COMP2501 Lab Sheet 1

## Instructions

* You are expected to answer the questions in the sheet in class
* You are encouraged to try different code or arguments
* You can discuss with other students for ideas but do not copy from others
* If you have any questions, ask Google, the two TAs, or me
* Sample answers will be given after the class on Moodle

### Environmental setup

You need to have the dplyr, ggplot2 and datasets packages installed. If not yet, please run install.packages(c("dplyr", "ggplot2", "datasets")) in your R environment. If you have installed the tidyverse package, dplyr and ggplot2 are installed by default.

# Load the packages.  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)  
library(datasets)

### 1. Write a function compute\_s\_n that for any given n computes the sum Sn = 1^2 + 2^2 + 3^2 + … + n^2. Report the value of the sum when n=100.

compute\_s\_n <- function(n) {  
 s <- 0  
 for (i in 1:n) {  
 s <- s + i^2  
 }  
 return(s)  
}  
  
compute\_s\_n(100)

## [1] 338350

### 2. Play with the built-in dataset mtcars to explore the package dplyr. Please use head() to show the results if there are too many rows in them.

#### a. Load the dataset mtcars. Print 1) the first 6 rows of the data, 2) how many observations (rows) and variables (columns) are in the dataset, 3) the names of all variables.

data(mtcars)  
head(mtcars)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

dim(mtcars)

## [1] 32 11

names(mtcars)

## [1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec" "vs" "am" "gear"  
## [11] "carb"

#### b. Create a new variable performance as the ratio of horsepower (hp) to weight (wt) (i.e. ). Please do operations on a dataset copy of mtcars named mtcars\_modified.

mtcars\_modified <- mtcars |> mutate(performance = hp / wt)  
head(mtcars\_modified)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1  
## performance  
## Mazda RX4 41.98473  
## Mazda RX4 Wag 38.26087  
## Datsun 710 40.08621  
## Hornet 4 Drive 34.21462  
## Hornet Sportabout 50.87209  
## Valiant 30.34682

#### c. 1) Use top\_n() function to find out the top 1 observation in mtcars\_modified with the highest value of performance, 2) Sort the mtcars\_modified by performance in descending order. Check if the result in 1) is the first row of the results in 2).

top\_n(mtcars\_modified, 1, performance)

## mpg cyl disp hp drat wt qsec vs am gear carb performance  
## Maserati Bora 15 8 301 335 3.54 3.57 14.6 0 1 5 8 93.83754

mtcars\_modified |> arrange(desc(performance)) |> head()

## mpg cyl disp hp drat wt qsec vs am gear carb performance  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 93.83754  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 83.28076  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 74.68605  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 68.62745  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 63.80208  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 63.17690

# yes

#### d. Filter the original dataset mtcars to show only the cars with a mpg value greater than 20. Please do operations on a dataset copy of mtcars named mtcars\_filtered.

mtcars\_filtered <- mtcars |> filter(mpg > 20)  
head(mtcars\_filtered)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2

#### e. Select only the mpg and cyl columns from the filtered dataset mtcars\_filtered. Please do operations on a dataset copy of mtcars\_filtered named mtcars\_selected.

mtcars\_selected <- mtcars\_filtered |> select(mpg, cyl)  
head(mtcars\_selected)

## mpg cyl  
## Mazda RX4 21.0 6  
## Mazda RX4 Wag 21.0 6  
## Datsun 710 22.8 4  
## Hornet 4 Drive 21.4 6  
## Merc 240D 24.4 4  
## Merc 230 22.8 4

#### f. Group the dataset mtcars\_selected by cyl and summarize the mean mpg for each group. Please do operations on a dataset copy of mtcars\_selected named mtcars\_grouped.

mtcars\_grouped <- mtcars\_selected  
mtcars\_grouped |> group\_by(cyl) |> summarize(mean\_mpg = mean(mpg))

## # A tibble: 2 × 2  
## cyl mean\_mpg  
## <dbl> <dbl>  
## 1 4 26.7  
## 2 6 21.1

### 3. Play with the built-in dataset iris to explore the package ggplot2. Please set an appropriate plot title and axis titles.

#### a. Load the dataset iris. Print 1) the first 6 rows of the data, 2) how many observations (rows) and variables (columns) are in the dataset, 3) the names of all variables.

data(iris)  
head(iris)

## 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

dim(iris)

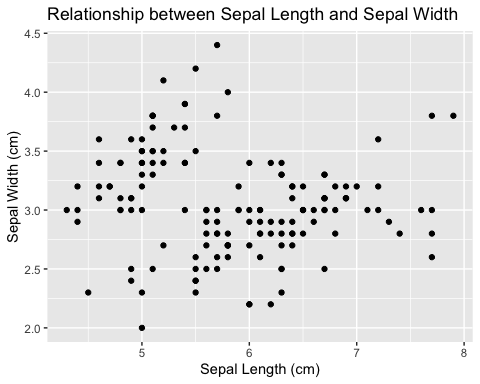
## [1] 150 5

names(iris)

## [1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"

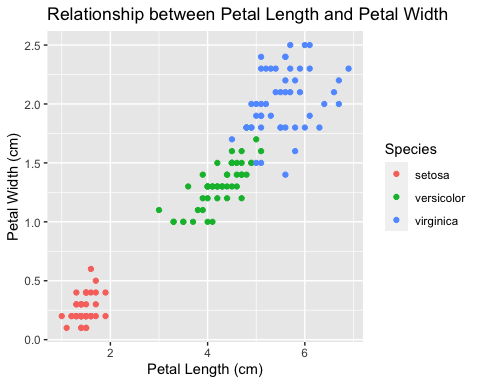
#### b. Create a scatter plot to visualize the relationship between Sepal.Length (x-axis) and Sepal.Width (y-axis).

ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
 geom\_point() +  
 ggtitle("Relationship between Sepal Length and Sepal Width") +  
 xlab("Sepal Length (cm)") +  
 ylab("Sepal Width (cm)")



#### c. Create a scatter plot to visualize the relationship between Petal.Length and Petal.Width, and color the points by Species.

ggplot(data = iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +  
 geom\_point() +  
 ggtitle("Relationship between Petal Length and Petal Width") +  
 xlab("Petal Length (cm)") +  
 ylab("Petal Width (cm)")



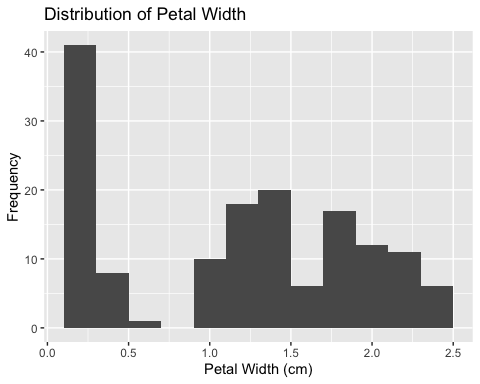
#### d. Create a boxplot to visualize the distribution of Petal.Length of each Species.

ggplot(data = iris, aes(x = Species, y = Petal.Length)) +  
 geom\_boxplot() +  
 ggtitle("Distribution of Petal Length by Species") +  
 xlab("Species") +  
 ylab("Petal Length (cm)")



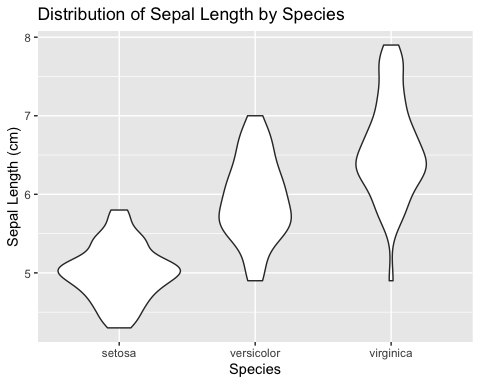
#### e. Create a histogram to visualize the distribution of Petal.Width.

ggplot(data = iris, aes(x = Petal.Width)) +  
 geom\_histogram(binwidth = 0.2) +  
 ggtitle("Distribution of Petal Width") +  
 xlab("Petal Width (cm)") +  
 ylab("Frequency")



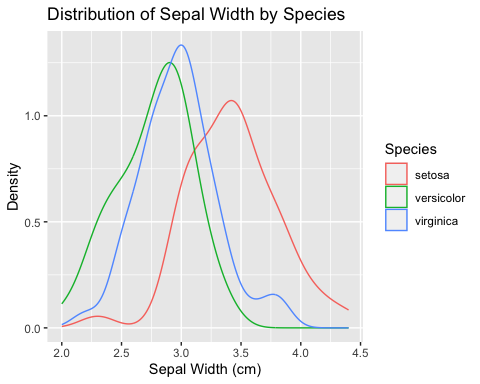
#### f. Create a violin plot to visualize the distribution of Sepal.Length by Species.

# Create a violin plot of Sepal Length by Species  
ggplot(data = iris, aes(x = Species, y = Sepal.Length)) +  
 geom\_violin() +  
 ggtitle("Distribution of Sepal Length by Species") +  
 xlab("Species") +  
 ylab("Sepal Length (cm)")



#### g. Create a density plot to visualize the distribution of Sepal.Width by Species.

ggplot(data = iris, aes(x = Sepal.Width, color = Species)) +  
 geom\_density() +  
 ggtitle("Distribution of Sepal Width by Species") +  
 xlab("Sepal Width (cm)") +  
 ylab("Density")



### 4. Practice with the real-world dataset covid\_hk\_data (“<http://www.chp.gov.hk/files/misc/enhanced_sur_covid_19_eng.csv>”). Please use head() to show the results if there are too many rows in them.

#### a. Load the dataset covid\_hk\_data. Print 1) the first 6 rows of the data, 2) how many observations (rows) and variables (columns) are in the dataset, 3) the names of all variables.

covid\_hk\_url <- "http://www.chp.gov.hk/files/misc/enhanced\_sur\_covid\_19\_eng.csv"  
covid\_hk\_data <- read.csv(covid\_hk\_url)  
head(covid\_hk\_data)

## Case.no. Report.date Date.of.onset Gender Age Name.of.hospital.admitted  
## 1 1 23/01/2020 21/01/2020 M 39 NA  
## 2 2 23/01/2020 18/01/2020 M 56 NA  
## 3 3 24/01/2020 20/01/2020 F 62 NA  
## 4 4 24/01/2020 23/01/2020 F 62 NA  
## 5 5 24/01/2020 23/01/2020 M 63 NA  
## 6 6 26/01/2020 21/01/2020 M 47 NA  
## Hospitalised.Discharged.Deceased HK.Non.HK.resident Classification.  
## 1 Discharged Non-HK resident Imported case  
## 2 Discharged HK resident Imported case  
## 3 Discharged Non-HK resident Imported case  
## 4 Discharged Non-HK resident Imported case  
## 5 Discharged Non-HK resident Imported case  
## 6 Discharged HK resident Imported case  
## Case.status.  
## 1 Confirmed  
## 2 Confirmed  
## 3 Confirmed  
## 4 Confirmed  
## 5 Confirmed  
## 6 Confirmed

dim(covid\_hk\_data)

## [1] 15441 10

names(covid\_hk\_data)

## [1] "Case.no." "Report.date"   
## [3] "Date.of.onset" "Gender"   
## [5] "Age" "Name.of.hospital.admitted"   
## [7] "Hospitalised.Discharged.Deceased" "HK.Non.HK.resident"   
## [9] "Classification." "Case.status."

#### b. Filter the dataset covid\_hk\_data with Case.status. equal to Confirmed. Please do operations on a dataset copy of covid\_hk\_data named covid\_hk\_data\_filtered.

covid\_hk\_data\_filtered <- covid\_hk\_data |> filter(Case.status. == "Confirmed")  
head(covid\_hk\_data\_filtered)

## Case.no. Report.date Date.of.onset Gender Age Name.of.hospital.admitted  
## 1 1 23/01/2020 21/01/2020 M 39 NA  
## 2 2 23/01/2020 18/01/2020 M 56 NA  
## 3 3 24/01/2020 20/01/2020 F 62 NA  
## 4 4 24/01/2020 23/01/2020 F 62 NA  
## 5 5 24/01/2020 23/01/2020 M 63 NA  
## 6 6 26/01/2020 21/01/2020 M 47 NA  
## Hospitalised.Discharged.Deceased HK.Non.HK.resident Classification.  
## 1 Discharged Non-HK resident Imported case  
## 2 Discharged HK resident Imported case  
## 3 Discharged Non-HK resident Imported case  
## 4 Discharged Non-HK resident Imported case  
## 5 Discharged Non-HK resident Imported case  
## 6 Discharged HK resident Imported case  
## Case.status.  
## 1 Confirmed  
## 2 Confirmed  
## 3 Confirmed  
## 4 Confirmed  
## 5 Confirmed  
## 6 Confirmed

#### c. Group the dataset covid\_hk\_data\_filtered by Report.date, and calculate the total number of confirmed cases each day (total\_confirmed). Please do operations on a dataset copy of covid\_hk\_data\_filtered named covid\_hk\_data\_grouped.

covid\_hk\_data\_grouped <- covid\_hk\_data\_filtered |> group\_by(Report.date) |> summarize(total\_confirmed = sum(Case.status. == "Confirmed"))  
head(covid\_hk\_data\_grouped)

## # A tibble: 6 × 2  
## Report.date total\_confirmed  
## <chr> <int>  
## 1 01/01/2021 42  
## 2 01/01/2022 8  
## 3 01/02/2020 2  
## 4 01/02/2021 34  
## 5 01/02/2022 94  
## 6 01/03/2020 5

#### d. Add a new column date with the standard date format “YYYY-MM-DD” to the covid\_hk\_data\_grouped dataset according to the Report.date column. Be reminded the format of Report.date is “DD/MM/YYYY”.

covid\_hk\_data\_grouped <- mutate(covid\_hk\_data\_grouped, date=as.Date(as.character(Report.date),"%d/%m/%Y"))  
head(covid\_hk\_data\_grouped)

## # A tibble: 6 × 3  
## Report.date total\_confirmed date   
## <chr> <int> <date>   
## 1 01/01/2021 42 2021-01-01  
## 2 01/01/2022 8 2022-01-01  
## 3 01/02/2020 2 2020-02-01  
## 4 01/02/2021 34 2021-02-01  
## 5 01/02/2022 94 2022-02-01  
## 6 01/03/2020 5 2020-03-01

#### e. Use ggplot2 to visualize the total number of confirmed cases over date with covid\_hk\_data\_grouped dataset.

ggplot(covid\_hk\_data\_grouped, aes(x = date, y = total\_confirmed)) +   
 geom\_line() +   
 ggtitle("Total number of confirmed cases in Hong Kong") +   
 xlab("Date") +   
 ylab("Confirmed Cases")

