Practical Exercise 0 - Introduction | Statistics for Premasters: DSS/CSAI

Ethel Pruss_u123456

All of your future practical exercises will be in this format. The zip file you downloaded includes three files: a rmd file, a pdf and a docx file. You only need to open the Rmarkdown file (.rmd) to complete the exercise. The word document needs to stay in the same folder as the Rmarkdown file but aside from that, you can ignore it. It is part of the knitting process which we'll get to a bit later. The pdf is simply another set of the same instructions seen here in a prettier format.

Part A - Getting Comfortable With Rmarkdown

First, take a look at the little section at the very top of this file. You'll see a little section between three dashes with some information. You only need to write your name and u number where it says "author" in the format of "your name_unumber". I have filled in my information as an example. Other than that, you don't need to change anything in the setup of this file.

To complete the practicals exercises, you will write your answers directly in this file where indicated with either "Your answer here" or "Your code here". You can delete the comment indicating where you should answer.

You will have two types of tasks:

Task 1. Questions with text-only answers

If you don't see a code block under a question and it says "your answer here", we expect a written answer with no code. Simply type your answer under the question like this.

Task 2. Coding questions

If you see a code block and a comment with "your code here", we expect an answer with code.

Executable code blocks are surrounded by "‘ on the bottom and top and have a header like {r task..}. Make sure you don't delete or change these elements, just write your code in the middle.

```
# Questions with code blocks like this need to be answered with code
# If you want to add some clarifying comments, add a # in front of them

print("Code goes here")
```

```
## [1] "Code goes here"
```

On the upper right side of the code block, you will see a little run button. When solving PEs, remember to press this to run your code and see if your results are correct. If the result don't make sense or you get an error, try correcting your code and running it again.

Part B - Loading Data

For most of your future exercises, you will need to import a dataset to work on. Ususally, this dataset is either given to you in a csv file (check the zip folder for your PE) or you can load it directly from a package or base R.

Task 3a. Loading data from R

```
# Loading the dataset called "iris", this is part of base R
# If your dataset is in an external package, load the package first as seen in Part C
data(iris)

# Let's check if it worked
# head() shows the first x rows of your data
# The second argument determines the number of rows
head(iris, 7)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
## 7          4.6         3.4          1.4         0.3  setosa
```

Task 3b. Loading data from a CSV file

```
# Loading the data
csv_data <- read.csv(paste(getwd(), "/expt.csv", sep=""), header = TRUE, sep = ",")
```

If you prefer to use the "Import Dataset" button on the right side to import your data from a file, notice that a line of code similar to the one above appears in your console. You need to copy and paste this line over here as seen above to avoid issues with knitting later on.

Task 3c. Try using head() to view the first 10 rows of csv_data

```
head(csv_data,10)
```

```
##    id age gender treatment hormone happy  sad
## 1   1  25   male   control     6.7  2.00 6.12
## 2   2  24   male     drug1    38.5  3.36 3.53
## 3   3  25   male     drug2    25.0  3.40 4.82
## 4   4  28   male   control    98.4  5.69 0.34
## 5   5  23   male     drug1    42.4  4.56 4.48
## 6   6  28   male     drug2    20.3  2.89 4.57
## 7   7  25 female   control    18.5  3.18 4.82
## 8   8  29 female     drug1    65.2  4.78 2.24
## 9   9  21 female     drug2    56.4  4.51 2.64
## 10 10  26 female   control    55.7  3.90 2.71
```

```
#Your code goes here
```

Part C - Installing and loading packages

Task 4. Installing and Loading Packages

Not all functions we use in this course are part of base R. If you want to use a function from an external package, you will first need to install it. This only needs to be done once.

Lsr, car and psych are three of the main packages you will need for your first practical exercises, let's install them now to get everything ready.

Task 4a. For installing packages, you can either use the "Packages" tab in your graphical interface or the line "install.packages('package')" either in your console or a code block like this. I have added some extra code in this case to make sure the package isn't already installed.

```
# require('lsr') returns "FALSE" if we don't have lsr
# The "!" in front means the if statement is only triggered if we get a "FALSE".
if (!require('lsr')) install.packages('lsr')
```

```
## Loading required package: lsr
```

```
# install.packages('lsr') is easier if you already know you don't have the package
```

Task 4b. Try installing the 'psych' and 'car' packages using install.packages() or the graphical interface.

```
if(!require('psych'))install.packages('psych')
```

```
## Loading required package: psych
```

```
install.packages('??car')
```

```
## Error in contrib.url(repos, "source"): trying to use CRAN without setting a mirror
```

```
# Your code goes here
```

Task 5. Loading packages

Task 5a. In addition to installing packages, we also need to load them every time we want to use them. We'll look at two ways of doing this. First, using the library function like this:

```
# We load the library first
library(lsr)

# Now we can use functions from the library
ciMean(c(1,2,5,6,3))
```

```
##           2.5%    97.5%
## [1,] 0.8252307 5.974769
```

You only need to use library() once per file for the same package and it needs to be in your .rmd file as seen above. If you use the console below the file or the graphical interface for loading libraries without copying the code over here, you will run into issues later on.

Task 5b. Another way is indicating the package needed for our function in the format of "package::function" like this:

```
lsr::ciMean(c(1,2,5,6,3))
```

```
##           2.5%    97.5%
## [1,] 0.8252307 5.974769
```

Task 6. Getting help

If you forget which package you need to load for a function, you can add two question marks in front of the function to search for it.

```
??bars
```

```
## starting httpd help server ... done
```

```
# Check the help box on the right side, you should see "lsr::bars" there
```

If you're not sure how a function works you can do the same with only one question mark.You will see some helpful information like the the expected input format for the function and the list of possible arguments.

```
?bi.bars
```

If you prefer, you can also get the same information using this website: https://www.rdocumentation.org/

Part D - Knitting

Note that the accepted submission formats for PEs on Canvas are pdf or docx. You only need to turn in this document for each PE, the rmd file is not needed. The document will include all of your code and answers from this file and we are going to generate it automatically using a process called knitting.

Before we get started, you will need to install a few prerequisites.

Before we do this, use ctrl + s or File -> Save to make sure you don't lose any progress. Then, run the code block below. Similarly to the packages we installed above, we only need to do this part once, so you don't need to worry about it for the following exercises if you've completed this one.

Task 7. Installing and loading prerequisites

For knitting to pdf, you will need to install a LaTeX distribution. The easiest way to do this is with the package 'tinytex'.

```
# 1. Installing the package tinytex
if (!require('tinytex')) install.packages('tinytex')
```

```
## Loading required package: tinytex
```

```
# 2. Installing the corresponding LaTeX distribution
tinytex::install_tinytex()
```

```
## Found 'C:\Users\mr\AppData\Roaming\TinyTeX\bin\windows\tlmgr.bat', which indicates a LaTeX distributi
```

```
## Error in tinytex::install_tinytex(): If you want to force installing TinyTeX anyway, use tinytex::ins
```

If you have trouble with knitting to pdf, you can also knit to docx. For this, you only need to have Microsoft Word installed. The Office 365 package is available for students here:

https://www.tilburguniversity.edu/students/it/office-365

Next, we need to install the packages 'rmarkdown' and 'knitr'.

```
if (!require('rmarkdown')) install.packages('rmarkdown')
```

```
## Loading required package: rmarkdown
```

```
if (!require('knitr')) install.packages('knitr')
```

```
## Loading required package: knitr
```

Let's load the packages we just installed to make sure everything went smoothly.

```
library(rmarkdown)
library(knitr)
```

Task 8. Knitting your document

After successfully installing and loading the packages, you should see a "Knit" button just above this document. If you hover over it, you will see a "Knit to PDF" (or "Knit to Word") option - click this.

That's it, you should see a document named "Practical-Exercise 0.pdf" (or .docx) in your PE folder now.

Task 9. Troubleshooting

If you had an error when trying to knit, take a look at "Error Solving Guide for PEs and Tutorials" under Modules on Canvas and see if you can find a solution there.

Here's a quick checklist:

1. Is the "reference_tasks.docx" still in the same folder as the .rmd file you're using?
2. Did you do anything using the graphical interface or the console? Make sure the corresponding code for everything you did is copy-pasted into the code chunks in this file.
3. Are all the packages you used loaded?