**Assignment 2 - Fundamentals of Data Science**

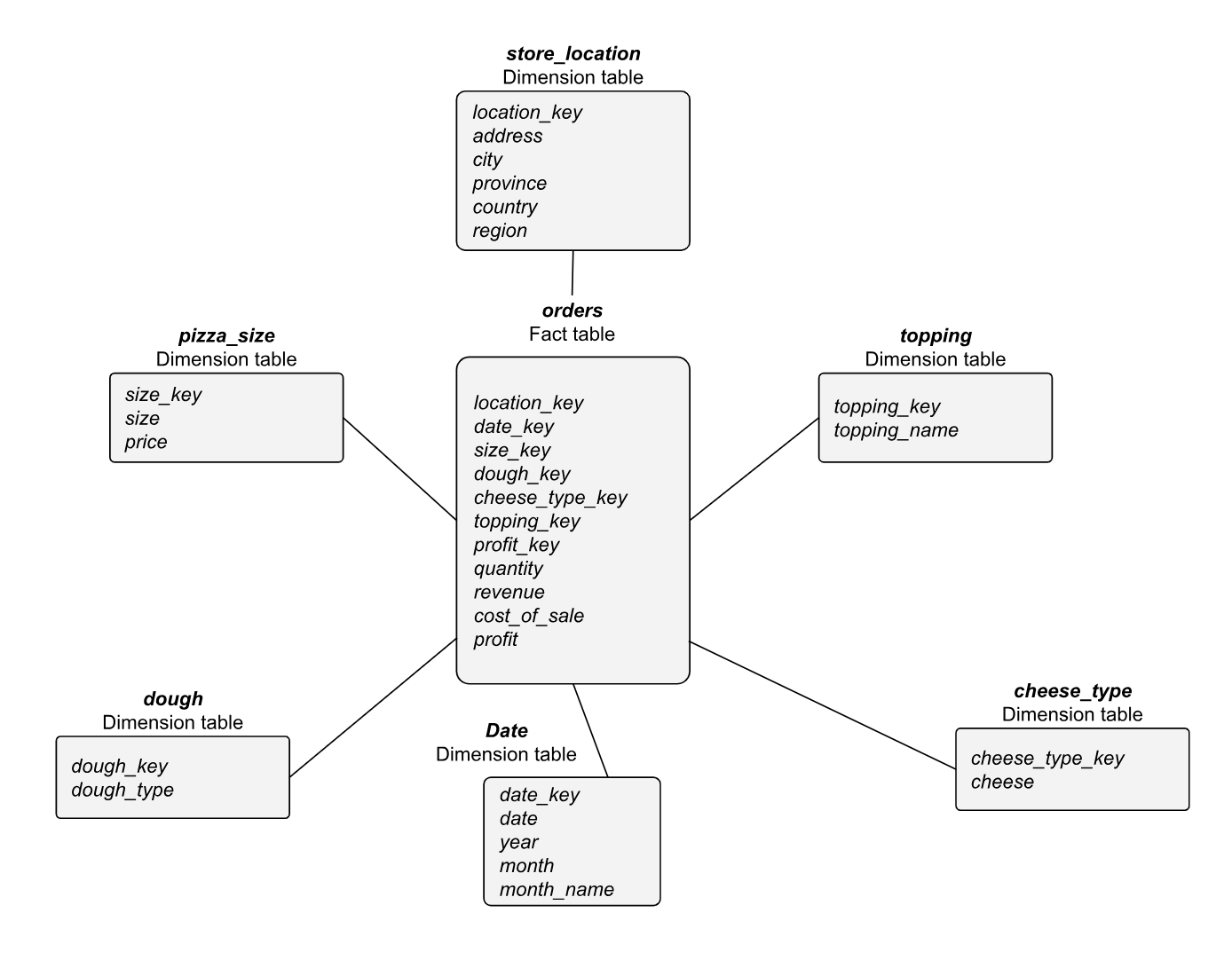
**Part A**

**1a. Star schema**

The following star schema includes Orders (Fact) as well as all of its dimensions. A date dimension has been added as well for the ease of calculations in subsequent questions.

It should be noted that the size dimension (labeled pizza\_size) also includes the prices of each pizza size. For the purpose of this activity, we are assuming that the price of the pizza is dependent on its size. Therefore, the other attributes of the pizza (topping, dough, cheese) do not contribute to the total price of the pizza.

Revenue and profit attributes are added to the schema as they calculate the revenue and profit of the order, using another attribute called cost-of-sale (i.e., profit = revenue - cost of sale).



**1b. Snowflake schema**

In the above star schema the store\_location dimension table is not normalized. Hence, we have introduced the city dimension table and country dimension table to make a snowflake, to normalize the snowflake schema and reduce redundancies. Such a table makes it easy to maintain and saves processing capabilities and thus resources.

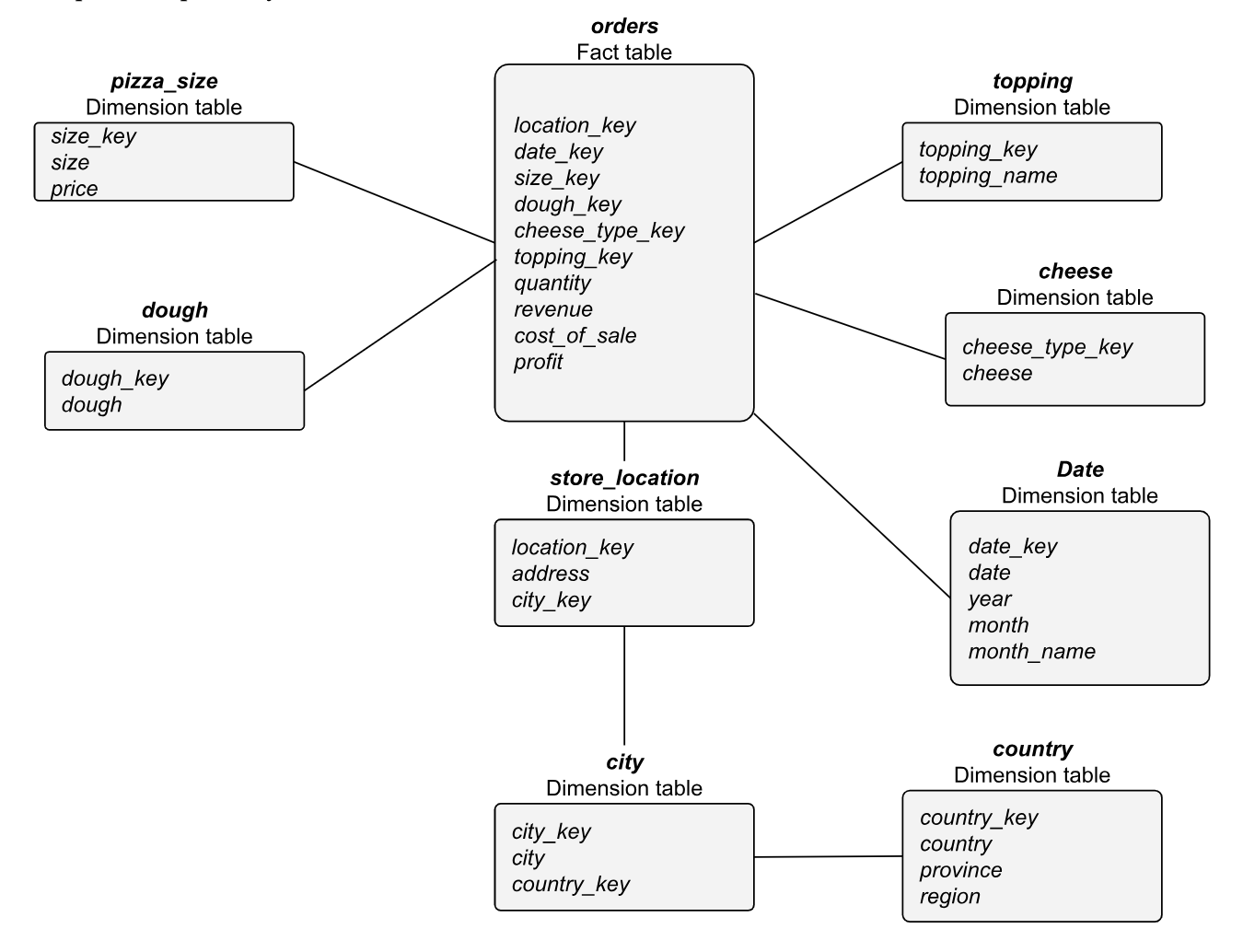
We included ‘city’ and ‘country’ as separate dimension tables because there are thousands of cities and hundreds of countries. Furthermore, the presence of region and province attributes would require the table to be further normalized and split.

The reason for linking the country dimension table to the city dimension table is to optimize the information retrieval from the data warehouse.

Again, note that we have assumed that the price of the pizza is dependent on only the size of the pizza ordered. Same is reflected in the dimension tables.

**Points to note:**

* In the above diagram, we altered the fact table to add a few more attributes, which are calculated fields (i.e., revenue, profit and cost of sale), in order to improve the quality of the fact table and the ultimate analysis that would be carried out using it.We also used the same schema in our R files to generate our dimensions and facts.
* In addition, we introduce the price and cost\_of\_sale into the orders fact table, to calculate revenue and profit, respectively.



* We also introduced quantity and profit fields in order to provide deeper analysis of the orders. These fields are derived fields and are populated during fact population.
* Revenue is a value derived using price x quantity (in orders table).
* Profit is a value derived using revenue - cost of sale.

The R file attached provides the code that was used to build the OLAP.

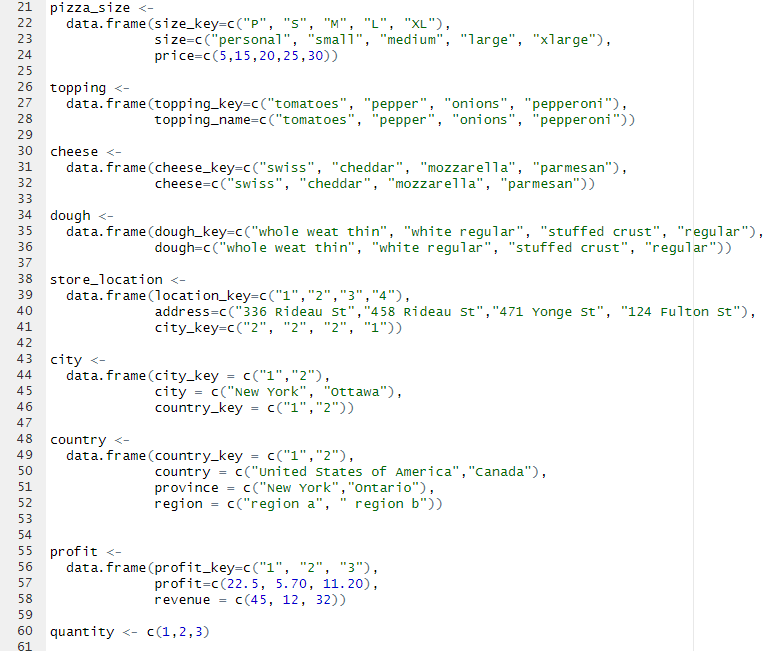
The following sections provide the details of the tasks completed.

**Note**: Even though the keys for each dimension table are supposed to be numeric, for the purpose of better understanding our cube, we have replaced numeric values with strings.

**1c. Generating data**

We first created a date dimension having dates in the first 15 days of October 2023. This range can be expanded to generate dates of multiple months or years, and the code attached would simply split it into relevant months, years, month names and dates accordingly. We experimented with this and generated a sample dataset for 1 year of data, however, we proceeded with 15 days of data to keep our analysis straightforward and to increase the readability of our results.

Next, we populated the rest of our dimensions.



All of the dimensions were with respective fields. In the case of topping, cheese, and dough our primary key was the same as the name to enhance the readability of our cube. In the case where we added the numeric values to the key, we proceeded to merge the fact with the respective dimension to retrieve the non numeric values (cheese names, dough name etc.) to improve readability. However, we have not included this into our code at this point.

The quantity field was introduced here but used while sampling.

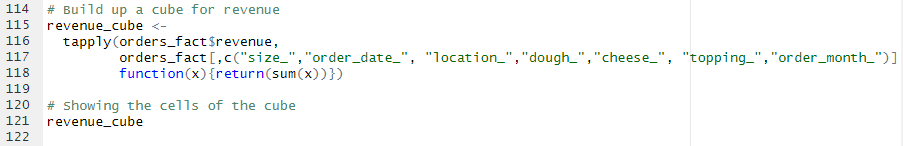
Next step was to create a function to generate orders which would populate our Orders fact. We completed this by using the following code:



The fields revenue, cost\_of\_sales and profit\_ are all calculated fields which have been calculated while populating the fact. This would certainly help in keeping the formulas for all of these fields constant. Even though this might be a costly activity in terms of resources, this approach makes the most sense since we are generating our own data and do not have data from a database providing us with these values.

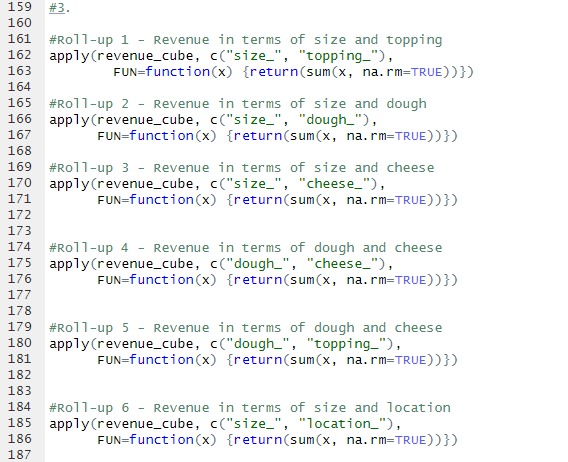
2. Once generated, we used the Orders fact to build our revenue cube.

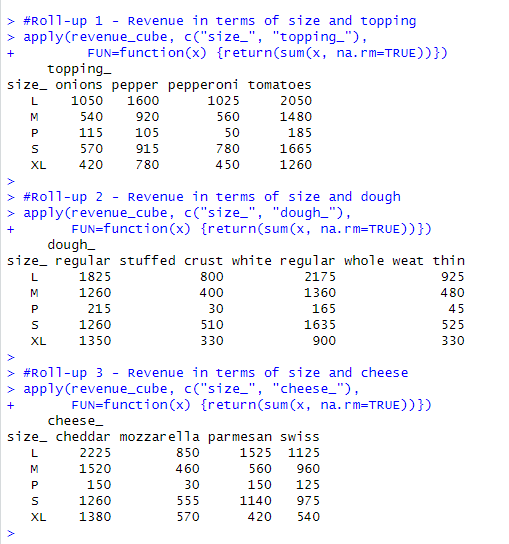
This revenue\_cube had the dimensions of size, order date, location, dough, cheese, topping and order month. To analyze other aspects of the order, for example profits, quantities sold or cost\_of\_sales, other cubes can also be built. We have included some of these in our code which we used for verification of our cube.

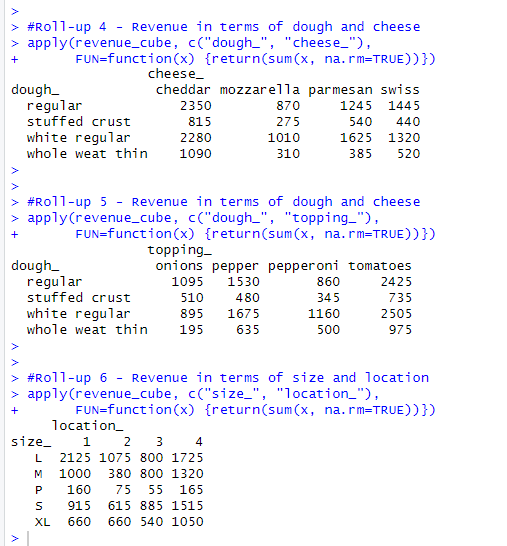


With our OLAP cube complete, we moved on to analyzing the data in it.

We started with applying a series of roll-up operations to bifurcate the fact data into dimension, leading us to finding some extremely interesting trends in our dataset.







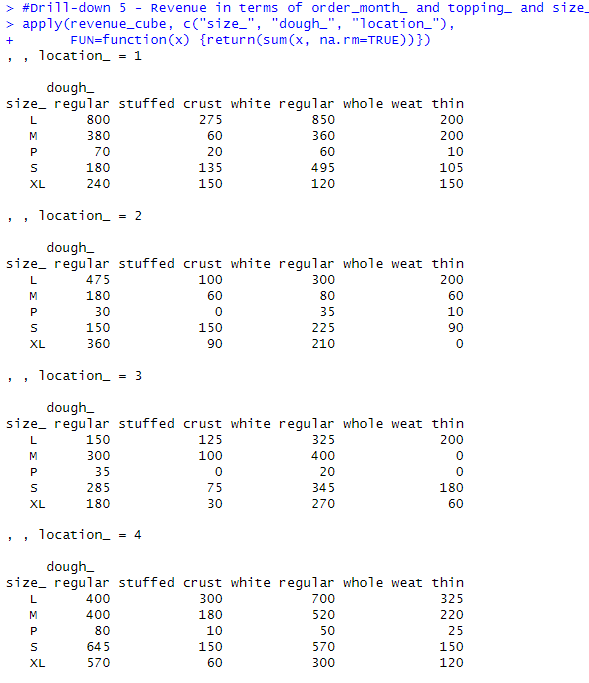
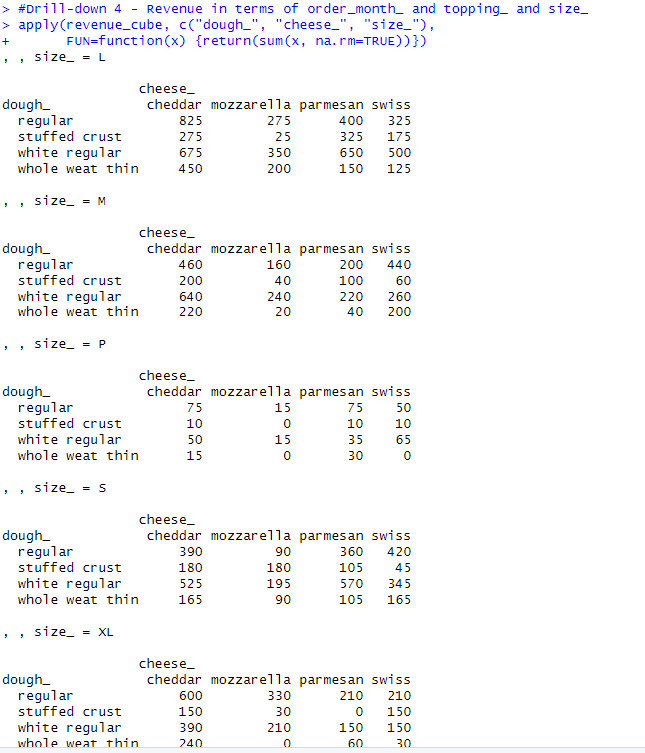
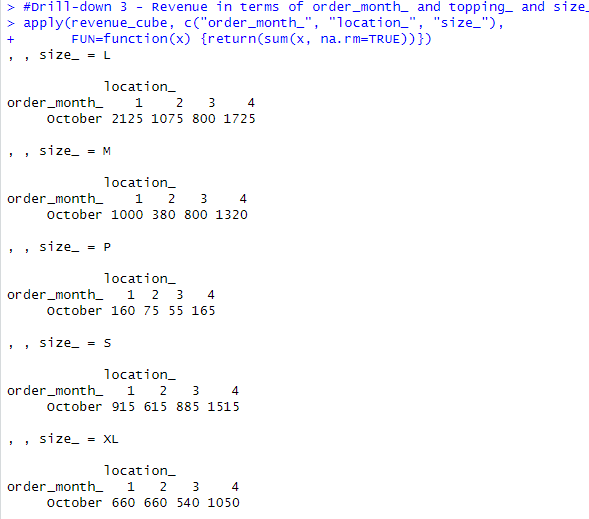
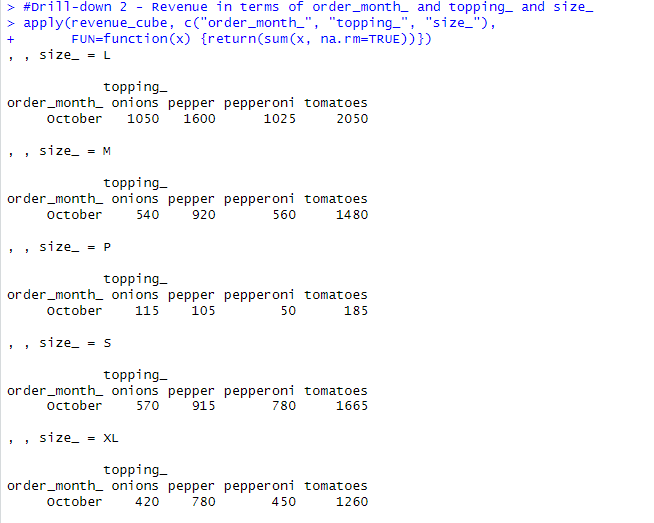
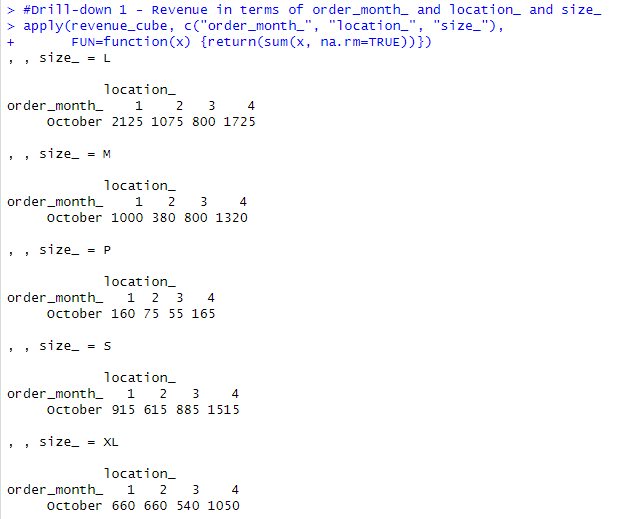
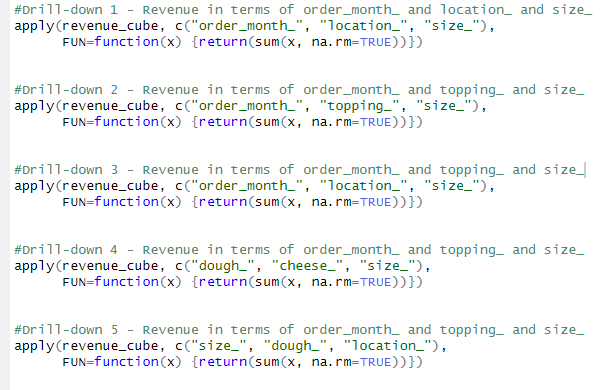
With our roll-up operations, we quickly analyzed the low revenue generated by “Personal” pizza size across all dimensions. And in every roll-up operation, the highest revenue was brought in by “Large” pizza size orders.

**Roll-up analysis**

1. In the size vs toppings roll-up, Large pizzas with tomatoes generated the highest revenue, while Personal pizzas with pepperoni topping was the least popular. Overall, the tomato toppings generated the most revenue across all pizza sizes.
2. Similarly, in size vs dough comparison, we noticed that regular and white regular dough performed equally as well, competing with each other very closely. While both stuffed crust and whole weat thin lagged behind, with stuffed crust performing the worst.
3. The most interesting roll-up results were brought by rollup 6 comparing size and location. It seemed that while all locations performed almost equally well, the Personal pizza performed the worst in all locations

.

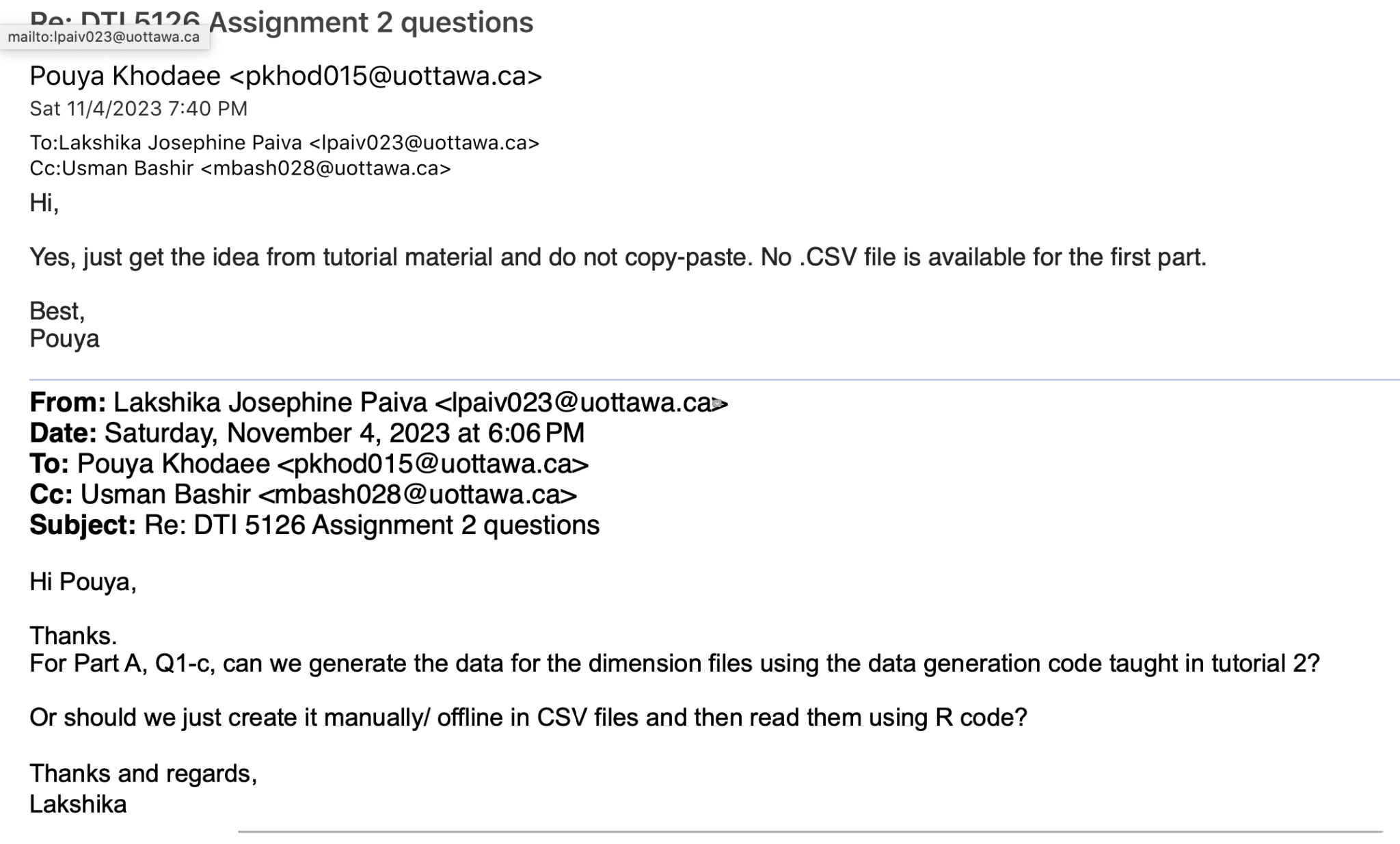
1. It is worth noting, that in all of the roll up operations performed, personal pizzas were the worst performing pizza sizes. Where as the large and extra large pizzas were the most sold regardless of the location, topping, order data. Therefore, customers are preferring larger pizzas over small or medium sized pizzas.



**Drill-down analysis**

1. Through our drill down operations, we were able to find staggering analysis patterns. In our first operation we drilled down into the OLAP cube on the 3 dimensions of order month, location and size. We found that as per our roll-up operations, Personal pizzas were sold extremely less than any other pizza. Therefore, the store should not focus on getting more components of Personal sized pizza and instead buy large and extra large in bulk quantities as they prove to be the most selling and revenue generating sizes.
2. When we drilled down into order month vs topping vs size, we analyzed that pepperoni and onion generated the least revenue in the entire month for all pizza sizes. Whereas tomatoes were the most favored. It would thus be advised to the store to buy more tomato topping components than pepperoni. Pepper toppings were closely following tomatoes and would require to be replenished.
3. Our most important finding was in drilling down dough vs cheese vs size. In this operation we noticed that pizzas with mozzarella cheese and stuffed crust were never ordered. Same was the case with whole wheat thin dough and mozzarella and swiss cheese. Therefore, these components should not be bought or bought in very less amounts as they did not generate ANY revenue for the store. Furthermore regular dough with cheddar cheese is the most selling pizza, and must be replenished in high amounts as that would generate the highest revenue for the store.
4. Much like our roll0up analysis, the drop-down operations confirmed that the personal pizza sizes were the least bought. Whereas the large and extra large were generating the most revenue. It can thus be concluded that the large and XL pizzas are preferred by he customers and the store should prefer buying components for these size.s

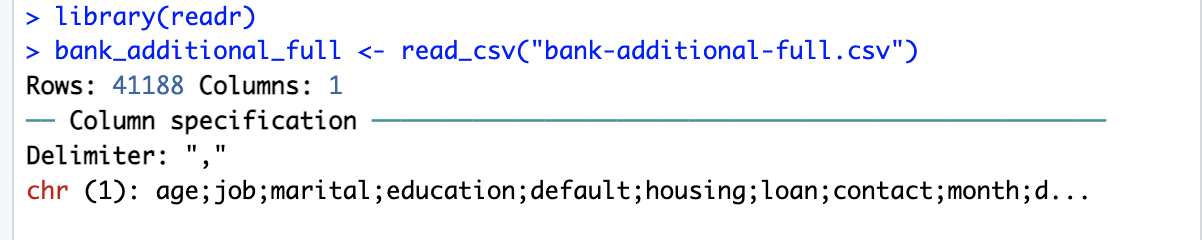
**Note:** Please note that, in Part A, question 1C and 2 (first part), the data was generated using R, as per the confirmation on email given below. We are mentioning this point in this report just to make sure there is no confusion in the assignment question and the corresponding responses presented in this document.



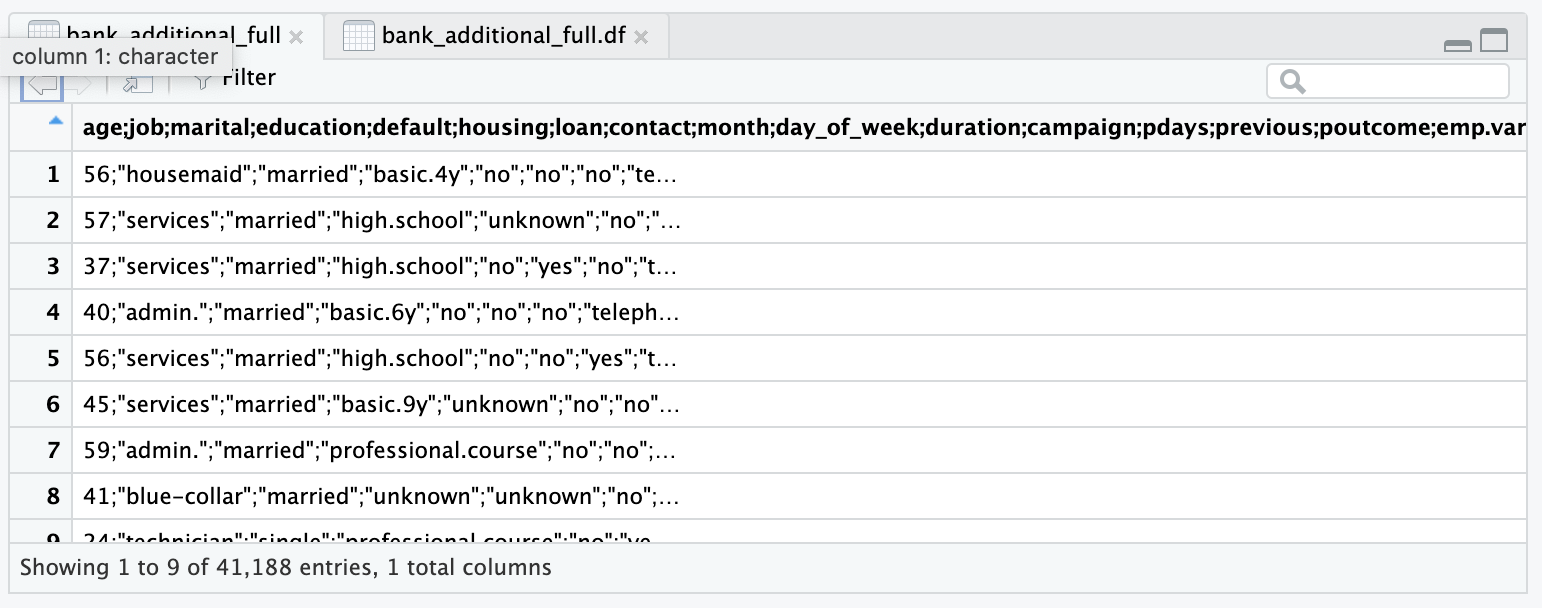
**Part B**

Prior to the tasks in the assignment, the data was imported into RStudio. Then the .csv file with delimiters, was converted to a table to ensure better readability in rows and columns. R code used for the process is shown:

Import:



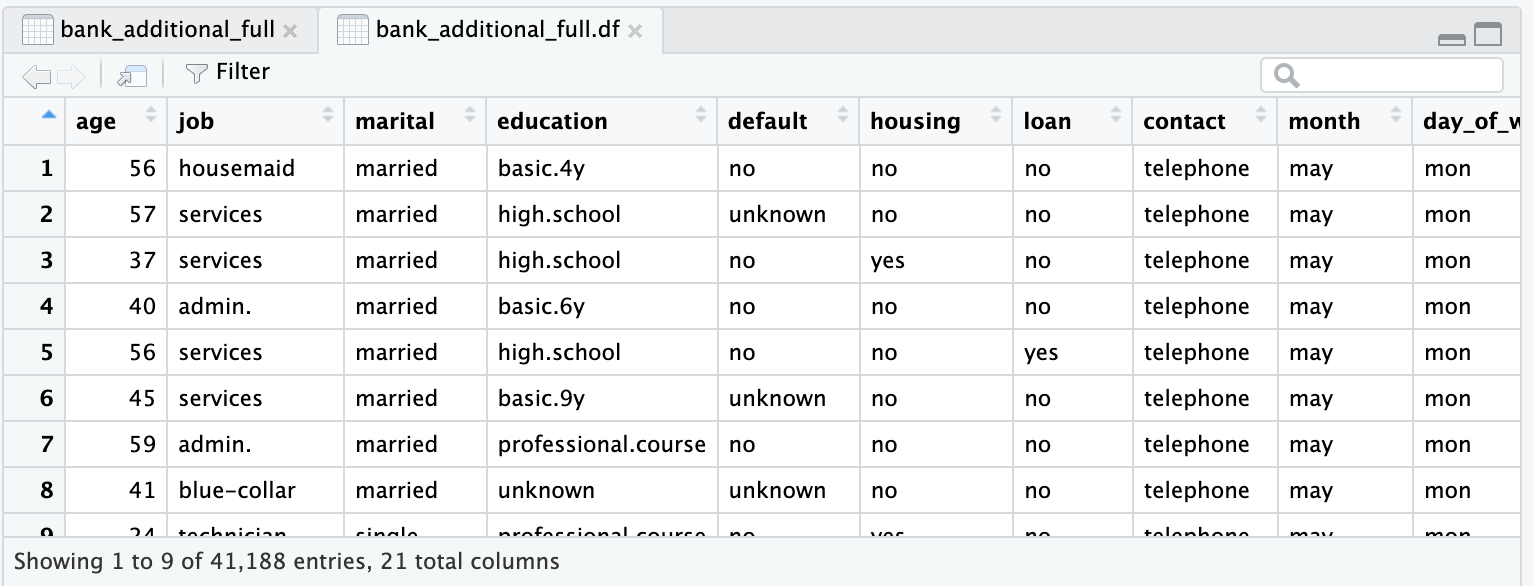
Imported data:



Converted data into a table and assigned to a data frame.

**> bank\_additional.df = read.csv(file="bank-additional-full.csv", sep=";", header=T)**

Converted data:

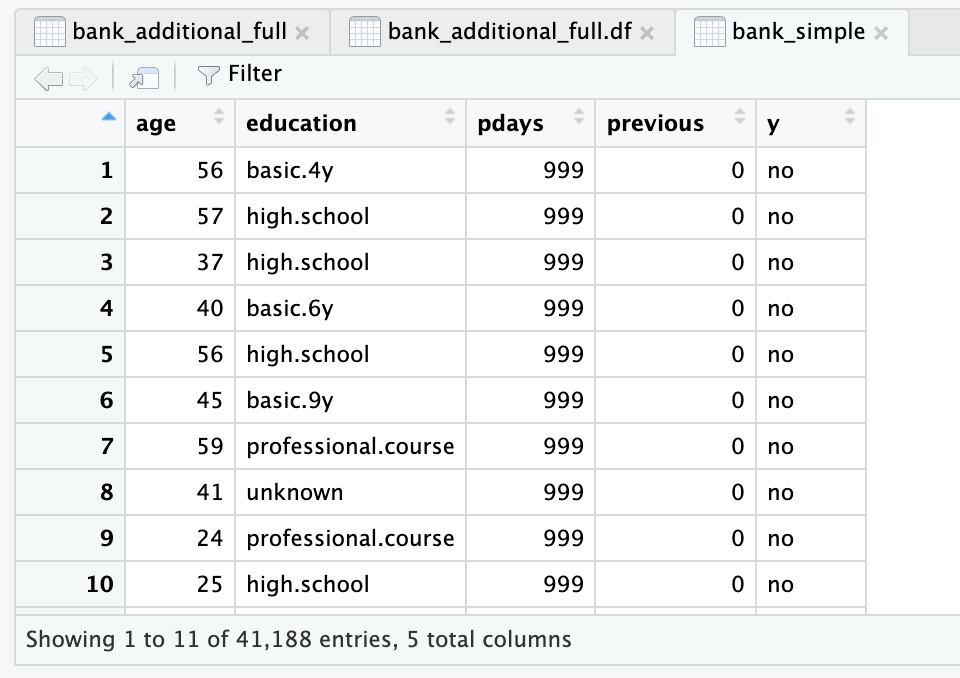


**Answers to questions:**

1. The data columns which are required were preserved and other columns were dropped.

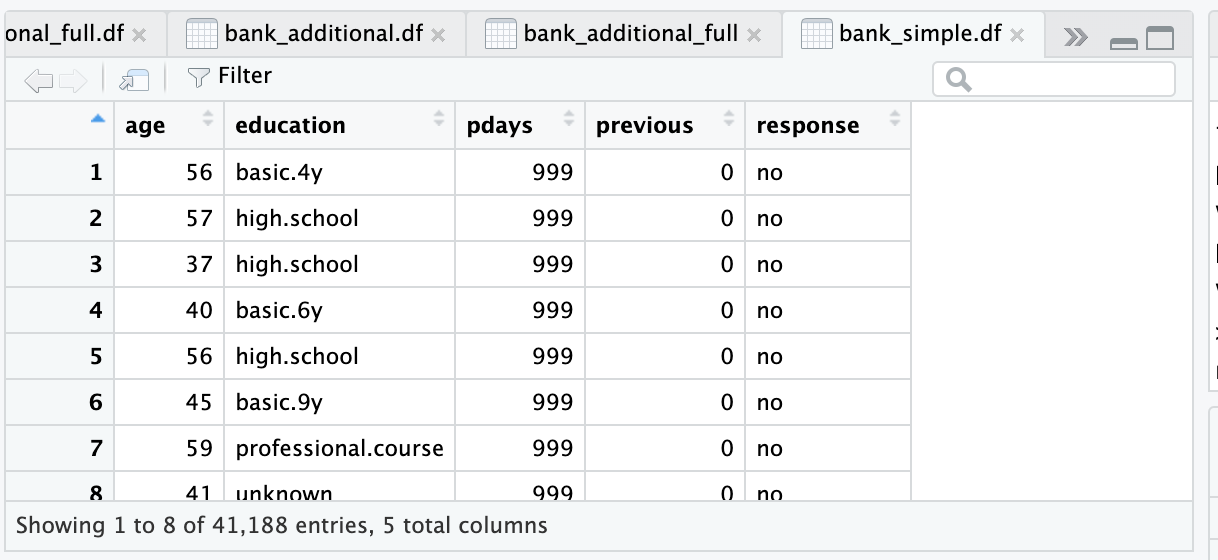
**> bank\_simple.df <- bank\_additional\_full.df[-c(2:3, 5:12, 15:20)]**

**> View(bank\_simple.df)**



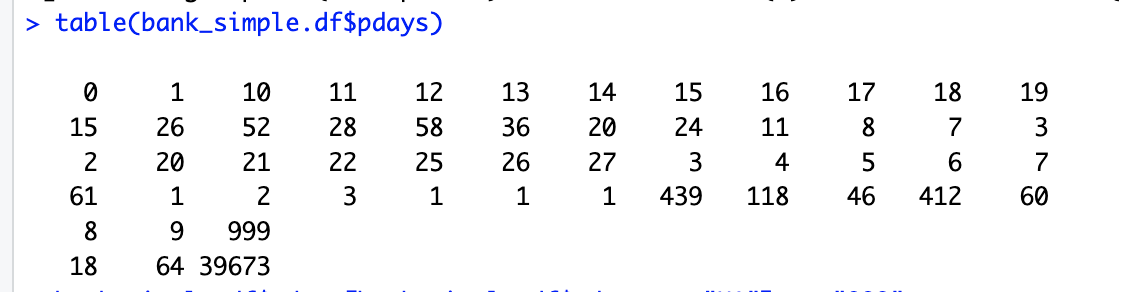
In addition, the “y” column was renamed to “response” as it was not descriptive of the data in the column.

**> names(bank\_simple.df)[names(bank\_simple.df) == "y"] <- "response"**



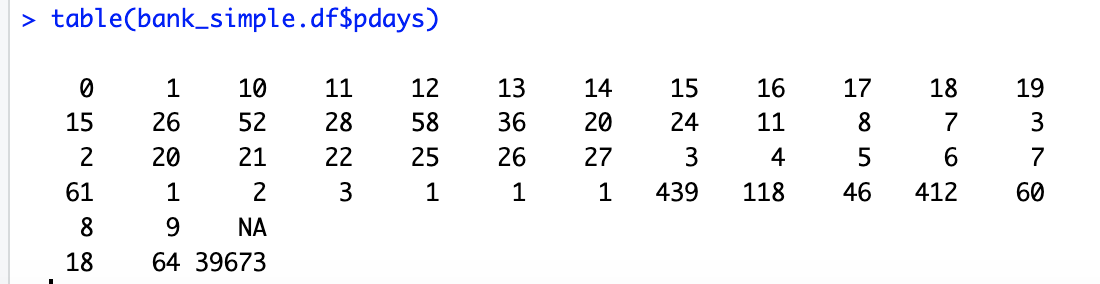
1. The value “999” is replaced with “NA”. The ‘before’ and ‘after’ output is shown below.

Before:

****

**bank\_simple.df$pdays[bank\_simple.df$pdays == "999"] <- "NA"**

After:

****

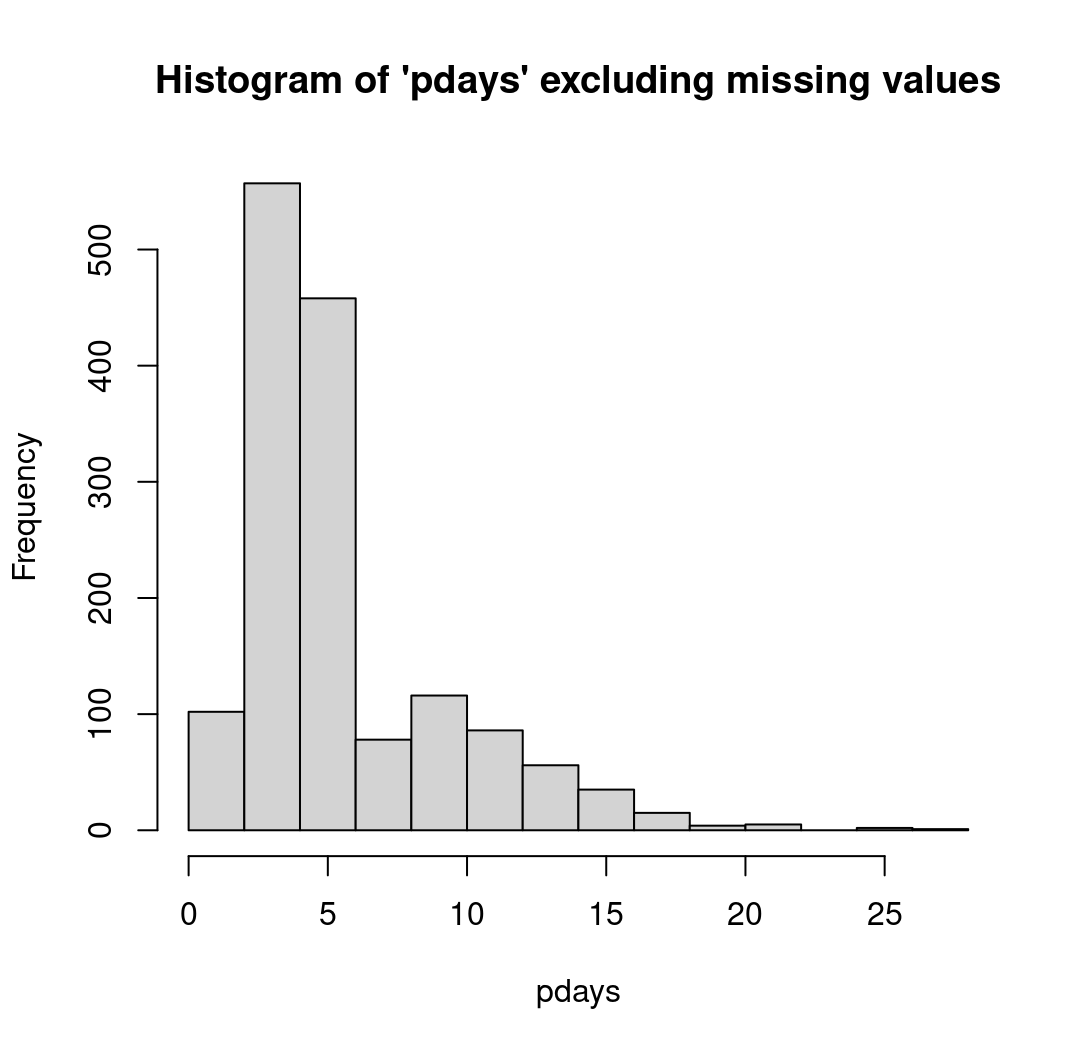
1. In reality ‘999’ denotes missing values. Because the ‘pdays’ column is numeric, having 999 in comparison to the other values which are drastically smaller, will make ‘999’ look like part of the data and affect/skew the data in calculations. For example, if the mean is calculated including ‘999’ values, the result will take an abnormal value with 999 in the calculation, whereas actually it is significantly lesser than that. Hence, it is better to substitute it with NA.
2. See the code and histogram below:

**> bank\_simple.df$pdays[bank\_simple.df$pdays == 999] <- "NA"**

**> bank\_simple.df$pdays <- as.numeric(as.character(bank\_simple.df$pdays))**

**> hist(bank\_simple.df$pdays[!is.na(bank\_simple.df$pdays)], xlab="pdays", ylab="Frequency", main="Histogram of pdays excluding missing values")**

The histogram is self-explanatory, and it shows the frequency of pdays.



1. Converting categorical values to numerical values in the education column.

**> bank\_simple.df$education[bank\_simple.df$education=="illiterate"]<-"0"**

**> bank\_simple.df$education[bank\_simple.df$education=="basic.4y"]<-"4"**

**> bank\_simple.df$education[bank\_simple.df$education=="basic.6y"]<-"6"**

**> bank\_simple.df$education[bank\_simple.df$education=="basic.9y"]<-"9"**

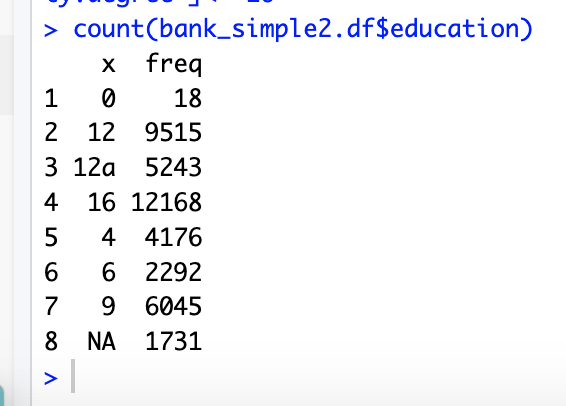
**> bank\_simple.df$education[bank\_simple.df$education=="high.school"]<-"12"**

**> bank\_simple.df$education[bank\_simple.df$education=="professional.course"]<-"12a"**

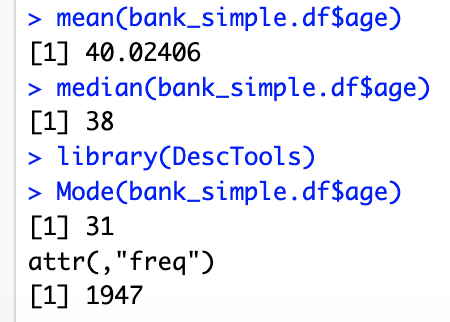
**> bank\_simple.df$education[bank\_simple.df$education=="university.degree"]<-"16"**

**> bank\_simple.df$education[bank\_simple.df$education=="unknown"]<-"NA"**

**> count(bank\_simple.df$education)**

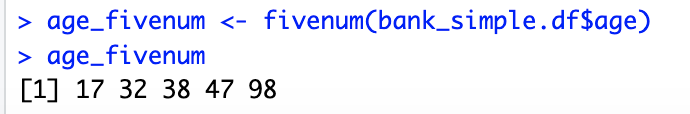
****

1. Following are the mean, median and mode calculations of the ‘age’ field:

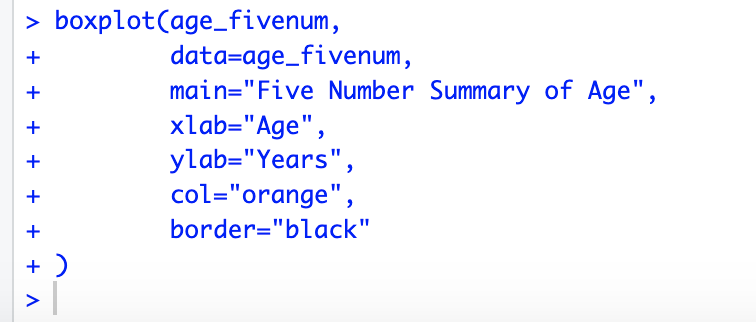


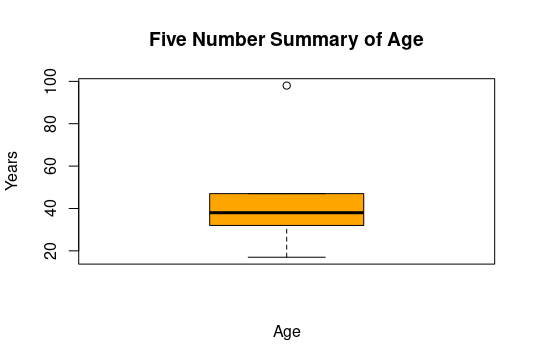
Please note that, ‘mode’ was calculated after using the library (DescTools)

Calculation of five number summary to show the data, and assigning it to a variable.

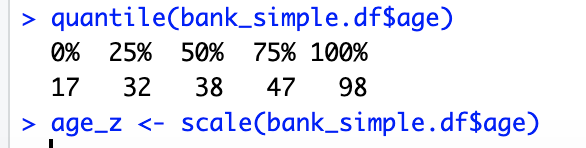


Boxplot of five number summary of age:



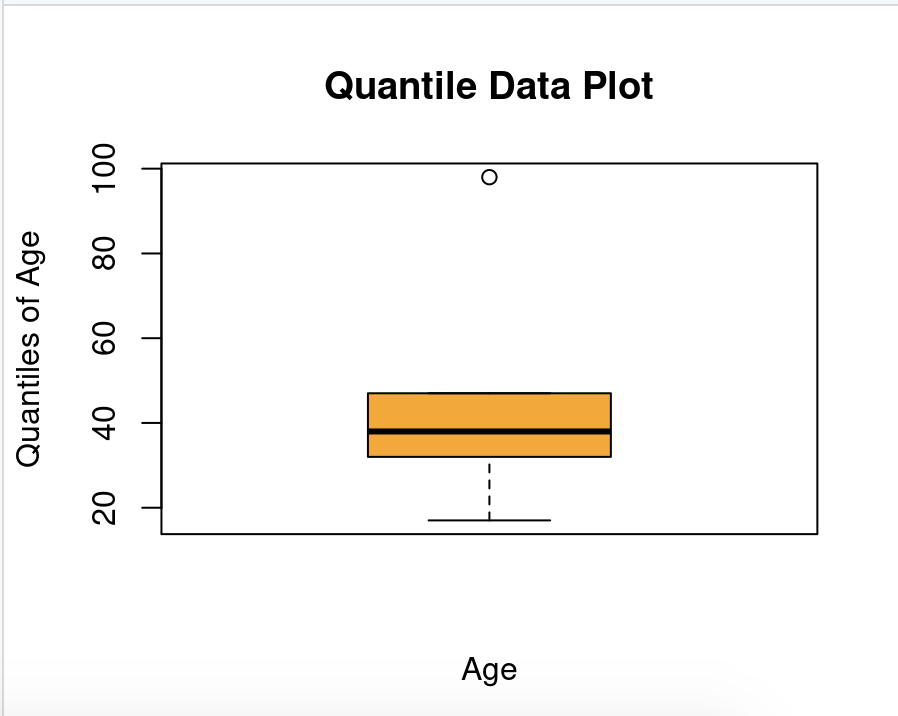


Quantile information:

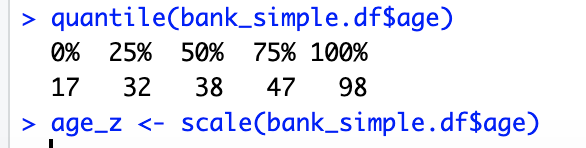


Following is the boxplot for quantiles in age.

**> boxplot(quantile\_plot, data=quantile\_plot, main="Quantile Data Plot", xlab="Age", ylab="Quantiles of Age", col="orange", border="black")**



1. Age variable standardized and saved in “age\_z”



1. Detecting outliers in age\_z.

**> age\_z <- scale(bank\_simple.df$age)**

**> out <- boxplot.stats(age\_z)$out**

**> out\_ind <- which(age\_z %in% c(out))**

**> bank\_simple.df[out\_ind, ]**

According to the output, individuals aged 70 and above are considered outliers in this dataset.

age education pdays previous response

27714 70 basic.4y NA 0 yes

27758 76 university.degree NA 0 no

27781 73 university.degree NA 1 no

27801 88 basic.4y NA 0 no

27803 88 basic.4y NA 0 yes

27806 88 basic.4y NA 0 yes

27809 88 basic.4y NA 0 no

27811 88 basic.4y NA 0 yes

27812 88 basic.4y NA 0 yes

27813 88 basic.4y NA 0 no

27814 88 basic.4y NA 0 yes

27815 88 basic.4y NA 0 no

27816 88 basic.4y NA 0 no

27817 88 basic.4y NA 0 yes

27818 88 basic.4y NA 0 yes

27819 88 basic.4y NA 0 yes

27827 95 basic.6y NA 0 no

27838 70 basic.4y NA 1 no

27839 70 basic.4y NA 1 no

27845 70 basic.4y NA 0 no

27852 77 unknown NA 0 yes

27876 75 basic.9y NA 0 no

27880 70 university.degree NA 0 no

27903 70 basic.4y NA 1 no

27931 73 university.degree NA 0 yes

27951 80 basic.4y NA 0 no

27952 80 basic.4y NA 0 no

27964 80 professional.course NA 0 yes

28221 72 basic.4y NA 0 no

28222 72 basic.4y NA 0 no

28313 82 unknown NA 0 yes

28457 73 basic.4y NA 0 yes

28505 71 basic.4y NA 0 no

28531 70 basic.4y NA 0 yes

28541 70 basic.4y NA 1 yes

28587 70 basic.4y NA 0 no

28620 71 high.school NA 0 no

28733 70 unknown NA 0 yes

28774 70 unknown NA 0 no

29226 71 unknown NA 0 yes

29264 75 basic.4y NA 0 no

29499 73 basic.4y 6 1 no

29626 73 basic.4y NA 0 no

29669 71 university.degree NA 0 yes

29683 75 basic.4y NA 0 yes

29974 75 basic.4y NA 0 no

29978 78 basic.4y NA 0 yes

29982 75 basic.4y NA 1 yes

29988 70 basic.6y NA 0 no

29991 78 basic.4y NA 0 no

30001 75 basic.4y NA 1 yes

30005 78 basic.4y NA 0 yes

30007 85 basic.4y NA 0 yes

30073 85 basic.4y NA 0 no

30079 85 basic.4y NA 0 no

30080 80 high.school NA 3 no

30089 71 university.degree NA 0 no

30104 85 basic.4y NA 0 no

30111 85 basic.4y NA 0 no

30134 79 basic.9y NA 0 yes

30172 77 basic.4y NA 1 no

30215 83 basic.4y NA 0 no

30226 81 professional.course NA 0 no

30228 71 university.degree 5 1 no

30242 81 professional.course NA 0 no

30335 73 university.degree NA 1 no

30336 71 basic.4y NA 1 no

30391 71 basic.4y NA 0 no

30431 88 high.school NA 0 no

30461 81 basic.9y NA 1 no

30590 81 basic.4y NA 0 yes

35834 81 unknown NA 0 yes

35849 71 unknown NA 1 no

35857 83 basic.9y 3 3 no

35879 75 basic.4y NA 0 no

35974 78 high.school 3 2 no

36184 88 basic.4y NA 0 no

36286 77 basic.9y NA 1 no

36312 72 university.degree NA 0 no

36384 79 high.school NA 1 no

36385 79 high.school NA 1 no

36817 74 basic.4y NA 0 yes

36999 75 basic.4y NA 0 no

37137 72 university.degree NA 0 no

37138 72 university.degree NA 0 no

37171 70 basic.4y NA 0 no

37187 79 unknown NA 0 no

37191 74 high.school NA 0 no

37193 74 high.school NA 0 no

37194 74 high.school NA 1 yes

37196 74 high.school NA 0 no

37207 76 basic.4y NA 0 yes

37208 76 basic.4y NA 0 yes

37214 82 university.degree NA 2 no

37220 75 basic.4y NA 0 yes

37228 70 basic.4y NA 0 yes

37236 73 basic.4y NA 0 yes

37238 73 basic.9y 15 1 no

37240 73 basic.4y NA 0 no

37258 73 basic.4y NA 0 no

37261 76 university.degree NA 0 yes

37317 70 basic.4y NA 1 no

37342 85 professional.course NA 0 yes

37356 80 illiterate 6 1 yes

37372 70 high.school NA 0 no

37404 74 university.degree NA 1 yes

37455 74 professional.course NA 0 no

37456 76 basic.4y NA 0 no

37473 88 basic.4y NA 0 no

37480 74 basic.4y NA 1 no

37494 81 basic.6y 4 1 no

37506 76 basic.6y NA 1 no

37510 74 professional.course NA 1 yes

37513 76 university.degree NA 1 no

37526 73 professional.course NA 0 no

37533 72 basic.4y NA 1 no

37546 70 professional.course NA 1 no

37569 71 basic.4y NA 0 yes

37571 70 professional.course NA 0 yes

37587 70 professional.course NA 0 no

37598 76 professional.course 15 1 yes

37602 72 basic.4y NA 0 no

37603 73 basic.4y NA 0 yes

37605 80 high.school NA 0 yes

37636 74 basic.9y NA 1 yes

37662 71 basic.4y NA 0 yes

37676 74 basic.4y 13 1 yes

37680 80 basic.4y NA 0 no

37691 74 university.degree NA 0 yes

37693 73 basic.4y NA 0 no

37716 74 basic.9y NA 1 no

37717 71 university.degree 6 2 yes

37736 76 basic.4y NA 0 no

37737 76 basic.4y NA 1 no

37744 87 basic.4y NA 0 yes

37757 79 basic.4y 3 2 yes

37766 70 university.degree NA 1 no

37770 74 basic.4y 6 1 yes

37776 88 university.degree NA 0 no

37785 81 high.school NA 0 no

37819 80 basic.4y NA 0 yes

37820 80 basic.4y 3 2 no

37821 78 basic.4y NA 0 no

37826 71 university.degree NA 0 no

37827 71 university.degree NA 0 no

37862 72 unknown NA 1 yes

37869 73 university.degree NA 0 no

37871 73 basic.4y NA 1 yes

37874 73 high.school NA 0 no

37906 79 basic.9y NA 1 yes

37921 72 unknown NA 0 no

37936 71 basic.4y NA 0 no

37947 83 unknown NA 0 no

37952 76 unknown NA 0 no

37953 76 unknown NA 0 yes

37955 72 basic.4y 3 2 yes

37959 71 professional.course NA 0 no

37998 71 basic.4y NA 0 yes

38000 76 basic.4y NA 0 no

38006 75 unknown NA 1 yes

38008 71 basic.4y NA 0 no

38020 78 unknown NA 0 no

38021 78 unknown NA 1 yes

38023 91 university.degree NA 2 no

38033 91 university.degree NA 0 no

38034 76 basic.4y 3 1 yes

38046 73 basic.4y NA 0 no

38053 76 university.degree NA 1 no

38055 73 basic.4y NA 0 no

38061 71 unknown NA 0 yes

38066 83 unknown NA 0 no

38072 70 basic.4y 3 2 yes

38075 70 basic.4y NA 2 yes

38082 70 basic.4y 3 1 yes

38089 70 basic.4y 3 1 no

38126 70 high.school NA 0 no

38128 70 university.degree NA 0 yes

38130 70 university.degree NA 0 no

38137 81 basic.4y NA 0 yes

38145 70 basic.4y NA 1 no

38146 70 basic.4y NA 0 no

38167 78 basic.9y NA 0 no

38170 71 basic.4y NA 2 no

38176 71 basic.4y NA 1 yes

38179 75 basic.4y NA 0 yes

38180 83 professional.course 4 1 yes

38184 71 basic.9y NA 0 no

38185 82 university.degree NA 0 yes

38192 82 university.degree NA 1 no

38193 82 university.degree NA 0 no

38194 80 basic.4y NA 0 no

38196 80 basic.4y NA 0 no

38207 86 basic.4y NA 0 no

38230 77 unknown NA 0 no

38242 75 basic.9y NA 0 no

38247 77 university.degree NA 0 yes

38248 70 university.degree NA 1 no

38253 80 basic.4y NA 1 no

38254 71 university.degree NA 0 no

38255 71 university.degree NA 0 no

[ reached 'max' / getOption("max.print") -- omitted 269 rows ]