CCS2233 LAB WEEK 3

```
# SLIDE 9 =============
getwd() # to see current working directory
setwd("C:/Users/ASUS/OneDrive/Documents/Beginner R") # to set working directory

# SLIDE 11 =============
# import csv file
ABC <- read.table(file = "iris_dirty.csv", header = TRUE, sep = ",") # use read.table
ABC <- read.csv("iris_dirty.csv") # use read.csv
class(ABC) # check the class of the csv file

# SLIDE 14 =============
Mydata <- read.table("student_performance.txt", sep = " ") # to import txt file
Mydata <- read.delim("student_performance.txt") # to import txt file

# SLIDE 16 =============
# file.choose() to import txt file
student_performance <- read.delim("C:/My Documents MAZ/UniMAP/TUGAS LUAR/AIU 2024/R Studio/Dataset/student_performance.txt")
head(student_performance)

# SLIDE 17 =============
library(readxl) # package to read excel file
read_excel("health.xlsx")

# SLIDE 18 =============
library(haven) # package to import and export 'SPSS', 'Stata' and 'SAS' files
read_spss("Dataset.sav")
read_dta("Dataset.dta")
read_sas("Dataset.sas7bdat")
```

```
# SLIDE 19 =============

# importing data from website
getLink <- "https://calmcode.io/static/data/fish.csv"
myData <- read.table(file = getLink, header = TRUE, sep = ",")

# download the data.
# Note that this will download to the current wd, but you can change it by specifying a path to "destfile"
download.file(url='https://fragilestatesindex.org/wp-content/uploads/2023/06/FSI-2023-DOWNLOAD.xlsx', destfile='excelData.xlsx', method='curl')
excelData <- read_excel("excelData.xlsx") # read data

# SLIDE 21 =============

health <- read.table(file = "health.csv", header = TRUE, sep = ",")  # read csv file
save(health, file="health.rdata") # save the health.frame to disk
rm(health) # remove health from memory
load("health.rdata") # read it from the rdata file
head(health) # check if it exists now

# SLIDE 23 =============

# r binary files
x <- 1:5
y <- letters[1:5]
z <- data.frame(x, y)

# save all three objects at once
save(x, y, z, file="multiple.rdata")
rm(x,y,z)
```

```
load("multiple.rdata")
x
y
z

# SLIDE 25 =============

x <- c(1, 5, 4) # create an object
x # view it
saveRDS(x, file='anObject.rds') # save to rds file
thatObject <- readRDS('anObject.rds') # read the file and save to a different object
thatObject # display it
identical(x, thatObject) # check they are the same

# SLIDE 27 =============

# load data from package ggplot2
data(diamonds, package='ggplot2')
head(diamonds)

# SLIDE 28 =============

library(readr) # package to read or write csv files
write.csv(health, "IntrotoR.final.csv", row.names = FALSE)
write_csv(health, 'healthExam.csv')
library(sjlabelled) # package to read and write SPSS, SAS and Stata files
write_spss(health, "my_spss.sav") # might need a longer time to execute
write_dta(health, "my_stata.dta") # might need a longer time to execute

# DATA PREPROCESSING ================================
# Data Validation ================================

iris_dirty <- read.csv("iris_dirty.csv")

summary(iris_dirty) # Summary statistics

# Data profiling
library('DataExplorer')
create_report(iris_dirty)

# Removing Duplicates ================================
iris_new <- unique(iris_dirty)
rownames(iris_new) <- 1:nrow(iris_new)
sum(duplicated(iris_new))

# Encoding Categorical Variables ================================

iris_new$Species <- tolower(iris_new$Species) # change to lowercase
iris_new$Species <- factor(iris_new$Species) # change to category

levels(iris_new$Species)[as.integer(iris_new$Species)]
levels(iris_new$Species)[1]<-"setosa"
levels(iris_new$Species)[2]<-"versicolor"
as.numeric(iris_new$Species) #category in numeric data type
iris_new

# Handling Missing Data ================================
# Replacing Using Mean

iris_new$Sepal.Length[is.na(iris_new$Sepal.Length)]<-mean(iris_new$Sepal.Length,
na.rm=TRUE)
iris_new$Sepal.Width[is.na(iris_new$Sepal.Width)]<-mean(iris_new$Sepal.Width,  na.rm=TRUE)
```

```
sum(is.na(iris_new))

# Outlier Detection and Treatment ===============================

# Detecting Outliers
iris_new <- subset(iris_new, select = -X) # remove X column
boxplot(iris_new)

# Deleting Outliers
Q1 <- quantile(iris_new$Sepal.Width, .25)
Q3 <- quantile(iris_new$Sepal.Width, .75)
IQR <- IQR(iris_new$Sepal.Width)

# only keep rows in dataframe that have values within 1.5*IQR of Q1 and Q3
iris_final <- subset(iris, iris_new$Sepal.Width> (Q1 - 1.5*IQR) & iris_new$Sepal.Width< (Q3 + 1.5*IQR))

# view row and column count of new data frame
dim(iris_final)
boxplot(iris_final)
```