Lab 1 - Intro to labs, R and RMarkdown

Your name

January 7, 2022

To produce PDF file, you need TeX files.

- Easy way: Install the `tinytex` package: `install.packages("tinytex")`. Then run `tinytex::install_tinytex()`.
- If you want full version of TeX: For Mac install MacTeX. For Windows install TeX Live.

Basic console output

To insert an R code chunk, you can type it manually or just press `Chunks - Insert chunks` or use the shortcut key.

You can quickly insert chunks into your R Markdown file with the keyboard shortcut Cmd + Option + I (Windows Ctrl + Alt + I). This will produce the following code chunk:

Prerequiste

After cleaning the memory, we will first call the library `tidyverse`. Tidyverse loads a set of commonly used packages for data science, including `ggplot2` and `dplyr`.

```
rm(list = ls()) # Clear memory

# Load packages

#install.packages("magrittr")
#install.packages("lattice")
#install.packages("stargazer")
#install.packages("pander")
#install.packages("kableExtra")
#install.packages("tidyverse")

library(magrittr)
library(lattice)
library(stargazer)
library(pander)
library(kableExtra)
library(tidyverse)
```

R Code chunk features

Create Markdown code from R

Frequently used chunk options

Option	Description
include	If FALSE, knitr will run the chunk but not include the chunk in the final document
echo	If FALSE, knitr will not display the code in the code chunk above it's results in the final document.
error	If FALSE, knitr will not display any error messages generated by the code.
message	If FALSE, knitr will not display any messages generated by the code.
warning	If FALSE, knitr will not display any warning messages generated by the code.

Recommendation for Homework

Option	HW setting
include	TRUE
echo	TRUE
error	FALSE
message	FALSE
warning	FALSE

Data Frames Practice 1

1. Load econ.csv in R

```
df <- read.csv("data/econ.csv")
```

2. What is the data structure? What does that tell us about type?

```
# Check structure

str(df) # character, integer, numeric, and integer.
```

```
## 'data.frame':    557 obs. of  4 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Albania" ...
##  $ GWn      : int  700 700 700 339 615 615 615 615 615 540 ...
##  $ year     : int  1983 1985 1991 2000 1967 1968 1977 1986 2006 1953 ...
##  $ gdpPercap: num  863 819 601 2962 1824 ...
```

3. Check the names and summary statistics of the data. Fix any names that are less than good.

```r
# Check and fix names

names(df)
```

```
## [1] "country"   "GWn"       "year"      "gdpPercap"
```

```r
# Removing an unteresting varibale:

df <- df %>% select(-GWn)

# Two ways to change the names of a df:

# Using dplyr:

df <-
  df %>% rename(
  COUNTRY = country,
  gdp_PC = gdpPercap
)

# Using base R

names(df) <- c("COUNTRY","Year","GDP_PC_PPP")


# Summary Statistics

summary(df)
```

```
##    COUNTRY               Year        GDP_PC_PPP
##  Length:557         Min.   :1900   Min.   :  228
##  Class :character   1st Qu.:1949   1st Qu.: 1357
##  Mode  :character   Median :1970   Median : 3386
##                     Mean   :1966   Mean   : 6297
##                     3rd Qu.:1989   3rd Qu.: 8657
##                     Max.   :2010   Max.   :39157
```

4. Load pop.csv in R

```r
df2 <- read.csv("data/pop.csv")
```

5. Check and remove observations with missing values

```r
# I like to use the package questionr, but you will first have to install it

#questionr::freq.na(df2)

# Remove NAs
df2 <- na.omit(df2)
```

6. Merging two datasets

```
# Makes sure that the reference columns have the same names:

names(df)
```

```
## [1] "COUNTRY"    "Year"       "GDP_PC_PPP"
```

```
names(df2)
```

```
## [1] "country" "GWn"     "year"    "pop"     "region"
```

```
df2 %<>% # Note the assignment pipe, from magrittr
  rename(
    COUNTRY = country,
    Year = year
  )

df_m <- merge(df,df2,by=c("COUNTRY","Year"))

# check out ?merge

# Moreover, cbind() and rbind() may be useful in another applications
```

7. Calculate the average GDP per capita for Brazil for the observed period. Repeat the calculation for all countries.

```
# Base R

mean(df_m[df_m$COUNTRY == "Brazil", "GDP_PC_PPP"])
```

```
## [1] 4184.175
```

```
# Tidy way

df_m %>%
  filter(COUNTRY == "Brazil") %>%
  summarize(mean(GDP_PC_PPP))
```

```
##   mean(GDP_PC_PPP)
## 1         4184.175
```

```
# Average gdp.per.cap for all countries

df_m %>%
  group_by(COUNTRY) %>%
  summarize(mean(GDP_PC_PPP))%>%
  head() %>%    # Let's only show the first six observations
  kbl()
```

COUNTRY	mean(GDP_PC_PPP)
Afghanistan	760.6971
Albania	2962.2794
Algeria	2649.4482
Angola	825.4988
Argentina	4798.4565
Australia	7914.3872

Sometimes we are interested in report a specific number from a dataset. To minimize human mistake, we can embed R code into an R Markdown document directly with `r`.

For example, the final data set `df_m` has 508 observations and the average GDP per capita in Brazil is 4184.2.

8. Let's create some data

```
rm(list = ls()) # Clear memory

set.seed(1234) # For replication

x <- 1:10
y <- round(rnorm(10, mean=x, sd=1), digits=2)
df <- data.frame(x, y)
```

9. Basic markdown functionality

For those not familiar with standard Markdown, the following may be useful. See the source code for how to produce such points. However, RStudio does include a Markdown quick reference button that adequatly covers this material.

Dot Points

Simple dot points:

- Point 1
- Point 2
- Point 3

and numeric dot points:

1. Number 1
2. Number 2
3. Number 3

and nested dot points:

- A
  - A.1
  - A.2
- B
  - B.1
  - B.2

10. Equations

Equations are included by using LaTeX notation and including them either between single dollar signs (inline equations) or double dollar signs (displayed equations). If you hang around the Q&A site CrossValidated you'll be familiar with this idea.

There are inline equations such as $y_i = \alpha + \beta x_i + e_i$.

And displayed formulas:

$$\frac{1}{1 + \exp(-x)}$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\begin{aligned} X &= (x + a)(x - b) \\ &= x(x - b) + a(x - b) \\ &= x^2 + x(a - b) - ab \end{aligned}$$

More info: LaTeX wiki

11. Tables

Tables can be included using the following notation

A	B	C
1	Male	Blue
2	Female	Pink

Or you want to show nice regression tables

```
Mod1 <- y ~ x
Res1 <-
  lm(formula = Mod1,
     data = df)

Mod2 <- y ~ x^2
Res2 <-
  lm(formula = Mod2,
     data = df)
```

```
stargazer(Res1, Res2)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Fri, Jan 07, 2022 - 1:36:02 AM

```
#For html
#stargazer(Res1, Res2, type = "html")
```

More info: Cheat Sheet

If you want to create a fancy table from data.frame, you can use "pander" or "kable"

Table 4:

	Dependent variable:	
	y	
	(1)	(2)
x	0.965***	0.965***
	(0.116)	(0.116)
Constant	−0.192	−0.192
	(0.718)	(0.718)
Observations	10	10
R^2	0.897	0.897
Adjusted R^2	0.884	0.884
Residual Std. Error (df = 8)	1.051	1.051
F Statistic (df = 1; 8)	69.541***	69.541***
Note:	*p<0.1; **p<0.05; ***p<0.01	

```
Table <-
df %>%
  mutate(z = if_else(y>5, 1, 0)) %>%
  t()

Table
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## x  1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00
## y -0.21 2.28 4.08 1.65 5.43 6.51 6.43 7.45 8.44  9.11
## z  0.00 0.00 0.00 0.00 1.00 1.00 1.00 1.00 1.00  1.00
```

With `pander`

```
Table  %>%
  pander(caption ="Fancy Table")
```

Table 5: Fancy Table

x	1	2	3	4	5	6	7	8	9	10
y	-0.21	2.28	4.08	1.65	5.43	6.51	6.43	7.45	8.44	9.11
z	0	0	0	0	1	1	1	1	1	1

With `kable`

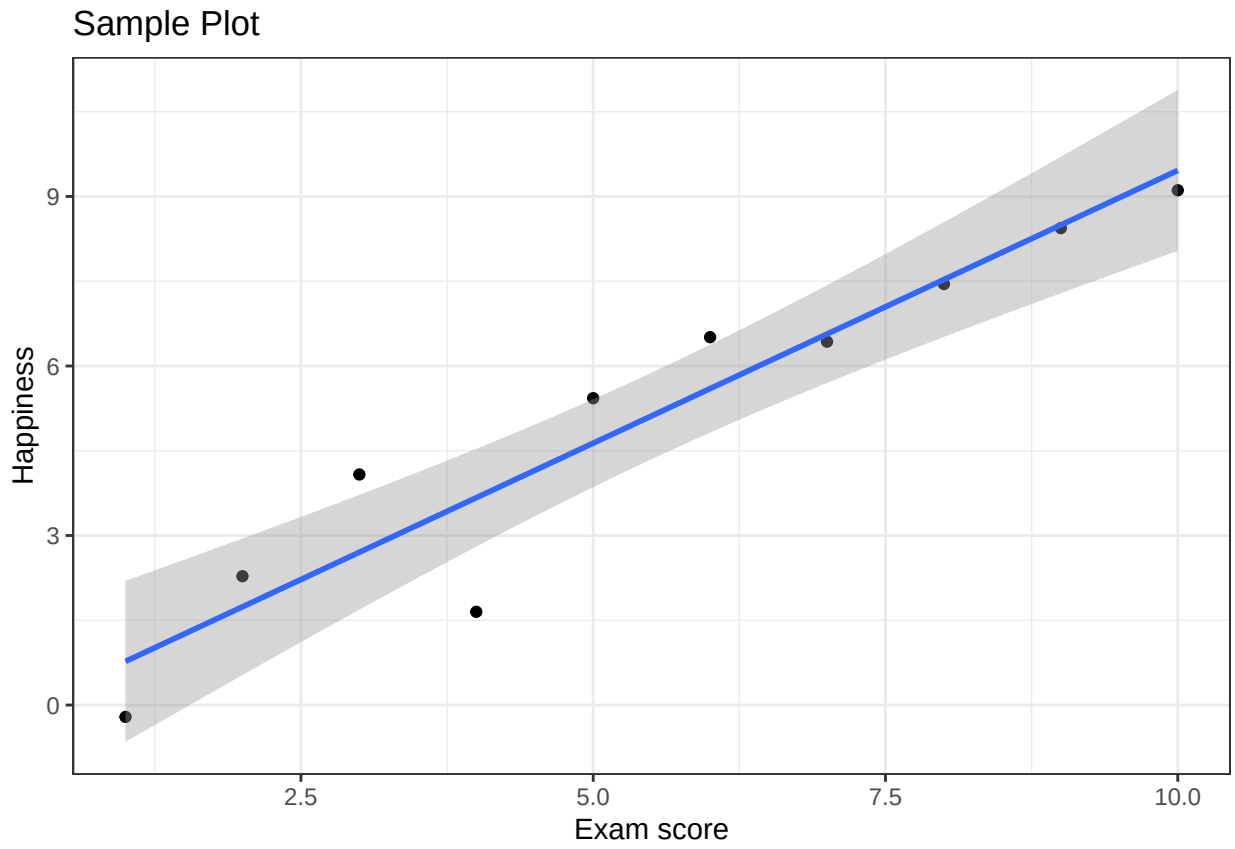
```
Table  %>% kableExtra::kable()
```

x	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00
y	-0.21	2.28	4.08	1.65	5.43	6.51	6.43	7.45	8.44	9.11
z	0.00	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00

12. Plots

You can also show plots

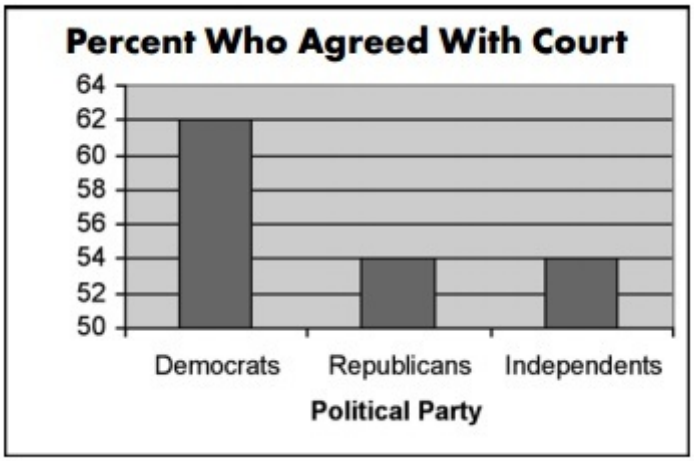
```
df %>%
  ggplot(aes(x = x, y = y))+
  geom_point()+
  geom_smooth(method = "lm", formula = y ~ x)+
  labs(title = "Sample Plot",
       y = "Happiness",
       x = "Exam score")+
  theme_bw()
```



13. Images

Images can be called using *include_graphics*.

```
knitr::include_graphics("images/terry-schiavo-misleading-graph.jpg")
```



Source: Statistics How To "Misleading Graphs: Real Life Examples"