 2. Finally, z-scores may be obtained for a single variable or an entire data frame, if those variables are continuous. A summary of these syntaxes may be found on Table 2. Figure 1 shows example output for R's summary command.

Table 2. Base *R* Syntax for Descriptive Statistics

Syntax	Purpose
summary(sr1)	Basic statistics; mean, median, minimum, maximum, first and third quartiles for the entire dataset.
summary(sr1$fg)	Same as above, but for the final grade (fg) variable only.
mean(sr1$fg, na.rm = FALSE)*, median(), min(), max(), IQR(), sd(), var()	Obtain the mean, median, minimum values, maximum values, interquartile range, standard deviation, and variance respectively, for a single variable. "na.rm = FALSE" argument essentially ignores missing values, which are designated as "NA" in *R*.
sort(table(sr1$fg), decreasing = TRUE)	Provides an abbreviated frequency table for a single variable in descending order. The first value will be the mode.
max(sr1$fg)-min(sr1$fg)	Obtains the range of a single variable.
scale(sr1$fg)	z-scores (single variable)
sr1szcores<-scale(sr1)	z-scores (entire dataset); "sr1zscores<-" command creates an object.



```
> summary(sr1)
       fg              hs              mbs              sse              car               anx
 Min.   :37.00   Min.   : 4.00   Min.   :50.00   Min.   : 41.00   Min.   : 70.00   Min.   : 20.00
 1st Qu.:47.25   1st Qu.: 7.25   1st Qu.:70.00   1st Qu.: 56.50   1st Qu.: 77.25   1st Qu.: 40.00
 Median :51.00   Median : 9.00   Median :75.50   Median : 70.50   Median : 87.00   Median : 53.00
 Mean   :51.50   Mean   : 9.36   Mean   :75.24   Mean   : 69.20   Mean   : 85.50   Mean   : 52.08
 3rd Qu.:57.25   3rd Qu.:11.00   3rd Qu.:80.75   3rd Qu.: 77.75   3rd Qu.: 90.75   3rd Qu.: 60.00
 Max.   :68.00   Max.   :16.00   Max.   :91.00   Max.   :100.00   Max.   :100.00   Max.   :100.00
```

Figure 1. Base *R* Descriptive Statistics Table using *Summary* Command

Descriptive Statistics using Add-on Packages

Other packages offer descriptive statistics tables of varying quality. The *psych* package offers *n*, mean, SD, median, trimmed mean, MAD, minimum, maximum, range, skewness, kurtosis, and standard error of the mean. These are offered by the *pastecs* package, in addition to variance, tests of normality, and the coefficient of variation. Of the packages discussed in this paper, *pastecs* offers the most detailed descriptive statistics table, and the *DescTools* output quality appears to be most similar to *psych.*

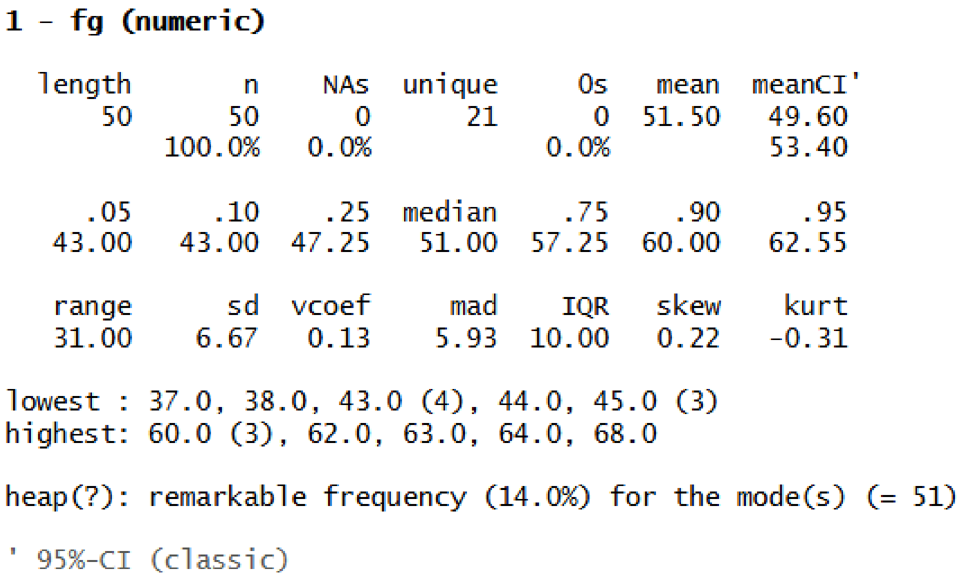
The *skimr* package also offers *n*, missing, a completion rate, mean, minimum, first quartile, median, 3rd quartile, and maximum values (designated as p0, p25, p50, p75, and p100 respectively). When analyzing several variables, it offers the number of columns as well. It is like the *summarytools* package in this regard, as well as offering histogram variations alongside descriptives output for all variables. The syntaxes for each package are listed in table 3.

Table 3. Descriptive Statistics Syntaxes for Other Packages

Package	Syntax	Output
	install.packages(psych) install.packages(pastecs) install.packages(summarytools) install.packages(DescTools)	Installs the respective packages.
	library(psych) library(pastecs) library(summarytools) library(DescTools)	Loads the respective packages.

Package	Syntax	Output
psych	describe(sr1)	Includes n*, mean, standard deviation, median, mode, trimmed means, median absolute deviation, minimum, maximum range, skewness, kurtosis, standard error. Overall, good descriptive statistics table.
pastecs	stat.desc(sr1)	Includes n*, missing, NA, minimum, maximum, range, sum, median, mean, standard error of the mean, confidence intervals for the mean, variance, standard deviation, coefficient of variation. Most comprehensive descriptive statistics table in this list.
	stat.desc(sr1, norm = TRUE)	Adds skewness, kurtosis, and Shapiro-Wilks test
summarytools	dfsummary(sr1)	Includes mean, standard deviation, median, minimum and maximum values, interquartile range, coefficient of variation, valid n*, abbreviated frequencies and visualization via the descriptives table. Good for quick checks.
skimr	skim(sr1)	Includes mean, missing, standard deviation, quintiles (minimum, median, maximum, 1st and 3rd quartile, and histograms. Good for quick checks.
DescTools	Desc(sr1)	Includes n*, n/a, missing, number of unique values, mean, mode, confidence interval for the mean, quartiles, and others. Good tandem option for descriptive statistics and visualization.

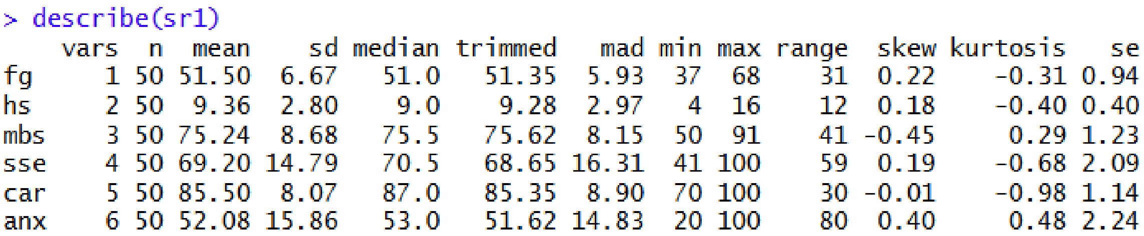
n*= sample size



```
1 - fg (numeric)

  length       n     NAs  unique      0s   mean  meanCI'
      50      50       0      21       0  51.50   49.60
          100.0%    0.0%            0.0%          53.40

     .05     .10     .25  median     .75    .90     .95
   43.00   43.00   47.25   51.00   57.25  60.00   62.55

   range      sd   vcoef     mad     IQR   skew    kurt
   31.00    6.67    0.13    5.93   10.00   0.22   -0.31

lowest : 37.0, 38.0, 43.0 (4), 44.0, 45.0 (3)
highest: 60.0 (3), 62.0, 63.0, 64.0, 68.0

heap(?): remarkable frequency (14.0%) for the mode(s) (= 51)

' 95%-CI (classic)
```

Figure 2. *desctools* Descriptives Output

.



```
> describe(sr1)
    vars  n  mean    sd median trimmed   mad min max range  skew kurtosis   se
fg     1 50 51.50  6.67   51.0   51.35  5.93  37  68    31  0.22    -0.31 0.94
hs     2 50  9.36  2.80    9.0    9.28  2.97   4  16    12  0.18    -0.40 0.40
mbs    3 50 75.24  8.68   75.5   75.62  8.15  50  91    41 -0.45     0.29 1.23
sse    4 50 69.20 14.79   70.5   68.65 16.31  41 100    59  0.19    -0.68 2.09
car    5 50 85.50  8.07   87.0   85.35  8.90  70 100    30 -0.01    -0.98 1.14
anx    6 50 52.08 15.86   53.0   51.62 14.83  20 100    80  0.40     0.48 2.24
```

Figure 3. *psych* Descriptives Output

Visualization

For this list, the *summarytools* package, by far, offers the most comprehensive frequency table, using a short command. The base package and *DescTools* also offer abbreviated frequency tables.

The base *R* package also offers customizable histograms and boxplots, using relatively easy syntaxes. As aforementioned, *summarytools* and *skimr* also offer histogram variants with their descriptive statistics tables. The base package also offers boxplots for single variables. With one command, the *DescTools* package will produce descriptive statistics, boxplots, as well as histograms with frequency and ogive curves. The *ggplot2* (Wickham, 2016) and *ggpubr* (Kassambara, 2025) packages also produce customized histograms and boxplots for single variables.

Table 4. Options for Frequency Tables

Package	Syntax	Notes
R (base)	table(sr1$fg) modefreq<-sort(table(sr1$fg), decreasing = TRUE)	Better for one variable at a time. Use this command to easily find mode(s).
summarytools	freq(sr1)	Comprehensive frequency tables, with counts, and valid/total percentages. Best in this list.
	freq(sr1$fg)	One variable at a time.
DescTools	Desc(sr1)	Abbreviated frequency table using this command.
	Freq(sr1$fg)	Grouped frequency distribution for individual variables.

Table 5. Options for Histograms

Package	Syntax	Notes
R (base)	hist(sr1$fg, main = “Final Grade Histogram”, xlab = “Final Grade”)	“main” and “xlab” syntaxes label histogram and x-axis respectively (see Figure 4).
DescTools	Desc(sr1)	Includes frequency curve and ogive curve.
ggplot2	gradehist<-ggplot(sr1, aes(x = fg)) + geom_histogram(binwidth = 1) + labs(title = "Final Grade Distribution", x = "Grades", y = "Frequency")	Longer code, but customizable plots; one variable at a time. “gradehist<-“ assigns it as an object.
	ggplot(sr1, aes(x=fg))+ geom_histogram(binwidth =	Same as above, with customized color scheme. Good option for

Package	Syntax	Notes
	1, fill = “aquamarine”, color = “black”)+ labs(title = “Final Grade Distribution”, x = “Grades”, y = “Frequency”)	customization when users become comfortable with coding.
ggpubr	ggpub_hist<- gghistogram(data = sr1, x = "fg", bins = 30, fill = "steelblue", color = "black", add = "mean", add_density = TRUE, rug = TRUE)+ labs(title = "Exam Score Distribution", x = "Exam Score", y = "Frequency")	Customizable plot. Note the syntaxes for number of bins, coloring, mean, density curve, labeling. Good option for customization.
	ggpub_hist	Displays histogram (see Figure 5).

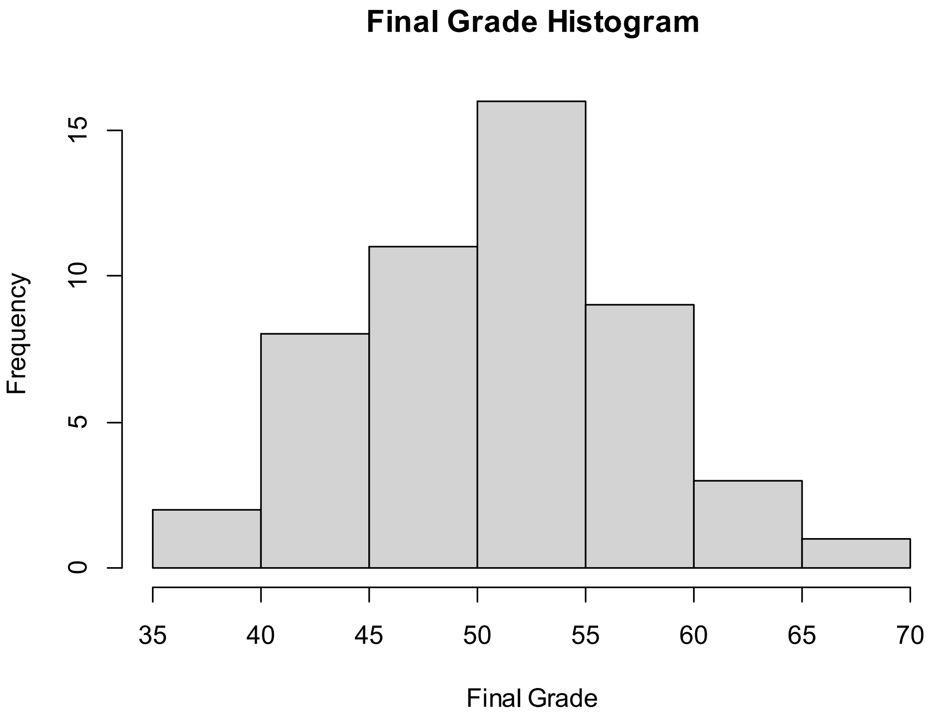


Figure 4. Base *R* Histogram

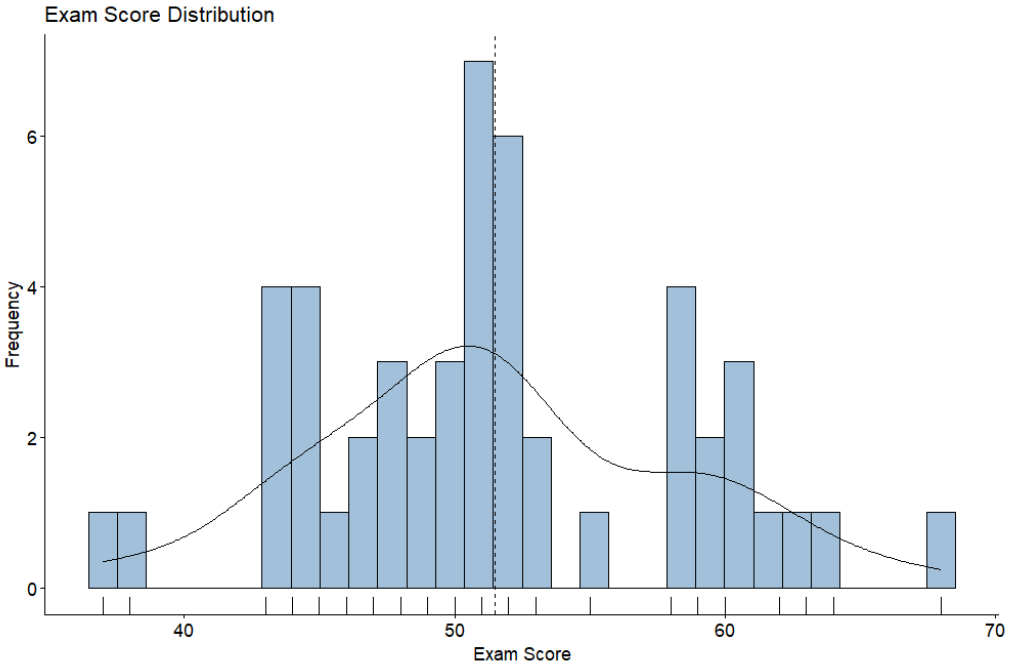


Figure 5. *ggpubr* Histogram

Table 6. Options for Boxplots

Package	Syntax	Notes
R (base)	boxplot(sr1$fg)	“main” and “xlab” syntaxes label histogram and x-axis respectively.
ggplot2	gradeboxplot<-ggplot(sr1, aes(y = fg, x = 1)) + geom_boxplot(fill = "maroon", color = "black") + labs(title = "Final Grade Boxplot", y = "Final Grade")	Longer code, but customizable plots.
ggpubr	ggboxplot(sr1, y = "fg", add = "jitter") + labs(title = "Final Grade Boxplot", x = "Sample", y = "Final Grade")	Longer code, but customizable plots.
DescTools	Desc(sr1)	

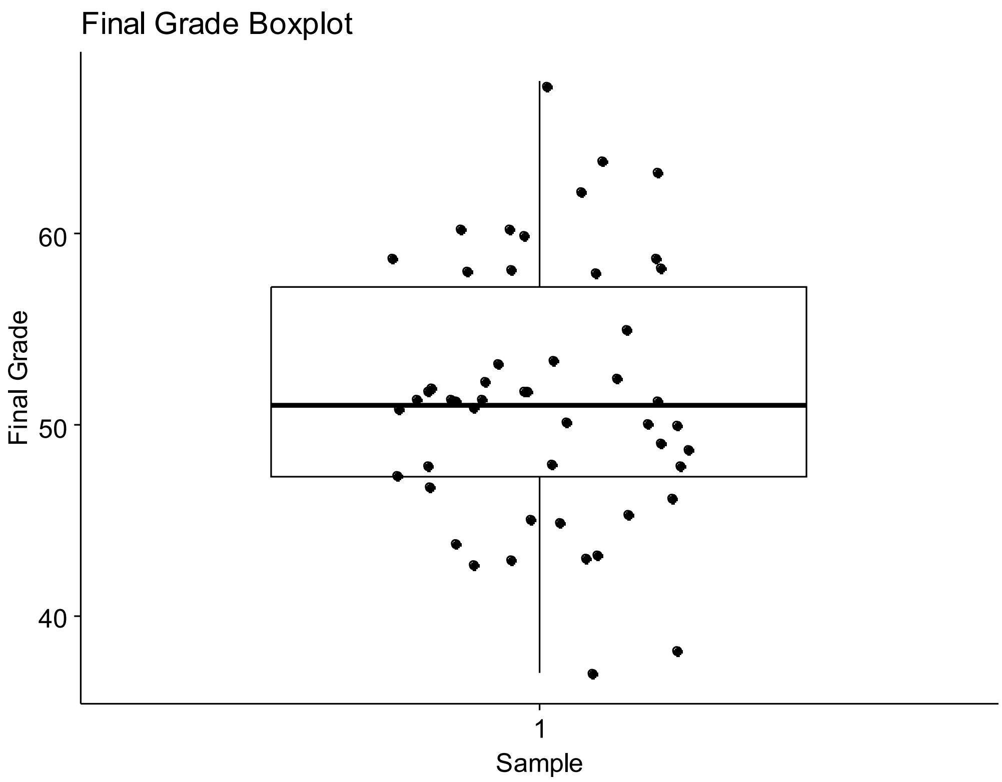


Figure 6. *ggpubr* Boxplot

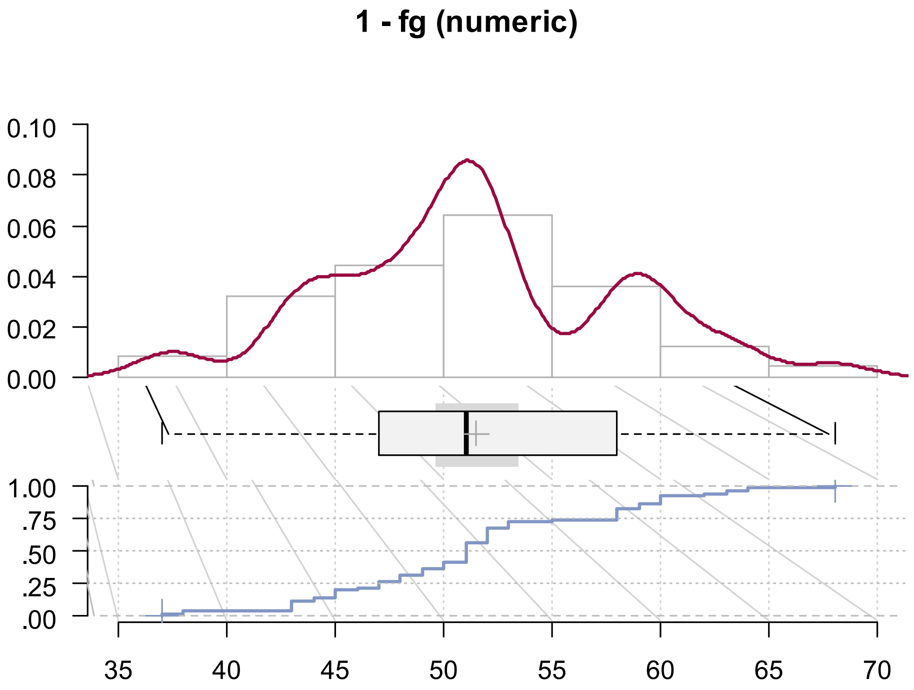


Figure 7. *DescTools* Visualization

Discussion and Conclusion

The purpose of this paper was to provide an introduction, review, and free tool for learning descriptive statistics and visualization in *R*. Examples of students who would benefit from this paper include statistics students new to coding, or computer science students learning statistics. It serves as a free resource to practice coding, and to reduce the effects of statistics and/or coding anxiety. The descriptive statistics and visualization outputs for the base *R* package, *DescTools*, *psych*, *skimr*, and *summarytools* packages were reviewed and compared. The *ggplot2* and *ggpubr* packages were also introduced as options for customized visualization. *R* offers multiple approaches to descriptive statistics, ranging from the base package's brief summaries to extremely detailed tables found in the *pastecs* package. The *DescTools* package offers the best combination of descriptive statistics and visualization, with short syntax. Packages such as *ggplot2*, and *ggpubr* allow visual customization, making *R* a flexible choice for teaching descriptive statistics.

While this tutorial is meant to be used as a teaching tool, it does not contain a comprehensive list of packages. Moreover, not all capabilities of the discussed packages have been explored in this paper. For example, similar papers on visualization may be written for single packages like *ggpubr*, *ggplot2*, or *DescTools*. This was intentional, as to not overwhelm new statistics and *R* students. The *skimr* and *summarytools* packages can be used to quickly check data. For descriptive statistics *pastecs* is best, while *DescTools* excels in overall descriptive statistics and visualization. As students gain confidence, they may replicate, and gradually customize, their own plots using *ggplot2* and *ggpubr.*

References

Comtois, D. (2025). *summarytools: Tools to Quickly and Neatly Summarize Data* (R package version 1.1.4) [Computer software]. Comprehensive R Archive Network (CRAN). https://CRAN.R-project.org/package=summarytools

Forrester, C., Schwikert, S., Foster, J., & Corwin, L. (2022). Undergraduate R programming anxiety in ecology: persistent gender gaps and coping strategies. *CBE—Life Sciences Education*, *21*(2). https://doi.org/10.1187/cbe.21-05-0133

Grosjean, P., Ibanez, F. (2024). *pastecs: Package for Analysis of Space-Time Ecological Series* (R package version 1.4.2) [Computer software]. Comprehensive R Archive

Network (CRAN). https://CRAN.R-project.org/package=pastecs

Kassambara, A. (2025). *ggpubr: 'ggplot2' Based Publication Ready Plots* (R package version 0.6.1) [Computer software]. https://doi.org/10.32614/CRAN.package.ggpubr

Posit (2025). *RStudio Desktop*. Posit. https://posit.co/download/rstudio-desktop/

R Core Team (2025). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Rindskopf, D., & Shiyko, M. (2010). Measures of Dispersion, Skewness and Kurtosis. *International Encyclopedia of Education (3rd ed.)*, 267–273. https://doi.org/10.1016/b978-0-08-044894-7.01344-0

Signorell, A. (2025). *DescTools: Tools for Descriptive Statistics* (R package version 0.99.60) [Computer software]. https://doi.org/10.32614/CRAN.package.DescTools

Waring, E., Quinn, M., McNamara, A., Arino de la Rubia, E., Zhu, H., & Ellis, S. (2025). *skimr: Compact and Flexible Summaries of Data* (R package version 2.2.1.9000) [Computer software]. rOpenSci. https://docs.ropensci.org/skimr/

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Citation

Lindsay, K. G. (2026). Descriptive statistics and visualization. In *Applied Statistics 101 using R: A primer for an introductory class in statistical programming* (pp. 3-16). ISTES.

Chapter 2 - Introduction to the Chi-Square Test

Chapter Highlights

- Use R to compute the chi-square goodness-of-fit and test of independence.
- Learn assumption checks (expected cell counts, independence, etc.).
- Includes syntax for handling assumption violations and alternative methods (e.g., Fisher’s test).
- Compute effect sizes (Cohen’s W, Cramer’s V) and interpret benchmarks.
- Use and compare packages such as base *R*, *gmodels*, and *vcd*.
- Use methods of visualization such as association plots and mosaic plots.
- Learn APA-style reporting of results.

Introduction

The chi-square test is a statistical method used to determine relationships between categorical variables. Created by Karl Pearson (Nihan, 2020), the chi-square test may be applied in several ways. The chi-square goodness of fit (GOF) test is used to see if observed proportions differ from expected proportions for a single variable. The chi-square test of independence (TOI) is used to find bivariate associations between categories. The data must be categorical (nominal/ordinal) for both variables and collected at random. It may be used for continuous outcomes that are collapsed into categories due to failure to meet the assumptions of parametric tests. Compared groups must be independent. Expected cell counts should be 5 or more, and no expected cell should be less than 1. Cohen's W and Cramer's V are the effect size measures for GOF and TOI respectively (Cohen, 1988). Their benchmarks for small, medium, and large effects are .1, .3, and .5 respectively. Note that these recommendations are general and may differ depending on context. This tutorial is designed for students in an introductory applied statistics or statistics programming course. By practicing the contents found here, readers will be able to:

- Compute and interpret the chi-square goodness of fit using *R*.
- Compute and interpret the chi-square test of independence using *R* and add-on packages.
- Visualize categorical associations using association and mosaic plots.
- Interpret Cohen's W and Cramer's V effect sizes.
- Compare outputs and effect size for tests of independence using base *R* and other packages.

Introduction to *R*

R (R Core Team, 2025) is a statistical language that is free and open source. It is supported by a community of scientists and programmers who contribute add-on packages via the Comprehensive *R* Archive Network (CRAN). Users may find add-on packages for virtually any purpose related to data analysis. Additionally, several packages do similar tasks, resulting in greater flexibility to customize various types of analyses. This gives *R* a distinct advantage over proprietary programs with licenses that require regular license renewals. The major disadvantage is that the user must learn coding, which can invoke coding anxiety (Forrester et al., 2022). This may compound statistical anxiety, especially in introductory courses where students learn *R* along with statistical concepts. *R* may be used with an integrated development environment (IDE) such as *RStudio* (Posit, 2025), to help with management of datasets and add-on packages. Note that although syntax is provided for the installation and loading of packages, *RStudio* has options to help with these. This paper is meant to serve as a tutorial for using the chi-square tests in *R*. The base package will be used in tandem with *dplyr* (for modifying datasets; Wickham et al., 2019). Other packages such as *DescTools* (Signorell, 2025), and *vcd* (Meyer et al., 2002) will be used.

Objects and Functions

An object is an umbrella term that can refer to most things in *R*. For example, in this paper, objects are created from datasets to manipulate, summarize, or analyze them, or to create subsets of data. Objects are created by assigning a manageable name or abbreviation to a dataset by using "<-" or "=". Objects may also be assigned to calculations for further use. For example, the syntax "statstable1<- table(ind$Attitude, ind$Math_Background)" creates a contingency table called "statstable". Another object "chi1<-chisq.test(statstable1)" is then created to store and document the analysis. A function is defined as some sort of operation or process. Later in this paper, a function is used to create a Cramer's V matrix. Note that an object is also assigned to this function. An IDE such as *RStudio* can also be used to manage objects and functions. The user is strongly encouraged to create and manage objects as they see fit, and other examples follow throughout this manuscript.

Datasets

The artificially generated data sets used herein are found as supplementary material accompanying this article. All datasets are artificial and contain no real student data. For GOF, a researcher wishes to know if the observed distribution of preferred study methods matches an expected uniform distribution. For TOI, a professor wishes to know if students' attitude toward statistics is associated with their genders, majors, learning formats, and mathematics backgrounds. They are simple and ideal for practicing chi-square tests without overwhelming complexity. They may be used for practice or teaching, ad libitum. They may be obtained using this link: https://osf.io/mrw5v/overview.

Method

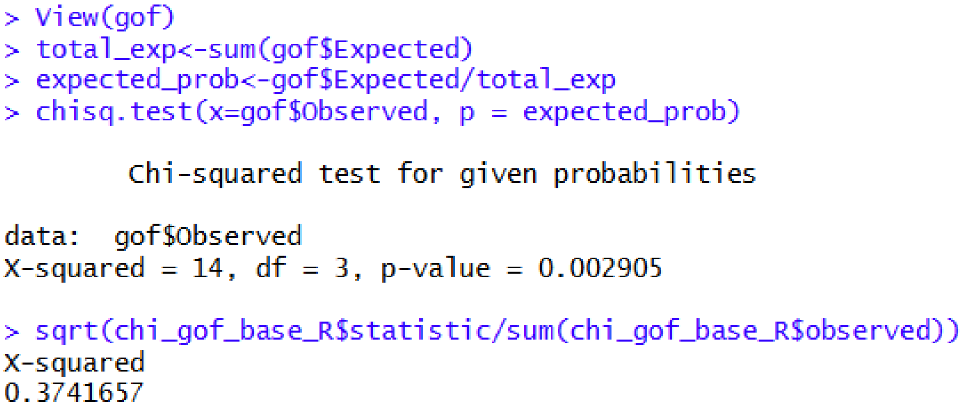
The following table shows the workflow for calculating GOF in *R*. Note the creation of objects for each step of the analysis. Once the data are loaded, the following syntaxes may be used to compute chi-square tests.

Table 1. Chi-square Goodness of fit Using Base *R*

Package	Syntax	Purpose
Base *R*	counts_base_R<-aggregate(Attitude ~ Gender + Major + Math_Background + Format, data = ind1, FUN = length)	Frequencies aggregated by attitude.
	> total_exp<-sum(gof$Expected) > expected_prob<-gof$Expected/total_exp > chisq.test(x=gof$Observed, p = expected_prob)	Chi-square goodness of fit. Note that there are three lines of syntax.
	chi_gof<-chisq.test(x=gof$Observed, p = expected_prob)	Creates object for the acquisition of effect size. Note that Base *R* is the only option in the list for goodness-of-fit.
	w<-sqrt(chi_gof$statistic/sum(chi_gof$observed))	Cohen's W effect size.

The chi-square goodness of fit analysis is shown below. Note the rejection of the null hypothesis. Currently, no package calculates an effect size for chi-square goodness of fit. However, it may be calculated using the syntax shown below, and in Table 1. It should be reported as follows: $\chi^2(3, n = 100) = 14$, $p < .01$, W = .37. The size of the effect is moderate. The null hypothesis is rejected ($p < .05$), indicating that observed proportions differ from expected proportions.

Similarly, the following chi-square test of independence explores the association between students' attitude toward statistics and their mathematics background. It should be reported as follows: $\chi^2(4, n = 300) = 39.7$, $p < .001$, V = .26. The null hypothesis is rejected ($p < .05$), therefore showing a significant association. Cramer's V is obtained by using *DescTools* and shows a weak bordering on moderate relationship according to Cohen's (1988) benchmarks.

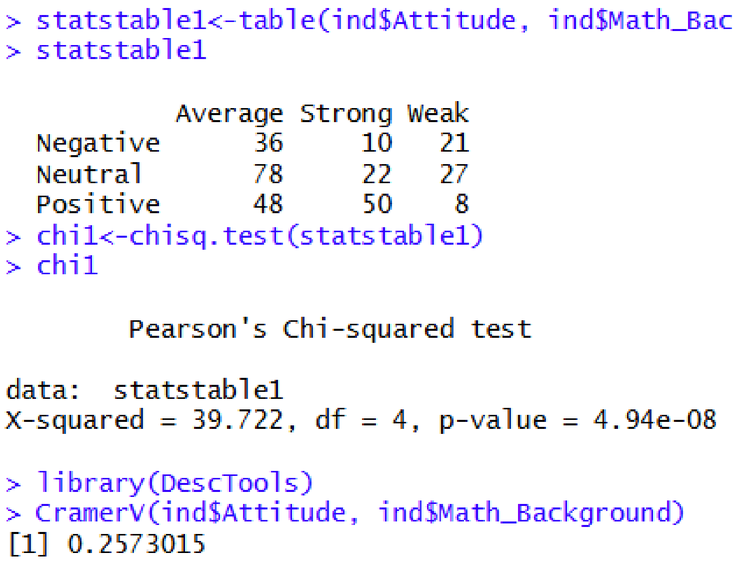


```
> View(gof)
> total_exp<-sum(gof$Expected)
> expected_prob<-gof$Expected/total_exp
> chisq.test(x=gof$Observed, p = expected_prob)

        Chi-squared test for given probabilities

data:  gof$Observed
X-squared = 14, df = 3, p-value = 0.002905

> sqrt(chi_gof_base_R$statistic/sum(chi_gof_base_R$observed))
X-squared
0.3741657
```

Figure 1. Output for Chi-square Goodness of Fit using Base *R*

Table 2. Chi-square Test of independence using Base *R* and *DescTools*

Package	Syntax	Purpose
	install.packages("DescTools") install.packages("dplyr") library(DescTools) library(dplyr)	Installs and opens the respective packages.
Base *R*	statstable1<-table(ind$Attitude, ind$Math_Background)	Contingency table for chi-square test of independence.
	chi1<-chisq.test(statstable1)	Running chi-square test on the object created above.
dplyr	ind1<-ind%>%select(-ID)	Duplicates original data with removed ID column.
DescTools	Desc(ind1) CramerV(ind$Attitude, ind$Math_Background)	Frequency tables for all categories. Cramer's V effect size



```
> statstable1<-table(ind$Attitude, ind$Math_Background)
> statstable1

          Average Strong Weak
  Negative     36     10   21
  Neutral      78     22   27
  Positive     48     50    8
> chi1<-chisq.test(statstable1)
> chi1

        Pearson's Chi-squared test

data:  statstable1
X-squared = 39.722, df = 4, p-value = 4.94e-08

> library(DescTools)
> CramerV(ind$Attitude, ind$Math_Background)
[1] 0.2573015
```

Figure 2: Two-way Table, χ^2 Test, and Effect Size for Attitude and Mathematics Background in Base *R*

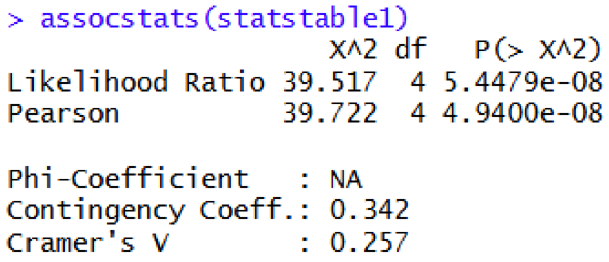
Table 3. Other Options for Chi-square Test of Independence

Package	Syntax	Purpose
	install.packages(“gmodels”) install.packages(“vcd”) library(gmodels) library(vcd)	Installs and opens the respective packages.
Base *R*	statstable2<-table(ind$Attitude, ind$Gender)	
	chi2<-chisq.test(statstable2, simulate.p.value = TRUE, B = 10000)	Option when cells have counts < 5.
	fisher2<-fisher.test(statstable2)	Fisher’s test; same as above.
vcd	chi1_vcd<-assocstats(statstable1)	Good option for obtaining statistics and effect sizes. Overall best pick in this list.
	assoc(statstable1, shade = TRUE, main = "Statistics Attitude vs Math Background")	Association Plot. Good option for visualizing contribution of each cell

		to χ^2 outcome.
	mosaic(statstable1, shade = TRUE, legend = TRUE, main = "Mosaic Plot, Statistics Attitude vs. Math Background")	Mosaic plot
gmodels	chi1_gmodels<-CrossTable(ind$Attitude, ind$Math_Background, chisq = TRUE)	Two-way table and chi-square test. Good option for quantifying contribution of each cell to χ^2 outcome.
DescTools	cv_matrix<-sapply(ind1, function(x) sapply(ind1, function(y) CramerV(table(x, y))))	Creates a "Cramer's V matrix". Optional task for experienced users, or those who wish to practice coding.
	round(cv_matrix, 2)	Rounds coefficients.

The *vcd* Package

The *vcd* package offers both the test statistics and effect size, as shown below.



```
> assocstats(statstable1)
                    X^2 df   P(> X^2)
Likelihood Ratio 39.517  4 5.4479e-08
Pearson          39.722  4 4.9400e-08

Phi-Coefficient   : NA 
Contingency Coeff.: 0.342 
Cramer's V        : 0.257 
```

Figure 3. χ^2 Test and Effect Size using the *vcd* Package

The *vcd* package is also useful for visualizing the results of the test, using an association plot (Zeileis et al., 2007). This is obtained using the following syntax: "assoc(statstable1, shade = TRUE, main = "Statistics Attitude vs Math Background")".

The direction of the bar indicates the direction in which the deviation occurred (i.e., more or less than expected). Grey bars show where deviations by chance. Blue or red bars indicate statistically significant deviations from observed vs. expected. The blue bars show cells where observed counts are more than expected counts. This applies to those with a negative attitude and a weak mathematics background, as well those with a positive mathematics attitude and a strong mathematics background. Hence there are associations between those with negative attitudes and a weak mathematics background, as well as associations between a positive attitude and a strong mathematics background.

The red bars show cells where observed counts are less than expected counts. There is a lower than expected count in those who have a neutral attitude and a strong mathematics background, as well as those with a positive mathematics attitude and a weak math background. Overall, those with a weak math background have negative attitudes toward statistics, while those with a strong mathematics background have positive attitudes toward statistics.

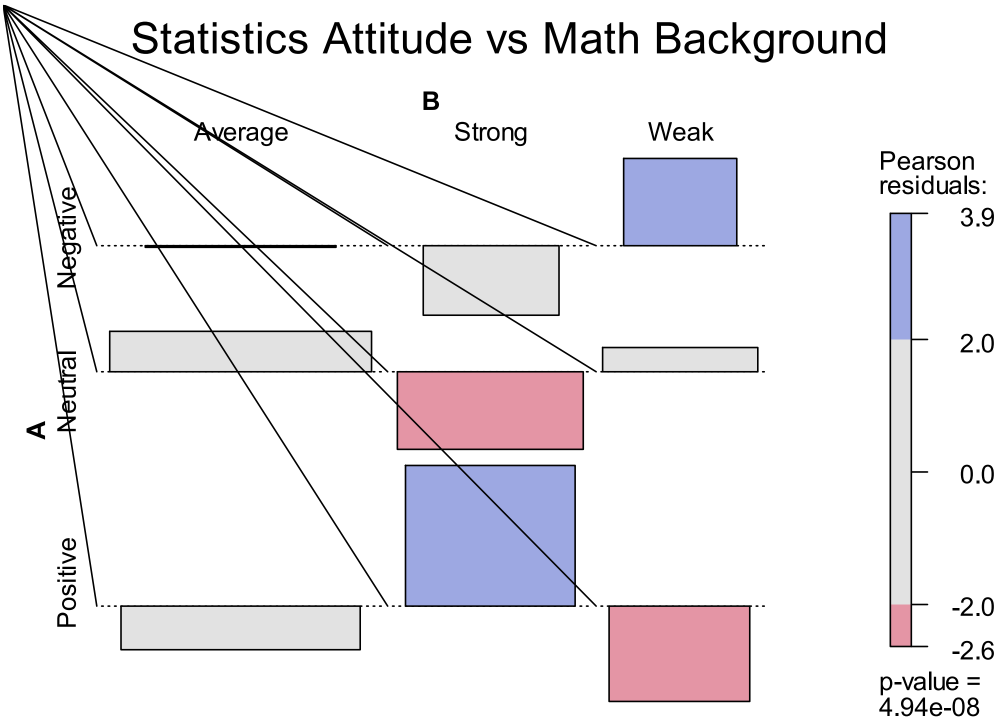


Figure 4. Association Plot Attitude and Mathematics Background using *vcd*

A mosaic plot (Zeileis et al., 2007) may also be obtained by using the following syntax: “mosaic(statstable1, shade = TRUE, legend = TRUE, main = "Mosaic Plot, Statistics Attitude vs. Math Background")”.

The color interpretation is similar to the association plot, but the size of the boxes represent the proportion of the total count.

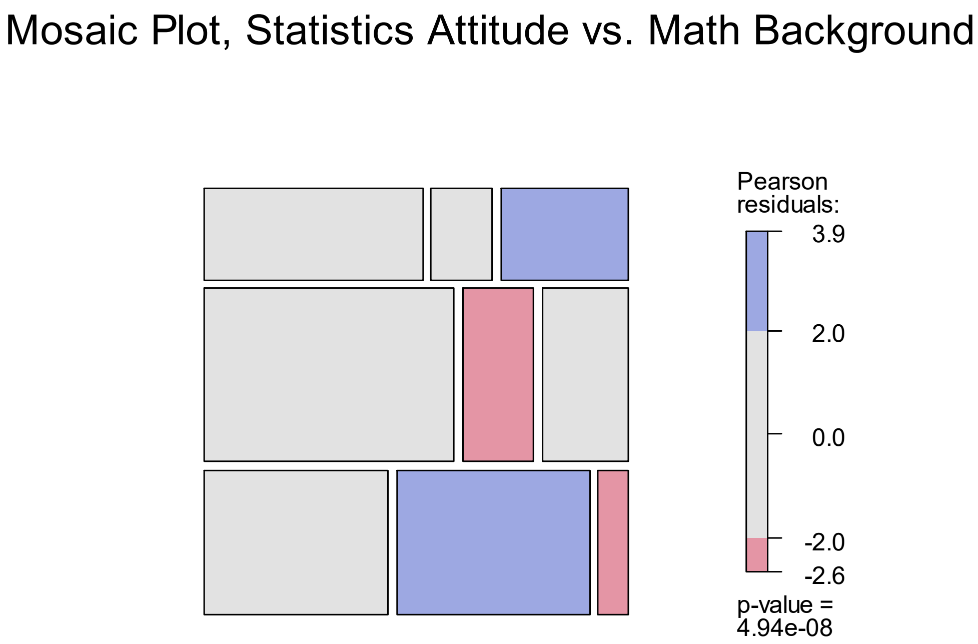
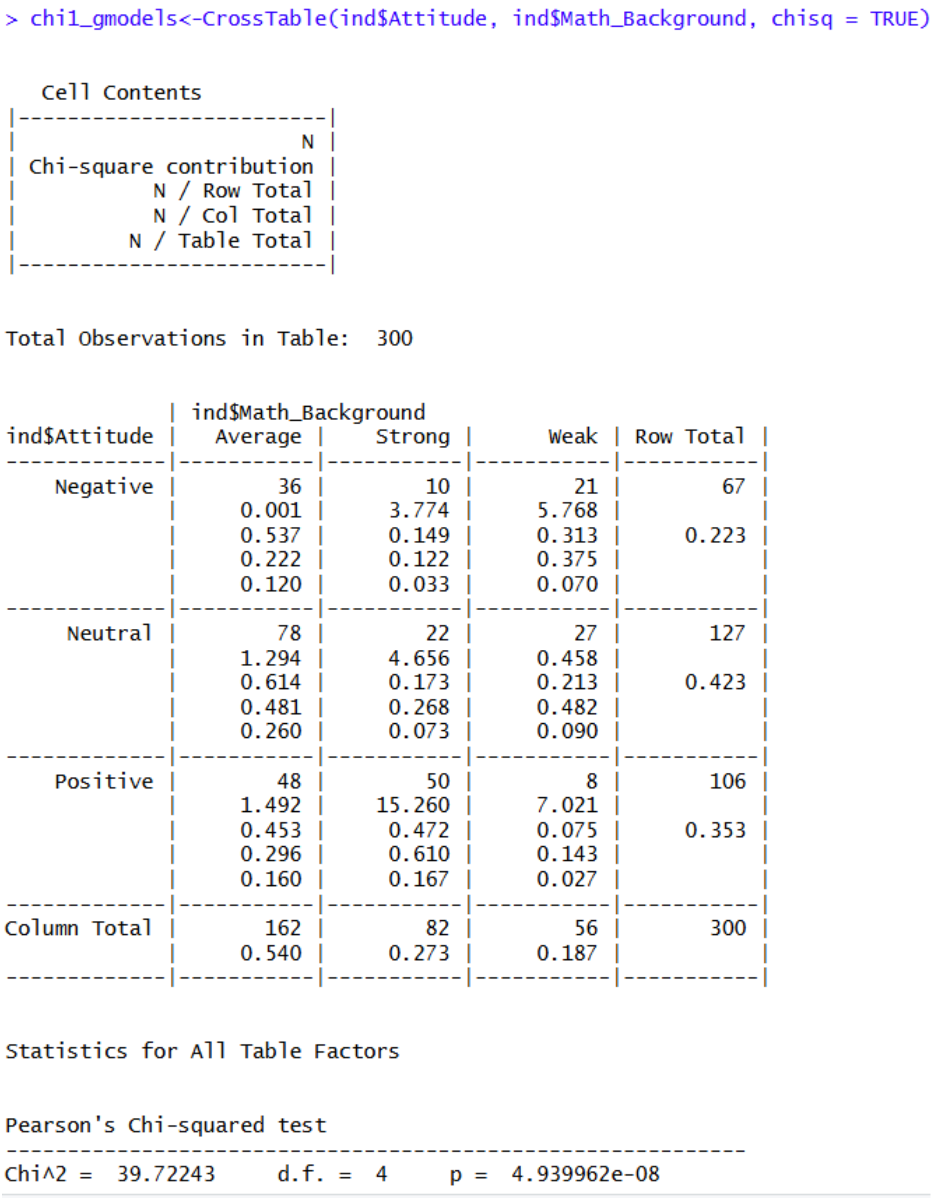


Figure 5. Mosaic Plot Attitude and Mathematics Background using *vcd*

These methods of visualization may be applied to all possible associations.

The *gmodels* Package

The *gmodels* package (Warnes at al., 2024) offers a very detailed 2 x 2 table, along with the chi-square test using one line of syntax. In addition to the raw counts for each cell, it quantifies contribution of the cell to the chi-square statistic. Note that the cell with positive attitudes and strong mathematics background makes the strongest contribution. The cells also offer row proportions, column proportions, and proportions for the entire sample. The cell with positive attitudes and strong mathematics background makes up 47.2 % of the positive attitude row, 61 % of the strong background column, and 16.7 % of the entire sample. The major disadvantage of this package, is that it does not offer Cramer’s V. However, it may be easily obtained using *DescTools*.



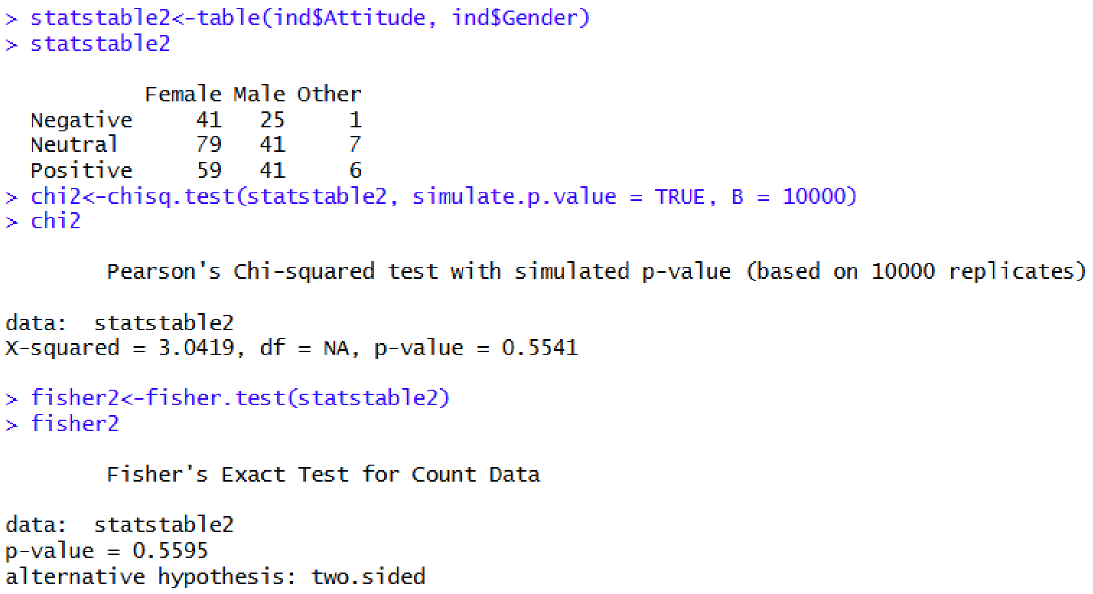
```
> chi1_gmodels<-CrossTable(ind$Attitude, ind$Math_Background, chisq = TRUE)


   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  300


             | ind$Math_Background
ind$Attitude |   Average |    Strong |      Weak | Row Total |
-------------|-----------|-----------|-----------|-----------|
    Negative |        36 |        10 |        21 |        67 |
             |     0.001 |     3.774 |     5.768 |           |
             |     0.537 |     0.149 |     0.313 |     0.223 |
             |     0.222 |     0.122 |     0.375 |           |
             |     0.120 |     0.033 |     0.070 |           |
-------------|-----------|-----------|-----------|-----------|
     Neutral |        78 |        22 |        27 |       127 |
             |     1.294 |     4.656 |     0.458 |           |
             |     0.614 |     0.173 |     0.213 |     0.423 |
             |     0.481 |     0.268 |     0.482 |           |
             |     0.260 |     0.073 |     0.090 |           |
-------------|-----------|-----------|-----------|-----------|
    Positive |        48 |        50 |         8 |       106 |
             |     1.492 |    15.260 |     7.021 |           |
             |     0.453 |     0.472 |     0.075 |     0.353 |
             |     0.296 |     0.610 |     0.143 |           |
             |     0.160 |     0.167 |     0.027 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |       162 |        82 |        56 |       300 |
             |     0.540 |     0.273 |     0.187 |           |
-------------|-----------|-----------|-----------|-----------|


Statistics for All Table Factors


Pearson's Chi-squared test
------------------------------------------------------------
Chi^2 =  39.72243     d.f. =  4     p =  4.939962e-08
```

Figure 6. Contingency Table from *gmodels* Package

Options for Violated Assumptions

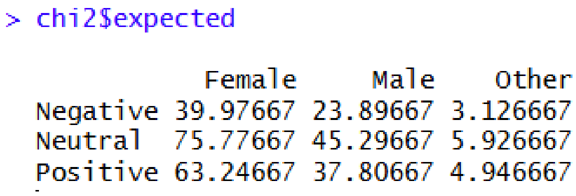
The following chi-square test of independence demonstrates an example of the violation of the assumption of expected cell counts ≥ 5. The syntax may be amended as shown below, or Fisher's exact test may be used. Report as follows: $\chi^2(4, n = 300) = 3.0$, $p > .05$, V = .07.

In Figure 9, the syntax is optional to obtain a "Cramer's V matrix" function in the style of a Pearson's correlation matrix. This enables simultaneous interpretation of the magnitude of associations between all variables. As a reminder, Cohen's benchmarks for Cramer's V are identical to those of Pearson's correlation and are .1, .3, and .5 for weak, moderate, and strong associations respectively. For example, the strongest association is between attitude and mathematics background and is a weak to moderate association based upon the aforementioned benchmarks. This may be helpful for advanced follow-up analyses that go beyond the scope of this paper, such as logistic regression or correspondence analysis.



```
> statstable2<-table(ind$Attitude, ind$Gender)
> statstable2

           Female Male Other
  Negative     41   25     1
  Neutral      79   41     7
  Positive     59   41     6
> chi2<-chisq.test(statstable2, simulate.p.value = TRUE, B = 10000)
> chi2

        Pearson's Chi-squared test with simulated p-value (based on 10000 replicates)

data:  statstable2
X-squared = 3.0419, df = NA, p-value = 0.5541

> fisher2<-fisher.test(statstable2)
> fisher2

        Fisher's Exact Test for Count Data

data:  statstable2
p-value = 0.5595
alternative hypothesis: two.sided
```

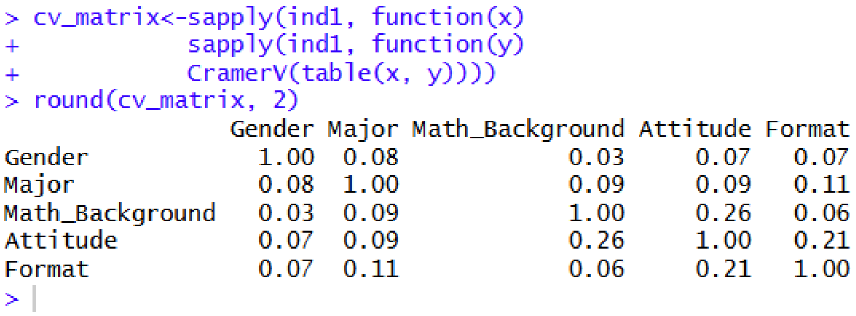
Figure 7. Two-way Table, χ^2 Test, and for Attitude and Gender using Base *R*



```
> chi2$expected

            Female     Male    Other
Negative 39.97667 23.89667 3.126667
Neutral  75.77667 45.29667 5.926667
Positive 63.24667 37.80667 4.946667
```

Figure 8. Expected Cell Counts using Base *R*

Note the syntax to acquire expected cell counts in Figure 7 above. The “Other” column shows two expected rows < 5. Check expected cell counts before interpreting results.



```
> cv_matrix<-sapply(ind1, function(x)
+           sapply(ind1, function(y)
+           CramerV(table(x, y))))
> round(cv_matrix, 2)
                Gender Major Math_Background Attitude Format
Gender            1.00  0.08            0.03     0.07   0.07
Major             0.08  1.00            0.09     0.09   0.11
Math_Background   0.03  0.09            1.00     0.26   0.06
Attitude          0.07  0.09            0.26     1.00   0.21
Format            0.07  0.11            0.06     0.21   1.00
> |
```

Figure 9. Cramer’s V Matrix

Discussion and Conclusion

The chi-square goodness of fit test and the chi-square test of independence may be obtained using R's base package and are accessible to beginners in coding. The *DescTools* package can be used to obtain Cramer's V. The *gmodels* package is a good option for contingency tables and may also be used with *DescTools*. The *vcd* package is a good option for obtaining chi-square statistics, Cramer's V, and visualization.

R does not currently contain a package that automatically calculates effect sizes for chi-square goodness of fit tests. However, it may be calculated using the syntax presented in this paper. Not all available packages or features were showcased or discussed. This was intentional, to not overwhelm new statistics and *R* students.

Chi-square tests and their effect sizes may easily be obtained using either the *vcd* package, or a combination of *DescTools* with the base package or *gmodels*. Students are encouraged to replicate examples and explore association or mosaic plots to build familiarity with analysis and visualization.

References

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Forrester, C., Schwikert, S., Foster, J., & Corwin, L. (2022). Undergraduate *R* programming anxiety in ecology: persistent gender gaps and coping strategies. *CBE—Life Sciences Education*, *21*(2). https://doi.org/10.1187/cbe.21-05-0133

Meyer, D., Achim Zeileis, Hornik, K., & Friendly, M. (2002). *vcd: Visualizing categorical data*. https://doi.org/10.32614/cran.package.vcd

Nihan, S. T. (2020). Karl Pearson's chi-square tests. *Educational Research and Reviews, 15*(9), 575–580. https://doi.org/10.5897/ERR2019.3817

Posit (2025). *RStudio Desktop*. Posit. https://posit.co/download/rstudio-desktop/

R Core Team (2025). *R: A language and environment for statistical computing. R* Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Signorell, A. (2025). *DescTools: Tools for descriptive statistics* (R package version 0.99.60) [Computer software]. https://doi.org/10.32614/CRAN.package.DescTools

Warnes G.R., Bolker B., Lumley T., Johnson R.C., Jain N., Schwartz M., Rogers, J. (2024). *gmodels: Various R Programming Tools for Model Fitting. R* package version 2.19.1, https://github.com/r-gregmisc/gmodels.

Wickham, H., François, R., Henry, L., & Müller, K. (2019). A grammar of data manipulation [R package *dplyr* version 0.8.3]. *R-Project.org*. https://cran.r-project.org/package=dplyr

Zeileis, A., Meyer, D., & Hornik, K. (2007). Residual-Based Shadings for Visualizing (Conditional) Independence. *Journal of Computational and Graphical Statistics*, *16*(3), 507–525. https://doi.org/10.1198/106186007x237856

Citation

Lindsay, K. G. (2026). Introduction to the chi-square test. In *Applied Statistics 101 using R: A primer for an introductory class in statistical programming* (pp. 17-29). ISTES.

Chapter 3 - Correlation and Regression

Chapter Highlights

- Learn to compute and interpret strength of relationships using Pearson's correlation coefficients using packages such as base R, psych, and others.
- Visualize relationships using scatterplots, regression lines and network graphs.
- Perform linear and multiple regression using base *R*, *easystats*, and *olsrr*.
- Check assumptions using diagnostic plots and *gvlma*.
- Learn non-parametric, bootstrapped, and robust options for assumption violations.
- Learn APA-style reporting of results.

Introduction

Pearson product moment correlation (PPMC) and ordinary least squares regression (OLSR) are widely used to assess associations between continuous variables. Correlation (specifically, PPMC) is used to measure the relationship between two continuous variables, while regression is used to predict one continuous outcome from one or more continuous predictors. The concept was first used by Sir Francis Galton in the late 1800's to study heredity in sweet peas, and mathematically refined by Karl Pearson (Stanton, 2001). Correlation may be used as a stand-alone analysis, a measure of effect size, or as a method to screen variables for more advanced statistical methods, such as regression, factor analysis, and multivariate analysis of variance. Correlation and regression methods are widely taught and used in research across many fields. Jacob Cohen's thresholds for r and R^2 are shown in the table below.

Table 1. Thresholds for r and R^2

Threshold	r	R^2
Small	.1	.01
Medium	.3	.09
Large	.5	.25

The objectives of this tutorial are to teach the reader how to:

- Use and interpret correlation and regression analyses in *R*.
- Visualize correlation and regression in *R*.
- Compare output from base *R* and other add-on packages.
- Identify and address violations of assumptions using robust and bootstrapped methods.
- Visualize correlation and regression analyses.

Null Hypotheses

The null hypothesis for PPMC states that there is no linear relationship between any pair of variables. The null hypothesis for OLSR is that none of the independent variables (hypothesized predictors) will predict the outcome variable. While connected, PPMC tends to be more exploratory, and OLSR is more theory driven.

Assumptions

The assumptions for PPMC include linearity (i.e., the scatterplot follows a linear trend) and homoscedasticity (constant variability across scatterplot). They also include independence of observations, where data of participants (i.e., rows of data) are not influenced by data of other participants. The assumptions of OLSR are similar: linearity, homoscedasticity, normality of error distribution around zero, and independence of errors. Many options now exist for assumption violations for both PPMC and OLSR. These include non-parametric, robust, and bootstrapped options, several of which will be discussed. The choice for an alternative method is based upon the context of the study and/or the nature of the data. In any event, if there are doubts about violation of assumptions, it may be prudent to report the PPMC, OLSR, and alternative analyses. Follow-up comparison and/or contrast of the methods may be necessary depending upon context.

Introduction to R

R (R Core Team, 2025) is a free and open-source statistical programming language and environment, supported by a worldwide community. Members often submit add-on packages for both classical and modern statistical methods. As aforementioned, its major advantage is that it is free. However, it involves coding, which may create anxiety for new users (Forrester et al., 2022). In the case of an introductory statistics course, there is potential for coding anxiety to compound statistics anxiety (Cruise, Cash & Bolton, 1985). An integrated development environment (IDE) such as *RStudio* (Posit, 2025), may be used to import and manage datasets, objects, and packages.

Objects

An object is a name, phrase or abbreviation assigned to a dataset, function, or some other entity within the *R* environment to work with it more easily. An object may be designated to a data set, and another one may be used for a subset of that dataset. Creation of an object is designated by the symbols “<-“ or “=”. *RStudio* may also be used to create and manage objects. The syntax “scores <- c(13, 17, 125)” creates an object called “scores”, that stores three scores. In this tutorial, the syntax “sr1<-sr%>% select(-ID)” creates an object from the original dataset that removes the ID variable. This is then used to get correlation matrices (e.g., “cor3<-corr.test(sr1)” using the *psych* package). Objects may also be assigned to methods of visualization such as scatterplots, and readers are encouraged to create and organize objects as they see fit. Other examples will follow throughout this tutorial.

Packages

The packages used for PPMC in this paper are the base *R* package, *DescTools* (Signorell, 2025), the *psych* package (Revelle, 2025), and *Hmisc* (Harrell, 2025). The base *R* package will also be used for OLSR, in addition to *easystats* (Lüdecke et al., 2022). The *olssr* (Hebbali, 2024), and *robustbase* (Maechler et al., 2025) packages will also be introduced. The *dplyr* package (Wickham et al., 2019) will be used for some data management, while *gvlma* (Pena & Slate, 2019) will be used to check assumptions for multiple regression. Finally, *DescTools*, *lattice* (Sarkar, 2008), *psych*, *GGally* (Schloerke et al., 2025), *PerformanceAnalytics* (Peterson & Carl, 2025), *sjPlot* (Lüdecke, 2025), *qgraph* (Epskamp et al., 2012) and *visreg* (Breheny & Burchett, 2017) will be used for visualization.

Data

The dataset used for this paper is an artificially generated dataset that carries the following premise. A statistics professor/researcher wants to see if and how number of hours a week spent studying, math background score, statistical self-efficacy, class attendance, and anxiety affect final grade in a statistics class. It is sufficiently simple for students practicing correlation and regression. All code examples are reproducible and may be adapted freely for the purpose of teaching and learning. The abbreviations used for each table are listed in the table below. The data may be obtained using this link: https://osf.io/mrw5v/overview.

Table 2. Variables

Variable	Meaning
fg	Final course grade (overall achievement in course)
hs	Number of hours per week spent studying (self-reported)
mbs	Mathematics background score (composite score from math placement

Variable	Meaning
	test)
sse	Statistical self-efficacy score (validated psychometric scale)
car	Class attendance rate (%)
anx	Test anxiety score (validated psychometric scale)

Method

Correlation

Users may begin by importing the data into *RStudio*. The syntaxes in Table 3 may be used to obtain a scatterplot using the base package.

Table 3. Data Management and Scatterplots in Base R

Package	Syntax	Purpose
	install.packages("dplyr") library(dplyr)	
dplyr	sr1<-sr%>% select(-ID)	Command to create a new object from the original data that removes the ID column.
Base *R*	r_baseplot<-plot(sr1$fg, sr1$hs, pch = 19, col = "steelblue", cex = 1.5, xlab = "Final Grade", ylab = "Weekly Hours Spent Studying", main = "Scatterplot of Grade vs. Hours Studying")	Step 1: creates base scatterplot. Note the syntax for the labels. The *pch* command is used to make solid circles, and *cex* refers to the size of the points. Use for two variables.
	linearmodel1<-lm(sr1$hs~sr1$fg)	Step 2: assigns an object to the model used to draw the line
	abline(linearmodel1, col = "red", lwd = 1.5, lty = 1)	Step 3: This draws the regression line and adds it to the model.
	pairs(~fg+hs+mbs+sse+car+anx, data = sr1, main="Variable Scatter Plot")	Creation of scatterplot matrix; relatively long syntax.

Figure 1 shows a scatterplot of final grade vs. hours studied, with a regression line added. The positive slope (i.e., increasing from left to right), suggests a positive relationship.

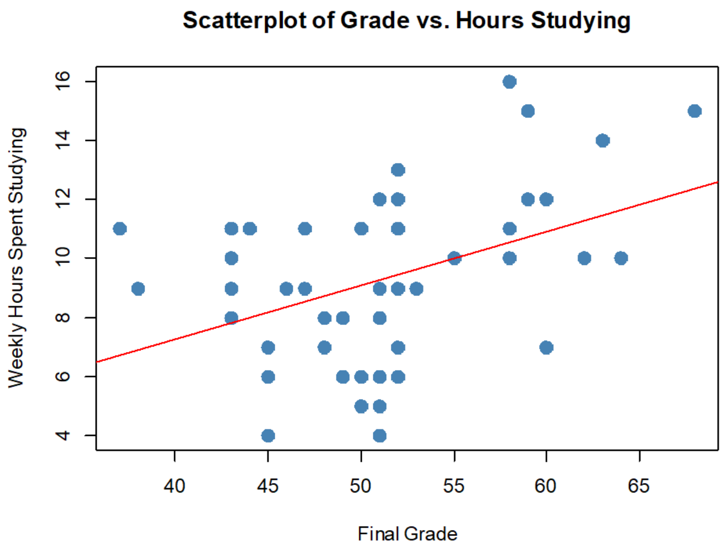


Figure 1. Scatterplot with Regression Line using *R*'s Base Package

The syntax used to create a scatterplot using base *R* is lengthy compared to other packages. Alternatively, other packages (*psych, GGally, DescTools, PerfromanceAnalytics*) make it relatively easy to obtain single or multiple scatterplots. Examples are shown in Table 4.

Table 4. Scatterplots and Visualization using Other Packages

Package	Syntax	Purpose
	install.packages("DescTools") install.packages("lattice") install.packages("psych") install.packages("ggally") install.packages("PerformanceAnalytics")	
	library(lattice) library(psych) library(ggally) library(performanceanalytics) library(qgraph)	Loads all packages used in this table.
lattice	lat_plot<-xyplot(fg ~ hs, data = sr1, type	Use for two

Package	Syntax	Purpose
	= c("p", "r"))	variables.
psych	pairs.panels(sr1)	Creates scatterplot matrix for all variables included.
GGally	ggpairs(sr1)	Same as above.
PerformanceAnalytics	chart.Correlation(sr1, histogram = TRUE, pch = 19)	Same as above. Convenient for quick comparisons and visualization.
qgraph	qgraph(cor1p, layout = "spring", minimum = 0.2)	Visualizes relationships as a network.
DescTools	DescToolscor<-Desc(fg~hs, data = sr1, main = "Grade vs. Hours Studying")	Use for two variables.
	DescToolsmatrix<-PlotPairs(sr1, main = "Scatterplot matrix of study variables")	Use for scatterplot matrix similar to *psych* and *GGally*.

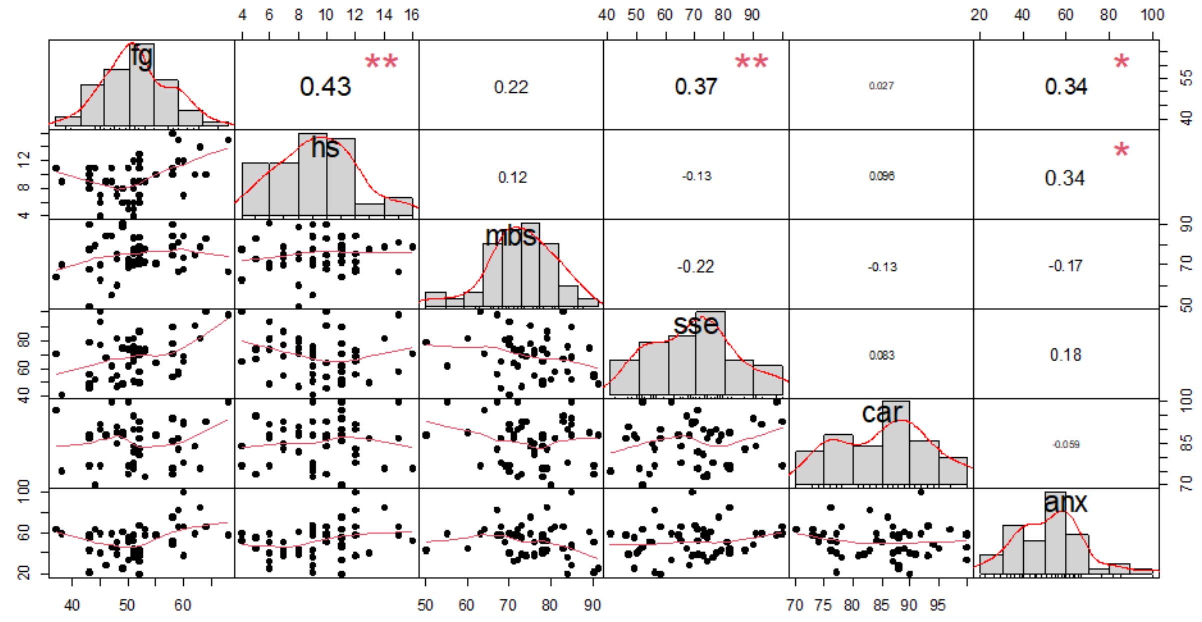


Figure 2. Scatterplot Matrix using the *GGally* Package

Correlation in the R Statistical Environment

The following table shows syntax for computing PPMC using the base package and several add-on packages. Base R does not provide p-values, but *Hmisc*, *psych*, and *DescTools* are options that do. Note the options for obtaining non-parametric (e.g., Spearman's) correlation.

Table 5. Correlation Syntax

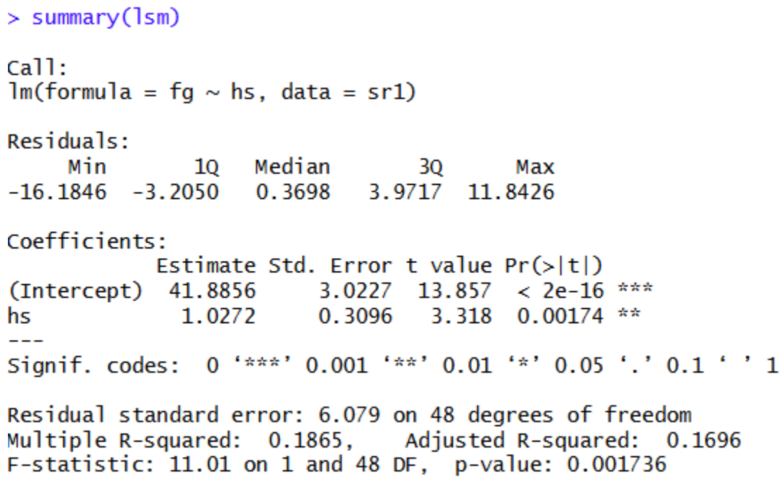
Package	Syntax	Purpose
Base *R*	cor(sr1, method = "pearson")	Parametric correlation; "cor(sr1)" yields same result as Pearson is default. Does not provide p-values.
	cor(sr1, method = "spearman")	Non-parametric correlation
Hmisc	cor2p<-rcorr(as.matrix(sr1))	Parametric
	cor2s<-rcorr(as.matrix(sr1), type = "spearman")	Non-parametric
psych	cor3<-corr.test(sr1) or cor3p<-corr.test(sr1, method = "pearson")	Assigns an object to parametric correlation in the *psych* package.
	cor3s<-corr.test(sr1, method = "spearman")	
	cor.plot(sr1)	Scatterplot matrix
DescTools	DescToolscor<-Desc(fg~hs, data = sr1, main = "Grade vs. Hours Studying")	Provides Pearson's, Spearman's, and Kendall's correlation coefficients.
	Cor(sr1, use = "pairwise.complete.obs")	Identical to Base *R* output

Regression

Linear Regression syntax using base R package

To run OLSR in *R,* the following lines of syntax may be used in the base package:
lsm<-lm(fg~hs, data = sr1)
summary(lsm)

The output is in Figure 3.



```
> summary(lsm)

Call:
lm(formula = fg ~ hs, data = sr1)

Residuals:
     Min       1Q   Median       3Q      Max 
-16.1846  -3.2050   0.3698   3.9717  11.8426 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  41.8856     3.0227  13.857  < 2e-16 ***
hs            1.0272     0.3096   3.318  0.00174 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.079 on 48 degrees of freedom
Multiple R-squared:  0.1865,	Adjusted R-squared:  0.1696 
F-statistic: 11.01 on 1 and 48 DF,  p-value: 0.001736
```

Figure 3. Simple Linear Regression using Base *R*

The null hypothesis of no effect of weekly hours spent studying on final grade has been rejected. It may be reported as follows: $F(1, 48) = 11.01$, $p < .01$, $R^2 = .19$, Adjusted $R^2 = .17$.

Multiple Regression

Multiple regression assesses the effect of multiple predictors on one outcome. The syntax for the base *R* package is listed in Table 6.

Table 6. Multiple Regression using Base *R* Package

Syntax	Output	Purpose
statsmod<-lm(fg ~ hs + mbs + sse + car + anx, data = sr1)	Provides an ANOVA report below the coefficients table.	Standard multiple regression analysis. This object is also used for other packages.
stepmodel_base_R<-step(statsmod, direction = "both")	Same as above; optional based upon correlation matrix.	Option for stepwise regression. Note removal of class attendance rate (car) variable.
plot(statsmod)	Q-Q plot	Checks normality.
	Residuals vs. fitted	Checks linearity.
	Homoscedasticity	Checks homoscedasticity.
	Residuals vs. leverage	Suggests influential cases.

Once an object has been created for the regression model, it may be visually evaluated using the *plot* command. Here is where most of the assumptions may be checked. The Q-Q plot tells if the assumption of normality is met, by showing whether residuals fall on the straight line. It appears that two participants (13 and 17) could result in a violation of this assumption, by deviating from the line.

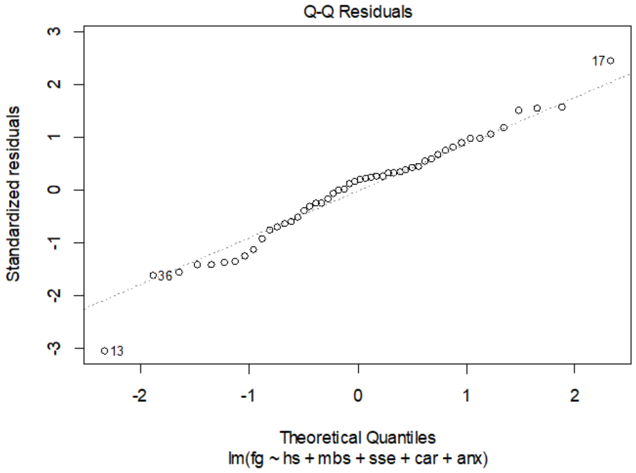


Figure 4. Base *R*'s Q-Q Plot

The residuals vs. fitted graph looks relatively straight, suggesting a linear relationship between the dependent variable and the predictors. Thus, the linearity assumption appears to be met.

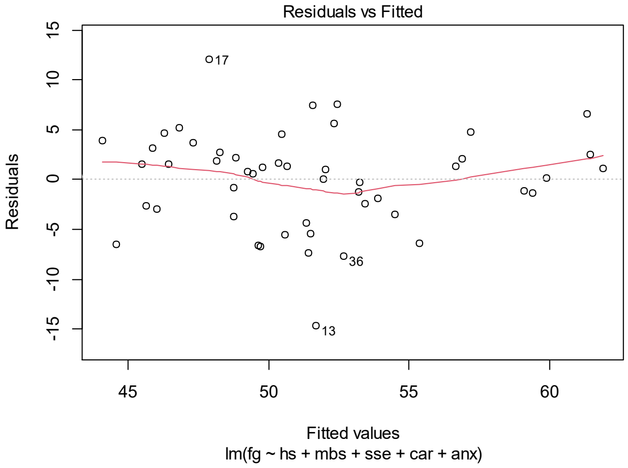


Figure 5. Base *R*'s Residuals vs. Fitted Graph

The scale-location graph addresses homoscedasticity and should show a haphazard band around a horizontal line. Thus, the assumption of homoscedasticity also appears to be met.

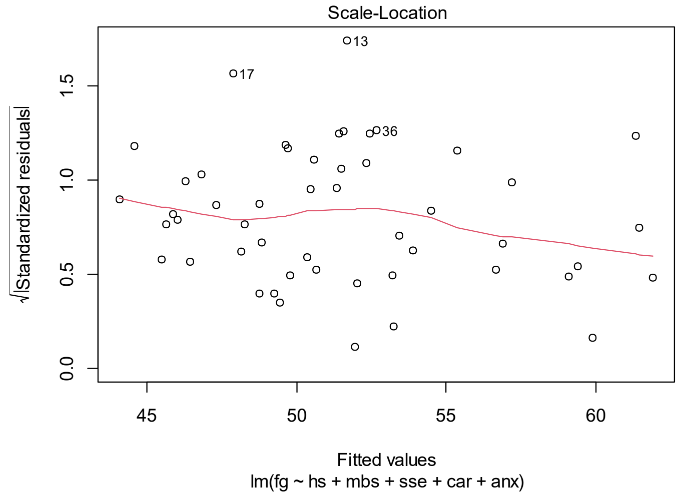


Figure 6. Base *R*'s Scale-location Graph

The important information about the residuals vs. leverage plot is that it shows potentially influential cases. This output shows that cases 7, 13, and 35 appear to be influential cases.

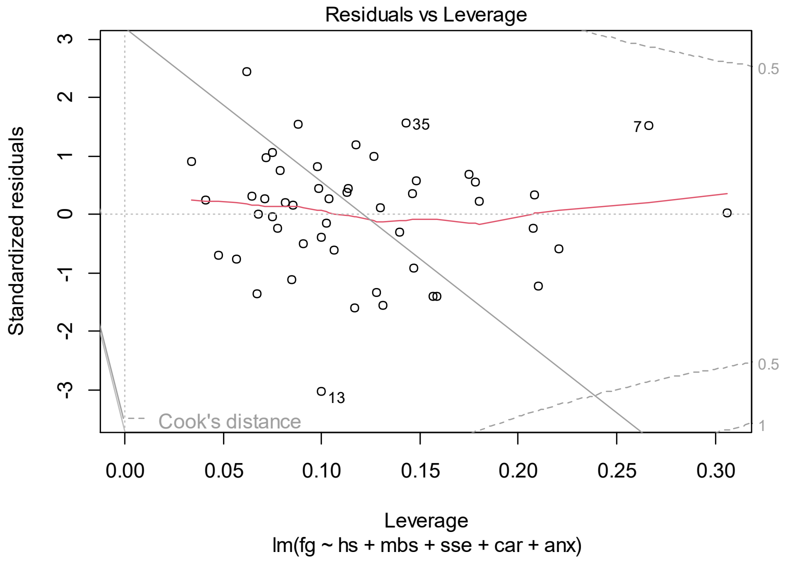
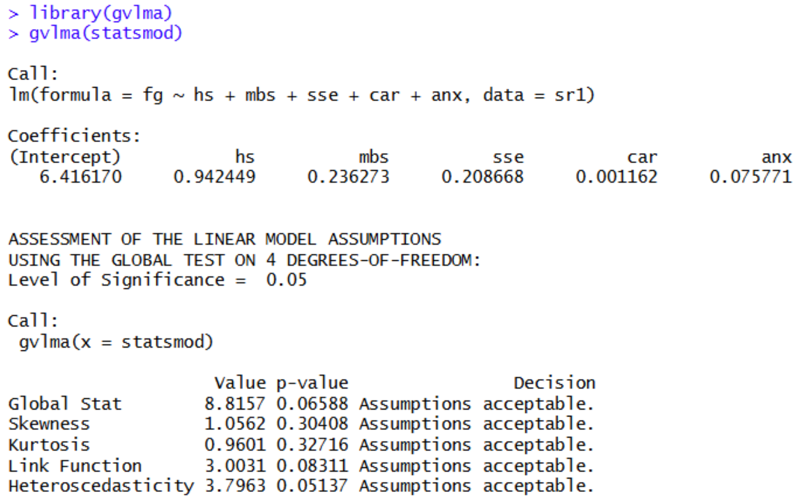


Figure 7. Base *R*'s Residuals vs. Leverage Graph

Regression diagnostics won't check the assumption of independence, as that is an element of research design. For example, if many participants in the study are related, other statistics methods should be used. Overall, the violations of assumptions appear to be moderate at best. The analyst is now forced to make the decision to interpret this analysis or use a robust approach. As visual inspection can be subjective, an objective option is the use of the *gvlma* package using the syntax below:

gvlma(statsmod)

Note the output for *gvlma* below. It says that assumptions of normality are met, so the original model may now be interpreted. Always check assumptions using diagnostic plots. If unsure, double check with *gvlma*.



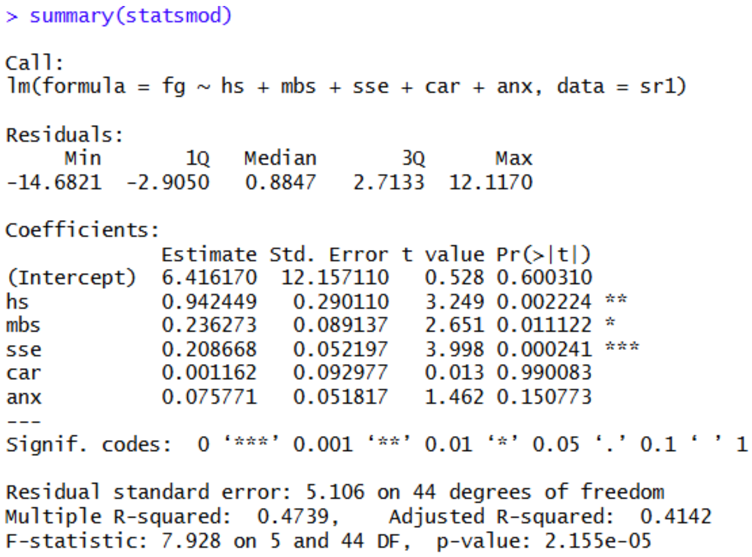
```
> library(gvlma)
> gvlma(statsmod)

Call:
lm(formula = fg ~ hs + mbs + sse + car + anx, data = sr1)

Coefficients:
(Intercept)           hs          mbs          sse          car          anx  
   6.416170     0.942449     0.236273     0.208668     0.001162     0.075771  


ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance =  0.05 

Call:
 gvlma(x = statsmod) 

                    Value p-value                Decision
Global Stat        8.8157 0.06588 Assumptions acceptable.
Skewness           1.0562 0.30408 Assumptions acceptable.
Kurtosis           0.9601 0.32716 Assumptions acceptable.
Link Function      3.0031 0.08311 Assumptions acceptable.
Heteroscedasticity 3.7963 0.05137 Assumptions acceptable.
```

Figure 8. Assumption Check using the *gvlma* Package

Interpretation

Similarly to the earlier linear regression analysis, the base package offers a coefficients table, as well as an abbreviated report in paragraph form. It should be reported as such: $F(5, 44) = 7.9$, $p < .001$, $R^2 = .47$, Adjusted $R^2 = .41$. Using Cohen's (1988) benchmarks, the predictors have an overall large effect on the final grade. From the coefficients table, the following may be determined. As the overall null hypothesis has been rejected, the coefficients table may now be interpreted. Each coefficient in the estimates column may be interpreted as long as the respective p-value is less than .05.

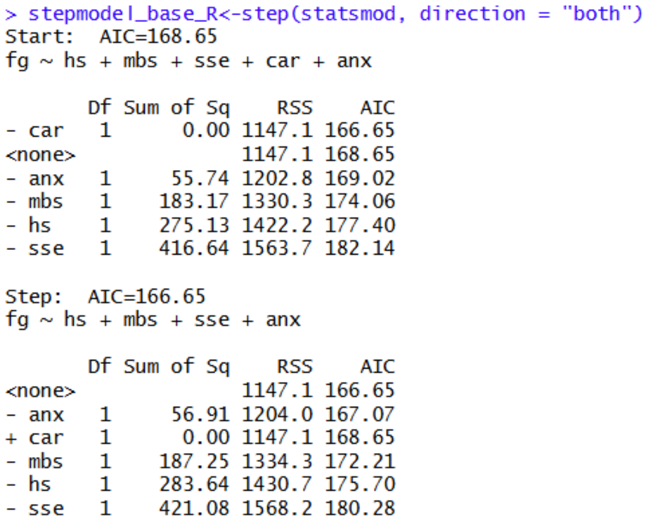
For every hour per week studied (hs), final grade increased by .94 units, keeping other variables constant. For every unit increase in mathematics background score (mbs), final grade increased by about 0.24 units, while keeping other variables constant. For every unit increase in statistical self-efficacy (sse), final grade increased by about 0.21 points, while keeping all others constant. As the p-values for class attendance record (car), and anxiety are > .05, they generally should not be interpreted.



```
> summary(statsmod)

Call:
lm(formula = fg ~ hs + mbs + sse + car + anx, data = sr1)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.6821  -2.9050   0.8847   2.7133  12.1170 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.416170  12.157110   0.528 0.600310    
hs           0.942449   0.290110   3.249 0.002224 ** 
mbs          0.236273   0.089137   2.651 0.011122 *  
sse          0.208668   0.052197   3.998 0.000241 ***
car          0.001162   0.092977   0.013 0.990083    
anx          0.075771   0.051817   1.462 0.150773    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.106 on 44 degrees of freedom
Multiple R-squared:  0.4739,	Adjusted R-squared:  0.4142 
F-statistic: 7.928 on 5 and 44 DF,  p-value: 2.155e-05
```

Figure 9. Multiple Regression Coefficients Table using Base *R*

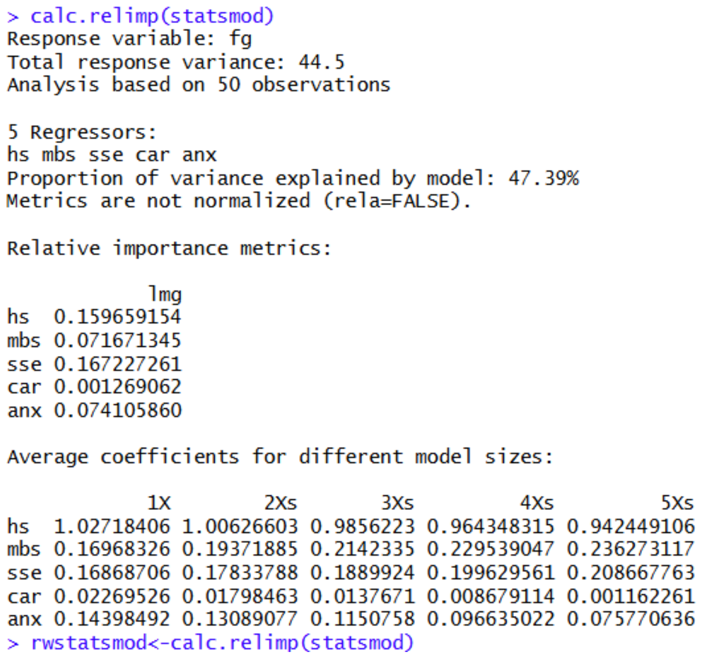
Given that some variables share marginal, or no, relationships with the outcome, a stepwise regression could be considered. The syntax for that analysis is shown below.



```
> stepmodel_base_R<-step(statsmod, direction = "both")
Start:  AIC=168.65
fg ~ hs + mbs + sse + car + anx

       Df Sum of Sq    RSS    AIC
- car   1      0.00 1147.1 166.65
<none>              1147.1 168.65
- anx   1     55.74 1202.8 169.02
- mbs   1    183.17 1330.3 174.06
- hs    1    275.13 1422.2 177.40
- sse   1    416.64 1563.7 182.14

Step:  AIC=166.65
fg ~ hs + mbs + sse + anx

       Df Sum of Sq    RSS    AIC
<none>              1147.1 166.65
- anx   1     56.91 1204.0 167.07
+ car   1      0.00 1147.1 168.65
- mbs   1    187.25 1334.3 172.21
- hs    1    283.64 1430.7 175.70
- sse   1    421.08 1568.2 180.28
```

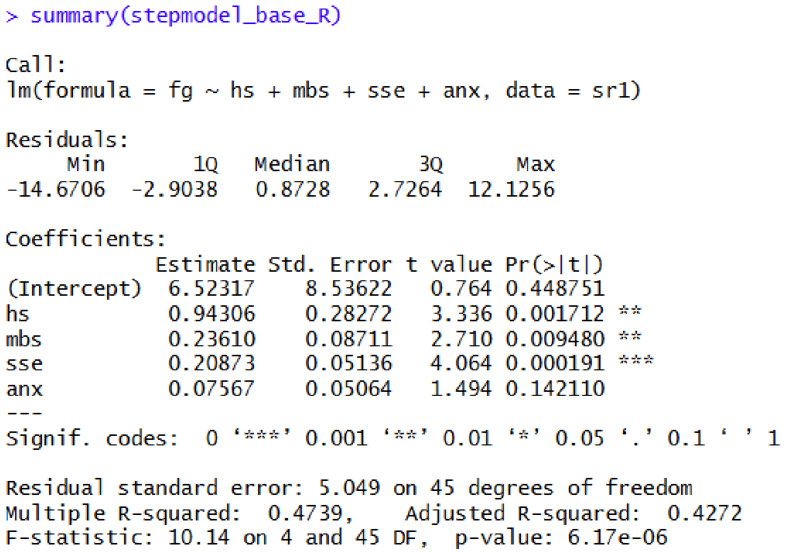
Figure 10. Stepwise Regression Model using Base *R*

Removal of class attendance lowered the AIC (Akaike’s Information Criterion; Akaike, 1974). Note that even though the null hypothesis for anxiety was not rejected, it was not removed from the stepwise model. Use of the *relaimpo* package (Groemping, 2006) provides insight as to why. Under the relative importance metrics in the figure below, statistical self-efficacy and number of hours studied equally contribute to about 33% of the variance in the model. They are followed by anxiety and mathematics background score, each of which contributes about 7% variance, while class attendance record contributes no variance. Also note that these sum to the total 47% of variance explained in the model. This example illustrates the importance of globally evaluating predictors based upon other parameters beyond the p-value. Even though it is not interpreted in this case, it contributes 7% of the variance in the model, which is a weak, bordering on moderate, effect according to Cohen's benchmarks (1988).



```
> calc.relimp(statsmod)
Response variable: fg 
Total response variance: 44.5 
Analysis based on 50 observations 

5 Regressors: 
hs mbs sse car anx 
Proportion of variance explained by model: 47.39%
Metrics are not normalized (rela=FALSE). 

Relative importance metrics: 

            lmg
hs  0.159659154
mbs 0.071671345
sse 0.167227261
car 0.001269062
anx 0.074105860

Average coefficients for different model sizes: 

            1X        2Xs       3Xs         4Xs         5Xs
hs  1.02718406 1.00626603 0.9856223 0.964348315 0.942449106
mbs 0.16968326 0.19371885 0.2142335 0.229539047 0.236273117
sse 0.16868706 0.17833788 0.1889924 0.199629561 0.208667763
car 0.02269526 0.01798463 0.0137671 0.008679114 0.001162261
anx 0.14398492 0.13089077 0.1150758 0.096635022 0.075770636
> rwstatsmod<-calc.relimp(statsmod)
```

Figure 11. Relative Importance of Each Predictor using *relaimpo* Package



```
> summary(stepmodel_base_R)

Call:
lm(formula = fg ~ hs + mbs + sse + anx, data = sr1)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.6706  -2.9038   0.8728   2.7264  12.1256 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.52317    8.53622   0.764 0.448751    
hs           0.94306    0.28272   3.336 0.001712 ** 
mbs          0.23610    0.08711   2.710 0.009480 ** 
sse          0.20873    0.05136   4.064 0.000191 ***
anx          0.07567    0.05064   1.494 0.142110    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.049 on 45 degrees of freedom
Multiple R-squared:  0.4739,	Adjusted R-squared:  0.4272 
F-statistic: 10.14 on 4 and 45 DF,  p-value: 6.17e-06
```

Figure 12. Coefficients of Stepwise Model in Base *R*

If the investigator chooses the stepwise option, there would be a new model to report as shown below. In this case, the report would now be: $F(4, 45) = 10.1$, $p < .001$, $R^2 = .47$, Adjusted $R^2 = .43$, AIC = 166.7. As changes in the model were marginal at best, the rest of this demonstration will use the full model.

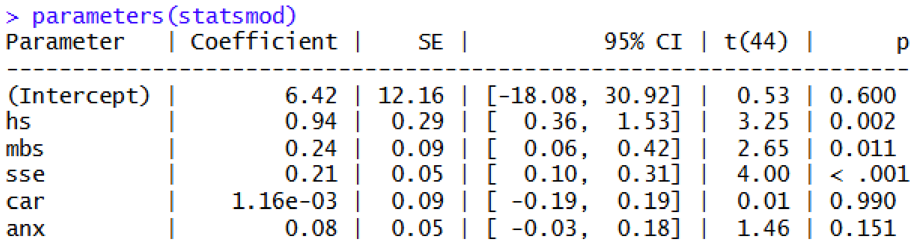
The following section will show another option using *easystats*. It provides several options for both correlation and multiple regression output. The syntaxes for *easystats* are shown in the table below.

Table 7. Multiple Regression using *easystats* and *relaimpo*

Package	Syntax	Purpose
	install.packages(“easystats”)	
	library(easystats)	Overall excellent package for obtaining model summaries, coefficients, collinearity diagnostics, and options for assumption violations.
easystats	correlation(sr1)	Default Pearson’s correlation.
	correlation(sr1, method = “kendall”) correlation(sr1, method = “spearman”)	Non-parametric options.
	parameters(statsmod)	Provides coefficients table.
	parameters(statsmod, bootstrap = 2000)	Same as above, but bootstrapped if assumptions are not met.
	model_performance(statsmod)	Model summary table. Effect sizes are found here.
	check_collinearity(statsmod)	Look for VIF above 10, or tolerance below 0.1.
	report(statsmod)	Optional for script report of findings.
relaimpo	calc.relimp(statsmod)	Shows contribution of each predictor to effect size. Very useful for interpretation.

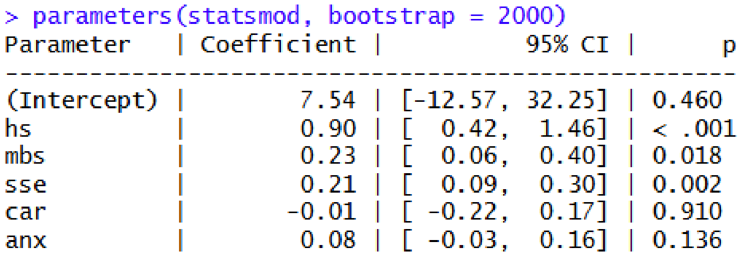
For the *easystats* package, multiple comparisons are controlled for in Pearson's correlation. Hence, the p-values here appear discouraging, especially compared to the output of previous packages. The user is encouraged to focus on the r-values for this output.

The *parameters* command from the *easystats* package offers a coefficients table with confidence intervals included. A notable advantage of *easystats* is the ability to bootstrap the model as a robust option if assumptions are violated. Examples of both analyses are shown below.



```
> parameters(statsmod)
Parameter   | Coefficient |    SE |          95% CI | t(44) |      p
--------------------------------------------------------------------
(Intercept) |        6.42 | 12.16 | [-18.08, 30.92] |  0.53 | 0.600
hs          |        0.94 |  0.29 | [  0.36,  1.53] |  3.25 | 0.002
mbs         |        0.24 |  0.09 | [  0.06,  0.42] |  2.65 | 0.011
sse         |        0.21 |  0.05 | [  0.10,  0.31] |  4.00 | < .001
car         |    1.16e-03 |  0.09 | [ -0.19,  0.19] |  0.01 | 0.990
anx         |        0.08 |  0.05 | [ -0.03,  0.18] |  1.46 | 0.151
```

Figure 13. Linear Regression Coefficients Table using *easystats*

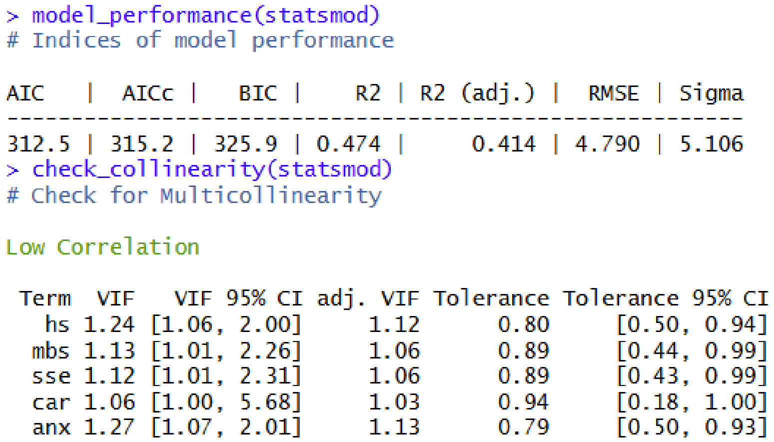


```
> parameters(statsmod, bootstrap = 2000)
Parameter   | Coefficient |          95% CI |      p
----------------------------------------------------
(Intercept) |        7.54 | [-12.57, 32.25] | 0.460
hs          |        0.90 | [  0.42,  1.46] | < .001
mbs         |        0.23 | [  0.06,  0.40] | 0.018
sse         |        0.21 | [  0.09,  0.30] | 0.002
car         |       -0.01 | [ -0.22,  0.17] | 0.910
anx         |        0.08 | [ -0.03,  0.16] | 0.136
```

Figure 14. Example of Coefficients Table after Bootstrapped Regression in *easystats*

Both models should be reported if the results differ. However, as assumptions were found to be met via the *gvlma* package, the bootstrapped analysis is not necessary.

The *easystats* package also offers a model summary table via the *model_performance* syntax, and collinearity diagnostics via the *check_collinearity* syntax as shown below. As discussed previously, R^2 states that 47.4% of variance in the model is explained by the predictors. Collinearity diagnostics are comprised of the variance inflation factor (VIF), and tolerance. A VIF above 10 and/or a tolerance below 0.1 indicate highly correlated predictors. Based on the table below, collinearity diagnostics show no collinearity. If collinearity exists, the researcher should decide which predictors to remove based upon theory and context.



```
> model_performance(statsmod)
# Indices of model performance

AIC     |  AICc |   BIC |    R2 | R2 (adj.) |  RMSE | Sigma
-----------------------------------------------------------
312.5 | 315.2 | 325.9 | 0.474 |     0.414 | 4.790 | 5.106
> check_collinearity(statsmod)
# Check for Multicollinearity

Low Correlation

 Term  VIF   VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
   hs 1.24 [1.06, 2.00]     1.12      0.80     [0.50, 0.94]
  mbs 1.13 [1.01, 2.26]     1.06      0.89     [0.44, 0.99]
  sse 1.12 [1.01, 2.31]     1.06      0.89     [0.43, 0.99]
  car 1.06 [1.00, 5.68]     1.03      0.94     [0.18, 1.00]
  anx 1.27 [1.07, 2.01]     1.13      0.79     [0.50, 0.93]
```

Figure 15. Model Summary and Collinearity Diagnostics Tables using *easystats*

Other packages

Besides *easystats*, an excellent option that offers comparable output is the *olsrr* package. The syntax is shared in table 8 below. Its output is similar to those of graphical user interface (GUI) packages like SPSS, Excel and others. Thus, it may be excellent for teaching, especially for students used to these programs. However, *olsrr* does not do robust or non-parametric alternatives like the *easystats* package. This may be ameliorated by using other packages, including bootstrapping via the *easystats* package as described earlier. The *robustbase* package is another option for robust regression in case of violated assumptions. The *visreg* package is another option for obtaining scatterplots showing the relationship of the outcome with each predictor. Finally, *sjPlot* visualizes the coefficient table, showing the relative contribution of each predictor. The syntaxes for these options are shown in Table 8.

Table 8. Other Options for Regression

Package	Syntax	Purpose
olsrr	library(olsrr) library(robustbase) library(sjPlot)	
	ols_regress(statsmod)	Model summary, ANOVA table, coefficients table.
	ols_vif_tol(statsmod)	Collinearity diagnostics
	ols_coll_diag(statsmod)	Same as above, but includes eigenvalue table.
	ols_plot_diagnostics_statsmod	Visualization of residuals; overall excellent

Package	Syntax	Purpose
		package for this purpose.
robustbase	statsmodrobust<-lmrob(statsmod)	Another good option for violated assumptions in regression.
sjPlot	plot_model(statsmod, show.values = TRUE)	Visualizes relative contribution of predictors.
visreg	visreg(statsmod)	Provides individual scatterplots of outcome vs each predictor.

Discussion

The objective of this paper was to provide options for doing PPMC and OLSR using *R* and its add-on packages. Options for performing both analyses were showcased using a synthetic dataset, including options for violation of assumptions for both statistical methods. Options for visualization were also discussed.

The base *R* package can perform parametric and nonparametric correlation analyses but does not provide p-values. However, the *Hmisc* and *psych* packages are more comprehensive options. The *lattice* and *DescTools* packages are also able to provide single scatterplots, and several packages can provide scatterplot matrices including *psych*, *DescTools*, *GGally*, and *PerformanceAnalytics*. The *qgraph* package is good for visualizing relationships as a network.

Base *R* also offers OLSR but does not offer collinearity diagnostics. Other comprehensive packages for OLSR exist, including the *easystats* environment, and *olsrr*. The *gvlma* package may be used to check assumptions, and other options for violated assumptions include *easystats*, and the *robustbase* package. The *relaimpo* package is invaluable for obtaining each predictor’s contribution to the effect size. The *visreg* package is good for creating individual scatterplots of an outcome and several predictors, while the *sjPlot* package is good for comparing the relative importance of predictors.

Overall, *R* has flexible options for PPMC, OLSR, and alternatives for violated assumptions. Beginners may start with base *R* for simplicity, explore packages like *psych* for correlation, and *easystats* or *olsrr* for regression. Others such as *gvlma*, *relaimpo*, *GGally*, and *sjPlot* enhance interpretation. The *easystats* environment is well-rounded for correlation and regression analyses.

Limitations

Although multiple regression is not taught in many introductory statistics courses, demonstrating this method may help students to understand how correlation matrices are used as exploratory analyses for more advanced methods. Instructors and students are encouraged to use or omit it as they see fit. This paper does not contain a comprehensive list of packages that perform correlation and regression. For example, another excellent package for scatterplots and regression is the *car* package (Fox & Weisberg, 2019), and there are several other packages that perform either robust, bootstrapped, or non-parametric forms of multiple regression. The *ggplot* family of packages are also excellent options for visualization. However, the packages described in this paper were chosen because of their rich output, and relative ease of use. Additionally, the full capability of each package was not described. This was intentional, to make the content more digestible for new *R* users. Readers are fully encouraged to explore these, and other packages, to practice as they see fit for their respective levels of interest and learning.

Conclusion

The *R* ecosystem is fully equipped for options to perform PPMC and OSLR. For quick correlation, use base *R* or *psych.* Practice regression using base *R,* checking overall assumptions using *gvlma*, and importance of predictors using *relaimpo*. Users may then progress to other packages like *easystats* or *olsrr* at their discretion.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, AC-19*, 716–723.

Breheny P., Burchett W. (2017). Visualization of regression models using visreg. *The R Journal*, 9(2), 56–71. https://doi.org/10.32614/RJ-2017-046

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Cruise, R. J., Cash, R. W., & Bolton, D. L. (1985, August). Development and validation of an instrument to measure statistical anxiety. In *American Statistical Association Proceedings of the Section on Statistical Education* (Vol. 4, No. 3, pp. 92-97).

Epskamp, S., Cramer, A. O. J., Waldorp, L. J., Schmittmann, V. D., & Borsboom, D. (2012). *qgraph: Network visualizations of relationships in psychometric data.* Journal of Statistical Software, 48(4), 1–18. https://www.jstatsoft.org/v48/i04/

Forrester, C., Schwikert, S., Foster, J., & Corwin, L. (2022). Undergraduate R programming anxiety in ecology: persistent gender gaps and coping strategies. *CBE—Life Sciences Education*, *21*(2). https://doi.org/10.1187/cbe.21-05-0133

Fox, J., & Weisberg, S. (2019). *An R Companion to Applied Regression*, Third edition. Sage, Thousand Oaks CA. https://www.john-fox.ca/Companion/

Groemping, U. (2006). Relative Importance for Linear Regression in R: The Package *relaimpo*. *Journal of Statistical Software, 17*(1), 1--27.

Harrell Jr., F. (2025). *Hmisc: Harrell Miscellaneous*. R package version 5.2-4. [Computer software]

Hebbali, A. (2024). *olsrr: Tools for Building OLS Regression Models*. R package version 0.6.1.9000,
https://github.com/rsquaredacademy/olsrr, https://olsrr.rsquaredacademy.com/

Lüdecke, D. (2025). *sjPlot: Data Visualization for Statistics in Social Science*. R package version 2.9.0. https://CRAN.R-project.org/package=sjPlot

Lüdecke, D., Patil, I., Ben-Shachar, M. S., Wiernik, B. M., Bacher, E., Thériault, R., & Makowski, D. (2022). *easystats: Framework for Easy Statistical Modeling, Visualization, and Reporting* (R package). https://easystats.github.io/easystats/

Maechler, M., Rousseeuw, P., Croux, C., Todorov, V., Ruckstuhl, A., Salibian-Barrera, M., Verbeke, T., Koller, M., Conceicao, E. L. T., & Anna di Palma, M. (2025). *robustbase: Basic Robust Statistics* (R package version 0.99-6). http://robustbase.r-forge.r-project.org/.

Pena, E. A., & Slate, E. H. (2019). *gvlma: Global validation of linear models assumptions (*R package version 1.0.0.3). https://cran.r-project.org/package=gvlma

Peterson, B. G., & Carl, P. (2025). *PerformanceAnalytics: Econometric tools for performance and risk analysis.* R package version 2.0.8-1

Posit (2025). *RStudio Desktop*. Posit. https://posit.co/download/rstudio-desktop/

R Core Team (2025). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Revelle, W. (2025). *psych: Procedures for Psychological, Psychometric, and Personality Research*. Northwestern University, Evanston, Illinois. R package version 2.5.6. https://CRAN.R-project.org/package=psych

Sarkar, D. (2008). Lattice: Multivariate Data Visualization with R. Springer. http://lmdvr.r-forge.r-project.org

Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., Elberg, A., & Crowley, J. (2025). *GGally: Extension to 'ggplot2'* (Version 2.4.0) [R package]. https://ggobi.github.io/ggally/

Signorell, A. (2025). *DescTools: Tools for Descriptive Statistics* (R package version 0.99.60) [Computer software]. https://doi.org/10.32614/CRAN.package.DescTools

Stanton, J. M. (2001). Galton, Pearson, and the Peas: A Brief History of Linear Regression for Statistics Instructors. *Journal of Statistics Education*, *9*(3). https://doi.org/10.1080/10691898.2001.11910537

Wickham, H., François, R., Henry, L., & Müller, K. (2019). *A Grammar of Data Manipulation* [R package dplyr version 0.8.3]. R-Project.org. https://cran.r-project.org/package=dplyr

Citation

Lindsay, K. G. (2026). Correlation and regression. In *Applied Statistics 101 using R: A primer for an introductory class in statistical programming* (pp. 30-50). ISTES.

Chapter 4 - Comparing Two Means with Z-Tests and T-Tests

Chapter Highlights

- Learn differences between z-tests and t-tests.
- Compute z-tests and t-tests using base R, BSDA, and *rstatix*.
- Interpret effect sizes for group comparisons (e.g., Cohen's D) using *effectsize* package.
- Visualize using *DescTools* and other options.
- Includes ANOVA syntax as an alternative approach for two means.
- Learn APA-style reporting of results.

Introduction

This tutorial introduces readers to comparing two means using z-tests and t-tests in *R*. Readers will use practical coding examples to compute one-sample and two-sample z-tests; one-sample, independent, and dependent t-tests; and obtain effect sizes. The z-tests for means are comprised of statistical methods that compare two means when population standard deviations are known.

When population standard deviations are unknown, the t-tests are used. *R* is a statistical programming language and environment that consists of a base package, and add-on packages that may optimize workflow. This tutorial introduces beginners to z-tests and t-tests using base *R*, and various add-on packages.

The one-sample z-test is used to compare a sample mean to a population mean when the standard deviation is known. The two-sample z-test compares the means of two samples when the population standard deviations are known for the variables being compared.

The t-test is a commonly used statistical method used for comparing two means. It was invented by William Sealy Gossett in the early 1900's, as an objective method of quality control for the Guinness brewery. As he was not allowed to publish using his real name or reveal that he worked for the company, he assumed the pen name "Student". In contrast to the z-test, the standard deviation of the population is unknown. The one-sample t-test compares a sample mean to a population mean when the standard deviation is unknown. The independent t-test compares two means, when population standard deviations are unknown, and the dependent t-test compares the means of the same individuals tested at two times, or under two conditions.

Assumptions for parametric tests include continuous data that are approximately normally distributed, random sampling, a known standard deviation, and (for z-tests) a known population standard deviation. Equality of variances and independent observations are assumptions for the independent t-test, while the paired or dependent t-test assumes related observations (e.g., same participants measured twice). Effect sizes for t-tests using Cohen's D and eta squared are listed below (Cohen, 1988). While these benchmarks may not be appropriate for all scenarios, they will be used to add context to this tutorial.

Table 1. Benchmarks for Cohen's D and Eta Squared

Threshold	d	η^2
Small	.2	.01
Medium	.5	.06
Large	.8	.14

The objectives of this tutorial are to teach the reader how to:

- Compute and interpret one and two-sample z-tests for means using *R*.
- Use one-sample, independent, dependent t-tests, and one-way ANOVA in *R* to compare two means.
- Compute effect sizes.
- Where applicable, compare output from base *R*, and other packages.

Introduction to R

The statistical programming language *R* (R Core Team, 2025) may be used for statistical analysis using z-tests and t-tests. *R* is a free and open-source statistical language and environment. It is supported and used by a wide range of scholars, analysts, and computer programmers, many of whom submit add-on packages suited for a variety of tasks. Packages may be found for virtually any statistical procedure, in addition to related tasks such as geographic information systems (Lovelace et al., 2019; University of Toronto Libraries, 2024). *R* may be used in tandem with an Integrated Development Environment, such as *RStudio* (Posit, 2025). While not necessary, benefits of its use include management of datasets, objects, and packages.

Objects and Functions

An object in *R* is a method of abbreviating and storing some entity found in R. For example, a dataset may be assigned as an object. For example, the syntax "scores <- c(15, 1.4, 37)" creates an object called "scores", that stores three scores. By doing this, it now becomes easier to subset the original data, and the dataset may then be more easily summarized or analyzed. Data subsets, summaries and analyses, may also be assigned as objects to be better stored and managed. *RStudio* is also helpful in this regard. Objects are created using the chosen abbreviation followed by “<-“ or “=”. The command “onesamplet<-t.test(one_s$finalscore, mu = 75)” is used to assign an object to the one-sample t-test analysis. The effect size is then obtained using the object, “cohens_d(onesamplet)”. An object may also be assigned to the latter command if the reader wishes. Users are encouraged to create, organize, and manage objects at their discretion, and other examples follow in the tables and figures of this paper. A function is defined as some sort of operation or process. Later in this paper, a function will be used to get descriptive statistics using the base package.

Data Sets and Premises

There are five datasets associated with this paper, which will be provided as supplementary material. The data may be obtained using this link: https://osf.io/mrw5v/overview. They were synthesized for instructional purposes and contain no real student data. For the dataset associated with the one-sample z and one-sample-t-test, a statistics teacher wants to know if the final exam score in a class differs from the population mean. The standard deviation is also provided for the z-test. For the two-sample z-test, a teacher wishes to compare a new method with a traditional teaching method on a standardized statistics exam score. For the paired t-test, students rated their confidence before and after taking a statistics course. These data are offered in long and wide formats. In long format, both levels of time are found in the same column, whereas the times are found in separate columns for wide format. Finally, the premise for the independent t-test dataset is a comparison between the final examination scores between students who attended and did not attend optional review sessions. The datasets are easy to use for practicing hypothesis testing. A summary of the z-tests and t-tests covered in this article is listed below.

Table 2. Null Hypotheses

Statistical Test	Null Hypothesis
One-sample z	Sample mean = population mean, with known σ^*
Two-sample z	Equal group means, with known σ
One-sample t	Sample mean = population mean, σ unknown.
Independent t	Equal group means; S^{**} is used.
Dependent t	Cohort means equal across times or contexts. S is used.

$^*\sigma$ = population standard deviation; $^{**}S$ = sample standard deviation.

Method

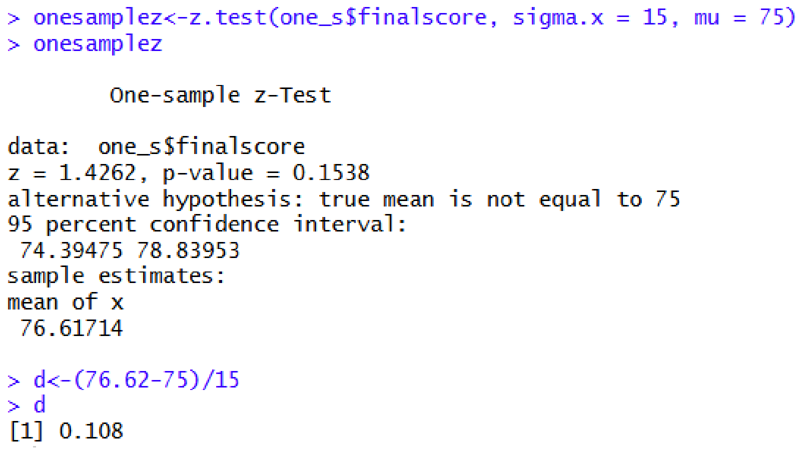
The z-tests

Users may upload the data sets into *RStudio*, and practice using the syntax in the following table. The *BSDA* (Basic Statistics and Data Analysis; Arnholt & Evans, 2023) package is used to calculate z-tests in this section, while the base *R* package is used to calculate the one-sample t-test. The *effectsize* package (Ben-Shachar et al., 2020) will be used to obtain effect sizes, where possible.

Table 3. Syntax for z-tests and One-sample t-test

Package	Syntax	Purpose
	install.packages(“BSDA”) install.packages(“effectsize”) library(BSDA) library(effectsize)	
Base *R*	onesamplet<-t.test(one_s$finalscore, mu = 75)	Calculates one-sample t-test
	d = (M – μ)/σ	Manually calculates effect size for one-sample z-test.
BSDA	onesamplez<-z.test(one_s$finalscore, sigma.x = 15, mu = 75)	Calculates one-sample z-test.
	twosamplez<-z.test(two_szNew_Method, two_szTraditional_Method, sigma.x = 10.2, sigma.y = 9.8)	Calculates two-sample z-test
effectsize	cohens_d(onesamplet)	Effect size for one-sample t-test.
	cohens_d(two_szNew_Method, two_szTraditional_Method, paired = FALSE)	Effect size for two-sample z

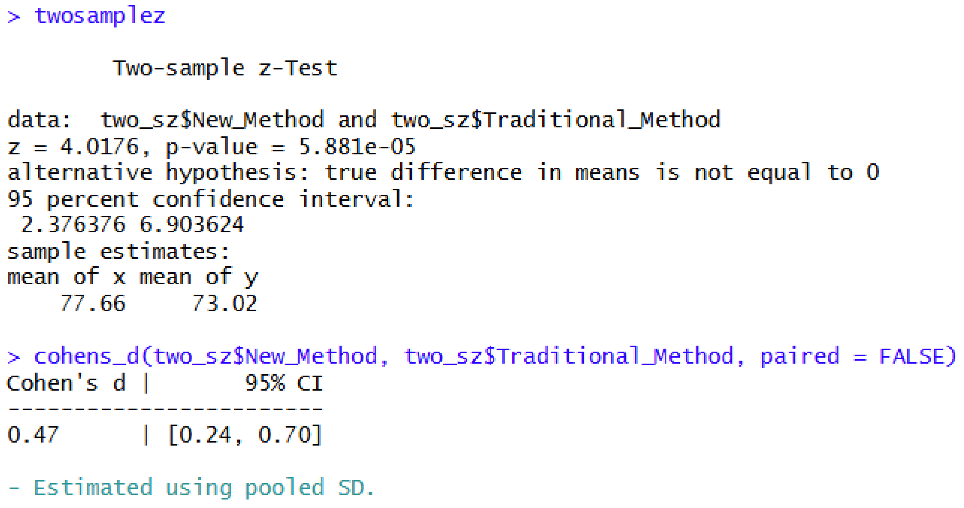
The one-sample z-test below uses the “one_s” data set and fails to reject the null hypothesis at the .05 level. Also note the small effect (using Cohen's benchmarks), suggesting that the sample mean's difference was by chance only. Report as follows: $z(174) = 1.4$, $p > .05$, d = 0.1. Note that the degrees of freedom is calculated by subtracting 1 from the sample size.



```
> onesamplez<-z.test(one_s$finalscore, sigma.x = 15, mu = 75)
> onesamplez

        One-sample z-Test

data:  one_s$finalscore
z = 1.4262, p-value = 0.1538
alternative hypothesis: true mean is not equal to 75
95 percent confidence interval:
 74.39475 78.83953
sample estimates:
mean of x 
 76.61714 

> d<-(76.62-75)/15
> d
[1] 0.108
```

Figure 1. One-sample z-test using *BSDA*

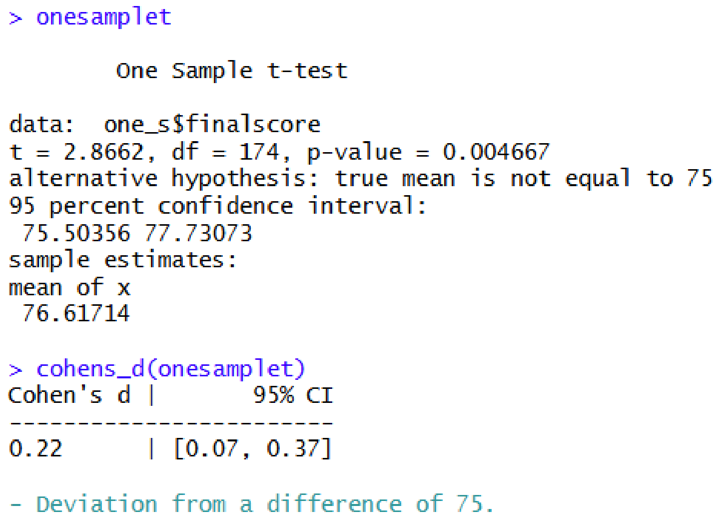
On the other hand, the two-sample z-test shows that the novel teaching method resulted in a higher average score than a traditional method, with a small, bordering on medium, effect. Report as follows: $z(148) = 4.0$, $p < .001$, d = 0.47.



```
> twosamplez

        Two-sample z-Test

data:  two_sz$New_Method and two_sz$Traditional_Method
z = 4.0176, p-value = 5.881e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.376376 6.903624
sample estimates:
mean of x mean of y 
    77.66     73.02 

> cohens_d(two_sz$New_Method, two_sz$Traditional_Method, paired = FALSE)
Cohen's d |       95% CI
------------------------
0.47      | [0.24, 0.70]

- Estimated using pooled SD.
```

Figure 2. Two-sample z-test using *BSDA*

Figure 3 below shows the output for the one-sample t-test. Note that this is the same dataset used for the previous one-sample z-test, albeit with σ excluded. Using this method, the null hypothesis is rejected with a small effect, suggesting the sample mean is marginally higher than the population mean. Report as follows: $t(174) = 2.8$, $p < .01$, d = .22.



```
> onesamplet

        One Sample t-test

data:  one_s$finalscore
t = 2.8662, df = 174, p-value = 0.004667
alternative hypothesis: true mean is not equal to 75
95 percent confidence interval:
 75.50356 77.73073
sample estimates:
mean of x
 76.61714

> cohens_d(onesamplet)
Cohen's d |       95% CI
-------------------------
0.22      | [0.07, 0.37]

- Deviation from a difference of 75.
```

Figure 3. One-sample t-test using Base *R*

Independent t-test

The following table shows the syntax for the independent t-test using base *R* and several packages. The *rstatix* package (Kassambara, 2025) provides all parameters for the t-test, while the base *R* package is used in tandem with other packages as shown below.

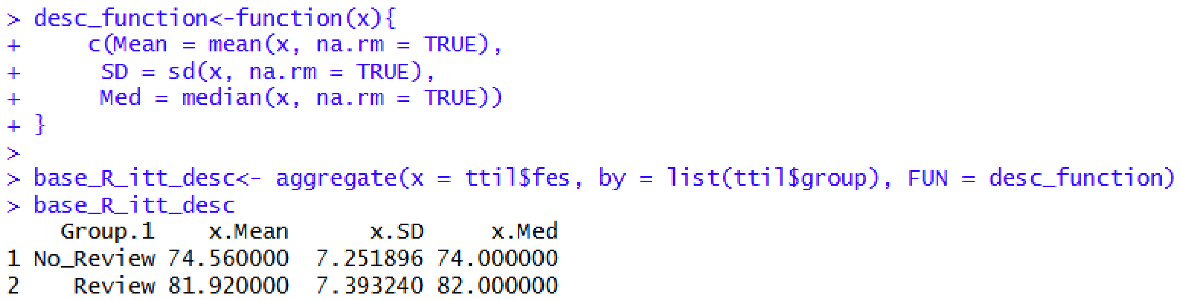
Table 4. Independent T-test Syntax

Package	Syntax	Purpose
	install.packages("rstatix") library(rstatix)	
Base *R*	aggregate(x = ttil$fes, by = list(ttil$group), FUN = mean)	Command to obtain means by group. Replace with sd, median, etc. for other parameters.
	t.test(fes~group, data = ttil, var.equal = TRUE)	Student's T-test; analysis also provides group means.
	itt_base_R<-t.test(fes~group, data = ttil, var.equal = TRUE)	Assigns an object to the syntax above.
	itt_base_R_w<-t.test(fes~group, data = ttil, var.equal = FALSE)	Welch's T-test in case of unequal variances.
effectsize	cohens_d(fes~group, data = ttil)	Cohen's D; note similarity in commands to rstatix package.

Package	Syntax	Purpose
rstatix	itt_descriptives<-ttil %>% + group_by(group) %>% + get_summary_stats(fes)	Descriptive statistics by group.
	levene_test(ttil, fes~group)	Levene's Test for equality of variances.
	itt_rstatix<-t_test(ttil, fes~group)	Student's T-test
	itt_rstatix_w<-t_test(ttil, fes~group, var.equal = FALSE)	Welch's T-test.
	esd<-cohens_d(ttil, fes~group)	Cohen's D
DescTools	Desc(fes~group, data = ttil)	Best option for quick descriptives and visualization.
	LeveneTest(ttil$fes, ttil$group)	Levene's test

Descriptive Statistics

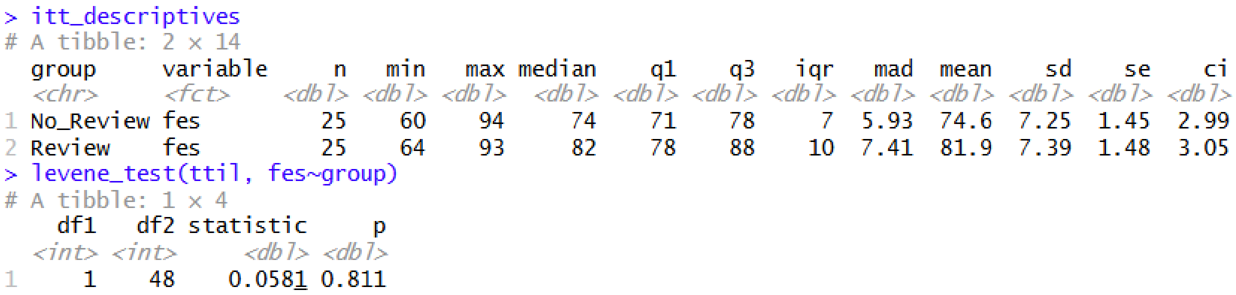
Getting a descriptive statistics table using the base *R* package requires relatively lengthy syntax. It may be obtained by using the *aggregate* command each for mean, standard deviation, median, etc. Additional parameters may be obtained from one aggregate command, if they are wrapped into a function as shown below.



```
> desc_function<-function(x){
+     c(Mean = mean(x, na.rm = TRUE),
+      SD = sd(x, na.rm = TRUE),
+      Med = median(x, na.rm = TRUE))
+ }
>
> base_R_itt_desc<- aggregate(x = ttil$fes, by = list(ttil$group), FUN = desc_function)
> base_R_itt_desc
    Group.1    x.Mean      x.SD     x.Med
1 No_Review 74.560000  7.251896 74.000000
2    Review 81.920000  7.393240 82.000000
```

Figure 4. Function used to Create Descriptive Statistics using Base *R*

In contrast, the *rstatix* package offers a comprehensive descriptive statistics table split by group, the Levene's test for equality of variances, t-tests, and the Cohen's D effect size. As group sizes are equal, the Levene's Test is shown for demonstrative purposes only. Unless sample sizes are equal, check Levene's Test before deciding between the Student's and Welch's tests (Field, 2013). If in doubt, run both tests and report both outcomes.



```
> itt_descriptives
# A tibble: 2 × 14
  group     variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se    ci
  <chr>     <fct>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 No_Review fes         25    60    94     74    71    78     7  5.93  74.6  7.25  1.45  2.99
2 Review    fes         25    64    93     82    78    88    10  7.41  81.9  7.39  1.48  3.05
> levene_test(ttil, fes~group)
# A tibble: 1 × 4
    df1   df2 statistic     p
  <int> <int>     <dbl> <dbl>
1     1    48    0.0581 0.811
```

Figure 5. Descriptive Statistics and the Levene's Test using *rstatix*

An even faster option for obtaining descriptive statistics is the *DescTools* package (Signorell, 2025). With short syntax, the user can quickly obtain descriptive statistics and boxplots.

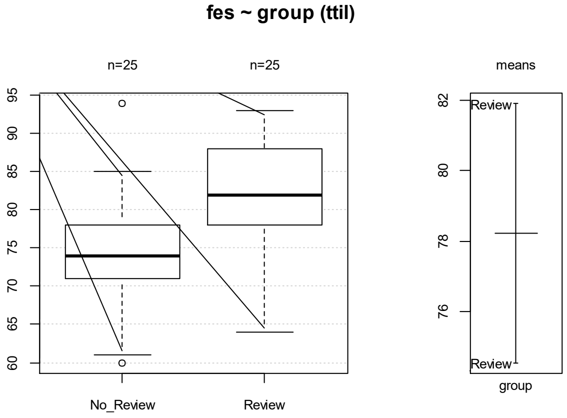
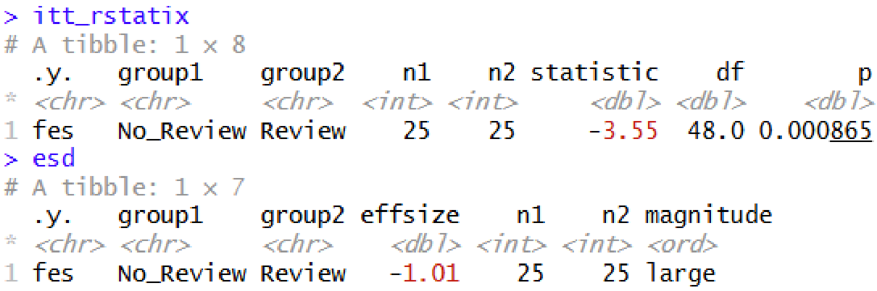


Figure 6. Boxplots by Group using *DescTools*

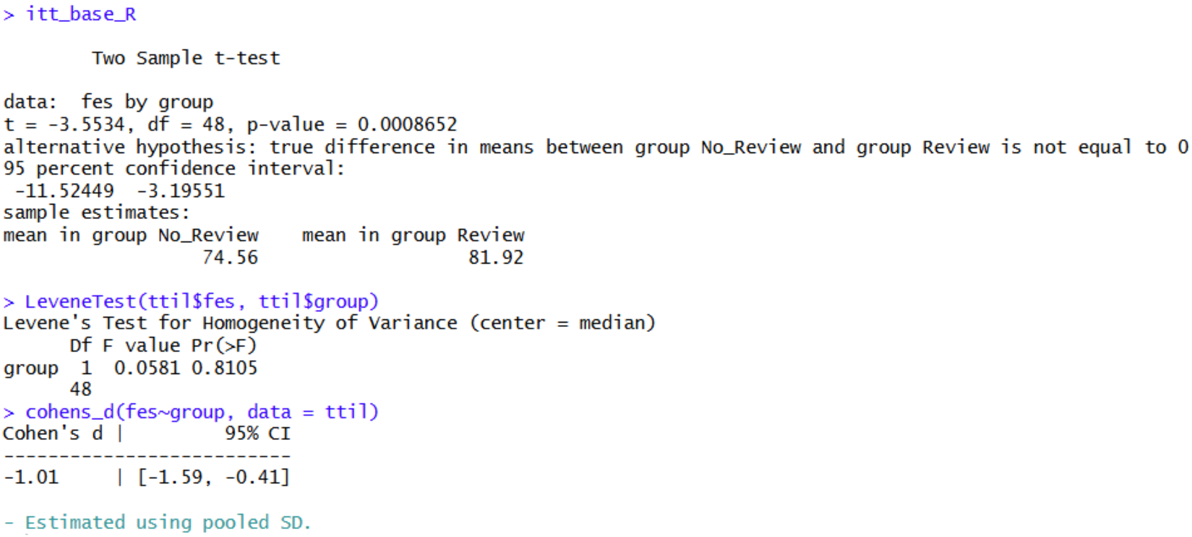
The t-test and Cohen's D show a rejection of the null hypothesis, and a large effect respectively. Report as follows: $t(48) = 3.55$, $p < .001$, $d = 1$. Only the magnitudes of the t-statistic and Cohen's D need to be reported. There was a difference between the students who attended the review sessions and those who did not. Those who attended the review session had a higher score.



```
> itt_rstatix
# A tibble: 1 × 8
  .y.   group1    group2     n1    n2 statistic    df        p
* <chr> <chr>     <chr>   <int> <int>     <dbl> <dbl>    <dbl>
1 fes   No_Review Review     25    25     -3.55  48.0 0.000865
> esd
# A tibble: 1 × 7
  .y.   group1    group2 effsize    n1    n2 magnitude
* <chr> <chr>     <chr>    <dbl> <int> <int> <ord>
1 fes   No_Review Review   -1.01    25    25 large
```

Figure 7. Independent t-test and Cohen's D using *rstatix*

Similar outputs for t-test and effect sizes may be obtained using the base *R* and *effectsize* packages, respectively. Although Levene's test is not offered using the base *R* package, it may be obtained via *DescTools*.



```
> itt_base_R

        Two Sample t-test

data:  fes by group
t = -3.5534, df = 48, p-value = 0.0008652
alternative hypothesis: true difference in means between group No_Review and group Review is not equal to 0
95 percent confidence interval:
 -11.52449  -3.19551
sample estimates:
mean in group No_Review    mean in group Review 
                  74.56                   81.92 

> LeveneTest(ttil$fes, ttil$group)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  0.0581 0.8105
      48               
> cohens_d(fes~group, data = ttil)
Cohen's d |         95% CI
--------------------------
-1.01     | [-1.59, -0.41]

- Estimated using pooled SD.
```

Figure 8. Independent t-test, Levene's Test, & Cohen's D using Base *R*, *DescTools* & *effectsize*

Dependent t-test

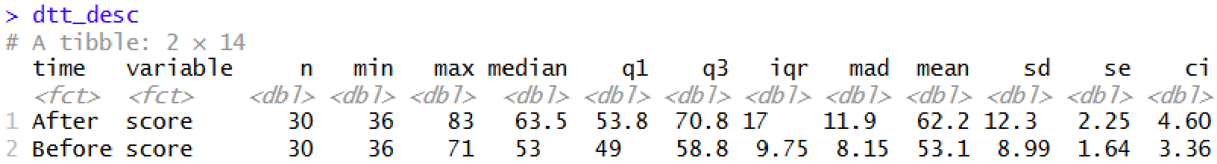
The dependent t-test is used to compare two means, typically from the same group measured twice, or to compare related groups (e.g., twins). Several packages offer options for the use of the dependent t-test. This section will compare the base *R* package with the *rstatix* package. Similarly to the independent t-test, the *rstatix* package was chosen because it provides a comprehensive descriptive statistics table split by time or context, performs the t-test, and provides effect size D.

Table 5. Dependent T-Test Syntax

Package	Syntax	Purpose
Base *R*	dtt_base_R<-t.test(ttpw$conf_before, ttpw$conf_after, paired = TRUE)	Dependent T-test; uses wide format.
effectsize	cohens_d(ttpw$conf_before, ttpw$conf_after, paired = TRUE)	Calculates effect size using wide format.
	repeated_measures_d(ttpw$conf_before,	Adjusts for small

Package	Syntax	Purpose
	ttpw$conf_after, paired = TRUE)	samples.
rstatix	dtt_desc<-ttpl %>% + group_by(time) %>% + get_summary_stats(score)	Descriptive statistics for each time/context.
	dtt_rstatix<-t_test(ttpl, score~time, paired = TRUE)	Dependent T-test; uses long format.
	ttpl %>% t_test(score~time, paired = TRUE)	Alternate command.
	esdp<-cohens_d(ttpl, score~time, paired = TRUE)	Cohen's D using long format.
ggstatsplot	ggwithinstats(data = ttpl, x = "time", y = "score", id = "ID", type = "parametric", title = "Statistics confidence before vs. after Statistics class")	Visualization.

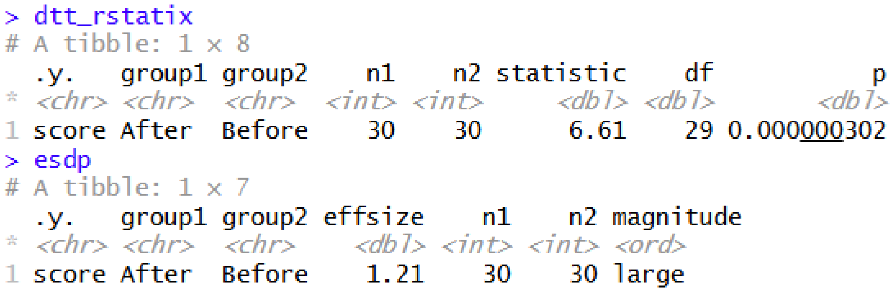
The *rstatix* package also offers descriptive statistics by time or context for paired t-tests with data in the long format.



```
> dtt_desc
# A tibble: 2 × 14
  time   variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se    ci
  <fct>  <fct>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 After  score       30    36    83   63.5  53.8  70.8 17     11.9  62.2 12.3   2.25  4.60
2 Before score       30    36    71   53    49    58.8  9.75  8.15  53.1  8.99  1.64  3.36
```

Figure 9. Descriptive Statistics for the Dependent t-test using *rstatix*

Like the independent t-test function, it also offers a t-table and the Cohen's D effect size. The null hypothesis is rejected at the $p < .001$ level, suggesting that the confidence differs before and after the statistics course. The means show that confidence is higher after the course. Report as follows: $t(29) = 6.6$, $p < .001$, $d = 1.2$.



```
> dtt_rstatix
# A tibble: 1 × 8
  .y.   group1 group2    n1    n2 statistic    df           p
* <chr> <chr>  <chr>  <int> <int>     <dbl> <dbl>       <dbl>
1 score After  Before    30    30      6.61    29 0.000000302
> esdp
# A tibble: 1 × 7
  .y.   group1 group2 effsize    n1    n2 magnitude
* <chr> <chr>  <chr>    <dbl> <int> <int> <ord>
1 score After  Before    1.21    30    30 large
```

Figure 10. Dependent t-test and Cohen's D using *rstatix*

The *ggstatsplot* package (Patil, 2021) is an excellent option that provides a statistical report of the outcome, along with plots. It requires more coding than *DescTools*, but is more customizable. It provides an opportunity for students to practice coding to create plots.

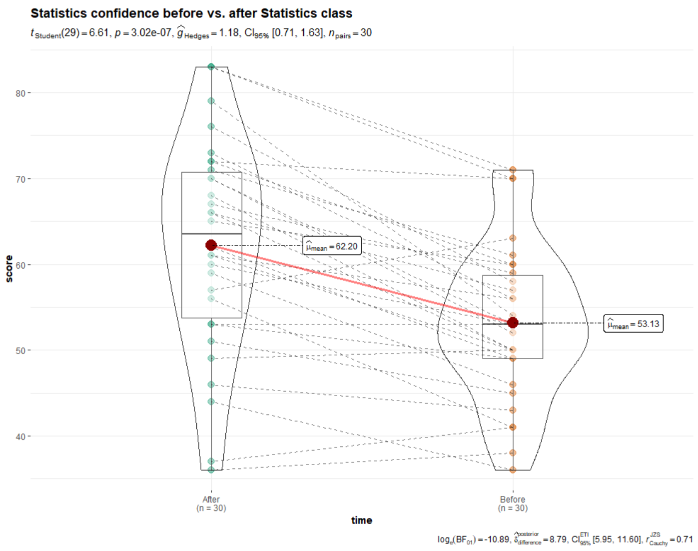


Figure 11: Visualization using *ggstatsplot*

Analysis of Variance (ANOVA) Approach

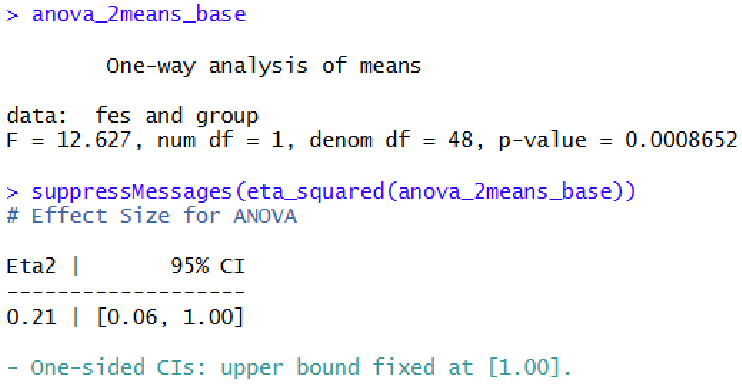
Analysis of variance may also be used to compare two means. he advantage of using this approach is the acquisition of the eta squared (η^2) effect size, which quantifies the percentage of variance explained by the independent variable. This is especially useful for effect size interpretation. This option may be found using both the base *R* package, and *rstatix*. The syntax is shown in the table below.

Table 7. ANOVA syntax for Two Means

Package	Syntax	Purpose
Base *R*	anova_2means_base<-oneway.test(fes~group, data = ttil, var.equal = TRUE)	One way independent ANOVA
effectsize	eta_squared(anova_2means_base)	Effect size obtained using object created above. Partial eta squared is equivalent to general eta squared for one independent variable.

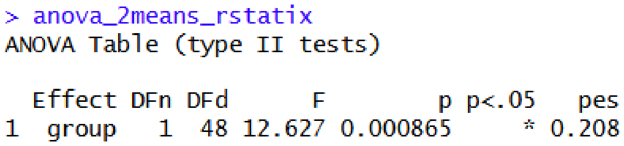
rstatix	anova_2means_rstatix<-anova_test(data = ttil, dv = fes, between = group, effect.size = "pes")	Provides effect size (general eta squared) in ANOVA table.
	anova_2meansRM_rstatix<-anova_test(data = ttpl, dv = score, wid = ID, within = time)	One-way repeated measures ANOVA

As expected, note the similar ANOVA output for both packages. The main difference is that base *R* will need another package (e.g., *effectsize*) to obtain eta squared, whereas *rstatix* appends it to the ANOVA table. Report as follows: $F(1, 48) = 12.6$, $p < .001$, $\eta^2 = .21$. Group membership explains 21 % of the variation in scores.



```
> anova_2means_base

        One-way analysis of means

data:  fes and group
F = 12.627, num df = 1, denom df = 48, p-value = 0.0008652

> suppressMessages(eta_squared(anova_2means_base))
# Effect Size for ANOVA

Eta2 |       95% CI
-------------------
0.21 | [0.06, 1.00]

- One-sided CIs: upper bound fixed at [1.00].
```

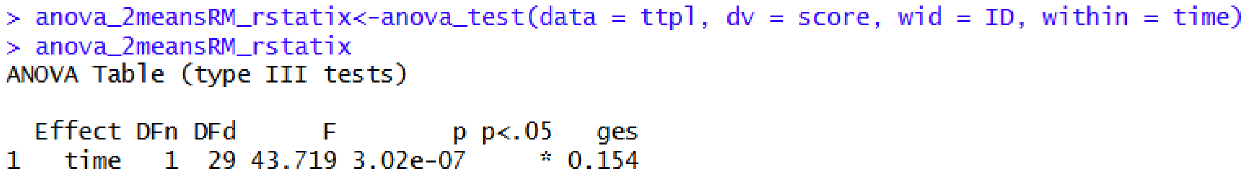
Figure 12: Independent ANOVA using Base *R*



```
> anova_2means_rstatix
ANOVA Table (type II tests)

  Effect DFn DFd      F        p p<.05   pes
1  group   1  48 12.627 0.000865     * 0.208
```

Figure 13. Independent ANOVA using *rstatix*

Repeated measures ANOVA is easily performed in *rstatix* as well. Report as follows: $F(1, 29) = 43.7$, $p < .001$, $\eta^2 = .15$. Time explains 15 % of the variation in confidence.



```
> anova_2meansRM_rstatix<-anova_test(data = ttpl, dv = score, wid = ID, within = time)
> anova_2meansRM_rstatix
ANOVA Table (type III tests)

  Effect DFn DFd      F        p p<.05   ges
1   time   1  29 43.719 3.02e-07     * 0.154
```

Figure 13. Dependent (Repeated Measures) ANOVA using *rstatix*

Discussion and Conclusion

The base *R* package offers one-sample t-tests in addition to paired and independent options. Descriptive statistics for t-tests can be best obtained by the *DescTools* package, which is also a good option for quick visualization of subgroups. The *BSDA* package is the option for using z-tests, and the *effectsize* package is an excellent option for getting most effect sizes for z-tests and t-tests. The *rstatix* package is a comprehensive package for utilizing independent and paired t-tests. It provides descriptive statistics, the Levene's test, t-tests, and effect sizes. It is also a good choice for using independent and repeated measures ANOVA to analyze differences between two means. Overall, *R* offers multiple options for comparing two means. Beginners should compute independent and dependent t-tests using both base *R* and *rstatix* and compare results.

R offers excellent tools for statistical analysis using t-tests using either the base package or add-on packages. Base *R* is best used with the *effectsize* package for effect sizes, while *rstatix* offers a convenient and comprehensive workflow. Users are encouraged to explore and practice using multiple options in order to improve familiarity with the *R* environment.

References

Arnholt, A., & Evans, B. (2023). *BSDA: Basic Statistics and Data Analysis.* doi:10.32614/CRAN.package.BSDA

Ben-Shachar M., Lüdecke D., Makowski, D. (2020). effectsize: Estimation of Effect Size Indices and Standardized Parameters. *Journal of Open Source Software, 5*(56), 2815. doi: 10.21105/joss.02815

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers

Field, A. (2013). *Discovering statistics using IBM SPSS statistics* (4th ed.). Sage Publications.

Kassambara, A. (2025). *rstatix: Pipe-Friendly framework for basic statistical tests.* R package version 0.7.2.

Lovelace, R., Nowosad, J., & Muenchow, J. (2019). *Geocomputation with R*. CRC Press.

Patil, I. (2021). Visualizations with statistical details: The '*ggstatsplot*' approach. *Journal of*

Open Source Software, 6(61), 3167, doi:10.21105/joss.03167

Posit (2025). *RStudio Desktop*. Posit. https://posit.co/download/rstudio-desktop/

R Core Team (2025). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Signorell, A. (2025). *DescTools: Tools for Descriptive Statistics* (R package version 0.99.60) [Computer software]. doi:10.32614/CRAN.package.DescTools https://doi.org/10.32614/CRAN.package.DescTools

University of Toronto Libraries (2024). *Introduction to GIS using R | Map and Data Library*. https://mdl.library.utoronto.ca/technology/tutorials/introduction-gis-using-r

Citation

Lindsay, K. G. (2026). Comparing two means with z-tests and t-tests. In *Applied Statistics 101 using R: A primer for an introductory class in statistical programming* (pp. 51-65). ISTES.

Chapter 5 - One-way Analysis of Variance I: Independent ANOVA

Chapter Highlights

- Compute and compare one-way independent ANOVA using base *R* and add-on packages such as *rstatix*.
- Check assumptions and apply non-parametric and robust alternatives.
- Learn and compute options for post hoc comparisons (e.g., Tukey, Bonferroni, Games-Howell, etc.).
- Obtain and interpret measures of effect size (eta squared, omega squared).
- Compare options for visualizing group differences using boxplots and violin plots.
- Learn APA-style reporting of results.

Introduction

One-way analysis of variance (ANOVA) is a statistical method that compares two or more means. It consists of an omnibus test, followed by pairwise comparisons of ≥ 3 means via a suite of post hoc tests (Kim, 2015). Effect sizes for ANOVA include η^2. The ANOVA procedure was developed by Ronald Fisher for the agricultural sciences. Assumptions of independent ANOVA include continuous outcome variables, a normal distribution of outcome scores for each level of the independent variable, independence of observations, and equality of variances between groups. The latter assumption is evaluated by Levene's test. Thresholds for small medium and large effects are .01, .06, and .14 respectively for partial eta squared (η^2_p; Cohen, 1988). The less biased omega squared (ω^2) generally uses the same thresholds. Though these thresholds may not apply to all contexts, they will be applied in this tutorial.

Post Hoc (Pairwise Comparison) Testing

ANOVA analyses are usually accompanied by an ANOVA table which is an omnibus test. For a comparison of three or more means, it means that at least one mean differs from at least one other mean. Various procedures exist in order to evaluate where differences exist, Most of these methods control for the inflation of Type I error for the comparison of multiple means. They generally fall into one of three categories: single step, step-up, and step-down. A single-step procedure compares all means simultaneously. A step down procedure starts with the pairwise p-value that is most significant, and sequentially compares means until all null hypotheses are rejected, or until one of them is not rejected. When the latter happens, the remaining null hypotheses are also retained. Conversely, a step up procedure moves in the other direction. It starts with the least significant and moves to the most significant one. When a pairwise null hypothesis is rejected, the rest are automatically rejected. The more conservative methods are less likely to reject a null hypothesis for pairwise comparisons, while more powerful ones are more likely to do so. Note that many procedures are modifications of other procedures. For example, the Sidak and Holm procedures are both modifications of the Bonferroni procedure.

Some of these methods are briefly described in the table below, though it is not comprehensive. Emphasis is placed on several pairwise comparisons found in the packages described in this tutorial. Further information may be obtained elsewhere (Holm, 1979; Hochberg, 1988; Hommel, 1988; Benjamini & Hochberg, 1995; Kim, 2015; Liguori & Moreira, 2018; Agbangba et al. 2024; R Core Team, 2025a).

Table 1. List of ANOVA Pairwise Comparisons/post Hoc Tests

Pairwise comparison	Description of method
Bonferroni	Single step. Strong control of Type I error/family-wise error rate (FWER). However, strong control may lead to Type II error.
Dunnett	Single step. Designed to compare treatment groups to a control group.
Benjamini-Hochberg/False Discovery Rate (BH/FDR)	Controls false discovery rate (FDR) rather than FWER; more powerful than Bonferroni, Holm.
Benjamini-Yekutieli (BY)	Same as above.
Hochberg	Step-up. Similar to Tukey, but better for unbalanced designs. More powerful than Bonferroni.
Holm	More powerful, step-down modification of Bonferroni procedure, while controlling FWER.
Hommel	Step up. Generally more powerful than Holm.
Sidak	Single step, though a step down version also exists. Stringent Type I error control, but more powerful than Bonferroni.
Scheffe	Single step. Flexible, with lower power compared to Tukey for pairwise comparisons.
Tukey HSD (Honestly Significant Difference)	Single step. Widely used method across several fields; use for balanced designs.
Games-Howell	Single step. Use when the assumption for equal variances is violated.
Dunn	Single step. Non-parametric test; use with non-parametric ANOVA.
Fisher's Least Significant Differences	Single step. Does not control for inflation of Type I error; generally not recommended for post hoc testing.

This tutorial's objectives are to teach the reader how to:

- Use and interpret one-way ANOVA using *R* and add-on packages.
- Check assumptions and apply nonparametric or robust alternatives for violated assumptions.
- Compute and interpret post hoc test analyses and effect sizes (η^2).
- Visualize group differences.

Introduction to *R*

R (R Core Team, 2025) is a free and open-source statistical programming language. It is supported and used by a worldwide community of scientists, many of whom submit add-on packages for virtually any statistical task, including data science (Zumel el al., 2020). This combination of zero cost and flexibility makes *R* a powerful resource for researchers. However, it involves coding which may compound statistical anxiety for classes with new learners of statistics in R (Forrester et al., 2022). Use of *R* with an Integrated Development Environment (IDE), such as *RStudio* (Posit, 2025) is recommended. While it is not meant to be a "point-and-click" interface like those found in other statistical programs, it is helpful for the download and management of datasets, objects, and packages.

Objects

An object is an umbrella term for a method of abbreviating an entity that may be found or created in *R,* to better use or manage it. A dataset may be assigned as an object. The object may then be manipulated, depending on the user's needs. For example, a subset of data may be created using another object. The object may also be summarized and analyzed, the results of which may also be assigned as objects. Objects are created using the abbreviation chosen by the user followed by "<-" or "=". Examples found in this tutorial include the following object to abbreviate an ANOVA analysis using the base package: "bsa_anova_base_R<-aov(fps~im, data = bsa)". This stores the model for future reference. The object is then used to obtain Tukey Honest Significant Differences post hoc tests using the syntax "TukeyHSD(bsa_anova_base_R)". An object may then be used to store this in a project in *RStudio*. Readers are encouraged to create and organize objects at their own discretion.

Packages

There are several *R* packages that compute one-way ANOVA. Packages used for standard ANOVA include R's base package, *afex* (Singmann et al., 2025), and *car* (Fox & Weisberg, 2019). In case assumptions are violated, the base package*,* and *WRS2* (Mair & Wilcox, 2019) offer nonparametric or robust versions of ANOVA. Options for post hoc tests can also be obtained via the base package, or the *emmeans* package. Finally, *afex*, *DescTools* (Signorell, 2025), and *effectsize* (Ben-Shachar et al., 2020) all offer options for obtaining effect sizes. This paper emphasizes the use of *rstatix* (Kassambara, 2025), as it offers all of the above options. However, syntax for the use of other packages will also be shown.

Datasets and Premises

There are two datasets used in these analyses. They are synthesized for learning purposes and contain no real student data. For the first dataset, a statistics professor wants to compare final project scores between a lecture class, an online class, and a hybrid class. The second dataset violates the assumption of homogenous variances and compares four instructional methods on student performance. These data may be obtained using this link:
https://osf.io/mrw5v/overview.

Method

Parametric One-way Independent ANOVA

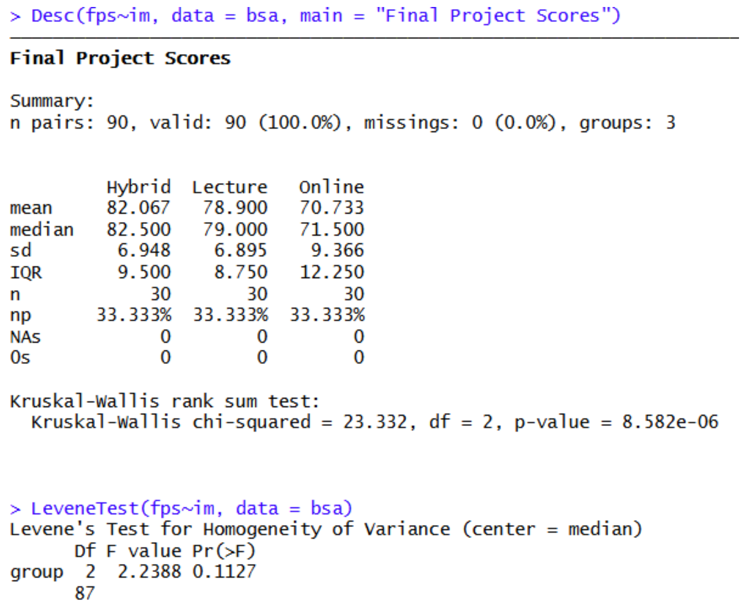
Users may load the dataset into *RStudio*, and follow the steps below to analyze data. Workflows using the base *R* and *rstatix* packages are emphasized.

Table 2. Parametric ANOVA Syntax

Package	Syntax	Purpose
	library(rstatix) library(effectsize) library(DescTools)	
Base *R*	bsa_anova_base_R<-aov(fps~im, data = bsa)	1-way ANOVA analysis

Package	Syntax	Purpose
	summary(bsa_anova_base_R)	ANOVA table
	TukeyHSD(bsa_anova_base_R)	Tukey post hoc test
	plot(TukeyHSD(bsa_anova_base_R))	Optional, for basic visualization.
effectsize	eta_squared(bsa_anova_base_R)	Partial eta squared. Use with base *R*.
rstatix	bsa_desc<-bsa %>% group_by(im) %>% get_summary_stats(fps)	Descriptive statistics. The *rstatix* package is well-rounded for ANOVA analysis.
	levene_test(bsa, fps~im)	Shows Levene's test.
	bsa_anova_rstatix<-anova_test(data = bsa, dv = fps, between = im, effect.size = "pes")	1-way ANOVA
	bsa_anova_rstatix	ANOVA table.
	bsa_post_hoc_rstatix<-bsa %>% pairwise_t_test(fps~im, p.adjust.method = "bonferroni")	Bonferroni post hoc test.
	bsa %>% emmeans_test(fps~im, p.adjust.method = "sidak")	Alternate syntax for post hoc analysis using *rstatix.*
DescTools	Desc(fps~im, data = bsa, main = "Final Project Scores")	Good for descriptives and boxplots. Use in tandem with base *R*.
	EtaSq(bsa_anova_base_R)	Partial eta squared. Use with base *R*.
ggstatsplot	bsaviolin<-ggbetweenstats(data = bsa, x = im, y = fps) + labs(x = "Instructional Method", y = "Final Project Score", title = "Comparison of Methods")	Customized visualization.

The *DescTools* package is a comprehensive option for descriptive statistics and visualization (boxplots) when running ANOVA in the base *R* package. It also offers Levene's test, and eta squared effect sizes. In contrast, getting similar descriptive statistics and visualization with the Base *R* package requires lengthier syntax. In contrast, the command: "Desc(fps~im, data = bsa, main = "Final Project Scores")", offers descriptive statistics, boxplots by group, and even an omnibus Kruskall-Wallis test. Note that acquiring Levene's test requires separate syntax, and is shown here for demonstrative purposes only. Interpret Levene's test cautiously, as it may be significant due to large sample sizes (Field, 2013). It has even been recommended to ignore Levene's test if group sizes are equal (Field, 2013). Cautious approaches include checking consistency of parametric ANOVA tables with those of Welch's ANOVA and checking effect sizes.



```
> Desc(fps~im, data = bsa, main = "Final Project Scores")
------------------------------------------------------------------
Final Project Scores

Summary:
n pairs: 90, valid: 90 (100.0%), missings: 0 (0.0%), groups: 3


         Hybrid  Lecture   Online
mean     82.067   78.900   70.733
median   82.500   79.000   71.500
sd        6.948    6.895    9.366
IQR       9.500    8.750   12.250
n            30       30       30
np      33.333%  33.333%  33.333%
NAs           0        0        0
0s            0        0        0

Kruskal-Wallis rank sum test:
  Kruskal-Wallis chi-squared = 23.332, df = 2, p-value = 8.582e-06


> LeveneTest(fps~im, data = bsa)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  2  2.2388 0.1127
      87
```

Figure 1. Descriptive Statistics and the Levene's Test using *DescTools*

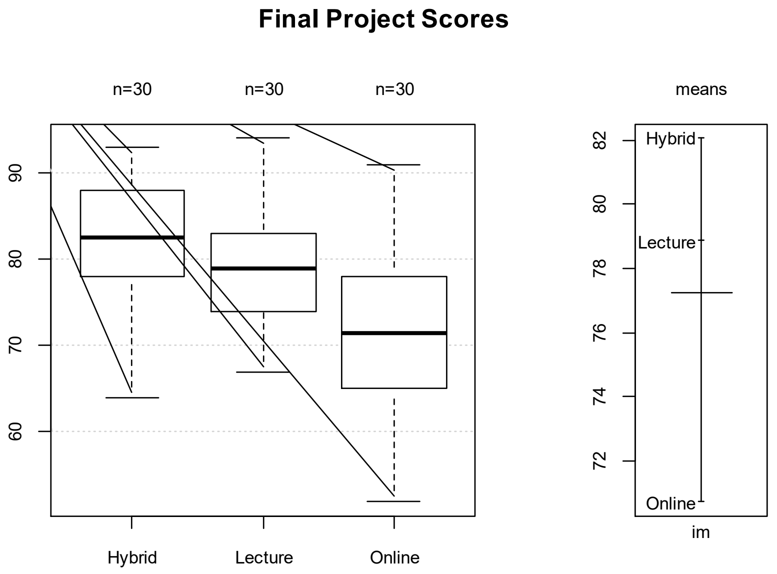
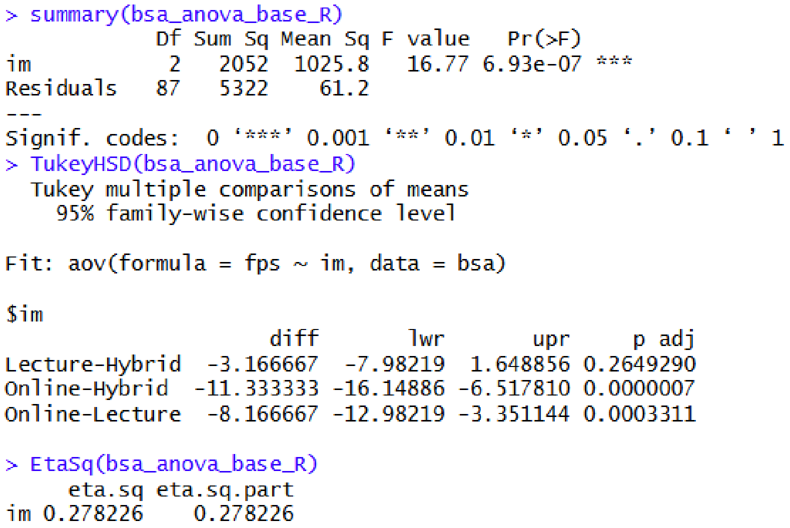


Figure 2. Boxplots by Group using *DescTools*

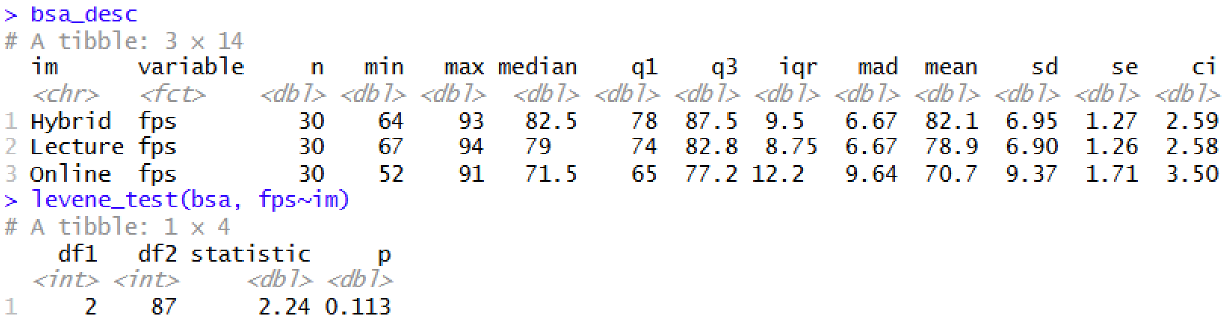
The data are analyzed below using ANOVA and the Tukey post hoc test in base *R,* as shown below. For this demonstration, the *aov* command was used. The effect size is obtained using the *EtaSq* command in *DescTools*.



```
> summary(bsa_anova_base_R)
            Df Sum Sq Mean Sq F value   Pr(>F)    
im           2   2052  1025.8   16.77 6.93e-07 ***
Residuals   87   5322    61.2                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> TukeyHSD(bsa_anova_base_R)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = fps ~ im, data = bsa)

$im
                     diff       lwr       upr     p adj
Lecture-Hybrid  -3.166667  -7.98219  1.648856 0.2649290
Online-Hybrid  -11.333333 -16.14886 -6.517810 0.0000007
Online-Lecture  -8.166667 -12.98219 -3.351144 0.0003311

> EtaSq(bsa_anova_base_R)
     eta.sq eta.sq.part
im 0.278226    0.278226
```

Figure 3. ANOVA Analysis using Base *R* and *DescTools*

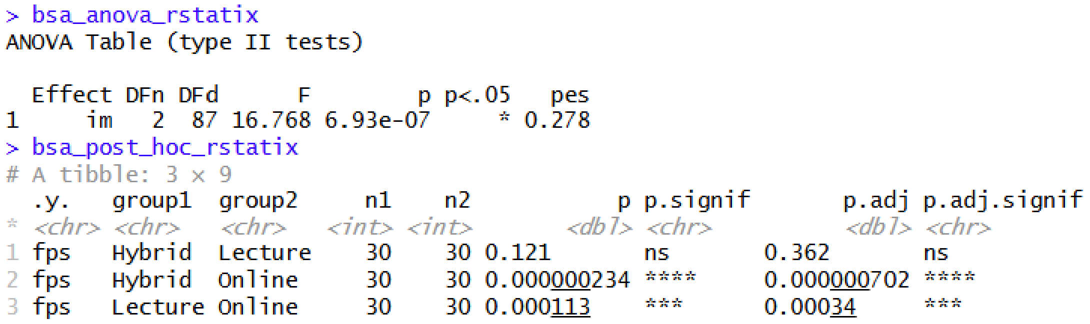
The same analysis is replicated below using the *rstatix* package. Descriptive statistics, Levene’s test, the ANOVA analysis, and the Bonferroni post hoc test are shown below.



```
> bsa_desc
# A tibble: 3 x 14
  im      variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se    ci
  <chr>   <fct>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Hybrid  fps         30    64    93   82.5    78  87.5  9.5   6.67  82.1  6.95  1.27  2.59
2 Lecture fps         30    67    94   79      74  82.8  8.75  6.67  78.9  6.90  1.26  2.58
3 Online  fps         30    52    91   71.5    65  77.2 12.2   9.64  70.7  9.37  1.71  3.50
> levene_test(bsa, fps~im)
# A tibble: 1 x 4
    df1   df2 statistic     p
  <int> <int>     <dbl> <dbl>
1     2    87      2.24 0.113
```

Figure 4. Descriptive Statistics and the Levene’s Test using *rstatix*

Note the added effect size in the ANOVA table and compare the post hoc tables for both packages. Report as follows: $F(2, 87) = 16.8$, $p < .001$, (partial) $\eta^2 = .28$. The ANOVA result suggests a difference between at least two groups. The post hoc test tells where the differences occur. Hybrid and lecture both differed from the online group, but not from each other. The means show that the final project score for the hybrid and lecture sections were higher than the online section. Per Cohen’s benchmarks (1988), the size of the effect is large. This interpretation aligns with practical significance, as the hybrid and lecture sections are higher than the online section by approximately a letter grade.



```
> bsa_anova_rstatix
ANOVA Table (type II tests)

  Effect DFn DFd      F        p p<.05   pes
1     im   2  87 16.768 6.93e-07     * 0.278
> bsa_post_hoc_rstatix
# A tibble: 3 x 9
  .y.   group1  group2     n1    n2           p p.signif       p.adj p.adj.signif
* <chr> <chr>   <chr>   <int> <int>       <dbl> <chr>          <dbl> <chr>
1 fps   Hybrid  Lecture    30    30 0.121       ns       0.362       ns
2 fps   Hybrid  Online     30    30 0.000000234 ****     0.000000702 ****
3 fps   Lecture Online     30    30 0.000113    ***      0.00034     ***
```

Figure 5. ANOVA Table and Bonferroni Post Hoc Tests using *rstatix*

Nonparametric One-way Independent ANOVA

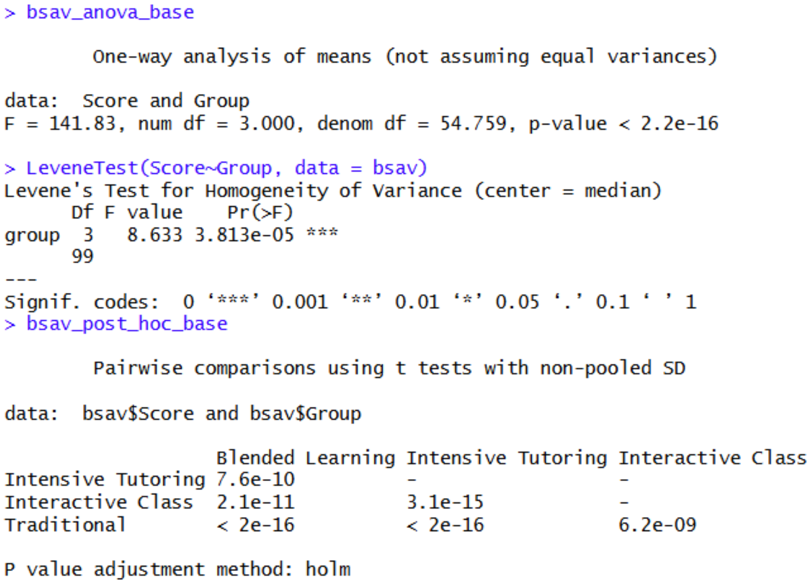
The base package, *DescTools*, and *rstatix* may also be used to compute ANOVA for violated assumptions; Welch's ANOVA is used for both packages. Welch's ANOVA is used when the assumption of equal variances is violated. It provides a more robust option by adjusting degrees of freedom. However, the commands for both packages differ slightly. The syntaxes for nonparametric ANOVA are shown in Table 3.

Table 3. Nonparametric ANOVA Syntax

Package	Syntax	Purpose
Base *R*	library(ggstatsplot)	
	bsav_anova_base<- oneway.test(Score~Group, data = bsav)	Command when equal variances are unassumed. Note the contrast with the "aov" command.
	bsav_anova_base	ANOVA table
	bsav_post_hoc_base<- pairwise.t.test(bsav$Score, bsav$Group, p.adjust.method = "holm", pool.sd = FALSE)	Welch's t-tests with Holm adjustments for multiple comparisons.
DescTools	Desc(Score ~ Group, data = bsav, main = "Scores by Group")	Descriptives, boxplots, abridged nonparametric analysis.
	LeveneTest(Score~Group, data = bsav)	Levene's test
effectsize	eta_squared(bsav_anova_base) omega_squared(bsav_anova_base)	Respective commands for eta squared and omega squared.

Package	Syntax	Purpose
rstatix	bsav_desc<-bsav %>% group_by(Group) %>% get_summary_stats(Score)	Descriptive statistics.
	levene_test(bsav, Score~Group)	Violated test shows nonparametric option needed.
	bsav$Group<-as.factor(bsav$Group)	
	bsav_anova_rstatix<-welch_anova_test(data = bsav, Score~Group)	1-way ANOVA.
	bsav_anova_rstatix	ANOVA table.
	bsav_gh_test<-games_howell_test(bsav, Score~Group)	Games-Howell post hoc test.
ggstatsplot	bsav_plot1<-ggbetweenstats(data = bsav, x = Group, y = Score, type = "welch", title = “Scores by Group”)	Comprehensive option for visualization.

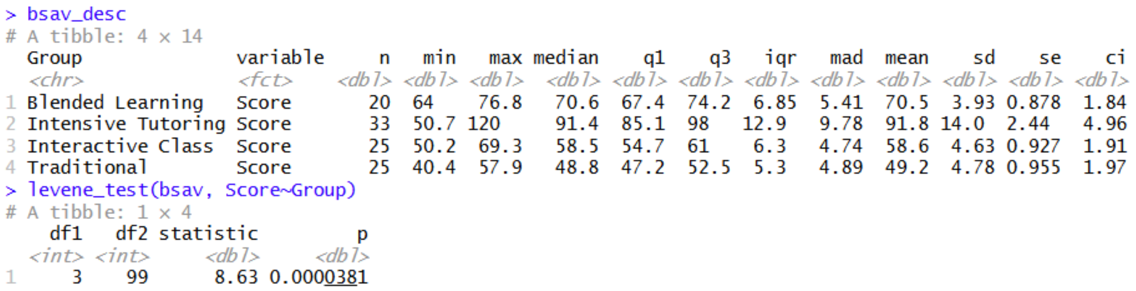
For this research question, a statistics professor wants to know if four teaching modalities differ in the outcome of final examination scores. Nonparametric ANOVA is computed in the base *R* package using the *oneway.test* command, as opposed to the *aov* command for parametric analysis. The resulting Welch’s ANOVA table is produced below. The post hoc tests used here are pairwise Welch’s t-tests with Holm adjustments. The Levene’s test is obtained using *DescTools*, while eta squared is obtained using the *effectsize* package.



```
> bsav_anova_base

        One-way analysis of means (not assuming equal variances)

data:  Score and Group
F = 141.83, num df = 3.000, denom df = 54.759, p-value < 2.2e-16

> LeveneTest(Score~Group, data = bsav)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value    Pr(>F)    
group  3   8.633 3.813e-05 ***
      99                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> bsav_post_hoc_base

        Pairwise comparisons using t tests with non-pooled SD 

data:  bsav$Score and bsav$Group 

                   Blended Learning Intensive Tutoring Interactive Class
Intensive Tutoring 7.6e-10          -                  -                
Interactive Class  2.1e-11          3.1e-15            -                
Traditional        < 2e-16          < 2e-16            6.2e-09          

P value adjustment method: holm 
```

Figure 6: Welch’s ANOVA using base *R* and *DescTools*

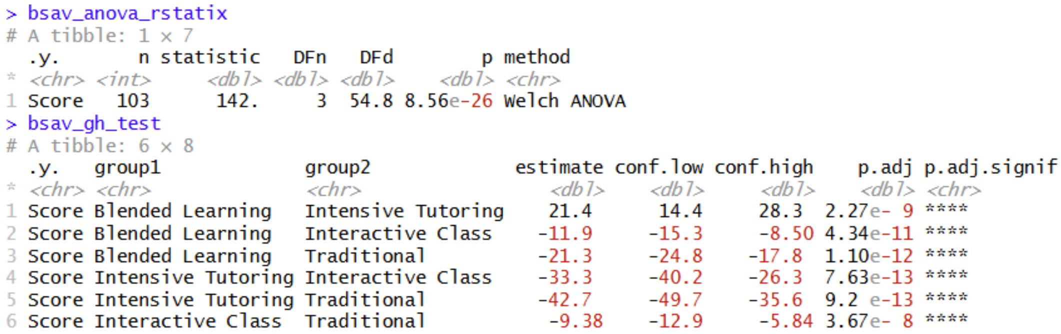
The analysis is replicated using the *rstatix* package as shown below. Like the base package, it is also capable of using Welch's ANOVA. Note the difference in syntax between the parametric and nonparametric analyses using this package.



```
> bsav_desc
# A tibble: 4 × 14
  Group              variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se    ci
  <chr>              <fct>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Blended Learning   Score       20  64    76.8   70.6  67.4  74.2  6.85  5.41  70.5  3.93 0.878  1.84
2 Intensive Tutoring Score       33  50.7 120     91.4  85.1  98    12.9   9.78  91.8 14.0  2.44   4.96
3 Interactive Class  Score       25  50.2  69.3   58.5  54.7  61     6.3   4.74  58.6  4.63 0.927  1.91
4 Traditional        Score       25  40.4  57.9   48.8  47.2  52.5   5.3   4.89  49.2  4.78 0.955  1.97
> levene_test(bsav, Score~Group)
# A tibble: 1 × 4
    df1   df2 statistic         p
  <int> <int>     <dbl>     <dbl>
1     3    99      8.63 0.0000381
```

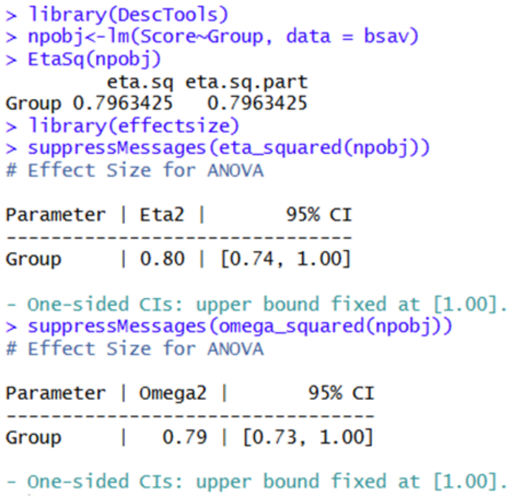
Figure 7. Descriptive Statistics and the Levene's Test using *rstatix*

Another notable difference with getting Welch's ANOVA using *rstatix* is that the effect size is not reported. It may be obtained using the *effectsize* or *DescTools* packages. The post hoc test chosen this time is the Games-Howell test.



```
> bsav_anova_rstatix
# A tibble: 1 × 7
  .y.       n statistic   DFn   DFd         p method
* <chr> <int>     <dbl> <dbl> <dbl>     <dbl> <chr>
1 Score   103      142.     3  54.8 8.56e-26 Welch ANOVA
> bsav_gh_test
# A tibble: 6 × 8
  .y.   group1             group2             estimate conf.low conf.high    p.adj p.adj.signif
* <chr> <chr>              <chr>                 <dbl>    <dbl>     <dbl>    <dbl> <chr>
1 Score Blended Learning   Intensive Tutoring    21.4      14.4      28.3  2.27e- 9 ****
2 Score Blended Learning   Interactive Class    -11.9     -15.3      -8.50 4.34e-11 ****
3 Score Blended Learning   Traditional          -21.3     -24.8     -17.8  1.10e-12 ****
4 Score Intensive Tutoring Interactive Class    -33.3     -40.2     -26.3  7.63e-13 ****
5 Score Intensive Tutoring Traditional          -42.7     -49.7     -35.6  9.2 e-13 ****
6 Score Interactive Class  Traditional           -9.38    -12.9      -5.84 3.67e- 8 ****
```

Figure 8. Welch's ANOVA and Games-Howell Post Hoc Tests in *rstatix*



```
> library(DescTools)
> npobj<-lm(Score~Group, data = bsav)
> EtaSq(npobj)
         eta.sq eta.sq.part
Group 0.7963425   0.7963425
> library(effectsize)
> suppressMessages(eta_squared(npobj))
# Effect Size for ANOVA

Parameter | Eta2 |       95% CI
-------------------------------
Group     | 0.80 | [0.74, 1.00]

- One-sided CIs: upper bound fixed at [1.00].
> suppressMessages(omega_squared(npobj))
# Effect Size for ANOVA

Parameter | Omega2 |       95% CI
---------------------------------
Group     |   0.79 | [0.73, 1.00]

- One-sided CIs: upper bound fixed at [1.00].
```

Figure 9. Effect Size Calculations using *DescTools* and *effectsize*

The effect size may be obtained by first creating a *lm* object using the base *R* package, then using either the *EtaSq* or *eta_squared* commands from *DescTools* or *effectsize* respectively. The results of the effect size measure ω^2 is also shown, as it is less biased than η^2 (Ellis, 2017).

Visualization

In addition to *DescTools*, *ggstatsplot* (Patil, 2021) is a comprehensive option for visualizing group differences. In addition to violin plots, it even offers an abridged ANOVA analysis and post hoc tests. The ANOVA analysis should be reported as follows: $F(3, 54.8) = 142$, p <.001, $\eta^2 = .80$. All modalities differed from one another (see post hoc tests), with intensive tutoring being the best modality, followed by blended learning, the interactive class, with the traditional method having the lowest mean score. The effect of modality on score was large.

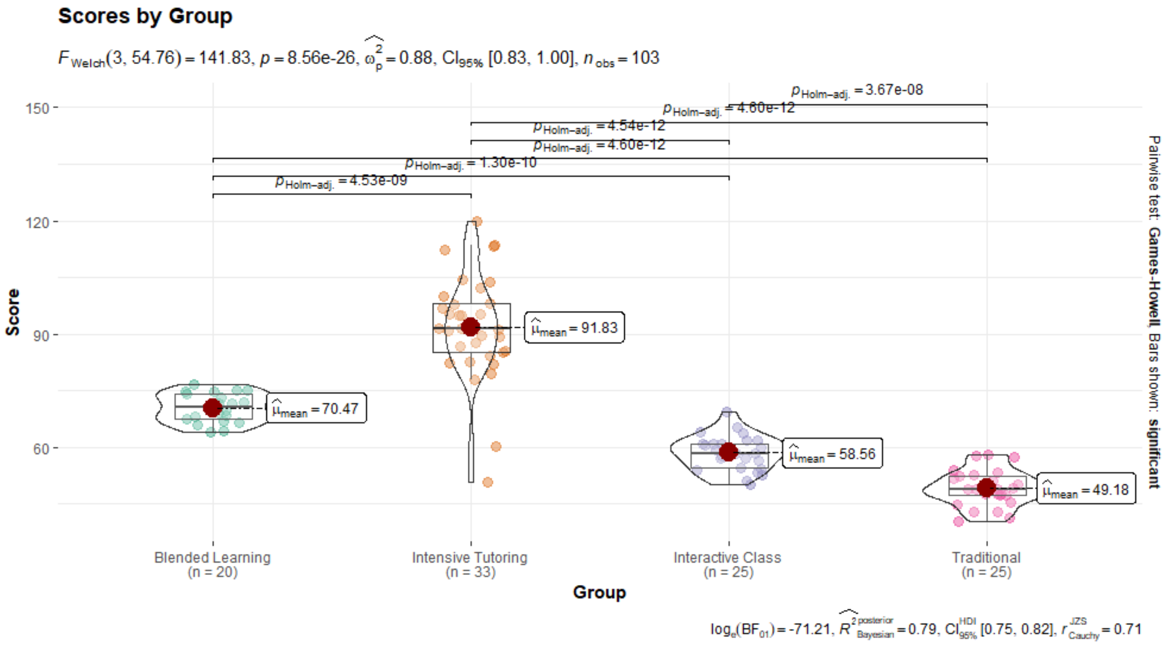


Figure 9. Visualization by Group using *ggstatsplot*

Other ANOVA Packages

In addition to the base package and *rstatix*, several other packages also compute ANOVA analyses. These include *afex* and *car* for parametric ANOVA, and *WRS2* for robust ANOVA. The syntaxes for these packages are shown in the table below.

Table 4. Syntaxes for Other Packages

Package	Syntax	Purpose
	library(afex) library(car) library(WRS2) library(emmeans)	
afex	bsa$im<-as.factor(bsa$im)	Syntax to transform independent variable into factor for *afex* analysis.
	bsa_anova_afex<-aov_ez(id="ID", dv = "fps", between = "im", data = bsa, anova_table = list(es = "pes"))	Parametric ANOVA analysis.
	bsa_anova_afex	ANOVA table.
	bsa_afexplot<-afex_plot(bsa_anova_afex, x="im", error = "model", dodge = 0.4)	Optional visualization using *afex*.
emmeans	emmeans(bsa_anova_afex, pairwise~im)	Tukey post hoc tests.
car	car_model<-lm(fps~im, data = bsa)	Creates object for ANOVA in *car*
	car_anova<-Anova(car_model)	ANOVA summary
multcomp	car_post_hoc<-summary(glht(car_model, linfct = mcp(im = "Tukey")))	Tukey post hoc test.
effectsize	eta_squared(bsa_anova_afex, partial = TRUE)	Eta squared effect size. "partial" command more pertinent to factorial ANOVA (*afex* object).
	omega_squared(bsa_anova_afex)	Omega squared effect size (*afex* object)
	eta_squared(car_model)	Eta squared for *car* object.
WRS2	bsav_anova_wrs2<-t1way(Score ~ Group, data = bsav)	Robust ANOVA using 20% trimmed means.

Package	Syntax	Purpose
	wrs2_effect<-yuen.effect.ci(Score ~ Group, data = bsav)	Yuen's effect size.
	bsav_post_hoc_wrs2<-mcppb20(Score ~ Group, data = bsav)	Robust post hoc analyses also using trimmed means (package uses Hochberg's approach).

Discussion and Conclusion

R offers several options for ANOVA analysis. In this tutorial, the *rstatix* package was used, as it offers a comprehensive suite of options for parametric and nonparametric ANOVA analysis, to include descriptive statistics, the Levene's test, effect sizes, and post hoc tests. Similarly, the base package offers options for parametric ANOVA, nonparametric ANOVA, and respective post hoc tests. While descriptive statistics are offered in the base package, they are more cumbersome to obtain.

Additionally, Levene's test, and measures of effect size are not offered as functions in the base package. The *DescTools* package is a good option for descriptive statistics, the Levene's Test, and effect sizes. The *effectsize* package is also a good option for obtaining effect sizes. Finally, *DescTools*, and *ggstatsplot* are options for visualization of group comparisons. Syntaxes for other packages such as *car* and *WRS2* (Mair & Wilcox, 2019) are also shown in Table 3.

R offers several other options for parametric and nonparametric one-way independent ANOVA, and not all packages are shown in this paper. This is a similar case for visualization. Additionally, not all features of *R*'s base package, *rstatix*, or other add-on packages were shown. This was intentional, to find a balance between comprehensiveness and accessibility. The reader is encouraged to explore other packages at their own discretion.

The *R* environment offers several packages for ANOVA analysis, post hoc tests, and visualization. The reader is encouraged to explore their preferences, and to practice using the packages that best suit their needs.

References

Agbangba, C. E., Sacla Aide, E., Honfo, H., & Glèlè Kakai, R. (2024). On the use of post-hoc tests in environmental and biological sciences: A critical review. *Heliyon*, *10*(3), e25131. https://doi.org/10.1016/j.heliyon.2024.e25131

Ben-Shachar, M., Lüdecke, D., Makowski, D. (2020). *effectsize*: Estimation of Effect Size Indices and Standardized Parameters. *Journal of Open Source Software, 5*(56), 2815. doi: 10.21105/joss.02815

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(1), 289–300.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Ellis, P. D. (2017). *The essential guide to effect sizes: statistical power, meta-analysis and the interpretation of research results*. Cambridge University Press.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics* (4th ed.). Sage Publications.

Forrester, C., Schwikert, S., Foster, J., & Corwin, L. (2022). Undergraduate R programming anxiety in ecology: persistent gender gaps and coping strategies. *CBE—Life Sciences Education*, *21*(2). https://doi.org/10.1187/cbe.21-05-0133

Fox, J., & Weisberg, S. (2019). *An R Companion to Applied Regression*, Third edition. Sage, Thousand Oaks CA. https://www.john-fox.ca/Companion/

Hochberg, Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, 75, 800–803. doi:10.2307/2336325.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6, 65–70. https://www.jstor.org/stable/4615733.

Hommel, G. (1988). A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika*, 75, 383–386. doi:10.2307/2336190.

Kassambara, A. (2025). *rstatix: Pipe-Friendly framework for basic statistical tests. R* package version 0.7.2.

Kim, H.-Y. (2015). Statistical notes for clinical researchers: post-hoc multiple comparisons. *Restorative Dentistry & Endodontics*, *40*(2), 172. https://doi.org/10.5395/rde.2015.40.2.172

Liguori, G. R., & Moreira, L. F. P. (2018). Operating with Data - Statistics for the

Cardiovascular Surgeon: Part III. Comparing Groups. *Brazilian Journal of Cardiovascular Surgery*, *33*(6). https://doi.org/10.21470/1678-9741-2018-0378

Mair, P., & Wilcox, R. (2019). Robust statistical methods in *R* using the WRS2 package. *Behavior Research Methods*, *52*. https://doi.org/10.3758/s13428-019-01246-w

Patil, I. (2021). Visualizations with statistical details: The '*ggstatsplot*' approach. *Journal of Open Source Software, 6*(61), 3167, doi:10.21105/joss.03167

Posit (2025). *RStudio Desktop*. Posit. https://posit.co/download/rstudio-desktop/

R Core Team. (2025a). *p.adjust: Adjust p-values for multiple comparisons* (Version 4.5.1) [Computer software]. R Foundation for Statistical Computing. https://www.R-project.org/

R Core Team (2025b). *R: A language and environment for statistical computing. R* Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Signorell, A. (2025). *DescTools: Tools for Descriptive Statistics* (R package version 0.99.60) [Computer software]. doi:10.32614/CRAN.package.DescTools https://doi.org/10.32614/CRAN.package.DescTools

Singmann H., Bolker B., Westfall J., Aust F., Ben-Shachar, M. (2025). *afex: Analysis of Factorial Experiments. R* package version 1.5-0, https://github.com/singmann/afex

Zumel, N., Mount, J., Howard, J., & Thomas, R. (2020). *Practical data science with R*. Manning Publications Company.

Citation

Lindsay, K. G. (2026). One-way analysis of variance I: Independent ANOVA. In *Applied Statistics 101 using R: A primer for an introductory class in statistical programming* (pp. 66-81). ISTES.

Chapter 6 - One-way Analysis of Variance II: Repeated Measures ANOVA

Chapter Highlights

- Compute repeated measures ANOVA using *rstatix*, *afex*, and *emmeans*.
- Check sphericity and apply Greenhouse-Geisser and Huynh-Feldt corrections.
- Compute post hoc tests using multiple adjustment methods (Holm, Hommel, Hochberg, etc.).
- Obtain and interpret effect sizes (generalized and partial eta squared).
- Visualization of repeated measures data
- Explore non-parametric (Friedman's ANOVA, aligned rank transformation) and robust options for repeated measures ANOVA using *rstatix*, *ARTool*, and *WRS2* respectively.
- Learn APA-style reporting of results.

Introduction: Repeated Measures ANOVA

Repeated measures ANOVA (RMA), also called dependent ANOVA, is a method for comparing two or more means of related groups, or (more commonly) the same group measured at two or more times, contexts, or conditions. Like independent ANOVA it may be used for two or more means and also uses the assumption of normality of outcomes at each level if the independent variable. Sphericity is defined as the equality of variances of the differences between each pair of treatment levels, and is also an assumption for the use of parametric RMA (Field, 2013). The assumption of sphericity therefore requires at least three levels of the independent variable. Sphericity is assessed by Mauchly's test of sphericity, and this is standard output for software for computing RMA. If it is non-significant, sphericity may be assumed. If significant, corrections exist. These include the Greenhouse-Geisser and Huynh-Feldt corrections (Field, 2013). For violations of normality, several methods exist such as Friedman's ANOVA. Whereas the assumption of independent observations exists for independent ANOVA, RMA assumes related observations (i.e., relatedness). Hence, this method is typically used in studies where participants are measured at two or more timepoints, contexts, or conditions. It may also be potentially used to study related samples (e.g., comparing the same traits of several generations of plants or bacteria). Repeated measures ANOVA also requires complete data for all timepoints. Any case with incomplete data will be excluded from analysis, and standard repeated measures ANOVA may not be used. For studies requiring repeated measurements, where the violations of related observations and complete data occur, methods such as multilevel models may be used. Finally, for analyses with more than two levels, post hoc comparisons may be used. A brief discussion of post hoc comparisons and their descriptions may be found in the previous chapter. For the reader's convenience, that table is replicated below.

Table 1. List of ANOVA Pairwise Comparisons/post Hoc Tests

Pairwise comparison	Description of method
Bonferroni	Single step. Strong control of Type I error/family-wise error rate (FWER). However, strong control may lead to Type II error.
Dunnett	Single step. Designed to compare treatment groups to a control group.

Pairwise comparison	Description of method
Benjamini-Hochberg/False Discovery Rate (BH/FDR)	Controls false discovery rate (FDR) rather than FWER; more powerful than Bonferroni, Holm.
Benjamini-Yekutieli (BY)	Same as above.
Hochberg	Step-up. Similar to Tukey, but better for unbalanced designs. More powerful than Bonferroni.
Holm	More powerful, step-down modification of Bonferroni procedure, while controlling FWER.
Hommel	Step up. Generally more powerful than Holm.
Sidak	Single step, though a step down version also exists. Stringent Type I error control, but more powerful than Bonferroni.
Scheffe	Single step. Flexible, with lower power compared to Tukey for pairwise comparisons.
Tukey HSD (Honestly Significant Difference)	Single step. Widely used method across several fields; use for balanced designs.
Games-Howell	Single step. Use when the assumption for equal variances is violated.
Dunn	Single step. Non-parametric test; use with non-parametric ANOVA.
Fisher's Least Significant Differences	Single step. Does not control for inflation of Type I error; generally not recommended for post hoc testing.

The *R* Environment

R is a free and open-source environment for statistical computing, supported by a worldwide network of scientists and programmers. It contains add-on packages for statistics, data science, and related areas such as geographical information systems. These qualities make *R* a powerful tool for teaching and data analysis. However, *R* requires coding which may intimidate new users, and may compound statistical anxiety in introductory statistics courses using *R*. *R* may be used with an integrated development environment such as *RStudio* (Posit,

2025) to more easily manage data, packages, and objects.

Repeated Measures ANOVA in *R*

There are several packages that implement RMA in *R*. The *rstatix* package (Kassambara, 2025) is a comprehensive option for both parametric and nonparametric RMA. Other packages include *afex* (Singmann, et al., 2025) combined with *datawizard* (Patil, et al, 2022), and *emmeans* (Lenth & Piakowski, 2025) for parametric RMA. The *ARTool* package (Wobbrock at al., 2011) combined with *emmeans* are also a good option for nonparametric analysis. The *WRS2* package also offers a robust option as well as robust post hoc tests. The *tidyr* package (Wickham et al., 2024) is used to demonstrate reformatting of data from long to wide, and vice versa. The *ggstatsplot* package (Patil, 2021) will be used for visualization.

For Type I sums of squares in ANOVA analysis, the order in which predictors are entered matters. This method is not recommended for repeated measures designs, as they do not evaluate main effects or interactions. The base *R* package uses this variant only (Field, Miles & Field, 2012). Conversely, Type II sums of squares takes all main effects into consideration, while ignoring higher order (interaction) effects. Type III sums of squares evaluates the effects (main and interaction), taking all other effects into context, and is also used for unbalanced designs. It is the default used by commercial packages that implement RMA like SPSS, and for *R* packages such as *rstatix* (Field, Miles & Field, 2012; Kassambara, 2025). The second problem with using the base *R* functions is that they do not produce tests of sphericity. Hence repeated measures ANOVA using the base *R* package will be omitted from this tutorial.

The packages presented here use the long data format. This means that the time or context variable has one dedicated column with all of its levels, and each observation's timepoint is in a separate row. Hence, for a study that measures four timepoints, each participant will have four rows of data. An ID number is used so that the software can discern every participant across all levels. This is in contrast to the wide format, where every level of time or condition has its own dedicated column, and therefore each participant has a dedicated row across several columns. This latter format is used for packages such as SPSS (Armonk, NY), and JASP (JASP Team, 2025). Syntaxes for conversion from wide long format and vice versa will be shown.

Objects

An object is a way of summarizing, storing, and managing some entity created, or found in *R*. For example, an object may be used for a dataset. For example, data1<-c(2, 4, 6, 17) assigns the object "data1" to four scores. In the table below, a dataset is converted from long to wide format, and a new object is assigned to the newly formatted data. Users are encouraged to create and manage objects at their discretion.

Data

The premise for this dataset is that a group of students engaged in a four-phase learning program designed to improve their performance in an undergraduate applied statistics course. Student performance was measured at each phase to evaluate the program's effectiveness. The data may be obtained using this link: https://osf.io/mrw5v/overview.

Method

Once the dataset has been uploaded into *RStudio*, the following steps may be taken to reformat data if needed, and for analysis. For parametric analysis, *rstatix* as well as *afex* and *emmeans* will be used. The *rstatix* package, WRS2, as well as *ARTool* and *emmeans* will be used. The syntaxes for converting from long to wide and vice versa using the *tidyr* package are shown below.

Table 2. Syntaxes for Conversion of Formats

Syntax	Purpose
library(tidyr)	Loads the package
pivot_wider(rma, names_from = phase, values_from = score)	Converts data from long format to wide format.
rma_wide<-pivot_wider(rma, names_from = phase, values_from = score)	Assigns a new object to the newly created data.
View(rma_wide)	Checks to make sure dataset was properly converted.
rma_long<-pivot_longer(rma_wide, cols = -	Converts data from wide to long

Syntax	Purpose
ID, names_to = "phase", values_to = "score")	format.

Parametric Repeated Measures ANOVA

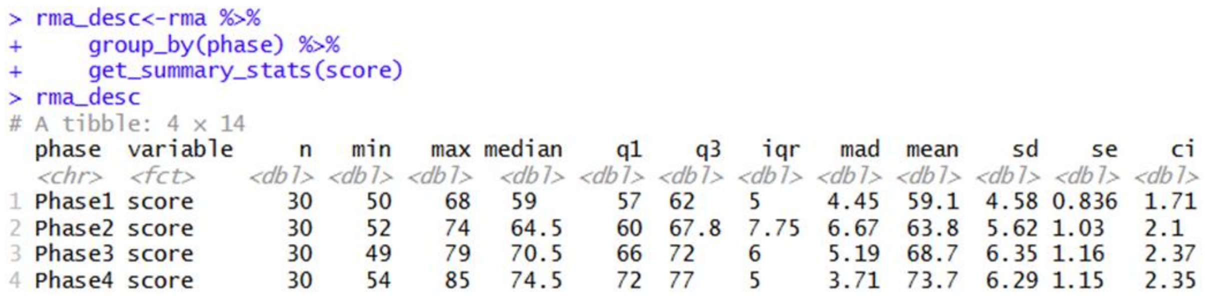
Repeated Measures ANOVA using the rstatix Package

The following syntaxes are used to compute RMA. Both the *rstatix* and *afex* packages require the long format for analysis. The *ggstatsplot* package is used to create a custom plot to visualize the analysis.

Table 3. Parametric Repeated Measures ANOVA syntax using *rstatix*

Package	Syntax	Purpose
	library(rstatix) library(afex) library(emmeans)	
rstatix	rma_desc<-rma %>% group_by(phase) %>% get_summary_stats(score)	Creates object for descriptive statistics table.
	rma_desc	Shows descriptive statistics.
	rm_rstatix<-anova_test(data = rma, dv = score, wid = ID, within = phase)	Repeated measures ANOVA in *rstatix*. General eta squared is the default effect size.
	rm_rstatix0<-anova_test(data = rma, dv = score, wid = ID, within = phase, effect.size = "pes")	Similar to the above. Although the general interpretation is similar, note the difference between general and partial eta squared.
	rm_rstatix_post_hoc<- rma%>% pairwise_t_test(score~phase, paired = TRUE, p.adjust.method = "holm")	Post hoc tests using holm adjustment. Other options include Bonferroni, Hochberg, Hommel, and others.

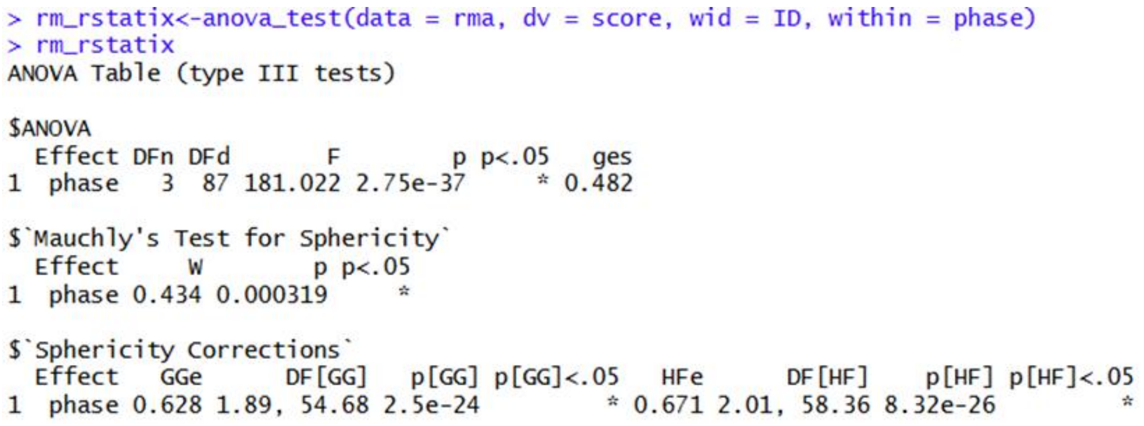
The *rstatix* package is a comprehensive option for computing repeated measures ANOVA as it provides descriptive statistics (Figure 1), ANOVA tables with effect sizes (Figures 2 & 3), and post hoc tests.



```
> rma_desc<-rma %>%
+     group_by(phase) %>%
+     get_summary_stats(score)
> rma_desc
# A tibble: 4 x 14
  phase  variable     n   min   max median    q1    q3   iqr   mad  mean    sd    se    ci
  <chr>  <fct>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Phase1 score       30    50    68   59      57  62    5     4.45  59.1  4.58 0.836  1.71
2 Phase2 score       30    52    74   64.5    60  67.8  7.75  6.67  63.8  5.62 1.03   2.1 
3 Phase3 score       30    49    79   70.5    66  72    6     5.19  68.7  6.35 1.16   2.37
4 Phase4 score       30    54    85   74.5    72  77    5     3.71  73.7  6.29 1.15   2.35
```

Figure 1. Descriptive Statistics

The default effect size for *rstatix* is general eta squared as shown below. Note the difference from partial eta squared, which tends to be more upward biased (Ellis, 2017).

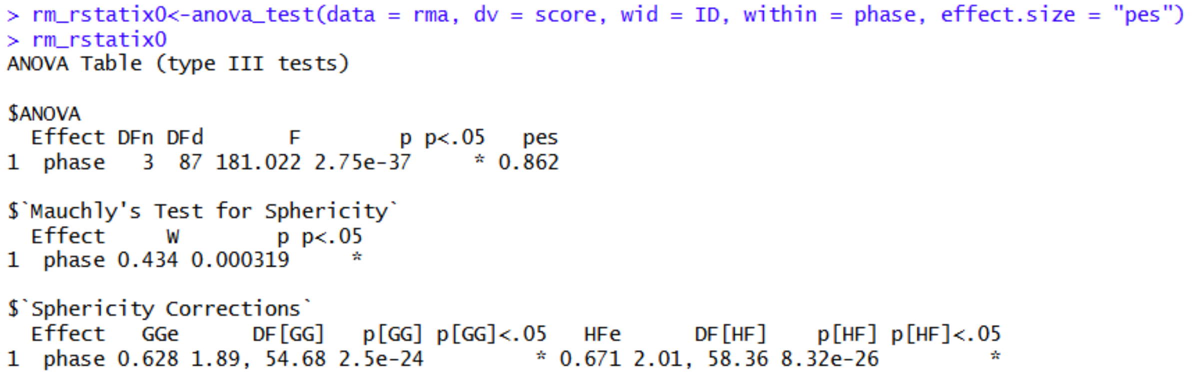


```
> rm_rstatix<-anova_test(data = rma, dv = score, wid = ID, within = phase)
> rm_rstatix
ANOVA Table (type III tests)

$ANOVA
  Effect DFn DFd       F        p p<.05   ges
1  phase   3  87 181.022 2.75e-37     * 0.482

$`Mauchly's Test for Sphericity`
  Effect     W        p p<.05
1  phase 0.434 0.000319     *

$`Sphericity Corrections`
  Effect   GGe      DF[GG]    p[GG] p[GG]<.05   HFe      DF[HF]    p[HF] p[HF]<.05
1  phase 0.628 1.89, 54.68 2.5e-24          * 0.671 2.01, 58.36 8.32e-26         *
```

Figure 2. RM ANOVA Summary Table

If the assumption of Sphericity were met, the ANOVA table could be used, and reported as follows: *F*(3, 87) = 181, *p* < .01, η^2_g = .48. However, Mauchly's Test of Sphericity is violated. Hence, either the Greenhouse-Geisser or Huynh-Feldt adjustment should be reported. The Greenhouse-Geisser adjustment is reported as follows: *F*(1.89. 54.68) = 181, *p* < .01, η^2_g = .48. In other words, the F value from the second line of the output is reported, along with the Greenhouse-Geisser adjusted degrees of freedom and the p-value from the "Sphericity Corrections" section. Alternatively, the Huynh-Feldt adjustment can be reported as follows: *F*(2.01, 58.36) = 181, *p* < .01, η^2_g = .48. Feel free to round as appropriate, e.g. *F*(2, 58.4) = 181, *p* < .01, η^2_g = .48.

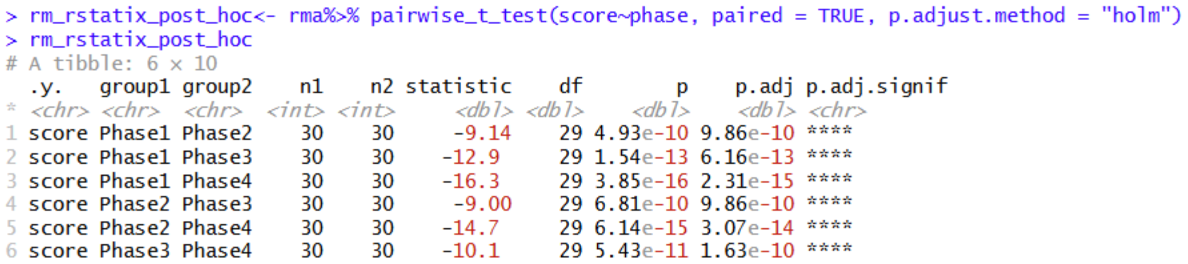
If partial eta squared is chosen as the effect size, then report as follows: *F*(1.9. 54.7) = 181, *p* < .01, η^2_p = .86. Note that the latter report uses the Greenhouse-Geisser adjustment, and the Huynh-Feldt adjustment may be reported instead.



```
> rm_rstatix0<-anova_test(data = rma, dv = score, wid = ID, within = phase, effect.size = "pes")
> rm_rstatix0
ANOVA Table (type III tests)

$ANOVA
  Effect DFn DFd       F        p p<.05   pes
1  phase   3  87 181.022 2.75e-37     * 0.862

$`Mauchly's Test for Sphericity`
  Effect     W        p p<.05
1  phase 0.434 0.000319     *

$`Sphericity Corrections`
  Effect   GGe      DF[GG]    p[GG] p[GG]<.05   HFe      DF[HF]    p[HF] p[HF]<.05
1  phase 0.628 1.89, 54.68  2.5e-24         * 0.671 2.01, 58.36 8.32e-26         *
```

Figure 3. ANOVA Table with Partial Eta Squared Effect Size

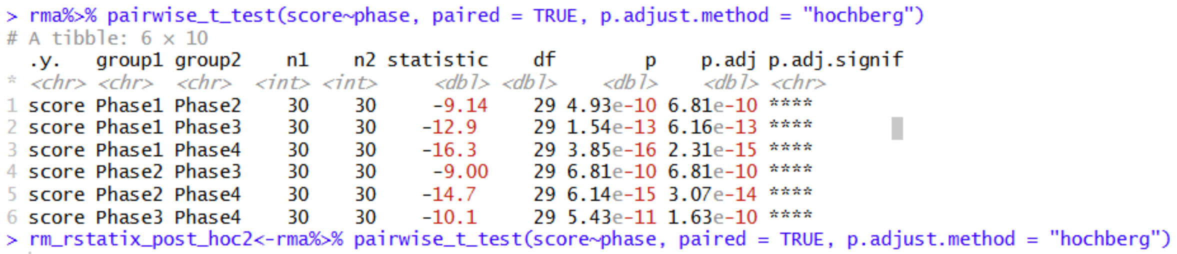
Finally, the output for the Holm post hoc test is shown below. Pairwise comparisons show that all means differed from one another. The descriptive statistics table shows that the score increased significantly from Phase 1 through Phase 4.



```
> rm_rstatix_post_hoc<- rma%>% pairwise_t_test(score~phase, paired = TRUE, p.adjust.method = "holm")
> rm_rstatix_post_hoc
# A tibble: 6 x 10
  .y.   group1 group2    n1    n2 statistic    df        p    p.adj p.adj.signif
* <chr> <chr>  <chr>  <int> <int>     <dbl> <dbl>    <dbl>    <dbl> <chr>
1 score Phase1 Phase2    30    30     -9.14    29 4.93e-10 9.86e-10 ****
2 score Phase1 Phase3    30    30    -12.9     29 1.54e-13 6.16e-13 ****
3 score Phase1 Phase4    30    30    -16.3     29 3.85e-16 2.31e-15 ****
4 score Phase2 Phase3    30    30     -9.00    29 6.81e-10 9.86e-10 ****
5 score Phase2 Phase4    30    30    -14.7     29 6.14e-15 3.07e-14 ****
6 score Phase3 Phase4    30    30    -10.1     29 5.43e-11 1.63e-10 ****
```

Figure 4. Post Hoc Tests using Holm Adjustment for Multiple Comparisons

Other options for post hoc testing are shown in Table 2. Simply use the adjustment you wish in the "*p.adjust.method*" command in all lower-case letters. For example, the Hochberg method is shown below.



```
> rma%>% pairwise_t_test(score~phase, paired = TRUE, p.adjust.method = "hochberg")
# A tibble: 6 x 10
  .y.   group1 group2    n1    n2 statistic    df        p    p.adj p.adj.signif
* <chr> <chr>  <chr>  <int> <int>     <dbl> <dbl>    <dbl>    <dbl> <chr>
1 score Phase1 Phase2    30    30     -9.14    29 4.93e-10 6.81e-10 ****
2 score Phase1 Phase3    30    30    -12.9     29 1.54e-13 6.16e-13 ****
3 score Phase1 Phase4    30    30    -16.3     29 3.85e-16 2.31e-15 ****
4 score Phase2 Phase3    30    30     -9.00    29 6.81e-10 6.81e-10 ****
5 score Phase2 Phase4    30    30    -14.7     29 6.14e-15 3.07e-14 ****
6 score Phase3 Phase4    30    30    -10.1     29 5.43e-11 1.63e-10 ****
> rm_rstatix_post_hoc2<-rma%>% pairwise_t_test(score~phase, paired = TRUE, p.adjust.method = "hochberg")
```

Figure 5. Post Hoc Tests using Hochberg Adjustment for Multiple Comparisons

Repeated Measures ANOVA using afex and emmeans

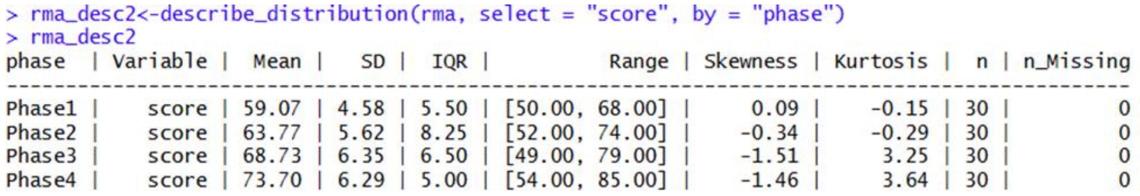
The *afex* package may also be used to compute RMA in tandem with the *emmeans* package to compute post hoc analysis. Visualization using the *ggstatsplot* package is also shown in this section.

Table 4. Parametric Repeated Measures ANOVA Syntax using *datawizard*, *afex* and *emmeans*

Package	Syntax	Purpose
	library(afex) library(datawizard) library(emmeans) library(ggstatsplot)	
datawizard	describe_distribution(rma, select = "score", by = "phase")	Descriptive statistics
afex	rm_afex<-aov_ez(data = rma, id = "ID", dv = "score", within = "phase", anova_table = list(es = "pes"))	Repeated measures ANOVA in *afex*.
	rm_afex	Shows ANOVA table with any adjustments. Greenhouse-Geisser is the default method.
	summary(rm_afex)	Same as above, but with more detail.
	rm_afex_HF<- aov_ez(data = rma, id = "ID", dv = "score", within = "phase", anova_table = list(es = "pes", correction = "HF"))	Use this syntax to get Huynh-Feldt adjustment instead.
emmeans	emm<-emmeans(rm_afex, ~ phase)	Shows means and other parameters for each level of the ANOVA.
	pairs(emm)	Pairwise comparisons; Tukey post hoc is the default.
	rm_afex_hochberg = pairs(emm, adjust = "hochberg")	Creates an object for Hochberg post hoc tests. Bonferroni, Hommel, Holm, Scheffe, Sidak, and other options may be found here.

ggstatsplot	rmplot<-ggwithinstats(data = rma, x = phase, y = score, type = "parametric", pairwise.comparisons = TRUE, p.adjust.method = "holm", title = "Scores by Phase")	Visualization.

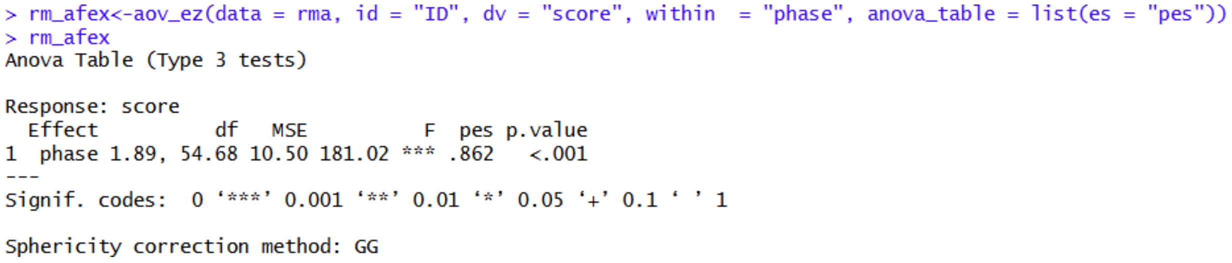
The *afex* package is also capable of computing parametric RMA analyses. Unlike *rstatix*, it does not offer descriptive statistics or post hoc tests. These may be respectively obtained using the *datawizard* and *emmeans* packages. Descriptive statistics using the *datawizard* package is shown below.



```
> rma_desc2<-describe_distribution(rma, select = "score", by = "phase")
> rma_desc2
phase   | Variable |  Mean |   SD |  IQR |          Range | Skewness | Kurtosis |  n | n_Missing
-----------------------------------------------------------------------------------------------
Phase1 |    score | 59.07 | 4.58 | 5.50 | [50.00, 68.00] |     0.09 |    -0.15 | 30 |         0
Phase2 |    score | 63.77 | 5.62 | 8.25 | [52.00, 74.00] |    -0.34 |    -0.29 | 30 |         0
Phase3 |    score | 68.73 | 6.35 | 6.50 | [49.00, 79.00] |    -1.51 |     3.25 | 30 |         0
Phase4 |    score | 73.70 | 6.29 | 5.00 | [54.00, 85.00] |    -1.46 |     3.64 | 30 |         0
```

Figure 6. Descriptive Statistics using the *datawizard* Package

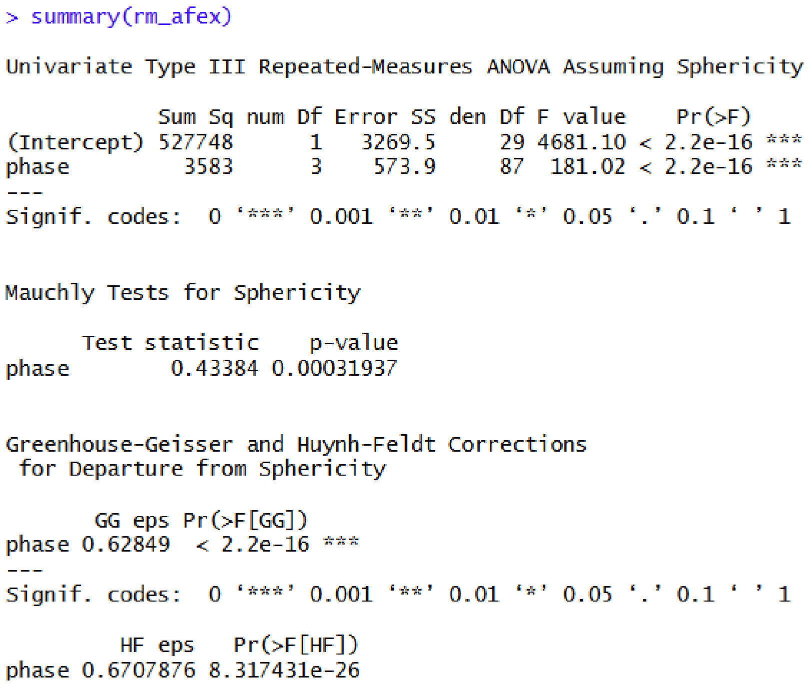
Note that the “aov_ez” command provides an abridged ANOVA table, while the summary command offers more detailed output. However, the abridged output mentions that the Greenhouse-Geisser adjustment is used, and also gives the adjusted degrees of freedom for this method. Therefore, this command intuitively suggests that Mauchly’s Test of Sphericity is violated.



```
> rm_afex<-aov_ez(data = rma, id = "ID", dv = "score", within  = "phase", anova_table = list(es = "pes"))
> rm_afex
Anova Table (Type 3 tests)

Response: score
  Effect          df   MSE          F  pes p.value
1  phase 1.89, 54.68 10.50 181.02 *** .862   <.001
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘+’ 0.1 ‘ ’ 1

Sphericity correction method: GG
```

Figure 7. Repeated Measures ANOVA Table using *afex*

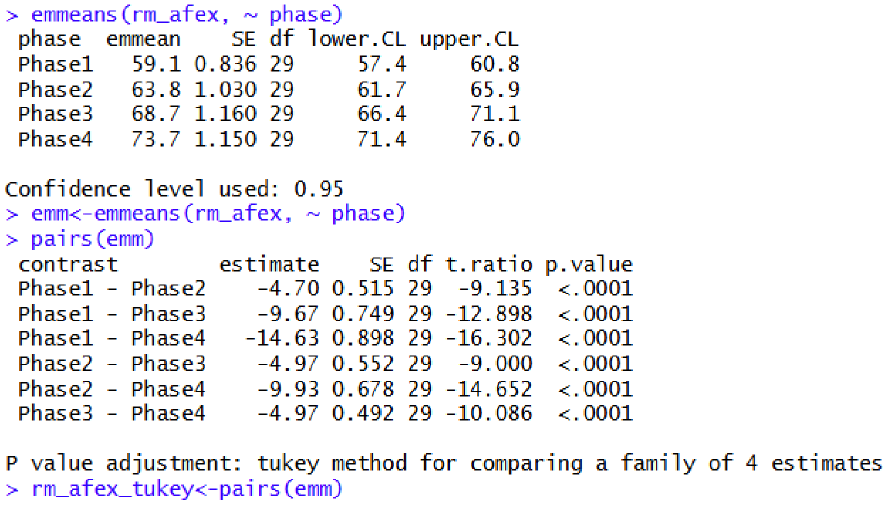
The more detailed output below does provide an actual value for Mauchly's Test, and p-values for both adjustments, but does not offer the adjusted degrees of freedom for either.



```
> summary(rm_afex)

Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

            Sum Sq num Df Error SS den Df F value    Pr(>F)    
(Intercept) 527748      1   3269.5     29 4681.10 < 2.2e-16 ***
phase         3583      3    573.9     87  181.02 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Mauchly Tests for Sphericity

      Test statistic    p-value
phase        0.43384 0.00031937


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

       GG eps Pr(>F[GG])    
phase 0.62849  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

         HF eps   Pr(>F[HF])
phase 0.6707876 8.317431e-26
```

Figure 8. Detailed *afex* ANOVA Output

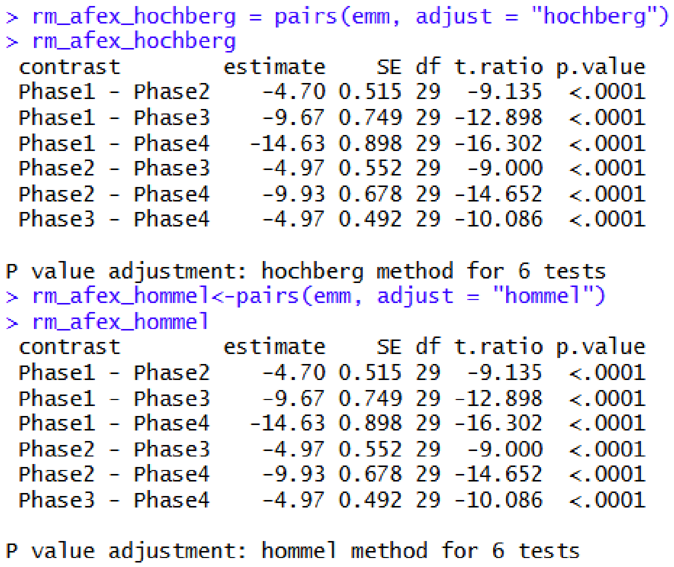
Post hoc testing may be obtained using the syntaxes below using the *emmeans* package. The Tukey adjustment is the default method.



```
> emmeans(rm_afex, ~ phase)
 phase  emmean    SE df lower.CL upper.CL
 Phase1   59.1 0.836 29     57.4     60.8
 Phase2   63.8 1.030 29     61.7     65.9
 Phase3   68.7 1.160 29     66.4     71.1
 Phase4   73.7 1.150 29     71.4     76.0

Confidence level used: 0.95 
> emm<-emmeans(rm_afex, ~ phase)
> pairs(emm)
 contrast        estimate    SE df t.ratio p.value
 Phase1 - Phase2    -4.70 0.515 29  -9.135  <.0001
 Phase1 - Phase3    -9.67 0.749 29 -12.898  <.0001
 Phase1 - Phase4   -14.63 0.898 29 -16.302  <.0001
 Phase2 - Phase3    -4.97 0.552 29  -9.000  <.0001
 Phase2 - Phase4    -9.93 0.678 29 -14.652  <.0001
 Phase3 - Phase4    -4.97 0.492 29 -10.086  <.0001

P value adjustment: tukey method for comparing a family of 4 estimates 
> rm_afex_tukey<-pairs(emm)
```

Figure 9. Tukey Post Hoc Comparisons using *emmeans*

Other post hoc tests may be obtained by specifying the adjustments as shown below.



```
> rm_afex_hochberg = pairs(emm, adjust = "hochberg")
> rm_afex_hochberg
 contrast        estimate    SE df t.ratio p.value
 Phase1 - Phase2    -4.70 0.515 29  -9.135  <.0001
 Phase1 - Phase3    -9.67 0.749 29 -12.898  <.0001
 Phase1 - Phase4   -14.63 0.898 29 -16.302  <.0001
 Phase2 - Phase3    -4.97 0.552 29  -9.000  <.0001
 Phase2 - Phase4    -9.93 0.678 29 -14.652  <.0001
 Phase3 - Phase4    -4.97 0.492 29 -10.086  <.0001

P value adjustment: hochberg method for 6 tests
> rm_afex_hommel<-pairs(emm, adjust = "hommel")
> rm_afex_hommel
 contrast        estimate    SE df t.ratio p.value
 Phase1 - Phase2    -4.70 0.515 29  -9.135  <.0001
 Phase1 - Phase3    -9.67 0.749 29 -12.898  <.0001
 Phase1 - Phase4   -14.63 0.898 29 -16.302  <.0001
 Phase2 - Phase3    -4.97 0.552 29  -9.000  <.0001
 Phase2 - Phase4    -9.93 0.678 29 -14.652  <.0001
 Phase3 - Phase4    -4.97 0.492 29 -10.086  <.0001

P value adjustment: hommel method for 6 tests
```

Figure 10. Hochberg and Hommel Post Hoc Comparisons using *emmeans*

Finally, visualization using the *ggstatsplot* package is shown below.

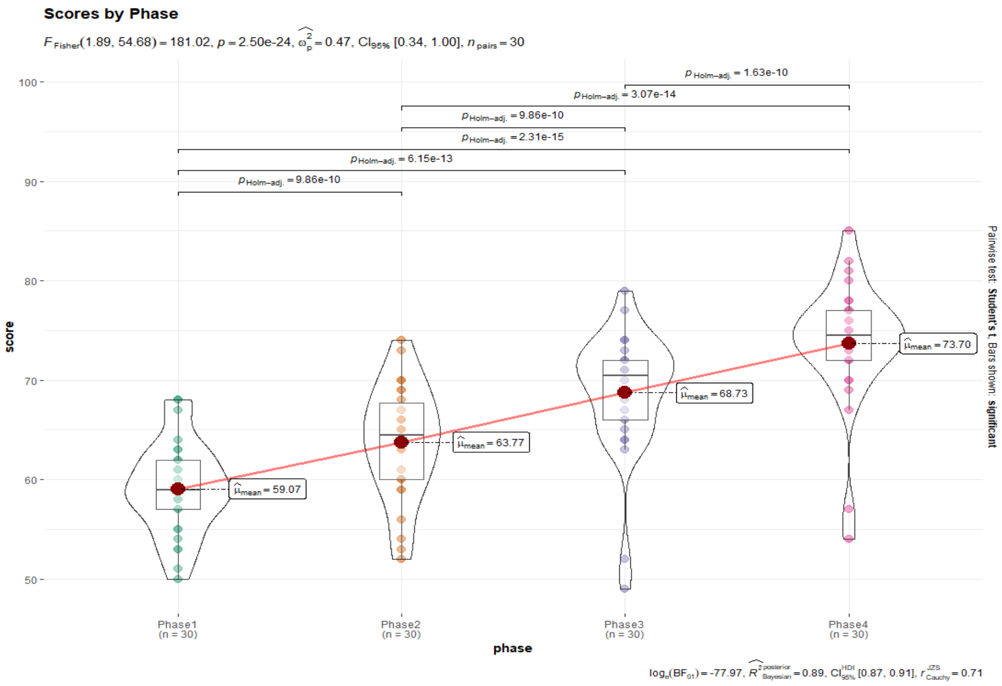


Figure 11. Visualization using *ggstatsplot*

Options for Violated Normality Assumptions

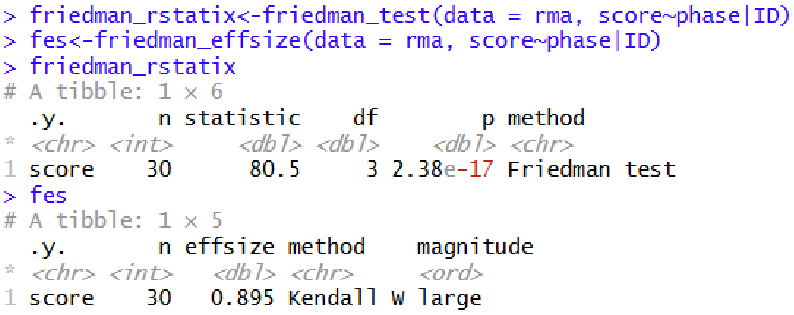
Friedman's ANOVA using afex

Friedman's ANOVA is a popular non-parametric option for heavily skewed data, or small sample sizes. The *rstatix* package also performs Friedman's ANOVA. The syntaxes are shown below, with *ggstatsplot* used for visualization.

Table 5. Friedman's ANOVA using *rstatix*

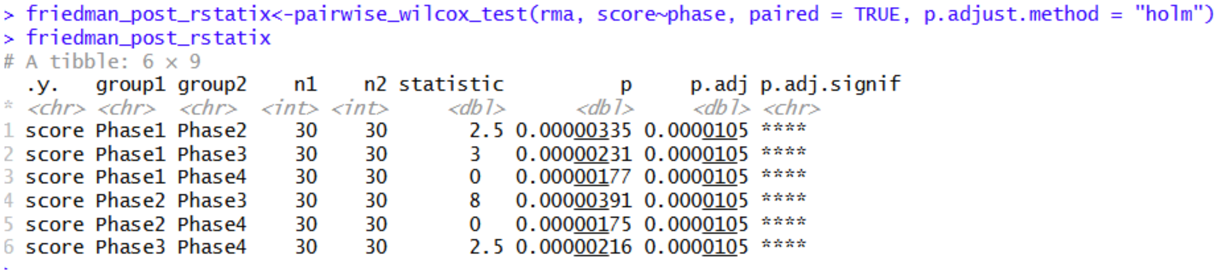
Package	Syntax	Purpose
rstatix	friedman_rstatix<-friedman_test(data = rma, score~phase|ID)	Friedman's ANOVA in *rstatix*.
	fes<-friedman_effsize(data = rma, score~phase|ID)	Generates Kendall's W.
	friedman_post_rstatix<-pairwise_wilcox_test(rma, score~phase, paired = TRUE, p.adjust.method = "holm")	Pairwise Wilcoxon tests with Holm adjustments.
	pairwise_wilcox_test(rma, score~phase, paired = TRUE, p.adjust.method = "bonferroni")	Same as above, except with Bonferroni adjustment.
ggstatsplot	friedmanplot<-ggwithinstats(data = rma, x = phase, y = score, type = "nonparametric", pairwise.comparisons = TRUE, p.adjust.method = "holm", title = "Scores by Phase")	Visualization. This package is helpful as it also gives an APA style report and post hoc comparisons.

The Friedman test and Kendall's W effect size outputs are shown below. They may be reported as follows: $\chi^2_{Friedman}$(3, n = 30) = 80.5, $p < .001$, W = .90.



```
> friedman_rstatix<-friedman_test(data = rma, score~phase|ID)
> fes<-friedman_effsize(data = rma, score~phase|ID)
> friedman_rstatix
# A tibble: 1 x 6
  .y.       n statistic    df        p method
* <chr> <int>     <dbl> <dbl>    <dbl> <chr>
1 score    30      80.5     3 2.38e-17 Friedman test
> fes
# A tibble: 1 x 5
  .y.       n effsize method    magnitude
* <chr> <int>   <dbl> <chr>     <ord>
1 score    30   0.895 Kendall W large
```

Figure 12. Friedman's ANOVA Output using *rstatix*



```
> friedman_post_rstatix<-pairwise_wilcox_test(rma, score~phase, paired = TRUE, p.adjust.method = "holm")
> friedman_post_rstatix
# A tibble: 6 x 9
  .y.   group1 group2    n1    n2 statistic          p     p.adj p.adj.signif
* <chr> <chr>  <chr>  <int> <int>     <dbl>      <dbl>     <dbl> <chr>
1 score Phase1 Phase2    30    30       2.5 0.00000335 0.0000105 ****
2 score Phase1 Phase3    30    30       3   0.00000231 0.0000105 ****
3 score Phase1 Phase4    30    30       0   0.00000177 0.0000105 ****
4 score Phase2 Phase3    30    30       8   0.00000391 0.0000105 ****
5 score Phase2 Phase4    30    30       0   0.00000175 0.0000105 ****
6 score Phase3 Phase4    30    30       2.5 0.00000216 0.0000105 ****
```

Figure 13. Pairwise Wilcoxon Post Hoc Comparisons with Holm Adjustments

Robust Repeated Measures ANOVA syntax using WRS2

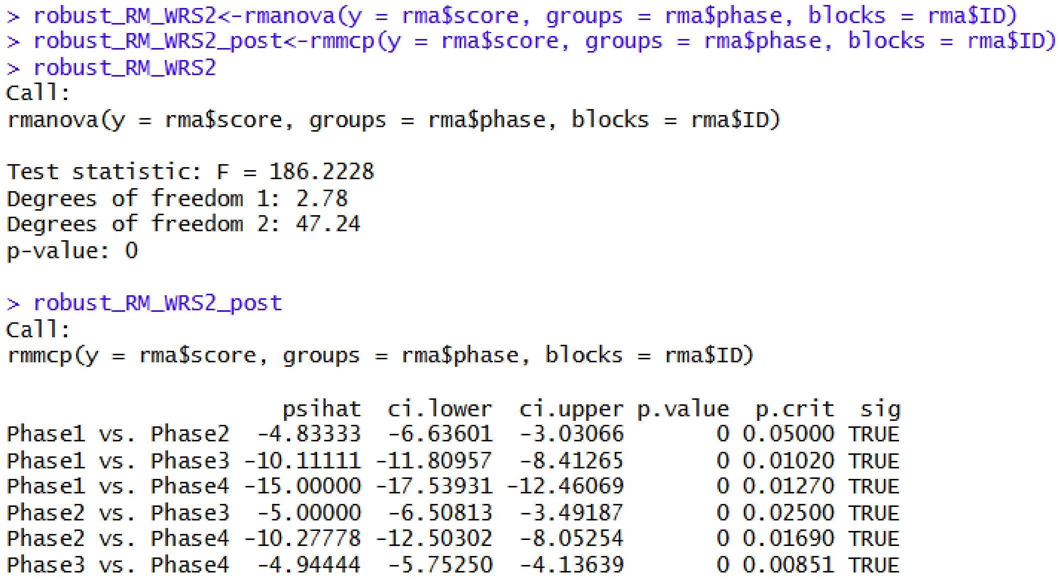
The *WRS2* package is another alternative. It offers robust and bootstrapped options for RMA and post hoc tests in case of violated assumptions. The syntaxes are shown in the table below.

Table 6. Syntax for Repeated Measures ANOVA using *WRS2*

Package	Syntax	Purpose
	library(WRS2)	
WRS2	robust_RM_WRS2<-rmanova(y = rma$score, groups = rma$phase, blocks = rma$ID)	Robust repeated measures ANOVA
	with(rma, (rmanova(y = score, groups = phase, blocks = ID)))	Alternate syntax for above.
	rmmcp(y = rma$score, groups = rma$phase, blocks = rma$ID)	Post hoc tests for trimmed means.
	rmanovab(y = rma$score, groups = rma$phase, blocks = rma$ID)	Bootstrapped robust ANOVA.
	rmanovab(y = rma$score, groups = rma$phase, blocks = rmaf$ID, nboot = 2000)	Same as above, specifying 2000 bootstrapped samples.
	pairdepb(y= rma$score, groups = rma$phase, blocks = rma$ID, nboot = 2000)	Bootstrapped post hoc tests for the above ANOVA.

The first option for *WRS2* is repeated measures ANOVA with 20% trimmed means. The null hypothesis is rejected, and should be reported as follows: $F(2.78, 47.24) = 186.2$, $p < .001$.

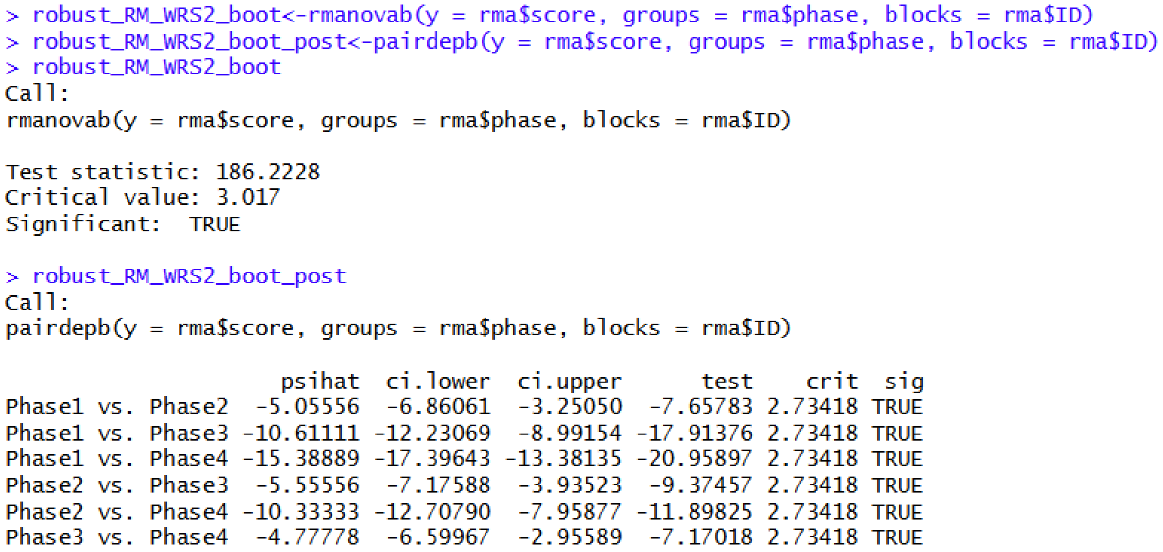
The interpretation is exactly the same as the Friedman's ANOVA in *rstatix*. However, a disadvantage of the *WRS2* package is that it does not offer an effect size for RMA.



```
> robust_RM_WRS2<-rmanova(y = rma$score, groups = rma$phase, blocks = rma$ID)
> robust_RM_WRS2_post<-rmmcp(y = rma$score, groups = rma$phase, blocks = rma$ID)
> robust_RM_WRS2
Call:
rmanova(y = rma$score, groups = rma$phase, blocks = rma$ID)

Test statistic: F = 186.2228
Degrees of freedom 1: 2.78
Degrees of freedom 2: 47.24
p-value: 0

> robust_RM_WRS2_post
Call:
rmmcp(y = rma$score, groups = rma$phase, blocks = rma$ID)

                     psihat  ci.lower  ci.upper p.value  p.crit  sig
Phase1 vs. Phase2  -4.83333  -6.63601  -3.03066       0 0.05000 TRUE
Phase1 vs. Phase3 -10.11111 -11.80957  -8.41265       0 0.01020 TRUE
Phase1 vs. Phase4 -15.00000 -17.53931 -12.46069       0 0.01270 TRUE
Phase2 vs. Phase3  -5.00000  -6.50813  -3.49187       0 0.02500 TRUE
Phase2 vs. Phase4 -10.27778 -12.50302  -8.05254       0 0.01690 TRUE
Phase3 vs. Phase4  -4.94444  -5.75250  -4.13639       0 0.00851 TRUE
```

Figure 14. Robust One-way Repeated Measures ANOVA using *WRS2*

The *WRS2* package also allows bootstrapped robust ANOVA and post hoc tests, as shown below. The overall null hypothesis is also rejected using this method. The bootstrapped robust RMA showed a significant effect of phase on score, as the test statistic (186.2) exceeded the critical value of 3.0, thus rejecting the null hypothesis. Note the interpretation of the post hoc tests is also similar to the *rstatix* analysis.



```
> robust_RM_WRS2_boot<-rmanovab(y = rma$score, groups = rma$phase, blocks = rma$ID)
> robust_RM_WRS2_boot_post<-pairdepb(y = rma$score, groups = rma$phase, blocks = rma$ID)
> robust_RM_WRS2_boot
Call:
rmanovab(y = rma$score, groups = rma$phase, blocks = rma$ID)

Test statistic: 186.2228
Critical value: 3.017
Significant:  TRUE

> robust_RM_WRS2_boot_post
Call:
pairdepb(y = rma$score, groups = rma$phase, blocks = rma$ID)

                     psihat  ci.lower  ci.upper      test    crit  sig
Phase1 vs. Phase2  -5.05556  -6.86061  -3.25050  -7.65783 2.73418 TRUE
Phase1 vs. Phase3 -10.61111 -12.23069  -8.99154 -17.91376 2.73418 TRUE
Phase1 vs. Phase4 -15.38889 -17.39643 -13.38135 -20.95897 2.73418 TRUE
Phase2 vs. Phase3  -5.55556  -7.17588  -3.93523  -9.37457 2.73418 TRUE
Phase2 vs. Phase4 -10.33333 -12.70790  -7.95877 -11.89825 2.73418 TRUE
Phase3 vs. Phase4  -4.77778  -6.59967  -2.95589  -7.17018 2.73418 TRUE
```

Figure 15. Bootstrapped Robust One-way Repeated Measures ANOVA using *WRS2*

Aligned Rank Transformed ANOVA using ARTool

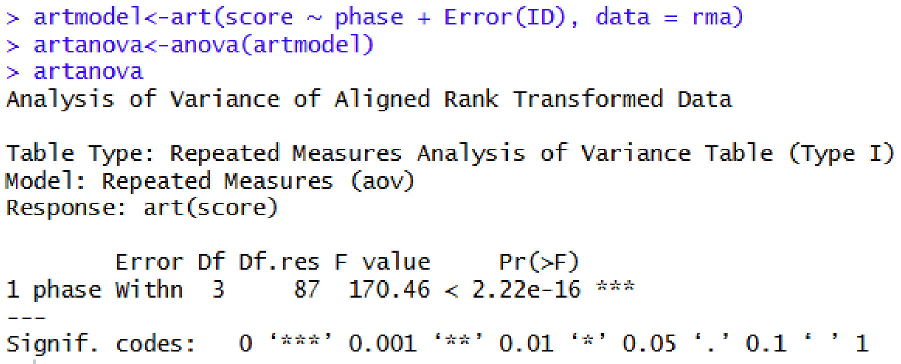
The *ARTool* package is another option for nonparametric analysis. While specialized for factorial designs (Wobbrock et al., 2011), its use for one-way RMA is demonstrated below. It is somewhat more labor intensive than *WRS2*, and may be suitable for more advanced *R* users. It may be used with the *effectsize* and *emmeans* packages to obtain effect sizes and post hoc tests respectively. The syntaxes and workflow are shown in the table below.

Table 7. Syntaxes for ANOVA of Aligned Rank Transformed Data using *ARTool*, *effectsize*, and *emmeans* Packages

Package	Syntax	Purpose
	library(ARTool)	
	rma$ID <-factor(rma$ID)	Convert the ID variable into a factor.
	rma$phase <-factor(rma$phase)	Same for independent variable.
ARTool	artmodel<-art(score ~ phase + Error(ID), data = rma)	Transformation of repeated measures data using *ARTool*.
	artanova<- anova(artmodel)	Repeated measures ANOVA of aligned rank transformed data in *ARTool*.
	artanova	Shows ANOVA table.
	art_phase<-artlm(artmodel, "phase")	
effectsize	eta_squared(art_phase)	Partial eta squared.
	art_effectsize2<-eta_squared(art_phase, partial = FALSE)	Generalized eta squared.
	eta_squared(art_phase, partial = FALSE, generalized = TRUE)	Same as above.
emmeans	art_post <-emmeans(art_phase, pairwise ~ phase)	Pairwise comparisons with Tukey adjustment as the default method. Multiple post hoc tests are shown for demonstration only. Choose one post hoc test ahead of time.

Package	Syntax	Purpose
	emmeans(art_phase, pairwise ~ phase, adjust = "bonferroni")	Bonferroni adjustment.
	art_post_holm<-emmeans(art_phase, pairwise ~ phase, adjust = "holm")	Holm adjustment. Also offers Scheffe, Sidak, and others.

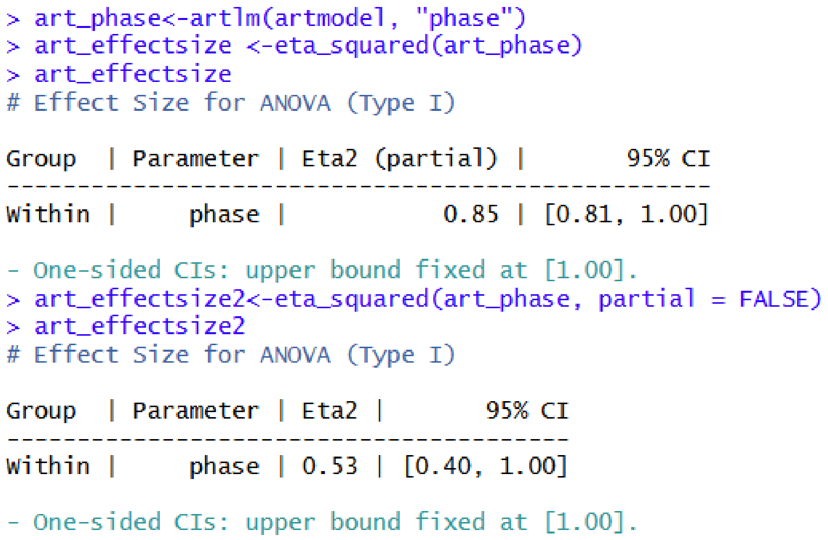
The figure below shows the ANOVA table. Note that obtaining the ANOVA table is comprised of two steps: transforming the data, followed by the ANOVA analysis.



```
> artmodel<-art(score ~ phase + Error(ID), data = rma)
> artanova<-anova(artmodel)
> artanova
Analysis of Variance of Aligned Rank Transformed Data

Table Type: Repeated Measures Analysis of Variance Table (Type I)
Model: Repeated Measures (aov)
Response: art(score)

        Error Df Df.res F value     Pr(>F)
1 phase Withn  3     87  170.46 < 2.22e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 16. Syntax and Output for ANOVA of Aligned Rank Transformed Data

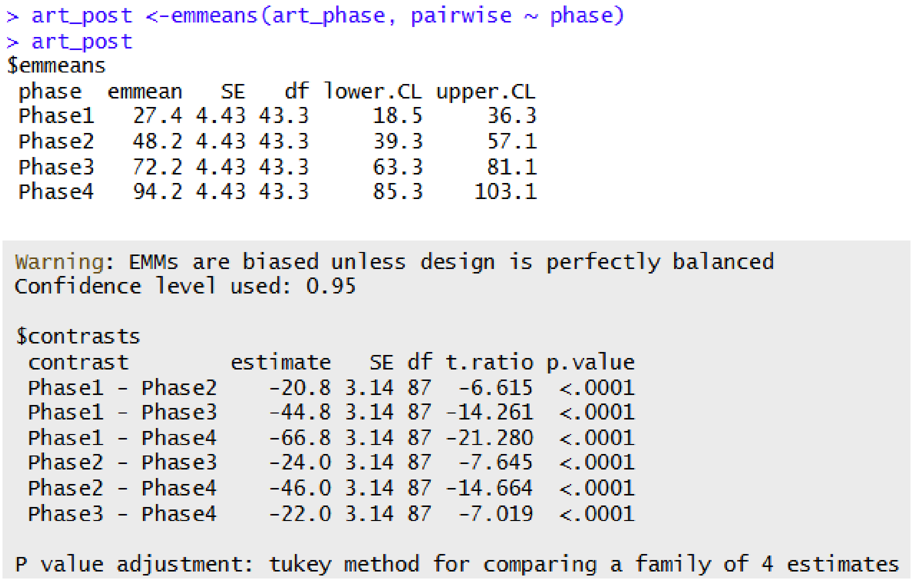
The *effectsize* package may be used to obtain either partial eta squared or generalized eta squared. Note that partial eta squared tends to be upwardly biased. The analysis may be reported as follows: $F(3, 87) = 170.5$, $p < .001$, $\eta^2_g = .53$ for generalized eta squared, or $F(3, 87) = 170.5$, $p < .001$, $\eta^2_p = .85$ for partial eta squared.



```
> art_phase<-artlm(artmodel, "phase")
> art_effectsize <-eta_squared(art_phase)
> art_effectsize
# Effect Size for ANOVA (Type I)

Group  | Parameter | Eta2 (partial) |       95% CI
--------------------------------------------------
Within |     phase |           0.85 | [0.81, 1.00]

- One-sided CIs: upper bound fixed at [1.00].
> art_effectsize2<-eta_squared(art_phase, partial = FALSE)
> art_effectsize2
# Effect Size for ANOVA (Type I)

Group  | Parameter | Eta2 |       95% CI
----------------------------------------
Within |     phase | 0.53 | [0.40, 1.00]

- One-sided CIs: upper bound fixed at [1.00].
```

Figure 17. Effect Sizes for ANOVA of Aligned Rank Transformed Data

As shown below, the Tukey adjustment is the default for pairwise comparisons. As aforementioned, other post hoc tests may be chosen in *emmeans* depending on the user's preference and objective.



```
> art_post <-emmeans(art_phase, pairwise ~ phase)
> art_post
$emmeans
 phase  emmean   SE   df lower.CL upper.CL
 Phase1   27.4 4.43 43.3     18.5     36.3
 Phase2   48.2 4.43 43.3     39.3     57.1
 Phase3   72.2 4.43 43.3     63.3     81.1
 Phase4   94.2 4.43 43.3     85.3    103.1

Warning: EMMs are biased unless design is perfectly balanced
Confidence level used: 0.95

$contrasts
 contrast        estimate   SE df t.ratio p.value
 Phase1 - Phase2    -20.8 3.14 87  -6.615  <.0001
 Phase1 - Phase3    -44.8 3.14 87 -14.261  <.0001
 Phase1 - Phase4    -66.8 3.14 87 -21.280  <.0001
 Phase2 - Phase3    -24.0 3.14 87  -7.645  <.0001
 Phase2 - Phase4    -46.0 3.14 87 -14.664  <.0001
 Phase3 - Phase4    -22.0 3.14 87  -7.019  <.0001

P value adjustment: tukey method for comparing a family of 4 estimates
```

Figure 18. Post Hoc Tests for ANOVA of Aligned Rank Transformed Data

Discussion and Conclusion

R offers several options for one-way RMA analysis. In this tutorial, the *rstatix* package was used as it is a comprehensive package for parametric and nonparametric options. It offers descriptive statistics, parametric and nonparametric RMA, effect sizes, and post hoc tests. The *afex* package also offers parametric RMA and effect sizes. However, another package such as *datawizard* may be used to get descriptive statistics, and *emmeans* may be used for post hoc tests. The *WRS2* package offers robust and bootstrapped RMA and post hoc tests, although it does not offer descriptive statistics or an effect size. The *ARTool* package may be used in tandem with the *effectsize* and *emmeans* packages to offer effect sizes and post hoc tests respectively.

While four options are shown for RMA in this paper, it is not a comprehensive list of packages for this method. Users are encouraged to explore other packages at their discretion. This method is included in this tutorial, as it naturally progresses from the paired t-test, just as independent ANOVA naturally progresses from the independent t-test. That said, RMA is not taught in all introductory courses, and instructors may choose to include or omit as they see fit. Multilevel models are an option for violations of the assumptions of relatedness and incomplete data. As this content is meant for an introductory course, these are not included.

R offers several options for RMA analysis and visualization. The *rstatix* package offers the most comprehensive option for parametric and nonparametric analyses. The *afex* package is a good option for parametric analysis, while *WRS2* and *ARTool* offer options for violated assumptions. Readers are encouraged to practice using these packages and explore others at their own discretion.

References

Ben-Shachar, M., Lüdecke, D., & Makowski, D. (2020). *effectsize*: Estimation of effect size indices and standardized parameters. *Journal of Open Source Software, 5*(56), 2815. doi: 10.21105/joss.02815

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences (2nd ed.)*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Ellis, P. D. (2017). *The essential guide to effect sizes : statistical power, meta-analysis and the interpretation of research results*. Cambridge University Press.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics* (4th ed.). Sage Publications.

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. SAGE.

Kassambara, A. (2025). *rstatix: Pipe-friendly framework for basic statistical tests. R* package version 0.7.2.

Kay, M., Elkin, L., Higgins, J., & Wobbrock, J. (2025). *ARTool: Aligned rank transformation for nonparametric factorial ANOVAs.* doi:10.5281/zenodo.594511

Lenth, R., & Piaskowski, J. (2025). *Emmeans: Estimated marginal means, aka least-squares means.* doi:10.32614/CRAN.package.emmeans

Mair, P., & Wilcox, R. (2019). Robust statistical methods in *R* using the *WRS2* package. *Behavior Research Methods*, *52*. https://doi.org/10.3758/s13428-019-01246-w

Patil, I. (2021). Visualizations with statistical details: The '*ggstatsplot*' approach. *Journal of Open Source Software, 6*(61), 3167, doi:10.21105/joss.03167

Patil, I., Makowski, D., Ben-Shachar, M. S., Wiernik, B. M., Bacher, E., & Ludecke, D. (2022). datawizard: An R Package for Easy Data Preparation and Statistical

Transformations. *Journal of Open Source Software,* 7(78), 4684, https://doi.org/10.21105/joss.04684

Posit (2025). *RStudio Desktop*. Posit. https://posit.co/download/rstudio-desktop/

R Core Team (2025). *R: A language and environment for statistical computing. R* Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Singmann H., Bolker B., Westfall J., Aust F., & Ben-Shachar, M. (2025). *afex: Analysis of factorial experiments. R* package version 1.5-0, https://github.com/singmann/afex

Wickham, H., Vaughan, D., & Girlich, M. (2024). *tidyr: Tidy Messy Data [R package tidyr version 1.1.3]* https://doi.org/10.32614/CRAN.package.tidyr. *R* package version 1.3.1

Wobbrock, J. O., Findlater, L., Gergle, D., & Higgins, J. J. (2011). The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures. *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI '11*. https://doi.org/10.1145/1978942.1978963

Citation

Lindsay, K. G. (2026). One-way analysis of variance II: Repeated measures ANOVA. In *Applied Statistics 101 using R: A primer for an introductory class in statistical programming* (pp. 82-101). ISTES.

The purpose of this book is to serve as a guide for students concurrently learning statistics and R. Students new to both programming and statistics may experience increased cognitive load as they have to learn both concepts for the first time. This book is meant to reduce that barrier. That said, students who are familiar with statistical content and not coding, or computer science students learning statistics may also benefit. The chapters focus on pragmaticworkflows in R, applied statistical reasoning, as well as interpretation of results. As R is free and open source, proprietary tools are not required. This also makes the book suitable for institutions with limited resources, and for anyone wishing to practice outside of formal coursework.

The methods introduced include descriptive statistics, t-tests, z-tests, chi-square tests, correlation, regression, and one-way analyses of variance. Assumption checks, non-parametric alternatives, and effect sizes are also included, though these are not strictly required for an introductory course. However, they may help to develop habits that transfer to additional coursework, research, hands-on data analysis, and interpretation. Where possible, several packages are discussed for each task in order to demonstrate R’s flexibility as an ecosystem. Users of this manual may use these examples as starting points, and adapt other packages as their confidence grows and as they see fit.The datasets featured in this manual may be obtained using this link: https://osf.io/mrw5v/overview.

Although the order of the chapters is intentional, the book is intended to be modular. Readers are not obligated to read through the book sequentially. Chapters are written to stand alone, and the book may be read in any order. All examples use synthetic instructional datasets.

Above all, this book is meant to be practical and easy to use. Learning R and statistics should be challenging in productive ways, but not discouraging. I hope that this book helps students to build both competence and confidence and helps instructors to build a replicable bridge between statistical concepts and computing.

Dr. Keston Lindsay

iSTES

ISBN: 978-1-952092-93-0