Lab Session 5 (Week 14)

2.12 Before you proceed for this lab session

Before we get started - in order to use your lab time more efficiently, make sure you complete the following steps first:

1. Open RStudio

2. Create a new Project. **File > New Project > New Directory > New Project**. Then, click on **Browse** and select your **Desktop**, then for **Directory Name** type in ‘Week 14 Lab’. Then click on Create Project.

3. On moodle, on the PSYC3000 page, go to Week 14. In the lab section there are two files, **internet.csv** and **exercisew7.csv**. Download both files and save them or copy them in your **Week 14 Lab** folder that is on your desktop. **IF YOU ARE A MAC USER PLEASE USE CHROME AND NOT SAFARI.**

4. If you have done everything right, you should be able to see both files at the bottom right window in your RStudio, in the tab **Files**.

5. Create a new script and save it as **week14**. Then type in the following commands and run them in order to load the **tidyverse**, **correlation**, and **ggpubr** packages.

```
library(tidyverse)
library(correlation)
library(ggpubr)
```

6. If you get any **warning messages** that is fine.

7. If you get an error for a specific package, then you probably have not installed it yet. You can install it using the **install.packages()** function by running `install.packages("packagename")` in your console.

2.13 Learning objectives

- **Load our own data files in RStudio**
- **Use the mutate() function to add new variables into the dataset**
- **Create a scatterplot for our variables**
- **Run correlation analysis in order to explore associations**
- **Create new dataframes that will only contain the variables we want**
- **Run multiple correlations in one step**

2.14 Load our own data

If you have followed the steps above, you should already have two data files in your folder. These are **internet.csv** and **exercisew7.csv**. We will execute the following script to load **internet.csv** and also get some useful initial summaries. The **internet.csv** contains data from a study that measured participants Internet use. If you have already created the **week14 lab** script above then just continue working from there. If not make sure you add the three lines above that contain the libraries.

```
## load the file internet.csv and assign to a dataframe called df
df <- read_csv("internet.csv")

## Let's have a quick look at our variables in df
summary(df)
```

2.15 Working with our data

The id variable corresponds to participant id, the variable gender and age are self-explanatory. The variables **use1** to **use6** represent 6 questions with regards to the use of internet. The variables **pre1** to **pre6** represent 6 questions with regards to how preoccupied with thoughts about Internet someone is. The variables **comfort1** and **comfort2** are two questions regarding how comforting someone finds the Internet to be. **Days** and **hours** represent how many days a week and how many hours a day someone spends on the Internet.

Even though we could run correlation analysis for each question, this is not particularly meaningful or helpful in this case. **What we want to do is calculate three new variables:**

- One variable that will be the average use of internet (mean of **use1** to **use6**)
- One variable that would be average preoccupation (mean of **pre1** to **pre6**)

- One variable that would be the average of comforting, mean of **comfort1** and **comfort2**.

So, in reality we want to add three new columns to our dataframe that will include the results of the following three calculations:

- **use_mean**: mean of use1 to use6 for each participant
- **pre_mean**: mean of pre1 to pre6 for each participant
- **comf_mean**: mean of comfort1 and comfort2 for each participant.
- **total_hr**: the total amount of hours in a week

We can achieve the above with many different ways, for this term we will focus on using the function **mutate()**.

```
## when using mutate, we have to declare the dataframe first
## then give the name of the new variable
## and then type in the calculation we want to do
## the mean is nothing else than adding all variables
## and then dividing by the number of variables
## NOTE: you do not have to break the mutate line in two lines
## like below, we did that so it fits in the pdf width

df <- mutate(df,
             use_mean = (use1 + use2 + use3 + use4 + use5 + use6)/6)



df <- mutate(df,
             pre_mean = (pre1 + pre2 + pre3 + pre4 + pre5 + pre6)/6)



df <- mutate(df,
             comf_mean = (comfort1 + comfort2)/2)

## we can also do any other type of arithmetic calculation
## for example: total_hr = days * hours
df <- mutate(df,
             total_hr = days*hours)
```

We can now create simple scatterplots between between our newly created variables using **ggscatter()**. Below, we have given you one example for creating your scatterplots, try to complete the rest on your own. We can also carry out a correlation analysis between two variables using **correlation()**. Again, we give you only one example, you should do the rest on your own during the lab session.

```
## a simple scatterplot between use_mean and pre_mean
ggscatter(df, x = "use_mean", y = "pre_mean")

## a simple correlation analysis between
cor_test(data = df,
         x = "use_mean", y = "pre_mean", method = "pearson")

## change the code as needed to see output for different pairs of
## variables. You should have all pairs of Use_mean, pre_mean, comf_mean, ##  and total_
```

2.16 Creating a new dataset and executing all correlations at once

As you have seen above, it is quite a lot of work to keep typing the same code for each pair of variables. One way to overcome this is to use the function, **correlation()**. This function takes our entire dataframe as its input and runs correlations for **all** the existing variables. The problem with this approach is that our dataframe may also have variables that we do not want to include. An easy way to fix this would be to create a new dataframe that only contains the variables we need. We can then run our correlation on the new dataframe. We can do that by using the function **select()** and declare the variables we want to copy to the new dataframe.

As you should recall, when we loading in our original data, we assigned it to a dataframe and called it **df**. So, we now want to create a new dataframe and call it something else, such as **df2**.

```
## When using the select() function,
##we first declare that the dataframe we want is df
## then we declare the variables we want to copy over
## to the new dataframe which we can give any name we want.
df2 <- select(df, use_mean, pre_mean, comf_mean, total_hr)

summary(df2)
```

Running the above code would create a new dataframe called **df2**. We can now use **correlation()** to run all the correlations between our four variables.

```
## all we have to do is declare the dataframe name
## in the correlation function

correlation(data = df2, method = "pearson")
```

If you observe the output, you will see that we have a row for each pair of variables.

2.17 Exercise

Create a new script in your week 14 project. Then, write the appropriate code that would allow you to load the datafile **exercisew7.csv**. This file contains the following columns: att1, att2, att3, bel1, bel2, bel3, bel4, stance1, stance2, stance3, stance4, ethic1, ethic2, ethic3. You should create new variables that would be the average of the following:

1. **att_mean**: att1, att2, att3
2. **bel_mean**: bel1, bel2, bel3, bel4
3. **stance_mean**: stance1,stance2,stance3,stance4
4. **ethic_mean**: ethic1, ethic2, ethic3

Using what you have learnt from today’s session, complete the following activities:

- **Exercise 1:** Explore the correlations between the above four variables by creating scatterplots.

- **Exercise 2:** Next, create a new datafile that will **only** include these four variables and run **all correlations at once** using the new dataframe.