Summary of Day 1 and 2 of PSY9510

Stine Nygård

2022-09-27

> Note: On the course page (https://www.uio.no/studier/emner/sv/psykologi/PSY9510/index.html) it specifically says that students must submit "a reproducible Rmarkdown report" (i.e. the requirement is 1 report). I have therefore collected my notes from both lectures in a single document.

1 Getting started

> Note: R is *case sensitive* - be mindful of this when writing code

> Note: set working directory = `set.wd` , show working directory = `get.wd`

1.1 Get to know R functions

> A function is a set of statements that, when put together, perform a specific task

Function: `help`

> The `help` -function is used to look up help-files for specific functions. NOTE: May be **unhelpful**.

E.g. use `help` -function to read up on `help` -function

```
help ("help")
```

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

1.2 Install and load R packages to be used

> In R, packages are collections of functions, code, and data

Function: `install.packages`

> Use `install.packages` -function to install specified R package

> Install tidyverse-package using `install.packages("tidyverse")`

Function: `library`

> Use `library` -function to load R packages

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

2 Introduction to basic R functions

2.1 Display values

Function: `print`

> Use `print` -function to print values or variables

E.g. print "Hello world"

```
print ("Hello world")
```

```
## [1] "Hello world"
```

2.2 Assign values

Function: `assign`

> Use `assign` -function to assign values to a variable

E.g. assign value 1 to A

```
assign ("A", 1)
print ("A")
```

```
## [1] "A"
```

> Can also be done as follows: `"variable" <- (value)`

E.g. assign value 2 to b

```
"B" <- 2
print (B)
```

```
## [1] 2
```

Function: `combine`

> Use `c (value, value, value, etc.)` to create a *vector* (or "train of objects") to assign multiple values to a variable

E.g. assign values 3, 6, and 9 to C

```
assign ("C", value = c (3, 6, 9))
print(C)
```

```
## [1] 3 6 9
```

E.g. assign values red, green, blue to D

```
assign ("D", value = c ("red", "green", "blue"))
print(D)
```

```
## [1] "red"   "green" "blue"
```

> You can also create a *vector* of integers in increasing or decreasing order using colon ":"

E.g. assign values 1 to 10 to E

```
E <- (1:10)
print(E)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

> You can also assign *logical operators* as values to a variable

E.g. assign 3 TRUE and 3 FALSE as values to variable TF

```
TF <- c(TRUE, TRUE, TRUE, FALSE, FALSE, FALSE)
print (TF)
```

```
## [1]  TRUE  TRUE  TRUE FALSE FALSE FALSE
```

2.3 Examine objects (e.g. variables)

Function: `typeof`

> Use `typeof` -function to examine type or storage mode of any object

E.g. examine the type of the variables that you have created

```
typeof(A)
```

```
## [1] "double"
```

```
typeof(D)
```

```
## [1] "character"
```

```
typeof(TF)
```

```
## [1] "logical"
```

Function: `length`

> Use `length` -function to examine lenght (number of values) of variables

E.g. examine length of the variables you have created

```
length(B)
```

```
## [1] 1
```

```
length(C)
```

```
## [1] 3
```

```
length(E)
```

```
## [1] 10
```

Function: logical operators

> Use logical operators to compare values

- `<` less than
- `<=` less than or equal to
- `>` greater than
- `>=` greater than or equal to
- `==` exactly equal to
- `!=` not euqal to

E.g. check if the values of TF are 1, 1, 1, 0, 0, 0
Note: true is coded as 1, FALSE is coded as zero

```
TF == c(1,1,1,0,0,0)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE
```

2.4 Create data frame

Function: `data.frame`

> Use `data.frame` -function to create a data frame consisting of multiple variables

E.g. name, age, and gender of students

```
name <- c("Emily", "Julie", "Stine", "Eirik", "Nadine", "Sara", "Ole", "Anders", "Nicklas", "Fredrik", "Haghish")
age <- c(27, 25, 31, 26, 31, 22, 27, 37, 44, 45, NA)
gender <- c("female", "female", "female", "male", "female", "female", "male", "male", "male", "male", "male")
df <- data.frame(name, age, gender)
print(df)
```

```
##       name age gender
## 1    Emily  27 female
## 2    Julie  25 female
## 3    Stine  31 female
## 4    Eirik  26   male
## 5   Nadine  31 female
## 6     Sara  22 female
## 7      Ole  27   male
## 8   Anders  37   male
## 9  Nicklas  44   male
## 10 Fredrik  45   male
## 11 Haghish  NA   male
```

3 Data sets in R

3.1 General functions

Function: `ls`

> Use `ls` -function to list objects in global environment

```
ls()
```

```
##  [1] "A"      "age"    "B"      "C"      "D"      "df"     "E"      "gender"
##  [9] "name"   "TF"
```

Function: `saveRDS`

> Use `saveRDS` to save a single R object as a file (in working directory)

e.g. save df as file

```
saveRDS(df, "students")
```

Function: `readRDS`

> Use `readRDS` -function to restore R object from a file (in working directory)

Function: `unlink`

> Use `unlink` -function to delete file (from working directory)

```
unlink("students")
```

3.2 CSV files

Function: `read.csv`

> Use `read.csv` -function to read data from a CSV file

Function: `write.csv`

> Use `write.csv` -function to create a CSV file from R objects

3.3 Excel sheets

> Use functions in the `readxl` -package to import excel sheets into R Check out the tidyverse website for more information about the `readxl` -package

3.4 Stata

> Download and use the functions of the `readstata13` -package to import Stata files into R

3.5. SPSS

> Download and use the functions of the `foreign` -package to import SPSS files into R

Data from the Internet

> It’s possible to import data directly from the web - read up on this if it ever becomes relevant

4 Analyze data

4.1 Data prep/cleaning

Function: `is.na`

> Use `is.na` to check for/locate missing values

E.g. check for missing values in variable “station” in data set “attenu”

```
data("attenu")
is.na(attenu$station)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
##  [97] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## [109] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
## [121] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## [157] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE
```

```
sum(is.na(attenu$station))
```

```
## [1] 16
```

> Use `sum`-function to calculate number of missing values

```
sum(is.na(attenu$station))
```

```
## [1] 16
```

4.2 Run descriptive statistics

Function: `dim`

> Use `dim`-function to assess dimensions of variables or data sets

E.g. check dimensions of df

```
dim(df)
```

```
## [1] 11  3
```

Function: `mean`

> Use `mean`-function to compute mean value of variables or subset of variables

Note. use na.rm = TRUE to remove missing values if any

E.g. check average age of female students

```
mean(df[df$gender == "female", "age"])
```

```
## [1] 27.2
```

Function: `sd`

> Use `sd`-function to compute standard deviations of a variable’s values

Note. use na.rm = TRUE to remove missing values, if any

E.g. check standard deviation of age among male student

```
sd(df[df$gender == "male", "age"], na.rm = T)
```

```
## [1] 9.038805
```

4.3 Loops

Function: `for loop`

> Use `for loop`-function to repeat a specific block of code/statement/set of statements

```
for(b in 1:5) {print(b)}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

```
s <- c ("red", "blue", "green", "yellow")
for (indiancurry in s) {print(indiancurry)}
```

```
## [1] "red"
## [1] "blue"
## [1] "green"
## [1] "yellow"
```

4.4 Create tables

> Load data sets to be used

```
data("cars")
data("iris")
```

> Install and load `pander` package using `install.packages("pander")` and

```
library(pander)
```

Function: `pander`

> Use `pander`-function to create table of datasets

```
pander(cars)
```

speed	dist
4	2
4	10
7	4
7	22
8	16
9	10
10	18
10	26
10	34
11	17
11	28
12	14
12	20
12	24
12	28
13	26
13	34
13	34
13	46
14	26
14	36
14	60
14	80
15	20
15	26
15	54
16	32
16	40
17	32
17	40
17	50
18	42
18	56
18	76
18	84

speed	dist
19	36
19	46
19	68
20	32
20	48
20	52
20	56
20	64
22	66
23	54
24	70
24	92
24	93
24	120
25	85

```
pander(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa
4.3	3	1.1	0.1	setosa
5.8	4	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa
5.4	3.4	1.7	0.2	setosa
5.1	3.7	1.5	0.4	setosa
4.6	3.6	1	0.2	setosa
5.1	3.3	1.7	0.5	setosa
4.8	3.4	1.9	0.2	setosa
5	3	1.6	0.2	setosa
5	3.4	1.6	0.4	setosa
5.2	3.5	1.5	0.2	setosa
5.2	3.4	1.4	0.2	setosa
4.7	3.2	1.6	0.2	setosa
4.8	3.1	1.6	0.2	setosa
5.4	3.4	1.5	0.4	setosa
5.2	4.1	1.5	0.1	setosa
5.5	4.2	1.4	0.2	setosa
4.9	3.1	1.5	0.2	setosa
5	3.2	1.2	0.2	setosa
5.5	3.5	1.3	0.2	setosa

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
4.9	3.6	1.4	0.1	setosa
4.4	3	1.3	0.2	setosa
5.1	3.4	1.5	0.2	setosa
5	3.5	1.3	0.3	setosa
4.5	2.3	1.3	0.3	setosa
4.4	3.2	1.3	0.2	setosa
5	3.5	1.6	0.6	setosa
5.1	3.8	1.9	0.4	setosa
4.8	3	1.4	0.3	setosa
5.1	3.8	1.6	0.2	setosa
4.6	3.2	1.4	0.2	setosa
5.3	3.7	1.5	0.2	setosa
5	3.3	1.4	0.2	setosa
7	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4	1.3	versicolor
6.5	2.8	4.6	1.5	versicolor
5.7	2.8	4.5	1.3	versicolor
6.3	3.3	4.7	1.6	versicolor
4.9	2.4	3.3	1	versicolor
6.6	2.9	4.6	1.3	versicolor
5.2	2.7	3.9	1.4	versicolor
5	2	3.5	1	versicolor
5.9	3	4.2	1.5	versicolor
6	2.2	4	1	versicolor
6.1	2.9	4.7	1.4	versicolor
5.6	2.9	3.6	1.3	versicolor
6.7	3.1	4.4	1.4	versicolor
5.6	3	4.5	1.5	versicolor
5.8	2.7	4.1	1	versicolor
6.2	2.2	4.5	1.5	versicolor
5.6	2.5	3.9	1.1	versicolor
5.9	3.2	4.8	1.8	versicolor
6.1	2.8	4	1.3	versicolor
6.3	2.5	4.9	1.5	versicolor
6.1	2.8	4.7	1.2	versicolor
6.4	2.9	4.3	1.3	versicolor
6.6	3	4.4	1.4	versicolor
6.8	2.8	4.8	1.4	versicolor
6.7	3	5	1.7	versicolor
6	2.9	4.5	1.5	versicolor
5.7	2.6	3.5	1	versicolor
5.5	2.4	3.8	1.1	versicolor
5.5	2.4	3.7	1	versicolor
5.8	2.7	3.9	1.2	versicolor
6	2.7	5.1	1.6	versicolor
5.4	3	4.5	1.5	versicolor
6	3.4	4.5	1.6	versicolor
6.7	3.1	4.7	1.5	versicolor
6.3	2.3	4.4	1.3	versicolor
5.6	3	4.1	1.3	versicolor
5.5	2.5	4	1.3	versicolor
5.5	2.6	4.4	1.2	versicolor
6.1	3	4.6	1.4	versicolor
5.8	2.6	4	1.2	versicolor

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5	2.3	3.3	1	versicolor
5.6	2.7	4.2	1.3	versicolor
5.7	3	4.2	1.2	versicolor
5.7	2.9	4.2	1.3	versicolor
6.2	2.9	4.3	1.3	versicolor
5.1	2.5	3	1.1	versicolor
5.7	2.8	4.1	1.3	versicolor
6.3	3.3	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3	5.9	2.1	virginica
6.3	2.9	5.6	1.8	virginica
6.5	3	5.8	2.2	virginica
7.6	3	6.6	2.1	virginica
4.9	2.5	4.5	1.7	virginica
7.3	2.9	6.3	1.8	virginica
6.7	2.5	5.8	1.8	virginica
7.2	3.6	6.1	2.5	virginica
6.5	3.2	5.1	2	virginica
6.4	2.7	5.3	1.9	virginica
6.8	3	5.5	2.1	virginica
5.7	2.5	5	2	virginica
5.8	2.8	5.1	2.4	virginica
6.4	3.2	5.3	2.3	virginica
6.5	3	5.5	1.8	virginica
7.7	3.8	6.7	2.2	virginica
7.7	2.6	6.9	2.3	virginica
6	2.2	5	1.5	virginica
6.9	3.2	5.7	2.3	virginica
5.6	2.8	4.9	2	virginica
7.7	2.8	6.7	2	virginica
6.3	2.7	4.9	1.8	virginica
6.7	3.3	5.7	2.1	virginica
7.2	3.2	6	1.8	virginica
6.2	2.8	4.8	1.8	virginica
6.1	3	4.9	1.8	virginica
6.4	2.8	5.6	2.1	virginica
7.2	3	5.8	1.6	virginica
7.4	2.8	6.1	1.9	virginica
7.9	3.8	6.4	2	virginica
6.4	2.8	5.6	2.2	virginica
6.3	2.8	5.1	1.5	virginica
6.1	2.6	5.6	1.4	virginica
7.7	3	6.1	2.3	virginica
6.3	3.4	5.6	2.4	virginica
6.4	3.1	5.5	1.8	virginica
6	3	4.8	1.8	virginica
6.9	3.1	5.4	2.1	virginica
6.7	3.1	5.6	2.4	virginica
6.9	3.1	5.1	2.3	virginica
5.8	2.7	5.1	1.9	virginica
6.8	3.2	5.9	2.3	virginica
6.7	3.3	5.7	2.5	virginica
6.7	3	5.2	2.3	virginica
6.3	2.5	5	1.9	virginica
6.5	3	5.2	2	virginica
6.2	3.4	5.4	2.3	virginica

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.9	3	5.1	1.8	virginica

> Use `head` or `tail` -functions to limit table to x number of first or final rows in data sets

```
pander(head(cars, n=3))
```

speed	dist
4	2
4	10
7	4

> Use `[row number:row number, column number:column number]` to specify rows and columns from which to make a table

```
pander(iris[12:14, 2:5])
```

	Sepal.Width	Petal.Length	Petal.Width	Species
12	3.4	1.6	0.2	setosa
13	3	1.4	0.1	setosa
14	3	1.1	0.1	setosa

4.5 Create graphs

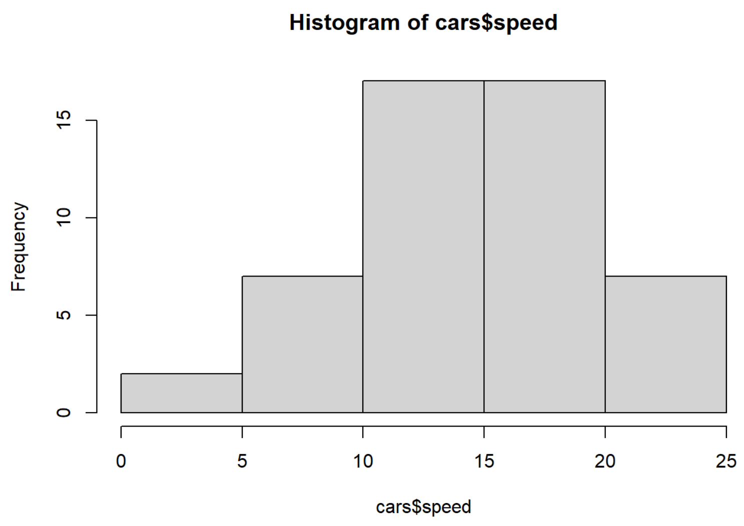
> Note: Use dollarsign `$` to access variable in dataset

Function: `hist`

> Use `hist` -function to create a histogram

E.g. create a histogram of the variable “speed” in the data set “cars”

```
hist(cars$speed)
```

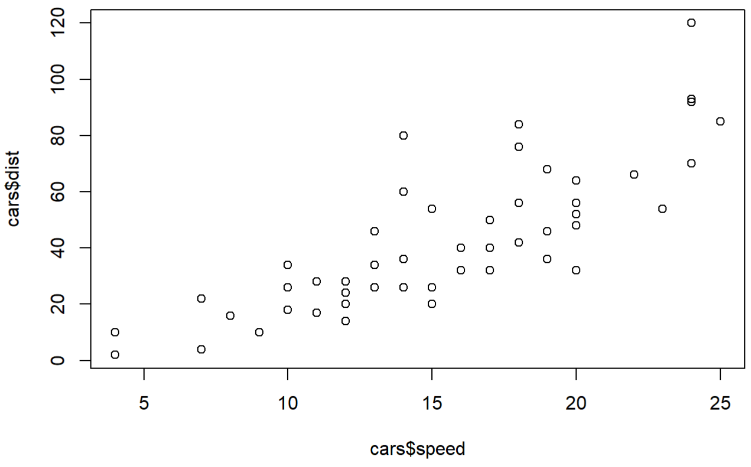


Function: `plot`

> Use `plot` -function to create scatterplot of variables

E.g. create scatterplot of variables “speed” and “dist” in data set “cars

```
plot(cars$speed, cars$dist)
```



Package: `GGplot`

> Read up on use of `GGplot` -package when you have the time