**INTRODUCTION TO DATA SCIENCE**

**R Programming Exercises**

**Member Names:**

**Cueva, Larry Miguel, R. Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Year and Section: \_\_\_\_\_\_\_\_\_\_\_\_\_**

**College Grades Data**

Remember the college student’s data? We'll compute their final grades.

The grade computation will be:

* **30%:** Average grade of long exams (2 long exams)
* **40%:** Final exam grade (1 final exam)
* **30%:** HW/Machine Problems (3 Homeworks)

**Class policy**

Before the course started, you also have the discussed to them the following policies:

1. The grading scale is:

|  |  |
| --- | --- |
| **Final Grade** | **Final Grade in Percentage** |
| 1.0 | [ 80 – 100 ] |
| 2.0 | [ 70 – 79 ] |
| 3.0 | [ 50 – 69 ] |
| 4.0 | [ 0 – 49 ] |

2. Max absences is 6. Students who exceeded the limit will get a grade of 5.0.

3. Missed exams or HW: 0

4. Cheating will get 0 points on that specific HW/Exam.

**Additional Premise**

1. You will see that almost everyone cheated in **HW 3**, except for **Chrisphee**. They all have the same answers and got **97**. Change the **HW 3** scores of students who cheated to **0**, and change Chrisphee's HW score to **100**.
2. You promised a bonus of **5** points to be added in their final percentage grades.

**1. Data Audit**

**1.1. Import library ‘dplyr’ and load College Students dataset in R**

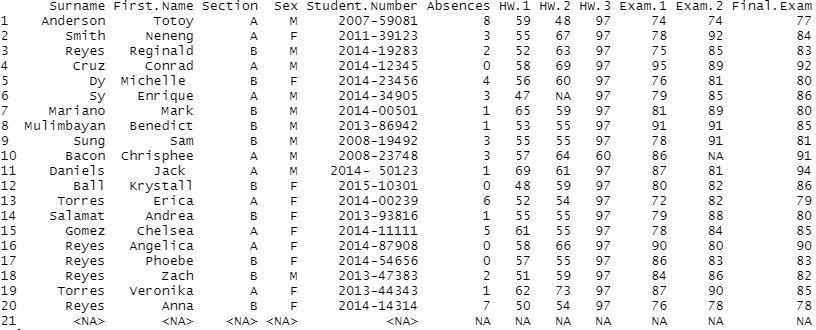
Load the **college\_students\_w\_header.csv** data here and store it as **college\_df.** Check the data frame then see the expected output below.

**library(dplyr)**

**college\_df <- read.csv(file.choose(), header=T)**

**college\_df**

**Expected Output:**



1098//8

//24

**1.2. Check how many entries do we have?**

cat("\"the number of entries are: \"", nrow(college\_df))

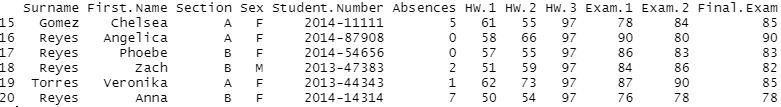
**Expected Output:**



**1.3. Remove the entries that have no “Surname” and “First Name”**

college\_df <- college\_df[!is.na(college\_df$Surname) & !is.na(college\_df$First.Name),]

**Expected Output:**



**1.4 How many students did not take the “HW 2”?**

nrow(college\_df[is.na(college\_df$HW.2),])

**Expected Output:**



**1.5. Who are they?**

hw2\_absentee <- college\_df[is.na(college\_df$HW.2),]

select(hw2\_absentee, Surname, First.Name)

**Expected Output:**



**1.6. How many students did not take the “Exam 2”?**

nrow(college\_df[is.na(college\_df$Exam.2),])

**Expected Output:**



**1.7. Who are they?**

exam2\_absentee <- college\_df[is.na(college\_df$Exam.2),]

select(exam2\_absentee, Surname, First.Name)

**Expected Output:**



**2. Data Cleaning and Processing**

Checklist:

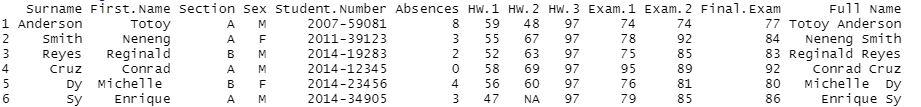
* **Add new columns** 
  + “***Full Name***” from “**Surname**” and “**First Name**”, please follow the format <*first\_name*> <*surname*>, example: Chrisphee Bacon
* **Resolve the cheating incident**
  + HW 3 grades should be 0
  + Chrisphee's HW 3 grade should be 100 because of her honestly.
* **Resolve the missed HWs and Exams**
  + HW 2
  + Exam 2

**2.1. Add new columns – “Full Name”**

full\_name <- paste(college\_df$First.Name, college\_df$Surname)

college\_df['Full name'] <- full\_name

**Expected Output:**



**2.2. Perform changes based on the cheating incident.**

cheaters\_to\_zero <- function(hw3\_x){

print(hw3\_x)

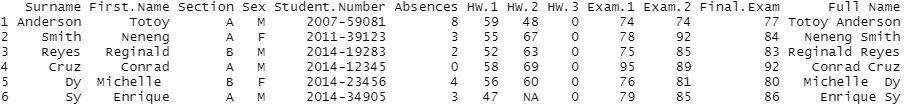
return (if(hw3\_x >= 97) 0 else 100)

}

college\_df$HW.3 <- sapply(college\_df$HW.3, cheaters\_to\_zero)

print(college\_df)

**Expected Output:**



**2.3. Chrisphee's HW 3 grade should be 100 because of her honesty.**

cheaters\_to\_zero <- function(hw3\_x){

print(hw3\_x)

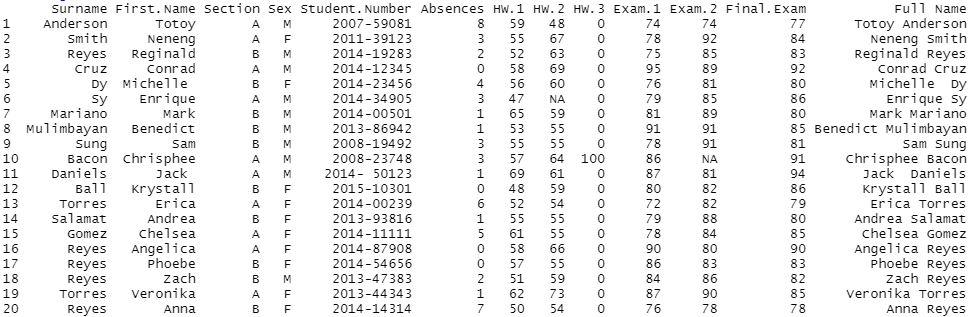
return (if(hw3\_x >= 97) 0 else 100)

}

college\_df$HW.3 <- sapply(college\_df$HW.3, cheaters\_to\_zero)

print(college\_df)

**Expected Output:**



**2.4. Resolve missed HW’s and Exams.**

resolve\_missed <- function(col\_i){

return (if(is.na(col\_i)) 0 else col\_i)

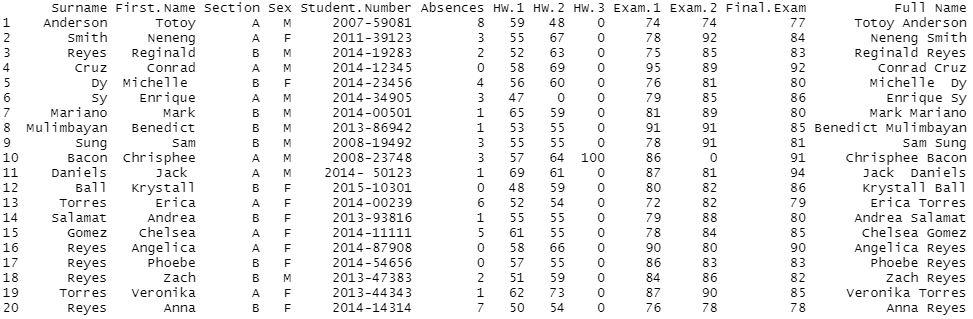
}

college\_df$HW.2 <- sapply(college\_df$HW.2, resolve\_missed)

college\_df$Exam.2 <- sapply(college\_df$Exam.2, resolve\_missed)

print(college\_df)

**Expected Output:**



**2.5. Computation of Grades**

**Checklist:**

* Compute for the final grades. Don't forget to add **5** points in their Final Grade (**%**), you kind-hearted person you.
* Convert their grades based on the scale.
* Set the grade to **5.0** for students with excessive absences.

**2.5.1. Get the average mean of the HWs and Exams.**

Before getting the **“HW mean”** and **“Exam mean”,** check first if these attributes: **‘HW.1’**, **‘HW.2’**, **‘HW.3’**, **‘Exam.1’**, and **‘Exam.2’** are in numeric or integer data type.

# check the data type of each column of college\_df

sapply(college\_df, class)

Just in case, if one of them are in a ***character data type***, this should be converted into **numeric data type**.

***Write your code here…***

Get the **mean** and store them in columns **“HW Mean”** and **“Exam Mean”**, respectively.

# create new columns HW Mean and Exam Mean that

# represent the mean of columns HW1, HW2, and

# HW3 and Exam1, and Exam2 respectively

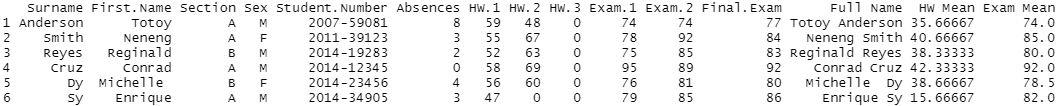
# select all the rows of college\_df but constraining

# the columns to only hw1, hw2, and hw3

college\_df['HW Mean'] <- rowMeans(college\_df[, c("HW.1", "HW.2", "HW.3")], na.rm=TRUE)

college\_df['Exam Mean'] <- rowMeans(college\_df[, c("Exam.1", "Exam.2")], na.rm=TRUE)

**Expected Output:**



**2.5.2. Compute Final Grade in Percentage (FG Per)**

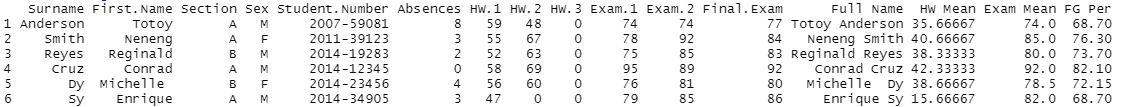
Use the columns **Final Exam**, **HW Mean**, and **Exam Mean**.

For reference:

* **30%**: Mean grade of long exams (2 long exams)
* **40%**: Final exam grade (1 final exam)
* **30%**: HW/Machine Problems (3 homeworks)

***Write your code here…***

**Expected Output:**



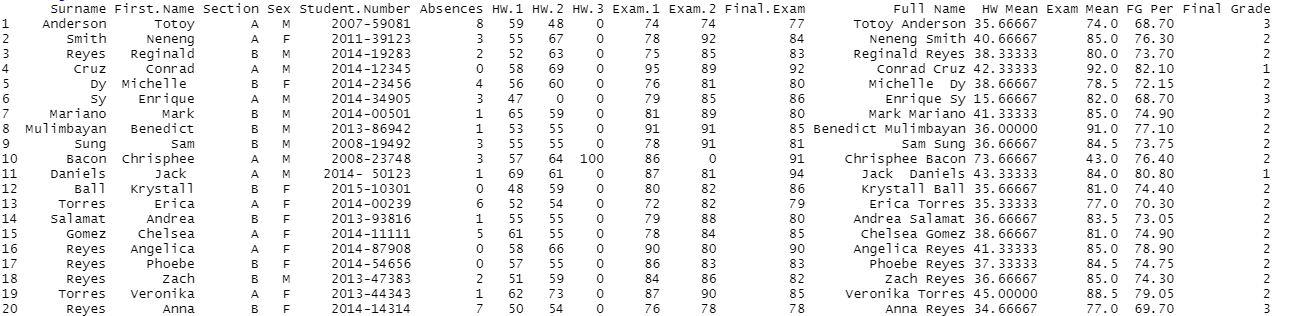
**2.5.3. Get the scaled grades**

This should be based from the **FG Per** column.

For reference:

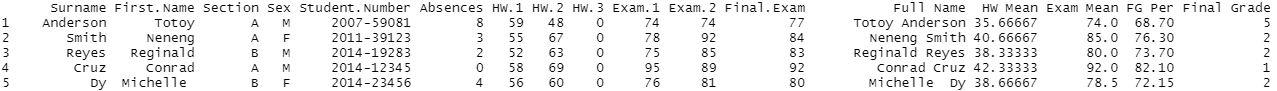
|  |  |
| --- | --- |
| **Final Grade** | **Final Grade in Percentage** |
| 1.0 | [ 80 – 100 ] |
| 2.0 | [ 70 – 79 ] |
| 3.0 | [ 50 – 69 ] |
| 4.0 | [ 0 – 49 ] |

***Write your code here…***

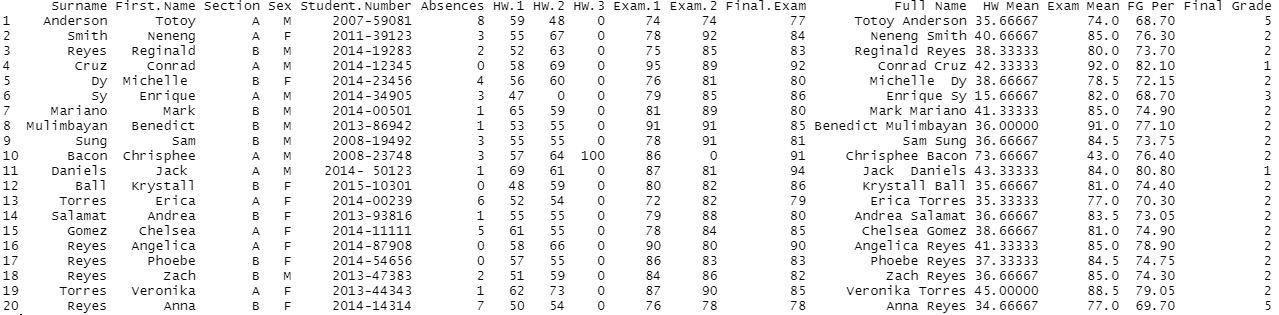
**Expected Output:**

**2.5.4. Resolve excessive absences**

***Write your code here…***



**Expected Output:**

**FINAL OUTPUT:**

**End of Exercise.**