Akdeniz Eğitim Araştırmaları Dergisi, 2021, Cilt 15 Sayı 35, Sayfa 1-17
Mediterranean Journal of Educational Research 2021, Volume 15 Issue 35, Page 1-17
https://doi.org/10.29329/mjer.2021.340.1

R Programming Language and a Sample Application

Taha Eser [1]

Abstract: The aim of this study is to introduce R programming language, which became popular especially in recent years and to show how to interpret Results obtained through a sample application. For this purpose, the interface of R Studio program, its superior aspects, how to import the data to the program, a sample analysis and how to get support for the program were discussed. After explaining how R programming language emerged and its historical development, RStudio interface will be mentioned. The study shows how to perform the analysis with the help of functions after the installation of RStudio. Within the scope of sample analysis, the same achievement test for the statistics course was administered on 30 students who were randomly assigned to experimental and control groups before and after the experimental procedure. After taking the pre-test scores of the students under control, the presence of a statistically significant difference between the post-test scores of the experimental and control groups, to which two different methods were applied, was tested by covariance analysis. Within the scope of the research, it has been concluded that the use of the R program, which is free, open source and therefore can be developed, has a strong online user community and where fast software updates can be performed, should be expanded.

Keywords: R Programming Language, Rstudio, Covariance Analysis

Geliş Tarihi: 31.07.2020 – **Kabul Tarihi:** 15.03.2021 – **Yayın Tarihi:** 25.03.2021

DOI: 10.29329/mjer.2021.340.1

[1] **Taha Eser,** Dr., Assessment and Evaluation in Education, Educational Sciences, ORCID: 0000-0001-7031-1953

Email: tahaeser@gmail.com

INTRODUCTION

There are many statistics programs used by researchers. Since the statistical software that researcher would like to use within a research context is closely related to what researcher wants to do, it is not possible to claim that any statistical software is superior to others. However, even though it is not possible to say that one statistical software is better than the others, there may be differences in terms of cost, service in the user's native language, ease of use, user-friendly interface, providing detailed results on relevant analysis and high-quality graphics regarding the selection of the software to be used. One of the biggest factors that play a role in the formation of these differences is the software language. It can be said that R, Python, SQL, Java, Scala, Matlab and Julia are the most commonly used programming languages in data analysis by researchers today (Zuur, Ieno & Meesters, 2009; Eser, Aksu & Güzeller, 2019). R is a powerful programming language for statistical computations and creating graphs. The most important source of this power of R is related to its constant change and development. At this point, R can be likened to a constantly growing cell with new small cells adding to it (Wenables & Smith, 2002). At the same time, R provides users with an environment for statistical computations and graphs. The platform where R is used is called “R Statistical Software Development Environment”. R is a GNU (GNU is not Unix) project similar to the S language developed by John Chambers and colleagues at Bell Labs (formerly AT&T, now Lucent Technologies). R can also be considered as an extension of the S language. The S language is a tool, often preferred for conducting research in statistical methodology and R offers an Open Source way to participate in this activity. There are some significant differences between R and S language, most code written for S cannot be changed under R. In order to learn how to work with R, it is necessary to have grasp of the S language in a basic sense. S programming language is a programming language that is easy to learn and enjoyable (Zuur et al., 2009; Knell, 2014).

Considering the fact that it is open source, free of charge, operates independent of the platform and the analyse performed with SPSS, Statistica, Minitab etc. can be carried out under a single program, it can be said that R Statistical Software Development Environment is widely used especially amongst statisticians. As an extension of being open source, the environment is supported by a comprehensive user community. R Development Core Team and the Comprehensive R Archive Network (CRAN) are at the center of these user communities. As a mission, R development core team supports the latest used software in every aspect in terms of functionality. “Comprehensive R Archive Network” CRAN can be considered as an archive containing packages, files and documents. Each package of CRAN is tested with the new core package. There is also a document for each package that provides information about the content of the package and how it is used. In case of experiencing any problems with the package, the problem is reported to and solved by the member who developed the package. Thus, a certain level of quality is offered to users (Chambers, 2008; Eser et al., 2019).

R programming language is not solely used in R statistical software development environment. The user can also use R programming language in any of the integrated software development environments (IDE) available on the market. Regarding the integrated software development environments in which R programming language can be used, RStudio is the most popular one. RStudio can be said to be the most widely used R integrated software development environment in many areas such as engineering, medicine and social sciences (Zuur et al., 2009; Campbell, 2019).

RStudio is an open source integrated software development environment that allows you to interact more easily with R statistical software development environment. This app has a user-friendly interface. The interface is designed to clearly display and manage graphs, data tables, R codes and outputs. RStudio has pop-up menus, windows with multiple tabs, and many customization options. RStudio application can easily import data files with Excel, SPSS, SAS, STATA, SYTAT extensions. With RStudio, users get an environment where they can easily write code. In this interface, ready-to-use code packages can be installed and used very easily. With RStudio, it is very easy to display, work on and store objects such as data set, vector, scalar, matrix, output (Campbell, 2019; Eser et al., 2019). With RStudio, researchers can easily access data sets. Even an entry-level user can easily access drawn graphics via RStudio. In addition, scripts can also be used within the scope of RStudio. By using scripts, an application record is created, the analysis is recorded while performing the analysis and then the script used in the analysis can be reused to perform the whole analysis again (Cirillo, 2016).

This study aimed to give information about R software in functional terms and reveal the differences of the software from other similar ones. For this purpose, the information about the superior aspects of R Statistical Software Development Environment, the superiorities of RStudio, the installation of R Statistical Software Development Environment, the installation of RStudio, the interface of RStudio, and the package installation using the packages tab within the scope of RStudio are given in the following sections; an application was carried out with sample analysis and outputs using the software.

Superior Aspects of R Statistical Software Development Environment

Being open source and free of charge, having different version compatible with many software and hardware, working in harmony with other programming languages, performing hypothesis testing, data management, machine learning, data mining methods (clustering, classification, decision trees, association rules, artificial neural networks, time series analysis, text mining, social network analysis, etc.), operational research, reporting and presentation are superior aspects of R that should be known by all researchers (Crawley, 2007; Knell, 2014).

The superior aspects of R statistical software development environment mentioned here should not be considered as a full list. The biggest benefit of being an open source system is the ability to constantly add something to the system. In this context, regardless of what type of operation is being processed on the data, it is likely to be performed in R statistical software development environment. Although the process in question may be performed more easily via menus in other programs, R offers various option to the researchers about performing that job by being open source.

Program support

Sites such as https://support.rstudio.com/hc/en-us, http://www.sthda.com/english/, https://rpubs.com/, https://stats.stackexchange.com/ are online platforms for fast and efficient response to code, function, object and package problems related to R environment and RStudio. While https://support.rstudio.com/hc/en-us contains detailed information about the use of RStudio, http://www.sthda.com/english/ and https://rpubs.com/ provide guidance about how most of the analysis that can be performed with RStudio takes place step by step and the interpretation of the outputs. In addition, https://stats.stackexchange.com/ is very helpful in solving code, function and package problems.

Statistical analysis can be performed either by using direct codes in R interface, or by using “RStudio”, a free, open source integrated development environment for R. RStudio basically helps to write R codes faster and more efficiently. The superior aspects of RStudio over statistical package programs are mentioned below.

Superior Aspects of Rstudio Over Statistical Software Packages

RStudio provides ease of writing scripts.

RStudio facilitates viewing and interacting with the objects stored in R environment.

RStudio makes graphics more accessible for an ordinary user.

RStudio has an available version that is compatible with all operating systems.

RStudio is constantly changing and evolving to meet all the needs of researchers.

Installation of R Statistical Software Development Environment of Personal Computers

The current version of R statistical software development environment is R 3.5.3. CRAN releases a new version of the program each year, where the number in the middle of the 3-digit coding template indicates the program version and it is increased by one each year, representing major changes. R 3.2.0 was released in 2015, 3.3.0 in 2016 and 3.4.0 in 2017. The current version of the program is R 3.6.0. The last digit in the version number represents minor updates. Most R functions are generally backward compatible with previous versions (Eser et al., 2019).

To install R statistical software development environment, one should login to https://cran.r-project.org/bin/windows/base/. There is a link “Download R 3.6.0 for Windows” at the top of the main page to download the set-up file to the computer.

Installation of RStudio

The user can install either R Statistical Software Development Environment or RStudio first to the computer. However, in order to run RStudio, R Statistical Software Development Environment should be installed on the computer following the aforementioned steps. Once R Statistical Software Development Environment is installed on the computer, you should login to the official site of RStudio, www.rstudio.com. Click the "Download" tab on the main page to download the setup file of the RStudio.

RStudio's Interface

After successful installation of RStudio on the computer, RStudio shortcut will be displayed in the Start menu of the computer. The menus in the main window of RStudio, which will appear after clicking on RStudio shortcut, are shown in Figure 1. This window is the main screen of RStudio and consists of four main sections.

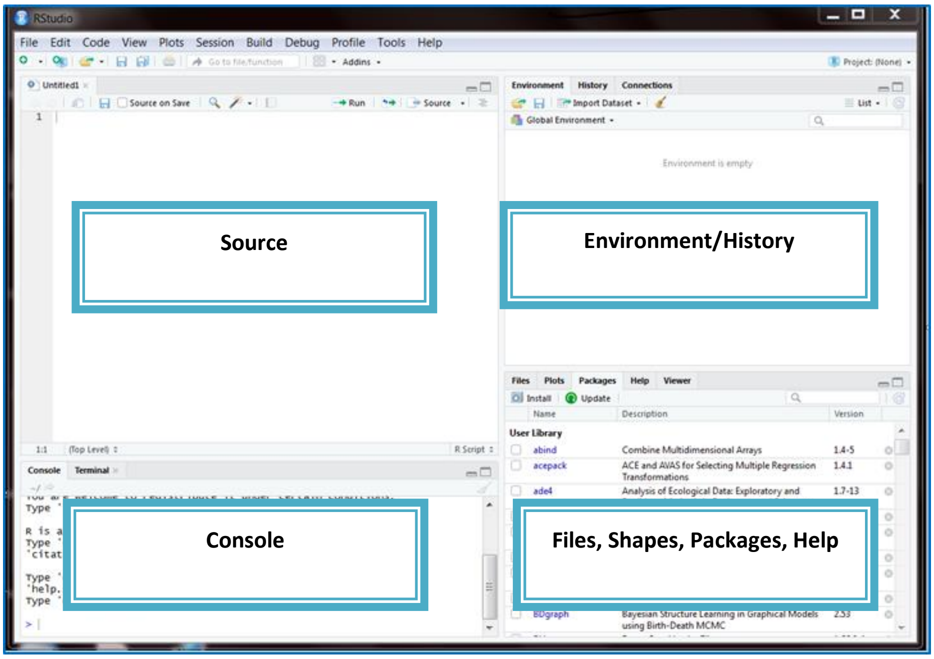

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
Project: (None)
Untitled1
Source on Save
Run
Source
Environment History Connections
Import Dataset
List
Global Environment
Environment is empty
Source
Environment/History
Files Plots Packages Help Viewer
Install Update
Name Description Version
User Library
abind Combine Multidimensional Arrays 1.4-5
acepack ACE and AVAS for Selecting Multiple Regression Transformations 1.4.1
ade4 Analysis of Ecological Data: Exploratory and 1.7-13
BDgraph Bayesian Structure Learning in Graphical Models using Birth-Death MCMC 2.53
(Top Level)
R Script
Console Terminal
Console
Files, Shapes, Packages, Help

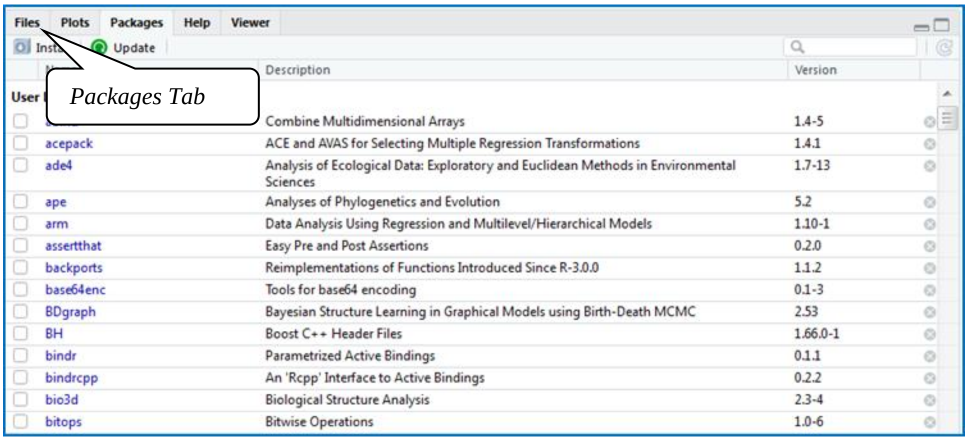
Figure 1 Main Windows of RStudio

The top left panel in RStudio interface in Figure 3, is the Source panel where commands can be written and saved, and notes can be taken about the works performed. When you run a command from the source panel, the command is sent to the Console panel for execution. It is possible to have more than one source or script in the Source panel. The data set, which was imported to the program via the Import Dataset tab on the Environment/History panel at the top right of the screen, is displayed in the Source panel. The panel at the top right of RStudio interface contains RStudio environment as well as the history of the earlier commands. The Environment / History panel is also the place where different objects created by the user can be seen and the data set is transferred to RStudio. The panel at the bottom left of RStudio interface is the place where the action occurs. This panel is often called as

Console. Each time you start RStudio, the console will have the same text at the top showing R version you are working with. Basically, it is for interacting with R statistical software development environment, writing commands, and interpreting the output. These commands and syntaxes have been created over the years and help many users in accessing data, and editing, defining and recalling statistical computations. The panel at the bottom right of RStudio interface is the panel where the files can be accessed, it includes the packages that allow the analysis to be performed, the shapes formed as a result of the analyse and the help panel. The next part of the study provides information about the installation steps of the packages used in RStudio.

Install a Package Using the Package Tab in RStudio

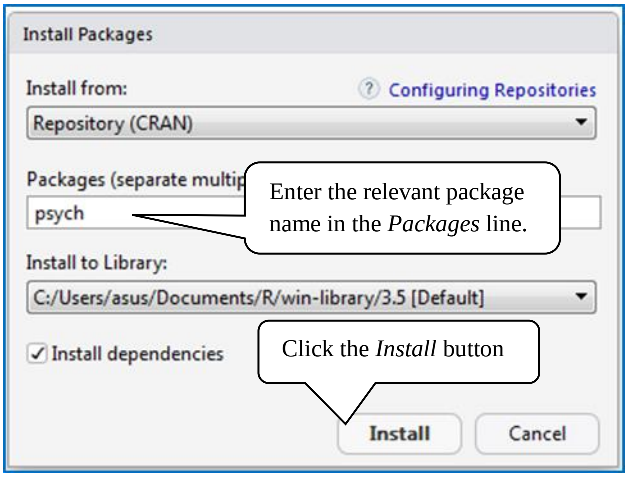
The first thing to do to install packages using the Packages tab is to activate the menu by clicking the Packages tab in RStudio interface. This tab is automatically displayed every time a RStudio session is opened. Figure 2 shows the Packages tab in RStudio interface.



		Description	Version
□		Combine Multidimensional Arrays	1.4-5
□	acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
□	ade4	Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences	1.7-13
□	ape	Analyses of Phylogenetics and Evolution	5.2
□	arm	Data Analysis Using Regression and Multilevel/Hierarchical Models	1.10-1
□	assertthat	Easy Pre and Post Assertions	0.2.0
□	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.2
□	base64enc	Tools for base64 encoding	0.1-3
□	BDgraph	Bayesian Structure Learning in Graphical Models using Birth-Death MCMC	2.53
□	BH	Boost C++ Header Files	1.66.0-1
□	bindr	Parametrized Active Bindings	0.1.1
□	bindrcpp	An 'Rcpp' Interface to Active Bindings	0.2.2
□	bio3d	Biological Structure Analysis	2.3-4
□	bitops	Bitwise Operations	1.0-6

Figure 2 RStudio Packages Menu

Click the Install tab under the Packages tab. Researchers should select the desired packages from the alphabetically ordered list and tick the box in front of this package. In this screen, the names of the packages in the library that researchers have created for themselves, explanations about the package and the version of the package are listed. Clicking Install tab displays RStudio package installation window. The package name to be used is written in the Packages line, at the middle of RStudio package installation window. For example, in figure 3, the “psych” package, which contains the codes for the calculations used in psychology, psychometry, and personality research, is written on the Packages line. The package written on the Packages line will be installed by clicking on the Install button.



Install Packages

Install from: Repository (CRAN)

Packages (separate multip…: psych

Install to Library: C:/Users/asus/Documents/R/win-library/3.5 [Default]

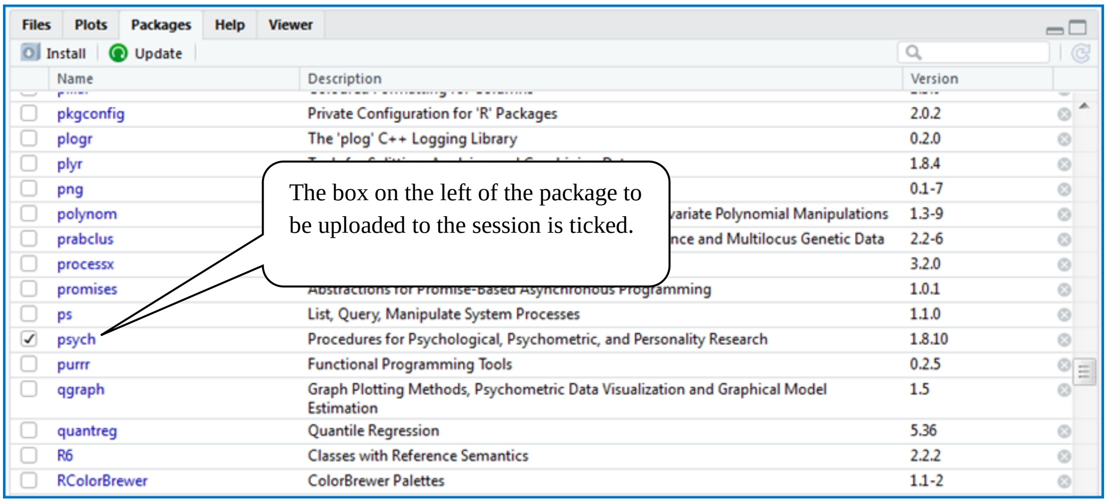
☑ Install dependencies

Enter the relevant package name in the *Packages* line.

Click the *Install* button

Figure 3 Installation of Packages in RStudio

After clicking the Install button in RStudio package installation window, the package you entered in the Packages line is quickly installed. When installing the package, the Console panel of RStudio interface displays information about the package being installed. This command line contains information such as the name of the package, where to install the package, from where the package was downloaded, the size of the package, and the installation status. Following the installation of the package, it should first be uploaded into the RStudio session in order to run the applications. To upload the package into RStudio session, tick the box on the left of the installed package in the list below the Packages tab. As an example, Figure 6 shows how to upload the "psych" package, installed in the previous step, into the session.



	Name	Description	Version
□	pkgconfig	Private Configuration for 'R' Packages	2.0.2
□	plogr	The 'plog' C++ Logging Library	0.2.0
□	plyr		1.8.4
□	png		0.1-7
□	polynom	variate Polynomial Manipulations	1.3-9
□	prabclus	nce and Multilocus Genetic Data	2.2-6
□	processx		3.2.0
□	promises	Abstractions for Promise-Based Asynchronous Programming	1.0.1
□	ps	List, Query, Manipulate System Processes	1.1.0
☑	psych	Procedures for Psychological, Psychometric, and Personality Research	1.8.10
□	purrr	Functional Programming Tools	0.2.5
□	qgraph	Graph Plotting Methods, Psychometric Data Visualization and Graphical Model Estimation	1.5
□	quantreg	Quantile Regression	5.36
□	R6	Classes with Reference Semantics	2.2.2
□	RColorBrewer	ColorBrewer Palettes	1.1-2

The box on the left of the package to be uploaded to the session is ticked.

Figure 3 Upload of "Psych" Package in RStudio

A sample Analysis Application by RSudio: Covariance Analysis

Within the scope of the research, One-way ANCOVA (covariance analysis) application, which is one of the statistical methods with many assumptions, was carried out to be an example in terms of seeing the capabilities of the R program more clearly by the researchers.

One-way ANCOVA (covariance analysis) can be considered as extending one-way variance analysis to include a covariate (Büyüköztürk, 2011). Like one-way ANOVA, one-way ANCOVA is used to determine whether there is a statistically significant difference in a dependent variable between two or more independent (unrelated) groups. However, ANOVA looks for differences in group means; whereas ANCOVA is investigating whether there are differences in adjusted means. Therefore, compared to one-way ANOVA, one-way ANCOVA has the advantage of allowing you to "statistically control" a third variable (commonly known as a covariate or sometimes a "confounding variable"), which you believe will affect your results. You can include the third variable, which you think may affect your results, in your one-way ANCOVA analysis by defining the third variable as a covariate.

When you select to analyze your data using One-Way ANCOVA, you need to check whether your data meets the assumptions of this analysis. In order to achieve highly valid outcomes via One-Way ANCOVA, eight assumptions should be met. The assumptions to be met for the one-way ANCOVA are given below (Büyüköztürk, 2011; Tabachnick ve Fidell, 2013):

Assumption 1: The dependent variable and the covariate should be continuous.

Assumption 2: The independent variable should consist of two or more independent groups.

Assumption 3: The independence of the observations should be ensured.

Assumption 4: Extreme points belonging to the dependent variable should not be numerous.

Assumption 5: Residuals/errors should show a near-normal distribution for each category of the independent variable.

Assumption 6: Homogeneity of the variances should be ensured.

Assumption 7: The covariate should have a linear relationship with the dependent variable for each category of the independent variable.

Assumption 8: The homogeneity of Regression curves should be ensured (there should be no mutual interaction between the covariate and the independent variable).

Application

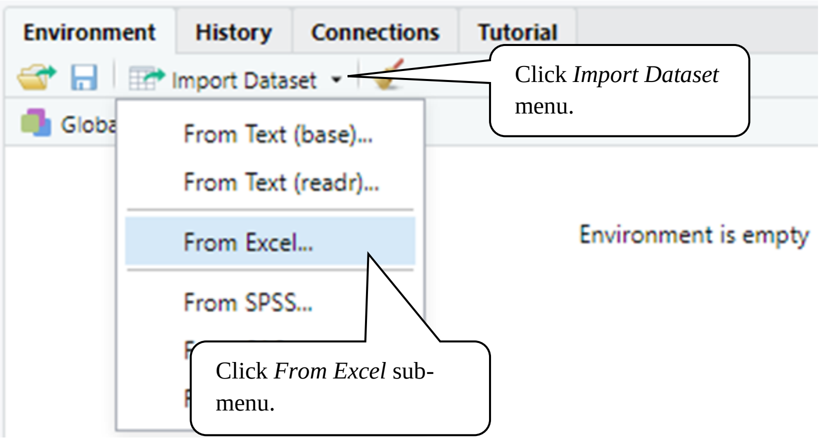
Within the scope of the application, ANCOVA was used to determine whether there was a significant difference between the post-test scores of the student groups when pre-test scores were under control, based on the data of the pre-test and post-test scores obtained from the statistics test administered to a total of 30 students. In this case, the problem statement can be written as “What can be said about the difference between the post-test scores of the student groups when pre-test scores were under control?” Zero and alternative hypotheses were constructed as follows:

H_0: There was no statistically significant difference in the experimental process, between the post-test scores of the groups when the pre-test scores were under control.

H_1: There was a statistically significant difference in the experimental process, between the post-test scores of the groups when the pre-test scores were under control.

Before starting the analysis, the assumptions related to the analysis were tested and the results obtained from the tests of these assumptions are given below.

The data set should be imported into RStudio before testing the assumptions. For ANCOVA, the data set with Excel extension is imported to RStudio by following the path Import Dataset> From Excel. Similarly, SPSS files can be imported to RStudio by following the path Import Dataset >From SPSS. Figure 4 shows how to import the dataset to RStudio.



Click *Import Dataset* menu.

Click *From Excel* sub-menu.

Figure 4 Importing ANCOVA Dataset to RStudio

After importing the data set into RStudio, the data set of ANCOVA will be displayed on the screen. The data set is saved as an object in R Environment with the same name (ANCOVA) after being imported into RStudio. After saving the data set, which contains the group variable of the students - experimental or control group - and the data of pre-test and post-test scores, as an object in R environment, the file is added to RStudio by using attach function and attach (ANCOVA) command. There are three variables in the data set: the group, pre_score and post_score. Within the scope of the sample, post_score will be included into the analysis as dependent variable, group as independent variable, and pre_score as covariate. After the data set is imported and added to RStudio, the assumptions are tested.

Assumption Control

Assumption 1 and 2: The dependent variable (post-test scores), and covariate (pre-test scores, are continuous.

Assumption 2: The group variable consists of two independent groups (experimental group and control group).

Assumption 3: Considering that each individual belongs to only one group, it can be said that the independence of observations assumption is met.

Assumption 4: The extreme values of the dependent variable should be of negligible quantity and magnitude. Extreme values in the dependent variable can be identified using the box plot, which helps to summarize the data visually. The Boxplot function is in the graphics package, located in the system library under the Packages tab. In order to use the Boxplot function, firstly the graphics package under the system library should be ticked and installed on the system. After the package is installed, write boxplot (post_score) to the console and click ENTER. Figure 5 shows the box graph for the extreme values of the dependent variable. The figure indicates that there is no extreme value.

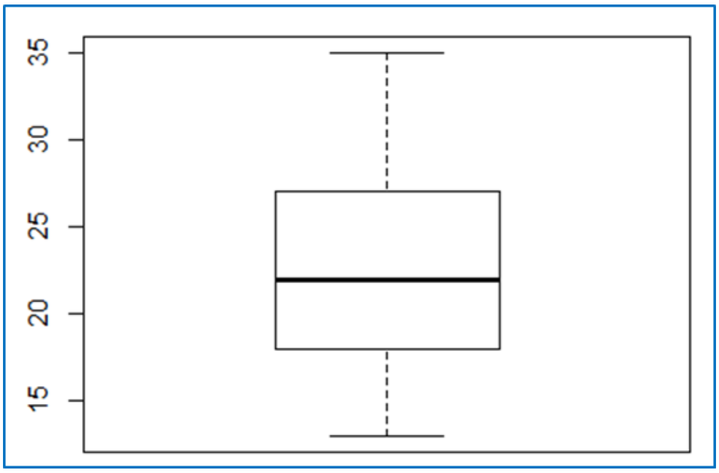
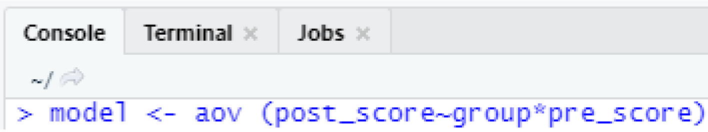


Figure 5 Box Graph of the Post-Test Scores

Assumption 5: Regarding the test of the assumption that residuals/errors should show a near-normal distribution for the established model, three different objects are created; one for the model, then another one for residuals/errors, and the last one for the standardized values of residuals/errors and finally, the hypothesis is tested using the shapiro wilk test.

The model of covariance analysis is formed by "aov" function, which is also used in variance analysis. The aov function is included in the "Stats" package, which is located in the system library under the Packages tab. In order to use the function, Stats package should be ticked and installed on the system. After the package is installed, write the script to the console and click ENTER. The script that needs to be written to the console to create the object called "model" is shown in Figure 9.

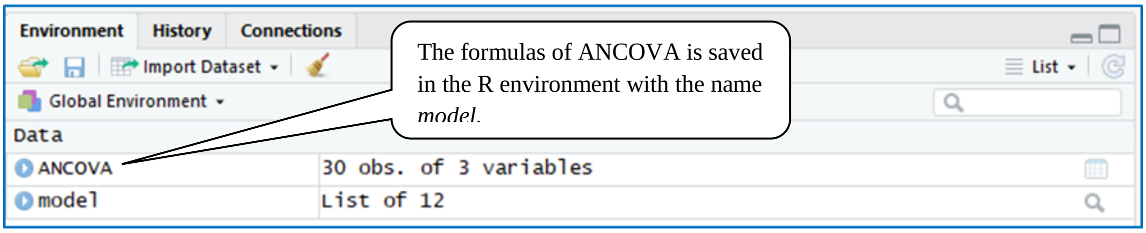


```
> model <- aov (post_score~group*pre_score)
```

Figure 6 Script for the Model Object

In Figure 6, aov is the function used and the variables in the parenthesis are dependent variable (post_score), independent variable (group) and covariate (pre_score)

When performing covariance analysis, the order of variable types should not be violated. Otherwise, incorrect and false analysis results are obtained. Figure 7 provides the visual representation of the object named "model" after being saved to R Environment.

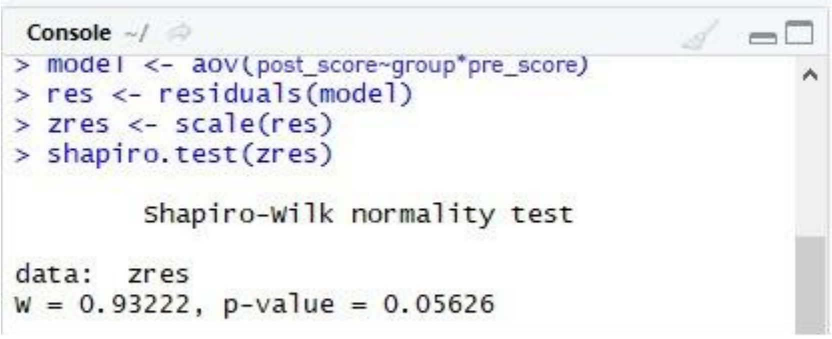


```
ANCOVA     30 obs. of 3 variables
model      List of 12
```

The formulas of ANCOVA is saved in the R environment with the name *model.*

Figure 7 Saving the Object Named "Model" to R Environment

After the model for ANCOVA is saved in R Environment by using aov function, the created model should be saved in R Environment as an object using the "residuals" function. Residuals function is included in the stats package, which is located in the system library under the Packages tab. In order to use the function, Stats package should be ticked and installed on the system. Let's call the object that we will save using Residuals function as "res". After the package is installed, write res <- residuals(model) to the console and click ENTER. Here, "res" is the name of the object to be saved, residuals is the function used, the model is the object created and saved in R Environment in the previous step. After creating the object called "res" for residuals/errors in the model, the third step is converting Residuals/errors into the standardized form, using the scale function included in the scale package. In order to use Relevant function, the scale package is downloaded and installed following the download and installation steps. After the package is downloaded and installed, write zres <- scale(res) to the console and click ENTER. Here, "zres" is the name of the new object, scale is the function used, res is the object saved in the previous step, using residuals function. After residuals/errors are standardized and saved in R Environment under the name zres, the assumption can be tested using the shapiro wilk test to see if residues/errors show a near-normal distribution. The shapiro.test function included in the stats package can be used for this purpose. Stats package is located in the system library under the Packages tab. In order to use the function, Stats package should be ticked and installed on the system. After the package is installed, write shapiro.test(zres) to the console and click ENTER. Here, shapiro.test is the function used, "zres" is the latest object created for residuals/errors and saved in R Environment. The normal distribution test results for residuals/errors are shown in Figure 8.

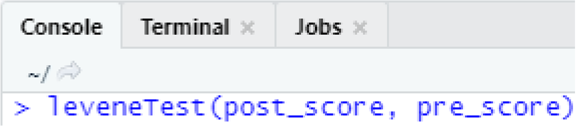


```
> model <- aov(post_score~group*pre_score)
> res <- residuals(model)
> zres <- scale(res)
> shapiro.test(zres)

        Shapiro-Wilk normality test

data:  zres
W = 0.93222, p-value = 0.05626
```

Figure 8 Normal Distribution Test Results for Residuals/Errors

Regarding Results of the analysis, it was concluded that Residuals showed normal distribution (W=0,93; p>.05).

Assumption 6: To test whether the homogeneity of variances assumption is met, leveneTest function included in the “car” package can be used. This function is included in the package named car. To use the mentioned function, the car package is downloaded and installed using the previously mentioned download and installation steps. After the package is downloaded and installed, write the script to the console and click ENTER. The script that needs to be written to the console is shown in Figure 9.

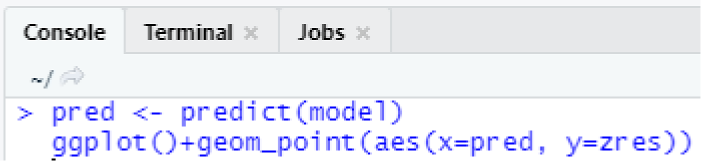


```
> leveneTest(post_score, pre_score)
```

Figure 9 Script for Levene Test

Regarding the results of the analysis, it was concluded that Residuals showed normal distribution (F(12,17)=1.53; p>.05). In other words, the variances show a homogeneous distribution in factor groups.

Assumption 7: To test the assumption that the covariate should have a linear relationship with the dependent variable for each category of the independent variable, the data of predictive values and residual/error values are needed. While testing the assumption of normality, the data of residual/error values was saved in the R Environment under the name of res object. The predict function included in the stats package can be used to obtain data of the predictive values. To use this function, Stats package should be ticked and installed on the system. After the package is installed, write pred <- predict(model) to the console and click ENTER. Here, “pred” is the name of the new object saved in R Environment, predict is the function used, model is the object that was previously saved in R environment. After the data of predictive values is saved as an object in R Environment under the name of pred using predict function, geom_point and aes functions included in the ggplot2 package can be used to obtain an image of the predictive and residual/error values for testing the assumption. To use the mentioned functions, ggplot2 package is downloaded and installed using the previously mentioned download and installation steps. Afterwards, write ggplot()+geom_point(aes(x=pred, y=zres)) to the console and click ENTER. Here, ggplot () is used to set the data frame of the entry for the chart to be drawn, geom_point function for the creation of scatter graph and aes function is used to indicate the mapping of variables. Pred in brackets is the object that forms the x-axis of the graph created under relevant assumption, whereas zres is the object that was formed for testing the normality assumption and which is related to residuals/errors that form the y axis of the graph. The scripts for checking the linearity are shown in Figure 10.



```
> pred <- predict(model)
  ggplot()+geom_point(aes(x=pred, y=zres))
```

Figure 10 Scripts for Linearity

The analysis results are shown in Figure 11.

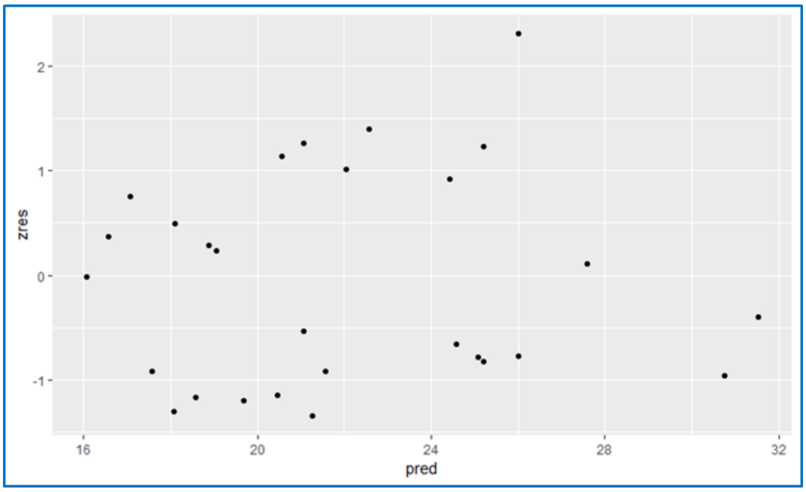
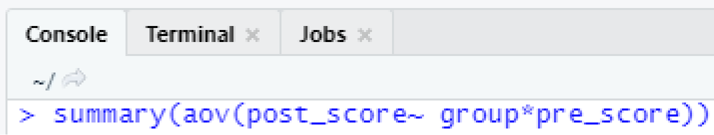


Figure 11 Linearity of Errors Test Results

To meet the assumption, the points in Figure 11 should be uniformly located. In general, the distances between the points appear to be relatively high. In short, we can say that there are clusters at different points.

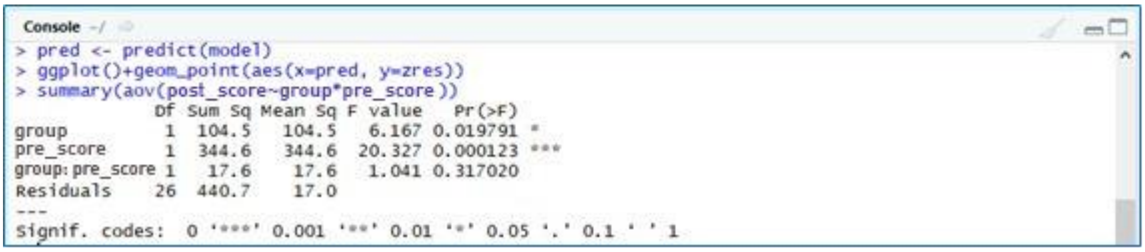
Assumption 8: Summary function can be used to see if the assumption involving the homogeneity of regression curves (there should be no mutual interaction between the covariate and the independent variable) is met. Summary function is used to obtain detailed outputs for analysis. The script that needs to be written to test the relevant assumption is in Figure 12.



```
> summary(aov(post_score~ group*pre_score))
```

Figure 12 Script for Homogeneity of Regression Curves

In Figure 12, summary is the function used to obtain detailed outputs for analysis, aov is the main function for the modeling of covariance analysis, post_score is dependent variable, group is independent variable, and pre_score is covariate. The output in Figure 13 is also the output screen displaying the analysis results of ANCOVA.



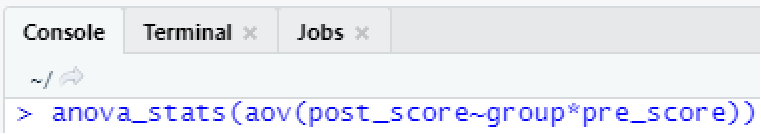
```
> pred <- predict(model)
> ggplot()+geom_point(aes(x=pred, y=zres))
> summary(aov(post_score~group*pre_score))
                Df Sum Sq Mean Sq F value   Pr(>F)    
group            1  104.5   104.5   6.167 0.019791 *  
pre_score        1  344.6   344.6  20.327 0.000123 ***
group:pre_score  1   17.6    17.6   1.041 0.317020    
Residuals       26  440.7    17.0                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 13 ANCOVA Results I

As mentioned above, Figure 13 is also the detailed analysis table of ANCOVA. For this reason, in order to determine whether there are statistically significant differences in the post-test scores of the two different groups we have covered in the study, we have to check the value of p in Row of the group variable. If this value is greater than .05, it can be concluded that there is no statistically significant difference between the groups. There is a single star next to this value in the first row of the table, which means that the obtained p value is statistically significant at .05 level. As a result, since the p value is less than .05, the hypothesis H0 is rejected; in other words, the hypothesis H1 is accepted. Accordingly, in this model where pre-test scores were determined as covariate, it was concluded that there is a statistically significant difference between the scores of the students in the experimental and control groups from the post-test of the statistics course. That is, there is a statistically significant difference between the post-test scores of the groups when the pre-test scores are under control.

In the given table, you should check the significance value (p) in the Row containing the variable “group: pre_score”, which signifies the interaction between group variable and pretest scores. If this value is less than or equal to .05, it means that the interaction between the pretest results and the group variable is statistically significant and therefore it can be concluded that the assumption has been violated. In this case, it is understood that there is a statistically significant relationship between experimental intervention and pre-test scores taken as covariate and therefore the assumption is violated. In such cases, researchers are not expected to obtain a significant value. In the given example, the p value was calculated as .317 and it was found that it is above the .05 accepted as the cut-off point. Therefore, it was concluded that the assumption of homogeneity of Regression curves was not violated.

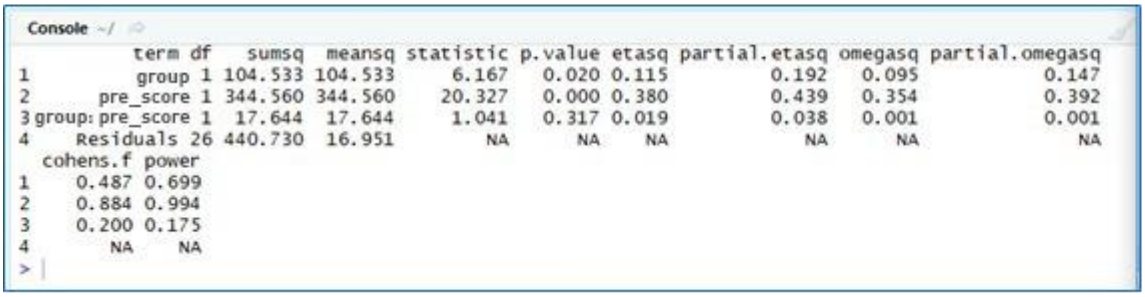
In addition to these, effect size values should also be checked in order to ensure the statistically significant difference in covariance analysis. The “anova_stats” function, which is one of the most popular functions used to calculate effect size, is included in the sjstats package of RStudio. To use the mentioned function, the sjstats package is downloaded and installed using the previously mentioned download and installation steps. The script that needs to be written to have the “effect sizes” is in Figure 14.



```
> anova_stats(aov(post_score~group*pre_score))
```

Figure 14 Script for Effect Sizes

Results of the analysis of the effect size are shown in Figure 15.



```
              term df   sumsq  meansq statistic p.value etasq partial.etasq omegasq partial.omegasq
1            group  1 104.533 104.533     6.167   0.020 0.115         0.192   0.095           0.147
2        pre_score  1 344.560 344.560    20.327   0.000 0.380         0.439   0.354           0.392
3 group:pre_score  1  17.644  17.644     1.041   0.317 0.019         0.038   0.001           0.001
4        Residuals 26 440.730  16.951        NA      NA    NA            NA      NA              NA
  cohens.f power
1    0.487 0.699
2    0.884 0.994
3    0.200 0.175
4       NA    NA
> 
```

Figure 15 Effect Sizes for ANCOVA

DISCUSSION, CONCLUSION AND SUGGESTIONS

Within the scope of the research, an exemplary ANCOVA application was carried out in order to promote the R program, in which the assumptions before the analysis were tested. Within the scope of sample application realized with the R program, the functions included in different packages were used. It was concluded that the scripts created by the functions included in the packages effectively incorporate the researcher into the analysis process and thus, the lack of theoretical knowledge that can be found in the researcher's mind about the analysis can easily be revealed. It is concluded that coding information is not needed while analysis is done through scripts. In addition, it was concluded that the program gives detailed information about what the error is if the researcher made a mistake during the application. While performing analysis with the R program, it was concluded that information can be obtained by using the program interface regarding how the functions are used. At the same time, it is concluded that the analysis can be obtained by creating scripts containing all the steps related to the analysis. Thus, many different ideas about the analysis can be tried quickly, if there is an error, the location of the error can be quickly determined and the analysis can be updated. Although not carried out within the scope of the analysis made in the research, most steps related to the analysis can be done with different functions. After the analysis process is over, the text written in RStudio can be converted into high quality documents, reports, presentations and panels with R Markdown. Thus, the scripts, results and explanation of the analysis performed with R can be done in a single document and this document can be translated into popular formats such as HTML, PDF or Microsoft Word upon request.

One of the prominent aspects of the R program is that the R software language, which is an open source project, is free and is continually developed by a team working in this field. There are no license restrictions for the program. The program can run smoothly on almost all operating systems.

Even researchers who are not familiar with coding can easily perform their analysis in RStudio. In addition, Researchers can suggest codes or functions to develop a new method for any analysis and, if accepted, share their code with about 2 million R users. In this way, a researcher can contribute both to himself and to researchers using R software language in the world. The program can easily work on very large and very complex datasets. The user is not limited to a single method of analysis because there are many packages for any analysis. At the same time, R has excellent tools to easily visualize and draw graphs of complex analysis results. Many users think of R as a statistics system, but environment is the correct word that describes it. R is an environment in which statistical formulas and techniques can be applied and developed by any user in time. R has its own LaTex-like documentation format, which is used to create comprehensive documentation online in various formats and as printed copies.

Besides the advantages of the program, there are some disadvantages. Within the scope of R program, the analyse can be performed step by step, by the functions written on the command line by the user and not by pop-up menus. In order for users to use R correctly, they have to learn the data types that correspond to different classes. At the same time, the user should be able to think critically about the use of the program. The user should learn commands and representations specific to different statistical models. Since R is an open source program and many people contribute to its development, there is no authority to complain about in case of any problem. R is not as popular as Microsoft Excel. Users have a prejudice that using R is more difficult than other statistical programs. However, any user who uses R software for a short time can easily control the program and get many advantages that the program brings. Another drawback of the program is that the lack of spreadsheet view of the data sets. It is striking that the R programming language is slower compared to other programming languages such as MATLAB and Python.

In studies where R and similar open source software are used, it will be appropriate to report the codes used, share them with other researchers and test the correctness of the analysis process. It is safer for the researcher to use well-known packages / libraries when an analysis is to be carried out using the R program. Because, well-known libraries such as SciPy and NumPy generally contain well-managed programmers who examine the codes very well and respond quickly to problems. The large user base of these packages ensures that open bugs are pre-captured, reported and corrected.

Although there has been an increase in the number of Turkish publications related to the use of R program in recent years, the number of publications is still insufficient. At this point, it can be said that the publications related to the use of R program should increase. In the field of science, there are courses in various departments (statistics, engineering, etc.) related to the introduction and use of R program. In the field of educational sciences, courses related to the introduction and use of R program should be opened, especially at postgraduate level. In the academic world, researchers often use non-original versions of these programs instead of purchasing paid statistical programs, which causes the violation of the code of ethics. Considering that R program is free, such an ethical violation can be prevented by the use of R program.

REFERENCES

Aydın, B., Algina, J., Leite, W., & Atılgan, H. (2018). An R companion: A compact introduction for social scientists. Ankara: Anı Publishing.

Büyüköztürk, Ş. (2011). Sosyal bilimler için veri analizi el kitabı-İstatistik, araştırma deseni, SPSS uygulamaları ve yorum(15. Baskı). Ankara: Pegem Akademi.

Campbell, M. (2019). Learn RStudio IDE. New York, NY: Apress.

Chambers, J. (2008). Software for data Analysis: Programming with R. New York, NY: Springer.

Cirillo, A. (2016). RStudio for R statistical computing cookbook. Birmingham, England: PACKT Publishing.

Crawley, M. J. (2007). The R book. Chichester, West Sussex, England: Wiley.

Eser, M. T., Aksu, G., & Güzeller, C. O. (2019). R Programlama Dili İle Temel İstatistikler ve Raporlama. Ankara: Pegem Akademi

Knell, J. R. (2014). Introductory R. United Kingdom: Hersham, Walton on Thames.

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