DTI5126[EG] Data Science Applications

20215

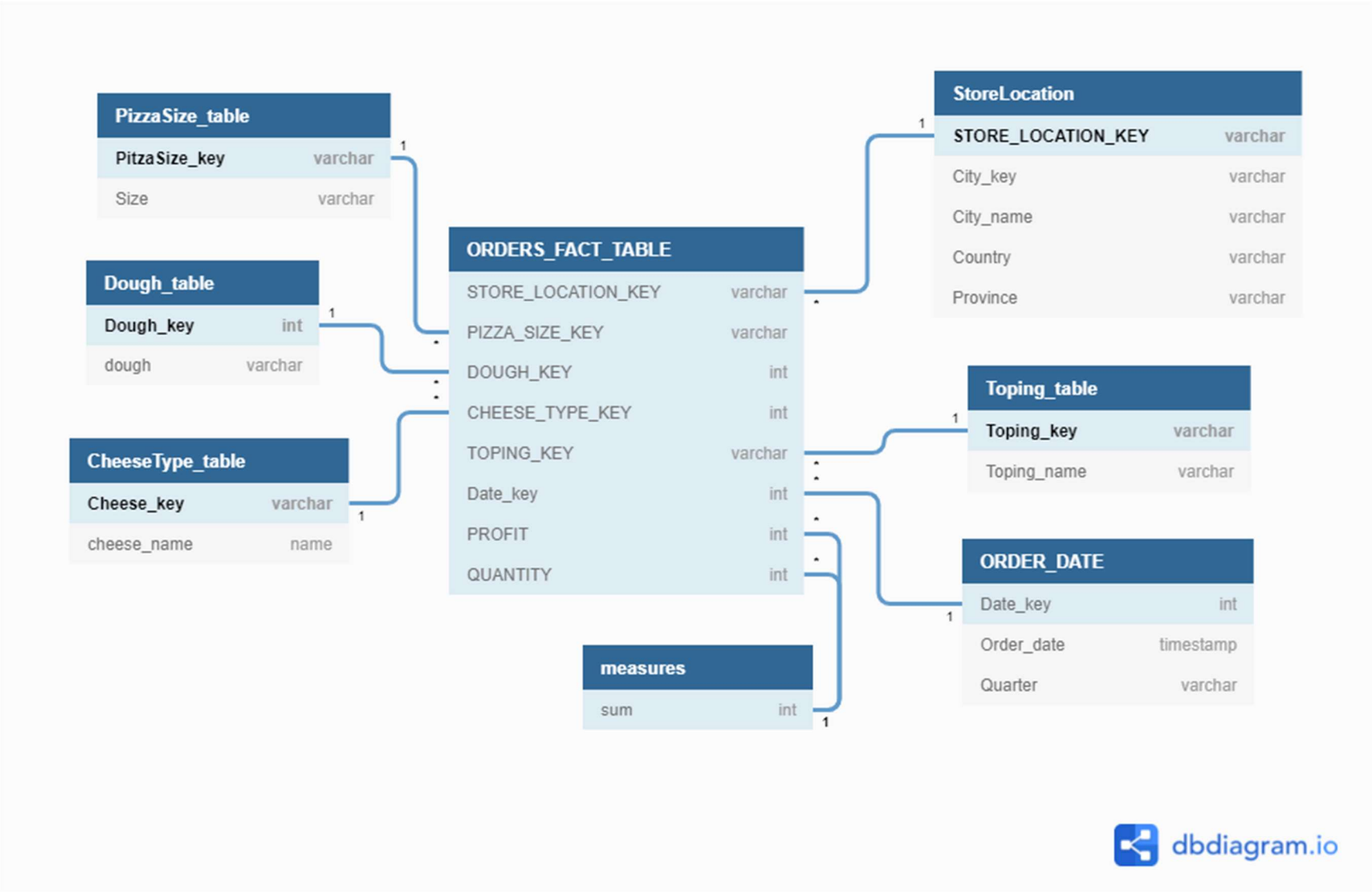
Assignment one

By

Omar Sorour

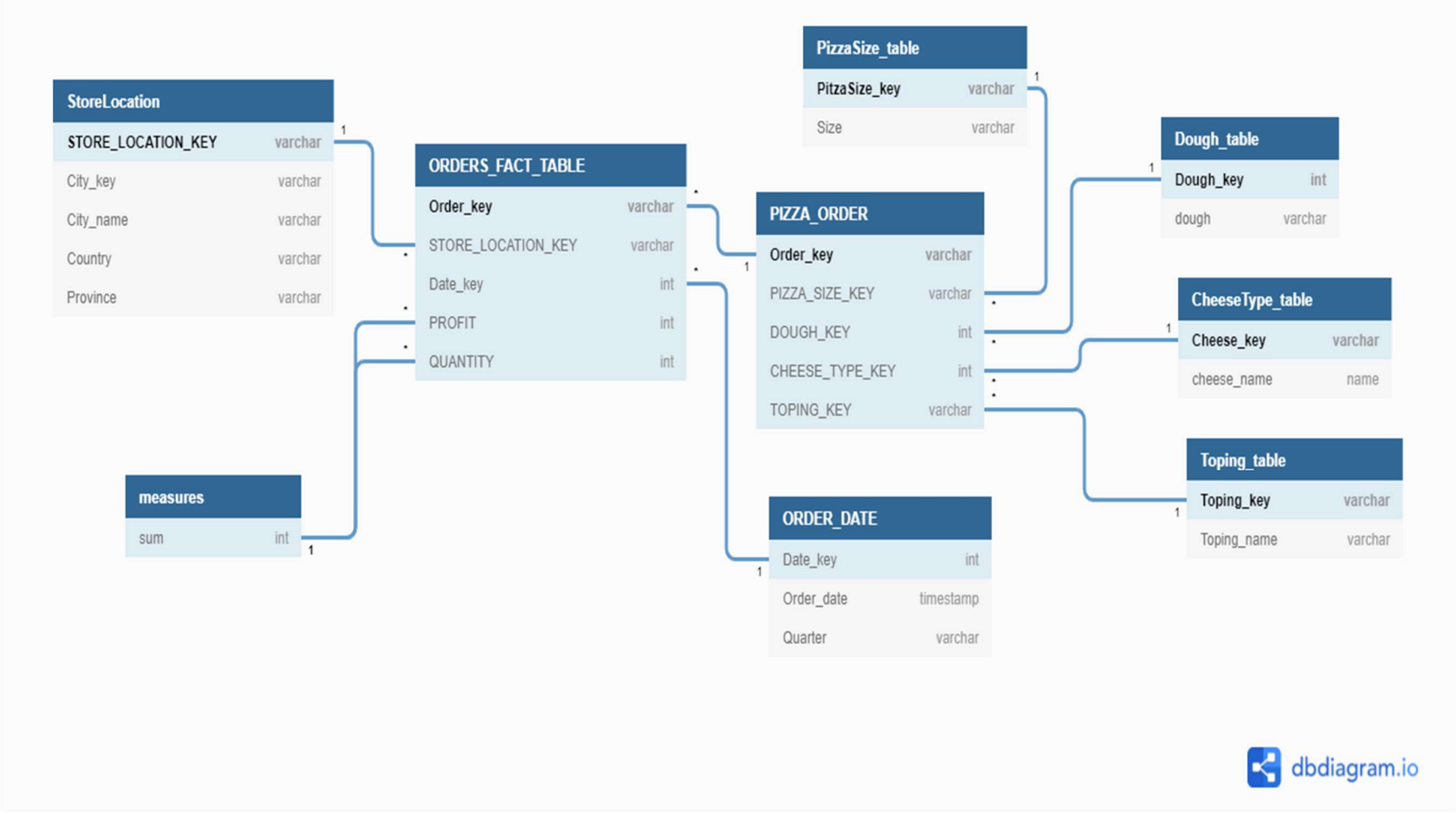
Part A Data warehousing and OLAP

(a) Star schema representing the listed problem



Part A Data warehousing and OLAP

(b) Generate a snowflake schema that represents this problem



(I used this site https://dbdiagram.io/d to create these two schemas)

(C) **Generate a set of sample data stored in csv files for the dimensions and fact table for the snowflake schema in c.**

This has been implemented in the attached R file that is named "dimensions and fact table.R"

(2) Using R, read the dimensions files and the profit fact table. Build an OLAP cube for your revenue and show the cells of a subset of the cells

The Cube has been created as following:

```
#Using R, read the dimensions files and the profit fact table. Build an OLAP cube for your re


revenue_cube_profit <-
  tapply(big_fact_table$Profit,
         (big_fact_table[,c("StoreLocation_key_ref","Date_key_ref")]
          ),
         FUN=function(x){return(sum(x))})

# Showing the cells of the cube
revenue_cube
```

Showing the subset of cells:

```
> revenue_cube
, , Profit = 6, Date_key_ref = 1, PitzaSize_key_ref = L, Toping_key_ref = t1

                     Quantity
StoreLocation_key_ref  1  2  3  4  5
                  S10 NA NA NA NA NA
                  S20 NA NA NA NA NA
                  S30 NA NA NA NA  5

, , Profit = 7, Date_key_ref = 1, PitzaSize_key_ref = L, Toping_key_ref = t1

                     Quantity
StoreLocation_key_ref  1  2  3  4  5
                  S10 NA NA NA NA NA
                  S20 NA NA NA NA NA
                  S30 NA NA NA NA NA

, , Profit = 8, Date_key_ref = 1, PitzaSize_key_ref = L, Toping_key_ref = t1
```

(3) Suppose that we want to examine the data of the above store to find trends and thus to predict which Pizza components the store should order more of.

In this scenario we can rollup on the Pizza toping dimension to see which toping was mostly order then drill down to each store because we have 3 stores. I did this one time for the toping, cheese type, and the dough type.

These figures show the results:

The output of the roll up step:

```
> apply(revenue_cube_quantity, c("Toping_key_ref"),FUN=function(x) {return(sum(x, na.rm=TRUE))})
 t1  t2  t3  t4
330 435 372 340
> apply(revenue_cube_quantity, c("Dough_key_ref"),FUN=function(x) {return(sum(x, na.rm=TRUE))})
  1   2   3
497 450 530
> apply(revenue_cube_quantity, c("Cheese_key_ref"),FUN=function(x) {return(sum(x, na.rm=TRUE))})
  1   2   3
526 526 425
>
```

The output of the drill down step:

```
> #apply(revenue_cube, c("Toping_key_ref"),FUN=function(x) {return(sum(x, na.rm=TRUE))})
> apply(revenue_cube_quantity, c("Toping_key_ref","StoreLocation_key_ref"),FUN=function(x) {return
(sum(x, na.rm=TRUE))})
              StoreLocation_key_ref
Toping_key_ref S10 S20 S30
            t1 134  71 125
            t2 129 142 164
            t3 120 154  98
            t4 160  84  96
> apply(revenue_cube_quantity, c("Dough_key_ref","StoreLocation_key_ref"),FUN=function(x) {return(s
um(x, na.rm=TRUE))})
             StoreLocation_key_ref
Dough_key_ref S10 S20 S30
            1 192 142 163
            2 145 145 160
            3 206 164 160
> apply(revenue_cube_quantity, c("Cheese_key_ref","StoreLocation_key_ref"),FUN=function(x) {return
(sum(x, na.rm=TRUE))})
              StoreLocation_key_ref
Cheese_key_ref S10 S20 S30
             1 230 140 156
             2 187 181 158
             3 126 130 169
```

(3) Describe a series of drill-down and roll-up operations that would lead to the conclusion that customers are beginning to prefer bigger pizzas.

In this scenario we should:

1- Roll up on the Pizza size dimension.

2- Drill down on the date dimension.

That way we will have a table of all the sizes that have been mostly purchased by the users in a specified duration after that we will decide whether the customers begin to prefer the large size or not. In my case the data was randomly selected and the most popular size was the XLarge size. We can extend this to show which size is popular by each store or branch as in the

```
> apply(revenue_cube_quantity, c("PitzaSize_key_ref"), FUN=function(x) {return(sum(x, na.rm=TRU
E))})
  L   M   P   S  XL
284 202  75 266 650
>
> apply(revenue_cube_quantity, c("PitzaSize_key_ref","Date_key_ref"), FUN=function(x) {return(sum
(x, na.rm=TRUE))})
                 Date_key_ref
PitzaSize_key_ref  1  2  3  4  5  6  7  8  9 10
               L  24 18 45 26 16 15 22 49 30 39
               M  25 16 38 23 21 17 15 15 18 14
               P  10 18 11  7  1  8  5  1 14  0
               S  30 22 24 19 17 22 32 37 44 19
               XL 52 38 69 52 77 95 83 50 58 76
```

We can extend this to show which size is popular by each store or branch as in this figure:

```
> apply(revenue_cube_quantity, c("PitzaSize_key_ref","StoreLocation_key_ref"), FUN=function(x) {ret
urn(sum(x, na.rm=TRUE))})
                 StoreLocation_key_ref
PitzaSize_key_ref S10 S20 S30
               L   96  91  97
               M   83  45  74
               P   18  39  18
               S   91  91  84
               XL 255 185 210
>
```

Part B Data Preparation

1- **Import the data set into RStudio and reduce the dataset to only four predictors (age, education, previous, and pdays), and the target, response.**

```
#Importing the data
setwd("F:\\uOttawa\\Fundamental\\Assignment\\Part B\\bank-additional")
bank_addtional_full <- read.delim(file="bank-additional-full.csv", header=TRUE, sep=";")

#dataset contains only four predictors (age, education, previous, and pdays)
reqd <- as.vector(c("age", "education","previous","pdays"))

Results <- bank_addtional_full[ ,reqd]
```

Results:

	age	education	previous	pdays
1	56	basic.4y	0	*NA*
2	57	high.school	0	*NA*
3	37	high.school	0	*NA*
4	40	basic.6y	0	*NA*
5	56	high.school	0	*NA*
6	45	basic.9y	0	*NA*
7	59	professional.course	0	*NA*
8	41	unknown	0	*NA*
9	24	professional.course	0	*NA*
10	25	high.school	0	*NA*
11	41	unknown	0	*NA*
12	25	high.school	0	*NA*
13	29	high.school	0	*NA*
14	57	basic.4y	0	*NA*

2- **The field pdays is a count of the number of days since the client was last contacted from a previous campaign. The code 999 in the value represents customers who had not been contacted previously. Change the field value 999 to "NA" to represent missing values.**

```
Results[which(Results[,4]==999, arr.ind=TRUE), 4] <- NA
```

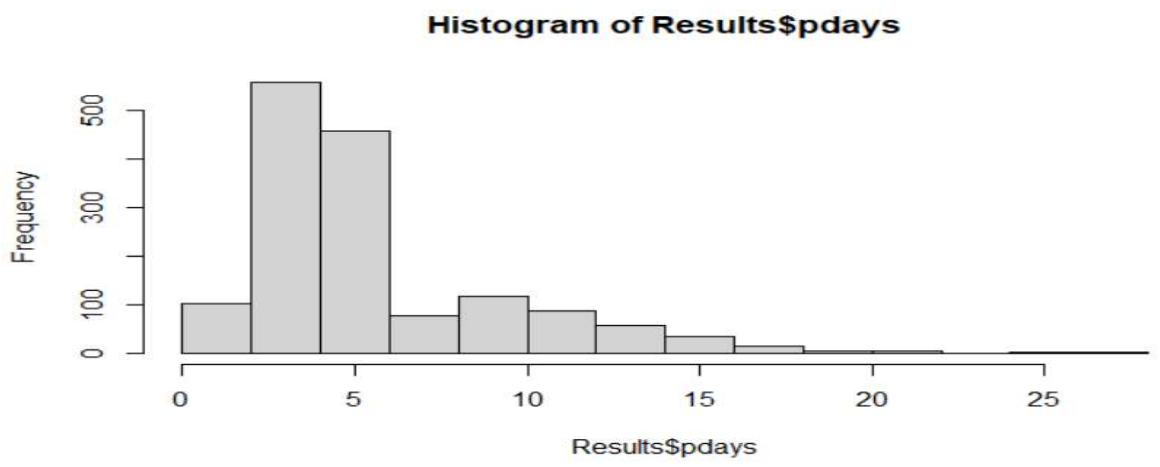
Result:

pdays
NA
NA
NA
NA
NA
NA
NA
NA
NA

3- Explain why the field pdays is essentially useless until you handle the 999 code

The pdays is useless without handling the code 999 because if we were to do any calculations on this column, this code would be treated as a numeric value that has its weight so this will be mis leading**.**

4- Create a histogram of the pdays variable showing the missing value excluded.




```
sum(is.na(Results$pdays))
```

The number of NA values is 39673

5- Transform the data values of the education field into numeric values using the chart in Table 1 below.

```
Results$education <- factor(Results$education,levels=c("illiterate","basic.4y","basic.6y",
                                                       "basic.9y","high.school",
                                                       "professional.course",
                                                       "university.degree","unknown"),
                                                labels = c(0,4,6,9,12,12,16,NA))
```

Results:

education
4
12
12
6
12
9
12
NA
12
12
NA
12
12
4

6- **Compute the mean, median & mode of the age variable. Using a boxplot, give the five-number summary of the data. Plot the quantile information.The following figure describes how to calculate the mean, the median and the mode.**

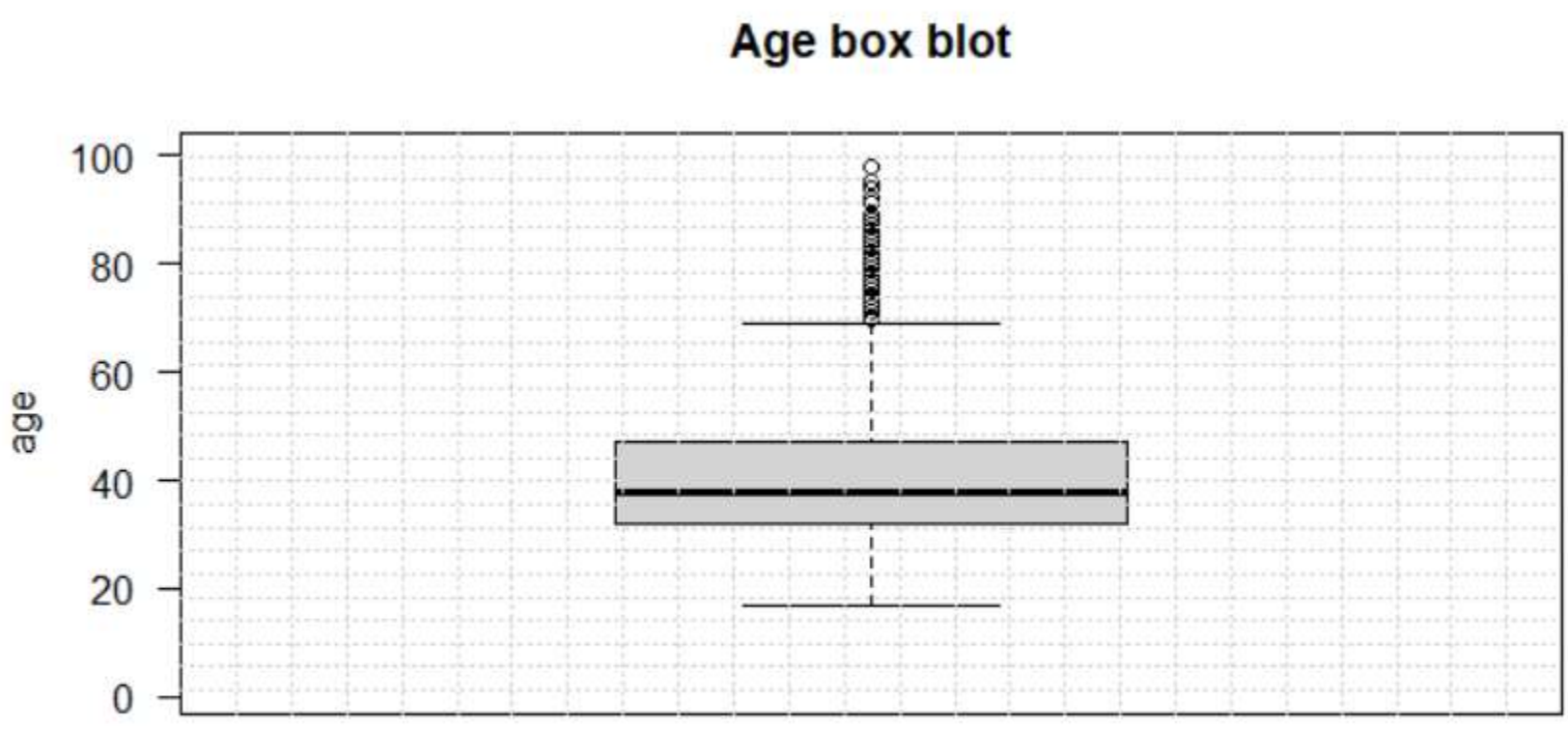
```
mean_age <- mean(Results$age)
mean_age
median_age <- median(Results$age)
median_age
#creating a function to get the mode
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
age_mode <- getmode(Results$age)
age_mode
```

These were the obtained results:

```
> age_mode
[1] 31
> mean_age <- mean(Results$age)
> mean_age
[1] 40.02406
> median_age <- median(Results$age)
> median_age
[1] 38
```

Plotting the boxplot of the variable age:

```
boxplot(Results$age, main="Age box blot", ylab="age",ylim=c(1,100),las=1) #Box plot of the da
```



For the figure the five number salaries are

The minimum value is: 17

The maximum value is: 69 (including the outliers the maximum is 98)

Median is: 38

Q1 = 32

Q2= 47

7- **Some machine learning algorithms perform better when the numeric fields are standardized. Standardize the age variable and save it as a new variable, age_z.**

```
#Creating the new field age_z and adding it to the dataframe
age_z <- scale(Results$age)
mean(age_z)
Results$age_z <- age_z
```

Results:

age_z
1.533015677
1.628973456
-0.290182119
-0.002308783
1.533015677
0.477480111
1.820889013
0.093648996
-1.537633243
-1.441675464

8- **Obtain a listing of all records that are outliers according to the field age_z.**

```
# Obtaining a listing of all records that are outliers according to the field age_z.
lower_bound <- quantile(Results$age_z, 0.01)
upper_bound <- quantile(Results$age_z, 0.99)

outlier_ind <- which(Results$age_z < lower_bound | Results$age_z > upper_bound)

outliers_record <- Results[outlier_ind, ]
```

Results:

	age	education	previous	pdays	y	age_z
38453	98		2	2	yes	5.563242
38456	98		0	NA	yes	5.563242
27827	95		0	NA	no	5.275369
38922	94		1	NA	no	5.179411
39656	92	UA	2	6	no	4.987496
39735	92	UA	1	NA	yes	4.987496
40451	92	UA	1	3	yes	4.987496
40470	92	UA	4	3	yes	4.987496
38023	91	6	2	NA	no	4.891538
38033	91	6	0	NA	no	4.891538

As we can see here, we got a new table that contains both lower and higher outliers.