Exercise 1 : Installing Packages

1. Install the tidyverse packages.
2. Install the magrittr packages.
3. load these pacakges using the `library()` function.

```
library(magrittr)
```

Exercise 2 : Check Your Working Directory

1. Using the `getwd()` command, check your working directory.
2. Using the `setwd()` function, change the working directory to a different path (just using "desktop" or "my documents").
3. Using the `getwd()` command, Change the working directory back to the original location.

Remark: Be careful with forward slashes and backslashes.

Exercise 3 : Load the Idaho CSV file

Prelimaries:

- Put the idaho CSV file in your working directory.
- We are going to use "`read_csv()`" from the readr tidverse package. This is not the same as "`read.csv()`", a base R function.
- Let's make sure that the CSV file is in the working directory. Run the following bit of code to find out.

```
list.files()
```

Exercise:

1. Using the `read_csv()`, load the Idaho CSV file into the R environment, saving it as "idaho".

As an aside:

Is `read_csv()` faster than `read.csv()`? Use the following code to find out.

```
# Test 1
T1 <- Sys.time() ; idaho <- read.csv("idaho.csv") ; Sys.time() - T1

# Test 2
T2 <- Sys.time() ; idaho <- read_csv("idaho.csv") ; Sys.time() - T2
```

Exercise 4 : Inspect the Dataset

```
summary(idaho)
```

1. Try out the following commands, using the previous block of code as an example, and try guess what they do.

- dim()
- nrow()
- ncol()
- names()
- tail()
- head()

The following command can tell how the data is stored in memory, and how it is structured.

- `class()` [*Base R*]
- `mode()` [*Base R*]
- `str()` [*Base R*]
- `glimpse()` [*dplyr*]

Exercise 5 : Inspect the Dataset (using Pipe Operator)

Repeat Exercise 4, but this time use the pipe operator. (You should have the magrittr R package loaded.)

```
idaho %>% summary

idaho %>% tail       # Default: Last 6

idaho %>% tail(10)   # Use last 10
```

Exercise 6 : Inspect A Specific Column

In this exercise, we will inspect the "BDS" column. This variable contains information on the number of bedrooms in each premises. To access this column, we use daho$BDS.

idaho is a data frame, made up of multiple columns.

idaho$BDS refers specifically to the BDS column in the idaho data frame.

Exercise:

1. Try out the commands from the last exercises on the BDS column.

Exercise 7 : Create A Frequency Table

The table() function is used to create a simple frequency table.

```
idaho$BDS %>% table

# or

table(idaho$BDS)
```

Remark: table() is really useful in detecting anomalous output.
`exercise`

- Create frequency tabl

Exercise 8 : Create Cross Tabulations

If given two valid inputs, rather than one, the `table()` command would create a cross-tabulation of both variables.

- This analysis is only useful if both variables are categorical variables. It would work with other types of variables, but the output would be useless for any meaningful analysis.

Exercise 9 : Compute Summary Statistics

This variable contains information on the number of bedrooms in each premises.

- mean()
- media()
- `sd()`
- `var()`
- `IDQ()`
- `mad()`

Exercise:

1. Try out the commands from the last exercises on the BDS column.

What do these functions do? What Statistical Result do they give us? To find out, we can use the help files.

```
help(mad)

#or

?mad
```

Exercise 10 : Compute Summary Statistics, in presence of Missing Data

For this exercise, we will use the OCPIP column. If you were to use the following code segment, you would see that there is lots of missing data.

```
summary(idaho$OCPIP)

# or

idaho$OCPIP  %>% summary
```

```
 %>% mean(na.rm = TRUE)

# or

mean(   ,na.rm = TRUE)
```

Exercise 11 : Selecting columns using select

In this exercise, we are going to use the dplyr R package.

We will use the select function to pick out certain columns, based on specific condition. Try out the following lines of code. Use head() and tail() to properly assess the output.

```
#First 4 Columns
temp1 <- select(idaho,1:4)
idaho %>% select(1:4) -> temp1

#All columns up to BDS
temp2 <- select(idaho,1:BDS)
idaho %>% select(1:BDS) -> temp2

#All columns that start with "wg")
temp3 <- select(idaho,starts_with("wg"))
idaho %>% select(starts_with("wg")) -> temp3

# Remark : is this case-sensitive? R usually is, but maybe not here.
```

Remark : In general, it is advised to avoid creating new objects (i.e. temporary dataframes like we used above), also it is advised to delete them as soon as they are no longer needed.

```
rm(temp1); rm(temp2);
```

Exercise 12 : Using summarize operation

```
idaho %>% summarise(mean(OCPIP))  # Wont Work in this case!

idaho %>% summarise(mean(OCPIP,na.rm=TRUE))

idaho %>% summarise(meanOP = mean(OCPIP,na.rm=TRUE))
```

Exercise 13 : Using group_by operation

There are three Types of premises, as listed in the “Type” Column.

```
table(idaho$TYPE)

# or
idaho$TYPE %>% table
```

This functions restructures the data.

```
iris %>% head(3)
iris %>% slice(1:3)


idaho %>% group_by(TYPE) %>% slice(1:3)
```

```
idaho %>% group_by(TYPE) %>% summarise(mean(OCPIP,na.rm=TRUE))

idaho %>% group_by(TYPE) %>% summarise(meanOP = mean(OCPIP,na.rm=TRUE))
```

Exercise 14 : Code Vocabulary (Part 1)

Part A: Relational Operators

- `==` Equal to
- `!=` Not Equal
- < Less than
- <= Less than or equal to
- > Greater than
- >= Greater than or equal to

Part B: Splicing

```
X <- c(1,2,5,4,6,4,12,15,10)

# Subset X to elements great than 10
X[X>10]
```

```
# Subset X to elements less than or equal to 5


X[X<=5]
```

Exercise 15 : Filter Operations

Exercise 16 : Code Vocabulary (Part 2)

Part A: Creating a Dataframe

```
Z <- c(1,2,5,4,NA,0,1,6,7,16)
myDF <- data.frame(Y,Z)
```

Part B: Using `is.na()`

```
Y <- c(1,2,5,4,NA,,1,6,7,16)
```

Part C: Using `complete.cases()`

The American Community Survey distributes downloadable data about United States communities.

Download the 2006 microdata survey about housing for the state of Idaho.

Question 1 will refer to a variable that is described on Page 11. Question 2 will refer to a variable on Page 12, and so on.

1. Estimate the median amount of property taxes paid.
2. What proportion of family households have no income in the house?
3. What proportion of family households have Food stamp amount (yearly amount) allocation
4. What proportion of premises have complete kitchen facilities on the property?
5.

How many housing units in this survey were worth more than $1,000,000?

159 164 53 24