

ALY2010 Project 2 Assignment

Instructor: Prof. Dee Chiliza, PhD

Due Date: Tuesday February 15 at 11:59 PM.

Grade: 50 points

For this project, you will perform an initial analysis of the data set wine, and you will use R Markdown to prepare and present your report in a HTML document.

Submit: (1) your original R Markdown file and (2) your HTML report.

Before you start this assignment:

1. Download the data set **"wine(ALY2010).xlsx"** and save it inside your computer's folder: ALY2010 R Project/DataSets.
2. From R Studio, create a new **R Markdown** file inside your class R project file. Select **HTML** as the output format.

Task 1.

Use the code `dplyr::glimpse()` to obtain information about your data set.

- a. Explain the information obtained with that code.

Task 2

Using the codes `nrow()` and `ncol()`, create objects to save the number of columns (variables) and number of rows (observations) your data set contains.

Using Inline R codes, present these values.

Make sure to use two back quotes to ensure that your values are highlighted on your report.

e.g., ``r variables``

Note: this is the back quote key:



Task 3

Using the code `names()`, make a list of all variables in the data set, and present it as a data frame.

Task 4

Present the number of categories contained in variable wine type.

For this, use the code `table()` as explained in class, but do not present the table (save the table with an object name).

Use the table to present the categories of wine type using a bar plot.

Provide different colors to each bar.

Use code `text()`, as indicated in my website, to add the frequencies on top of each bar.

Explain the codes you used and the data you obtained on your graph.

Task 5.

For each category of wine type, ask a simple question:

What is the mean color intensity per wine type?

To answer that question, create an object named `mean_color` (or any other name of your preference) and use the `tapply()` code to answer that question. This code allows you to apply a function, like mean, median, to subsets of a variable. (<https://youtu.be/9ZWHfozPn6k>)

Remember to add the numerical Variable name first, then the categorical Variable name, and finally indicate the function(mean, sum, sd).

Do not present the outcome of the `tapply()` code, using the object you created, present the values in a `carplot()`.

Using `color =` , add different colors to each category.

Using code `text()`, add the mean values to each variable.

Since this is the first time you will do this, I will help you. See `tapply()` at the end of this document.

Task 6.

Similar to task 5, and using the `tapply()` code, create four objects to store the following data:

- The mean phenols per wine type.
- The standard deviation of phenols per wine type.
- The median phenols per wine type.
- The variance of phenols per wine type.

Observe the file "Vectors and Matrices R" in canvas, use it as a guide to present all the values using a table (first create a matrix, add column names, row names, then present the matrix).

Clue: Create three vectors: vector 1 contains the names of all four objects, vector 2 contains the names for the columns (wine types), and vector 3 contains the names of statistics (mean, sd, median, variance).

Create a matrix using vector 1.

Use vector 2 to add column names using code `colnames()`.

Use vector 3 to add row names using code `rownames()`.

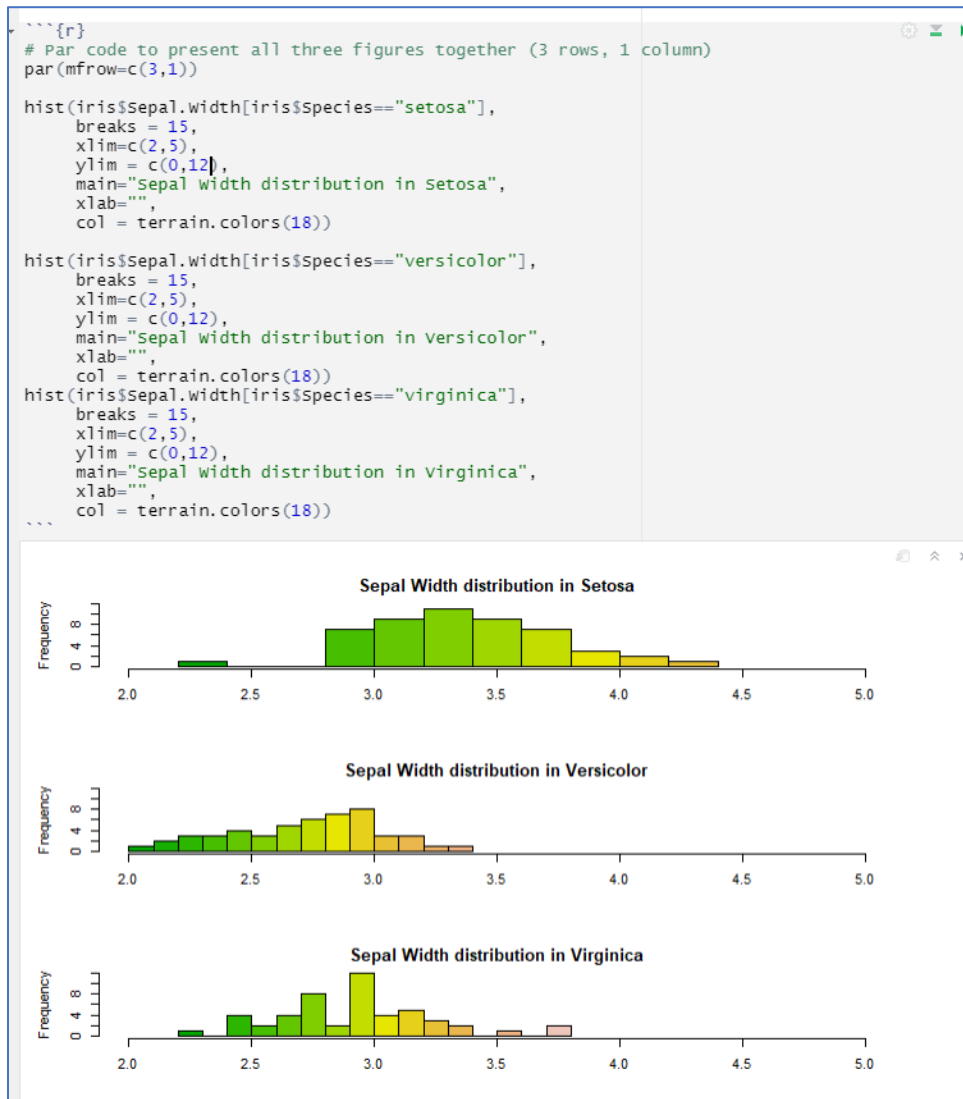
All these codes and their use are explained in the R file mentioned above.

Provide the Matrix with a name and be sure to present only the final product on your report.

Task 7

Now you want to know the frequency distribution of each continuous variable per wine type. For example: what is the distribution of sepal width per species in the data set iris?

One way to accomplish this task it by using histograms. In the x-axis you observe the sepal width distribution presented as bins, and in the y-axis you observe the frequency of observations per bin. I will help you with these codes.



Copy and save those codes on your support files, this will help you to practice and learn.

Note 1: I entered the codes for the three histograms on the same R chunk.

Note 2: I used code `par(mfrow=c(3,1))`. What this code does is the following: it is telling R that you want to present the figures using 3 rows and 1 column.

Note 3: To plot a histogram with a numerical variable and only one of the groups inside any categorical variable, you can use the code square brackets to specify the group you want to plot, for example, I want to plot sepal width only for the species setosa:

`hist(iris$Sepal.Width[iris$Species=="setosa"])`.

Note 4: When I first ran the histograms, I did not specify x-axes limits, and therefore the figures contained different dimensions. Remember what I mentioned in class, to make proper comparisons of data using different figures, the values must be presented using the same magnitude. In this case, I noticed what was the minimum value and the maximum value among the 3 graphs, and I used them to enter a common `xlim=c()` code. Observe above.

I did the same with the y-axis limits, notice that all three have the same `ylim=c(0,12)` code. Remember to first run your histograms without x- or y-axes limit codes, then decide what is the

best range for your data visualization.

Now that you know how to do create this figure, do the same for our data set wine.

Task: Present a figure containing three histograms to display the distribution of **magnesium** per wine type. Remember to change the number of bins (breaks) add proper x and y limits, x and Y axes labels, and colors to your figure.

Remember to make meaningful observations of your data results after each task.

Task 8

Present one box plot figure to display the distribution of **magnesium** per wine type. In this case, you only need one `boxplot()` code and inside indicate the numerical and categorical variables separated by a wavy dash `~`.

Remember to add a professional presentation to your graph.

Remember to make meaningful observations of your data results after each task.

Task 9

Compare the data results you obtained on task 7 (histogram) and task 8 (box plot).

What conclusions you can draw for magnesium based on its distribution per wine type?

Task 10. Conclusions

Of all the graphs you produced above, which one called most your attention? Choose at least 2 graphs, search on internet the meaning of that variable in terms of wine quality and make a short analysis of what you consider the quality of each wine type depending on the graphs you choose.

Also, make an overall observation of the whole project, the meaning of the results you obtained regarding the direction of the project, explain any new skills you gained.

Also, imagine you are preparing this report for a company or research institution, therefore, you must make meaningful contributions, think about what recommendations you can provide to other people using this data set.

Task 11. Bibliography.

Present all reading sources you used to support your project.

References must be used on the main body of your report: Technically speaking, if you do not mention any references in the main text of your report, then it is like you did not use any, even if you add a list at the end.

Present references in the main body of your reports in the place where you use them as an information source, use either only the first author's last name and year, e.g., (Bluman, 2017) and then list them in the bibliography section in alphabetical order, or use a number in order of appearance (1), (2), or (1,2), etc., then list them in the bibliography section in that numerical order.

Additional information

How to apply code: `tapply()`

If you run `tapply` to ask the standard deviation of petal width per species in the data set iris:

```
##{r}
object1 = tapply(iris$Petal.width, iris$Species, sd)
object1
```

	setosa	versicolor	virginica
sd	0.1053856	0.1977527	0.2746501

I created an object named `object1` and I activated. I need to subtract the actual sd values if I want to use them on the bar plot. To do this I transform the object into a matrix. Look how `object2` looks like:

```
##{r}
object1 = tapply(iris$Petal.width, iris$Species, sd)
object2 = matrix(object1)
object2
```

	[,1]
[1,]	0.1053856
[2,]	0.1977527
[3,]	0.2746501

From `object2` I need to extract all values in the column, to do this I can use square brackets.

```
##{r}
object1 = tapply(iris$Petal.width, iris$Species, sd)
object2 = matrix(object1)
object3 = object2[1:3]
object3
```

[1]	0.1053856	0.1977527	0.2746501
-----	-----------	-----------	-----------

Now, I can use `object1` to create my bar plot, and `object3` to add the standard deviation values. In the graph below, observe and analyze carefully the codes. Also notice that this is an R chunk and that the only information presented is the graph, I do not present the outcomes of each object because the graph is enough.

```

```{r}
object1 = tapply(iris$Petal.width, iris$Species, sd)
object2 = matrix(object1)
object3 = object2[1:3]

graph1 = barplot(object1,
 ylim=c(0,0.4),
 col=brewer.pal(3, "Set1"),
 las=1,
 ylab="Standard Deviation")
text(y=object3,
 graph1,
 round(object3, 3),
 pos = 3)
```

```

