Deliverable 4

Final deliverable

Júlia Gasull i Claudia Sánchez

January 4, 2021

Contents

1 First setups

```
if(!is.null(dev.list())) dev.off()  # Clear plots
rm(list=ls())                       # Clean workspace
```

1.1 Some useful functions

```
calcQ <- function(x) { # Function to calculate the different quartiles
  s.x <- summary(x)
  iqr<-s.x[5]-s.x[2]
  list(souti=s.x[2]-3*iqr, mouti=s.x[2]-1.5*iqr, min=s.x[1], q1=s.x[2], q2=s.x[3],
       q3=s.x[5], max=s.x[6], mouts=s.x[5]+1.5*iqr, souts=s.x[5]+3*iqr )
}

countNA <- function(x) { # Function to count the NA values
  mis_x <- NULL
  for (j in 1:ncol(x)) {mis_x[j] <- sum(is.na(x[,j])) }
  mis_x <- as.data.frame(mis_x)
  rownames(mis_x) <- names(x)
  mis_i <- rep(0,nrow(x))
  for (j in 1:ncol(x)) {mis_i <- mis_i + as.numeric(is.na(x[,j])) }
  list(mis_col=mis_x,mis_ind=mis_i)
}

countX <- function(x,X) { # Function to count a specific number of appearences
  n_x <- NULL
  for (j in 1:ncol(x)) {n_x[j] <- sum(x[,j]==X) }
  n_x <- as.data.frame(n_x)
  rownames(n_x) <- names(x)
  nx_i <- rep(0,nrow(x))
```

```
  for (j in 1:ncol(x)) {nx_i <- nx_i + as.numeric(x[,j]==X) }
  list(nx_col=n_x,nx_ind=nx_i)
}
```

2 Data description

- Description http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml
- Data Dictionary - SHL Trip Records -This data dictionary describes SHL trip data in visit http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

2.1 Variables

- VendorID
  - A code indicating the LPEP provider that provided the record.
  - Values:
    - 1= Creative Mobile Technologies, LLC
    - 2= VeriFone Inc.
- lpep_pickup_datetime
  - The date and time when the meter was engaged.
- lpep_dropoff_datetime
  - The date and time when the meter was disengaged.
- Passenger_count
  - The number of passengers in the vehicle.
  - This is a driver-entered value.
- Trip_distance
  - The elapsed trip distance in miles reported by the taximeter.
- Pickup_longitude
  - Longitude where the meter was engaged.
- Pickup_latitude
  - Latitude where the meter was engaged.
- RateCodeID
  - The final rate code in effect at the end of the trip.
  - Values:
    - 1=Standard rate
    - 2=JFK
    - 3=Newark
    - 4=Nassau or Westchester
    - 5=Negotiated fare
    - 6=Group ride
- Store_and_fwd_flag
  - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka “store and forward,” because the vehicle did not have a connection to the server:
  - Values
    - Y= store and forward trip
    - N= not a store and forward trip
- Dropoff_longitude
  - Longitude where the meter was timed off.
- Dropoff_latitude
  - Latitude where the meter was timed off.

- Payment_type
  - A numeric code signifying how the passenger paid for the trip.
  - Values:
    - 1= Credit card
    - 2= Cash
    - 3= No charge
    - 4= Dispute
- Fare_amount
  - The time-and-distance fare calculated by the meter.

- Extra
  - Miscellaneous extras and surcharges.

  - Currently, this only includes the $0.50 and $1 rush hour and overnight charges.
- MTA_tax
  - $0.50 MTA tax that is automatically triggered based on the metered rate in use.

- Improvement_surcharge
  - $0.30 improvement surcharge assessed on hailed trips at the flag drop.
  - The improvement surcharge began being levied in 2015.

- Tip_amount
  - This field is automatically populated for credit card tips.
  - Cash tips are not included.

- Tolls_amount
  - Total amount of all tolls paid in trip.

- Total_amount
  - The total amount charged to passengers.
  - Does not include cash tips.

- Trip_type
  - A code indicating whether the trip was a street-hail or a dispatch that is automatically assigned based on the metered rate in use but can be altered by the driver.
  - Values:
    - 1= Street-hail
    - 2= Dispatch

3 Load Required Packages for this deliverable

We load the necessary packages and set working directory

```
setwd("~/Documents/uni/FIB-ADEI-LAB/deliverable4")
filepath<-"~/Documents/uni/FIB-ADEI-LAB/deliverable4"

options(contrasts=c("contr.treatment","contr.treatment"))
requiredPackages <- c("missMDA","chemometrics","mvoutlier","effects","FactoMineR","car","lmtest","ggplot
missingPackages <- requiredPackages[!(requiredPackages %in% installed.packages()[,"Package"])]
if(length(missingPackages)) install.packages(missingPackages)
lapply(requiredPackages, require, character.only = TRUE)
```

4 Select a sample of 5000 records

From the proposed database, we need to select a sample of 5000 records randomly so we can start analyzing our data.

!!!!! PER DESCOMENTAR AL FINAL

```
#df<-read.table(paste0(filepath,"/green_tripdata_2016-01.csv"),header=T, sep=",")
#set.seed(180998)
#sam<-as.vector(sort(sample(1:nrow(df),5000)))
#df<-df[sam,]
```

!!! ESBORRAR AL DFINAL

```
load(paste0(filepath,"/Taxi5000_raw.RData"))
```

5 Rename variables and clean data

```
summary(df)
```

```
##     VendorID     lpep_pickup_datetime Lpep_dropoff_datetime Store_and_fwd_flag
##  Min.   :1.000   Length:5000          Length:5000           Length:5000
##  1st Qu.:2.000   Class :character     Class :character      Class :character
##  Median :2.000   Mode  :character     Mode  :character      Mode  :character
##  Mean   :1.788
##  3rd Qu.:2.000
##  Max.   :2.000
##    RateCodeID  Pickup_longitude Pickup_latitude Dropoff_longitude
##  Min.   :1.0   Min.   :-75.39   Min.   : 0.00   Min.   :-75.31
##  1st Qu.:1.0   1st Qu.:-73.96   1st Qu.:40.70   1st Qu.:-73.97
##  Median :1.0   Median :-73.95   Median :40.75   Median :-73.94
##  Mean   :1.1   Mean   :-73.89   Mean   :40.72   Mean   :-73.80
##  3rd Qu.:1.0   3rd Qu.:-73.92   3rd Qu.:40.80   3rd Qu.:-73.91
##  Max.   :5.0   Max.   :  0.00   Max.   :41.04   Max.   :  0.00
##  Dropoff_latitude Passenger_count Trip_distance     Fare_amount
##  Min.   : 0.00    Min.   :0.000   Min.   : 0.000   Min.   :-52.0
##  1st Qu.:40.70    1st Qu.:1.000   1st Qu.: 1.020   1st Qu.:  6.0
##  Median :40.75    Median :1.000   Median : 1.800   Median :  9.0
##  Mean   :40.67    Mean   :1.375   Mean   : 2.765   Mean   : 11.9
##  3rd Qu.:40.79    3rd Qu.:1.000   3rd Qu.: 3.420   3rd Qu.: 14.5
##  Max.   :41.18    Max.   :6.000   Max.   :52.790   Max.   :200.0
##      Extra            MTA_tax          Tip_amount      Tolls_amount
##  Min.   :-1.0000   Min.   :-0.5000   Min.   : 0.000   Min.   : 0.00000
##  1st Qu.: 0.0000   1st Qu.: 0.5000   1st Qu.: 0.000   1st Qu.: 0.00000
##  Median : 0.5000   Median : 0.5000   Median : 0.000   Median : 0.00000
##  Mean   : 0.3517   Mean   : 0.4857   Mean   : 1.217   Mean   : 0.08369
##  3rd Qu.: 0.5000   3rd Qu.: 0.5000   3rd Qu.: 2.000   3rd Qu.: 0.00000
##  Max.   : 1.0000   Max.   : 0.5000   Max.   :96.000   Max.   :18.04000
##  Ehail_fee      improvement_surcharge  Total_amount     Payment_type
##  Mode:logical   Min.   :-0.3000       Min.   :-52.80   Min.   :1.00
##  NA's:5000      1st Qu.: 0.3000       1st Qu.:  7.80   1st Qu.:1.00
##                 Median : 0.3000       Median : 11.16   Median :2.00
##                 Mean   : 0.2914       Mean   : 14.33   Mean   :1.52
##                 3rd Qu.: 0.3000       3rd Qu.: 17.16   3rd Qu.:2.00
##                 Max.   : 0.3000       Max.   :260.00   Max.   :4.00
##    Trip_type
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.023
##  3rd Qu.:1.000
##  Max.   :2.000
```

```
names(df)[names(df) == "VendorID"] <- "q.vendor_id"
names(df)[names(df) == "lpep_pickup_datetime"] <- "qual.lpep_pickup_datetime"
names(df)[names(df) == "Lpep_dropoff_datetime"] <- "qual.lpep_dropoff_datetime"
names(df)[names(df) == "Store_and_fwd_flag"] <- "qual.store_and_fwd_flag"
names(df)[names(df) == "RateCodeID"] <- "q.rate_code_id"
names(df)[names(df) == "Pickup_longitude"] <- "q.pickup_longitude"
names(df)[names(df) == "Pickup_latitude"] <- "q.pickup_latitude"
names(df)[names(df) == "Dropoff_longitude"] <- "q.dropoff_longitude"
names(df)[names(df) == "Dropoff_latitude"] <- "q.dropoff_latitude"
names(df)[names(df) == "Passenger_count"] <- "q.passenger_count"
names(df)[names(df) == "Trip_distance"] <- "q.trip_distance"
names(df)[names(df) == "Fare_amount"] <- "q.fare_amount"
```

```r
names(df)[names(df) == "Extra"] <- "q.extra"
names(df)[names(df) == "MTA_tax"] <- "q.mta_tax"
names(df)[names(df) == "Tip_amount"] <- "q.tip_amount"
names(df)[names(df) == "Tolls_amount"] <- "q.tolls_amount"
df$Ehail_fee <- NULL # deleting it --> only NA's
names(df)[names(df) == "improvement_surcharge"] <- "q.improvement_surcharge"
names(df)[names(df) == "Total_amount"] <- "q.target.total_amount"
names(df)[names(df) == "Payment_type"] <- "q.payment_type"
names(df)[names(df) == "Trip_type"] <- "q.trip_type"
summary(df); names(df)
```

```
##   q.vendor_id    qual.lpep_pickup_datetime qual.lpep_dropoff_datetime
##  Min.   :1.000   Length:5000               Length:5000
##  1st Qu.:2.000   Class :character          Class :character
##  Median :2.000   Mode  :character          Mode  :character
##  Mean   :1.788
##  3rd Qu.:2.000
##  Max.   :2.000
##  qual.store_and_fwd_flag q.rate_code_id q.pickup_longitude q.pickup_latitude
##  Length:5000             Min.   :1.0    Min.   :-75.39     Min.   : 0.00
##  Class :character        1st Qu.:1.0    1st Qu.:-73.96     1st Qu.:40.70
##  Mode  :character        Median :1.0    Median :-73.95     Median :40.75
##                          Mean   :1.1    Mean   :-73.89     Mean   :40.72
##                          3rd Qu.:1.0    3rd Qu.:-73.92     3rd Qu.:40.80
##                          Max.   :5.0    Max.   :  0.00     Max.   :41.04
##  q.dropoff_longitude q.dropoff_latitude q.passenger_count q.trip_distance
##  Min.   :-75.31      Min.   : 0.00      Min.   :0.000     Min.   : 0.000
##  1st Qu.:-73.97      1st Qu.:40.70      1st Qu.:1.000     1st Qu.: 1.020
##  Median :-73.94      Median :40.75      Median :1.000     Median : 1.800
##  Mean   :-73.80      Mean   :40.67      Mean   :1.375     Mean   : 2.765
##  3rd Qu.:-73.91      3rd Qu.:40.79      3rd Qu.:1.000     3rd Qu.: 3.420
##  Max.   :  0.00      Max.   :41.18      Max.   :6.000     Max.   :52.790
##  q.fare_amount      q.extra          q.mta_tax        q.tip_amount
##  Min.   :-52.0   Min.   :-1.0000   Min.   :-0.5000   Min.   : 0.000
##  1st Qu.:  6.0   1st Qu.: 0.0000   1st Qu.: 0.5000   1st Qu.: 0.000
##  Median :  9.0   Median : 0.5000   Median : 0.5000   Median : 0.000
##  Mean   : 11.9   Mean   : 0.3517   Mean   : 0.4857   Mean   : 1.217
##  3rd Qu.: 14.5   3rd Qu.: 0.5000   3rd Qu.: 0.5000   3rd Qu.: 2.000
##  Max.   :200.0   Max.   : 1.0000   Max.   : 0.5000   Max.   :96.000
##  q.tolls_amount     q.improvement_surcharge q.target.total_amount
##  Min.   : 0.00000   Min.   :-0.3000         Min.   :-52.80
##  1st Qu.: 0.00000   1st Qu.: 0.3000         1st Qu.:  7.80
##  Median : 0.00000   Median : 0.3000         Median : 11.16
##  Mean   : 0.08369   Mean   : 0.2914         Mean   : 14.33
##  3rd Qu.: 0.00000   3rd Qu.: 0.3000         3rd Qu.: 17.16
##  Max.   :18.04000   Max.   : 0.3000         Max.   :260.00
##  q.payment_type  q.trip_type
##  Min.   :1.00   Min.   :1.000
##  1st Qu.:1.00   1st Qu.:1.000
##  Median :2.00   Median :1.000
##  Mean   :1.52   Mean   :1.023
##  3rd Qu.:2.00   3rd Qu.:1.000
##  Max.   :4.00   Max.   :2.000
```

```
##  [1] "q.vendor_id"                "qual.lpep_pickup_datetime"
##  [3] "qual.lpep_dropoff_datetime" "qual.store_and_fwd_flag"
##  [5] "q.rate_code_id"             "q.pickup_longitude"
##  [7] "q.pickup_latitude"          "q.dropoff_longitude"
##  [9] "q.dropoff_latitude"         "q.passenger_count"
## [11] "q.trip_distance"            "q.fare_amount"
## [13] "q.extra"                    "q.mta_tax"
## [15] "q.tip_amount"               "q.tolls_amount"
## [17] "q.improvement_surcharge"    "q.target.total_amount"
## [19] "q.payment_type"             "q.trip_type"
```

6 DELIVERABLE I

6.1 Initiating missings, outliers and errors

Initialization of counts for missings, outliers and errors. All numerical variables have to be checked before

```
imis<-rep(0,nrow(df)); mis1<-countNA(df); imis<-mis1$mis_ind
jmis<-rep(0,2*ncol(df))

iouts<-rep(0,nrow(df))
jouts<-rep(0,2*ncol(df))

ierrs<-rep(0,nrow(df))
jerrs<-rep(0,2*ncol(df))
```

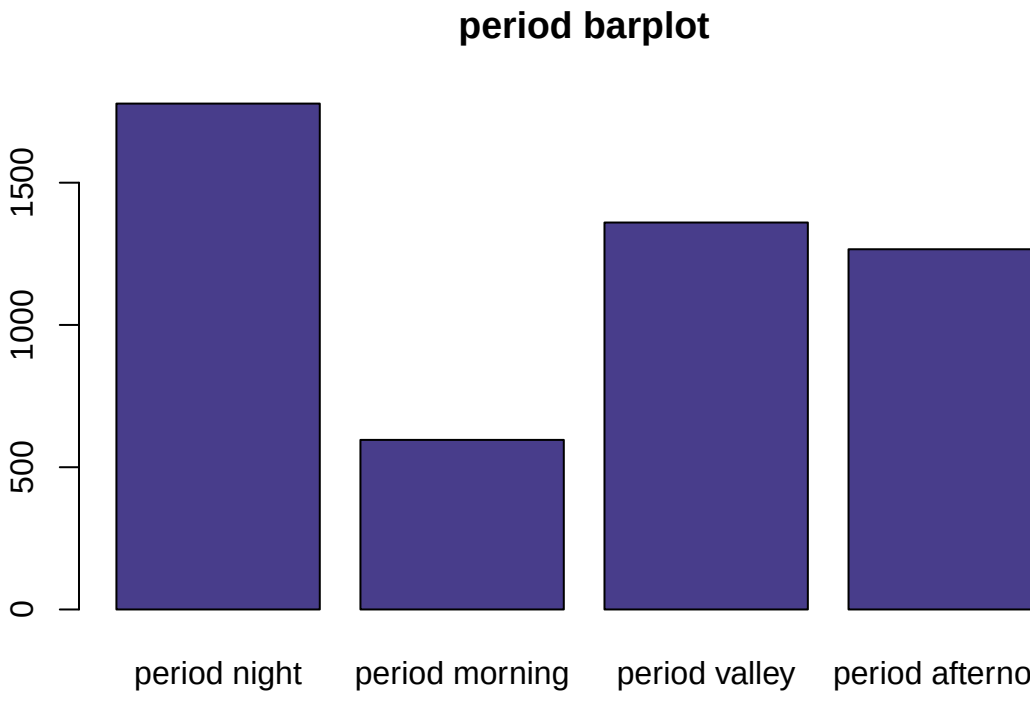
6.2 Univariate Descriptive Analysis

6.2.1 Qualitative Variables (Factors) / Categorical

Description: Original numeric variables corresponding to qualitative concepts have to be converted to factors. New factors grouping original levels will be considered very positively.

We need to do an analysis of all the variables to be able to identify missings, errors and outliers. We will also try to factorize each variable to make it easier to understand the sample.

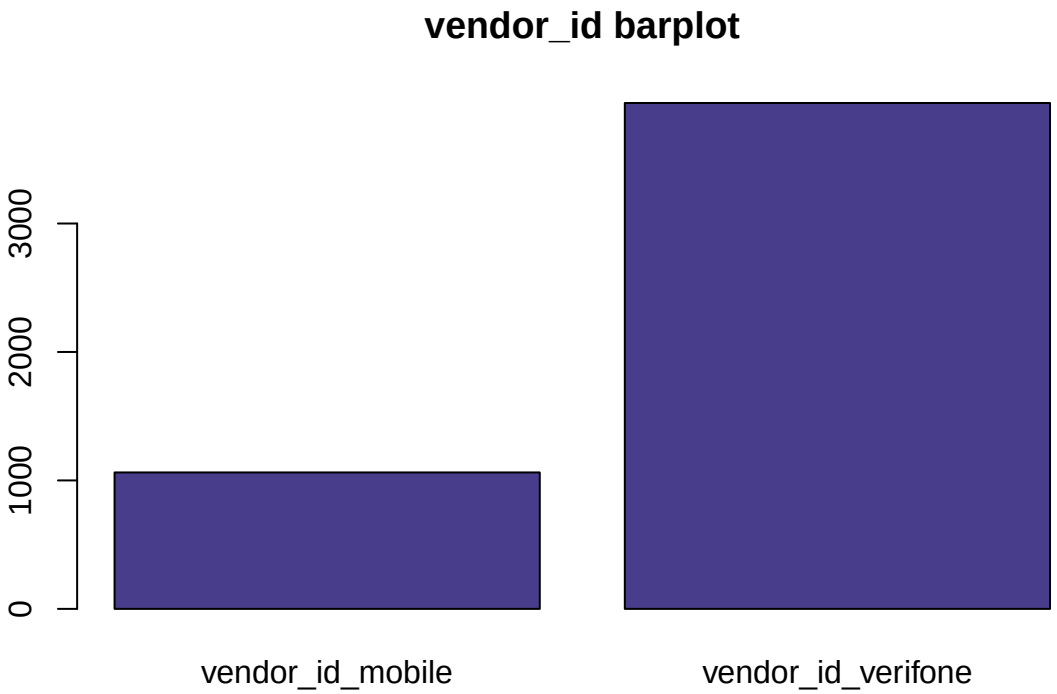
```
df$q.hour<-as.numeric(substr(strptime(df$qual.lpep_pickup_datetime, "%Y-%m-%d %H:%M:%S"),12,13))
df$f.period<-1
df$f.period[df$q.hour>7]<-2
df$f.period[df$q.hour>10]<-3
df$f.period[df$q.hour>16]<-4
df$f.period[df$q.hour>20]<-1
df$f.period<-factor(df$f.period,labels=paste("period",c("night","morning","valley","afternoon")))
barplot(summary(df$f.period),main="period barplot",col="darkslateblue")
```



6.2.1.1 New variable: Period

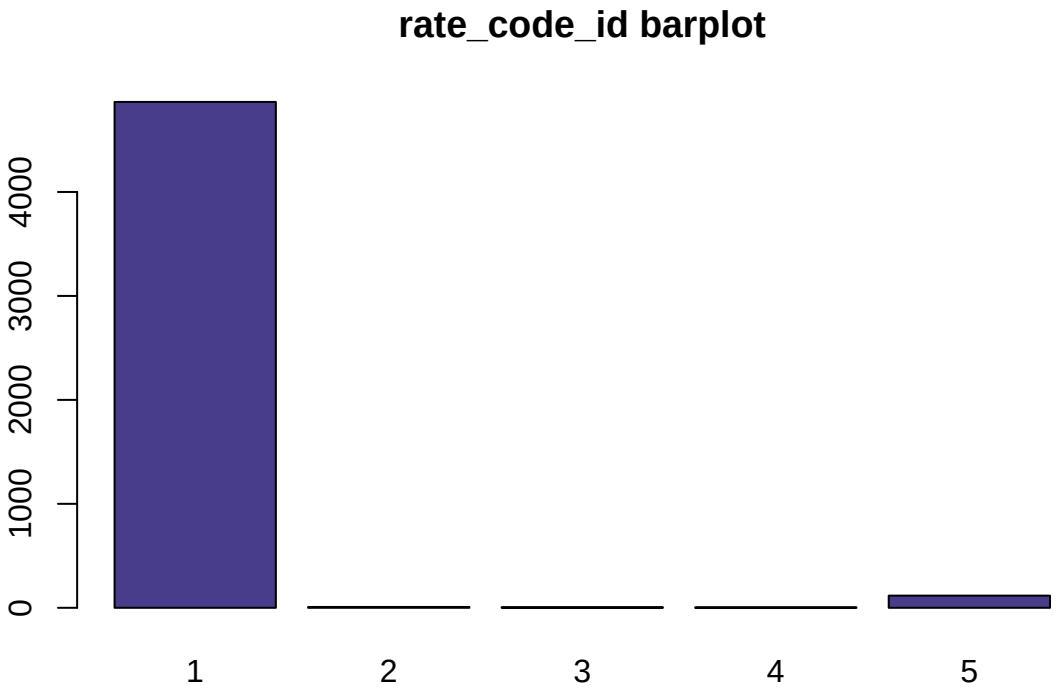
6.2.1.2 VendorID This variable expresses the Creative Mobile Technologies, LLC as 1 and Verifone Inc as 2, so we create a factor to make it more readable. With the initial summary we see that this variable does not have any missing value, so we proceed to factor it.

```
names(df)[names(df) == "q.vendor_id"] <- "f.vendor_id"
df$f.vendor_id<-factor(df$f.vendor_id,labels=c("vendor_id_mobile","vendor_id_verifone"))
barplot(summary(df$f.vendor_id),main="vendor_id barplot",col="darkslateblue")
```



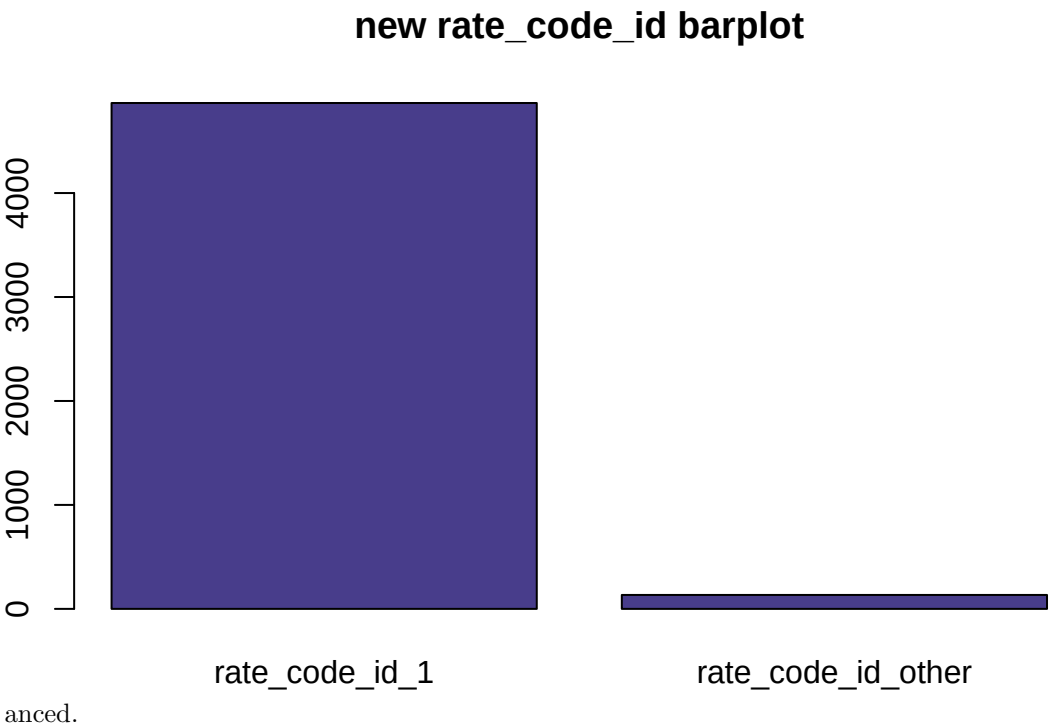
6.2.1.3 RateCodeID This variable expresses the different RateCodeIDs that we can have as numerical values, so we need to categorize them in order to be able to work with them.

```
names(df)[names(df) == "q.rate_code_id"] <- "f.rate_code_id"
df$f.rate_code_id<-factor(df$f.rate_code_id)
barplot(summary(df$f.rate_code_id),main="rate_code_id barplot",col="darkslateblue")
```



We see that most samples are in RateCodeID = 1, which is what we are interested in. Therefore, we factorize and create only two groups, the one with RateCodeID = 1 and the rest.

```
df$f.rate_code_id[df$f.rate_code_id != 1] = 2
df$f.rate_code_id <- factor(df$f.rate_code_id, labels=c("rate_code_id_1","rate_code_id_other"))
barplot(summary(df$f.rate_code_id),main="new rate_code_id barplot",col="darkslateblue")
```



Now is more balanced.

6.2.1.4 Store_and_fwd_flag

This is a categorical variable with the values Y and N, so we need to factor it.

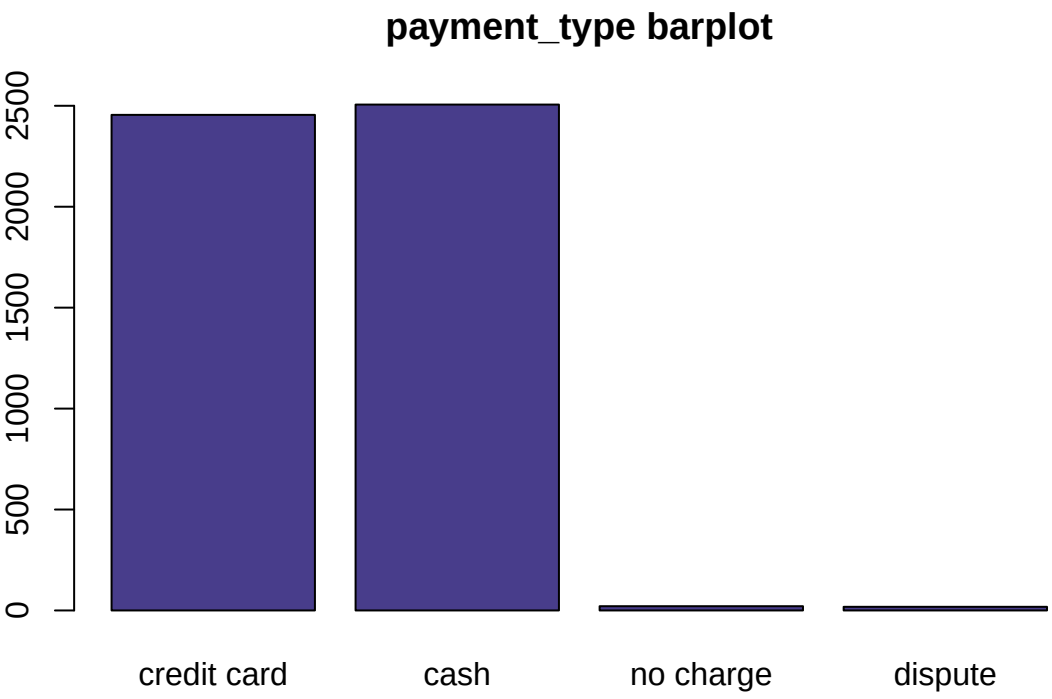
```
names(df)[names(df) == "qual.store_and_fwd_flag"] <- "f.store_and_fwd_flag"
df$f.store_and_fwd_flag<-factor(df$f.store_and_fwd_flag, labels=c("flag-no","flag-yes"))
summary(df$f.store_and_fwd_flag)
```

```
##  flag-no flag-yes
##     4982       18
```

6.2.1.5 Payment_type

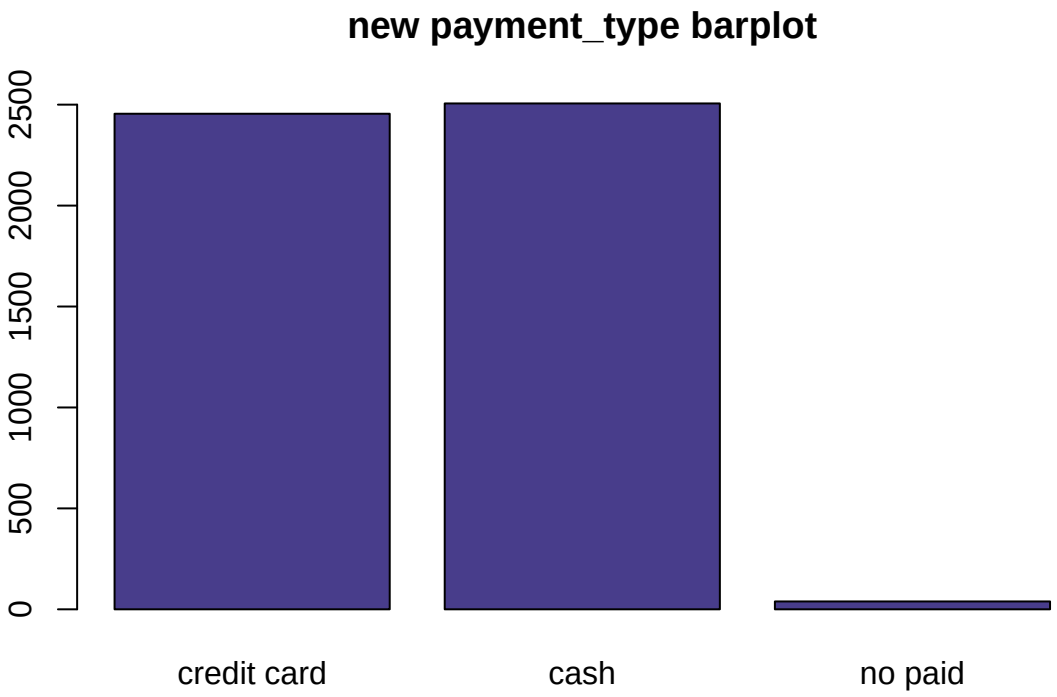
This variable is categorical but it is expressed as numerical, so we need to factor it in order to be able to work with it.

```
names(df)[names(df) == "q.payment_type"] <- "f.payment_type"
df$f.payment_type<-factor(df$f.payment_type,labels=c("credit card","cash","no charge","dispute"))
barplot(summary(df$f.payment_type),main="payment_type barplot",col="darkslateblue")
```



As we can see, there are few values with "No charge" or "Dispute" category, so we decided to categorize it into a new category ("No paid").

```
levels(df$f.payment_type) <- c("credit card","cash","no paid","no paid")
barplot(summary(df$f.payment_type),main="new payment_type barplot",col="darkslateblue")
```



Now is more balanced.

6.2.1.6 Trip_type This variable is categorical but it is expressed as numerical, so we need to factor it in order to be able to work with it.

```
names(df)[names(df) == "q.trip_type"] <- "f.trip_type"
df$f.trip_type<-factor(df$f.trip_type,labels=c("trip_street_hail","trip_dispatch"))
barplot(summary(df$f.trip_type),main="trip_type barplot",col="darkslateblue")
```

6.2.2 Quantitative Variables

Description: Original numeric variables corresponding to real quantitative concepts are kept as numeric but additional factors should also be created as a discretization of each numeric variable.

We only keep the hours (variables 2 and 3) to be able to work with time slots in the future.

Create new variables derived from the original ones, as effective speed, travel time, hour of request, period of request, effective trip distance (in km)

6.2.2.1 New variables: Trip Length in km, Travel time un min and Effective speed

```
df$q.tlenkm<-df$q.trip_distance*1.609344 # Miles to km
```

6.2.2.1.1 Trip length in km

```
df$q.travel_time<-(as.numeric(as.POSIXct(df$qual.lpep_dropoff_datetime)) - as.numeric(as.POSIXct(df$qual
```

6.2.2.1.2 Travel time in min

```
df$q.espeed<-(df$q.tlenkm/(df$q.travel_time))*60
```

6.2.2.1.3 Effective speed in km/h Missing data

```
sel<-which(is.na(df$q.espeed<=0))
imis[sel]<-imis[sel]+1
jmis[25]<-length(sel)
```

Error detection

We detect as error those speeds smaller than 0 and bigger than 200

```
summary(df$q.espeed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   14.60   18.58   23.07   23.70 3881.74       2
```

```
sel<-which((df$q.espeed<=0)|(df$q.espeed > 200))
ierrs[sel]<-ierrs[sel]+1
jerrs[25]<-length(sel)
```

Sel contains the rownames of the individuals with "0" as value for longitude

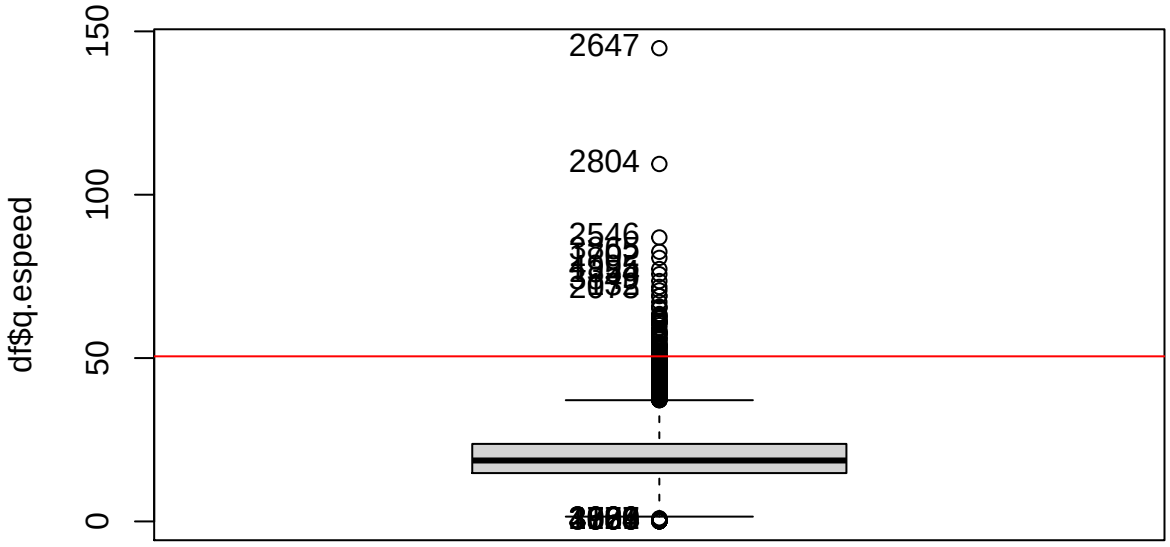
```
df[sel,"q.espeed"]<-NA
```

Outlier detection

```
Boxplot(df$q.espeed)
```

```
##  [1] 4780 3001 3066 1936  120 3578 1767 4824 2685 3009 2647 2804 2546 3865 1702
## [16] 4995 1354 3849  132 2075
```

```
var_out<-calcQ(df$q.espeed)
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
```




```
llout<-which((df$q.espeed<=3)|(df$q.espeed>80))
iouts[llout]<-iouts[llout]+1
```

```
jouts[25]<-length(llout)
df[llout,"q.espeed"]<-NA
```

6.2.2.2 lpep_pickup_datetime

We just keep the hours

```
df$qual.pickup<-substr(strptime(df$qual.lpep_pickup_datetime, "%Y-%m-%d %H:%M:%S"), 12, 13)
```

6.2.2.3 lpep_dropoff_datetime

We just keep the hours

```
df$qual.dropoff<-substr(strptime(df$qual.lpep_dropoff_datetime, "%Y-%m-%d %H:%M:%S"), 12, 13)
```

```
summary(df$q.passenger_count)
```

6.2.2.4 Passenger_count

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   1.000   1.375   1.000   6.000
```

We set the 0 as an error because it is not possible to have a trip without passengers

```
sel<-which(df$q.passenger_count == 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[10]<-length(sel)
```

Sel contains the rownames of the individuals with "0" as value for passengers

```
df[sel,"q.passenger_count"]<-NA
```

```
summary(df$q.trip_distance)
```

6.2.2.5 Trip_distance

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.020   1.800   2.765   3.420  52.790
```

We see on the summary that there are not NA values, so we proceed to the outlier and error detection.

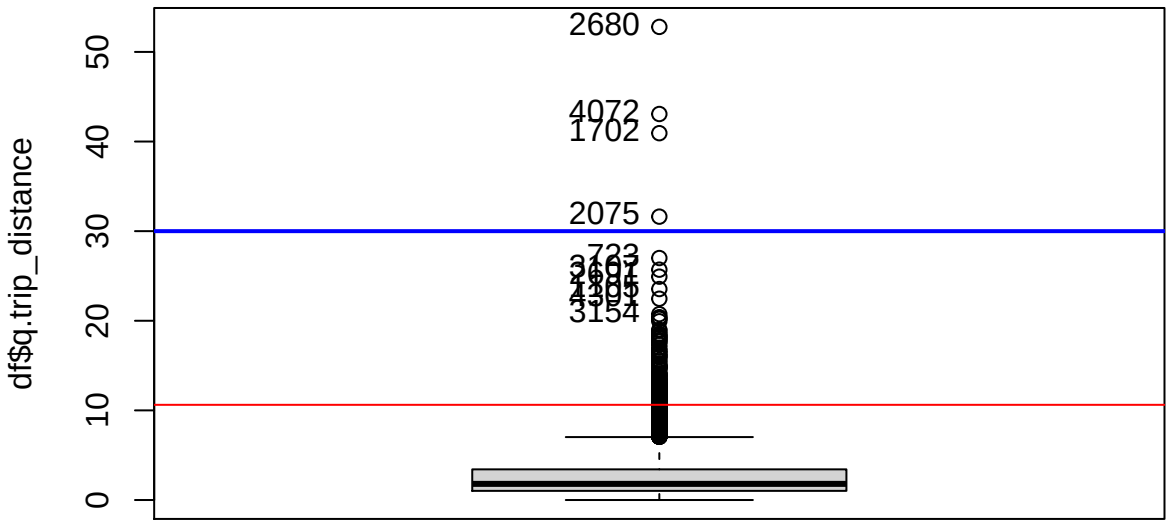
6.2.2.5.1 Outlier detection

In order to evalute or data, we decide to set the maximum trip distance to 30, so we proceed to delete the outliers.

```
Boxplot(df$q.trip_distance)
```

```
##  [1] 2680 4072 1702 2075  723 3107 2691 1105 4301 3154
```

```
var_out<-calcQ(df$q.trip_distance)
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
abline(h=30,col="blue",lwd=2)
```



```
llout<-which(df$q.trip_distance>30)
iouts[llout]<-iouts[llout]+1
jouts[11]<-length(llout)
```

6.2.2.5.2 Error detection We decide that an incorrect trip distance is the one with 0 miles or less. In order to be aware of this error we store it at ierrs, and jerrs. ierrs stores the number of errors in a row, and jerrs stores the total amount of errors in a variable.

```
sel<-which(df$q.trip_distance <= 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[11]<-length(sel)
```

6.2.2.5.3 Errors and outliers Now, we set NA values in order to remove errors and outliersfrom the dataset

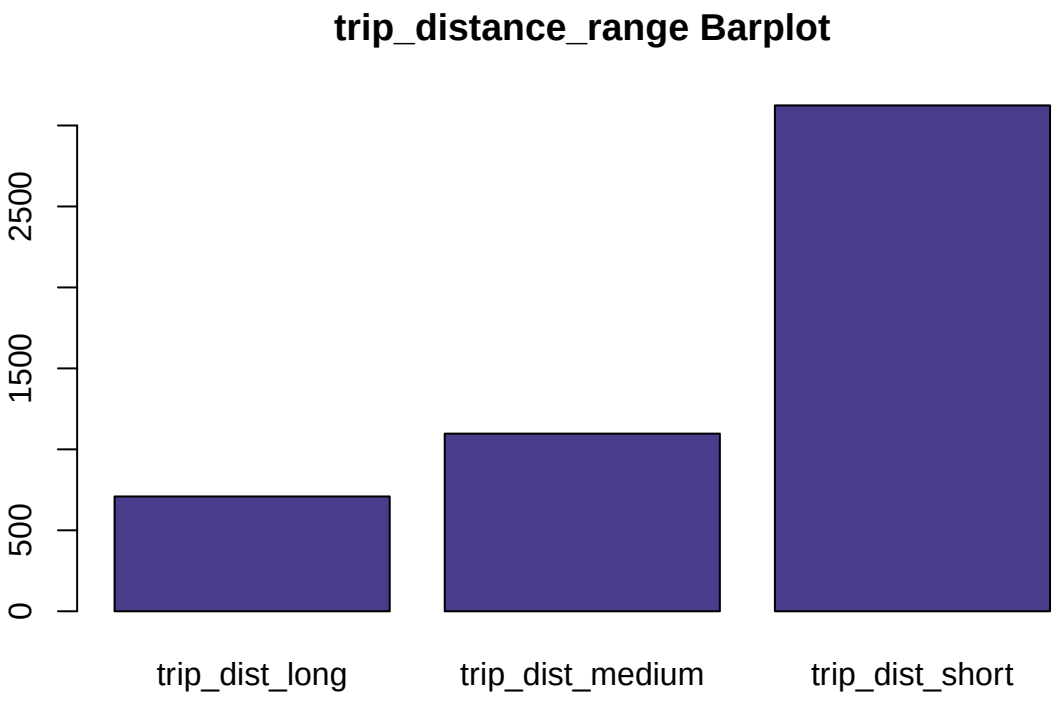
```
setNA<-which((df$q.trip_distance<=0) | (df$q.trip_distance > 30))
df[setNA,"q.trip_distance"]<-NA
```

6.2.2.5.4 Caterogial variable for Trip_distance We are going to set a categorical variable for the f.trip_distance_range We decided to create 3 levels: "trip_dist_short", "trip_dist_medium" and"trip_dist_long". - trip_dist_short: <= 2.5 - trip_dist_medium: 2.5 < q.trip_distance <= 5 - trip_dist_long: > 5

```
df$f.trip_distance_range[df$q.trip_distance <= 2.5] = "trip_dist_short"
df$f.trip_distance_range[(df$q.trip_distance > 2.5) & (df$q.trip_distance <= 5)] = "trip_dist_medium"
df$f.trip_distance_range[df$q.trip_distance > 5] = "trip_dist_long"
df$f.trip_distance_range <- factor(df$f.trip_distance_range)
```

We see a barplot for the factor we created.

```
barplot(table(df$f.trip_distance_range),main="trip_distance_range Barplot",col="darkslateblue")
```



6.2.2.6 Pickup_longitude We know that New York's longitude is -73.9385, so values that differ a lot from this value is an error or an outlier.

```
summary(df$q.pickup_longitude)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -75.39  -73.96  -73.95  -73.89  -73.92    0.00
```

0.00 looks to be an error Seeing the individuals with this "0" value: df[which(df[,"q.pickup_longitude"]==0),] it is a quantitive variable. Non-possible values will be recoded as errors, so will be transformed to NA.

```
sel<-which(df$q.pickup_longitude == 0)
ierrs[sel]<-ierrs[sel]+1
```

```
jerrs[6]<-length(sel)
df[sel,"q.pickup_longitude"]<-NA
```

Non-possible values are replaced by NA, missing value symbol in R.

We are deleting trips from outside New York. This means we are not using longitudes bigger than -73.80 and smaller than -74.02.

```
llout <-which((df$q.pickup_longitude < -74.02) | (df$q.pickup_longitude > -73.80))
iouts[llout]<-iouts[llout]+1
jouts[6]<-length(llout)
df[llout,"q.pickup_longitude"]<-NA
```

6.2.2.7 Pickup_latitude

We know that New York's latitude is 40.6643, so values that differ a lot from this value is an error or an outlier.

```
summary(df$q.pickup_latitude)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   40.70   40.75   40.72   40.80   41.04
```

0.00 looks to be an error. Seeing the individuals with this "0" value: df[which(df[,"q.pickup_latitude"]==0),] it is a quantitive variable. non-possible values will be recoded as errors, so will be transformed to NA.

```
sel<-which(df$q.pickup_latitude == 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[7]<-length(sel)
df[sel,"q.pickup_latitude"]<-NA
```

Non-possible values are replaced by NA, missing value symbol in R. We are deleting trips from outside New York. This means we are not using latitudes bigger than 40.54 and smallerthan 40.86

```
llout <-which((df$q.pickup_latitude < 40.54) | (df$q.pickup_latitude > 40.86))
iouts[llout]<-iouts[llout]+1
jouts[7]<-length(llout)
df[llout,"q.pickup_latitude"]<-NA
```

6.2.2.8 Dropoff_longitude

We know that New York's longitude is -73.9385, so values that differ a lot from this value is an error or an outlier.

```
summary(df$q.dropoff_longitude)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -75.31  -73.97  -73.94  -73.80  -73.91    0.00
```

0.00 looks to be an error Seeing the individuals with this "0" value: df[which(df[,"q.dropoff_longitude"]==0),] it is a quantitive variable.
Non-possible values will be recoded as errors, so will be transformed to NA.

```
sel<-which(df$q.dropoff_longitude == 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[8]<-length(sel)
df[sel,"q.dropoff_longitude"]<-NA
```

Non-possible values are replaced by NA, missing value symbol in R. We are deleting trips from outside New York. This means we are not using longitudes bigger than -73.80 and smaller than -74.02.

```
llout <-which((df$q.dropoff_longitude < -74.02) | (df$q.dropoff_longitude > -73.80))
iouts[llout]<-iouts[llout]+1
jouts[8]<-length(llout)
df[llout,"q.dropoff_longitude"]<-NA
```

6.2.2.9 Dropoff_latitude

We know that New York's latitude is 40.6643, so values that differ a lot from this value is an error or an outlier.

```
summary(df$q.dropoff_latitude)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   40.70   40.75   40.67   40.79   41.18
```

0.00 looks to be an error Seeing the individuals with this "0" value: df[which(df[,"q.dropoff_latitude"]==0),] it is a quantitive variable. Non-possible values will be recoded as errors, so will be transformed to NA.

```
sel<-which(df$q.dropoff_latitude == 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[9]<-length(sel)
```

Sel contains the rownames of the individuals with "0" as value for longitude

```
df[sel,"q.dropoff_latitude"]<-NA
```

Non-possible values are replaced by NA, missing value symbol in R. We are deleting trips from outside New York. This means we are not using latitude bigger than 40.54 and smaller than 40.86

```
llout <-which((df$q.dropoff_latitude < 40.54) | (df$q.dropoff_latitude > 40.86))
iouts[llout]<-iouts[llout]+1
jouts[9]<-length(llout)
```

Now that we have the outliers, we are setting them as NA

```
df[llout,"q.dropoff_latitude"]<-NA
```

6.2.2.10 13. Fare_amount

We know that the fare should be positive, as it is the price of the trip, so we'll treat as error those values. The next we'll do is decide the outliers.

```
summary(df$q.fare_amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -52.0     6.0     9.0    11.9    14.5   200.0
```

```
sel<-which(df$q.fare_amount <= 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[12]<-length(sel)
df[sel,"q.fare_amount"]<-NA
```

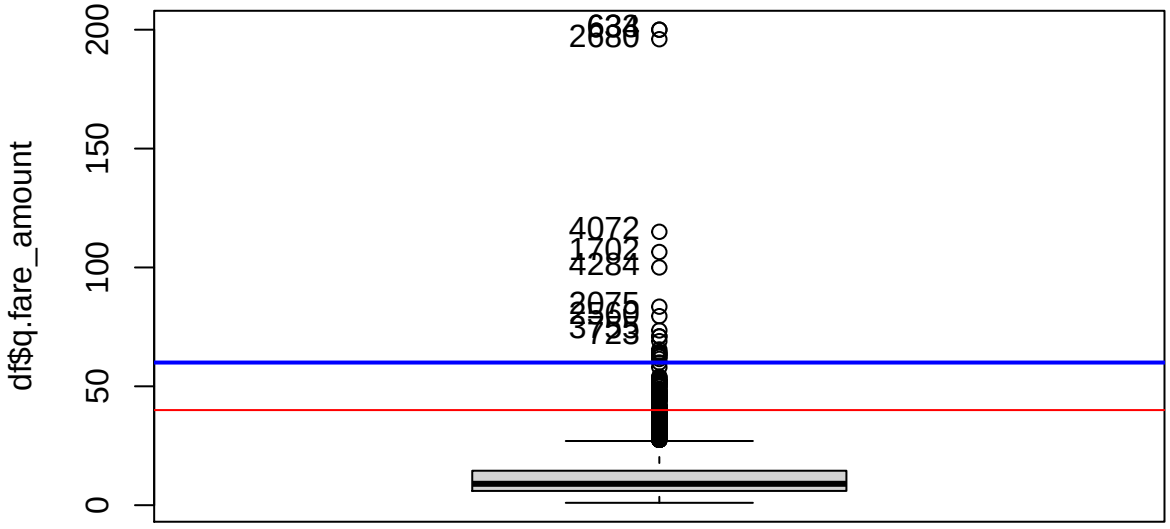
Non-possible values are replaced by NA, missing value symbol in R

```
Boxplot(df$q.fare_amount)
```

6.2.2.10.1 Outlier detection

```
##  [1]  633  634 2680 4072 1702 4284 2075 2560 3755  723
```

```
var_out<-calcQ(df$q.fare_amount)
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
abline(h=60,col="blue",lwd=2)
```



We decide to set outliers for fare amounts bigger than 60, because the majority of the values are concentrated between 0 and 60.

```
llout<-which(df$q.fare_amount>60)
iouts[llout]<-iouts[llout]+1
jouts[12]<-length(llout)
df[llout,"q.fare_amount"]<-NA
```

6.2.2.11 Extra As this variable is price related, it cannot have negative values, so this individuals will be treated as errors.

```
table(df$q.extra)
```

```
##
##   -1 -0.5    0  0.5    1
##    2    5 2296 1868  829
```

As it is a price related variable, negative values should be treated as errors, and the other values are the ones defined for this variable, so there are not outliers.

```
sel<-which(df$q.extra < 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[13]<-length(sel)
df[sel,"q.extra"]<-NA
```

6.2.2.12 MTA_tax This variable corresponds to a tax that must be charged in every trip and its cost is $0.50, so values different from this are errors, and we don't have to take into account outliers because after the errors detection all values should be the MTA_tax.

```
table(df$q.mta_tax)
```

```
##
## -0.5    0  0.5
##   10  123 4867
```

Important note: We assume that when this tax is smaller than 0, it is an error. If tax is 0, we say that payment in these cases is equivalent to "no paid".

```
sel<-which(df$q.mta_tax < 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[14]<-length(sel)
df[sel,"q.mta_tax"]<-NA
```

6.2.2.13 Improvement_surcharge This variable corresponds to a charge that must be charged in every trip and its cost is $0.30, so values smaller than 0 are errors, and we don't have to take into account outliers because after the errors detection all values should be the Improvement surcharge.

```
table(df$q.improvement_surcharge)
```

```
##
## -0.3    0  0.3
##   11  121 4868
```

We see that the 0 individuals are errors.

```
sel<-which(df$q.improvement_surcharge < 0)
ierrs[sel]<-ierrs[sel]+1
jerrs[17]<-length(sel)
df[sel,"q.improvement_surcharge"]<-NA
```

6.2.2.14 Tip_amount As this is a price related variable, negative values should be considered as errors, and big tips should be considered as outliers. Also tip amounts bigger than 0 for individuals with payment_type = "Cash" should be considered as errors as well.

```
summary(df$q.tip_amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   1.217   2.000  96.000
```

We proceed to check if the 0 values are related with payment_type = "credit card" and the passenger did not tip.

```
table(df$q.tip_amount>0, df$f.payment_type)
```

```
##
##         credit card cash no paid
##   FALSE         357 2506      39
##   TRUE         2098    0       0
```

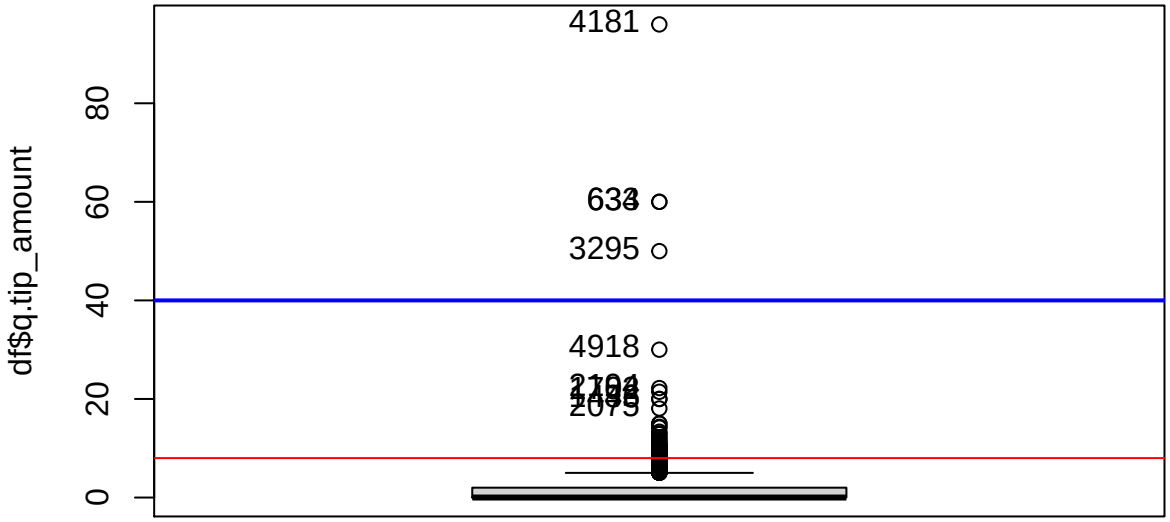
Now, we proceed to the outlier detection.

```
Boxplot(df$q.tip_amount)
```

6.2.2.15 Outlier detection

```
##  [1] 4181  633  634 3295 4918 2194 1702   46 1433 2075
```

```
var_out<-calcQ(df$q.tip_amount)
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
abline(h=40,col="blue",lwd=2)
```




```
llout<-which(df$q.tip_amount>40)
iouts[llout]<-iouts[llout]+1
jouts[15]<-length(llout)
df[llout,"q.tip_amount"]<-NA
```

6.2.2.16 Tolls_amount

As this is a price related variable, negative values should be considered as errors.

```
table(df$q.tolls_amount)
```

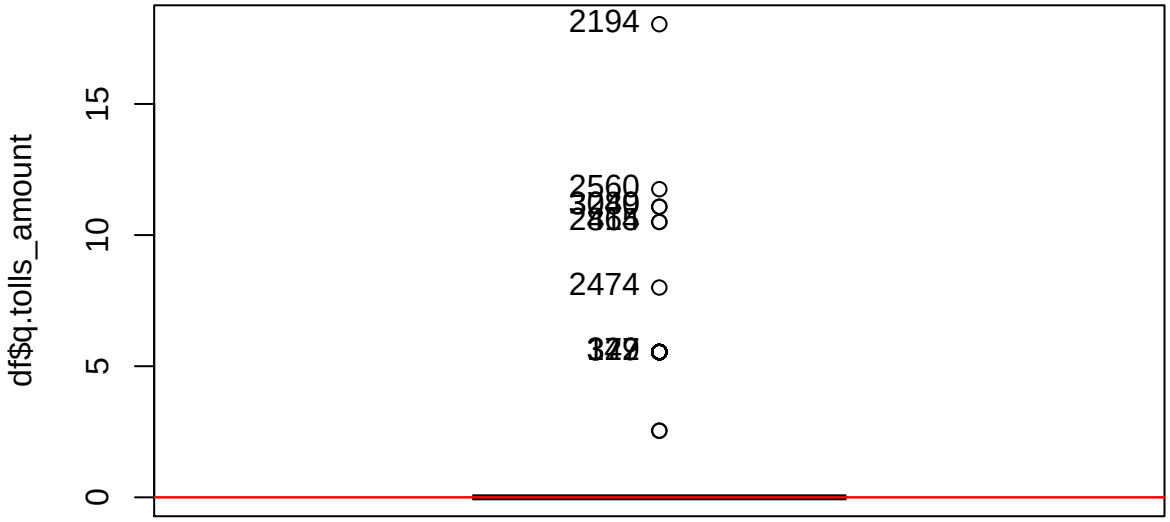
```
##
##     0  2.54  5.54     8  10.5 11.08 11.75 18.04
##  4931     2    60     1     2     2     1     1
```

We see that there are not negative values, so we do not have errors. We proceed now to the outlier detection.

```
Boxplot(df$q.tolls_amount)
```

```
##  [1] 2194 2560 3040 3289  415 2864 2474  122  347  379
```

```
var_out<-calcQ(df$q.tolls_amount)
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
```



As we see in the boxplot and the table, the majority of the individuals are 0, so the values bigger than 5.54 will be outliers.

```
llout<-which(df$q.tolls_amount>5.54)
iouts[llout]<-iouts[llout]+1
jouts[16]<-length(llout)
df[llout,"q.tolls_amount"]<-NA
```

6.2.2.17 20. Total_amount

This is a price related variable, so negative values should be treated as errors. Also, we need to sum the "Fare_amount", "Extra","MTA_tax", "Improvement_surcharge", "Tip_amount" and the "Tolls_amount" in order to see if the Total_amount matches with this sum.

```
summary(df$q.target.total_amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -52.80    7.80   11.16   14.33   17.16  260.00
```

Negative values seem to be errors - 0 Total_amount is possible when Payment_type ==“No charge”

We proceed to check if total amount is correctsumming the other variables and checking negatives values:

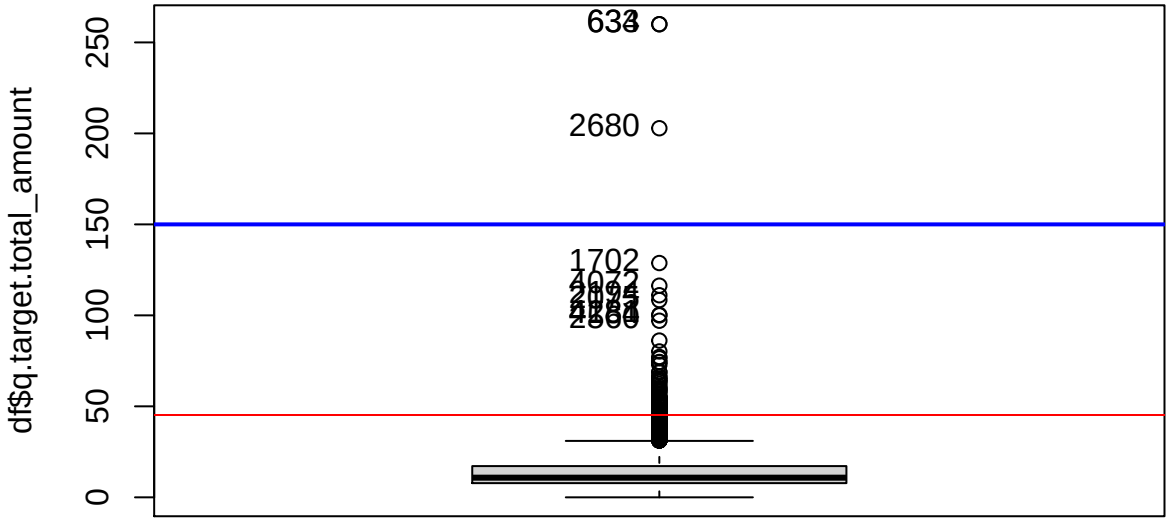
```
sum_total_amount = (
  df$q.fare_amount +
  df$q.extra +
  df$q.mta_tax +
  df$q.improvement_surcharge +
  df$q.tip_amount +
  df$q.tolls_amount
)

sel<-which((df$q.target.total_amount != sum_total_amount) | (df$q.target.total_amount<0))
if (length(sel)>0) {
  ierrs[sel]<-ierrs[sel]+1
  jerrs[18]<-length(sel)
}
df[sel,"q.target.total_amount"]<-NA
```

```
Boxplot(df$q.target.total_amount)
```

6.2.2.18 Outlier detection

```
##  [1]  633  634 2680 1702 4072 2194 2075 4181 4284 2560
```

```
var_out<-calcQ(df$q.target.total_amount)
abline(h=var_out$souts,col="red")
abline(h=var_out$souti,col="red")
abline(h=150,col="blue",lwd=2)
```




```
llout<-which(df$q.target.total_amount>150)
iouts[llout]<-iouts[llout]+1
jouts[18]<-length(llout)
df[llout,"q.target.total_amount"]<-NA
```

6.3 Data Quality Report

6.3.1 Per variable

Per each variable, we have to count the following:

- number of missing values
- number of errors (including inconsistencies)
- number of outliers
- rank variables according the sum of missing values (and errors).

```
missings_ranking_sortlist <- sort.list(mis1$mis_col, decreasing = TRUE)
for (i in missings_ranking_sortlist) {
  print(paste(names(df)[i], " : ", mis1$mis_col$mis_x[i]))
}
```

6.3.1.1 Number of missing values of each variable (with ranking)

```
## [1] "f.vendor_id  :  0"
## [1] "qual.lpep_pickup_datetime  :  0"
## [1] "qual.lpep_dropoff_datetime  :  0"
## [1] "f.store_and_fwd_flag  :  0"
## [1] "f.rate_code_id  :  0"
## [1] "q.pickup_longitude  :  0"
## [1] "q.pickup_latitude  :  0"
## [1] "q.dropoff_longitude  :  0"
## [1] "q.dropoff_latitude  :  0"
## [1] "q.passenger_count  :  0"
## [1] "q.trip_distance  :  0"
## [1] "q.fare_amount  :  0"
## [1] "q.extra  :  0"
## [1] "q.mta_tax  :  0"
## [1] "q.tip_amount  :  0"
## [1] "q.tolls_amount  :  0"
## [1] "q.improvement_surcharge  :  0"
## [1] "q.target.total_amount  :  0"
## [1] "f.payment_type  :  0"
## [1] "f.trip_type  :  0"
```

```
errors_ranking_sortlist <- sort.list(jerrs, decreasing = TRUE)
for (i in errors_ranking_sortlist) {
  if(!is.na(names(df)[i])) { print(paste(names(df)[i], " : ", jerrs[i])) }
}
```

6.3.1.2 Number of errors per each variable (with ranking)

```
## [1] "q.target.total_amount  :  374"
## [1] "q.espeed  :  73"
## [1] "q.trip_distance  :  66"
## [1] "q.fare_amount  :  24"
## [1] "q.improvement_surcharge  :  11"
## [1] "q.mta_tax  :  10"
## [1] "q.dropoff_longitude  :  9"
## [1] "q.dropoff_latitude  :  9"
## [1] "q.extra  :  7"
## [1] "q.pickup_longitude  :  3"
## [1] "q.pickup_latitude  :  3"
## [1] "q.passenger_count  :  2"
## [1] "f.vendor_id  :  0"
## [1] "qual.lpep_pickup_datetime  :  0"
## [1] "qual.lpep_dropoff_datetime  :  0"
## [1] "f.store_and_fwd_flag  :  0"
## [1] "f.rate_code_id  :  0"
## [1] "q.tip_amount  :  0"
## [1] "q.tolls_amount  :  0"
## [1] "f.payment_type  :  0"
## [1] "f.trip_type  :  0"
## [1] "q.hour  :  0"
## [1] "f.period  :  0"
## [1] "q.tlenkm  :  0"
## [1] "q.travel_time  :  0"
## [1] "qual.pickup  :  0"
## [1] "qual.dropoff  :  0"
## [1] "f.trip_distance_range  :  0"
```

```
errors_ranking_sortlist <- sort.list(jouts, decreasing = TRUE)
for (i in errors_ranking_sortlist) {
  if(!is.na(names(df)[i])) print(paste(names(df)[i], " : ", jouts[i]))
}
```

6.3.1.3 Number of outliers per each variable (with ranking)

```
## [1] "q.dropoff_latitude  :  116"
## [1] "q.dropoff_longitude  :  113"
## [1] "q.pickup_latitude  :  84"
## [1] "q.espeed  :  39"
## [1] "q.fare_amount  :  20"
## [1] "q.pickup_longitude  :  19"
## [1] "q.tolls_amount  :  7"
## [1] "q.trip_distance  :  4"
## [1] "q.tip_amount  :  4"
## [1] "q.target.total_amount  :  3"
## [1] "f.vendor_id  :  0"
## [1] "qual.lpep_pickup_datetime  :  0"
## [1] "qual.lpep_dropoff_datetime  :  0"
## [1] "f.store_and_fwd_flag  :  0"
## [1] "f.rate_code_id  :  0"
## [1] "q.passenger_count  :  0"
## [1] "q.extra  :  0"
## [1] "q.mta_tax  :  0"
## [1] "q.improvement_surcharge  :  0"
## [1] "f.payment_type  :  0"
## [1] "f.trip_type  :  0"
```

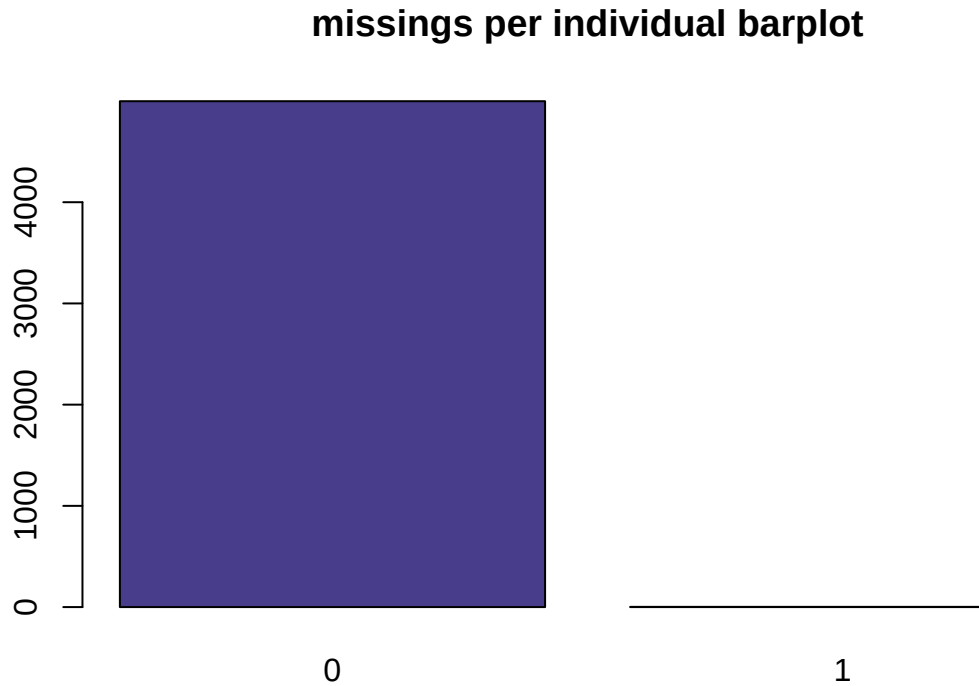
```
## [1] "q.hour  :  0"
## [1] "f.period  :  0"
## [1] "q.tlenkm  :  0"
## [1] "q.travel_time  :  0"
## [1] "qual.pickup  :  0"
## [1] "qual.dropoff  :  0"
## [1] "f.trip_distance_range  :  0"
```

6.3.2 Per individual

Per each individuals, we have to count the following:

- number of missing values
- number of errors
- number of outliers

```
barplot(table(imis),main="missings per individual barplot",col="darkslateblue")
```



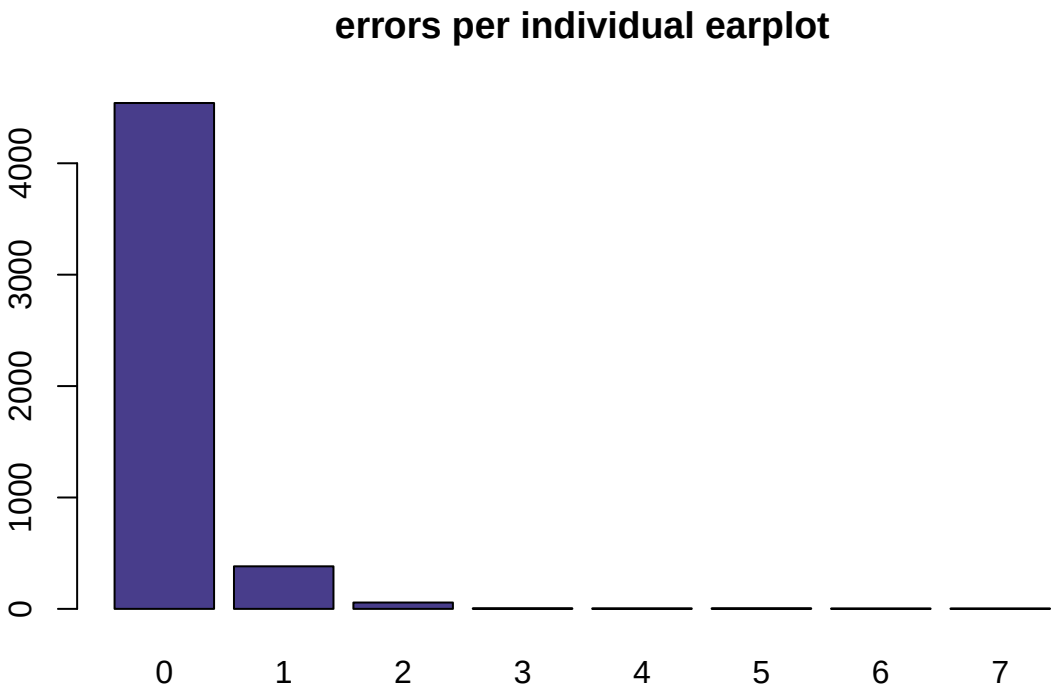
6.3.2.1 Number of missing values

We see that there are no native missing values (remember we deleted Ehail_fee).

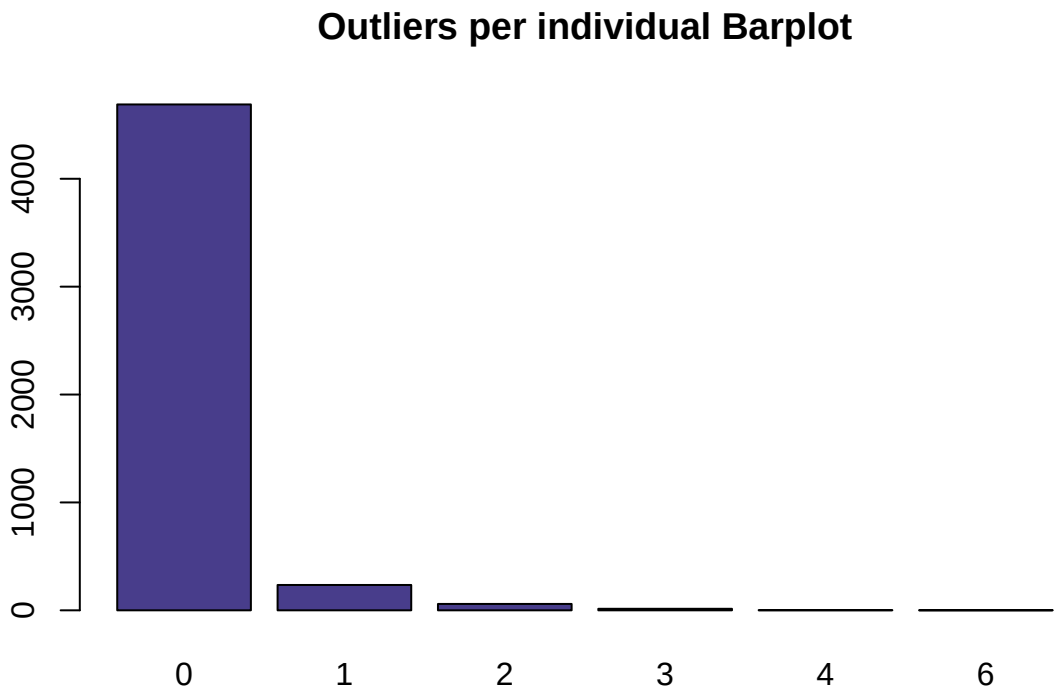
6.3.2.2 Number of errors

As we can see, most individuals have no mistakes.

```
barplot(table(ierrs),main="errors per individual earplot",col="darkslateblue")
```




```
barplot(table(iouts),main="Outliers per individual Barplot",col="darkslateblue")
```



6.3.2.3 Number of outliers

6.3.3 Create variable adding the total number missing values, outliers and errors

```
total_missings <- 0; total_outliers <- 0; total_errors <- 0;
for (m in imis) {total_missings <- total_missings + m}
for (o in iouts) {total_outliers <- total_outliers + o}
for (e in ierrs) {total_errors <- total_errors + e}
```

Now, let's print this variables:

```
total_missings
```

```
## [1] 2
```

```
total_outliers
```

```
## [1] 409
```

```
total_errors
```

```
## [1] 591
```

```
## Delete some unecessary variables
r df$qual.lpep_pickup_datetime <- NULL df$qual.lpep_dropoff_datetime <- NULL names(df)
##  [1] "f.vendor_id"             "f.store_and_fwd_flag" ##  [3] "f.rate_code_id"
"q.pickup_longitude" ##  [5] "q.pickup_latitude"       "q.dropoff_longitude" ##  [7]
"q.dropoff_latitude"      "q.passenger_count" ##  [9] "q.trip_distance"
"q.fare_amount" ## [11] "q.extra"                 "q.mta_tax" ## [13] "q.tip_amount"
"q.tolls_amount" ## [15] "q.improvement_surcharge" "q.target.total_amount" ## [17]
"f.payment_type"          "f.trip_type" ## [19] "q.hour"                  "f.period" ##
[21] "q.tlenkm"               "q.travel_time" ## [23] "q.espeed"
"qual.pickup" ## [25] "qual.dropoff"            "f.trip_distance_range"
```

6.4 Imputation

```
library(missMDA)
```

What we do with imputation is be able to eliminate all those values that may be missings, outliers or errors to turn them into values that can be realistic within our sample.

6.4.1 Numeric variables

We will now do the study by variables and try to impute the necessary observations.

Note: we do not include MTA_tax (14) nor improvement_surcharge(18). We proceed to delete NA values from Total_amount because it is our target variable, so we do not impute it, but we need to have this variable without NAs.

```
df <- df[!is.na(df$q.target.total_amount),]
names(df)
```

```
##  [1] "f.vendor_id"             "f.store_and_fwd_flag"
##  [3] "f.rate_code_id"          "q.pickup_longitude"
##  [5] "q.pickup_latitude"       "q.dropoff_longitude"
##  [7] "q.dropoff_latitude"      "q.passenger_count"
##  [9] "q.trip_distance"         "q.fare_amount"
## [11] "q.extra"                 "q.mta_tax"
## [13] "q.tip_amount"            "q.tolls_amount"
## [15] "q.improvement_surcharge" "q.target.total_amount"
## [17] "f.payment_type"          "f.trip_type"
## [19] "q.hour"                  "f.period"
## [21] "q.tlenkm"                "q.travel_time"
## [23] "q.espeed"                "qual.pickup"
## [25] "qual.dropoff"            "f.trip_distance_range"
```

```
vars_quantitatives <- names(df)[c(4,5,6,7,8,9,10,11,12,13,14,15,16,21,22,23)]
#  [1] "q.pickup_longitude"      "q.pickup_latitude"
#  [3] "q.dropoff_longitude"     "q.dropoff_latitude"
#  [5] "q.passenger_count"       "q.trip_distance"
#  [7] "q.fare_amount"           "q.extra"
#  [9] "q.mta_tax"               "q.tip_amount"
# [11] "q.tolls_amount"          "q.improvement_surcharge"
# [13] "q.tlenkm"                "q.travel_time"
# [15] "q.espeed"
```

```
summary(df[,vars_quantitatives])
```

```
##  q.pickup_longitude q.pickup_latitude q.dropoff_longitude q.dropoff_latitude
##  Min.   :-74.02     Min.   :40.58     Min.   :-74.02      Min.   :40.58
##  1st Qu.:-73.96     1st Qu.:40.70     1st Qu.:-73.97      1st Qu.:40.70
##  Median :-73.94     Median :40.75     Median :-73.94      Median :40.75
##  Mean   :-73.93     Mean   :40.75     Mean   :-73.94      Mean   :40.74
##  3rd Qu.:-73.92     3rd Qu.:40.80     3rd Qu.:-73.91      3rd Qu.:40.79
```

```
##  Max.   :-73.80     Max.   :40.86     Max.   :-73.80      Max.   :40.86
##  NA's   :20         NA's   :81        NA's   :110         NA's   :119
##  q.passenger_count q.trip_distance  q.fare_amount      q.extra
##  Min.   :1.000     Min.   : 0.010   Min.   : 1.00   Min.   :0.0000
##  1st Qu.:1.000     1st Qu.: 1.020   1st Qu.: 6.00   1st Qu.:0.0000
##  Median :1.000     Median : 1.760   Median : 9.00   Median :0.5000
##  Mean   :1.371     Mean   : 2.719   Mean   :11.47   Mean   :0.3523
##  3rd Qu.:1.000     3rd Qu.: 3.420   3rd Qu.:14.50   3rd Qu.:0.5000
##  Max.   :6.000     Max.   :27.000   Max.   :60.00   Max.   :1.0000
##  NA's   :2         NA's   :62       NA's   :30
##    q.mta_tax       q.tip_amount    q.tolls_amount    q.improvement_surcharge
##  Min.   :0.0000   Min.   : 0.000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.5000   1st Qu.: 0.000   1st Qu.:0.00000   1st Qu.:0.3000
##  Median :0.5000   Median : 0.000   Median :0.00000   Median :0.3000
##  Mean   :0.4871   Mean   : 1.029   Mean   :0.04671   Mean   :0.2923
##  3rd Qu.:0.5000   3rd Qu.: 1.700   3rd Qu.:0.00000   3rd Qu.:0.3000
##  Max.   :0.5000   Max.   :30.000   Max.   :5.54000   Max.   :0.3000
##                   NA's   :2        NA's   :7
##  q.target.total_amount    q.tlenkm      q.travel_time         q.espeed
##  Min.   :  0.00        Min.   : 0.000   Min.   :   0.000   Min.   : 3.239
##  1st Qu.:  7.80        1st Qu.: 1.609   1st Qu.:   5.767   1st Qu.:14.826
##  Median : 10.80        Median : 2.800   Median :   9.550   Median :18.613
##  Mean   : 13.93        Mean   : 4.358   Mean   :  19.863   Mean   :20.490
##  3rd Qu.: 17.00        3rd Qu.: 5.472   3rd Qu.:  16.125   3rd Qu.:23.647
##  Max.   :128.76        Max.   :69.314   Max.   :1438.183   Max.   :75.657
##                                                            NA's   :105
```

```
res.imputation<-imputePCA(df[,vars_quantitatives],ncp=5)
summary(res.imputation$completeObs)
```

```
##  q.pickup_longitude q.pickup_latitude q.dropoff_longitude q.dropoff_latitude
##  Min.   :-74.05     Min.   :40.58     Min.   :-74.06      Min.   :40.58
##  1st Qu.:-73.96     1st Qu.:40.70     1st Qu.:-73.97      1st Qu.:40.70
##  Median :-73.94     Median :40.75     Median :-73.94      Median :40.75
##  Mean   :-73.93     Mean   :40.75     Mean   :-73.94      Mean   :40.74
##  3rd Qu.:-73.92     3rd Qu.:40.80     3rd Qu.:-73.91      3rd Qu.:40.79
##  Max.   :-73.80     Max.   :40.86     Max.   :-73.80      Max.   :40.86
##  q.passenger_count q.trip_distance  q.fare_amount      q.extra
##  Min.   :1.000     Min.   : 0.010   Min.   : 1.00   Min.   :0.0000
##  1st Qu.:1.000     1st Qu.: 1.020   1st Qu.: 6.00   1st Qu.:0.0000
##  Median :1.000     Median : 1.770   Median : 9.00   Median :0.5000
##  Mean   :1.371     Mean   : 2.737   Mean   :11.65   Mean   :0.3523
##  3rd Qu.:1.000     3rd Qu.: 3.430   3rd Qu.:14.50   3rd Qu.:0.5000
##  Max.   :6.000     Max.   :32.462   Max.   :99.58   Max.   :1.0000
##    q.mta_tax       q.tip_amount    q.tolls_amount    q.improvement_surcharge
##  Min.   :0.0000   Min.   : 0.000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.5000   1st Qu.: 0.000   1st Qu.:0.00000   1st Qu.:0.3000
##  Median :0.5000   Median : 0.000   Median :0.00000   Median :0.3000
##  Mean   :0.4871   Mean   : 1.029   Mean   :0.04761   Mean   :0.2923
##  3rd Qu.:0.5000   3rd Qu.: 1.700   3rd Qu.:0.00000   3rd Qu.:0.3000
##  Max.   :0.5000   Max.   :30.000   Max.   :5.54000   Max.   :0.3000
##  q.target.total_amount    q.tlenkm      q.travel_time         q.espeed
##  Min.   :  0.00        Min.   : 0.000   Min.   :   0.000   Min.   :-46.26
##  1st Qu.:  7.80        1st Qu.: 1.609   1st Qu.:   5.767   1st Qu.: 14.81
##  Median : 10.80        Median : 2.800   Median :   9.550   Median : 18.60
##  Mean   : 13.93        Mean   : 4.358   Mean   :  19.863   Mean   : 20.20
##  3rd Qu.: 17.00        3rd Qu.: 5.472   3rd Qu.:  16.125   3rd Qu.: 23.64
##  Max.   :128.76        Max.   :69.314   Max.   :1438.183   Max.   : 77.48
```

We proceed now to fix all the numeric variables that have errors or outliers:

```
summary(res.imputation$completeObs[,"q.pickup_longitude"])
```

6.4.1.1 q.pickup_longitude

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -74.05  -73.96  -73.94  -73.93  -73.92  -73.80
```

```
summary(res.imputation$completeObs[,"q.pickup_latitude"])
```

6.4.1.2 q.pickup_latitude

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40.58   40.70   40.75   40.75   40.80   40.86
```

```
summary(res.imputation$completeObs[,"q.dropoff_longitude"])
```

6.4.1.3 q.dropoff_longitude

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -74.06  -73.97  -73.94  -73.94  -73.91  -73.80
```

```
summary(res.imputation$completeObs[,"q.dropoff_latitude"])
```

6.4.1.4 q.dropoff_latitude

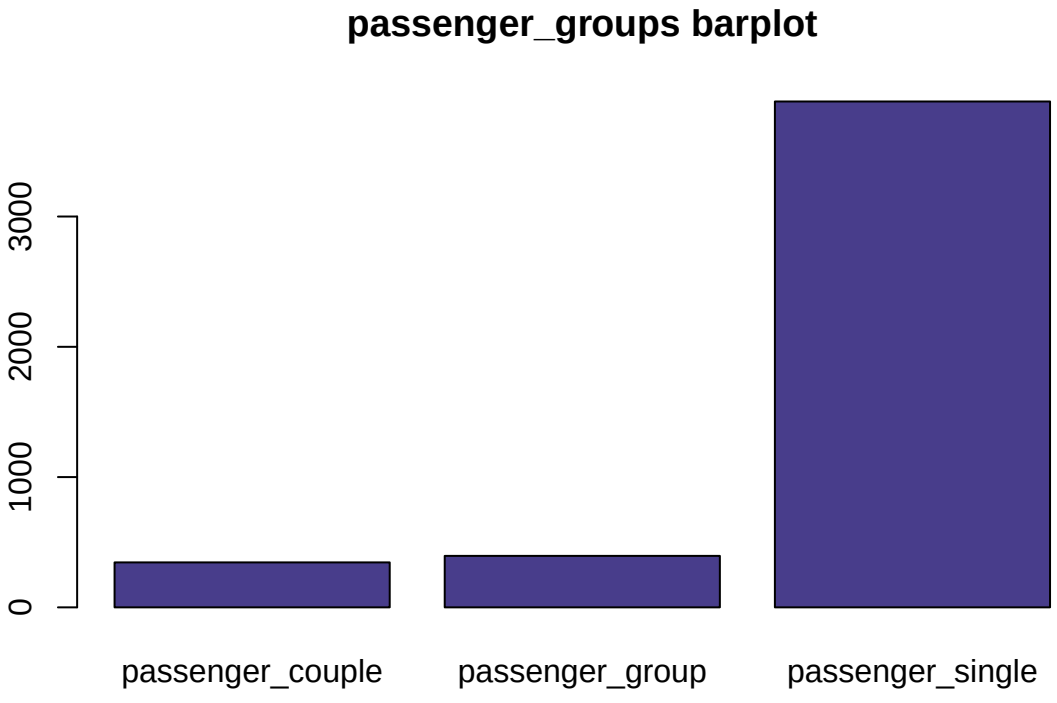
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40.58   40.70   40.75   40.74   40.79   40.86
```

6.4.1.5 q.passenger_count We decided to create categorical for this variable so we categorize it for single passengers, couple and groups (3 or more)

```
df$f.passenger_groups[res.imputation$completeObs[,"q.passenger_count"]  == 1] = "passenger_single"
df$f.passenger_groups[res.imputation$completeObs[,"q.passenger_count"] > 1 & res.imputation$completeObs[
df$f.passenger_groups[res.imputation$completeObs[,"q.passenger_count"] >= 3] = "passenger_group"
df$f.passenger_groups <- factor(df$f.passenger_groups)
```

We see the barplot in order to see the distribution of passenger per trip

```
barplot(table(df$f.passenger_groups),main="passenger_groups barplot",col="darkslateblue")
```




```
ll<-which(res.imputation$completeObs[,"q.trip_distance"] < 0)
res.imputation$completeObs[ll,"q.trip_distance"] <- 1
ll<-which(res.imputation$completeObs[,"q.trip_distance"] > 30)
res.imputation$completeObs[ll,"q.trip_distance"] <- 30
```

6.4.1.6 q.trip_distance

```
ll<-which(res.imputation$completeObs[,"q.fare_amount"] > 60)
res.imputation$completeObs[ll,"q.fare_amount"] <- 60
```

6.4.1.7 q.fare_amount

6.4.1.8 q.extra If we execute a table, we'll see that we have 0, 0'5 and 1 values, so we proceed to categorize this variable to see if has extra or not.

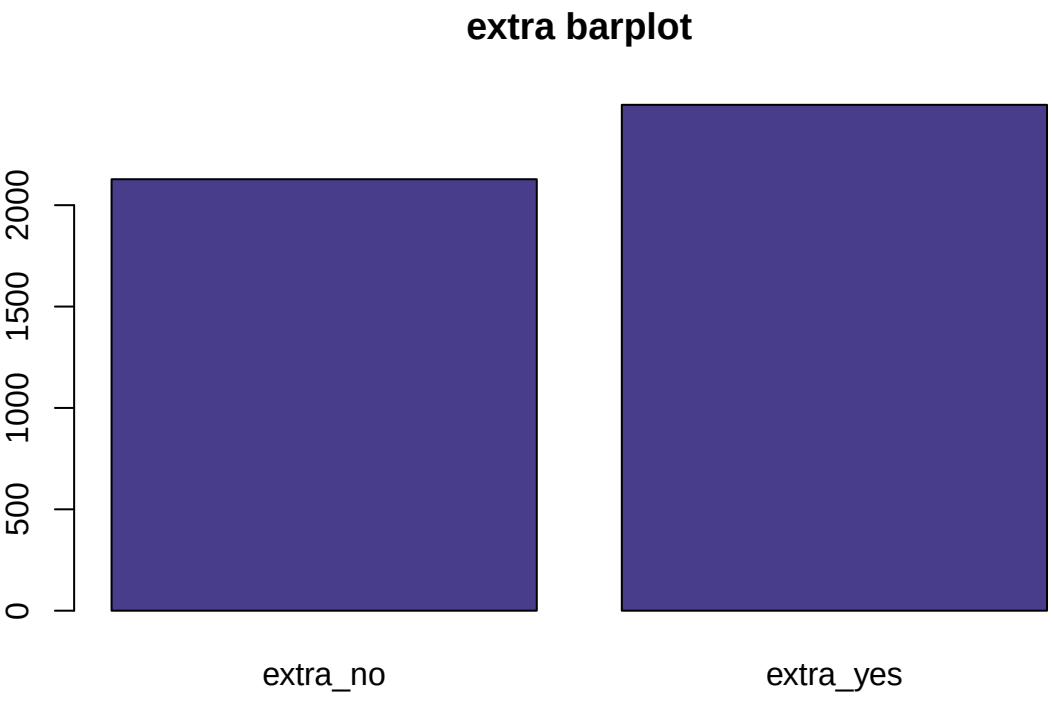
```
table(df$q.extra)
```

```
##
##    0  0.5    1
## 2128 1733  762
```

```
df$f.extra[df$q.extra == 0] = 0
df$f.extra[df$q.extra > 0] = 1
df$f.extra<-factor(df$f.extra, labels=c("extra_no","extra_yes"))
```

We see the barplot in order to see the distribution.

```
barplot(table(df$f.extra),main="extra barplot",col="darkslateblue")
```



q.mta_tax If we execute a summary, we'll see that every value should be 0.5 or 0, so we proceed to categorize this variable in order to see if the tax has been paid or not.

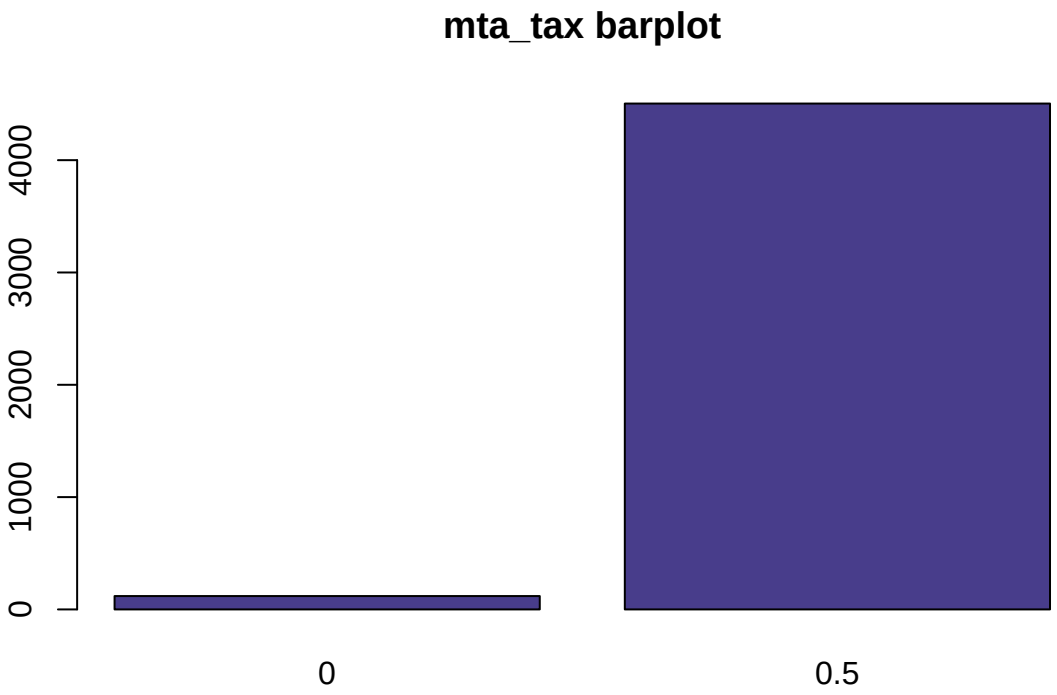
```
table(df$q.mta_tax)
```

```
##
##    0  0.5
##  119 4504
```

```
df$f.mta_tax<-factor(df$q.mta_tax, labels =c("mta_no","mta_yes"))
```

We see the barplot in order to see the distribution.

```
barplot(table(df$q.mta_tax),main="mta_tax barplot",col="darkslateblue")
```



q.tip_amount

```
ll<-which(res.imputation$completeObs[,"q.tip_amount"] > 17)
res.imputation$completeObs[ll,"q.tip_amount"] <- 17
```

We see that we have correct data, so we proceed to create the binary factor TipIsGiven.

```
df$f.target.tip_is_given[(res.imputation$completeObs[,"q.tip_amount"] > 0)] = "tip_yes"
df$f.target.tip_is_given[(res.imputation$completeObs[,"q.tip_amount"] == 0)] = "tip_no"
df$f.target.tip_is_given <- factor(df$f.target.tip_is_given)
summary(df$f.target.tip_is_given)
```

```
##  tip_no tip_yes
##    2882    1741
```

6.4.1.9 q.tolls_amount As we checked before the imputation and detected as errors those individuals with negative amount, the negative values found now are going to be set as 0 because they result negative during the imputation. After treating this values, we proceed to categorize this variable to see if an individual has paid or not for a toll.

```
ll<-which(res.imputation$completeObs[,"q.tolls_amount"] < 0)
res.imputation$completeObs[ll,"q.tolls_amount"] <- 0

df$f.paid_tolls[res.imputation$completeObs[,"q.tolls_amount"] == 0] = "tolls_no"
df$f.paid_tolls[res.imputation$completeObs[,"q.tolls_amount"] > 0] = "tolls_yes"
df$f.paid_tolls <- factor(df$f.paid_tolls)
```

6.4.1.10 q.improvement_surcharge If we execute a table, we'll see that every value should be 0.3 or 0, so we proceed to categorize this variable in order to see if the surcharge has been paid or not.

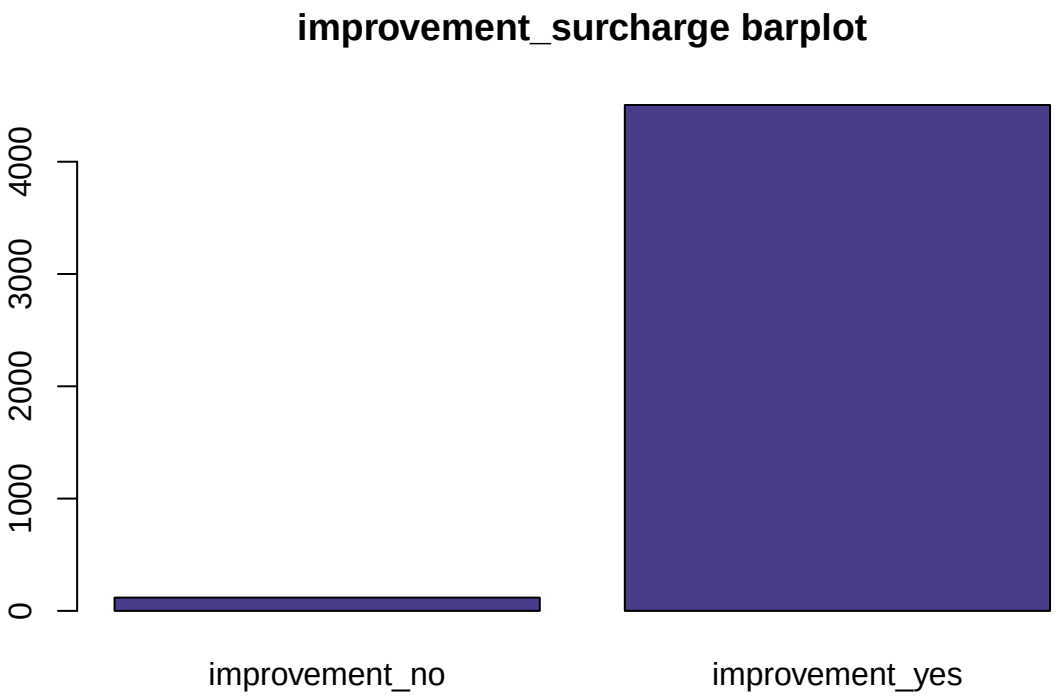
```
table(df$q.improvement_surcharge)
```

```
##
##    0  0.3
##  118 4505
```

```
df$f.improvement_surcharge<-factor(df$q.improvement_surcharge, labels=c("improvement_no","improvement_ye
```

We see the barplot in order to see the distribution.

```
barplot(table(df$f.improvement_surcharge),main="improvement_surcharge barplot",col="darkslateblue")
```




```
ll<-which(res.imputation$completeObs[,"q.tlenkm"] <= 1)
res.imputation$completeObs[ll,"q.tlenkm"] <- 1
ll<-which(res.imputation$completeObs[,"q.tlenkm"] > 48.28)
res.imputation$completeObs[ll,"q.tlenkm"] <- 48.28
```

6.4.1.11 q.tlenkm

```
ll<-which(res.imputation$completeObs[,"q.tlenkm"] > 60)
res.imputation$completeObs[ll,"q.tlenkm"] <- 60
```

6.4.1.12 q.travel_time

```
ll<-which(res.imputation$completeObs[,"q.espeed"] < 3)
res.imputation$completeObs[ll,"q.espeed"] <- 3
ll<-which(res.imputation$completeObs[,"q.espeed"] > 55)
res.imputation$completeObs[ll,"q.espeed"] <- 55
```

6.4.1.13 q.espeed

6.4.1.14 Store imputation We proceed to impute all NAs in our numerical variables that are stored in: `res.imputation$completeObs`

```
df[,vars_quantitatives] <- res.imputation$completeObs
```

6.4.2 Categorical variables / Factors

```
vars_categorical<-names(df)[c(1,2,3,17,18,20,26,27,28,29,30,31,32)]
summary(df[,vars_categorical])
```

```
##              f.vendor_id   f.store_and_fwd_flag            f.rate_code_id
##  vendor_id_mobile  : 973   flag-no :4605        rate_code_id_1    :4496
##  vendor_id_verifone:3650   flag-yes:  18        rate_code_id_other: 127
##
##
##      f.payment_type           f.trip_type               f.period
##  credit card:2096   trip_street_hail:4511   period night    :1642
##  cash       :2497   trip_dispatch   : 112   period morning  : 542
##  no paid    :  30                          period valley   :1260
##                                            period afternoon:1179
```

```
##       f.trip_distance_range        f.passenger_groups      f.extra
##  trip_dist_long  : 645      passenger_couple: 345     extra_no :2128
##  trip_dist_medium: 986      passenger_group : 395     extra_yes:2495
##  trip_dist_short :2930      passenger_single:3883
##  NA's            :  62
##    f.mta_tax    f.target.tip_is_given    f.paid_tolls
##  mta_no : 119   tip_no :2882          tolls_no :4576
##  mta_yes:4504   tip_yes:1741          tolls_yes:  47
##
##
##     f.improvement_surcharge
##  improvement_no : 118
##  improvement_yes:4505
##
##
```

```
res.input<-imputeMCA(df[,vars_categorical],ncp=10)
summary(res.input$completeObs)
```

```
##              f.vendor_id   f.store_and_fwd_flag            f.rate_code_id
##  vendor_id_mobile  : 973   flag-no :4605        rate_code_id_1    :4496
##  vendor_id_verifone:3650   flag-yes:  18        rate_code_id_other: 127
##
##
##      f.payment_type           f.trip_type               f.period
##  credit card:2096   trip_street_hail:4511   period night    :1642
##  cash       :2497   trip_dispatch   : 112   period morning  : 542
##  no paid    :  30                           period valley   :1260
##                                             period afternoon:1179
##       f.trip_distance_range        f.passenger_groups      f.extra
##  trip_dist_long  : 650      passenger_couple: 345     extra_no :2128
##  trip_dist_medium: 988      passenger_group : 395     extra_yes:2495
##  trip_dist_short :2985      passenger_single:3883
##
##    f.mta_tax    f.target.tip_is_given    f.paid_tolls
##  mta_no : 119   tip_no :2882          tolls_no :4576
##  mta_yes:4504   tip_yes:1741          tolls_yes:  47
##
##
##     f.improvement_surcharge
##  improvement_no : 118
##  improvement_yes:4505
##
##
```

6.4.2.1 Store imputation

We proceed to impute all NAs in our numerical variables that are stored in: `res.input$completeObs`

```
df[,vars_categorical] <- res.input$completeObs
```

6.4.3 Create some other factors after imputation

```
df$f.dist[df$q.trip_distance<=1.6] = "(0, 1.6]"
df$f.dist[(df$q.trip_distance>1.6) & (df$q.trip_distance<=3)] = "(1.6, 3]"
df$f.dist[(df$q.trip_distance>3) & (df$q.trip_distance<=5.5)] = "(3, 5.5]"
df$f.dist[(df$q.trip_distance>5.5) & (df$q.trip_distance<=30)] = "(5.5, 30]"
df$f.dist<-factor(df$f.dist)
```

6.4.3.1 f.dist

```
df$f.hour[(df$q.hour>=17) & (df$q.hour<18)] = "17"
df$f.hour[(df$q.hour>=18) & (df$q.hour<19)] = "18"
```

```
df$f.hour[(df$q.hour>=19) & (df$q.hour<20)] = "19"
df$f.hour[(df$q.hour>=20) & (df$q.hour<21)] = "20"
df$f.hour[(df$q.hour>=21) & (df$q.hour<22)] = "21"
df$f.hour[(df$q.hour>=22) & (df$q.hour<23)] = "22"
df$f.hour[(df$q.hour<17)] = "other"
df$f.hour[(df$q.hour>=23)] = "other"
df$f.hour<-factor(df$f.hour)
```

6.4.3.2 f.hour

```
df$f.espeed[(df$q.espeed>=3) & (df$q.espeed<20)]  = "[03,20)"
df$f.espeed[(df$q.espeed>=20) & (df$q.espeed<40)] = "[20,40)"
df$f.espeed[(df$q.espeed>=40) & (df$q.espeed<=55)] = "[40,55]"
df$f.espeed<-factor(df$f.espeed)
```

6.4.3.3 f.espeed

6.4.4 Describe these variables, to which other variables exist higher associations

6.4.4.1 Compute the correlation with all other variables.

We are skipping longitudes and latitudes.

```
library(mvoutlier)
library(FactoMineR)
vars_quantitatives_no_coords <- names(df)[c(8,9,10,11,12,13,14,15,16,21,22,23)]
res <- cor(df[,vars_quantitatives_no_coords])
round(res, 2)
```

```
##                         q.passenger_count q.trip_distance q.fare_amount q.extra
## q.passenger_count                    1.00            0.02          0.01    0.05
## q.trip_distance                      0.02            1.00          0.92   -0.05
## q.fare_amount                        0.01            0.92          1.00   -0.06
## q.extra                              0.05           -0.05         -0.06    1.00
## q.mta_tax                            0.00           -0.08         -0.10    0.15
## q.tip_amount                        -0.01            0.42          0.42    0.01
## q.tolls_amount                       0.02            0.20          0.20   -0.03
## q.improvement_surcharge              0.01           -0.07         -0.08    0.15
## q.target.total_amount                0.02            0.91          0.95   -0.01
## q.tlenkm                             0.02            0.99          0.91   -0.05
## q.travel_time                        0.00            0.11          0.12    0.03
## q.espeed                             0.02            0.57          0.41   -0.05
##                         q.mta_tax q.tip_amount q.tolls_amount
## q.passenger_count            0.00        -0.01           0.02
## q.trip_distance             -0.08         0.42           0.20
## q.fare_amount               -0.10         0.42           0.20
## q.extra                      0.15         0.01          -0.03
## q.mta_tax                    1.00         0.04           0.01
## q.tip_amount                 0.04         1.00           0.18
## q.tolls_amount               0.01         0.18           1.00
## q.improvement_surcharge      0.96         0.05           0.02
## q.target.total_amount       -0.05         0.57           0.25
## q.tlenkm                    -0.04         0.41           0.21
## q.travel_time                0.01         0.02           0.00
## q.espeed                    -0.08         0.21           0.16
##                         q.improvement_surcharge q.target.total_amount q.tlenkm
## q.passenger_count                          0.01                 0.02     0.02
## q.trip_distance                           -0.07                 0.91     0.99
## q.fare_amount                             -0.08                 0.95     0.91
## q.extra                                    0.15                -0.01    -0.05
## q.mta_tax                                  0.96                -0.05    -0.04
## q.tip_amount                               0.05                 0.57     0.41
## q.tolls_amount                             0.02                 0.25     0.21
## q.improvement_surcharge                    1.00                -0.03    -0.03
```

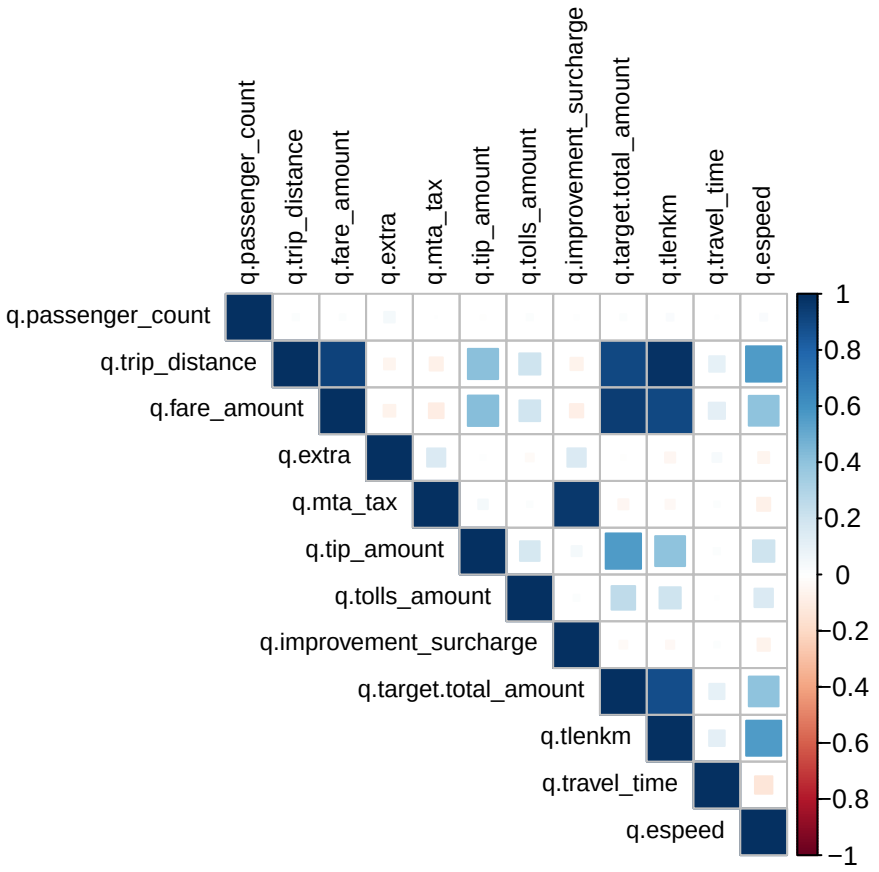
```
## q.target.total_amount                    -0.03                 1.00      0.88
## q.tlenkm                                 -0.03                 0.88      1.00
## q.travel_time                             0.01                 0.11      0.11
## q.espeed                                 -0.07                 0.40      0.56
##                          q.travel_time q.espeed
## q.passenger_count                 0.00     0.02
## q.trip_distance                   0.11     0.57
## q.fare_amount                     0.12     0.41
## q.extra                           0.03    -0.05
## q.mta_tax                         0.01    -0.08
## q.tip_amount                      0.02     0.21
## q.tolls_amount                    0.00     0.16
## q.improvement_surcharge           0.01    -0.07
## q.target.total_amount             0.11     0.40
## q.tlenkm                          0.11     0.56
## q.travel_time                     1.00    -0.14
## q.espeed                         -0.14     1.00
```

```
library(corrplot)
```

6.4.4.2 Rank these variables according the correlation:

```
## corrplot 0.84 loaded
```

```
corrplot(res,method="square",type="upper",tl.col="black",tl.cex=0.75,)
```



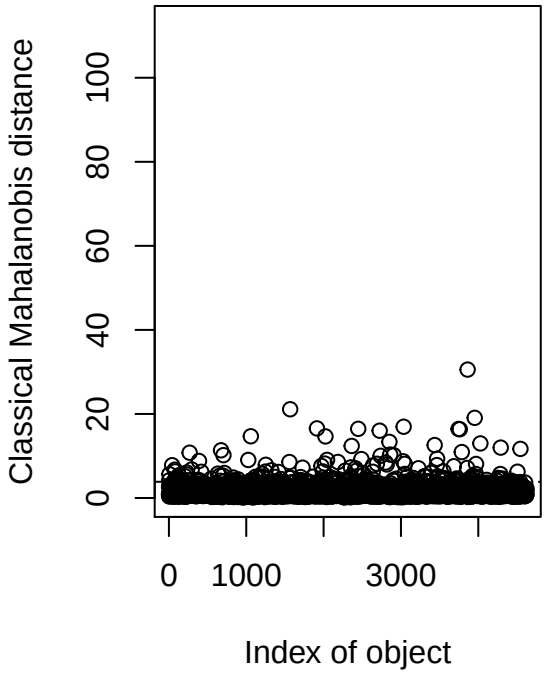
As we can see in this graph, we have the correlation between all quantitative variables. We must say, however, that there are two variables (espeed and traveltime) which we had to modify when making the imputation.

Now, let's describe each correlation we obtained in the graph (we will only mention one relation once): * Diagonals: Being exactly the same variable, it is directly related to itself. * q.passanger_count: not too related to any other not seen before * q.trip_distance + w/ q.fare_amount: More distance, more time, therefore more price. + w/ q.tip_amount: If the trip has been longer, there may be more reason to tip. + w/ q.target.total_amount: As before, more distance, more time, therefore more price. + w/ q.tlenkm: They are exactly the same, only with a metric change. + w/ q.travel_time: The further away, the longer. + w/ q.espeed: The reason we think these variables are related to a direct and positive proportion is that since short trips have to be, logically cheaper, what taxi drivers do is slow down so that the trip take longer and thus charge more. Therefore, by increasing the

distance of the journey, taxi drivers do not need to go so slow and therefore the speed increases. * q.fare_amount: + w/ q.tip_amount: In the USA it is normal to give a tip proportional to the price of the service that has been offered. + w/ q.target.total_amount: The variable q.target.total_amount is equivalent to q.fare_amount plus the fees, tips, among others, that have been applied to the trip. + w/ q.tlenkm: As before, more distance, more time, therefore more price + w/ q.travel_time: More time, more price. + w/ q.espeed: As we said before, more speed means more distance, therefore more travel time, causing more price. * q.extra: not too related to any other not seen before * q.mta_tax: + w/ q.improvement_subcharge: if there's a tax, the most probable thing to happen is that there's an improvement subcharge too * q.tip_amount: + w/ q.target.total_amount: As before, in the USA it is normal to give a tip proportional to the price of the service that has been offered. + w/ q.tlenkm: If the trip has been longer, there may be more reason to tip. + w/ q.travel_time: The longer it takes, the more price, and therefore the more tip given the proportionality. + w/ q.espeed: The more speed, as we said before, the more distance, and therefore the longer it takes. This causes more price and therefore more tip. * q.tolls_amount: not too related to any other not seen before * q.improvement_subcharge: not too related to any other not seen before * q.target.total_amount: + w/ q.tlenkm: More distance, more time, therefore more price. + w/ q.espeed: As we said before, more speed means more distance, therefore more travel time, causing more price. * q.tlenkm + Same as for q.trip_distance + q.espeed correlation. * q.travel_time: not too related to any other not seen before * q.espeed: not too related to any other not seen before

---

```
library(mvoutlier)
library(chemometrics)

#"Trip_distance" "Fare_amount"   "Total_amount"  "espeed"

multivariant_outliers <- Moutlier(df[, c(9,10,16,23)], quantile = 0.995)
```

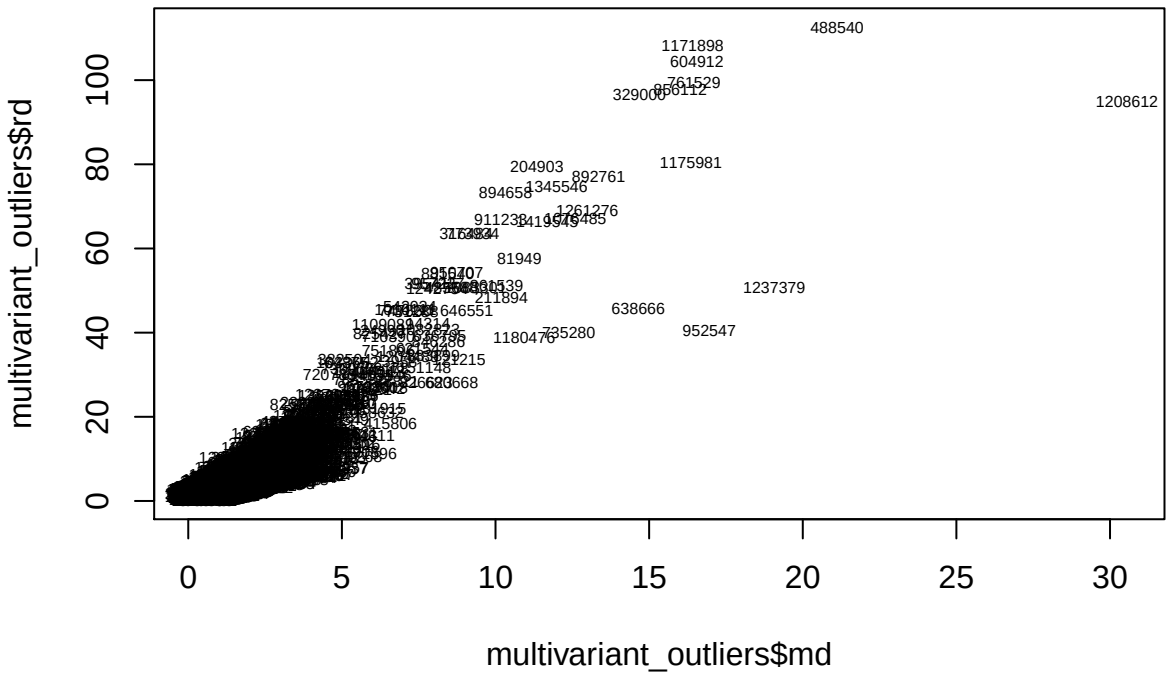


6.4.4.3 Identify individuals considered as multivariant outliers

```
multivariant_outliers$cutoff
```

```
## [1] 3.854901
```

```
par(mfrow=c(1,1))
plot(multivariant_outliers$md, multivariant_outliers$rd, type="n")
text(multivariant_outliers$md, multivariant_outliers$rd, labels=rownames(df[, c(9,10,16,23)]), cex=0.5)
```




```
ll_mvoutliers <- subset(df, (multivariant_outliers$md>10 | multivariant_outliers$rd>60)); summary(ll_mvo
```

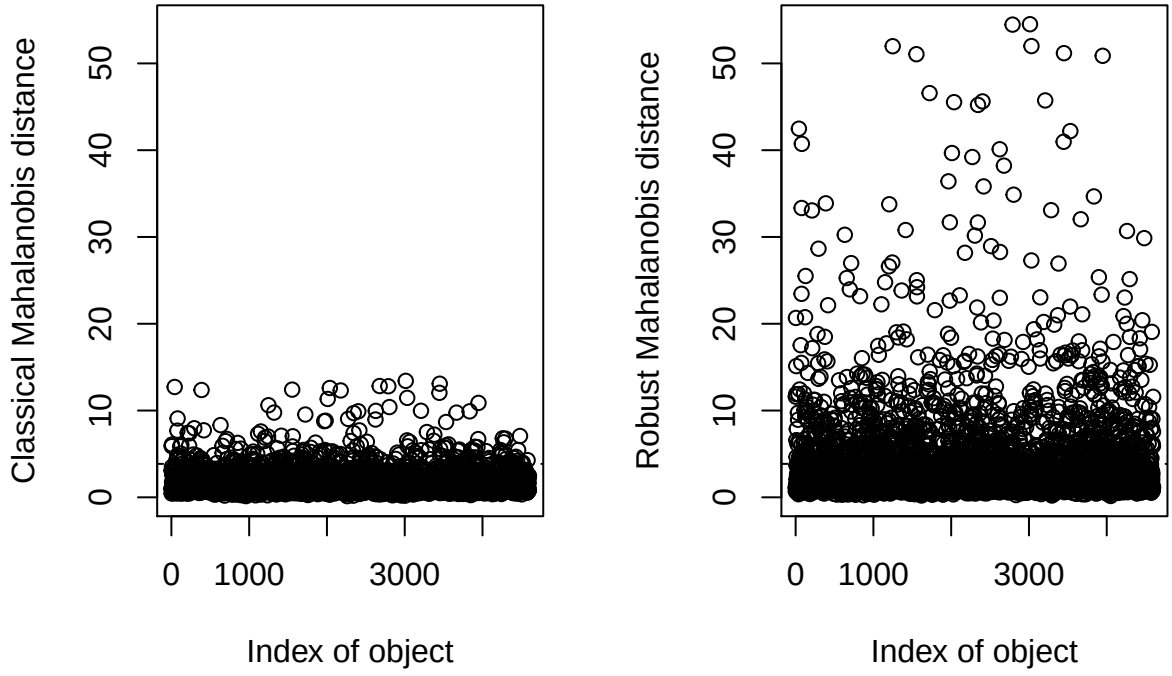
```
##              f.vendor_id f.store_and_fwd_flag            f.rate_code_id
##  vendor_id_mobile  : 6   flag-no :24          rate_code_id_1    :14
##  vendor_id_verifone:20   flag-yes: 2          rate_code_id_other:12
##
##
##
##
##
##  q.pickup_longitude q.pickup_latitude q.dropoff_longitude q.dropoff_latitude
##  Min.   :-74.01     Min.   :40.58     Min.   :-74.01      Min.   :40.59
##  1st Qu.:-73.95     1st Qu.:40.67     1st Qu.:-73.98      1st Qu.:40.65
##  Median :-73.93     Median :40.71     Median :-73.95      Median :40.71
##  Mean   :-73.93     Mean   :40.72     Mean   :-73.95      Mean   :40.71
##  3rd Qu.:-73.91     3rd Qu.:40.80     3rd Qu.:-73.92      3rd Qu.:40.76
##  Max.   :-73.81     Max.   :40.85     Max.   :-73.81      Max.   :40.84
##
##  q.passenger_count q.trip_distance q.fare_amount      q.extra
##  Min.   :1.000     Min.   : 0.05   Min.   : 2.50   Min.   :0.0000
##  1st Qu.:1.000     1st Qu.: 5.48   1st Qu.:26.37   1st Qu.:0.0000
##  Median :1.000     Median :10.52   Median :52.00   Median :0.0000
##  Mean   :1.308     Mean   :13.58   Mean   :41.02   Mean   :0.2692
##  3rd Qu.:1.000     3rd Qu.:22.84   3rd Qu.:59.41   3rd Qu.:0.5000
##  Max.   :5.000     Max.   :30.00   Max.   :60.00   Max.   :1.0000
##
##    q.mta_tax       q.tip_amount    q.tolls_amount   q.improvement_surcharge
##  Min.   :0.0000   Min.   : 0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1250   1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.:0.0750
##  Median :0.5000   Median : 1.007   Median :0.0000   Median :0.3000
##  Mean   :0.3654   Mean   : 5.343   Mean   :0.7118   Mean   :0.2192
##  3rd Qu.:0.5000   3rd Qu.: 9.570   3rd Qu.:0.0000   3rd Qu.:0.3000
##  Max.   :0.5000   Max.   :17.000   Max.   :5.5400   Max.   :0.3000
##
##  q.target.total_amount     f.payment_type           f.trip_type     q.hour
##  Min.   :  0.00        credit card:15     trip_street_hail:19   Min.   : 0.00
##  1st Qu.: 51.00        cash       :10     trip_dispatch   : 7   1st Qu.: 7.50
##  Median : 64.65        no paid    : 1                          Median :14.50
##  Mean   : 65.25                                                Mean   :13.38
##  3rd Qu.: 94.33                                                3rd Qu.:19.75
##  Max.   :128.76                                                Max.   :23.00
##
```

```
##              f.period     q.tlenkm      q.travel_time         q.espeed
##  period night    :11   Min.   : 1.000   Min.   : 0.03333   Min.   : 6.148
##  period morning  : 2   1st Qu.: 1.132   1st Qu.: 0.46250   1st Qu.:26.057
##  period valley   : 6   Median :16.834   Median :38.16667   Median :31.603
##  period afternoon: 7   Mean   :20.055   Mean   :35.93782   Mean   :33.373
##                        3rd Qu.:36.757   3rd Qu.:52.46250   3rd Qu.:49.525
##                        Max.   :48.280   Max.   :96.40000   Max.   :55.000
##
##  qual.pickup        qual.dropoff           f.trip_distance_range
##  Length:26          Length:26          trip_dist_long  :14
##  Class :character   Class :character   trip_dist_medium: 2
##  Mode  :character   Mode  :character   trip_dist_short :10
##
##
##
##
##        f.passenger_groups      f.extra     f.mta_tax  f.target.tip_is_given
##  passenger_couple: 1       extra_no :16   mta_no : 7   tip_no :13
##  passenger_group : 2       extra_yes:10   mta_yes:19   tip_yes:13
##  passenger_single:23
##
##
##
##
##     f.paid_tolls    f.improvement_surcharge       f.dist     f.hour
##  tolls_no :21    improvement_no : 7         (0, 1.6] : 4   17   : 0
##  tolls_yes: 5    improvement_yes:19         (1.6, 3] : 0   18   : 3
##                                             (3, 5.5] : 3   19   : 1
##                                             (5.5, 30]:19   20   : 3
##                                                            21   : 2
##                                                            22   : 0
##                                                            other:17
##     f.espeed
##  [03,20): 5
##  [20,40):14
##  [40,55]: 7
##
##
##
##
```

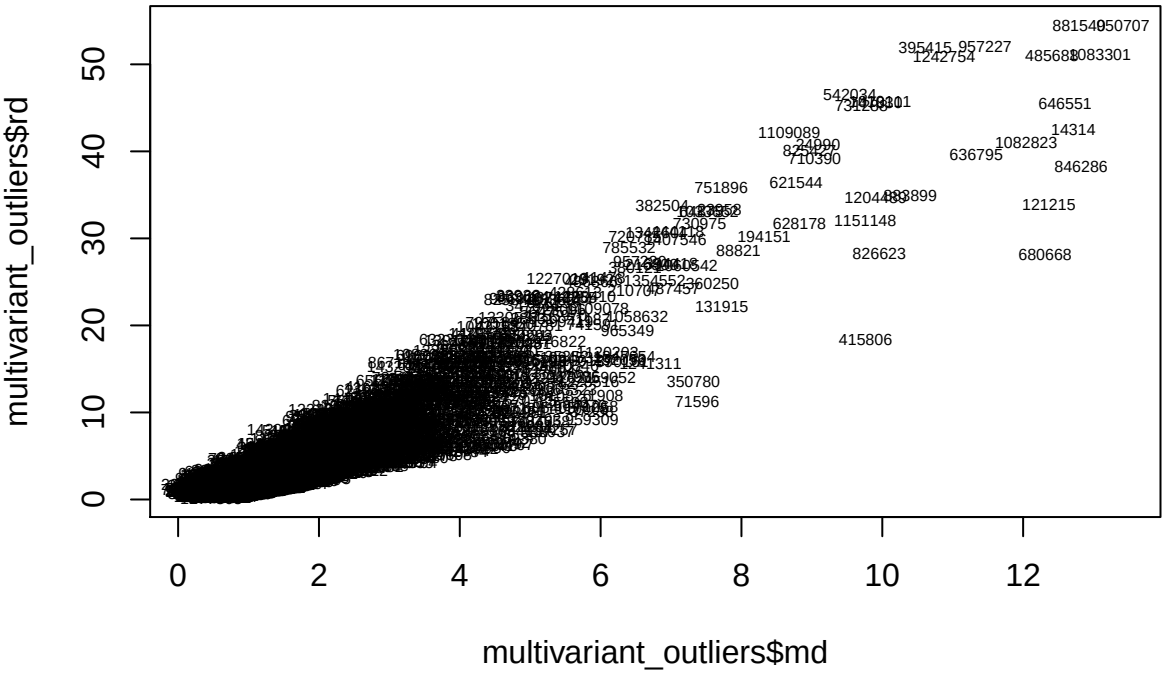
As we can see, above the defined line we have all the possible observations that we call multivariate outliers. These mean that, viewed only from the point of view of a variable, it does not have to be an outlier, but that viewed with various dimensions (variables), it may be so.

// !!! FALTA COMENTAR OUTLIERS MULTIVARIANTS

```
df <- subset(df, !(multivariant_outliers$md>10 | multivariant_outliers$rd>60))

multivariant_outliers <- Moutlier(df[, c(9,10,16,23)], quantile = 0.995)
```




```
par(mfrow=c(1,1))
plot(multivariant_outliers$md, multivariant_outliers$rd, type="n")
text(multivariant_outliers$md, multivariant_outliers$rd, labels=rownames(df[, c(9,10,16,23)]), cex=0.5)
```



We can see now that there are not multivariant outliers in our dataframe.

6.5 Profiling

6.5.1 Numeric target: q.target.total_amount

Profiling is used to finish profiling our sample.

We will now proceed to the profiling that asks us for our numeric target (Total_amount) and then we have to use the original variables and factors.

In order to observe the relationship of our numerical target with the other variables we use the condes tool that provides us with information about the relationships between the indicated variables and the target.

```
library(FactoMineR)
summary(df$q.target.total_amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##    0.00    7.80   10.56   13.64   16.80   77.16
```

```
vars_res<-names(df)[c(16,30)]
vars_quantitatives<-names(df)[c(8:15,21:23)]
vars_categorical<-names(df)[c(1:3,17,18,20,26:29,31,32)]

res.condes <- condes(df[, c(vars_res,vars_quantitatives, vars_categorical)],1)
```

Let's now look at the correlations between our Total_amount target and the variables in the following groups. We will basically look at p.value, which we know that the smaller the correlation between the variables.

```
res.condes$quanti
```

6.5.1.0.1 Numerical variables

```
##                  correlation       p.value
## q.fare_amount     0.97485988  0.000000e+00
## q.trip_distance   0.93406225  0.000000e+00
## q.tlenkm          0.92187648  0.000000e+00
## q.tip_amount      0.56127714  0.000000e+00
## q.espeed          0.39613486 1.421071e-172
## q.tolls_amount    0.23802440  3.144377e-60
## q.travel_time     0.11290029  1.625434e-14
## q.mta_tax        -0.03030698  3.990201e-02
```

For the lowest p.values:

- q.fare_amount: The variable q.target.total_amount is equivalent to q.fare_amount plus the fees, tips, among others, that have been applied to the trip.
- q.trip_distance: As before, more distance, more time, therefore more price.
- q.tlenkm: More distance, more time, therefore more price.
- q.tip_amount: The more you pay, since the tip is a proportion of the final price, the more it will increase.
- q.espeed: As we said before, more speed means more distance, therefore more travel time, causing more price.

```
res.condes$quali
```

6.5.1.0.2 Qualitative variables

```
##                                 R2      p.value
## f.trip_distance_range 0.6827079230 0.000000e+00
## f.paid_tolls          0.0754334605 2.364785e-80
## f.target.tip_is_given 0.0657915150 5.688596e-70
## f.payment_type        0.0593330010 9.600215e-62
## f.rate_code_id        0.0038737420 2.412314e-05
## f.mta_tax             0.0009185128 3.990201e-02
```

For the lowest p.values:

- f.trip_distance_range: Obviously, the longer the journey, the longer it will take and the more price it will have.
- f.paid_tolls: The variable q.target.total_amount is equivalent to f.paid_tolls plus the fees, tips, among others, that have been applied to the trip.
- f.target.tip_is_given: Like before, the more you pay, since the tip is a proportion of the final price, the more it will increase.

```
res.condes$category
```

6.5.1.0.3 Categorical variables

```
##                                        Estimate      p.value
## f.trip_distance_range=trip_dist_long  11.3373225 0.000000e+00
## f.paid_tolls=tolls_yes                12.8813000 2.364785e-80
## f.target.tip_is_given=tip_yes          2.3631665 5.688596e-70
## f.payment_type=credit card             2.2944595 8.798382e-63
```

```
## f.rate_code_id=rate_code_id_other        1.7783900 2.412314e-05
## f.period=period morning                  0.6908169 2.778052e-02
## f.mta_tax=mta_no                         0.8772005 3.990201e-02
## f.mta_tax=mta_yes                       -0.8772005 3.990201e-02
## f.rate_code_id=rate_code_id_1           -1.7783900 2.412314e-05
## f.trip_distance_range=trip_dist_medium  -1.5397811 2.318709e-46
## f.payment_type=cash                     -2.0845145 1.489206e-62
## f.target.tip_is_given=tip_no            -2.3631665 5.688596e-70
## f.paid_tolls=tolls_no                  -12.8813000 2.364785e-80
## f.trip_distance_range=trip_dist_short   -9.7975414 0.000000e+00
```

For the lowest p.values:

- f.trip_distance_range=trip_dist_long: We can see that, the further away, the more correlation, as it takes longer to travel.
- f.paid_tolls=tolls_yes: If tolls are paid, then there's more cost at the end.
- f.target.tip_is_given=tip_yes:We see that it is more likely to tip if the price is high.
- f.payment_type=credit card: We see that it is easier for the guy to be with credit card if the trip costs more.
- f.rate_code_id: As we have seen before, virtually all observations were of type 1. Therefore it is not worth looking at the correlation.
- f.period=period morning: We see that in the morning travel costs less.

6.5.2 Factor (Y.bin - f.target.tip_is_given)

And now, we are profiling the qualitative target:

```
res.catdes <- catdes(df[, c(vars_res,vars_quantitatives, vars_categorical)],2)
```

Let's now look at the correlations between our f.target.tip_is_given target and the variables in the following groups. We will basically look at p.value, which we know that the smaller the correlation between the variables.

```
res.catdes$test.chi2
```

6.5.2.0.1 Test.Chi2

```
##                               p.value df
## f.payment_type           0.000000e+00  2
## f.trip_distance_range    2.785249e-23  2
## f.mta_tax                5.054079e-06  1
## f.improvement_surcharge  6.545475e-06  1
## f.trip_type              1.208825e-05  1
## f.rate_code_id           1.463909e-05  1
## f.period                 5.138587e-05  3
## f.paid_tolls             3.327123e-04  1
```

For the lowest p.values:

- f.payment_type: We see that it is very likely that there will be a tip if it is paid in a concise manner.
- f.trip_distance_range: As we can see, there is tip as long as the trip is, or very short, or very long.

```
res.catdes$quanti.var
```

6.5.2.0.2 Quantitative variables

```
##                                Eta2      P-value
## q.tip_amount            0.545641143 0.000000e+00
## q.target.total_amount   0.065791515 5.688596e-70
## q.fare_amount           0.015030865 7.324378e-17
## q.trip_distance         0.013289537 4.499661e-15
## q.tlenkm                0.013272638 4.683133e-15
## q.espeed                0.007569834 3.449643e-09
## q.mta_tax               0.004528319 4.960153e-06
## q.improvement_surcharge 0.004420639 6.430281e-06
## q.tolls_amount          0.003369017 8.228375e-05
```

For the lowest p.values:

- q.tip_amount: If there is a tip, it must have value.
- q.target.total_amount: We see that it is more likely to tip if the price is high.
- q.fare_amount: We see that it is more likely to tip if the price is high.
- q.trip_distance: Exactly the same as above.
- q.tlenkm: The more distance, the more time, therefore the more price. So, more chances of there being a tip.

```
res.catdes$category
```

6.5.2.0.3 > Categorical variables

```
## $tip_no
##                                           Cla/Mod    Mod/Cla     Global
## f.payment_type=cash                     100.00000 86.6852562 54.1005003
## f.trip_distance_range=trip_dist_short    67.73109 70.2335308 64.7161192
## f.payment_type=no paid                  100.00000  1.0108052  0.6308462
## f.mta_tax=mta_no                         83.03571  3.2415476  2.4363715
## f.improvement_surcharge=improvement_no   82.88288  3.2066922  2.4146182
## f.trip_type=trip_dispatch                82.85714  3.0324155  2.2840983
## f.rate_code_id=rate_code_id_other        81.73913  3.2764029  2.5016315
## f.period=period valley                   67.30463 29.4179157 27.2786600
## f.paid_tolls=tolls_no                    62.65642 99.4771697 99.0863607
## f.period=period morning                  56.48148 10.6308818 11.7467914
## f.paid_tolls=tolls_yes                   35.71429  0.5228303  0.9136393
## f.rate_code_id=rate_code_id_1            61.91432 96.7235971 97.4983685
## f.trip_type=trip_street_hail             61.93232 96.9675845 97.7159017
## f.improvement_surcharge=improvement_yes  61.90370 96.7933078 97.5853818
## f.mta_tax=mta_yes                        61.89521 96.7584524 97.5636285
## f.trip_distance_range=trip_dist_medium   54.05680 18.5779017 21.4487709
## f.trip_distance_range=trip_dist_long     50.47170 11.1885674 13.8351099
## f.payment_type=credit card               16.96300 12.3039387 45.2686535
##                                             p.value    v.test
## f.payment_type=cash                    0.000000e+00       Inf
## f.trip_distance_range=trip_dist_short  1.081649e-23 10.033894
## f.payment_type=no paid                 1.094456e-06  4.873847
## f.mta_tax=mta_no                       1.634763e-06  4.794015
## f.improvement_surcharge=improvement_no 2.209458e-06  4.733257
## f.trip_type=trip_dispatch              4.384881e-06  4.592252
## f.rate_code_id=rate_code_id_other      5.893910e-06  4.530160
## f.period=period valley                 2.460929e-05  4.218353
## f.paid_tolls=tolls_no                  5.031113e-04  3.479094
## f.period=period morning                2.667731e-03 -3.003637
## f.paid_tolls=tolls_yes                 5.031113e-04 -3.479094
## f.rate_code_id=rate_code_id_1          5.893910e-06 -4.530160
## f.trip_type=trip_street_hail           4.384881e-06 -4.592252
## f.improvement_surcharge=improvement_yes 2.209458e-06 -4.733257
## f.mta_tax=mta_yes                      1.634763e-06 -4.794015
## f.trip_distance_range=trip_dist_medium 1.393474e-09 -6.056232
## f.trip_distance_range=trip_dist_long   3.929634e-11 -6.606707
## f.payment_type=credit card             0.000000e+00      -Inf
##
## $tip_yes
##                                           Cla/Mod    Mod/Cla     Global
## f.payment_type=credit card               83.03700 100.000000 45.2686535
## f.trip_distance_range=trip_dist_long     49.52830  18.229167 13.8351099
## f.trip_distance_range=trip_dist_medium   45.94320  26.215278 21.4487709
## f.mta_tax=mta_yes                        38.10479  98.900463 97.5636285
## f.improvement_surcharge=improvement_yes  38.09630  98.900463 97.5853818
## f.trip_type=trip_street_hail             38.06768  98.958333 97.7159017
## f.rate_code_id=rate_code_id_1            38.08568  98.784722 97.4983685
## f.paid_tolls=tolls_yes                   64.28571   1.562500  0.9136393
## f.period=period morning                  43.51852  13.599537 11.7467914
## f.paid_tolls=tolls_no                    37.34358  98.437500 99.0863607
## f.period=period valley                   32.69537  23.726852 27.2786600
```

```
## f.rate_code_id=rate_code_id_other       18.26087   1.215278  2.5016315
## f.trip_type=trip_dispatch               17.14286   1.041667  2.2840983
## f.improvement_surcharge=improvement_no  17.11712   1.099537  2.4146182
## f.mta_tax=mta_no                        16.96429   1.099537  2.4363715
## f.payment_type=no paid                   0.00000   0.000000  0.6308462
## f.trip_distance_range=trip_dist_short   32.26891  55.555556 64.7161192
## f.payment_type=cash                      0.00000   0.000000 54.1005003
##                                              p.value     v.test
## f.payment_type=credit card              0.000000e+00        Inf
## f.trip_distance_range=trip_dist_long    3.929634e-11   6.606707
## f.trip_distance_range=trip_dist_medium  1.393474e-09   6.056232
## f.mta_tax=mta_yes                       1.634763e-06   4.794015
## f.improvement_surcharge=improvement_yes 2.209458e-06   4.733257
## f.trip_type=trip_street_hail            4.384881e-06   4.592252
## f.rate_code_id=rate_code_id_1           5.893910e-06   4.530160
## f.paid_tolls=tolls_yes                  5.031113e-04   3.479094
## f.period=period morning                 2.667731e-03   3.003637
## f.paid_tolls=tolls_no                   5.031113e-04  -3.479094
## f.period=period valley                  2.460929e-05  -4.218353
## f.rate_code_id=rate_code_id_other       5.893910e-06  -4.530160
## f.trip_type=trip_dispatch               4.384881e-06  -4.592252
## f.improvement_surcharge=improvement_no  2.209458e-06  -4.733257
## f.mta_tax=mta_no                        1.634763e-06  -4.794015
## f.payment_type=no paid                  1.094456e-06  -4.873847
## f.trip_distance_range=trip_dist_short   1.081649e-23 -10.033894
## f.payment_type=cash                     0.000000e+00       -Inf
```

- f.payment_type: As we saw before, there is only a tip if the payment is done with a credit card.
- f.trip_distance_range: As we can see, there is tip as long as the trip is, or very short, or very long.
- f.mta_tax: We see that it is very likely that there will be a tip if there is a tax included.
- f.improvement_surcharge: We see that it is very likely that there will be a tip if there is the improvement subcharge included.
- f.trip_type: We don't think the type of trip is important.
- f.rate_code_id: As we have seen before, virtually all observations were of type 1. Therefore it is not worth looking at the correlation.
- f.period: We see that in the morning people are not in a very good mood and are more inclined to tip the "valley".