Deliverable 1

Data Processing, Description, Validation and Profiling

Júlia Gasull i Claudia Sánchez

# 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>:

## 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

# Load Required Packages for this deliverable

We load the necessary packages and set working directory

setwd("~/Documents/uni/FIB-ADEI-LAB/finaldeliverable")  
  
# Load Required Packages  
options(contrasts=c("contr.treatment","contr.treatment"))  
  
requiredPackages <- c("missMDA","chemometrics","mvoutlier","effects","FactoMineR","car", "factoextra","RColorBrewer","dplyr","ggmap","ggthemes","knitr")  
missingPackages <- requiredPackages[!(requiredPackages %in% installed.packages()[,"Package"])]  
  
if(length(missingPackages)) install.packages(missingPackages)  
lapply(requiredPackages, require, character.only = TRUE)

## 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.

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

Data: green\_tripdata\_2016-01

setwd("~/Documents/uni/FIB-ADEI-LAB/finaldeliverable")  
filepath<-"~/Documents/uni/FIB-ADEI-LAB/finaldeliverable"  
df<-read.table(paste0(filepath,"/green\_tripdata\_2016-01.csv"),header=T, sep=",")

Select your 5000 register sample (random sample). Use birthday of 1 member of the group –> Júlia’s one

set.seed(180998)  
sam<-as.vector(sort(sample(1:nrow(df),5000)))

Verification and storage of the sample

head(df)

## VendorID lpep\_pickup\_datetime Lpep\_dropoff\_datetime Store\_and\_fwd\_flag  
## 1 2 2016-01-01 00:29:24 2016-01-01 00:39:36 N  
## 2 2 2016-01-01 00:19:39 2016-01-01 00:39:18 N  
## 3 2 2016-01-01 00:19:33 2016-01-01 00:39:48 N  
## 4 2 2016-01-01 00:22:12 2016-01-01 00:38:32 N  
## 5 2 2016-01-01 00:24:01 2016-01-01 00:39:22 N  
## 6 2 2016-01-01 00:32:59 2016-01-01 00:39:35 N  
## RateCodeID Pickup\_longitude Pickup\_latitude Dropoff\_longitude  
## 1 1 -73.92864 40.68061 -73.92428  
## 2 1 -73.95267 40.72318 -73.92392  
## 3 1 -73.97161 40.67611 -74.01316  
## 4 1 -73.98950 40.66958 -74.00065  
## 5 1 -73.96473 40.68285 -73.94072  
## 6 1 -73.89114 40.74646 -73.86774  
## Dropoff\_latitude Passenger\_count Trip\_distance Fare\_amount Extra MTA\_tax  
## 1 40.69804 1 1.46 8.0 0.5 0.5  
## 2 40.76138 1 3.56 15.5 0.5 0.5  
## 3 40.64607 1 3.79 16.5 0.5 0.5  
## 4 40.68903 1 3.01 13.5 0.5 0.5  
## 5 40.66301 1 2.55 12.0 0.5 0.5  
## 6 40.74211 1 1.37 7.0 0.5 0.5  
## Tip\_amount Tolls\_amount Ehail\_fee improvement\_surcharge Total\_amount  
## 1 1.86 0 NA 0.3 11.16  
## 2 0.00 0 NA 0.3 16.80  
## 3 4.45 0 NA 0.3 22.25  
## 4 0.00 0 NA 0.3 14.80  
## 5 0.00 0 NA 0.3 13.30  
## 6 0.00 0 NA 0.3 8.30  
## Payment\_type Trip\_type  
## 1 1 1  
## 2 2 1  
## 3 1 1  
## 4 2 1  
## 5 2 1  
## 6 2 1

df<-df[sam,]  
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

Save the image

save.image("Taxi5000\_raw.RData")

## 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)   
}

# 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)) # rows - trips  
jmis<-rep(0,2\*ncol(df)) # columns - variables  
  
mis1<-countNA(df)  
imis<-mis1$mis\_ind   
  
iouts<-rep(0,nrow(df)) # rows - trips  
jouts<-rep(0,2\*ncol(df)) # columns - variables  
  
ierrs<-rep(0,nrow(df)) # rows - trips  
jerrs<-rep(0,2\*ncol(df)) # columns - variables

# Univariate Descriptive Analysis

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)

## [1] "VendorID" "lpep\_pickup\_datetime" "Lpep\_dropoff\_datetime"  
## [4] "Store\_and\_fwd\_flag" "RateCodeID" "Pickup\_longitude"   
## [7] "Pickup\_latitude" "Dropoff\_longitude" "Dropoff\_latitude"   
## [10] "Passenger\_count" "Trip\_distance" "Fare\_amount"   
## [13] "Extra" "MTA\_tax" "Tip\_amount"   
## [16] "Tolls\_amount" "Ehail\_fee" "improvement\_surcharge"  
## [19] "Total\_amount" "Payment\_type" "Trip\_type"

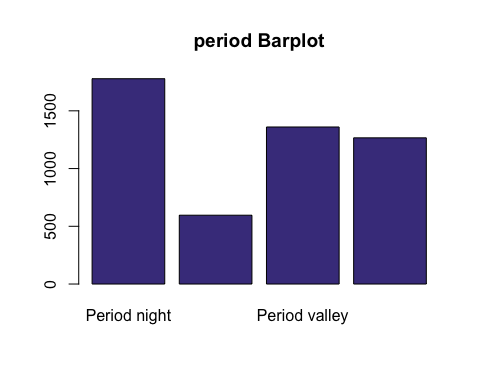
## Qualitative Variables (Factors) / Categorical

**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.

### New variable: Period

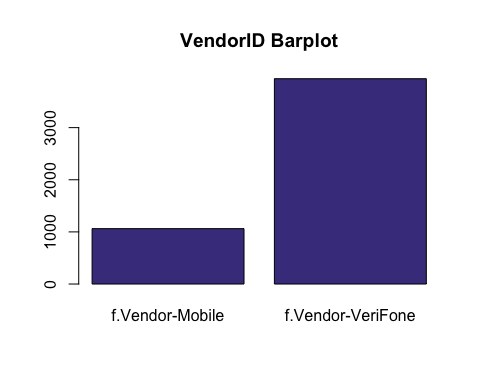
df$hour<-as.numeric(substr(strptime(df$lpep\_pickup\_datetime, "%Y-%m-%d %H:%M:%S"),12,13))  
df$period<-1  
df$period[df$hour>7]<-2  
df$period[df$hour>10]<-3  
df$period[df$hour>16]<-4  
df$period[df$hour>20]<-1  
df$period<-factor(df$period,labels=paste("Period",c("night","morning","valley","afternoon")))  
barplot(summary(df$period),main="period Barplot",col = "DarkSlateBlue")



### 1. 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.

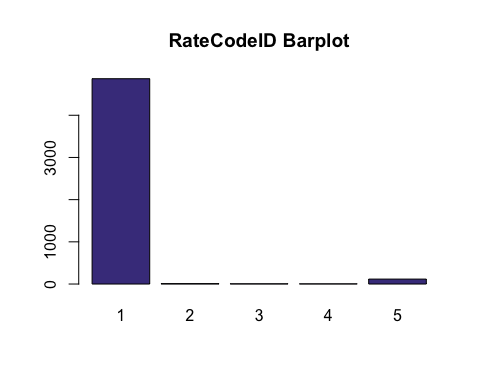
df$VendorID<-factor(df$VendorID,labels=c("Mobile","VeriFone"))  
levels(df$VendorID)<-paste0("f.Vendor-",levels(df$VendorID))  
barplot(summary(df$VendorID),main="VendorID Barplot",col = "DarkSlateBlue")



### 8. RateCodeID

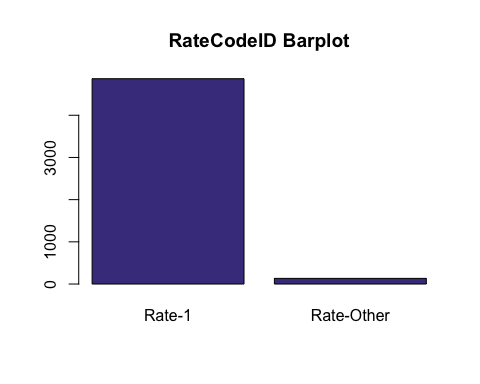
This variable expresses the different RateCodeIDs that we can have as numerical values, so we need to categorize them in order to be able to work with them.

df$RateCodeID<-factor(df$RateCodeID)  
barplot(summary(df$RateCodeID),main="RateCodeID 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$RateCodeID[df$RateCodeID != 1] = 2  
df$RateCodeID <- factor(df$RateCodeID, labels =c("Rate-1","Rate-Other"))  
barplot(summary(df$RateCodeID),main="RateCodeID Barplot",col = "DarkSlateBlue")

 Now is more balanced.

### 9. Store\_and\_fwd\_flag

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

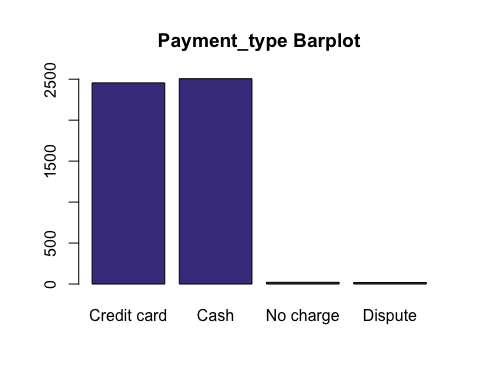
df$Store\_and\_fwd\_flag<-factor(df$Store\_and\_fwd\_flag)  
barplot(summary(df$Store\_and\_fwd\_flag),main="Store\_and\_fwd\_flag Barplot",col = "DarkSlateBlue")



### 12. Payment\_type

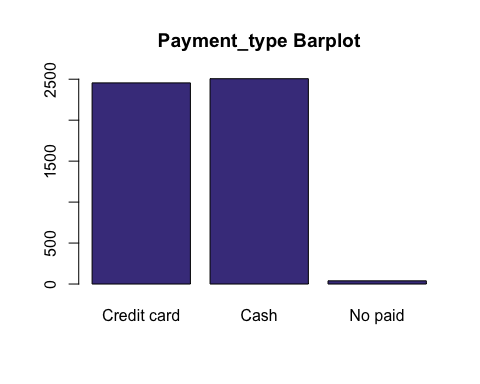
This variable is categorical but it is expressed as numerical, so we need to factor it in order to be able to work with it.

df$Payment\_type<-factor(df$Payment\_type,labels=c("Credit card","Cash","No charge","Dispute"))  
barplot(summary(df$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”).

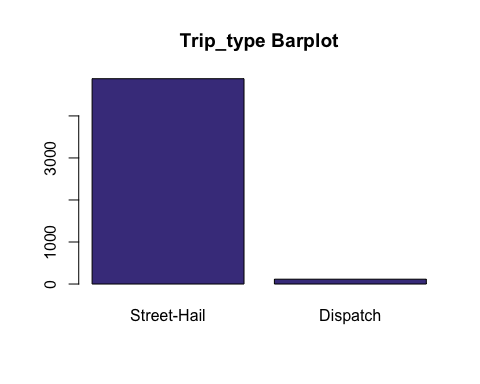
levels(df$Payment\_type) <- c("Credit card","Cash","No paid","No paid")  
barplot(summary(df$Payment\_type),main="Payment\_type Barplot",col = "DarkSlateBlue")



### 21. 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.

df$Trip\_type<-factor(df$Trip\_type,labels=c("Street-Hail","Dispatch"))  
barplot(summary(df$Trip\_type),main="Trip\_type Barplot",col = "DarkSlateBlue")



## 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)

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

#### Trip length in km

df$tlenkm<-df$Trip\_distance\*1.609344 # Miles to km

#### Travel time in min

df$traveltime<-(as.numeric(as.POSIXct(df$Lpep\_dropoff\_datetime)) - as.numeric(as.POSIXct(df$lpep\_pickup\_datetime)))/60

#### Effective speed in km/h

df$espeed<-(df$tlenkm/(df$traveltime))\*60

##### Missing data

sel<-which(is.na(df$espeed<=0)) #;length(sel)  
imis[sel]<-imis[sel]+1  
jmis[26]<-length(sel)

##### Error detection

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

summary(df$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$espeed<=0)|(df$espeed > 200))  
ierrs[sel]<-ierrs[sel]+1  
jerrs[26]<-length(sel)

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

df[sel,"espeed"]<-NA

##### Check outliers

calcQ(df$espeed)

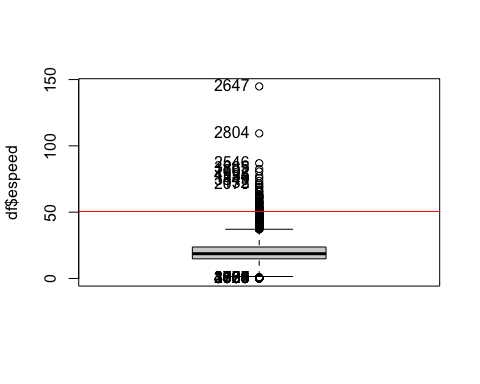
## $souti  
## 1st Qu.   
## -12.00637   
##   
## $mouti  
## 1st Qu.   
## 1.394097   
##   
## $min  
## Min.   
## 0.03530885   
##   
## $q1  
## 1st Qu.   
## 14.79457   
##   
## $q2  
## Median   
## 18.65269   
##   
## $q3  
## 3rd Qu.   
## 23.72821   
##   
## $max  
## Max.   
## 144.841   
##   
## $mouts  
## 3rd Qu.   
## 37.12868   
##   
## $souts  
## 3rd Qu.   
## 50.52915

##### Outlier detection

Boxplot(df$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$espeed)  
abline(h=var\_out$souts,col="red")  
abline(h=var\_out$souti,col="red")



llout<-which((df$espeed<=3)|(df$espeed>80))  
iouts[llout]<-iouts[llout]+1  
jouts[26]<-length(llout)  
df[llout,"espeed"]<-NA

### 2. lpep\_pickup\_datetime

We just keep the hours

df$pickup<-substr(strptime(df$lpep\_pickup\_datetime, "%Y-%m-%d %H:%M:%S"), 12, 13)

### 3. lpep\_dropoff\_datetime

We just keep the hours

df$dropoff<-substr(strptime(df$Lpep\_dropoff\_datetime, "%Y-%m-%d %H:%M:%S"), 12, 13)

### 4. Passenger\_count

summary(df$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$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,"Passenger\_count"]<-NA

### 5. Trip\_distance

summary(df$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.

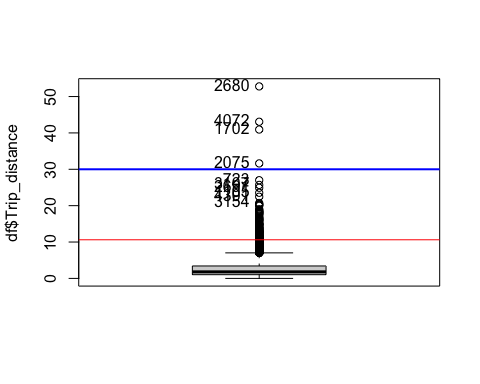
#### Outlier detection

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$Trip\_distance)

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

var\_out<-calcQ(df$Trip\_distance)  
abline(h=var\_out$souts,col="red")  
abline(h=var\_out$souti,col="red")  
abline(h=30,col="blue",lwd=2)



llout<-which(df$Trip\_distance>30)  
iouts[llout]<-iouts[llout]+1  
jouts[11]<-length(llout)

#### 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$Trip\_distance <= 0)  
ierrs[sel]<-ierrs[sel]+1  
jerrs[11]<-length(sel)

#### Errors and outliers

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

setNA<-which((df$Trip\_distance<=0) | (df$Trip\_distance > 30))  
df[setNA,"Trip\_distance"]<-NA

#### Caterogial variable for Trip\_distance

We are going to set a categorical variable for the Trip\_distancerange.

We decided to create 3 levels: “Short\_dist”, “Medium\_dist” and“Long\_dist”.

* Short\_dist <= 2.5
* Medium\_dist 2.5 < Trip\_distance <= 5
* Long\_dist > 5

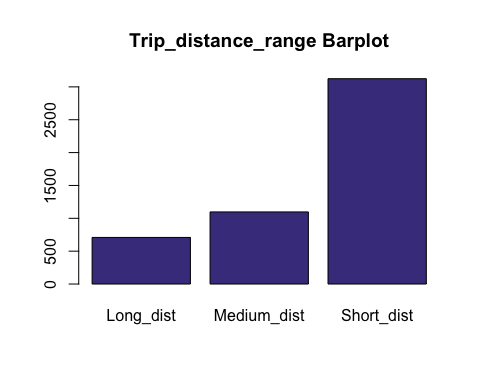
df$Trip\_distance\_range[df$Trip\_distance <= 2.5] = "Short\_dist"  
df$Trip\_distance\_range[(df$Trip\_distance > 2.5) & (df$Trip\_distance <= 5)] = "Medium\_dist"  
df$Trip\_distance\_range[df$Trip\_distance > 5] = "Long\_dist"

We see, though, that it is not a factor yet, so we factor it.

df$Trip\_distance\_range <- factor(df$Trip\_distance\_range)

We see a barplot for the factor we created.

barplot(table(df$Trip\_distance\_range),main="Trip\_distance\_range Barplot",col = "DarkSlateBlue")



### 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$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[,“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$Pickup\_longitude == 0)  
ierrs[sel]<-ierrs[sel]+1  
jerrs[6]<-length(sel)

Sel contains the rownames of the individuals with “0” as value for longitude.

df[sel,"Pickup\_longitude"]<-NA

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

#### Which trips are not running in New-York?

Consider if, at least, one of the pick-up and drop-off points belong to New-York area. if not, this trip is an “out-of-scope” individual and has to be eliminated of the basis. Nevertheless, you have to justify thiselimination and count how many individuals were in this situation. Look at that!! possibly, starting from the outliers…“0” is missing value, outliers can help to detect trips running outside of New York…

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$Pickup\_longitude < -74.02) | (df$Pickup\_longitude > -73.80))  
iouts[llout]<-iouts[llout]+1  
jouts[6]<-length(llout)

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

df[llout,"Pickup\_longitude"]<-NA

### 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$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[,“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$Pickup\_latitude == 0)  
ierrs[sel]<-ierrs[sel]+1  
jerrs[7]<-length(sel)

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

df[sel,"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$Pickup\_latitude < 40.54) | (df$Pickup\_latitude > 40.86))  
iouts[llout]<-iouts[llout]+1  
jouts[7]<-length(llout)

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

df[llout,"Pickup\_latitude"]<-NA

### 10. 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$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[,“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$Dropoff\_longitude == 0)  
ierrs[sel]<-ierrs[sel]+1  
jerrs[8]<-length(sel)

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

df[sel,"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$Dropoff\_longitude < -74.02) | (df$Dropoff\_longitude > -73.80))  
iouts[llout]<-iouts[llout]+1  
jouts[8]<-length(llout)

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

df[llout,"Dropoff\_longitude"]<-NA

### 11. 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$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[,“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$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,"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$Dropoff\_latitude < 40.54) | (df$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,"Dropoff\_latitude"]<-NA

### 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$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$Fare\_amount <= 0)  
ierrs[sel]<-ierrs[sel]+1  
jerrs[12]<-length(sel)  
df[sel,"Fare\_amount"]<-NA

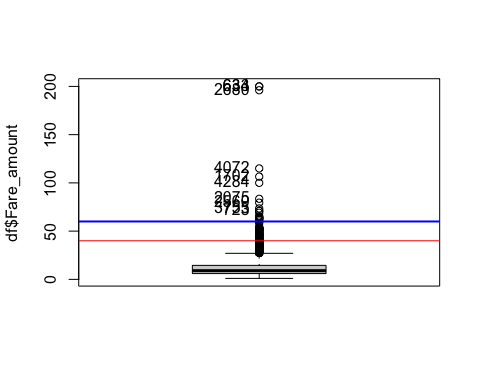
Non-possible values are replaced by NA, missing value symbol in R

#### Outlier detection

Boxplot(df$Fare\_amount)

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

var\_out<-calcQ(df$Fare\_amount)  
abline(h=var\_out$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$Fare\_amount>60)  
iouts[llout]<-iouts[llout]+1  
jouts[12]<-length(llout)  
df[llout,"Fare\_amount"]<-NA

### 14. Extra

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

summary(df$Extra)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.0000 0.0000 0.5000 0.3517 0.5000 1.0000

We execute table in order to see every different value in the sample

table(df$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$Extra < 0)  
ierrs[sel]<-ierrs[sel]+1  
jerrs[13]<-length(sel)  
df[sel,"Extra"]<-NA

### 15. 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.

summary(df$MTA\_tax)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.5000 0.5000 0.5000 0.4857 0.5000 0.5000

**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$MTA\_tax < 0)  
ierrs[sel]<-ierrs[sel]+1  
jerrs[14]<-length(sel)  
df[sel,"MTA\_tax"]<-NA

### 16. 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.

summary(df$improvement\_surcharge)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.3000 0.3000 0.3000 0.2914 0.3000 0.3000

table(df$improvement\_surcharge)

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

We know that this surcharge was leived in 2015, so we need to check if the 0 values correspond to trips before this year. That is what we are going to do.

df$yearGt2015[(df$lpep\_pickup\_datetime >= "2015-01-01 00:00:00") & (df$improvement\_surcharge == 0.3)] = 1  
df$yearGt2015[(df$lpep\_pickup\_datetime < "2015-01-01 00:00:00") | (df$improvement\_surcharge != 0.3)] = 0  
  
table(df$yearGt2015)

##   
## 0 1   
## 132 4868

We see that the 0 individuals are errors.

sel<-which(df$improvement\_surcharge < 0)  
ierrs[sel]<-ierrs[sel]+1  
jerrs[18]<-length(sel)  
df[sel,"improvement\_surcharge"]<-NA

### 17. Ehail\_fee

We don’t take this into account because every value of our sample is NA.

summary(df$Ehail\_fee)

## Mode NA's   
## logical 5000

### 18. 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$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.

df$CashTips[(df$Tip\_amount > 0) & (df$Payment\_type == "Cash")] = 1  
df$CashTips[(df$Payment\_type == "Credit card")] = 0  
table(df$CashTips)

##   
## 0   
## 2455

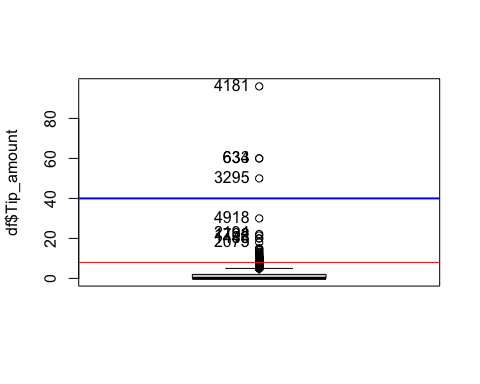
Now, we proceed to the outlier detection.

#### Outlier detection

Boxplot(df$Tip\_amount)

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

var\_out<-calcQ(df$Tip\_amount)  
abline(h=var\_out$souts,col="red")  
abline(h=var\_out$souti,col="red")  
abline(h=40,col="blue",lwd=2)



llout<-which(df$Tip\_amount>40)  
iouts[llout]<-iouts[llout]+1  
jouts[15]<-length(llout)  
df[llout,"Tip\_amount"]<-NA

### 19. Tolls\_amount

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

summary(df$Tolls\_amount)

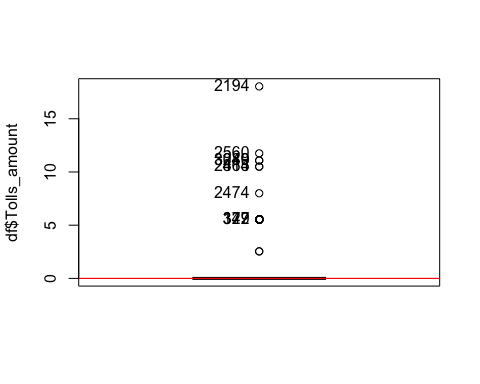
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00000 0.00000 0.00000 0.08369 0.00000 18.04000

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

Boxplot(df$Tolls\_amount)

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

var\_out<-calcQ(df$Tolls\_amount)  
abline(h=var\_out$souts,col="red")  
abline(h=var\_out$souti,col="red")



table(df$Tolls\_amount)

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

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$Tolls\_amount>5.54)  
iouts[llout]<-iouts[llout]+1  
jouts[16]<-length(llout)  
df[llout,"Tolls\_amount"]<-NA

### 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$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:

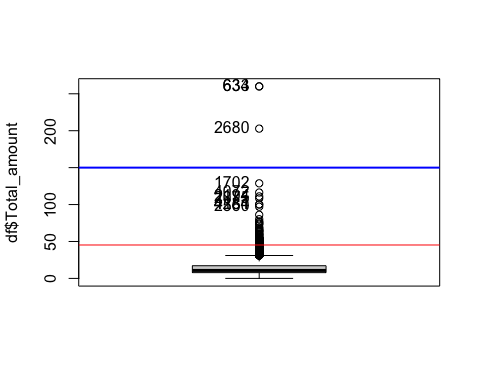
df$Sum\_total\_amount = (df$Fare\_amount + df$Extra + df$MTA\_tax + df$improvement\_surcharge + df$Tip\_amount + df$Tolls\_amount)  
  
sel<-which((df$Total\_amount != df$Sum\_total\_amount) | (df$Total\_amount<0))  
if (length(sel)>0) {  
 ierrs[sel]<-ierrs[sel]+1  
 jerrs[19]<-length(sel)  
}  
df[sel,"Total\_amount"]<-NA

#### Outlier detection

Boxplot(df$Total\_amount)

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

var\_out<-calcQ(df$Total\_amount)  
abline(h=var\_out$souts,col="red")  
abline(h=var\_out$souti,col="red")  
abline(h=150,col="blue",lwd=2)



llout<-which(df$Total\_amount>150)  
iouts[llout]<-iouts[llout]+1  
jouts[19]<-length(llout)  
df[llout,"Total\_amount"]<-NA

# Data Quality Report

## 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).

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

missings\_ranking\_sortlist <- sort.list(mis1$mis\_col, decreasing = TRUE)  
for (j in missings\_ranking\_sortlist) {  
 print(paste(names(df)[j], " : ", mis1$mis\_col$mis\_x[j]))  
}

## [1] "Ehail\_fee : 5000"  
## [1] "VendorID : 0"  
## [1] "lpep\_pickup\_datetime : 0"  
## [1] "Lpep\_dropoff\_datetime : 0"  
## [1] "Store\_and\_fwd\_flag : 0"  
## [1] "RateCodeID : 0"  
## [1] "Pickup\_longitude : 0"  
## [1] "Pickup\_latitude : 0"  
## [1] "Dropoff\_longitude : 0"  
## [1] "Dropoff\_latitude : 0"  
## [1] "Passenger\_count : 0"  
## [1] "Trip\_distance : 0"  
## [1] "Fare\_amount : 0"  
## [1] "Extra : 0"  
## [1] "MTA\_tax : 0"  
## [1] "Tip\_amount : 0"  
## [1] "Tolls\_amount : 0"  
## [1] "improvement\_surcharge : 0"  
## [1] "Total\_amount : 0"  
## [1] "Payment\_type : 0"  
## [1] "Trip\_type : 0"

### Number of errors per each variable (with ranking)

errors\_ranking\_sortlist <- sort.list(jerrs, decreasing = TRUE)  
for (j in errors\_ranking\_sortlist) {  
 if(!is.na(names(df)[j])) { print(paste(names(df)[j], " : ", jerrs[j])) }  
}

## [1] "Total\_amount : 374"  
## [1] "espeed : 73"  
## [1] "Trip\_distance : 66"  
## [1] "Fare\_amount : 24"  
## [1] "improvement\_surcharge : 11"  
## [1] "MTA\_tax : 10"  
## [1] "Dropoff\_longitude : 9"  
## [1] "Dropoff\_latitude : 9"  
## [1] "Extra : 7"  
## [1] "Pickup\_longitude : 3"  
## [1] "Pickup\_latitude : 3"  
## [1] "Passenger\_count : 2"  
## [1] "VendorID : 0"  
## [1] "lpep\_pickup\_datetime : 0"  
## [1] "Lpep\_dropoff\_datetime : 0"  
## [1] "Store\_and\_fwd\_flag : 0"  
## [1] "RateCodeID : 0"  
## [1] "Tip\_amount : 0"  
## [1] "Tolls\_amount : 0"  
## [1] "Ehail\_fee : 0"  
## [1] "Payment\_type : 0"  
## [1] "Trip\_type : 0"  
## [1] "hour : 0"  
## [1] "period : 0"  
## [1] "tlenkm : 0"  
## [1] "traveltime : 0"  
## [1] "pickup : 0"  
## [1] "dropoff : 0"  
## [1] "Trip\_distance\_range : 0"  
## [1] "yearGt2015 : 0"  
## [1] "CashTips : 0"  
## [1] "Sum\_total\_amount : 0"

### Number of outliers per each variable (with ranking)

errors\_ranking\_sortlist <- sort.list(jouts, decreasing = TRUE)  
for (j in errors\_ranking\_sortlist) {  
 if(!is.na(names(df)[j])) print(paste(names(df)[j], " : ", jouts[j]))  
}

## [1] "Dropoff\_latitude : 116"  
## [1] "Dropoff\_longitude : 113"  
## [1] "Pickup\_latitude : 84"  
## [1] "espeed : 39"  
## [1] "Fare\_amount : 20"  
## [1] "Pickup\_longitude : 19"  
## [1] "Tolls\_amount : 7"  
## [1] "Trip\_distance : 4"  
## [1] "Tip\_amount : 4"  
## [1] "Total\_amount : 3"  
## [1] "VendorID : 0"  
## [1] "lpep\_pickup\_datetime : 0"  
## [1] "Lpep\_dropoff\_datetime : 0"  
## [1] "Store\_and\_fwd\_flag : 0"  
## [1] "RateCodeID : 0"  
## [1] "Passenger\_count : 0"  
## [1] "Extra : 0"  
## [1] "MTA\_tax : 0"  
## [1] "Ehail\_fee : 0"  
## [1] "improvement\_surcharge : 0"  
## [1] "Payment\_type : 0"  
## [1] "Trip\_type : 0"  
## [1] "hour : 0"  
## [1] "period : 0"  
## [1] "tlenkm : 0"  
## [1] "traveltime : 0"  
## [1] "pickup : 0"  
## [1] "dropoff : 0"  
## [1] "Trip\_distance\_range : 0"  
## [1] "yearGt2015 : 0"  
## [1] "CashTips : 0"  
## [1] "Sum\_total\_amount : 0"

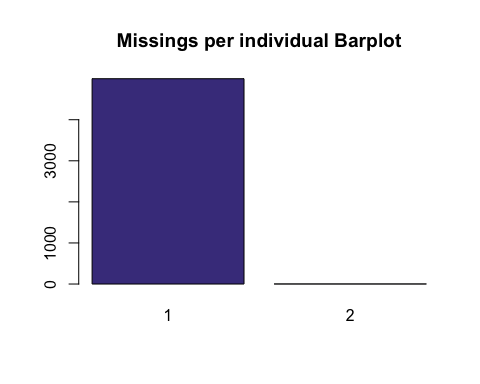
## Per individual

Per each individuals, we have to count the following:

* number of missing values
* number of errors
* number of outliers

### Number of missing values

barplot(table(imis),main="Missings per individual Barplot",col = "DarkSlateBlue")

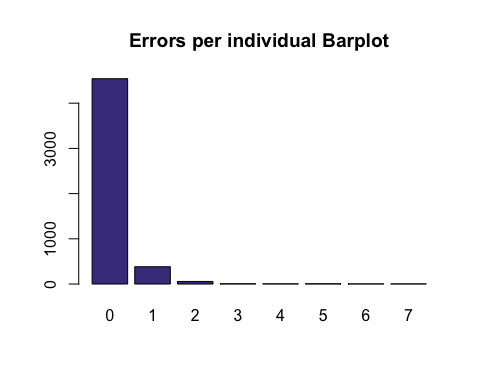


The one is from from the variable “Ehail\_fee” and the observations that have two missing values are because of the “espeed” variable (maybe because the traveltime was 0 and nothing can be divided by 0).

### Number of errors

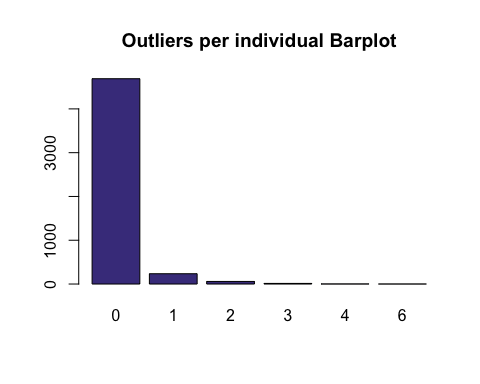
As we can see, most individuals have no mistakes. Those who do have errors, they tend to have more than one.

barplot(table(ierrs),main="Errors per individual Barplot",col = "DarkSlateBlue")



### Number of outliers

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



## 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] 5002

total\_outliers

## [1] 409

total\_errors

## [1] 591

# 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.

## 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$Total\_amount),]  
vars\_quantitatives<-names(df)[c(10:13,15,16,24:26)]

summary(df[,vars\_quantitatives])

## Passenger\_count Trip\_distance Fare\_amount 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   
## Tip\_amount Tolls\_amount tlenkm traveltime   
## Min. : 0.000 Min. :0.00000 Min. : 0.000 Min. : 0.000   
## 1st Qu.: 0.000 1st Qu.:0.00000 1st Qu.: 1.609 1st Qu.: 5.767   
## Median : 0.000 Median :0.00000 Median : 2.800 Median : 9.550   
## Mean : 1.029 Mean :0.04671 Mean : 4.358 Mean : 19.863   
## 3rd Qu.: 1.700 3rd Qu.:0.00000 3rd Qu.: 5.472 3rd Qu.: 16.125   
## Max. :30.000 Max. :5.54000 Max. :69.314 Max. :1438.183   
## NA's :2 NA's :7   
## espeed   
## Min. : 3.239   
## 1st Qu.:14.826   
## Median :18.613   
## Mean :20.490   
## 3rd Qu.:23.647   
## Max. :75.657   
## NA's :105

res.imputation<-imputePCA(df[,vars\_quantitatives],ncp=5)  
summary(res.imputation$completeObs)

## Passenger\_count Trip\_distance Fare\_amount Extra   
## Min. :1.000 Min. : 0.010 Min. : 1.00 Min. :0.0000   
## 1st Qu.:1.000 1st Qu.: 1.010 1st Qu.: 6.00 1st Qu.:0.0000   
## Median :1.000 Median : 1.760 Median : 9.00 Median :0.5000   
## Mean :1.371 Mean : 2.726 Mean : 11.64 Mean :0.3523   
## 3rd Qu.:1.000 3rd Qu.: 3.400 3rd Qu.: 14.50 3rd Qu.:0.5000   
## Max. :6.000 Max. :35.939 Max. :108.01 Max. :1.0000   
## Tip\_amount Tolls\_amount tlenkm traveltime   
## Min. : 0.000 Min. :-0.4860 Min. : 0.000 Min. : 0.000   
## 1st Qu.: 0.000 1st Qu.: 0.0000 1st Qu.: 1.609 1st Qu.: 5.767   
## Median : 0.000 Median : 0.0000 Median : 2.800 Median : 9.550   
## Mean : 1.028 Mean : 0.0474 Mean : 4.358 Mean : 19.863   
## 3rd Qu.: 1.700 3rd Qu.: 0.0000 3rd Qu.: 5.472 3rd Qu.: 16.125   
## Max. :30.000 Max. : 5.5400 Max. :69.314 Max. :1438.183   
## espeed   
## Min. :-84.61   
## 1st Qu.: 14.83   
## Median : 18.56   
## Mean : 19.97   
## 3rd Qu.: 23.58   
## Max. : 88.07

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

#### > Trip\_distance

ll<-which(res.imputation$completeObs[,"Trip\_distance"] < 0)  
res.imputation$completeObs[ll,"Trip\_distance"] <- 1  
ll<-which(res.imputation$completeObs[,"Trip\_distance"] > 30)  
res.imputation$completeObs[ll,"Trip\_distance"] <- 30

#### > Fare\_amount

ll<-which(res.imputation$completeObs[,"Fare\_amount"] > 60)  
res.imputation$completeObs[ll,"Fare\_amount"] <- 60

#### > Tip\_amount

ll<-which(res.imputation$completeObs[,"Tip\_amount"] > 17)  
res.imputation$completeObs[ll,"Tip\_amount"] <- 17

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

df$TipIsGiven[(res.imputation$completeObs[,"Tip\_amount"] > 0)] = "Yes"  
df$TipIsGiven[(res.imputation$completeObs[,"Tip\_amount"] == 0)] = "No"  
df$TipIsGiven <- factor(df$TipIsGiven)  
summary(df$TipIsGiven)

## No Yes   
## 2882 1741

#### > tlenkm

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

#### > traveltime

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

#### > espeed

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

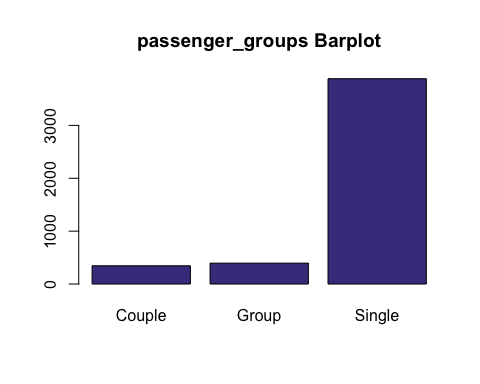
### > Passenger\_count

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

df$passenger\_groups[res.imputation$completeObs[,"Passenger\_count"] == 1] = "Single"  
df$passenger\_groups[res.imputation$completeObs[,"Passenger\_count"] > 1 & res.imputation$completeObs[,"Passenger\_count"] <= 2] = "Couple"  
df$passenger\_groups[res.imputation$completeObs[,"Passenger\_count"] >= 3] = "Group"  
df$passenger\_groups <- factor(df$passenger\_groups)

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

barplot(table(df$passenger\_groups),main="passenger\_groups Barplot",col = "DarkSlateBlue")



### > 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.

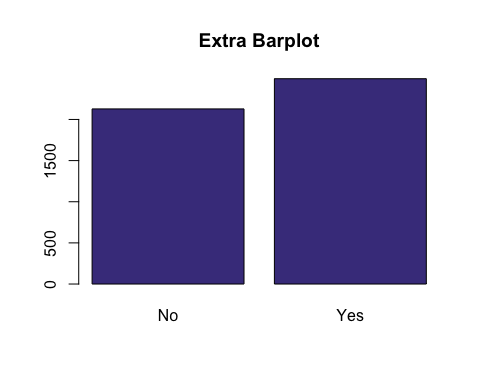
table(df$Extra)

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

df$Extra[df$Extra == 0] = 0  
df$Extra[df$Extra > 0] = 1  
df$Extra <- factor(df$Extra, labels =c("No","Yes"))

We see the barplot in order to see the distribution.

barplot(table(df$Extra),main="Extra Barplot",col = "DarkSlateBlue")



### > 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.

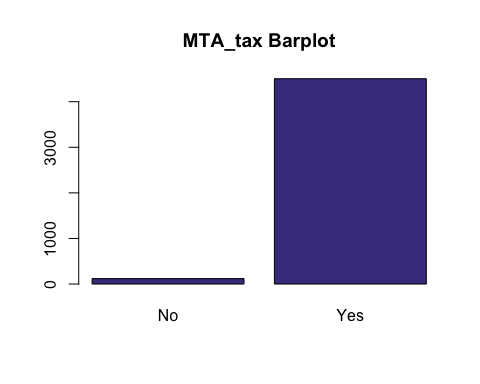
summary(df$MTA\_tax)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.5000 0.5000 0.4871 0.5000 0.5000

df$MTA\_tax <- factor(df$MTA\_tax, labels =c("No","Yes"))

We see the barplot in order to see the distribution.

barplot(table(df$MTA\_tax),main="MTA\_tax Barplot",col = "DarkSlateBlue")



### > 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.

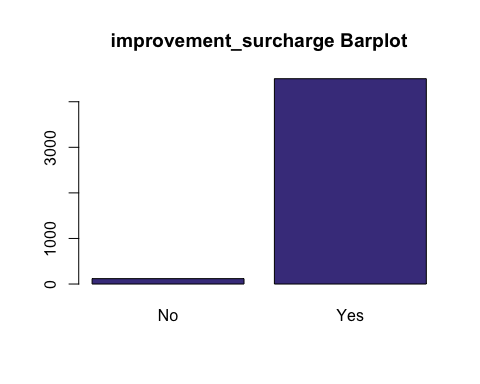
table(df$improvement\_surcharge)

##   
## 0 0.3   
## 118 4505

df$improvement\_surcharge <- factor(df$improvement\_surcharge)  
df$improvement\_surcharge <- factor(df$improvement\_surcharge, labels =c("No","Yes"))

We see the barplot in order to see the distribution.

barplot(table(df$improvement\_surcharge),main="improvement\_surcharge Barplot",col = "DarkSlateBlue")



### > 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$completeObs[,"Tolls\_amount"] < 0)  
res.imputation$completeObs[ll,"Tolls\_amount"] <- 0  
  
df$paidTolls[res.imputation$completeObs[,"Tolls\_amount"] == 0] = "No"  
df$paidTolls[res.imputation$completeObs[,"Tolls\_amount"] > 0] = "Yes"  
df$paidTolls <- factor(df$paidTolls)

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

df[,vars\_quantitatives] <- res.imputation$completeObs

## Categorical variables / Factors

vars\_categorical<-names(df)[c(1,4,5,20:21,23,29,35)]  
 summary(df[,vars\_categorical])

## VendorID Store\_and\_fwd\_flag RateCodeID   
## f.Vendor-Mobile : 973 N:4605 Rate-1 :4496   
## f.Vendor-VeriFone:3650 Y: 18 Rate-Other: 127   
##   
##   
## Payment\_type Trip\_type period   
## Credit card:2096 Street-Hail:4511 Period night :1642   
## Cash :2497 Dispatch : 112 Period morning : 542   
## No paid : 30 Period valley :1260   
## Period afternoon:1179   
## Trip\_distance\_range paidTolls   
## Long\_dist : 645 No :4580   
## Medium\_dist: 986 Yes: 43   
## Short\_dist :2930   
## NA's : 62

res.input<-imputeMCA(df[,vars\_categorical],ncp=10)  
 summary(res.input$completeObs)

## VendorID Store\_and\_fwd\_flag RateCodeID   
## f.Vendor-Mobile : 973 N:4605 Rate-1 :4496   
## f.Vendor-VeriFone:3650 Y: 18 Rate-Other: 127   
##   
##   
## Payment\_type Trip\_type period   
## Credit card:2096 Street-Hail:4511 Period night :1642   
## Cash :2497 Dispatch : 112 Period morning : 542   
## No paid : 30 Period valley :1260   
## Period afternoon:1179   
## Trip\_distance\_range paidTolls   
## Long\_dist : 666 No :4580   
## Medium\_dist: 986 Yes: 43   
## Short\_dist :2971   
##

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

df[,"VendorID"] <- res.input$completeObs[,"VendorID"]  
df[,"Store\_and\_fwd\_flag"] <- res.input$completeObs[,"Store\_and\_fwd\_flag"]  
df[,"RateCodeID"] <- res.input$completeObs[,"RateCodeID"]  
df[,"Payment\_type"] <- res.input$completeObs[,"Payment\_type"]  
df[,"Trip\_type"] <- res.input$completeObs[,"Trip\_type"]  
df[,"period"] <- res.input$completeObs[,"period"]  
df[,"Trip\_distance\_range"] <- res.input$completeObs[,"Trip\_distance\_range"]  
# df[,"passenger\_groups"] <- res.input$completeObs[,"passenger\_groups"]

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

### Compute the correlation with all other variables.

library(mvoutlier)  
library(FactoMineR)  
res <- cor(df[,vars\_quantitatives])  
round(res, 2)

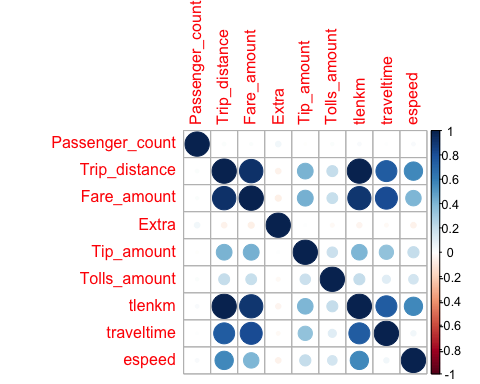
## Passenger\_count Trip\_distance Fare\_amount Extra Tip\_amount  
## Passenger\_count 1.00 0.02 0.01 0.05 -0.01  
## Trip\_distance 0.02 1.00 0.92 -0.05 0.42  
## Fare\_amount 0.01 0.92 1.00 -0.06 0.42  
## Extra 0.05 -0.05 -0.06 1.00 0.01  
## Tip\_amount -0.01 0.42 0.42 0.01 1.00  
## Tolls\_amount 0.02 0.20 0.20 -0.03 0.19  
## tlenkm 0.02 1.00 0.91 -0.05 0.41  
## traveltime 0.01 0.74 0.82 -0.02 0.35  
## espeed 0.02 0.57 0.40 -0.05 0.21  
## Tolls\_amount tlenkm traveltime espeed  
## Passenger\_count 0.02 0.02 0.01 0.02  
## Trip\_distance 0.20 1.00 0.74 0.57  
## Fare\_amount 0.20 0.91 0.82 0.40  
## Extra -0.03 -0.05 -0.02 -0.05  
## Tip\_amount 0.19 0.41 0.35 0.21  
## Tolls\_amount 1.00 0.20 0.11 0.15  
## tlenkm 0.20 1.00 0.74 0.56  
## traveltime 0.11 0.74 1.00 0.05  
## espeed 0.15 0.56 0.05 1.00

### Rank these variables according the correlation:

library(corrplot)

## corrplot 0.84 loaded

corrplot(res)



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.

In case of not having made the imputation of espeed and traveltime, we would have the following plot:

[insert image],

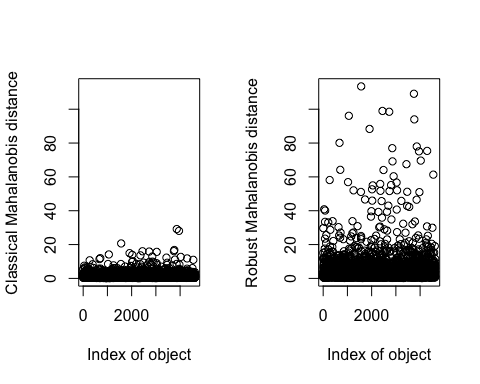
which means that there is a negative correlation between these two variables, since the longer the time, the slower the trip. However, we think it is necessary to remove the outliers we have had from these variables because they are unrealistic.

Now, let’s describe each correlation we obtained in the first graph:

* Diagonals:
  + Being exactly the same variable, it is directly related to itself.
* Fare\_amount + Trip\_distance:
  + More distance, more time, therefore more price.
* Tip\_amount + Trip\_distance:
  + If the trip has been longer, there may be more reason to tip.
* Total\_amount + Trip\_distance:
  + As before, more distance, more time, therefore more price.
* tlenkm + Trip\_distance:
  + They are exactly the same, only with a metric change.
* traveltime + Trip\_distance:
  + The further away, the longer.
* espeed + Trip\_distance:
  + 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.
* Amount\_type + Amount\_mount:
  + In the USA it is normal to give a tip proportional to the price of the service that has been offered.
* Total\_amount + Fare\_amount:
  + The variable Total\_amount is equivalent to Fare\_amount plus the fees, tips, among others, that have been applied to the trip.
* tlenkm + Fare\_amount:
  + As before, more distance, more time, therefore more price.
* traveltime + Fare\_amount:
  + More time, more price.
* espeed + Fare\_amount:
  + As we said before, more speed means more distance, therefore more travel time, causing more price.
* Total\_amount + Type\_amount:
  + As before, in the USA it is normal to give a tip proportional to the price of the service that has been offered.
* tlenkm + Mount\_type:
  + If the trip has been longer, there may be more reason to tip.
* traveltime + Tip\_amount:
  + The longer it takes, the more price, and therefore the more tip given the proportionality.
* espeed + Tip\_amount:
  + The more speed, as we said before, the more distance, and therefore the longer it takes. This causes more price and therefore more tip.
* tlenkm + Total\_amount:
  + More distance, more time, therefore more price.
* traveltime + Total\_amount:
  + More time, more price.
* espeed + Total\_amount:
  + As we said before, more speed means more distance, therefore more travel time, causing more price.
* traveltime + tlenkm:
  + The more km to travel, the longer it takes.
* speed + tlenkm:
  + Same as for espeed + Trip\_distance correlation.

### Identify individuals considered as multivariant outliers

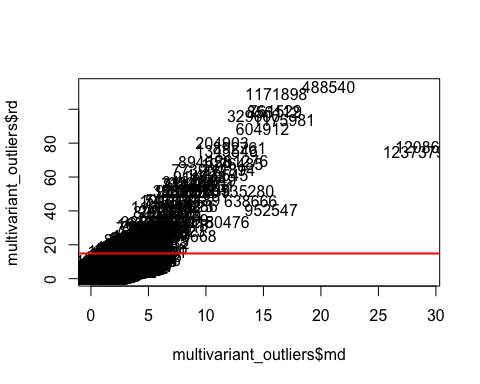
library(chemometrics)  
multivariant\_outliers <- Moutlier(df[, c(11:12, 19, 26)], quantile = 0.995)



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(11:12, 19, 26)]))   
abline(col="red",lwd=2, h=qchisq(0.995, ncol(df[, c(11:12, 19, 26)])))



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.

We want to look at two observations that have caught our attention. The first is 488540 and the second is 1180476.

As we can see, observation 488540 is the one at the boundary of the two axes. So that means it’s most likely a multivariate outlier. On the other hand, the 1180476 is not exactly very central on both axes. This may lead us to think that he is not as likely as the other observation to be a multivariate outlier.

df[which(row.names(df)=="488540"), 1:35]

## VendorID lpep\_pickup\_datetime Lpep\_dropoff\_datetime  
## 488540 f.Vendor-VeriFone 2016-01-11 06:57:31 2016-01-11 07:46:31  
## Store\_and\_fwd\_flag RateCodeID Pickup\_longitude Pickup\_latitude  
## 488540 N Rate-1 -73.91121 NA  
## Dropoff\_longitude Dropoff\_latitude Passenger\_count Trip\_distance  
## 488540 NA NA 1 30  
## Fare\_amount Extra MTA\_tax Tip\_amount Tolls\_amount Ehail\_fee  
## 488540 60 0 Yes 17 0 NA  
## improvement\_surcharge Total\_amount Payment\_type Trip\_type hour  
## 488540 Yes 128.76 Credit card Street-Hail 6  
## period tlenkm traveltime espeed pickup dropoff Trip\_distance\_range  
## 488540 Period night 48.28 49 55 06 07 Short\_dist  
## yearGt2015 CashTips Sum\_total\_amount TipIsGiven passenger\_groups  
## 488540 1 0 NA Yes Single  
## paidTolls  
## 488540 No

df[which(row.names(df)=="1180476"), 1:35]

## VendorID lpep\_pickup\_datetime Lpep\_dropoff\_datetime  
## 1180476 f.Vendor-Mobile 2016-01-27 05:48:11 2016-01-27 06:19:48  
## Store\_and\_fwd\_flag RateCodeID Pickup\_longitude Pickup\_latitude  
## 1180476 N Rate-Other -73.90414 40.85212  
## Dropoff\_longitude Dropoff\_latitude Passenger\_count Trip\_distance  
## 1180476 -73.9847 40.75537 1 10.5  
## Fare\_amount Extra MTA\_tax Tip\_amount Tolls\_amount Ehail\_fee  
## 1180476 28.76075 0 No 0 0 NA  
## improvement\_surcharge Total\_amount Payment\_type Trip\_type hour  
## 1180476 No 0 Cash Dispatch 5  
## period tlenkm traveltime espeed pickup dropoff  
## 1180476 Period night 16.89811 31.61667 32.06811 05 06  
## Trip\_distance\_range yearGt2015 CashTips Sum\_total\_amount TipIsGiven  
## 1180476 Long\_dist 0 NA NA No  
## passenger\_groups paidTolls  
## 1180476 Single No

# Profiling

## Numeric 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$Total\_amount)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 7.80 10.80 13.93 17.00 128.76

vars\_res<-names(df)[c(19,34)]  
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.

#### > Numerical variables

res.condes$quanti # Global association to numeric variables

## correlation p.value  
## Fare\_amount 0.9414826 0.000000e+00  
## Trip\_distance 0.8936243 0.000000e+00  
## tlenkm 0.8833291 0.000000e+00  
## traveltime 0.7620364 0.000000e+00  
## Tip\_amount 0.5665931 0.000000e+00  
## espeed 0.3962838 1.114335e-173  
## Tolls\_amount 0.2572094 9.464350e-71

* Fare\_amount:
  + The variable Total\_amount is equivalent to Fare\_amount plus the fees, tips, among others, that have been applied to the trip.
* Trip\_distance:
  + As before, more distance, more time, therefore more price.
* tlenkm
  + More distance, more time, therefore more price.
* traveltime
  + More time, more price.
* Tip\_amount
  + The more you pay, since the tip is a proportion of the final price, the more it will increase.
* espeed
  + As we said before, more speed means more distance, therefore more travel time, causing more price.

#### > Qualitative variables

res.condes$quali # Global association to factors

## R2 p.value  
## Trip\_distance\_range 0.563109948 0.000000e+00  
## paidTolls 0.079390158 4.123704e-85  
## Payment\_type 0.053488291 7.096416e-56  
## RateCodeID 0.013014975 7.244863e-15  
## Trip\_type 0.001221351 1.748826e-02

* Trip\_distance\_range
  + Obviously, the longer the journey, the longer it will take and the more price it will have.
* TipIsGiven
  + Like before, the more you pay, since the tip is a proportion of the final price, the more it will increase.
* Payment\_type
  + This is the least related variable. However, we can predict that the more the trip is worth, the more likely it is to be paid by credit card.
* RateCodeID
  + As we have seen before, virtually all observations were of type 1. Therefore it is not worth looking at the correlation.

#### > Categorical variables

res.condes$category # Partial association to significative levels in factors

## Estimate p.value  
## Trip\_distance\_range=Long\_dist 11.495819 0.000000e+00  
## paidTolls=Yes 14.741940 4.123704e-85  
## Payment\_type=Credit card 2.653121 4.728144e-57  
## RateCodeID=Rate-Other 3.505459 7.244863e-15  
## Trip\_type=Dispatch 1.141598 1.748826e-02  
## Trip\_type=Street-Hail -1.141598 1.748826e-02  
## RateCodeID=Rate-1 -3.505459 7.244863e-15  
## Trip\_distance\_range=Medium\_dist -1.632671 1.572556e-33  
## Payment\_type=Cash -2.024634 1.858977e-56  
## paidTolls=No -14.741940 4.123704e-85  
## Trip\_distance\_range=Short\_dist -9.863148 0.000000e+00

* Trip\_distance\_range
  + We can see that, the further away, the more correlation, as it takes longer to travel.
* TipIsGiven
  + We see that it is more likely to tip if the price is high.
* Payment\_type
  + We see that it is easier for the guy to be with CreditCard if the trip costs more.
* RateCodeID
  + As we have seen before, virtually all observations were of type 1. Therefore it is not worth looking at the correlation.
* period
  + We see that in the morning travel costs less.

## Factor (Y.bin - TipIsGiven)

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 TipIsGiven 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.

#### > Test.Chi2

res.catdes$test.chi2

## p.value df  
## Trip\_type 8.471361e-08 2  
## VendorID 1.105844e-07 2  
## RateCodeID 4.215654e-06 2  
## period 1.409652e-05 6

* Payment\_type
  + We see that it is very likely that there will be a tip if it is paid in a concise manner.
* Trip\_distance\_range
  + As we can see, there is tip as long as the trip is, or very short, or very long.
* Trip\_type
  + We don’t think the type of trip is important.
* RateCodeID
  + As we have seen before, virtually all observations were of type 1. Therefore it is not worth looking at the correlation.
* period
  + We see that in the morning people are not in a very good mood and are more inclined to tip the “valley”.

#### > Quantitative variables

res.catdes$quanti.var

## Eta2 P-value  
## Passenger\_count 0.901653419 0.000000000  
## Extra 0.002462239 0.003363457

* Tip\_amount
  + If there is a tip, it must have value.
* Total\_amount
  + We see that it is more likely to tip if the price is high.
* Fare\_amount
  + We see that it is more likely to tip if the price is high.
* tlenkm
  + The more distance, the more time, therefore the more price. So, more chances of there being a tip.
* Trip\_distance
  + Exactly the same as above.
* traveltime
  + The longer, therefore the more price. So, more chances of there being a tip.
* espeed
  + The faster you get to the site, the more satisfaction and therefore the likelihood of tipping.

#### > Categorical variables

res.catdes$category

## $Couple  
## Cla/Mod Mod/Cla Global p.value v.test  
## Trip\_type=Dispatch 21.428571 6.956522 2.422669 1.982364e-06 4.755214  
## RateCodeID=Rate-Other 18.897638 6.956522 2.747134 2.061983e-05 4.258071  
## VendorID=f.Vendor-Mobile 9.866393 27.826087 21.046939 1.842972e-03 3.114436  
## period=Period morning 4.981550 7.826087 11.723989 1.510117e-02 -2.429944  
## VendorID=f.Vendor-VeriFone 6.821918 72.173913 78.953061 1.842972e-03 -3.114436  
## RateCodeID=Rate-1 7.139680 93.043478 97.252866 2.061983e-05 -4.258071  
## Trip\_type=Street-Hail 7.115939 93.043478 97.577331 1.982364e-06 -4.755214  
##   
## $Group  
## Cla/Mod Mod/Cla Global p.value v.test  
## VendorID=f.Vendor-VeriFone 9.589041 88.607595 78.95306 1.756148e-07 5.223455  
## period=Period night 10.475030 43.544304 35.51806 5.820838e-04 3.439828  
## period=Period valley 6.746032 21.518987 27.25503 6.443873e-03 -2.724296  
## period=Period morning 4.981550 6.835443 11.72399 8.403776e-04 -3.339141  
## VendorID=f.Vendor-Mobile 4.624872 11.392405 21.04694 1.756148e-07 -5.223455  
##   
## $Single  
## Cla/Mod Mod/Cla Global p.value  
## period=Period morning 90.03690 12.5676024 11.7239888 1.840704e-05  
## Trip\_type=Street-Hail 84.28286 97.9139840 97.5773307 1.546175e-03  
## RateCodeID=Rate-1 84.27491 97.5791913 97.2528661 3.548442e-03  
## period=Period valley 86.19048 27.9680659 27.2550292 1.180517e-02  
## Payment\_type=No paid 96.66667 0.7468452 0.6489293 4.067769e-02  
## Trip\_distance\_range=Long\_dist 81.23123 13.9325264 14.4062297 3.853762e-02  
## RateCodeID=Rate-Other 74.01575 2.4208087 2.7471339 3.548442e-03  
## Trip\_type=Dispatch 72.32143 2.0860160 2.4226693 1.546175e-03  
## period=Period night 81.24239 34.3548802 35.5180619 1.762950e-04  
## v.test  
## period=Period morning 4.283386  
## Trip\_type=Street-Hail 3.165875  
## RateCodeID=Rate-1 2.915742  
## period=Period valley 2.517915  
## Payment\_type=No paid 2.046800  
## Trip\_distance\_range=Long\_dist -2.069090  
## RateCodeID=Rate-Other -2.915742  
## Trip\_type=Dispatch -3.165875  
## period=Period night -3.750766

* TipIsGiven
  + Same variable.
* Payment\_type
  + We see that it is very likely that there will be a tip if it is paid in a concise manner.
* Trip\_distance\_range
  + As we can see, there is tip as long as the trip is, or very short, or very long.
* Trip\_type
  + We don’t think the type of trip is important.
* RateCodeID
  + As we have seen before, virtually all observations were of type 1. Therefore it is not worth looking at the correlation.
* period
  + We see that in the morning people are not in a very good mood and are more inclined to tip the “valley”.

# Finally, save the data

save.image("Taxi5000\_del1.RData")