Deliverable 3

Numeric and Binary targets Forecasting Models

Júlia Gasull i Claudia Sánchez

# First setups

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

## Load Required Packages for this deliverable

We load the necessary packages and set working directory

#setwd("~/Documents/uni/FIB-ADEI-LAB/deliverable3")  
#filepath<-"~/Documents/uni/FIB-ADEI-LAB/deliverable3"  
setwd("C:/Users/Claudia Sánchez/Desktop/FIB/TARDOR 2020-2021/ADEI/DELIVERABLE1/FIB-ADEI-LAB/deliverable2")  
filepath<-"C:/Users/Claudia Sánchez/Desktop/FIB/TARDOR 2020-2021/ADEI/DELIVERABLE1/FIB-ADEI-LAB/deliverable2"  
  
# Load Required Packages  
options(contrasts=c("contr.treatment","contr.treatment"))  
requiredPackages <- c("missMDA","chemometrics","mvoutlier","effects","FactoMineR","car","lmtest","ggplot2","moments","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)

## Load processed data from last deliverable

#load(paste0(filepath,"/Taxi5000\_del2.RData"))  
load("C:/Users/Claudia Sánchez/Desktop/FIB/TARDOR 2020-2021/ADEI/DELIVERABLE1/FIB-ADEI-LAB/deliverable3/Taxi5000\_del2.RData")  
summary(df)

## VendorID RateCodeID Pickup\_longitude Pickup\_latitude  
## f.Vendor-Mobile : 973 Rate-1 :4496 Min. :-74.02 Min. :40.58   
## f.Vendor-VeriFone:3650 Rate-Other: 127 1st Qu.:-73.96 1st Qu.:40.70   
## Median :-73.94 Median :40.75   
## Mean :-73.93 Mean :40.75   
## 3rd Qu.:-73.92 3rd Qu.:40.80   
## Max. :-73.80 Max. :40.86   
## Dropoff\_longitude Dropoff\_latitude Passenger\_count Trip\_distance   
## Min. :-74.02 Min. :40.58 Min. :1.000 Min. : 0.010   
## 1st Qu.:-73.96 1st Qu.:40.70 1st Qu.:1.000 1st Qu.: 1.010   
## Median :-73.94 Median :40.75 Median :1.000 Median : 1.760   
## Mean :-73.93 Mean :40.75 Mean :1.371 Mean : 2.724   
## 3rd Qu.:-73.91 3rd Qu.:40.79 3rd Qu.:1.000 3rd Qu.: 3.400   
## Max. :-73.80 Max. :40.86 Max. :6.000 Max. :30.000   
## Fare\_amount Extra MTA\_tax Tip\_amount Tolls\_amount   
## Min. : 1.00 Min. :0.0000 No : 119 Min. : 0.000 Min. :0.0000   
## 1st Qu.: 6.00 1st Qu.:0.0000 Yes:4504 1st Qu.: 0.000 1st Qu.:0.0000   
## Median : 9.00 Median :0.5000 Median : 0.000 Median :0.0000   
## Mean :11.61 Mean :0.3523 Mean : 1.022 Mean :0.0477   
## 3rd Qu.:14.50 3rd Qu.:0.5000 3rd Qu.: 1.700 3rd Qu.:0.0000   
## Max. :60.00 Max. :1.0000 Max. :17.000 Max. :5.5400   
## improvement\_surcharge Total\_amount Payment\_type Trip\_type   
## No : 118 Min. : 0.00 Credit card:2096 Street-Hail:4511   
## Yes:4505 1st Qu.: 7.80 Cash :2497 Dispatch : 112   
## Median : 10.80 No paid : 30   
## Mean : 13.93   
## 3rd Qu.: 17.00   
## Max. :128.76   
## hour period tlenkm traveltime   
## Min. : 0.0 Period night :1642 Min. : 1.000 Min. : 0.000   
## 1st Qu.: 9.0 Period morning : 542 1st Qu.: 1.609 1st Qu.: 5.767   
## Median :15.0 Period valley :1260 Median : 2.800 Median : 9.550   
## Mean :13.4 Period afternoon:1179 Mean : 4.385 Mean :12.487   
## 3rd Qu.:19.0 3rd Qu.: 5.472 3rd Qu.:16.125   
## Max. :23.0 Max. :48.280 Max. :60.000   
## espeed pickup dropoff Trip\_distance\_range  
## Min. : 3.00 Length:4623 Length:4623 Long\_dist : 666   
## 1st Qu.:14.83 Class :character Class :character Medium\_dist: 986   
## Median :18.56 Mode :character Mode :character Short\_dist :2971   
## Mean :20.34   
## 3rd Qu.:23.58   
## Max. :55.00   
## TipIsGiven passenger\_groups paidTolls hcpck claKM   
## No :2882 Couple: 345 No :4580 kHP-1:1930 kKM-3: 844   
## Yes:1741 Group : 395 Yes: 43 kHP-2:1634 kKM-5:2353   
## Single:3883 kHP-3: 262 kKM-2: 486   
## kHP-4: 758 kKM-1: 831   
## kHP-5: 39 kKM-4: 109   
##   
## f.cost f.tt hcpckMCA hcpckMCA\_hcpck hcpckMCA\_claKM  
## (11,18] :1188 (10,15]: 913 1: 30 kHPmca-4: 43 kHPmca-2:1620   
## (18,30] : 724 (15,20]: 549 2:1620 kHPmca-3:2813 kHPmca-3:2813   
## (30,50] : 221 (20,60]: 756 3:2813 kHPmca-2:1620 kHPmca-1: 30   
## (50,129): 63 (5,10] :1511 4: 43 kHPmca-1: 30 kHPmca-4: 43   
## (8,11] :1151 [0,5] : 894 5: 117 kHPmca-5: 117 kHPmca-5: 117   
## [0,8] :1276

# Refactor

names(df)

## [1] "VendorID" "RateCodeID" "Pickup\_longitude"   
## [4] "Pickup\_latitude" "Dropoff\_longitude" "Dropoff\_latitude"   
## [7] "Passenger\_count" "Trip\_distance" "Fare\_amount"   
## [10] "Extra" "MTA\_tax" "Tip\_amount"   
## [13] "Tolls\_amount" "improvement\_surcharge" "Total\_amount"   
## [16] "Payment\_type" "Trip\_type" "hour"   
## [19] "period" "tlenkm" "traveltime"   
## [22] "espeed" "pickup" "dropoff"   
## [25] "Trip\_distance\_range" "TipIsGiven" "passenger\_groups"   
## [28] "paidTolls" "hcpck" "claKM"   
## [31] "f.cost" "f.tt" "hcpckMCA"   
## [34] "hcpckMCA\_hcpck" "hcpckMCA\_claKM"

names(df)[names(df) == "VendorID"] <- "f.vendor\_id"  
names(df)[names(df) == "RateCodeID"] <- "f.code\_rate\_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"  
names(df)[names(df) == "Extra"] <- "q.extra"  
names(df)[names(df) == "MTA\_tax"] <- "f.mta\_tax"  
names(df)[names(df) == "Tip\_amount"] <- "q.tip\_amount"  
names(df)[names(df) == "Tolls\_amount"] <- "q.tolls\_amount"  
names(df)[names(df) == "improvement\_surcharge"] <- "f.improvement\_surcharge"  
names(df)[names(df) == "Total\_amount"] <- "target.total\_amount"  
names(df)[names(df) == "Payment\_type"] <- "f.payment\_type"  
names(df)[names(df) == "Trip\_type"] <- "f.trip\_type"  
names(df)[names(df) == "hour"] <- "q.hour"  
names(df)[names(df) == "period"] <- "f.period"  
names(df)[names(df) == "tlenkm"] <- "q.tlenkm"  
names(df)[names(df) == "traveltime"] <- "q.traveltime"  
names(df)[names(df) == "espeed"] <- "q.espeed"  
names(df)[names(df) == "pickup"] <- "qual.pickup"  
names(df)[names(df) == "dropoff"] <- "qual.dropoff"  
names(df)[names(df) == "Trip\_distance\_range"] <- "f.trip\_distance\_range"  
names(df)[names(df) == "paidTolls"] <- "f.paid\_tolls"  
names(df)[names(df) == "TipIsGiven"] <- "target.tip\_is\_given"  
names(df)[names(df) == "passenger\_groups"] <- "f.passenger\_groups"  
#names(df)[names(df) == "f.cost"] <- ""  
#names(df)[names(df) == "f.tt"] <- ""  
  
df$hcpck <- NULL  
df$claKM <- NULL  
df$hcpckMCA <- NULL  
df$hcpckMCA\_hcpck <- NULL  
df$hcpckMCA\_claKM <- NULL  
  
names(df)

## [1] "f.vendor\_id" "f.code\_rate\_id"   
## [3] "q.pickup\_longitude" "q.pickup\_latitude"   
## [5] "q.dropoff\_longitude" "q.dropoff\_latitude"   
## [7] "q.passenger\_count" "q.trip\_distance"   
## [9] "q.fare\_amount" "q.extra"   
## [11] "f.mta\_tax" "q.tip\_amount"   
## [13] "q.tolls\_amount" "f.improvement\_surcharge"  
## [15] "target.total\_amount" "f.payment\_type"   
## [17] "f.trip\_type" "q.hour"   
## [19] "f.period" "q.tlenkm"   
## [21] "q.traveltime" "q.espeed"   
## [23] "qual.pickup" "qual.dropoff"   
## [25] "f.trip\_distance\_range" "target.tip\_is\_given"   
## [27] "f.passenger\_groups" "f.paid\_tolls"   
## [29] "f.cost" "f.tt"

Remove total amount equal to 0

df<-df[!(df$target.total\_amount=="0"),]

# Create factors needed for this deliverable (according to teacher’s video recording)

We must create: f.cost, f.dist, f.tt and f.hour. We already have f.cost and f.tt, so we will only have to create f.dist and f.hour:

## f.dist

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)

## f.hour

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)

## f.espeed

df$f.espeed[(df$q.espeed>=3) & (df$q.espeed<10)] = "[03,10)"  
df$f.espeed[(df$q.espeed>=10) & (df$q.espeed<20)] = "[10,20)"  
df$f.espeed[(df$q.espeed>=20) & (df$q.espeed<30)] = "[20,30)"  
df$f.espeed[(df$q.espeed>=30) & (df$q.espeed<40)] = "[30,40)"  
df$f.espeed[(df$q.espeed>=40) & (df$q.espeed<50)] = "[40,50)"  
df$f.espeed[(df$q.espeed>=50) & (df$q.espeed<=55)] = "[50,55]"  
df$f.espeed<-factor(df$f.espeed)

# Listing out variables

vars\_con<-names(df)[c(3:10,12:13,15,18,20:22)];  
vars\_dis<-names(df)[c(1:2,16,19,27:32)];   
vars\_res<-names(df)[c(15,27)];  
vars\_cexp<-vars\_con[c(5:10,12:15)];

# Useful information

## Y (Numeric Target).

This variable will be the target for linear model building (connected to blocks Statistical Modeling I and II).

# Quantitative Logistics Regression (explanatory variables numeric only)

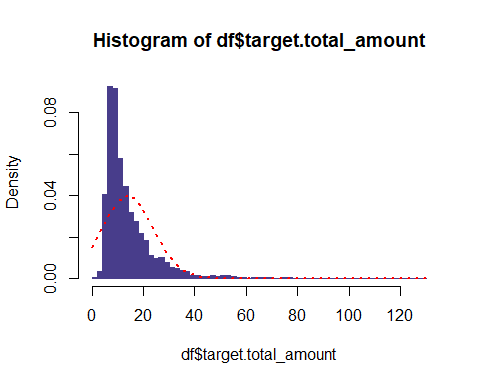
Steps to follow:

1. Enter all relevant numerical variables in the model
2. See if you need to replace a number with its equivalent factor
3. Add to the best model of step 2, the main effects of the factors and retain the significant net effects.
4. Add interactions: between factor-factor and between factor-numeric (doubles).
5. Diagnosis of waste and observations. Lack of adjustment and / or influential.

Before we begin to see correlations with our target, we should consider the normality of this.

## Normality

hist(df$target.total\_amount,50,freq=F,col="darkslateblue",border = "darkslateblue")  
mm<-mean(df$target.total\_amount);ss<-sd(df$target.total\_amount)  
curve(dnorm(x,mean=mm,sd=ss),col="red",lwd=2,lty=3, add=T)



shapiro.test(df$target.total\_amount)

##   
## Shapiro-Wilk normality test  
##   
## data: df$target.total\_amount  
## W = 0.73071, p-value < 2.2e-16

We see that the target total\_amount is not normally distributed for the following reasons:

* graph: there is no symmetry in the plot
* shapiro: we see that the p-value is too large to accept the assumption that target.total\_amount is normally distributed

### Symmetry

skewness(df$target.total\_amount)

## [1] 3.176789

Normal data should have 0 skewness: we see that our data is right skewed (3.18).

### Kurtosis

kurtosis(df$target.total\_amount)

## [1] 21.09556

Normal data should be 3. We have 21.1, so, in this case, our data is not normal.

## Method 1: take the most correlated variables

# we use spearman method since out target is not normally distributed  
round(cor(df[,c("target.total\_amount",vars\_cexp)], method="spearman"),dig=2)

## target.total\_amount q.passenger\_count q.trip\_distance  
## target.total\_amount 1.00 0.01 0.93  
## q.passenger\_count 0.01 1.00 0.01  
## q.trip\_distance 0.93 0.01 1.00  
## q.fare\_amount 0.97 0.01 0.95  
## q.extra 0.03 0.05 -0.05  
## q.tip\_amount 0.41 -0.01 0.26  
## q.tolls\_amount 0.15 0.01 0.14  
## q.hour -0.01 0.01 -0.05  
## q.tlenkm 0.91 0.00 0.98  
## q.traveltime 0.90 -0.01 0.87  
## q.espeed 0.29 0.02 0.46  
## q.fare\_amount q.extra q.tip\_amount q.tolls\_amount q.hour  
## target.total\_amount 0.97 0.03 0.41 0.15 -0.01  
## q.passenger\_count 0.01 0.05 -0.01 0.01 0.01  
## q.trip\_distance 0.95 -0.05 0.26 0.14 -0.05  
## q.fare\_amount 1.00 -0.06 0.25 0.14 -0.04  
## q.extra -0.06 1.00 0.02 -0.02 0.32  
## q.tip\_amount 0.25 0.02 1.00 0.11 0.02  
## q.tolls\_amount 0.14 -0.02 0.11 1.00 -0.01  
## q.hour -0.04 0.32 0.02 -0.01 1.00  
## q.tlenkm 0.94 -0.03 0.25 0.14 -0.04  
## q.traveltime 0.93 -0.03 0.22 0.11 -0.02  
## q.espeed 0.28 -0.01 0.14 0.12 -0.07  
## q.tlenkm q.traveltime q.espeed  
## target.total\_amount 0.91 0.90 0.29  
## q.passenger\_count 0.00 -0.01 0.02  
## q.trip\_distance 0.98 0.87 0.46  
## q.fare\_amount 0.94 0.93 0.28  
## q.extra -0.03 -0.03 -0.01  
## q.tip\_amount 0.25 0.22 0.14  
## q.tolls\_amount 0.14 0.11 0.12  
## q.hour -0.04 -0.02 -0.07  
## q.tlenkm 1.00 0.88 0.45  
## q.traveltime 0.88 1.00 0.05  
## q.espeed 0.45 0.05 1.00

We see that the diagonal is full of ‘1’, since this command gives us the correlation between the same variable. Apart from this diagonal, however, there are more high correlations. Let’s see which ones are correlated with our target:

* q.fare\_amount: 0.97
* q.trip\_distance: 0.93
* q.tlenkm: 0.91 (like trip\_distance)
* q.traveltime: 0.90
* q.tip\_amount: 0.41 (not much, but must be taken into account)
* q.espeed: 0.29 (not much, but must be taken into account)
* q.tolls\_amount: 0.15 (not much, but must be taken into account)
* we can see that some of them are not correlated:
  + q.extra (0.03)
  + q.passenger\_count (0.01)
  + q.hour (-0.01)

After seeing the correlation, to make an initial model, we should select the ones that are most correlated, which are:

* q.fare\_amount
* q.trip\_distance (we are not taking tlenkm because of redundance)
* q.traveltime
* q.tip\_amount
* q.espeed
* q.tolls\_amount

## Method 2: take the entire dataset with a condes

res.con <- condes(df,num.var=which(names(df)=="target.total\_amount"))

res.con$quanti

## correlation p.value  
## q.fare\_amount 0.94425003 0.000000e+00  
## q.trip\_distance 0.89702734 0.000000e+00  
## q.tlenkm 0.88671294 0.000000e+00  
## q.traveltime 0.76448863 0.000000e+00  
## q.tip\_amount 0.56622837 0.000000e+00  
## q.espeed 0.39683909 9.313540e-174  
## q.tolls\_amount 0.25751662 9.659999e-71  
## q.hour -0.03110910 3.465376e-02  
## q.pickup\_longitude -0.04064371 5.775239e-03  
## q.dropoff\_longitude -0.06391905 1.401371e-05  
## q.pickup\_latitude -0.12322848 4.560732e-17  
## q.dropoff\_latitude -0.14812217 4.926074e-24

Com hem pogut veure abans, les variables més correlacionades són:

* q.fare\_amount: 0.94
  + it is normal for the rate to go up when the price goes up
* q.trip\_distance: 0.90
  + the more distance, the more time, and therefore the more price
* q.tlenkm: 0.88
  + just like the previous one
* q.traveltime: 0.76
  + the longer, the more price
* q.tip\_amount: 0.57
  + not so much related, but we can keep in mind that people tend to give a percentage of the total price
* q.espeed: 0.40
* q.tolls\_amount: 0.26

res.con$quali

## R2 p.value  
## f.trip\_distance\_range 0.567177647 0.000000e+00  
## f.cost 0.908376615 0.000000e+00  
## f.tt 0.539010171 0.000000e+00  
## f.dist 0.636791987 0.000000e+00  
## f.espeed 0.171132867 1.210354e-184  
## f.paid\_tolls 0.079593357 4.072991e-85  
## target.tip\_is\_given 0.057803014 1.250800e-61  
## f.payment\_type 0.052910669 4.024719e-55  
## f.code\_rate\_id 0.018930689 6.290954e-21  
## f.mta\_tax 0.005160632 1.044478e-06  
## f.trip\_type 0.003203349 1.204051e-04  
## f.improvement\_surcharge 0.002760154 3.583467e-04  
## qual.dropoff 0.008369578 2.171667e-02

To talk about factor variables, we need to visualize res.con$quali. So let’s see:

* f.trip\_distance\_range
  + we see that they are totally related, just as we see with que.trip\_distance, since the longer distance, the longer time, and therefore the more price
* f.cost
  + is equivalent to our target
* f.tt
  + he longer time, the more price
* f.dist
  + just like with f.trip\_distance\_range
* f.paid\_tolls
  + if you pay more, it means that the trip has lasted longer, and therefore has been longer, and is more likely to have gone through more tolls
* target.tip\_is\_given
  + just like before, but we can keep in mind that people tend to give a percentage of the total price

## Method 3: if few explanatory variables are available -> take all of them

vars\_cexp

## [1] "q.passenger\_count" "q.trip\_distance" "q.fare\_amount"   
## [4] "q.extra" "q.tip\_amount" "q.tolls\_amount"   
## [7] "q.hour" "q.tlenkm" "q.traveltime"   
## [10] "q.espeed"

cor(df$q.trip\_distance,df$q.tlenkm)

## [1] 0.9951289

To give an example, we see that the two distances we have, trip\_distance and tlenkm, are closely related, since they represent the same.

### Model 1

model\_1 <- lm(  
 target.total\_amount~.  
 ,data=df[,c("target.total\_amount",vars\_cexp)]  
);summary(model\_1)

##   
## Call:  
## lm(formula = target.total\_amount ~ ., data = df[, c("target.total\_amount",   
## vars\_cexp)])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.562 -0.198 -0.055 0.071 94.934   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.153602 0.189353 11.373 < 2e-16 \*\*\*  
## q.passenger\_count 0.008078 0.036749 0.220 0.826033   
## q.trip\_distance 0.241864 0.160027 1.511 0.130756   
## q.fare\_amount 0.907127 0.014705 61.687 < 2e-16 \*\*\*  
## q.extra 1.072076 0.107278 9.993 < 2e-16 \*\*\*  
## q.tip\_amount 1.045374 0.023134 45.189 < 2e-16 \*\*\*  
## q.tolls\_amount 1.032744 0.077728 13.287 < 2e-16 \*\*\*  
## q.hour -0.000386 0.005808 -0.066 0.947009   
## q.tlenkm 0.303267 0.091687 3.308 0.000948 \*\*\*  
## q.traveltime -0.062887 0.008534 -7.369 2.02e-13 \*\*\*  
## q.espeed -0.070566 0.007275 -9.700 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.581 on 4600 degrees of freedom  
## Multiple R-squared: 0.934, Adjusted R-squared: 0.9338   
## F-statistic: 6506 on 10 and 4600 DF, p-value: < 2.2e-16

Model\_1 explains 93.4% of the variability of the target. We also see, according to the F-statistic, that it should be rejected.

We cannot use variables that are so correlated at the same time to act as explanatory variables. Therefore, we need to make a model in which we do not have these correlations.

But first, let’s see which of them are that correlated:

vif(model\_1) # Check association between explanatory vars

## q.passenger\_count q.trip\_distance q.fare\_amount q.extra   
## 1.004241 137.215426 10.203484 1.071071   
## q.tip\_amount q.tolls\_amount q.hour q.tlenkm   
## 1.247479 1.069987 1.073015 116.473412   
## q.traveltime q.espeed   
## 5.069225 2.779880

When the variance inflation factor is greater than 5, we need to consider whether or not we keep a variable.

* q.trip\_distance: 137.215426
* q.tlenkm: 116.473412
* q.fare\_amount: 10.203484
* q.traveltime: 5.069225

In this case we have to choose how far we stay. Since we work better with km than with miles (or inches, or whatever it is), we could choose the variable q.tlenkm.

### Model 1 with BIC

model\_1\_bic <- step( model\_1, k=log(nrow(df)) )

## Start: AIC=8826.82  
## target.total\_amount ~ q.passenger\_count + q.trip\_distance + q.fare\_amount +   
## q.extra + q.tip\_amount + q.tolls\_amount + q.hour + q.tlenkm +   
## q.traveltime + q.espeed  
##   
## Df Sum of Sq RSS AIC  
## - q.hour 1 0.0 30650 8818.4  
## - q.passenger\_count 1 0.3 30650 8818.4  
## - q.trip\_distance 1 15.2 30665 8820.7  
## <none> 30649 8826.8  
## - q.tlenkm 1 72.9 30722 8829.3  
## - q.traveltime 1 361.8 31011 8872.5  
## - q.espeed 1 626.9 31276 8911.8  
## - q.extra 1 665.4 31315 8917.4  
## - q.tolls\_amount 1 1176.2 31826 8992.0  
## - q.tip\_amount 1 13605.8 44255 10512.3  
## - q.fare\_amount 1 25354.6 56004 11597.9  
##   
## Step: AIC=8818.39  
## target.total\_amount ~ q.passenger\_count + q.trip\_distance + q.fare\_amount +   
## q.extra + q.tip\_amount + q.tolls\_amount + q.tlenkm + q.traveltime +   
## q.espeed  
##   
## Df Sum of Sq RSS AIC  
## - q.passenger\_count 1 0.3 30650 8810.0  
## - q.trip\_distance 1 15.3 30665 8812.2  
## <none> 30650 8818.4  
## - q.tlenkm 1 72.9 30722 8820.9  
## - q.traveltime 1 362.0 31012 8864.1  
## - q.espeed 1 629.8 31279 8903.7  
## - q.extra 1 702.0 31351 8914.4  
## - q.tolls\_amount 1 1176.2 31826 8983.6  
## - q.tip\_amount 1 13611.9 44261 10504.5  
## - q.fare\_amount 1 25371.8 56021 11590.9  
##   
## Step: AIC=8810  
## target.total\_amount ~ q.trip\_distance + q.fare\_amount + q.extra +   
## q.tip\_amount + q.tolls\_amount + q.tlenkm + q.traveltime +   
## q.espeed  
##   
## Df Sum of Sq RSS AIC  
## - q.trip\_distance 1 15.2 30665 8803.9  
## <none> 30650 8810.0  
## - q.tlenkm 1 73.0 30723 8812.5  
## - q.traveltime 1 362.1 31012 8855.7  
## - q.espeed 1 629.6 31279 8895.3  
## - q.extra 1 705.4 31355 8906.5  
## - q.tolls\_amount 1 1176.9 31827 8975.3  
## - q.tip\_amount 1 13614.4 44264 10496.3  
## - q.fare\_amount 1 25372.8 56023 11582.6  
##   
## Step: AIC=8803.85  
## target.total\_amount ~ q.fare\_amount + q.extra + q.tip\_amount +   
## q.tolls\_amount + q.tlenkm + q.traveltime + q.espeed  
##   
## Df Sum of Sq RSS AIC  
## <none> 30665 8803.9  
## - q.traveltime 1 387 31052 8853.2  
## - q.espeed 1 615 31280 8886.9  
## - q.extra 1 700 31365 8899.5  
## - q.tolls\_amount 1 1165 31830 8967.4  
## - q.tlenkm 1 1873 32538 9068.8  
## - q.tip\_amount 1 13724 44389 10500.9  
## - q.fare\_amount 1 33519 64184 12201.2

The BIC has been eliminating the variables it has considered, without worsening the AIC. However, since it does not take into account either correlations or concepts, it is probably not optimal.

Let’s see how it turned out:

vif(model\_1\_bic)

## q.fare\_amount q.extra q.tip\_amount q.tolls\_amount q.tlenkm   
## 7.898396 1.008633 1.241575 1.065918 9.377307   
## q.traveltime q.espeed   
## 4.984224 2.717538

Note that tlenkm still has a vif greater than 5 (9.377307), and so does fare\_amount (7.898396).

summary(model\_1\_bic)

##   
## Call:  
## lm(formula = target.total\_amount ~ q.fare\_amount + q.extra +   
## q.tip\_amount + q.tolls\_amount + q.tlenkm + q.traveltime +   
## q.espeed, data = df[, c("target.total\_amount", vars\_cexp)])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.203 -0.196 -0.053 0.070 94.855   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.103354 0.160998 13.064 < 2e-16 \*\*\*  
## q.fare\_amount 0.917656 0.012937 70.932 < 2e-16 \*\*\*  
## q.extra 1.067019 0.104097 10.250 < 2e-16 \*\*\*  
## q.tip\_amount 1.047409 0.023077 45.387 < 2e-16 \*\*\*  
## q.tolls\_amount 1.025892 0.077574 13.225 < 2e-16 \*\*\*  
## q.tlenkm 0.436186 0.026014 16.768 < 2e-16 \*\*\*  
## q.traveltime -0.064484 0.008461 -7.621 3.04e-14 \*\*\*  
## q.espeed -0.069090 0.007192 -9.606 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.581 on 4603 degrees of freedom  
## Multiple R-squared: 0.9339, Adjusted R-squared: 0.9338   
## F-statistic: 9295 on 7 and 4603 DF, p-value: < 2.2e-16

However, we see that it continues to explain much of the variability of our target (93.39%).

Therefore, we will try to make a model manually based on what model\_1\_bic has shown us and our knowledge of the data:

### Model 2

model\_2 <- lm(  
 target.total\_amount~   
 q.passenger\_count +  
 q.fare\_amount +  
 q.extra +   
 q.tip\_amount +  
 q.tolls\_amount +  
 q.hour +   
 q.tlenkm +  
 q.traveltime +  
 q.espeed   
 ,  
 data=df[,c("target.total\_amount",vars\_cexp)]  
);summary(model\_2)

##   
## Call:  
## lm(formula = target.total\_amount ~ q.passenger\_count + q.fare\_amount +   
## q.extra + q.tip\_amount + q.tolls\_amount + q.hour + q.tlenkm +   
## q.traveltime + q.espeed, data = df[, c("target.total\_amount",   
## vars\_cexp)])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.205 -0.197 -0.052 0.071 94.859   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.1016961 0.1862386 11.285 < 2e-16 \*\*\*  
## q.passenger\_count 0.0074884 0.0367525 0.204 0.839   
## q.fare\_amount 0.9176846 0.0129422 70.907 < 2e-16 \*\*\*  
## q.extra 1.0684221 0.1072657 9.961 < 2e-16 \*\*\*  
## q.tip\_amount 1.0475525 0.0230918 45.365 < 2e-16 \*\*\*  
## q.tolls\_amount 1.0257256 0.0775996 13.218 < 2e-16 \*\*\*  
## q.hour -0.0005778 0.0058073 -0.100 0.921   
## q.tlenkm 0.4361459 0.0260205 16.762 < 2e-16 \*\*\*  
## q.traveltime -0.0645068 0.0084674 -7.618 3.1e-14 \*\*\*  
## q.espeed -0.0691571 0.0072157 -9.584 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.582 on 4601 degrees of freedom  
## Multiple R-squared: 0.9339, Adjusted R-squared: 0.9338   
## F-statistic: 7226 on 9 and 4601 DF, p-value: < 2.2e-16

We see that the explainability is now 93.39%.

vif(model\_2) # Check association between explanatory vars

## q.passenger\_count q.fare\_amount q.extra q.tip\_amount   
## 1.004128 7.901266 1.070527 1.242636   
## q.tolls\_amount q.hour q.tlenkm q.traveltime   
## 1.066168 1.072503 9.378271 4.989265   
## q.espeed   
## 2.734212

Even so, owning one is still beyond the reach of the average person.

We try to make a new model without the distance:

### Model 3

model\_3 <- lm(  
 target.total\_amount~   
 q.passenger\_count +  
 q.fare\_amount +  
 q.extra +   
 q.tip\_amount +  
 q.tolls\_amount +  
 q.hour +   
 q.traveltime +  
 q.espeed   
 ,  
 data=df[,c("target.total\_amount",vars\_cexp)]  
);summary(model\_3)

##   
## Call:  
## lm(formula = target.total\_amount ~ q.passenger\_count + q.fare\_amount +   
## q.extra + q.tip\_amount + q.tolls\_amount + q.hour + q.traveltime +   
## q.espeed, data = df[, c("target.total\_amount", vars\_cexp)])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.322 -0.251 0.000 0.117 95.540   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.2903616 0.1562258 1.859 0.0631 .   
## q.passenger\_count 0.0132996 0.0378522 0.351 0.7253   
## q.fare\_amount 1.0440693 0.0108341 96.369 <2e-16 \*\*\*  
## q.extra 1.1208455 0.1104332 10.150 <2e-16 \*\*\*  
## q.tip\_amount 1.0607708 0.0237700 44.627 <2e-16 \*\*\*  
## q.tolls\_amount 1.0842604 0.0798441 13.580 <2e-16 \*\*\*  
## q.hour -0.0001983 0.0059813 -0.033 0.9736   
## q.traveltime -0.0089434 0.0080250 -1.114 0.2651   
## q.espeed 0.0052878 0.0058573 0.903 0.3667   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.659 on 4602 degrees of freedom  
## Multiple R-squared: 0.9299, Adjusted R-squared: 0.9298   
## F-statistic: 7630 on 8 and 4602 DF, p-value: < 2.2e-16

We see that the explainability is now 92.99%.

vif(model\_3) # Check association between explanatory vars

## q.passenger\_count q.fare\_amount q.extra q.tip\_amount   
## 1.004039 5.219389 1.069616 1.241186   
## q.tolls\_amount q.hour q.traveltime q.espeed   
## 1.064009 1.072486 4.224578 1.698328

The live ones are fine now. Still, we’ve pulled the distance, which conceptually we can’t afford. Therefore, we will try to remove another variable with a high vif (q.fare\_amount), instead of q.tlenkm:

### Model 4

model\_4 <- lm(  
 target.total\_amount~   
 q.passenger\_count +  
 q.extra +   
 q.tip\_amount +  
 q.tolls\_amount +  
 q.hour +   
 q.tlenkm +  
 q.traveltime +  
 q.espeed   
 ,  
 data=df[,c("target.total\_amount",vars\_cexp)]  
);summary(model\_4)

##   
## Call:  
## lm(formula = target.total\_amount ~ q.passenger\_count + q.extra +   
## q.tip\_amount + q.tolls\_amount + q.hour + q.tlenkm + q.traveltime +   
## q.espeed, data = df[, c("target.total\_amount", vars\_cexp)])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -44.146 -0.613 -0.248 0.192 94.727   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.548119 0.264727 17.180 < 2e-16 \*\*\*  
## q.passenger\_count 0.004933 0.053162 0.093 0.92607   
## q.extra 0.552686 0.154800 3.570 0.00036 \*\*\*  
## q.tip\_amount 1.227130 0.033200 36.961 < 2e-16 \*\*\*  
## q.tolls\_amount 1.308155 0.112098 11.670 < 2e-16 \*\*\*  
## q.hour 0.007250 0.008399 0.863 0.38806   
## q.tlenkm 1.511058 0.030591 49.396 < 2e-16 \*\*\*  
## q.traveltime 0.182147 0.011167 16.312 < 2e-16 \*\*\*  
## q.espeed -0.054416 0.010433 -5.216 1.91e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.734 on 4602 degrees of freedom  
## Multiple R-squared: 0.8617, Adjusted R-squared: 0.8615   
## F-statistic: 3585 on 8 and 4602 DF, p-value: < 2.2e-16

We see that the explainability is now 86.17%.

vif(model\_4) # Check association between explanatory vars

## q.passenger\_count q.extra q.tip\_amount q.tolls\_amount   
## 1.004128 1.065604 1.227688 1.063359   
## q.hour q.tlenkm q.traveltime q.espeed   
## 1.072115 6.195063 4.147204 2.731942

Despite having high vifs, we still have high explicability of the variability of our target and, given that the variable we have taken out we can remove with time and distance from the trip, we do not need it.

So we continue to stay with this variable and make new models. We apply BIC to help us a little:

model\_4\_bic <- step( model\_4, k=log(nrow(df)) )

## Start: AIC=12217.36  
## target.total\_amount ~ q.passenger\_count + q.extra + q.tip\_amount +   
## q.tolls\_amount + q.hour + q.tlenkm + q.traveltime + q.espeed  
##   
## Df Sum of Sq RSS AIC  
## - q.passenger\_count 1 0 64174 12209  
## - q.hour 1 10 64184 12210  
## <none> 64174 12217  
## - q.extra 1 178 64351 12222  
## - q.espeed 1 379 64553 12236  
## - q.tolls\_amount 1 1899 66073 12343  
## - q.traveltime 1 3710 67884 12468  
## - q.tip\_amount 1 19051 83224 13408  
## - q.tlenkm 1 34025 98198 14170  
##   
## Step: AIC=12208.94  
## target.total\_amount ~ q.extra + q.tip\_amount + q.tolls\_amount +   
## q.hour + q.tlenkm + q.traveltime + q.espeed  
##   
## Df Sum of Sq RSS AIC  
## - q.hour 1 10 64184 12201  
## <none> 64174 12209  
## - q.extra 1 179 64352 12213  
## - q.espeed 1 379 64553 12228  
## - q.tolls\_amount 1 1900 66073 12335  
## - q.traveltime 1 3710 67884 12460  
## - q.tip\_amount 1 19056 83230 13399  
## - q.tlenkm 1 34030 98204 14162  
##   
## Step: AIC=12201.24  
## target.total\_amount ~ q.extra + q.tip\_amount + q.tolls\_amount +   
## q.tlenkm + q.traveltime + q.espeed  
##   
## Df Sum of Sq RSS AIC  
## <none> 64184 12201  
## - q.extra 1 211 64395 12208  
## - q.espeed 1 391 64575 12221  
## - q.tolls\_amount 1 1902 66086 12328  
## - q.traveltime 1 3703 67887 12451  
## - q.tip\_amount 1 19088 83272 13393  
## - q.tlenkm 1 34063 98247 14156

Following BIC, we have to eliminate variables until the vif’s are less than 5. Therefore, the model that meets this is:

### Model 5

model\_5 <- lm(  
 target.total\_amount~   
 q.passenger\_count +  
 q.extra +  
 q.tip\_amount +  
 q.tolls\_amount +  
 q.tlenkm +  
 q.traveltime  
 ,  
 data=df  
);summary(model\_5)

##   
## Call:  
## lm(formula = target.total\_amount ~ q.passenger\_count + q.extra +   
## q.tip\_amount + q.tolls\_amount + q.tlenkm + q.traveltime,   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -43.380 -0.644 -0.251 0.211 94.956   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.582803 0.125371 28.578 < 2e-16 \*\*\*  
## q.passenger\_count 0.001889 0.053304 0.035 0.972   
## q.extra 0.605472 0.150868 4.013 6.08e-05 \*\*\*  
## q.tip\_amount 1.223749 0.033279 36.773 < 2e-16 \*\*\*  
## q.tolls\_amount 1.307289 0.112420 11.629 < 2e-16 \*\*\*  
## q.tlenkm 1.385255 0.019221 72.070 < 2e-16 \*\*\*  
## q.traveltime 0.221884 0.008248 26.901 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.745 on 4604 degrees of freedom  
## Multiple R-squared: 0.8609, Adjusted R-squared: 0.8607   
## F-statistic: 4748 on 6 and 4604 DF, p-value: < 2.2e-16

We see that the explainability is now 86.09%

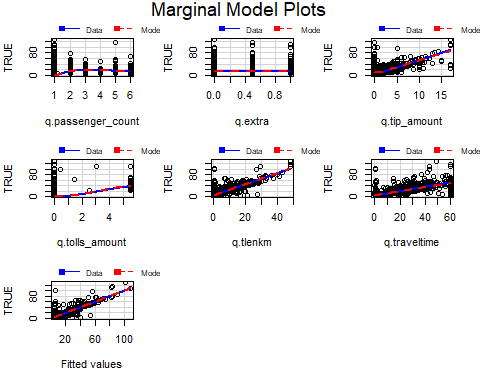
vif(model\_5) # Check association between explanatory vars

## q.passenger\_count q.extra q.tip\_amount q.tolls\_amount   
## 1.003687 1.006299 1.226347 1.063286   
## q.tlenkm q.traveltime   
## 2.431645 2.249571

There is no vif that exceeds 5.

Let’s now discriminate the variables independently:

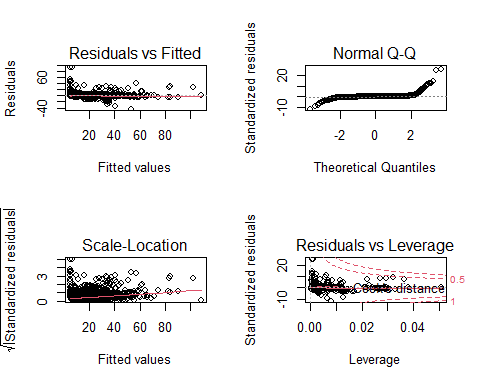
marginalModelPlots(model\_5)



We see that there is not much mismatch of the marginal variables. If there were any, we would have to transform our explanatory variables.

## Diagnostics

par(mfrow=c(2,2))  
plot(model\_5, id.n=0 )



par(mfrow=c(1,1))

Looking at the results, we can say that:

* There is no normality
* And, in terms of the Residual vs Leverage graph, our variables are within the R model, but it’s not very reliable, so it doesn’t help us much.

All this is due to the fact that our target variable was no longer normally distributed. To solve this, we apply the logarithm:

model\_6 <- lm(  
 log(target.total\_amount)~   
 q.passenger\_count +  
 q.extra +  
 q.tip\_amount +  
 q.tolls\_amount +  
 q.tlenkm +  
 q.traveltime  
 ,  
 data=df  
);summary(model\_6)

##   
## Call:  
## lm(formula = log(target.total\_amount) ~ q.passenger\_count + q.extra +   
## q.tip\_amount + q.tolls\_amount + q.tlenkm + q.traveltime,   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.49383 -0.10927 0.03793 0.14491 2.68692   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.8572872 0.0084592 219.558 < 2e-16 \*\*\*  
## q.passenger\_count -0.0014091 0.0035967 -0.392 0.695   
## q.extra 0.0704555 0.0101797 6.921 5.09e-12 \*\*\*  
## q.tip\_amount 0.0624228 0.0022454 27.800 < 2e-16 \*\*\*  
## q.tolls\_amount 0.0308942 0.0075854 4.073 4.72e-05 \*\*\*  
## q.tlenkm 0.0550138 0.0012969 42.419 < 2e-16 \*\*\*  
## q.traveltime 0.0220808 0.0005565 39.676 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2527 on 4604 degrees of freedom  
## Multiple R-squared: 0.7951, Adjusted R-squared: 0.7948   
## F-statistic: 2978 on 6 and 4604 DF, p-value: < 2.2e-16

We see that when doing the logarithm, the coefficient of determination is getting lower and lower, now it is 79.51%. We have seen that it has gotten worse than the previous model. Therefore, we discard it. We will work with model\_5.

However, let’s remember the last three models we used:

* Model 4
  + Coefficient of determination = 86,17%
  + VIFs:
    - q.passenger\_count: 1.004128
    - q.extra: 1.065604
    - q.tip\_amount: 1.227688
    - q.tolls\_amount: 1.063359
    - q.hour: 1.072115
    - q.tlenkm: 6.195063
    - q.traveltime: 4.147204
    - q.espeed: 2.731942
* Model 5
  + Coefficient of determination = 86.09%
  + VIFs:
    - q.passenger\_count: 1.003687
    - q.extra: 1.006299
    - q.tip\_amount: 1.226347
    - q.tolls\_amount: 1.063286
    - q.tlenkm: 2.431645
    - q.traveltime: .249571
* Model 6
  + Coefficient of determination = 79.51%
  + VIFs:
    - q.passenger\_count: 1.003687
    - q.extra: 1.006299
    - q.tip\_amount: 1.226347
    - q.tolls\_amount: 1.063286
    - q.tlenkm: 2.431645
    - q.traveltime: 2.249571

According to the coefficient of explicability, the ranking is: model\_4 >> model\_5 >> model\_6. As for the VIFs, however, the ranking is: model\_6 >> model\_5 >> model\_4. Since VIFs are acceptable on both model\_5 and model\_6, and not acceptable on model\_4, the smartest option is to choose model\_5.

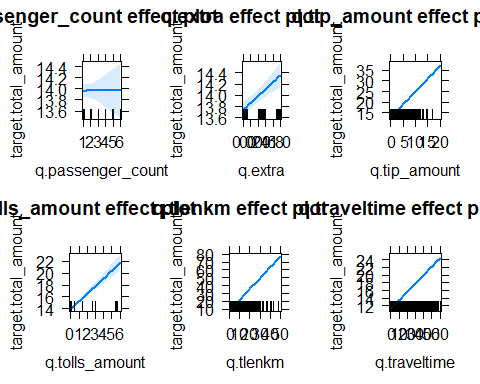
So, let’s look at the effects of this model:

Anova(model\_5)

## Anova Table (Type II tests)  
##   
## Response: target.total\_amount  
## Sum Sq Df F value Pr(>F)   
## q.passenger\_count 0 1 0.0013 0.9717   
## q.extra 226 1 16.1062 6.084e-05 \*\*\*  
## q.tip\_amount 18966 1 1352.2380 < 2.2e-16 \*\*\*  
## q.tolls\_amount 1897 1 135.2241 < 2.2e-16 \*\*\*  
## q.tlenkm 72851 1 5194.0555 < 2.2e-16 \*\*\*  
## q.traveltime 10150 1 723.6844 < 2.2e-16 \*\*\*  
## Residuals 64575 4604   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We see that now the net effects are significant.

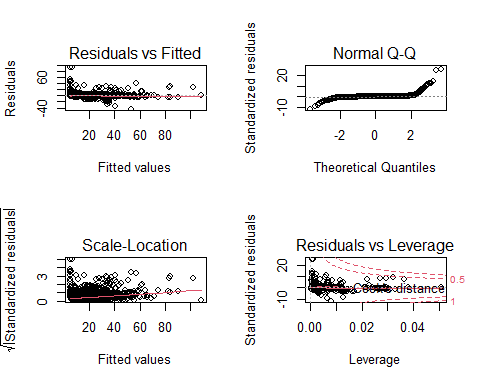
library(effects)  
plot(allEffects(model\_5))



We see that our model defines the following:

* q.passenger\_count does not depend on target.total\_amount
* q.extra grows if target.total\_amount grows
* q.tip\_amount grows if target.total\_amount grows
* q.tolls\_amount grows if target.total\_amount grows
* q.tlenkm grows if target.total\_amount grows
* q.traveltime grows if target.total\_amount grows

par(mfrow=c(2,2))  
plot(model\_5, id.n=0 )



par(mfrow=c(1,1))

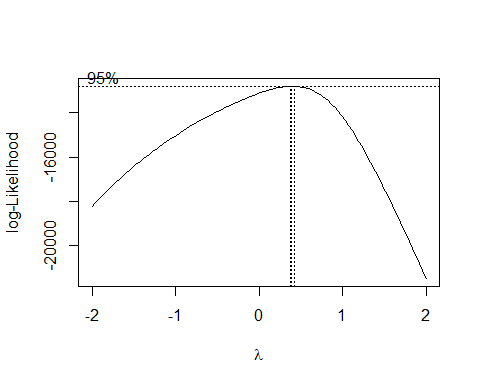
We see that the residues are not completely optimal.

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

boxcox(  
 target.total\_amount~   
 q.passenger\_count +  
 q.extra +  
 q.tip\_amount +  
 q.tolls\_amount +  
 q.tlenkm +  
 q.traveltime  
 ,  
 data=df  
)



We see the lambda parameter estimation method in the boxcox method. This gives us an idea of the power to which we need to raise the target variable in order to improve the properties of the linear model.

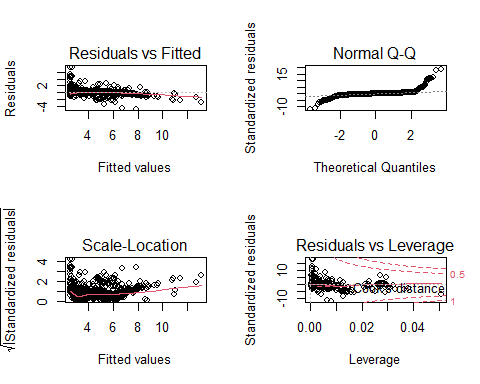
It is worth trying a new model with a square root in the target variable:

model\_7 <- lm(  
 sqrt(target.total\_amount)~   
 q.passenger\_count +  
 q.extra +  
 q.tip\_amount +  
 q.tolls\_amount +  
 q.tlenkm +  
 q.traveltime  
 ,  
 data=df  
);summary(model\_7)

##   
## Call:  
## lm(formula = sqrt(target.total\_amount) ~ q.passenger\_count +   
## q.extra + q.tip\_amount + q.tolls\_amount + q.tlenkm + q.traveltime,   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.7437 -0.1380 0.0139 0.1508 7.4872   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.3699317 0.0136357 173.804 < 2e-16 \*\*\*  
## q.passenger\_count -0.0013314 0.0057976 -0.230 0.818   
## q.extra 0.0977427 0.0164089 5.957 2.77e-09 \*\*\*  
## q.tip\_amount 0.1318869 0.0036195 36.438 < 2e-16 \*\*\*  
## q.tolls\_amount 0.1030452 0.0122272 8.428 < 2e-16 \*\*\*  
## q.tlenkm 0.1322517 0.0020905 63.262 < 2e-16 \*\*\*  
## q.traveltime 0.0357927 0.0008971 39.899 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4073 on 4604 degrees of freedom  
## Multiple R-squared: 0.8641, Adjusted R-squared: 0.8639   
## F-statistic: 4879 on 6 and 4604 DF, p-value: < 2.2e-16

We see that the coefficient has improved, from 85.09% (model\_5) to 86.41% (model\_7). But … is it worth it from a residual point of view?

par(mfrow=c(2,2))  
plot( model\_7, id.n=0 )



par(mfrow=c(1,1))

We see we haven’t won too much. So we stick to model\_5.

## Using factors as explanatory variables

### Try to change numerical each regressor by its discretized factor

model\_8<-lm(log(target.total\_amount)~ q.extra + q.tip\_amount + q.tolls\_amount + f.improvement\_surcharge + q.espeed + log(q.tlenkm), data=df)  
summary(model\_8)

##   
## Call:  
## lm(formula = log(target.total\_amount) ~ q.extra + q.tip\_amount +   
## q.tolls\_amount + f.improvement\_surcharge + q.espeed + log(q.tlenkm),   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.14903 -0.06792 -0.01991 0.05069 2.77861   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.0982020 0.0205582 102.061 < 2e-16 \*\*\*  
## q.extra 0.0884882 0.0079393 11.146 < 2e-16 \*\*\*  
## q.tip\_amount 0.0655898 0.0017109 38.337 < 2e-16 \*\*\*  
## q.tolls\_amount 0.0428318 0.0058348 7.341 2.5e-13 \*\*\*  
## f.improvement\_surchargeYes -0.2523217 0.0194490 -12.974 < 2e-16 \*\*\*  
## q.espeed -0.0091816 0.0003899 -23.550 < 2e-16 \*\*\*  
## log(q.tlenkm) 0.6191131 0.0044464 139.239 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1953 on 4604 degrees of freedom  
## Multiple R-squared: 0.8777, Adjusted R-squared: 0.8775   
## F-statistic: 5505 on 6 and 4604 DF, p-value: < 2.2e-16

We see that the explainability is now 87.77%. The more influent effects in this models are the length in km of the trip and the tip amount given.

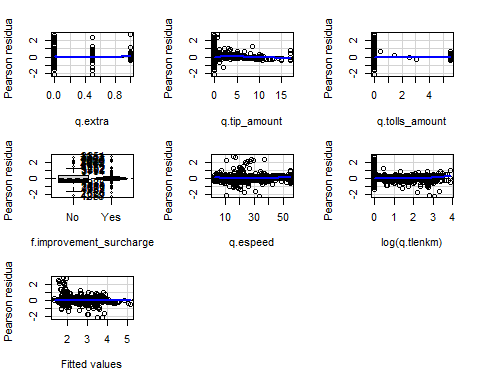
Anova(model\_8)

## Anova Table (Type II tests)  
##   
## Response: log(target.total\_amount)  
## Sum Sq Df F value Pr(>F)   
## q.extra 4.74 1 124.225 < 2.2e-16 \*\*\*  
## q.tip\_amount 56.03 1 1469.717 < 2.2e-16 \*\*\*  
## q.tolls\_amount 2.05 1 53.886 2.499e-13 \*\*\*  
## f.improvement\_surcharge 6.42 1 168.312 < 2.2e-16 \*\*\*  
## q.espeed 21.14 1 554.595 < 2.2e-16 \*\*\*  
## log(q.tlenkm) 739.16 1 19387.533 < 2.2e-16 \*\*\*  
## Residuals 175.53 4604   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

vif(model\_8)

## q.extra q.tip\_amount q.tolls\_amount   
## 1.025199 1.192442 1.053741   
## f.improvement\_surcharge q.espeed log(q.tlenkm)   
## 1.027504 1.395417 1.545375

residualPlots(model\_8)



## Test stat Pr(>|Test stat|)   
## q.extra 5.5432 3.135e-08 \*\*\*  
## q.tip\_amount -4.5251 6.189e-06 \*\*\*  
## q.tolls\_amount 0.0307 0.9755   
## f.improvement\_surcharge   
## q.espeed 13.5154 < 2.2e-16 \*\*\*  
## log(q.tlenkm) 13.8598 < 2.2e-16 \*\*\*  
## Tukey test -0.6750 0.4997   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# vars\_enum<-c("q.extra","q.tip\_amount","q.tolls\_amount","f.improvement\_surcharge","tlenkm")  
# vars\_edis<-c("VendorID","RateCodeID","Payment\_type","period")  
#   
df$f.extra <- factor(df$q.extra)  
  
model\_9<-lm(  
 log(target.total\_amount)~   
 f.extra +   
 q.tip\_amount +   
 q.tolls\_amount +   
 f.improvement\_surcharge +   
 q.espeed +   
 log(q.tlenkm)   
 ,data=df  
)  
BIC(model\_8,model\_9)

## df BIC  
## model\_8 8 -1917.617  
## model\_9 9 -1939.860

We can see from the BIC that the model\_9 is better than the model\_8, so it is correct to consider extra as factor. Next, we will do the same with the tolls\_amount and use the factor we had already created (paid\_tolls).

model\_10<-lm(  
 log(target.total\_amount)~   
 f.extra +   
 q.tip\_amount +   
 f.paid\_tolls +   
 f.improvement\_surcharge +   
 q.espeed +   
 log(q.tlenkm)   
 ,data=df  
)  
BIC(model\_8,model\_9,model\_10)

## df BIC  
## model\_8 8 -1917.617  
## model\_9 9 -1939.860  
## model\_10 9 -1944.606

We see can see that it is correct to use the paid\_tolls factor to improve our model. We will try it now with the effective speed.

model\_11<-lm(  
 log(target.total\_amount)~   
 f.extra +   
 q.tip\_amount +   
 f.paid\_tolls +   
 f.improvement\_surcharge +   
 f.espeed +   
 log(q.tlenkm)   
 ,data=df  
)  
BIC(model\_8,model\_9,model\_10,model\_11)

## df BIC  
## model\_8 8 -1917.617  
## model\_9 9 -1939.860  
## model\_10 9 -1944.606  
## model\_11 13 -1963.320

We can see that the best approach is the model\_10, so we are going to stick to it for now.

model\_12 <- model\_10  
  
Anova(model\_12)

## Anova Table (Type II tests)  
##   
## Response: log(target.total\_amount)  
## Sum Sq Df F value Pr(>F)   
## f.extra 5.89 2 77.880 < 2.2e-16 \*\*\*  
## q.tip\_amount 55.28 1 1460.732 < 2.2e-16 \*\*\*  
## f.paid\_tolls 2.12 1 55.915 9.007e-14 \*\*\*  
## f.improvement\_surcharge 5.88 1 155.314 < 2.2e-16 \*\*\*  
## q.espeed 18.07 1 477.567 < 2.2e-16 \*\*\*  
## log(q.tlenkm) 730.06 1 19292.288 < 2.2e-16 \*\*\*  
## Residuals 174.19 4603   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

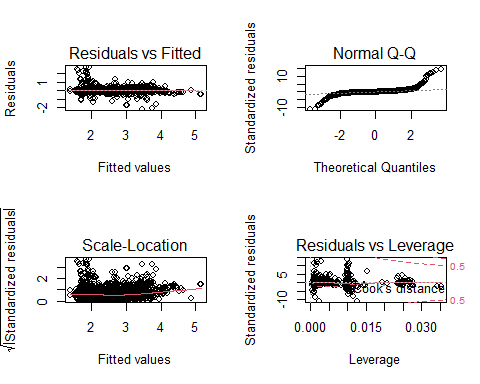
summary(model\_12)

##   
## Call:  
## lm(formula = log(target.total\_amount) ~ f.extra + q.tip\_amount +   
## f.paid\_tolls + f.improvement\_surcharge + q.espeed + log(q.tlenkm),   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.13181 -0.06786 -0.01713 0.04833 2.75572   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.0895877 0.0205470 101.698 < 2e-16 \*\*\*  
## f.extra0.5 0.0158044 0.0064600 2.446 0.0145 \*   
## f.extra1 0.1027775 0.0083225 12.349 < 2e-16 \*\*\*  
## q.tip\_amount 0.0653075 0.0017087 38.220 < 2e-16 \*\*\*  
## f.paid\_tollsYes 0.2296901 0.0307168 7.478 9.01e-14 \*\*\*  
## f.improvement\_surchargeYes -0.2424837 0.0194571 -12.462 < 2e-16 \*\*\*  
## q.espeed -0.0087026 0.0003982 -21.853 < 2e-16 \*\*\*  
## log(q.tlenkm) 0.6171457 0.0044432 138.897 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1945 on 4603 degrees of freedom  
## Multiple R-squared: 0.8786, Adjusted R-squared: 0.8784   
## F-statistic: 4759 on 7 and 4603 DF, p-value: < 2.2e-16

We can see from the Anova test that f.extra has 2 freedom degrees and globally it does have a significant net effect once the other variables are in the model.

We are going to take a look at the residues.

par(mfrow=c(2,2))  
plot( model\_12, id.n=0 )



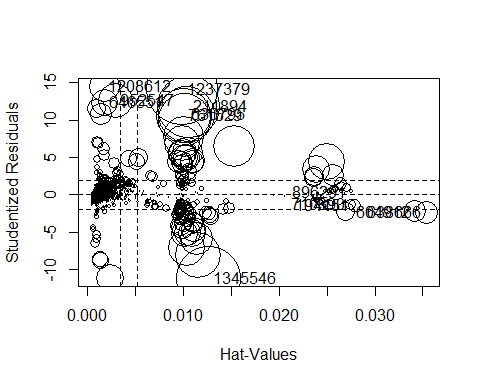
par(mfrow=c(1,1))

Looking at the results, we can say that:

* There is no normality
* And, in terms of the Residual vs Leverage graph, our variables are within the R model, but it’s not very reliable, so it doesn’t help us much.

We proceed to take a look at the influence plot to check our influent residuals for model\_12.

influencePlot( model\_12, id=c(list="noteworthy",n=5))

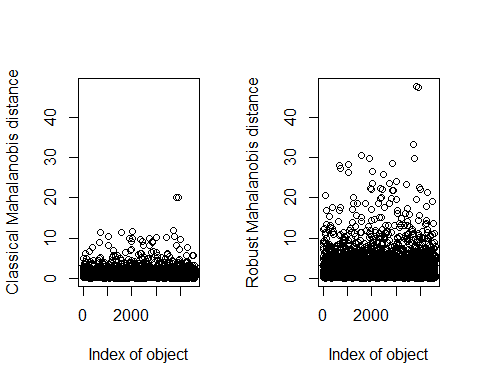


## StudRes Hat CookD  
## 194151 -1.4165961 0.027830775 0.0071794393  
## 211894 11.8720921 0.010543346 0.1821960413  
## 604912 -2.2551896 0.034191469 0.0224862744  
## 636795 10.7789754 0.010322738 0.1477856180  
## 638666 -2.3209123 0.035346786 0.0246486143  
## 646551 12.3519800 0.001824547 0.0337490049  
## 710390 -1.2640287 0.027396540 0.0056250635  
## 761529 10.6109774 0.010077102 0.1398787908  
## 896291 0.4256356 0.027534736 0.0006413128  
## 952547 12.6484050 0.002931994 0.0568424885  
## 1208612 14.4971050 0.001762369 0.0443645191  
## 1237379 14.0568677 0.009901130 0.2368806919  
## 1345546 -11.1335622 0.012642289 0.1932327624

We see this model as a disaster. That is, we have a student waste of the order of 35. We can confirm that this is too much. We have to compare student waste with a normal standard. Therefore, we would say that the model we have so far is a model that has a serious waste problem.

Intento treure outliers multivariants de total\_amount i tlenkm:

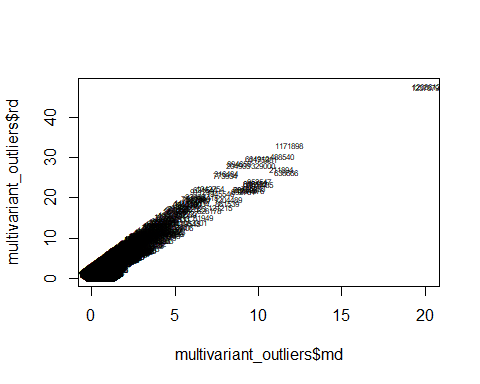
library(mvoutlier)  
library(chemometrics)  
multivariant\_outliers <- Moutlier(df[, c(15,20)], quantile = 0.995)



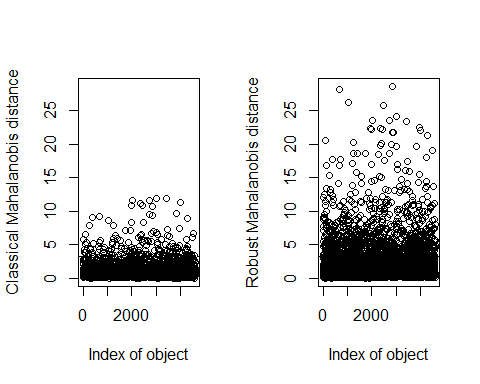
multivariant\_outliers$cutoff

## [1] 3.255247

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(15,20)]), cex=0.5)



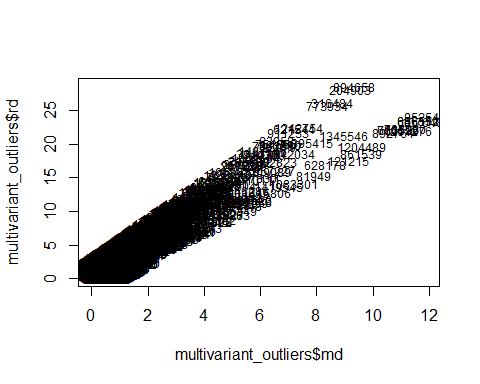
ll\_mvoutliers<-c('1237379', '1208612', '1171898', '488540', '211894', '638666', '329000', '1175981', '604912')  
  
df <- df[!(row.names(df) %in% ll\_mvoutliers),]  
  
multivariant\_outliers <- Moutlier(df[, c(15,20)], quantile = 0.995)



multivariant\_outliers$cutoff

## [1] 3.255247

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(15,20)]), cex=0.75)



In order for this not to happen to us, we need to work on the variable q.tlenkm.

So let’s create a new model that does not give so many problems:

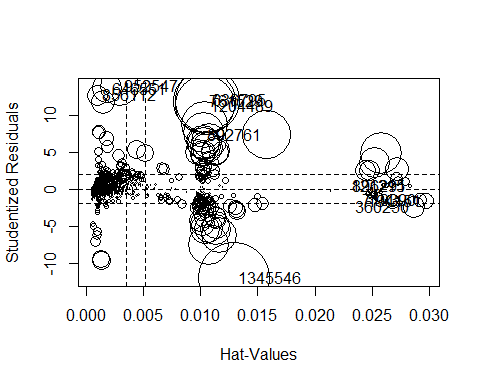
model\_13<-lm(  
 log(target.total\_amount)~   
 f.extra +   
 q.tip\_amount +   
 f.paid\_tolls +   
 f.improvement\_surcharge +   
 q.espeed +   
 log(q.tlenkm)  
 ,data=df  
)  
  
summary(model\_13)

##   
## Call:  
## lm(formula = log(target.total\_amount) ~ f.extra + q.tip\_amount +   
## f.paid\_tolls + f.improvement\_surcharge + q.espeed + log(q.tlenkm),   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.10502 -0.06679 -0.01703 0.04902 2.42599   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.0557085 0.0190514 107.903 < 2e-16 \*\*\*  
## f.extra0.5 0.0175034 0.0059203 2.957 0.00313 \*\*   
## f.extra1 0.0999597 0.0076298 13.101 < 2e-16 \*\*\*  
## q.tip\_amount 0.0654379 0.0015946 41.038 < 2e-16 \*\*\*  
## f.paid\_tollsYes 0.2460097 0.0286456 8.588 < 2e-16 \*\*\*  
## f.improvement\_surchargeYes -0.2110400 0.0180607 -11.685 < 2e-16 \*\*\*  
## q.espeed -0.0089655 0.0003656 -24.521 < 2e-16 \*\*\*  
## log(q.tlenkm) 0.6234997 0.0040831 152.702 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1782 on 4594 degrees of freedom  
## Multiple R-squared: 0.8959, Adjusted R-squared: 0.8957   
## F-statistic: 5648 on 7 and 4594 DF, p-value: < 2.2e-16

vif(model\_13)

## GVIF Df GVIF^(1/(2\*Df))  
## f.extra 1.084371 2 1.020456  
## q.tip\_amount 1.182362 1 1.087365  
## f.paid\_tolls 1.050503 1 1.024941  
## f.improvement\_surcharge 1.034810 1 1.017256  
## q.espeed 1.457073 1 1.207093  
## log(q.tlenkm) 1.544211 1 1.242663

influencePlot( model\_13, id=c(list="noteworthy",n=5))

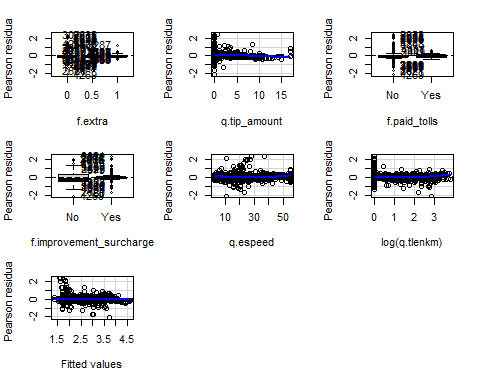


## StudRes Hat CookD  
## 121215 0.4203763 0.0283169950 0.0006438531  
## 194151 -1.6787049 0.0297354741 0.0107912464  
## 360250 -2.4687439 0.0287198240 0.0225018374  
## 636795 12.0427760 0.0106079001 0.1884592859  
## 646551 13.5889482 0.0018308364 0.0407101663  
## 710390 -1.5203565 0.0292595595 0.0087064599  
## 761529 11.8520824 0.0103596683 0.1783934856  
## 856112 12.6825286 0.0009709111 0.0188829083  
## 892761 7.3439088 0.0157633181 0.1067424678  
## 896291 0.3892795 0.0283833874 0.0005534554  
## 952547 13.9202623 0.0029422996 0.0685992790  
## 1204489 11.2790483 0.0104108324 0.1628225229  
## 1345546 -12.0786387 0.0129275449 0.2315405486

After doing certain tests, taking into account the influences, the coefficients of explicability and the vifs, we decided that the best we can get is a model where q.tlenkm does not apply any operation.

So let’s analyze it:

residualPlots(model\_13)



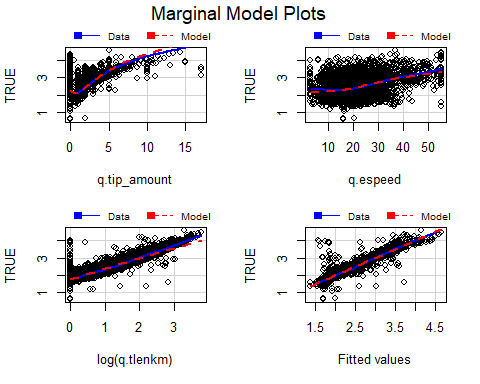
## Test stat Pr(>|Test stat|)   
## f.extra   
## q.tip\_amount -4.3322 1.508e-05 \*\*\*  
## f.paid\_tolls   
## f.improvement\_surcharge   
## q.espeed 14.0221 < 2.2e-16 \*\*\*  
## log(q.tlenkm) 15.5948 < 2.2e-16 \*\*\*  
## Tukey test 1.0019 0.3164   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

In the residualPlots, what we find is, for each factor, a boxplot of its categories and, for each quantitative variable, a pearson graph.

Let’s use another tool to fully understand our model:

marginalModelPlots(model\_13)

## Warning in mmps(...): Interactions and/or factors skipped

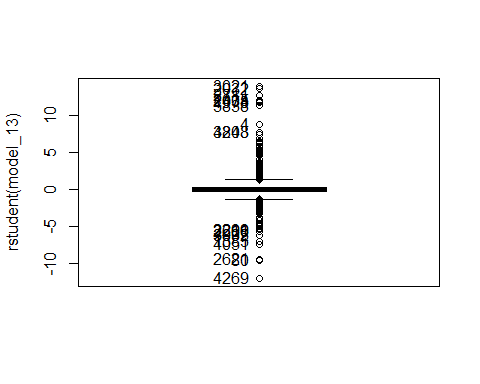


In relation to the variable q.tip\_amount, we see that there is a bit of mismatch, but not much, since tips given in cash are always declared as 0. Therefore, the data are not entirely real.

As for the variable q.tlenkm, we see that some observations do not follow the required pattern, and we have to modify them in some way.

How do we do that?

ll1<-Boxplot(rstudent(model\_13));ll1



## [1] 4269 80 2621 4051 1385 3035 3802 2666 3211 2299 3021 2032 2711 2005 2434  
## [16] 1978 3838 4 3808 4243

ll1<-c(4269, 80, 2621)  
df[ll1,]

## f.vendor\_id f.code\_rate\_id q.pickup\_longitude q.pickup\_latitude  
## 1345546 f.Vendor-VeriFone Rate-Other -73.92619 40.76569  
## 24990 f.Vendor-Mobile Rate-1 -73.95438 40.80410  
## 825427 f.Vendor-Mobile Rate-1 -73.93534 40.63492  
## q.dropoff\_longitude q.dropoff\_latitude q.passenger\_count  
## 1345546 -73.93353 40.76379 1  
## 24990 -73.95515 40.80468 1  
## 825427 -73.93534 40.63492 1  
## q.trip\_distance q.fare\_amount q.extra f.mta\_tax q.tip\_amount  
## 1345546 10.42 5.0 0.0 No 0  
## 24990 5.60 2.5 0.5 Yes 0  
## 825427 5.50 2.5 0.5 Yes 0  
## q.tolls\_amount f.improvement\_surcharge target.total\_amount  
## 1345546 0 No 5.0  
## 24990 0 Yes 3.8  
## 825427 0 Yes 3.8  
## f.payment\_type f.trip\_type q.hour f.period q.tlenkm q.traveltime  
## 1345546 Cash Dispatch 9 Period morning 16.769364 60.0000000  
## 24990 No paid Street-Hail 3 Period night 9.012326 0.5333333  
## 825427 No paid Street-Hail 0 Period night 8.851392 0.2666667  
## q.espeed qual.pickup qual.dropoff f.trip\_distance\_range  
## 1345546 11.06889 09 11 Long\_dist  
## 24990 23.16672 03 03 Long\_dist  
## 825427 23.05353 00 00 Long\_dist  
## target.tip\_is\_given f.passenger\_groups f.paid\_tolls f.cost f.tt  
## 1345546 No Single No [0,8] (20,60]  
## 24990 No Single No [0,8] [0,5]  
## 825427 No Single No [0,8] [0,5]  
## f.dist f.hour f.espeed f.extra  
## 1345546 (5.5, 30] other [10,20) 0  
## 24990 (5.5, 30] other [20,30) 0.5  
## 825427 (3, 5.5] other [20,30) 0.5

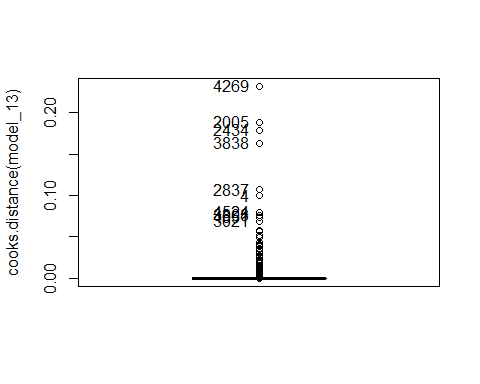
Let’s see the strangest:

* 4269
  + target.total\_amount: 5.0
  + q.tip\_amount: 0
  + q.espeed: 11.06889
  + q.tlenkm: 16.769364
* 80
  + target.total\_amount: 3.8
  + q.tip\_amount: 0
  + q.espeed: 23.16672
  + q.tlenkm: 9.012326
* 2621
  + target.total\_amount: 3.8
  + q.tip\_amount: 0
  + q.espeed: 23.05353
  + q.tlenkm: 8.851392

Veiem que són observacionsa vastant normals. Tot i això, per exemple, podem destacar que la observació 4269, a la qual ja se li aplica una tarifa de 5$, per molts km és que hagi fet, el preu no ha pujat.

We do the same for the cook distance:

ll4 <- Boxplot(cooks.distance(model\_13));ll4



## [1] 4269 2005 2434 3838 2837 4 4524 3808 4051 3021

ll4<-c(4269, 2005, 2434)  
df[ll4,]

## f.vendor\_id f.code\_rate\_id q.pickup\_longitude q.pickup\_latitude  
## 1345546 f.Vendor-VeriFone Rate-Other -73.92619 40.76569  
## 636795 f.Vendor-VeriFone Rate-Other -73.96568 40.68322  
## 761529 f.Vendor-VeriFone Rate-Other -73.94013 40.71141  
## q.dropoff\_longitude q.dropoff\_latitude q.passenger\_count  
## 1345546 -73.93353 40.76379 1  
## 636795 -73.96699 40.68422 1  
## 761529 -73.93863 40.71203 4  
## q.trip\_distance q.fare\_amount q.extra f.mta\_tax q.tip\_amount  
## 1345546 10.42000 5.00 0 No 0  
## 636795 6.39489 50.00 0 No 0  
## 761529 0.05000 49.99 0 No 0  
## q.tolls\_amount f.improvement\_surcharge target.total\_amount  
## 1345546 0 No 5.00  
## 636795 0 No 50.00  
## 761529 0 No 49.99  
## f.payment\_type f.trip\_type q.hour f.period q.tlenkm q.traveltime  
## 1345546 Cash Dispatch 9 Period morning 16.76936 60.00000000  
## 636795 Cash Dispatch 16 Period valley 1.00000 1.26666667  
## 761529 Credit card Dispatch 21 Period night 1.00000 0.03333333  
## q.espeed qual.pickup qual.dropoff f.trip\_distance\_range  
## 1345546 11.06889 09 11 Long\_dist  
## 636795 27.33968 16 16 Short\_dist  
## 761529 23.79045 21 21 Short\_dist  
## target.tip\_is\_given f.passenger\_groups f.paid\_tolls f.cost f.tt  
## 1345546 No Single No [0,8] (20,60]  
## 636795 No Single No (30,50] [0,5]  
## 761529 No Group No (30,50] [0,5]  
## f.dist f.hour f.espeed f.extra  
## 1345546 (5.5, 30] other [10,20) 0  
## 636795 (5.5, 30] other [20,30) 0  
## 761529 (0, 1.6] 21 [20,30) 0

* 4269
  + target.total\_amount: 5.0
  + q.tip\_amount: 0
  + q.espeed: 11.06889
  + q.tlenkm: 16.769364
* 2005
  + target.total\_amount: 50.00
  + q.tip\_amount: 0
  + q.espeed: 27.33968
  + q.tlenkm: 1.00000
* 2434
  + target.total\_amount: 49.99
  + q.tip\_amount: 0
  + q.espeed: 23.79045
  + q.tlenkm: 1.00000

We see that, apart from the first, explained above, the other two observations have a trip length of 1km, but instead has been paid about $ 50. We see that this is not possible.

It is necessary to eliminate these observations that do not have the same tendency as our model:

dfred<-df[-ll4,]  
  
model\_14<-lm(  
 log(target.total\_amount)~   
 f.extra +   
 q.tip\_amount +   
 f.paid\_tolls +   
 f.improvement\_surcharge +   
 q.espeed +   
 log(q.tlenkm)  
 ,data=dfred  
)  
  
summary(model\_14)

##   
## Call:  
## lm(formula = log(target.total\_amount) ~ f.extra + q.tip\_amount +   
## f.paid\_tolls + f.improvement\_surcharge + q.espeed + log(q.tlenkm),   
## data = dfred)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.69585 -0.06668 -0.01671 0.04908 2.43663   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.0373806 0.0184125 110.652 < 2e-16 \*\*\*  
## f.extra0.5 0.0184093 0.0056474 3.260 0.00112 \*\*   
## f.extra1 0.0997061 0.0072780 13.700 < 2e-16 \*\*\*  
## q.tip\_amount 0.0650028 0.0015213 42.730 < 2e-16 \*\*\*  
## f.paid\_tollsYes 0.2453415 0.0273246 8.979 < 2e-16 \*\*\*  
## f.improvement\_surchargeYes -0.1914635 0.0174708 -10.959 < 2e-16 \*\*\*  
## q.espeed -0.0093036 0.0003492 -26.642 < 2e-16 \*\*\*  
## log(q.tlenkm) 0.6286084 0.0039030 161.059 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1699 on 4591 degrees of freedom  
## Multiple R-squared: 0.905, Adjusted R-squared: 0.9049   
## F-statistic: 6248 on 7 and 4591 DF, p-value: < 2.2e-16

Anova(model\_14)

## Anova Table (Type II tests)  
##   
## Response: log(target.total\_amount)  
## Sum Sq Df F value Pr(>F)   
## f.extra 5.48 2 94.850 < 2.2e-16 \*\*\*  
## q.tip\_amount 52.73 1 1825.836 < 2.2e-16 \*\*\*  
## f.paid\_tolls 2.33 1 80.619 < 2.2e-16 \*\*\*  
## f.improvement\_surcharge 3.47 1 120.101 < 2.2e-16 \*\*\*  
## q.espeed 20.50 1 709.789 < 2.2e-16 \*\*\*  
## log(q.tlenkm) 749.16 1 25940.109 < 2.2e-16 \*\*\*  
## Residuals 132.59 4591   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

vif(model\_14)

## GVIF Df GVIF^(1/(2\*Df))  
## f.extra 1.083640 2 1.020284  
## q.tip\_amount 1.182486 1 1.087422  
## f.paid\_tolls 1.050503 1 1.024941  
## f.improvement\_surcharge 1.033891 1 1.016804  
## q.espeed 1.460196 1 1.208386  
## log(q.tlenkm) 1.547842 1 1.244123

We see that the coefficient of determination has increased a bit and it seems that we have no collinearity problems.

### Adding factors: main effects

names(df)

## [1] "f.vendor\_id" "f.code\_rate\_id"   
## [3] "q.pickup\_longitude" "q.pickup\_latitude"   
## [5] "q.dropoff\_longitude" "q.dropoff\_latitude"   
## [7] "q.passenger\_count" "q.trip\_distance"   
## [9] "q.fare\_amount" "q.extra"   
## [11] "f.mta\_tax" "q.tip\_amount"   
## [13] "q.tolls\_amount" "f.improvement\_surcharge"  
## [15] "target.total\_amount" "f.payment\_type"   
## [17] "f.trip\_type" "q.hour"   
## [19] "f.period" "q.tlenkm"   
## [21] "q.traveltime" "q.espeed"   
## [23] "qual.pickup" "qual.dropoff"   
## [25] "f.trip\_distance\_range" "target.tip\_is\_given"   
## [27] "f.passenger\_groups" "f.paid\_tolls"   
## [29] "f.cost" "f.tt"   
## [31] "f.dist" "f.hour"   
## [33] "f.espeed" "f.extra"

model\_15<-lm(  
 log(target.total\_amount) ~   
 q.tip\_amount +   
 log(q.tlenkm)+   
 f.paid\_tolls+   
 f.improvement\_surcharge +   
 f.espeed +   
 f.extra +   
 f.code\_rate\_id +   
 f.vendor\_id +   
 f.payment\_type+  
 f.period   
 ,data=df  
)  
summary(model\_15)

##   
## Call:  
## lm(formula = log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) +   
## f.paid\_tolls + f.improvement\_surcharge + f.espeed + f.extra +   
## f.code\_rate\_id + f.vendor\_id + f.payment\_type + f.period,   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.07100 -0.06106 -0.01212 0.05413 2.33447   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.4250185 0.0428280 33.273 < 2e-16 \*\*\*  
## q.tip\_amount 0.0517313 0.0019163 26.995 < 2e-16 \*\*\*  
## log(q.tlenkm) 0.6209633 0.0038383 161.782 < 2e-16 \*\*\*  
## f.paid\_tollsYes 0.1448719 0.0273812 5.291 1.27e-07 \*\*\*  
## f.improvement\_surchargeYes 0.5178918 0.0402982 12.852 < 2e-16 \*\*\*  
## f.espeed[10,20) -0.1944393 0.0114657 -16.958 < 2e-16 \*\*\*  
## f.espeed[20,30) -0.2883868 0.0122033 -23.632 < 2e-16 \*\*\*  
## f.espeed[30,40) -0.3398952 0.0149364 -22.756 < 2e-16 \*\*\*  
## f.espeed[40,50) -0.3606616 0.0189198 -19.063 < 2e-16 \*\*\*  
## f.espeed[50,55] -0.4385135 0.0261803 -16.750 < 2e-16 \*\*\*  
## f.extra0.5 0.0259337 0.0090278 2.873 0.00409 \*\*   
## f.extra1 0.1020348 0.0085383 11.950 < 2e-16 \*\*\*  
## f.code\_rate\_idRate-Other 0.7687656 0.0387554 19.836 < 2e-16 \*\*\*  
## f.vendor\_idf.Vendor-VeriFone -0.0026786 0.0061663 -0.434 0.66402   
## f.payment\_typeCash -0.0680012 0.0064312 -10.574 < 2e-16 \*\*\*  
## f.payment\_typeNo paid -0.2428288 0.0320752 -7.571 4.46e-14 \*\*\*  
## f.periodPeriod morning 0.0009375 0.0113906 0.082 0.93441   
## f.periodPeriod valley 0.0069741 0.0097913 0.712 0.47634   
## f.periodPeriod afternoon 0.0029100 0.0085276 0.341 0.73293   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1677 on 4583 degrees of freedom  
## Multiple R-squared: 0.908, Adjusted R-squared: 0.9076   
## F-statistic: 2513 on 18 and 4583 DF, p-value: < 2.2e-16

Anova(model\_15)

## Anova Table (Type II tests)  
##   
## Response: log(target.total\_amount)  
## Sum Sq Df F value Pr(>F)   
## q.tip\_amount 20.49 1 728.7238 < 2.2e-16 \*\*\*  
## log(q.tlenkm) 735.91 1 26173.4058 < 2.2e-16 \*\*\*  
## f.paid\_tolls 0.79 1 27.9939 1.274e-07 \*\*\*  
## f.improvement\_surcharge 4.64 1 165.1611 < 2.2e-16 \*\*\*  
## f.espeed 22.49 5 159.9773 < 2.2e-16 \*\*\*  
## f.extra 4.08 2 72.5752 < 2.2e-16 \*\*\*  
## f.code\_rate\_id 11.06 1 393.4798 < 2.2e-16 \*\*\*  
## f.vendor\_id 0.01 1 0.1887 0.6640   
## f.payment\_type 4.19 2 74.5335 < 2.2e-16 \*\*\*  
## f.period 0.02 3 0.2629 0.8522   
## Residuals 128.86 4583   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We see that, of all the new explanatory variables introduced, the ones we can save are:

* f.espeed: 22.49
* f.code\_rate\_id: 11.06
* f.payment\_type: 4.19

We create a new model with them:

model\_16<-lm(  
 log(target.total\_amount) ~   
 q.tip\_amount +   
 log(q.tlenkm)+   
 f.paid\_tolls+   
 f.espeed +   
 f.extra +   
 f.code\_rate\_id +   
 f.payment\_type+  
 f.period   
 ,data=df  
)  
  
anova(model\_15, model\_16)

## Analysis of Variance Table  
##   
## Model 1: log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.improvement\_surcharge + f.espeed + f.extra + f.code\_rate\_id +   
## f.vendor\_id + f.payment\_type + f.period  
## Model 2: log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + f.period  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 4583 128.86   
## 2 4585 133.50 -2 -4.6445 82.594 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We see that we haven’t lost anything.

#### Interactions

model\_17<-lm(  
 log(target.total\_amount) ~   
 (q.tip\_amount + log(q.tlenkm))\*(f.paid\_tolls + f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + f.period)  
 ,data=df  
)  
  
model\_17<-step( model\_17, k=log(nrow(df)))

## Start: AIC=-17256.64  
## log(target.total\_amount) ~ (q.tip\_amount + log(q.tlenkm)) \* (f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + f.period)  
##   
##   
## Step: AIC=-17256.64  
## log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + f.period +   
## q.tip\_amount:f.paid\_tolls + q.tip\_amount:f.espeed + q.tip\_amount:f.extra +   
## q.tip\_amount:f.code\_rate\_id + q.tip\_amount:f.period + log(q.tlenkm):f.paid\_tolls +   
## log(q.tlenkm):f.espeed + log(q.tlenkm):f.extra + log(q.tlenkm):f.code\_rate\_id +   
## log(q.tlenkm):f.payment\_type + log(q.tlenkm):f.period  
##   
## Df Sum of Sq RSS AIC  
## - log(q.tlenkm):f.period 3 0.0047 100.05 -17282  
## - q.tip\_amount:f.period 3 0.0259 100.07 -17281  
## - q.tip\_amount:f.extra 2 0.0639 100.11 -17271  
## - log(q.tlenkm):f.paid\_tolls 1 0.0581 100.10 -17262  
## <none> 100.05 -17257  
## - q.tip\_amount:f.paid\_tolls 1 0.2062 100.25 -17256  
## - log(q.tlenkm):f.extra 2 0.9401 100.99 -17231  
## - log(q.tlenkm):f.espeed 5 1.7854 101.83 -17217  
## - q.tip\_amount:f.espeed 5 1.7942 101.84 -17217  
## - log(q.tlenkm):f.payment\_type 2 2.7241 102.77 -17150  
## - q.tip\_amount:f.code\_rate\_id 1 3.2467 103.29 -17118  
## - log(q.tlenkm):f.code\_rate\_id 1 24.4450 124.49 -16259  
##   
## Step: AIC=-17281.72  
## log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + f.period +   
## q.tip\_amount:f.paid\_tolls + q.tip\_amount:f.espeed + q.tip\_amount:f.extra +   
## q.tip\_amount:f.code\_rate\_id + q.tip\_amount:f.period + log(q.tlenkm):f.paid\_tolls +   
## log(q.tlenkm):f.espeed + log(q.tlenkm):f.extra + log(q.tlenkm):f.code\_rate\_id +   
## log(q.tlenkm):f.payment\_type  
##   
## Df Sum of Sq RSS AIC  
## - q.tip\_amount:f.period 3 0.0232 100.07 -17306  
## - q.tip\_amount:f.extra 2 0.0616 100.11 -17296  
## - log(q.tlenkm):f.paid\_tolls 1 0.0584 100.11 -17288  
## <none> 100.05 -17282  
## - q.tip\_amount:f.paid\_tolls 1 0.2076 100.26 -17281  
## - log(q.tlenkm):f.espeed 5 1.7923 101.84 -17242  
## - q.tip\_amount:f.espeed 5 1.7956 101.85 -17242  
## - log(q.tlenkm):f.extra 2 1.6509 101.70 -17223  
## - log(q.tlenkm):f.payment\_type 2 2.7324 102.78 -17175  
## - q.tip\_amount:f.code\_rate\_id 1 3.2471 103.30 -17143  
## - log(q.tlenkm):f.code\_rate\_id 1 25.3794 125.43 -16250  
##   
## Step: AIC=-17305.96  
## log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + f.period +   
## q.tip\_amount:f.paid\_tolls + q.tip\_amount:f.espeed + q.tip\_amount:f.extra +   
## q.tip\_amount:f.code\_rate\_id + log(q.tlenkm):f.paid\_tolls +   
## log(q.tlenkm):f.espeed + log(q.tlenkm):f.extra + log(q.tlenkm):f.code\_rate\_id +   
## log(q.tlenkm):f.payment\_type  
##   
## Df Sum of Sq RSS AIC  
## - f.period 3 0.1722 100.25 -17323  
## - q.tip\_amount:f.extra 2 0.1242 100.20 -17317  
## - log(q.tlenkm):f.paid\_tolls 1 0.0590 100.13 -17312  
## <none> 100.07 -17306  
## - q.tip\_amount:f.paid\_tolls 1 0.2092 100.28 -17305  
## - log(q.tlenkm):f.espeed 5 1.7873 101.86 -17267  
## - q.tip\_amount:f.espeed 5 1.8682 101.94 -17263  
## - log(q.tlenkm):f.extra 2 1.6516 101.72 -17248  
## - log(q.tlenkm):f.payment\_type 2 2.7497 102.82 -17198  
## - q.tip\_amount:f.code\_rate\_id 1 3.2953 103.37 -17165  
## - log(q.tlenkm):f.code\_rate\_id 1 25.3969 125.47 -16274  
##   
## Step: AIC=-17323.35  
## log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + q.tip\_amount:f.paid\_tolls +   
## q.tip\_amount:f.espeed + q.tip\_amount:f.extra + q.tip\_amount:f.code\_rate\_id +   
## log(q.tlenkm):f.paid\_tolls + log(q.tlenkm):f.espeed + log(q.tlenkm):f.extra +   
## log(q.tlenkm):f.code\_rate\_id + log(q.tlenkm):f.payment\_type  
##   
## Df Sum of Sq RSS AIC  
## - q.tip\_amount:f.extra 2 0.1268 100.37 -17334  
## - log(q.tlenkm):f.paid\_tolls 1 0.0574 100.30 -17329  
## <none> 100.25 -17323  
## - q.tip\_amount:f.paid\_tolls 1 0.2058 100.45 -17322  
## - log(q.tlenkm):f.espeed 5 1.7958 102.04 -17284  
## - q.tip\_amount:f.espeed 5 1.8834 102.13 -17280  
## - log(q.tlenkm):f.extra 2 1.6356 101.88 -17266  
## - log(q.tlenkm):f.payment\_type 2 2.7496 103.00 -17216  
## - q.tip\_amount:f.code\_rate\_id 1 3.3059 103.55 -17183  
## - log(q.tlenkm):f.code\_rate\_id 1 25.3144 125.56 -16296  
##   
## Step: AIC=-17334.4  
## log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + q.tip\_amount:f.paid\_tolls +   
## q.tip\_amount:f.espeed + q.tip\_amount:f.code\_rate\_id + log(q.tlenkm):f.paid\_tolls +   
## log(q.tlenkm):f.espeed + log(q.tlenkm):f.extra + log(q.tlenkm):f.code\_rate\_id +   
## log(q.tlenkm):f.payment\_type  
##   
## Df Sum of Sq RSS AIC  
## - log(q.tlenkm):f.paid\_tolls 1 0.0537 100.43 -17340  
## <none> 100.37 -17334  
## - q.tip\_amount:f.paid\_tolls 1 0.2097 100.58 -17333  
## - q.tip\_amount:f.espeed 5 1.7712 102.14 -17296  
## - log(q.tlenkm):f.espeed 5 1.7817 102.15 -17296  
## - log(q.tlenkm):f.extra 2 1.8213 102.19 -17269  
## - log(q.tlenkm):f.payment\_type 2 2.7823 103.16 -17225  
## - q.tip\_amount:f.code\_rate\_id 1 3.3274 103.70 -17193  
## - log(q.tlenkm):f.code\_rate\_id 1 25.4051 125.78 -16304  
##   
## Step: AIC=-17340.37  
## log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + q.tip\_amount:f.paid\_tolls +   
## q.tip\_amount:f.espeed + q.tip\_amount:f.code\_rate\_id + log(q.tlenkm):f.espeed +   
## log(q.tlenkm):f.extra + log(q.tlenkm):f.code\_rate\_id + log(q.tlenkm):f.payment\_type  
##   
## Df Sum of Sq RSS AIC  
## - q.tip\_amount:f.paid\_tolls 1 0.1745 100.60 -17341  
## <none> 100.43 -17340  
## - q.tip\_amount:f.espeed 5 1.7304 102.16 -17304  
## - log(q.tlenkm):f.espeed 5 1.8561 102.28 -17298  
## - log(q.tlenkm):f.extra 2 1.8241 102.25 -17274  
## - log(q.tlenkm):f.payment\_type 2 2.7554 103.18 -17233  
## - q.tip\_amount:f.code\_rate\_id 1 3.3149 103.74 -17199  
## - log(q.tlenkm):f.code\_rate\_id 1 25.3540 125.78 -16313  
##   
## Step: AIC=-17340.81  
## log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) + f.paid\_tolls +   
## f.espeed + f.extra + f.code\_rate\_id + f.payment\_type + q.tip\_amount:f.espeed +   
## q.tip\_amount:f.code\_rate\_id + log(q.tlenkm):f.espeed + log(q.tlenkm):f.extra +   
## log(q.tlenkm):f.code\_rate\_id + log(q.tlenkm):f.payment\_type  
##   
## Df Sum of Sq RSS AIC  
## <none> 100.60 -17341  
## - log(q.tlenkm):f.espeed 5 1.8740 102.47 -17298  
## - q.tip\_amount:f.espeed 5 1.9522 102.55 -17295  
## - f.paid\_tolls 1 1.3113 101.91 -17290  
## - log(q.tlenkm):f.extra 2 1.8579 102.46 -17274  
## - log(q.tlenkm):f.payment\_type 2 2.7226 103.32 -17235  
## - q.tip\_amount:f.code\_rate\_id 1 3.1412 103.74 -17208  
## - log(q.tlenkm):f.code\_rate\_id 1 25.7500 126.35 -16300

This method tells us that:

* log(target.total\_amount) depends on:
  + q.tip\_amount
  + log(q.tlenkm)
  + f.paid\_tolls
  + f.espeed
  + f.extra
  + f.code\_rate\_id
  + f.payment\_type
* and there are interactionsa between:
  + q.tip\_amount:f.espeed
  + q.tip\_amount:f.code\_rate\_id
  + log(q.tlenkm):f.espeed
  + log(q.tlenkm):f.extra
  + log(q.tlenkm):f.code\_rate\_id
  + log(q.tlenkm):f.payment\_type

Anova(model\_17)

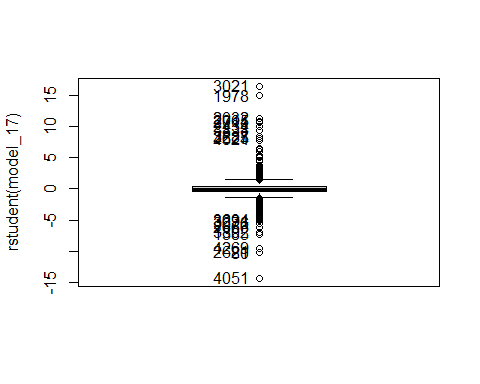
## Anova Table (Type II tests)  
##   
## Response: log(target.total\_amount)  
## Sum Sq Df F value Pr(>F)   
## q.tip\_amount 22.42 1 1018.820 < 2.2e-16 \*\*\*  
## log(q.tlenkm) 713.55 1 32428.747 < 2.2e-16 \*\*\*  
## f.paid\_tolls 1.31 1 59.596 1.421e-14 \*\*\*  
## f.espeed 22.93 5 208.405 < 2.2e-16 \*\*\*  
## f.extra 5.62 2 127.699 < 2.2e-16 \*\*\*  
## f.code\_rate\_id 8.87 1 402.972 < 2.2e-16 \*\*\*  
## f.payment\_type 2.79 2 63.393 < 2.2e-16 \*\*\*  
## q.tip\_amount:f.espeed 1.95 5 17.744 < 2.2e-16 \*\*\*  
## q.tip\_amount:f.code\_rate\_id 3.14 1 142.756 < 2.2e-16 \*\*\*  
## log(q.tlenkm):f.espeed 1.87 5 17.034 < 2.2e-16 \*\*\*  
## log(q.tlenkm):f.extra 1.86 2 42.217 < 2.2e-16 \*\*\*  
## log(q.tlenkm):f.code\_rate\_id 25.75 1 1170.261 < 2.2e-16 \*\*\*  
## log(q.tlenkm):f.payment\_type 2.72 2 61.867 < 2.2e-16 \*\*\*  
## Residuals 100.60 4572   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model\_17)

##   
## Call:  
## lm(formula = log(target.total\_amount) ~ q.tip\_amount + log(q.tlenkm) +   
## f.paid\_tolls + f.espeed + f.extra + f.code\_rate\_id + f.payment\_type +   
## q.tip\_amount:f.espeed + q.tip\_amount:f.code\_rate\_id + log(q.tlenkm):f.espeed +   
## log(q.tlenkm):f.extra + log(q.tlenkm):f.code\_rate\_id + log(q.tlenkm):f.payment\_type,   
## data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.05558 -0.05518 -0.00962 0.05245 2.35141   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1.926591 0.015134 127.298 < 2e-16  
## q.tip\_amount 0.062247 0.005199 11.972 < 2e-16  
## log(q.tlenkm) 0.619204 0.015692 39.459 < 2e-16  
## f.paid\_tollsYes 0.191814 0.024847 7.720 1.42e-14  
## f.espeed[10,20) -0.200905 0.014350 -14.000 < 2e-16  
## f.espeed[20,30) -0.286171 0.015864 -18.039 < 2e-16  
## f.espeed[30,40) -0.392635 0.023457 -16.738 < 2e-16  
## f.espeed[40,50) -0.668755 0.058657 -11.401 < 2e-16  
## f.espeed[50,55] -0.596715 0.073621 -8.105 6.70e-16  
## f.extra0.5 0.074778 0.008583 8.713 < 2e-16  
## f.extra1 0.169251 0.010904 15.522 < 2e-16  
## f.code\_rate\_idRate-Other 0.807476 0.022679 35.604 < 2e-16  
## f.payment\_typeCash -0.114061 0.008248 -13.829 < 2e-16  
## f.payment\_typeNo paid -0.303472 0.047206 -6.429 1.42e-10  
## q.tip\_amount:f.espeed[10,20) 0.006449 0.005720 1.127 0.2597  
## q.tip\_amount:f.espeed[20,30) -0.001163 0.005747 -0.202 0.8397  
## q.tip\_amount:f.espeed[30,40) -0.009265 0.006152 -1.506 0.1322  
## q.tip\_amount:f.espeed[40,50) -0.027796 0.006883 -4.039 5.47e-05  
## q.tip\_amount:f.espeed[50,55] -0.039137 0.007461 -5.246 1.63e-07  
## q.tip\_amount:f.code\_rate\_idRate-Other 0.089331 0.007477 11.948 < 2e-16  
## log(q.tlenkm):f.espeed[10,20) -0.004650 0.015727 -0.296 0.7675  
## log(q.tlenkm):f.espeed[20,30) -0.009975 0.016075 -0.621 0.5349  
## log(q.tlenkm):f.espeed[30,40) 0.038537 0.017955 2.146 0.0319  
## log(q.tlenkm):f.espeed[40,50) 0.155447 0.028369 5.479 4.50e-08  
## log(q.tlenkm):f.espeed[50,55] 0.149001 0.032483 4.587 4.62e-06  
## log(q.tlenkm):f.extra0.5 -0.045898 0.006164 -7.446 1.14e-13  
## log(q.tlenkm):f.extra1 -0.063196 0.008455 -7.475 9.22e-14  
## log(q.tlenkm):f.code\_rate\_idRate-Other -0.483411 0.014131 -34.209 < 2e-16  
## log(q.tlenkm):f.payment\_typeCash 0.070128 0.006313 11.109 < 2e-16  
## log(q.tlenkm):f.payment\_typeNo paid 0.061644 0.030379 2.029 0.0425  
##   
## (Intercept) \*\*\*  
## q.tip\_amount \*\*\*  
## log(q.tlenkm) \*\*\*  
## f.paid\_tollsYes \*\*\*  
## f.espeed[10,20) \*\*\*  
## f.espeed[20,30) \*\*\*  
## f.espeed[30,40) \*\*\*  
## f.espeed[40,50) \*\*\*  
## f.espeed[50,55] \*\*\*  
## f.extra0.5 \*\*\*  
## f.extra1 \*\*\*  
## f.code\_rate\_idRate-Other \*\*\*  
## f.payment\_typeCash \*\*\*  
## f.payment\_typeNo paid \*\*\*  
## q.tip\_amount:f.espeed[10,20)   
## q.tip\_amount:f.espeed[20,30)   
## q.tip\_amount:f.espeed[30,40)   
## q.tip\_amount:f.espeed[40,50) \*\*\*  
## q.tip\_amount:f.espeed[50,55] \*\*\*  
## q.tip\_amount:f.code\_rate\_idRate-Other \*\*\*  
## log(q.tlenkm):f.espeed[10,20)   
## log(q.tlenkm):f.espeed[20,30)   
## log(q.tlenkm):f.espeed[30,40) \*   
## log(q.tlenkm):f.espeed[40,50) \*\*\*  
## log(q.tlenkm):f.espeed[50,55] \*\*\*  
## log(q.tlenkm):f.extra0.5 \*\*\*  
## log(q.tlenkm):f.extra1 \*\*\*  
## log(q.tlenkm):f.code\_rate\_idRate-Other \*\*\*  
## log(q.tlenkm):f.payment\_typeCash \*\*\*  
## log(q.tlenkm):f.payment\_typeNo paid \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1483 on 4572 degrees of freedom  
## Multiple R-squared: 0.9282, Adjusted R-squared: 0.9277   
## F-statistic: 2037 on 29 and 4572 DF, p-value: < 2.2e-16

# !!!! FALTA DIAGNOSI EXHAUSTIVA !!!

ll1<-Boxplot(rstudent(model\_17));ll1



## [1] 4051 80 2621 4269 1385 3802 2666 3676 3291 3634 3021 1978 2032 2005 2711  
## [16] 2434 3838 3287 2625 4524

sel2<-which(hatvalues(model\_17)>5\*length(model\_17$coefficients)/nrow(df));sel2;length(sel2)

## 1908 14314 23421 23932 23958 24990 28982 33046 37238 41478   
## 7 42 73 76 77 80 97 106 114 128   
## 49078 64149 71268 71596 81949 88821 98170 101184 110979 115296   
## 157 204 228 231 268 295 317 326 357 373   
## 121215 125894 128613 131915 132102 154087 166154 169380 194151 201926   
## 389 401 410 418 421 500 536 547 633 658   
## 204903 209928 210357 210707 221913 228729 244755 244971 252056 274645   
## 674 692 697 699 738 772 830 831 855 914   
## 300524 316484 322178 327762 329452 360250 382504 395415 404073 415806   
## 980 1024 1038 1053 1057 1150 1204 1247 1278 1319   
## 423307 423839 428613 437922 443592 449320 453619 486866 487457 492805   
## 1350 1354 1362 1385 1408 1427 1445 1557 1560 1574   
## 516357 529475 533937 535041 542034 559358 564751 572644 575739 577950   
## 1638 1682 1694 1696 1720 1774 1788 1802 1808 1816   
## 590161 620293 621420 621544 625503 632100 645141 654257 657624 658738   
## 1850 1954 1958 1960 1968 1990 2028 2065 2075 2080   
## 663694 683052 689000 710390 724424 725701 728096 730897 730975 731288   
## 2105 2183 2199 2266 2320 2324 2328 2334 2337 2339   
## 735280 741591 747830 751896 771658 773934 785532 793294 794902 810930   
## 2349 2378 2398 2412 2468 2475 2510 2528 2532 2572   
## 825427 826623 829742 861539 881540 892761 894658 896291 920461 957227   
## 2621 2625 2631 2723 2788 2837 2847 2853 2927 3035   
## 965349 976822 986459 986910 1010111 1010826 1040346 1051194 1060542 1076485   
## 3065 3105 3142 3147 3211 3215 3314 3353 3382 3421   
## 1082823 1083301 1095371 1109089 1110005 1120203 1120401 1140092 1150441 1159509   
## 3449 3453 3497 3535 3538 3566 3567 3633 3667 3696   
## 1181893 1227019 1227021 1233051 1242754 1254924 1261276 1287570 1334927 1340781   
## 3776 3902 3903 3919 3951 3988 4007 4089 4241 4260   
## 1342604 1345546 1347654 1354552 1354822 1356261 1377906 1396114 1407546 1419545   
## 4264 4269 4278 4299 4301 4305 4376 4449 4486 4524   
## 1421036 1439743   
## 4529 4585

## [1] 152

ll2<-which(row.names(model\_17) %in% names(hatvalues(model\_17)[sel2]));ll2

## integer(0)

sel3<-which(cooks.distance(model\_17)> 0.5 );sel3;length(sel3)

## named integer(0)

## [1] 0

ll3<-which(row.names(df) %in% names(cooks.distance(model\_17)[sel3]));ll3

## integer(0)

# Binary Logistics Regression

vars\_cexp <- vars\_cexp[c(1:4,6:10)]; vars\_cexp

## [1] "q.passenger\_count" "q.trip\_distance" "q.fare\_amount"   
## [4] "q.extra" "q.tolls\_amount" "q.hour"   
## [7] "q.tlenkm" "q.traveltime" "q.espeed"

table(df$target.tip\_is\_given, df$f.payment\_type)

##   
## Credit card Cash No paid  
## No 352 2484 29  
## Yes 1737 0 0

We can see from the table that it is no credible the fact that any of the people that paid in cash did not leave any tip.

res.cat <- catdes(df, num.var = which(names(df)=="target.tip\_is\_given"))  
res.cat$quanti.var

## Eta2 P-value  
## q.tip\_amount 0.530313236 0.000000e+00  
## target.total\_amount 0.062475234 1.704519e-66  
## q.dropoff\_longitude 0.045623769 1.241947e-48  
## q.pickup\_longitude 0.035898477 1.874433e-38  
## q.fare\_amount 0.014755168 1.353812e-16  
## q.trip\_distance 0.012901088 1.091013e-14  
## q.tlenkm 0.012500007 2.820041e-14  
## q.dropoff\_latitude 0.011813680 1.432540e-13  
## q.pickup\_latitude 0.010850411 1.403276e-12  
## q.traveltime 0.009292813 5.638316e-11  
## q.espeed 0.007947848 1.376257e-09  
## q.tolls\_amount 0.004085851 1.427990e-05

res.cat$test.chi2

## p.value df  
## f.payment\_type 0.000000e+00 2  
## f.cost 1.855099e-93 5  
## f.dist 3.632199e-23 3  
## f.trip\_distance\_range 2.119770e-22 2  
## f.tt 7.339353e-14 4  
## f.espeed 1.128783e-08 5  
## f.paid\_tolls 2.595115e-06 1  
## qual.pickup 5.563582e-05 23  
## f.period 6.473080e-05 3  
## f.mta\_tax 8.160062e-05 1  
## f.improvement\_surcharge 1.041592e-04 1  
## f.trip\_type 1.182591e-04 1  
## qual.dropoff 3.987953e-04 23  
## f.code\_rate\_id 5.237279e-04 1  
## f.hour 4.399605e-02 6

From the quanti.var we can see that tip\_is\_given depends on tip\_amount which seems obvious, due to the fact that they are the same variable treated in different ways.

From the test.chi2 we can see that payment\_type has something really clear with the tip\_is\_given, as we have p-value of 0. Which means that we cannot use payment\_type as a predictor.

## Filter

ll<-which(df$f.payment\_type=="Cash"); length(ll)

## [1] 2484

dff<-df[-ll,]  
set.seed(12345)  
llwork<-sample(1:nrow(dff),0.70\*nrow(dff),replace=FALSE)  
llwork<-sort(llwork);length(llwork)

## [1] 1482

dffwork<-dff[llwork,]  
dfftest<-dff[-llwork,]

# maybe no cal posar això de steps to follow????

Steps to follow:

1. Enter all relevant numerical variables in the model
2. See if you need to replace a number with its equivalent factor
3. Add to the best model of step 2, the main effects of the factors and retain the significant net effects.
4. Add interactions: between factor-factor and between factor-numeric (doubles).
5. Diagnosis of waste and observations. Lack of adjustment and / or influential.

## Numerical variables // Enter all relevant numerical variables in the model

### Model 20

model\_20 <- glm(  
 target.tip\_is\_given~.  
 ,family = "binomial"  
 ,data=dffwork[,c("target.tip\_is\_given",vars\_cexp)]  
);summary(model\_20)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ ., family = "binomial", data = dffwork[,   
## c("target.tip\_is\_given", vars\_cexp)])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1696 0.5349 0.6141 0.6584 1.0045   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.789176 0.338897 2.329 0.0199 \*  
## q.passenger\_count 0.087787 0.073100 1.201 0.2298   
## q.trip\_distance -0.129272 0.217070 -0.596 0.5515   
## q.fare\_amount 0.003783 0.026264 0.144 0.8855   
## q.extra -0.020544 0.196999 -0.104 0.9169   
## q.tolls\_amount 0.066491 0.141704 0.469 0.6389   
## q.hour 0.017466 0.010258 1.703 0.0886 .  
## q.tlenkm 0.083903 0.131675 0.637 0.5240   
## q.traveltime 0.010833 0.015944 0.679 0.4969   
## q.espeed 0.008365 0.013213 0.633 0.5267   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1386.6 on 1472 degrees of freedom  
## AIC: 1406.6  
##   
## Number of Fisher Scoring iterations: 4

Anova(model\_20, test="Wald") #binary target

## Analysis of Deviance Table (Type II tests)  
##   
## Response: target.tip\_is\_given  
## Df Chisq Pr(>Chisq)   
## q.passenger\_count 1 1.4422 0.22978   
## q.trip\_distance 1 0.3547 0.55149   
## q.fare\_amount 1 0.0207 0.88548   
## q.extra 1 0.0109 0.91694   
## q.tolls\_amount 1 0.2202 0.63891   
## q.hour 1 2.8990 0.08863 .  
## q.tlenkm 1 0.4060 0.52400   
## q.traveltime 1 0.4617 0.49685   
## q.espeed 1 0.4008 0.52667   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Explicació output del summary: --> s'haurà d'esborrar!!!!  
#   
# \* Deviance Residuals: diferència entre prediccions i observacions.  
# \* Coeficients: com abans  
# \* (Dispersion parameter for binomial family taken to be 1): ni cas  
# \* Residual deviance: suma de quadrats residuals

Comments:

* We can see that the most influent variable, in our case, is the q.hour.
* We can see that the residual deviance is of 1386.6 on 1472 degrees of freedom.

vif(model\_20)

## q.passenger\_count q.trip\_distance q.fare\_amount q.extra   
## 1.009767 66.673198 9.481893 1.104293   
## q.tolls\_amount q.hour q.tlenkm q.traveltime   
## 1.050135 1.098553 63.087992 5.163194   
## q.espeed   
## 2.918476

We can see that we have some variables with very high vifs:

* q.trip\_distance (66.67)
* q.tlenkm (63.09) –> correlated with the previous
* q.fare\_amount (9.48)
* q.traveltime (5.16)

### Model 21

*NOTE: we are aware that we should not have factors in this section, but we have decided to include them due to the fact that we overwrote their numeric values and created their factors in the previous deliverables.*

We know there is not colinearity, so we create a new model:

model\_21 <- glm(  
 target.tip\_is\_given~  
 f.improvement\_surcharge+  
 f.mta\_tax+  
 q.passenger\_count+  
 q.extra+  
 q.tolls\_amount+  
 q.hour+  
 q.espeed+  
 q.tlenkm+  
 q.traveltime   
 ,family = "binomial"  
 ,data=dffwork  
);summary(model\_21)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ f.improvement\_surcharge +   
## f.mta\_tax + q.passenger\_count + q.extra + q.tolls\_amount +   
## q.hour + q.espeed + q.tlenkm + q.traveltime, family = "binomial",   
## data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1925 0.5236 0.6089 0.6505 1.3166   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.679210 0.521184 -1.303 0.1925   
## f.improvement\_surchargeYes 0.746400 1.553187 0.481 0.6308   
## f.mta\_taxYes 0.855221 1.551655 0.551 0.5815   
## q.passenger\_count 0.102739 0.074620 1.377 0.1686   
## q.extra -0.113608 0.198116 -0.573 0.5663   
## q.tolls\_amount 0.060750 0.141781 0.428 0.6683   
## q.hour 0.016996 0.010274 1.654 0.0981 .  
## q.espeed 0.006003 0.013134 0.457 0.6476   
## q.tlenkm 0.021254 0.040504 0.525 0.5998   
## q.traveltime 0.005897 0.013937 0.423 0.6722   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1374.2 on 1472 degrees of freedom  
## AIC: 1394.2  
##   
## Number of Fisher Scoring iterations: 4

vif(model\_21)

## f.improvement\_surcharge f.mta\_tax q.passenger\_count   
## 13.752586 13.725474 1.011409   
## q.extra q.tolls\_amount q.hour   
## 1.118068 1.034653 1.095075   
## q.espeed q.tlenkm q.traveltime   
## 2.831254 5.779818 4.048661

Anova(model\_21, test="Wald") #binary target

## Analysis of Deviance Table (Type II tests)  
##   
## Response: target.tip\_is\_given  
## Df Chisq Pr(>Chisq)   
## f.improvement\_surcharge 1 0.2309 0.63083   
## f.mta\_tax 1 0.3038 0.58152   
## q.passenger\_count 1 1.8957 0.16856   
## q.extra 1 0.3288 0.56634   
## q.tolls\_amount 1 0.1836 0.66830   
## q.hour 1 2.7366 0.09807 .  
## q.espeed 1 0.2089 0.64762   
## q.tlenkm 1 0.2753 0.59977   
## q.traveltime 1 0.1790 0.67220   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

anova(model\_21, model\_20, test="Chisq") # only for nested models

## Analysis of Deviance Table  
##   
## Model 1: target.tip\_is\_given ~ f.improvement\_surcharge + f.mta\_tax + q.passenger\_count +   
## q.extra + q.tolls\_amount + q.hour + q.espeed + q.tlenkm +   
## q.traveltime  
## Model 2: target.tip\_is\_given ~ q.passenger\_count + q.trip\_distance + q.fare\_amount +   
## q.extra + q.tolls\_amount + q.hour + q.tlenkm + q.traveltime +   
## q.espeed  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)  
## 1 1472 1374.2   
## 2 1472 1386.6 0 -12.44

We can transform tlenkm and remove improvement\_surcharge in order to have lower vifs:

### Model 22

model\_22 <- glm(  
 target.tip\_is\_given~  
 f.mta\_tax+  
 q.passenger\_count+  
 q.extra+  
 q.tolls\_amount+  
 q.hour+  
 q.espeed+  
 poly(q.tlenkm,2)+  
 q.traveltime   
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_22)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ f.mta\_tax + q.passenger\_count +   
## q.extra + q.tolls\_amount + q.hour + q.espeed + poly(q.tlenkm,   
## 2) + q.traveltime, family = "binomial", data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2454 0.5035 0.6010 0.6581 1.3451   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.063438 0.630329 -0.101 0.919834   
## f.mta\_taxYes 1.598945 0.430147 3.717 0.000201 \*\*\*  
## q.passenger\_count 0.103937 0.074592 1.393 0.163500   
## q.extra -0.100561 0.197968 -0.508 0.611478   
## q.tolls\_amount 0.056478 0.142674 0.396 0.692213   
## q.hour 0.016300 0.010315 1.580 0.114045   
## q.espeed -0.006787 0.013738 -0.494 0.621311   
## poly(q.tlenkm, 2)1 11.175853 7.164996 1.560 0.118811   
## poly(q.tlenkm, 2)2 -6.647205 2.778483 -2.392 0.016739 \*   
## q.traveltime -0.010694 0.014623 -0.731 0.464568   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1369.3 on 1472 degrees of freedom  
## AIC: 1389.3  
##   
## Number of Fisher Scoring iterations: 4

vif(model\_22)

## GVIF Df GVIF^(1/(2\*Df))  
## f.mta\_tax 1.044554 1 1.022034  
## q.passenger\_count 1.011298 1 1.005633  
## q.extra 1.112981 1 1.054979  
## q.tolls\_amount 1.034816 1 1.017259  
## q.hour 1.098904 1 1.048286  
## q.espeed 3.215503 1 1.793182  
## poly(q.tlenkm, 2) 6.953595 2 1.623874  
## q.traveltime 4.814589 1 2.194217

anova(model\_21, model\_22, test="Chisq") # only for nested models

## Analysis of Deviance Table  
##   
## Model 1: target.tip\_is\_given ~ f.improvement\_surcharge + f.mta\_tax + q.passenger\_count +   
## q.extra + q.tolls\_amount + q.hour + q.espeed + q.tlenkm +   
## q.traveltime  
## Model 2: target.tip\_is\_given ~ f.mta\_tax + q.passenger\_count + q.extra +   
## q.tolls\_amount + q.hour + q.espeed + poly(q.tlenkm, 2) +   
## q.traveltime  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)  
## 1 1472 1374.2   
## 2 1472 1369.3 0 4.88

Anova(model\_22, test="Wald") #binary target

## Analysis of Deviance Table (Type II tests)  
##   
## Response: target.tip\_is\_given  
## Df Chisq Pr(>Chisq)   
## f.mta\_tax 1 13.8176 0.0002014 \*\*\*  
## q.passenger\_count 1 1.9416 0.1634996   
## q.extra 1 0.2580 0.6114779   
## q.tolls\_amount 1 0.1567 0.6922126   
## q.hour 1 2.4973 0.1140452   
## q.espeed 1 0.2440 0.6213106   
## poly(q.tlenkm, 2) 2 5.8276 0.0542687 .   
## q.traveltime 1 0.5349 0.4645682   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Now, we can do a step:

### Model 23

model\_23 <- step(model\_22, k=log(nrow(dffwork)))

## Start: AIC=1442.33  
## target.tip\_is\_given ~ f.mta\_tax + q.passenger\_count + q.extra +   
## q.tolls\_amount + q.hour + q.espeed + poly(q.tlenkm, 2) +   
## q.traveltime  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2) 2 1374.7 1433.1  
## - q.tolls\_amount 1 1369.5 1435.2  
## - q.espeed 1 1369.6 1435.3  
## - q.extra 1 1369.6 1435.3  
## - q.traveltime 1 1369.8 1435.5  
## - q.passenger\_count 1 1371.4 1437.1  
## - q.hour 1 1371.8 1437.5  
## <none> 1369.3 1442.3  
## - f.mta\_tax 1 1382.2 1447.9  
##   
## Step: AIC=1433.12  
## target.tip\_is\_given ~ f.mta\_tax + q.passenger\_count + q.extra +   
## q.tolls\_amount + q.hour + q.espeed + q.traveltime  
##   
## Df Deviance AIC  
## - q.tolls\_amount 1 1375.0 1426.1  
## - q.extra 1 1375.0 1426.1  
## - q.passenger\_count 1 1376.7 1427.8  
## - q.espeed 1 1376.9 1428.0  
## - q.hour 1 1377.4 1428.5  
## - q.traveltime 1 1378.0 1429.1  
## <none> 1374.7 1433.1  
## - f.mta\_tax 1 1387.0 1438.2  
##   
## Step: AIC=1426.1  
## target.tip\_is\_given ~ f.mta\_tax + q.passenger\_count + q.extra +   
## q.hour + q.espeed + q.traveltime  
##   
## Df Deviance AIC  
## - q.extra 1 1375.3 1419.1  
## - q.passenger\_count 1 1377.0 1420.8  
## - q.espeed 1 1377.4 1421.2  
## - q.hour 1 1377.6 1421.4  
## - q.traveltime 1 1378.5 1422.3  
## <none> 1375.0 1426.1  
## - f.mta\_tax 1 1387.3 1431.2  
##   
## Step: AIC=1419.12  
## target.tip\_is\_given ~ f.mta\_tax + q.passenger\_count + q.hour +   
## q.espeed + q.traveltime  
##   
## Df Deviance AIC  
## - q.passenger\_count 1 1377.2 1413.8  
## - q.hour 1 1377.7 1414.2  
## - q.espeed 1 1377.8 1414.3  
## - q.traveltime 1 1378.9 1415.4  
## <none> 1375.3 1419.1  
## - f.mta\_tax 1 1387.3 1423.9  
##   
## Step: AIC=1413.76  
## target.tip\_is\_given ~ f.mta\_tax + q.hour + q.espeed + q.traveltime  
##   
## Df Deviance AIC  
## - q.espeed 1 1379.8 1409.0  
## - q.hour 1 1379.8 1409.0  
## - q.traveltime 1 1380.8 1410.0  
## <none> 1377.2 1413.8  
## - f.mta\_tax 1 1388.9 1418.1  
##   
## Step: AIC=1408.99  
## target.tip\_is\_given ~ f.mta\_tax + q.hour + q.traveltime  
##   
## Df Deviance AIC  
## - q.hour 1 1381.8 1403.7  
## - q.traveltime 1 1383.3 1405.2  
## <none> 1379.8 1409.0  
## - f.mta\_tax 1 1391.0 1412.9  
##   
## Step: AIC=1403.71  
## target.tip\_is\_given ~ f.mta\_tax + q.traveltime  
##   
## Df Deviance AIC  
## - q.traveltime 1 1385.0 1399.6  
## <none> 1381.8 1403.7  
## - f.mta\_tax 1 1393.6 1408.2  
##   
## Step: AIC=1399.63  
## target.tip\_is\_given ~ f.mta\_tax  
##   
## Df Deviance AIC  
## <none> 1385.0 1399.6  
## - f.mta\_tax 1 1397.9 1405.2

summary(model\_23)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ f.mta\_tax, family = "binomial",   
## data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8674 0.6201 0.6201 0.6201 1.1774   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -8.815e-14 4.082e-01 0.000 1.000000   
## f.mta\_taxYes 1.551e+00 4.140e-01 3.747 0.000179 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1385.0 on 1480 degrees of freedom  
## AIC: 1389  
##   
## Number of Fisher Scoring iterations: 4

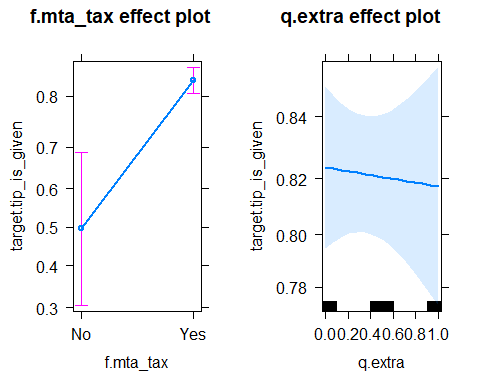
Due to the fact that this model is really ppor, we will take also the q.extra variable in order to be able to extract more information. For instance, we could do the marginal plots:

model\_23 <- glm(  
 target.tip\_is\_given~  
 f.mta\_tax+  
 q.extra  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_23)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ f.mta\_tax + q.extra, family = "binomial",   
## data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.8743 0.6157 0.6157 0.6218 1.1863   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.0009104 0.4082683 0.002 0.998221   
## f.mta\_taxYes 1.5660824 0.4190015 3.738 0.000186 \*\*\*  
## q.extra -0.0437003 0.1893364 -0.231 0.817464   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1385.0 on 1479 degrees of freedom  
## AIC: 1391  
##   
## Number of Fisher Scoring iterations: 4

### Understanding the model

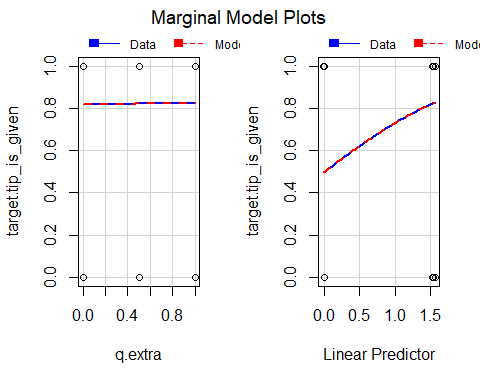
plot(allEffects(model\_23))



* For the f.mta\_tax: we that if the value of the variable is “Yes”, it is more probable that the target.tip\_is\_given value will be “Yes” as well.
* For the q.extra: as we have said before, this variable does not really affect to the target, but we will include it in order to be able to do more plots. At most, we could say that it is inversely proportional to the target.

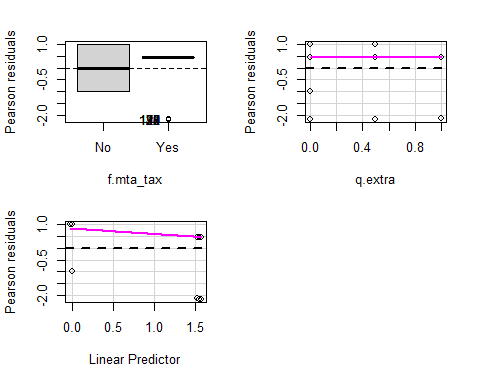
marginalModelPlots(model\_23)

## Warning in mmps(...): Interactions and/or factors skipped



We can observe that q.extra is a candidate to be a factor.

residualPlots(model\_23)



## Test stat Pr(>|Test stat|)  
## f.mta\_tax   
## q.extra 0.3308 0.5652

We see that the smoothers are relatively plain, so we could say that, for now, everything is ok.

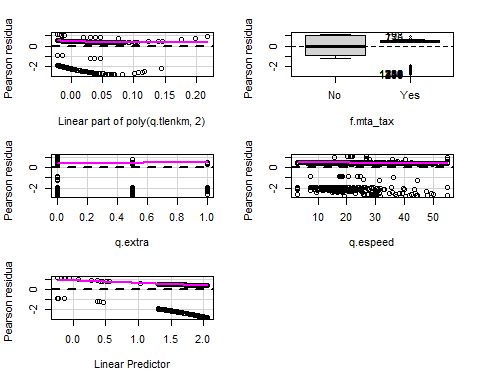
We are going, though, to propose a model which brings us more chances:

### Model 24

model\_24 <- glm(  
 target.tip\_is\_given~  
 poly(q.tlenkm, 2)+  
 f.mta\_tax+  
 q.extra+  
 q.espeed  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_24)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax +   
## q.extra + q.espeed, family = "binomial", data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0830 0.5183 0.6029 0.6620 1.2773   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.0262246 0.4739302 0.055 0.955872   
## poly(q.tlenkm, 2)1 6.4381003 3.3220235 1.938 0.052623 .   
## poly(q.tlenkm, 2)2 -5.6266502 2.3548152 -2.389 0.016875 \*   
## f.mta\_taxYes 1.5402952 0.4245567 3.628 0.000286 \*\*\*  
## q.extra 0.0058437 0.1901872 0.031 0.975488   
## q.espeed -0.0001024 0.0093700 -0.011 0.991278   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1374.8 on 1476 degrees of freedom  
## AIC: 1386.8  
##   
## Number of Fisher Scoring iterations: 4

residualPlots(model\_24)



## Test stat Pr(>|Test stat|)  
## poly(q.tlenkm, 2)   
## f.mta\_tax   
## q.extra 0.1797 0.6717  
## q.espeed 1.8612 0.1725

* q.tlenkm:
  + we see that the smoothe is plain, so it is ok.
  + the “weird” shapes that appear are because of the binery response model.
* q.extra:
  + we observe that the smoother is plain, so it is ok.
* q.espeed:
  + we see that the smoother is plain, so it is ok.
  + the “weird” shapes that appear are because of the binery response model.
* the whole model:
  + we see that the smoothe is not completely straight, but as it was said in class, we can work with unfitted values in the model, due to the fact that it is a really dense topic.

Anova(model\_24, test="Wald")

## Analysis of Deviance Table (Type II tests)  
##   
## Response: target.tip\_is\_given  
## Df Chisq Pr(>Chisq)   
## poly(q.tlenkm, 2) 2 8.1026 0.0173996 \*   
## f.mta\_tax 1 13.1624 0.0002856 \*\*\*  
## q.extra 1 0.0009 0.9754881   
## q.espeed 1 0.0001 0.9912776   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We have to ensure that we do not have any variable with a non significant net effect.

Thus, we are going to redp the model:

model\_24 <- glm(  
 target.tip\_is\_given~  
 poly(q.tlenkm, 2)+  
 f.mta\_tax  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_24)

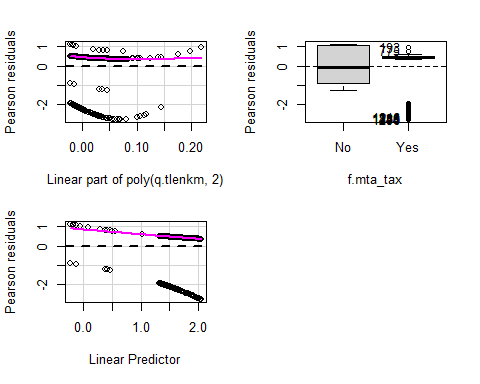
##   
## Call:  
## glm(formula = target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax,   
## family = "binomial", data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0825 0.5184 0.6030 0.6617 1.2771   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.02379 0.41272 0.058 0.954042   
## poly(q.tlenkm, 2)1 6.40974 2.72195 2.355 0.018531 \*   
## poly(q.tlenkm, 2)2 -5.62235 2.34340 -2.399 0.016430 \*   
## f.mta\_taxYes 1.54264 0.41841 3.687 0.000227 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1374.8 on 1478 degrees of freedom  
## AIC: 1382.8  
##   
## Number of Fisher Scoring iterations: 4

vif(model\_24)

## GVIF Df GVIF^(1/(2\*Df))  
## poly(q.tlenkm, 2) 1.000229 2 1.000057  
## f.mta\_tax 1.000229 1 1.000115

residualPlots(model\_24)

## Warning in residualPlots.default(model, ...): No possible lack-of-fit tests



Anova(model\_24, test="Wald")

## Analysis of Deviance Table (Type II tests)  
##   
## Response: target.tip\_is\_given  
## Df Chisq Pr(>Chisq)   
## poly(q.tlenkm, 2) 2 9.8765 0.007167 \*\*   
## f.mta\_tax 1 13.5936 0.000227 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

With Anova(model\_24), we see that it is fulfilled.

## Factors //See if you need to replace a number with its equivalent factor

We look if any of the numeric variables can be substituted by a factor.

The first thing we will do, it would be change the “q.mta\_tax” (if it existed in our dataset) for a “f.mta\_tax”. Due to the fact that mta\_tax is already a factor, we do not need to do this step.

Given that the other variable that could be a factor depends on a polynomial, we keep as it is. The code that should be done in case of a new model with an added factor, would be the following:

model\_25 <- glm(  
 target.tip\_is\_given~  
 poly(q.tlenkm, 2)+  
 f.mta\_tax  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_25)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax,   
## family = "binomial", data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0825 0.5184 0.6030 0.6617 1.2771   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.02379 0.41272 0.058 0.954042   
## poly(q.tlenkm, 2)1 6.40974 2.72195 2.355 0.018531 \*   
## poly(q.tlenkm, 2)2 -5.62235 2.34340 -2.399 0.016430 \*   
## f.mta\_taxYes 1.54264 0.41841 3.687 0.000227 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1374.8 on 1478 degrees of freedom  
## AIC: 1382.8  
##   
## Number of Fisher Scoring iterations: 4

BIC(model\_24, model\_25) # same model --> same bic

## df BIC  
## model\_24 4 1404.024  
## model\_25 4 1404.024

Thanks to the BIC(model\_24, model\_25) we could see te changes generated by the new model. The less the BIC is, the better the model will be. We need to remember that, in case of have done an exchange of a numeric variable to a factor, we could not have done it with an anova test, due to the fact that there is an exchange, which means that any model is bigger than the other.

## Decision //Add to the best model of step 2, the main effects of the factors and retain the significant net effects.

We decide to keep with the model\_25.

## Interactions //Add interactions: between factor-factor and between factor-numeric (doubles).

### factor-numeric

Now that we have a defined model, we are going to do some interactions with all of the factor variables we think are the relevant:

model\_26 <- glm(  
 target.tip\_is\_given~  
 ( poly(q.tlenkm, 2) ) \*   
 (  
 f.mta\_tax+  
 f.vendor\_id+  
 f.period+  
 f.espeed+  
 f.paid\_tolls+  
 f.tt+  
 f.extra  
 )  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_26)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ (poly(q.tlenkm, 2)) \* (f.mta\_tax +   
## f.vendor\_id + f.period + f.espeed + f.paid\_tolls + f.tt +   
## f.extra), family = "binomial", data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.5111 0.3976 0.5667 0.6588 1.5477   
##   
## Coefficients:  
## Estimate Std. Error z value  
## (Intercept) -0.82643 0.80291 -1.029  
## poly(q.tlenkm, 2)1 -48.86181 62.37786 -0.783  
## poly(q.tlenkm, 2)2 -67.70393 78.80136 -0.859  
## f.mta\_taxYes 1.33040 0.60053 2.215  
## f.vendor\_idf.Vendor-VeriFone 0.17117 0.17610 0.972  
## f.periodPeriod morning 0.05307 0.32187 0.165  
## f.periodPeriod valley -0.26682 0.27911 -0.956  
## f.periodPeriod afternoon -0.04948 0.28335 -0.175  
## f.espeed[10,20) 0.56001 0.41266 1.357  
## f.espeed[20,30) 0.81065 0.42368 1.913  
## f.espeed[30,40) 0.59718 0.52679 1.134  
## f.espeed[40,50) 0.39674 0.96765 0.410  
## f.espeed[50,55] 2.05026 1.36596 1.501  
## f.paid\_tollsYes 10.08756 8.98297 1.123  
## f.tt(15,20] 0.49946 0.63838 0.782  
## f.tt(20,60] 0.40590 0.55999 0.725  
## f.tt(5,10] 0.31296 0.62378 0.502  
## f.tt[0,5] -1.93019 1.71453 -1.126  
## f.extra0.5 -0.10017 0.25689 -0.390  
## f.extra1 0.25909 0.44227 0.586  
## poly(q.tlenkm, 2)1:f.mta\_taxYes -3.52444 36.74099 -0.096  
## poly(q.tlenkm, 2)2:f.mta\_taxYes 28.00984 56.28102 0.498  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone 3.63680 10.97264 0.331  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone 15.18241 13.02671 1.165  
## poly(q.tlenkm, 2)1:f.periodPeriod morning -22.28585 17.97438 -1.240  
## poly(q.tlenkm, 2)2:f.periodPeriod morning -44.45093 21.87311 -2.032  
## poly(q.tlenkm, 2)1:f.periodPeriod valley -10.77194 16.27258 -0.662  
## poly(q.tlenkm, 2)2:f.periodPeriod valley -25.05193 18.87027 -1.328  
## poly(q.tlenkm, 2)1:f.periodPeriod afternoon -9.27239 26.77483 -0.346  
## poly(q.tlenkm, 2)2:f.periodPeriod afternoon -0.89669 29.58146 -0.030  
## poly(q.tlenkm, 2)1:f.espeed[10,20) 25.42995 32.91696 0.773  
## poly(q.tlenkm, 2)2:f.espeed[10,20) -7.43103 34.55887 -0.215  
## poly(q.tlenkm, 2)1:f.espeed[20,30) 15.39756 24.99437 0.616  
## poly(q.tlenkm, 2)2:f.espeed[20,30) -17.71186 30.24786 -0.586  
## poly(q.tlenkm, 2)1:f.espeed[30,40) 1.38881 19.86645 0.070  
## poly(q.tlenkm, 2)2:f.espeed[30,40) -14.60762 23.96168 -0.610  
## poly(q.tlenkm, 2)1:f.espeed[40,50) -7.30645 23.72922 -0.308  
## poly(q.tlenkm, 2)2:f.espeed[40,50) -38.16754 29.48562 -1.294  
## poly(q.tlenkm, 2)1:f.espeed[50,55] -5.42983 33.43054 -0.162  
## poly(q.tlenkm, 2)2:f.espeed[50,55] -23.44641 32.10124 -0.730  
## poly(q.tlenkm, 2)1:f.paid\_tollsYes -119.22646 106.69711 -1.117  
## poly(q.tlenkm, 2)2:f.paid\_tollsYes 259.28772 211.00439 1.229  
## poly(q.tlenkm, 2)1:f.tt(15,20] 92.19663 53.37765 1.727  
## poly(q.tlenkm, 2)2:f.tt(15,20] 72.20211 67.51906 1.069  
## poly(q.tlenkm, 2)1:f.tt(20,60] 59.57123 45.79488 1.301  
## poly(q.tlenkm, 2)2:f.tt(20,60] 61.60412 51.24213 1.202  
## poly(q.tlenkm, 2)1:f.tt(5,10] 26.71046 72.61932 0.368  
## poly(q.tlenkm, 2)2:f.tt(5,10] 60.78930 69.64260 0.873  
## poly(q.tlenkm, 2)1:f.tt[0,5] -31.19286 207.76033 -0.150  
## poly(q.tlenkm, 2)2:f.tt[0,5] 92.47473 161.97721 0.571  
## poly(q.tlenkm, 2)1:f.extra0.5 -3.61736 14.76853 -0.245  
## poly(q.tlenkm, 2)2:f.extra0.5 -14.55494 15.56608 -0.935  
## poly(q.tlenkm, 2)1:f.extra1 34.64810 52.63775 0.658  
## poly(q.tlenkm, 2)2:f.extra1 -1.18737 48.91063 -0.024  
## Pr(>|z|)   
## (Intercept) 0.3033   
## poly(q.tlenkm, 2)1 0.4334   
## poly(q.tlenkm, 2)2 0.3902   
## f.mta\_taxYes 0.0267 \*  
## f.vendor\_idf.Vendor-VeriFone 0.3311   
## f.periodPeriod morning 0.8690   
## f.periodPeriod valley 0.3391   
## f.periodPeriod afternoon 0.8614   
## f.espeed[10,20) 0.1748   
## f.espeed[20,30) 0.0557 .  
## f.espeed[30,40) 0.2570   
## f.espeed[40,50) 0.6818   
## f.espeed[50,55] 0.1334   
## f.paid\_tollsYes 0.2615   
## f.tt(15,20] 0.4340   
## f.tt(20,60] 0.4686   
## f.tt(5,10] 0.6159   
## f.tt[0,5] 0.2603   
## f.extra0.5 0.6966   
## f.extra1 0.5580   
## poly(q.tlenkm, 2)1:f.mta\_taxYes 0.9236   
## poly(q.tlenkm, 2)2:f.mta\_taxYes 0.6187   
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone 0.7403   
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone 0.2438   
## poly(q.tlenkm, 2)1:f.periodPeriod morning 0.2150   
## poly(q.tlenkm, 2)2:f.periodPeriod morning 0.0421 \*  
## poly(q.tlenkm, 2)1:f.periodPeriod valley 0.5080   
## poly(q.tlenkm, 2)2:f.periodPeriod valley 0.1843   
## poly(q.tlenkm, 2)1:f.periodPeriod afternoon 0.7291   
## poly(q.tlenkm, 2)2:f.periodPeriod afternoon 0.9758   
## poly(q.tlenkm, 2)1:f.espeed[10,20) 0.4398   
## poly(q.tlenkm, 2)2:f.espeed[10,20) 0.8297   
## poly(q.tlenkm, 2)1:f.espeed[20,30) 0.5379   
## poly(q.tlenkm, 2)2:f.espeed[20,30) 0.5582   
## poly(q.tlenkm, 2)1:f.espeed[30,40) 0.9443   
## poly(q.tlenkm, 2)2:f.espeed[30,40) 0.5421   
## poly(q.tlenkm, 2)1:f.espeed[40,50) 0.7582   
## poly(q.tlenkm, 2)2:f.espeed[40,50) 0.1955   
## poly(q.tlenkm, 2)1:f.espeed[50,55] 0.8710   
## poly(q.tlenkm, 2)2:f.espeed[50,55] 0.4652   
## poly(q.tlenkm, 2)1:f.paid\_tollsYes 0.2638   
## poly(q.tlenkm, 2)2:f.paid\_tollsYes 0.2191   
## poly(q.tlenkm, 2)1:f.tt(15,20] 0.0841 .  
## poly(q.tlenkm, 2)2:f.tt(15,20] 0.2849   
## poly(q.tlenkm, 2)1:f.tt(20,60] 0.1933   
## poly(q.tlenkm, 2)2:f.tt(20,60] 0.2293   
## poly(q.tlenkm, 2)1:f.tt(5,10] 0.7130   
## poly(q.tlenkm, 2)2:f.tt(5,10] 0.3827   
## poly(q.tlenkm, 2)1:f.tt[0,5] 0.8807   
## poly(q.tlenkm, 2)2:f.tt[0,5] 0.5681   
## poly(q.tlenkm, 2)1:f.extra0.5 0.8065   
## poly(q.tlenkm, 2)2:f.extra0.5 0.3498   
## poly(q.tlenkm, 2)1:f.extra1 0.5104   
## poly(q.tlenkm, 2)2:f.extra1 0.9806   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1312.2 on 1428 degrees of freedom  
## AIC: 1420.2  
##   
## Number of Fisher Scoring iterations: 9

Anova(model\_26, test="Wald")

## Analysis of Deviance Table (Type II tests)  
##   
## Response: target.tip\_is\_given  
## Df Chisq Pr(>Chisq)   
## poly(q.tlenkm, 2) 2 1.9772 0.372099   
## f.mta\_tax 1 9.5162 0.002037 \*\*  
## f.vendor\_id 1 1.1244 0.288982   
## f.period 3 2.4845 0.478101   
## f.espeed 5 7.7073 0.173123   
## f.paid\_tolls 1 1.1206 0.289779   
## f.tt 4 4.4872 0.344066   
## f.extra 2 0.1760 0.915771   
## poly(q.tlenkm, 2):f.mta\_tax 2 1.1016 0.576479   
## poly(q.tlenkm, 2):f.vendor\_id 2 1.5547 0.459625   
## poly(q.tlenkm, 2):f.period 6 6.0681 0.415607   
## poly(q.tlenkm, 2):f.espeed 10 7.8326 0.645181   
## poly(q.tlenkm, 2):f.paid\_tolls 2 1.5643 0.457411   
## poly(q.tlenkm, 2):f.tt 8 13.1314 0.107408   
## poly(q.tlenkm, 2):f.extra 4 3.7993 0.433848   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We remove the non significant variables:

model\_27 <- step(model\_26, k=log(nrow(dffwork)))

## Start: AIC=1706.48  
## target.tip\_is\_given ~ (poly(q.tlenkm, 2)) \* (f.mta\_tax + f.vendor\_id +   
## f.period + f.espeed + f.paid\_tolls + f.tt + f.extra)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.espeed 10 1321.2 1642.5  
## - poly(q.tlenkm, 2):f.tt 8 1331.7 1667.5  
## - poly(q.tlenkm, 2):f.period 6 1318.5 1668.9  
## - poly(q.tlenkm, 2):f.extra 4 1316.4 1681.5  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1313.4 1693.1  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1313.8 1693.4  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1316.5 1696.1  
## <none> 1312.2 1706.5

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1642.48  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.period + f.espeed + f.paid\_tolls + f.tt + f.extra + poly(q.tlenkm,   
## 2):f.mta\_tax + poly(q.tlenkm, 2):f.vendor\_id + poly(q.tlenkm,   
## 2):f.period + poly(q.tlenkm, 2):f.paid\_tolls + poly(q.tlenkm,   
## 2):f.tt + poly(q.tlenkm, 2):f.extra

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.period 6 1327.9 1605.4  
## - poly(q.tlenkm, 2):f.tt 8 1342.6 1605.5  
## - f.espeed 5 1330.1 1614.8  
## - poly(q.tlenkm, 2):f.extra 4 1326.3 1618.3  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1322.7 1629.4  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1323.0 1629.6  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1325.2 1631.8  
## <none> 1321.2 1642.5

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1605.38  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.period + f.espeed + f.paid\_tolls + f.tt + f.extra + poly(q.tlenkm,   
## 2):f.mta\_tax + poly(q.tlenkm, 2):f.vendor\_id + poly(q.tlenkm,   
## 2):f.paid\_tolls + poly(q.tlenkm, 2):f.tt + poly(q.tlenkm,   
## 2):f.extra

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.tt 8 1347.7 1566.8  
## - f.espeed 5 1336.2 1577.1  
## - poly(q.tlenkm, 2):f.extra 4 1331.7 1579.9  
## - f.period 3 1330.6 1586.2  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1328.8 1591.6  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1329.5 1592.3  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1332.4 1595.2  
## <none> 1327.9 1605.4  
##   
## Step: AIC=1566.78  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.period + f.espeed + f.paid\_tolls + f.tt + f.extra + poly(q.tlenkm,   
## 2):f.mta\_tax + poly(q.tlenkm, 2):f.vendor\_id + poly(q.tlenkm,   
## 2):f.paid\_tolls + poly(q.tlenkm, 2):f.extra

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - f.espeed 5 1354.2 1536.7  
## - poly(q.tlenkm, 2):f.extra 4 1352.2 1542.0  
## - f.tt 4 1353.7 1543.5  
## - f.period 3 1350.3 1547.5  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1348.0 1552.5  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1348.9 1553.3  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1351.4 1555.8  
## <none> 1347.7 1566.8

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1536.73  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.period + f.paid\_tolls + f.tt + f.extra + poly(q.tlenkm,   
## 2):f.mta\_tax + poly(q.tlenkm, 2):f.vendor\_id + poly(q.tlenkm,   
## 2):f.paid\_tolls + poly(q.tlenkm, 2):f.extra

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.extra 4 1358.9 1512.2  
## - f.tt 4 1359.3 1512.6  
## - f.period 3 1357.2 1517.8  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1354.3 1522.2  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1356.2 1524.1  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1358.0 1526.0  
## <none> 1354.2 1536.7  
##   
## Step: AIC=1512.2  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.period + f.paid\_tolls + f.tt + f.extra + poly(q.tlenkm,   
## 2):f.mta\_tax + poly(q.tlenkm, 2):f.vendor\_id + poly(q.tlenkm,   
## 2):f.paid\_tolls  
##   
## Df Deviance AIC  
## - f.tt 4 1363.7 1487.8  
## - f.period 3 1362.0 1493.4  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1358.9 1497.7  
## - f.extra 2 1359.0 1497.7  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1360.2 1498.9  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1362.7 1501.4  
## <none> 1358.9 1512.2  
##   
## Step: AIC=1487.77  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.period + f.paid\_tolls + f.extra + poly(q.tlenkm, 2):f.mta\_tax +   
## poly(q.tlenkm, 2):f.vendor\_id + poly(q.tlenkm, 2):f.paid\_tolls  
##   
## Df Deviance AIC  
## - f.period 3 1367.0 1469.2  
## - f.extra 2 1363.8 1473.3  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1363.9 1473.4  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1365.0 1474.6  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1367.5 1477.0  
## <none> 1363.7 1487.8  
##   
## Step: AIC=1469.23  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.paid\_tolls + f.extra + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + poly(q.tlenkm, 2):f.paid\_tolls  
##   
## Df Deviance AIC  
## - f.extra 2 1367.2 1454.8  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1367.2 1454.8  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1368.6 1456.2  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1370.8 1458.5  
## <none> 1367.0 1469.2  
##   
## Step: AIC=1454.77  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.paid\_tolls + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + poly(q.tlenkm, 2):f.paid\_tolls  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1367.4 1440.4  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1368.7 1441.7  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1371.0 1444.0  
## <none> 1367.2 1454.8  
##   
## Step: AIC=1440.37  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.paid\_tolls + poly(q.tlenkm, 2):f.vendor\_id + poly(q.tlenkm,   
## 2):f.paid\_tolls

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1369.0 1427.4  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1371.1 1429.5  
## <none> 1367.4 1440.4  
## - f.mta\_tax 1 1379.7 1445.4  
##   
## Step: AIC=1427.42  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.paid\_tolls + poly(q.tlenkm, 2):f.paid\_tolls

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.paid\_tolls 2 1372.5 1416.3  
## - f.vendor\_id 1 1370.6 1421.7  
## <none> 1369.0 1427.4  
## - f.mta\_tax 1 1381.2 1432.3  
##   
## Step: AIC=1416.35  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.paid\_tolls  
##   
## Df Deviance AIC  
## - f.paid\_tolls 1 1373.1 1409.6  
## - f.vendor\_id 1 1374.2 1410.7  
## - poly(q.tlenkm, 2) 2 1381.9 1411.1  
## <none> 1372.5 1416.3  
## - f.mta\_tax 1 1385.4 1421.9  
##   
## Step: AIC=1409.63  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id  
##   
## Df Deviance AIC  
## - f.vendor\_id 1 1374.8 1404.0  
## - poly(q.tlenkm, 2) 2 1383.4 1405.3  
## <none> 1373.1 1409.6  
## - f.mta\_tax 1 1385.8 1415.0  
##   
## Step: AIC=1404.02  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2) 2 1385.0 1399.6  
## <none> 1374.8 1404.0  
## - f.mta\_tax 1 1387.4 1409.3  
##   
## Step: AIC=1399.63  
## target.tip\_is\_given ~ f.mta\_tax  
##   
## Df Deviance AIC  
## <none> 1385.0 1399.6  
## - f.mta\_tax 1 1397.9 1405.2

From what we can see, it only stays with the tax, but in order to have more freedom, we will keep what we had before.

Hence:

model\_27 <- glm(  
 target.tip\_is\_given~  
 ( poly(q.tlenkm, 2) ) \*   
 ( f.mta\_tax )  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_27)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ (poly(q.tlenkm, 2)) \* (f.mta\_tax),   
## family = "binomial", data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0752 0.5212 0.6039 0.6605 1.3204   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.01392 0.44130 0.032 0.974827   
## poly(q.tlenkm, 2)1 14.55513 29.03614 0.501 0.616177   
## poly(q.tlenkm, 2)2 -1.01357 42.42778 -0.024 0.980941   
## f.mta\_taxYes 1.55167 0.44679 3.473 0.000515 \*\*\*  
## poly(q.tlenkm, 2)1:f.mta\_taxYes -8.37146 29.16854 -0.287 0.774110   
## poly(q.tlenkm, 2)2:f.mta\_taxYes -4.50564 42.49358 -0.106 0.915558   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1374.7 on 1476 degrees of freedom  
## AIC: 1386.7  
##   
## Number of Fisher Scoring iterations: 4

Anova(model\_27, test="Wald")

## Analysis of Deviance Table (Type II tests)  
##   
## Response: target.tip\_is\_given  
## Df Chisq Pr(>Chisq)   
## poly(q.tlenkm, 2) 2 9.8658 0.0072057 \*\*   
## f.mta\_tax 1 13.2941 0.0002662 \*\*\*  
## poly(q.tlenkm, 2):f.mta\_tax 2 0.1592 0.9234768   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We do a comparison:

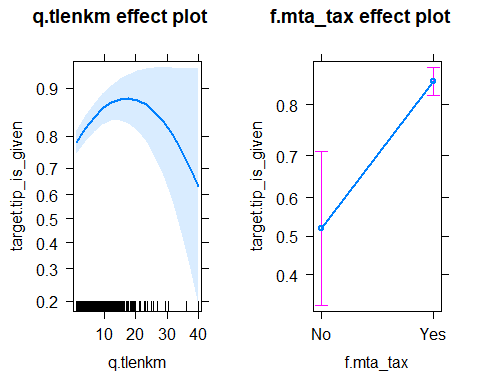
BIC(model\_27, model\_25)

## df BIC  
## model\_27 6 1418.460  
## model\_25 4 1404.024

We keep with the 25.

We can see now the effects of it:

plot(allEffects(model\_25))



* We can observe that only the tips is given in certain range of driven km, due to the fact that for few km, it makes no sense to give it, and for many km it is too much.
* As we have previously commented, it is more likely to give some tips if a tax is present.

### factor-factor

Although, for this deliverable it is asked to do some interactions between factors, we will do it even though the results could not be realistic:

# interaccions dobles entre factors:  
model\_factors\_1 <- glm(  
 target.tip\_is\_given~  
 ( poly(q.tlenkm, 2) + q.extra) + ( f.mta\_tax + f.vendor\_id + f.espeed)^2  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_factors\_1)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ (poly(q.tlenkm, 2) + q.extra) +   
## (f.mta\_tax + f.vendor\_id + f.espeed)^2, family = "binomial",   
## data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2113 0.4981 0.5928 0.6636 1.7147   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error z value  
## (Intercept) -1.275e+01 3.247e+02 -0.039  
## poly(q.tlenkm, 2)1 6.278e+00 3.391e+00 1.852  
## poly(q.tlenkm, 2)2 -6.122e+00 2.626e+00 -2.331  
## q.extra -9.451e-04 1.901e-01 -0.005  
## f.mta\_taxYes 1.363e+01 3.247e+02 0.042  
## f.vendor\_idf.Vendor-VeriFone 4.430e-01 1.388e+00 0.319  
## f.espeed[10,20) 1.143e+01 3.247e+02 0.035  
## f.espeed[20,30) 1.341e+01 3.247e+02 0.041  
## f.espeed[30,40) 1.142e+01 3.247e+02 0.035  
## f.espeed[40,50) 1.176e+01 3.247e+02 0.036  
## f.espeed[50,55] 1.192e+00 1.383e+00 0.862  
## f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone -4.541e-02 1.238e+00 -0.037  
## f.mta\_taxYes:f.espeed[10,20) -1.088e+01 3.247e+02 -0.034  
## f.mta\_taxYes:f.espeed[20,30) -1.274e+01 3.247e+02 -0.039  
## f.mta\_taxYes:f.espeed[30,40) -1.108e+01 3.247e+02 -0.034  
## f.mta\_taxYes:f.espeed[40,50) -1.184e+01 3.247e+02 -0.036  
## f.mta\_taxYes:f.espeed[50,55] NA NA NA  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) -2.463e-01 6.672e-01 -0.369  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) -1.373e-01 6.994e-01 -0.196  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) -3.194e-02 8.303e-01 -0.038  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 1.250e-01 9.969e-01 0.125  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] -5.731e-01 1.547e+00 -0.370  
## Pr(>|z|)   
## (Intercept) 0.9687   
## poly(q.tlenkm, 2)1 0.0641 .  
## poly(q.tlenkm, 2)2 0.0197 \*  
## q.extra 0.9960   
## f.mta\_taxYes 0.9665   
## f.vendor\_idf.Vendor-VeriFone 0.7497   
## f.espeed[10,20) 0.9719   
## f.espeed[20,30) 0.9671   
## f.espeed[30,40) 0.9720   
## f.espeed[40,50) 0.9711   
## f.espeed[50,55] 0.3887   
## f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 0.9707   
## f.mta\_taxYes:f.espeed[10,20) 0.9733   
## f.mta\_taxYes:f.espeed[20,30) 0.9687   
## f.mta\_taxYes:f.espeed[30,40) 0.9728   
## f.mta\_taxYes:f.espeed[40,50) 0.9709   
## f.mta\_taxYes:f.espeed[50,55] NA   
## f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 0.7120   
## f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 0.8444   
## f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 0.9693   
## f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 0.9002   
## f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 0.7111   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1361.6 on 1461 degrees of freedom  
## AIC: 1403.6  
##   
## Number of Fisher Scoring iterations: 11

model\_factors\_1\_step <- step(model\_factors\_1, k=log(nrow(dffwork)))

## Start: AIC=1514.94  
## target.tip\_is\_given ~ (poly(q.tlenkm, 2) + q.extra) + (f.mta\_tax +   
## f.vendor\_id + f.espeed)^2  
##   
## Df Deviance AIC  
## - f.vendor\_id:f.espeed 5 1362.1 1478.9  
## - f.mta\_tax:f.espeed 4 1365.6 1489.7  
## - q.extra 1 1361.6 1507.6  
## - f.mta\_tax:f.vendor\_id 1 1361.6 1507.6  
## - poly(q.tlenkm, 2) 2 1369.7 1508.4  
## <none> 1361.6 1514.9  
##   
## Step: AIC=1478.92  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed  
##   
## Df Deviance AIC  
## - f.mta\_tax:f.espeed 4 1366.0 1453.6  
## - q.extra 1 1362.1 1471.6  
## - f.mta\_tax:f.vendor\_id 1 1362.1 1471.6  
## - poly(q.tlenkm, 2) 2 1370.1 1472.3  
## <none> 1362.1 1478.9  
##   
## Step: AIC=1453.63  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id  
##   
## Df Deviance AIC  
## - f.espeed 5 1372.8 1424.0  
## - q.extra 1 1366.0 1446.3  
## - f.mta\_tax:f.vendor\_id 1 1366.2 1446.5  
## - poly(q.tlenkm, 2) 2 1373.8 1446.8  
## <none> 1366.0 1453.6  
##   
## Step: AIC=1423.96  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.mta\_tax:f.vendor\_id  
##   
## Df Deviance AIC  
## - q.extra 1 1372.8 1416.7  
## - f.mta\_tax:f.vendor\_id 1 1373.1 1416.9  
## - poly(q.tlenkm, 2) 2 1382.8 1419.3  
## <none> 1372.8 1424.0  
##   
## Step: AIC=1416.66  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id +   
## f.mta\_tax:f.vendor\_id  
##   
## Df Deviance AIC  
## - f.mta\_tax:f.vendor\_id 1 1373.1 1409.6  
## - poly(q.tlenkm, 2) 2 1382.9 1412.1  
## <none> 1372.8 1416.7  
##   
## Step: AIC=1409.63  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id  
##   
## Df Deviance AIC  
## - f.vendor\_id 1 1374.8 1404.0  
## - poly(q.tlenkm, 2) 2 1383.4 1405.3  
## <none> 1373.1 1409.6  
## - f.mta\_tax 1 1385.8 1415.0  
##   
## Step: AIC=1404.02  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2) 2 1385.0 1399.6  
## <none> 1374.8 1404.0  
## - f.mta\_tax 1 1387.4 1409.3  
##   
## Step: AIC=1399.63  
## target.tip\_is\_given ~ f.mta\_tax  
##   
## Df Deviance AIC  
## <none> 1385.0 1399.6  
## - f.mta\_tax 1 1397.9 1405.2

We stick with what we had.

# interaccions dobles entre factor-numèrica  
model\_factors\_2 <- glm(  
 target.tip\_is\_given~  
 ( poly(q.tlenkm, 2) + q.extra) \* ( f.mta\_tax + f.vendor\_id + f.espeed)  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_factors\_2)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ (poly(q.tlenkm, 2) + q.extra) \*   
## (f.mta\_tax + f.vendor\_id + f.espeed), family = "binomial",   
## data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7524 0.4510 0.5990 0.6586 1.3945   
##   
## Coefficients:  
## Estimate Std. Error z value  
## (Intercept) -1.0682 0.6215 -1.719  
## poly(q.tlenkm, 2)1 7.2714 36.7875 0.198  
## poly(q.tlenkm, 2)2 18.0176 51.3934 0.351  
## q.extra 24.3486 649.4904 0.037  
## f.mta\_taxYes 1.8229 0.4793 3.803  
## f.vendor\_idf.Vendor-VeriFone 0.1515 0.2252 0.673  
## f.espeed[10,20) 0.7793 0.4311 1.808  
## f.espeed[20,30) 1.0388 0.4180 2.485  
## f.espeed[30,40) 1.0806 0.5386 2.006  
## f.espeed[40,50) 0.1425 0.9805 0.145  
## f.espeed[50,55] 0.5455 1.2749 0.428  
## poly(q.tlenkm, 2)1:f.mta\_taxYes -7.7111 32.4480 -0.238  
## poly(q.tlenkm, 2)2:f.mta\_taxYes -10.8291 47.2310 -0.229  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone 0.9534 9.6855 0.098  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone 5.9339 11.2991 0.525  
## poly(q.tlenkm, 2)1:f.espeed[10,20) 31.1016 33.2114 0.936  
## poly(q.tlenkm, 2)2:f.espeed[10,20) 3.8258 32.8584 0.116  
## poly(q.tlenkm, 2)1:f.espeed[20,30) 9.5247 22.0306 0.432  
## poly(q.tlenkm, 2)2:f.espeed[20,30) -19.6899 26.5292 -0.742  
## poly(q.tlenkm, 2)1:f.espeed[30,40) 2.3595 19.5979 0.120  
## poly(q.tlenkm, 2)2:f.espeed[30,40) -21.8783 22.4290 -0.975  
## poly(q.tlenkm, 2)1:f.espeed[40,50) 0.2109 23.1366 0.009  
## poly(q.tlenkm, 2)2:f.espeed[40,50) -38.1689 27.1164 -1.408  
## poly(q.tlenkm, 2)1:f.espeed[50,55] 5.0641 27.9556 0.181  
## poly(q.tlenkm, 2)2:f.espeed[50,55] -7.5058 26.5425 -0.283  
## q.extra:f.mta\_taxYes -24.0875 649.4899 -0.037  
## q.extra:f.vendor\_idf.Vendor-VeriFone 0.1873 0.4330 0.433  
## q.extra:f.espeed[10,20) -0.2740 0.7254 -0.378  
## q.extra:f.espeed[20,30) -0.7469 0.8007 -0.933  
## q.extra:f.espeed[30,40) -1.8492 1.0443 -1.771  
## q.extra:f.espeed[40,50) 0.4632 1.3795 0.336  
## q.extra:f.espeed[50,55] 0.8395 2.8025 0.300  
## Pr(>|z|)   
## (Intercept) 0.085674 .   
## poly(q.tlenkm, 2)1 0.843311   
## poly(q.tlenkm, 2)2 0.725901   
## q.extra 0.970095   
## f.mta\_taxYes 0.000143 \*\*\*  
## f.vendor\_idf.Vendor-VeriFone 0.501042   
## f.espeed[10,20) 0.070624 .   
## f.espeed[20,30) 0.012961 \*   
## f.espeed[30,40) 0.044825 \*   
## f.espeed[40,50) 0.884481   
## f.espeed[50,55] 0.668748   
## poly(q.tlenkm, 2)1:f.mta\_taxYes 0.812156   
## poly(q.tlenkm, 2)2:f.mta\_taxYes 0.818651   
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone 0.921589   
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone 0.599466   
## poly(q.tlenkm, 2)1:f.espeed[10,20) 0.349030   
## poly(q.tlenkm, 2)2:f.espeed[10,20) 0.907309   
## poly(q.tlenkm, 2)1:f.espeed[20,30) 0.665493   
## poly(q.tlenkm, 2)2:f.espeed[20,30) 0.457968   
## poly(q.tlenkm, 2)1:f.espeed[30,40) 0.904169   
## poly(q.tlenkm, 2)2:f.espeed[30,40) 0.329338   
## poly(q.tlenkm, 2)1:f.espeed[40,50) 0.992729   
## poly(q.tlenkm, 2)2:f.espeed[40,50) 0.159252   
## poly(q.tlenkm, 2)1:f.espeed[50,55] 0.856250   
## poly(q.tlenkm, 2)2:f.espeed[50,55] 0.777343   
## q.extra:f.mta\_taxYes 0.970416   
## q.extra:f.vendor\_idf.Vendor-VeriFone 0.665296   
## q.extra:f.espeed[10,20) 0.705640   
## q.extra:f.espeed[20,30) 0.350939   
## q.extra:f.espeed[30,40) 0.076603 .   
## q.extra:f.espeed[40,50) 0.737021   
## q.extra:f.espeed[50,55] 0.764527   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1348.8 on 1450 degrees of freedom  
## AIC: 1412.8  
##   
## Number of Fisher Scoring iterations: 11

model\_factors\_2\_step <- step(model\_factors\_2, k=log(nrow(dffwork)))

## Start: AIC=1582.42  
## target.tip\_is\_given ~ (poly(q.tlenkm, 2) + q.extra) \* (f.mta\_tax +   
## f.vendor\_id + f.espeed)  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.espeed 10 1358.2 1518.8  
## - q.extra:f.espeed 5 1354.3 1551.4  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1348.8 1567.9  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1349.1 1568.2  
## - q.extra:f.vendor\_id 1 1349.0 1575.3  
## - q.extra:f.mta\_tax 1 1349.6 1576.0  
## <none> 1348.8 1582.4  
##   
## Step: AIC=1518.79  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + q.extra:f.mta\_tax + q.extra:f.vendor\_id +   
## q.extra:f.espeed  
##   
## Df Deviance AIC  
## - q.extra:f.espeed 5 1364.0 1488.1  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1358.2 1504.2  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1360.1 1506.1  
## - q.extra:f.vendor\_id 1 1358.3 1511.6  
## - q.extra:f.mta\_tax 1 1359.0 1512.4  
## <none> 1358.2 1518.8  
##   
## Step: AIC=1488.11  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + q.extra:f.mta\_tax + q.extra:f.vendor\_id  
##   
## Df Deviance AIC  
## - f.espeed 5 1370.3 1458.0  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1364.1 1473.6  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1364.9 1474.4  
## - q.extra:f.vendor\_id 1 1364.2 1481.0  
## - q.extra:f.mta\_tax 1 1364.8 1481.6  
## <none> 1364.0 1488.1  
##   
## Step: AIC=1457.95  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + q.extra:f.mta\_tax + q.extra:f.vendor\_id  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1370.4 1443.4  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1371.6 1444.7  
## - q.extra:f.vendor\_id 1 1370.6 1450.9  
## - q.extra:f.mta\_tax 1 1371.3 1451.6  
## <none> 1370.3 1458.0  
##   
## Step: AIC=1443.39  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + poly(q.tlenkm, 2):f.vendor\_id + q.extra:f.mta\_tax +   
## q.extra:f.vendor\_id  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1371.8 1430.2  
## - q.extra:f.vendor\_id 1 1370.6 1436.3  
## - q.extra:f.mta\_tax 1 1371.4 1437.1  
## <none> 1370.4 1443.4  
##   
## Step: AIC=1430.16  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + q.extra:f.mta\_tax + q.extra:f.vendor\_id  
##   
## Df Deviance AIC  
## - q.extra:f.vendor\_id 1 1372.1 1423.2  
## - q.extra:f.mta\_tax 1 1372.8 1423.9  
## - poly(q.tlenkm, 2) 2 1381.6 1425.4  
## <none> 1371.8 1430.2  
##   
## Step: AIC=1423.23  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + q.extra:f.mta\_tax  
##   
## Df Deviance AIC  
## - q.extra:f.mta\_tax 1 1373.1 1416.9  
## - f.vendor\_id 1 1373.8 1417.6  
## - poly(q.tlenkm, 2) 2 1381.9 1418.4  
## <none> 1372.1 1423.2  
##   
## Step: AIC=1416.93  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id  
##   
## Df Deviance AIC  
## - q.extra 1 1373.1 1409.6  
## - f.vendor\_id 1 1374.8 1411.3  
## - poly(q.tlenkm, 2) 2 1383.3 1412.5  
## <none> 1373.1 1416.9  
## - f.mta\_tax 1 1385.5 1422.0  
##   
## Step: AIC=1409.63  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id  
##   
## Df Deviance AIC  
## - f.vendor\_id 1 1374.8 1404.0  
## - poly(q.tlenkm, 2) 2 1383.4 1405.3  
## <none> 1373.1 1409.6  
## - f.mta\_tax 1 1385.8 1415.0  
##   
## Step: AIC=1404.02  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2) 2 1385.0 1399.6  
## <none> 1374.8 1404.0  
## - f.mta\_tax 1 1387.4 1409.3  
##   
## Step: AIC=1399.63  
## target.tip\_is\_given ~ f.mta\_tax  
##   
## Df Deviance AIC  
## <none> 1385.0 1399.6  
## - f.mta\_tax 1 1397.9 1405.2

We stick with what we had.

# interaccions dobles entre factor-numèrica + dobles entre factors  
model\_factors\_3 <- glm(  
 target.tip\_is\_given~  
 ( poly(q.tlenkm, 2) + q.extra) \* ( f.mta\_tax + f.vendor\_id + f.espeed)^2  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_factors\_3)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ (poly(q.tlenkm, 2) + q.extra) \*   
## (f.mta\_tax + f.vendor\_id + f.espeed)^2, family = "binomial",   
## data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6908 0.4084 0.5969 0.6586 1.3472   
##   
## Coefficients: (13 not defined because of singularities)  
## Estimate  
## (Intercept) 2.861e+03  
## poly(q.tlenkm, 2)1 2.564e+05  
## poly(q.tlenkm, 2)2 1.555e+05  
## q.extra 1.570e+04  
## f.mta\_taxYes -2.866e+03  
## f.vendor\_idf.Vendor-VeriFone -2.689e+03  
## f.espeed[10,20) 1.300e+02  
## f.espeed[20,30) -2.696e+03  
## f.espeed[30,40) -1.819e+02  
## f.espeed[40,50) -6.112e+02  
## f.espeed[50,55] 1.380e+01  
## f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 2.696e+03  
## f.mta\_taxYes:f.espeed[10,20) -1.218e+02  
## f.mta\_taxYes:f.espeed[20,30) 2.703e+03  
## f.mta\_taxYes:f.espeed[30,40) 1.896e+02  
## f.mta\_taxYes:f.espeed[40,50) 4.891e+02  
## f.mta\_taxYes:f.espeed[50,55] NA  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) -7.764e+00  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) -6.235e+00  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) -6.761e+00  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 1.227e+02  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 1.312e+03  
## poly(q.tlenkm, 2)1:f.mta\_taxYes -2.572e+05  
## poly(q.tlenkm, 2)2:f.mta\_taxYes -1.562e+05  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone -2.510e+05  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone -1.592e+05  
## poly(q.tlenkm, 2)1:f.espeed[10,20) 1.978e+04  
## poly(q.tlenkm, 2)2:f.espeed[10,20) 7.782e+02  
## poly(q.tlenkm, 2)1:f.espeed[20,30) -2.583e+05  
## poly(q.tlenkm, 2)2:f.espeed[20,30) -1.511e+05  
## poly(q.tlenkm, 2)1:f.espeed[30,40) -6.618e+03  
## poly(q.tlenkm, 2)2:f.espeed[30,40) 6.165e+02  
## poly(q.tlenkm, 2)1:f.espeed[40,50) 6.601e+02  
## poly(q.tlenkm, 2)2:f.espeed[40,50) -4.258e+03  
## poly(q.tlenkm, 2)1:f.espeed[50,55] 1.233e+03  
## poly(q.tlenkm, 2)2:f.espeed[50,55] 5.019e+02  
## q.extra:f.mta\_taxYes -1.569e+04  
## q.extra:f.vendor\_idf.Vendor-VeriFone -3.489e+00  
## q.extra:f.espeed[10,20) -3.699e+00  
## q.extra:f.espeed[20,30) -3.293e+00  
## q.extra:f.espeed[30,40) -6.015e+00  
## q.extra:f.espeed[40,50) 9.066e+01  
## q.extra:f.espeed[50,55] -3.815e+01  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 2.518e+05  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 1.599e+05  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[10,20) -1.881e+04  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[10,20) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[20,30) 2.592e+05  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[20,30) 1.518e+05  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[30,40) 7.405e+03  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[30,40) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[40,50) NA  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[40,50) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[50,55] NA  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[50,55] NA  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) -9.359e+02  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) -7.781e+02  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) -8.155e+02  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) -7.121e+02  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) -7.834e+02  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) -6.440e+02  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) -6.633e+02  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 4.214e+03  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] -2.353e+04  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 8.514e+03  
## q.extra:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone NA  
## q.extra:f.mta\_taxYes:f.espeed[10,20) NA  
## q.extra:f.mta\_taxYes:f.espeed[20,30) NA  
## q.extra:f.mta\_taxYes:f.espeed[30,40) NA  
## q.extra:f.mta\_taxYes:f.espeed[40,50) NA  
## q.extra:f.mta\_taxYes:f.espeed[50,55] NA  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 4.215e+00  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 3.240e+00  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 4.685e+00  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) -9.031e+01  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 1.124e+03  
## Std. Error  
## (Intercept) 1.469e+08  
## poly(q.tlenkm, 2)1 1.426e+10  
## poly(q.tlenkm, 2)2 8.499e+09  
## q.extra 9.016e+08  
## f.mta\_taxYes 1.469e+08  
## f.vendor\_idf.Vendor-VeriFone 1.469e+08  
## f.espeed[10,20) 1.956e+05  
## f.espeed[20,30) 1.469e+08  
## f.espeed[30,40) 1.511e+05  
## f.espeed[40,50) 4.217e+05  
## f.espeed[50,55] 9.089e+03  
## f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 1.469e+08  
## f.mta\_taxYes:f.espeed[10,20) 1.956e+05  
## f.mta\_taxYes:f.espeed[20,30) 1.469e+08  
## f.mta\_taxYes:f.espeed[30,40) 1.511e+05  
## f.mta\_taxYes:f.espeed[40,50) 4.217e+05  
## f.mta\_taxYes:f.espeed[50,55] NA  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 4.142e+00  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 3.945e+00  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 4.090e+00  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 1.500e+02  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 1.588e+06  
## poly(q.tlenkm, 2)1:f.mta\_taxYes 1.426e+10  
## poly(q.tlenkm, 2)2:f.mta\_taxYes 8.499e+09  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone 1.426e+10  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone 8.499e+09  
## poly(q.tlenkm, 2)1:f.espeed[10,20) 9.142e+06  
## poly(q.tlenkm, 2)2:f.espeed[10,20) 4.664e+02  
## poly(q.tlenkm, 2)1:f.espeed[20,30) 1.426e+10  
## poly(q.tlenkm, 2)2:f.espeed[20,30) 8.499e+09  
## poly(q.tlenkm, 2)1:f.espeed[30,40) 6.057e+06  
## poly(q.tlenkm, 2)2:f.espeed[30,40) 4.486e+02  
## poly(q.tlenkm, 2)1:f.espeed[40,50) 6.885e+02  
## poly(q.tlenkm, 2)2:f.espeed[40,50) 5.726e+03  
## poly(q.tlenkm, 2)1:f.espeed[50,55] 3.830e+05  
## poly(q.tlenkm, 2)2:f.espeed[50,55] 1.484e+05  
## q.extra:f.mta\_taxYes 9.016e+08  
## q.extra:f.vendor\_idf.Vendor-VeriFone 2.201e+00  
## q.extra:f.espeed[10,20) 2.119e+00  
## q.extra:f.espeed[20,30) 2.208e+00  
## q.extra:f.espeed[30,40) 2.762e+00  
## q.extra:f.espeed[40,50) 1.118e+02  
## q.extra:f.espeed[50,55] 3.380e+04  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 1.426e+10  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 8.499e+09  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[10,20) 9.142e+06  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[10,20) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[20,30) 1.426e+10  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[20,30) 8.499e+09  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[30,40) 6.057e+06  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[30,40) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[40,50) NA  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[40,50) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[50,55] NA  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[50,55] NA  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 5.754e+02  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 4.679e+02  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 5.514e+02  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 4.506e+02  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 5.505e+02  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 4.494e+02  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 6.890e+02  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 5.727e+03  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 1.743e+07  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 3.154e+07  
## q.extra:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone NA  
## q.extra:f.mta\_taxYes:f.espeed[10,20) NA  
## q.extra:f.mta\_taxYes:f.espeed[20,30) NA  
## q.extra:f.mta\_taxYes:f.espeed[30,40) NA  
## q.extra:f.mta\_taxYes:f.espeed[40,50) NA  
## q.extra:f.mta\_taxYes:f.espeed[50,55] NA  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 2.272e+00  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 2.391e+00  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 3.029e+00  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 1.118e+02  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 3.114e+06  
## z value  
## (Intercept) 0.000  
## poly(q.tlenkm, 2)1 0.000  
## poly(q.tlenkm, 2)2 0.000  
## q.extra 0.000  
## f.mta\_taxYes 0.000  
## f.vendor\_idf.Vendor-VeriFone 0.000  
## f.espeed[10,20) 0.001  
## f.espeed[20,30) 0.000  
## f.espeed[30,40) -0.001  
## f.espeed[40,50) -0.001  
## f.espeed[50,55] 0.002  
## f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 0.000  
## f.mta\_taxYes:f.espeed[10,20) -0.001  
## f.mta\_taxYes:f.espeed[20,30) 0.000  
## f.mta\_taxYes:f.espeed[30,40) 0.001  
## f.mta\_taxYes:f.espeed[40,50) 0.001  
## f.mta\_taxYes:f.espeed[50,55] NA  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) -1.874  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) -1.580  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) -1.653  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 0.818  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 0.001  
## poly(q.tlenkm, 2)1:f.mta\_taxYes 0.000  
## poly(q.tlenkm, 2)2:f.mta\_taxYes 0.000  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone 0.000  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone 0.000  
## poly(q.tlenkm, 2)1:f.espeed[10,20) 0.002  
## poly(q.tlenkm, 2)2:f.espeed[10,20) 1.669  
## poly(q.tlenkm, 2)1:f.espeed[20,30) 0.000  
## poly(q.tlenkm, 2)2:f.espeed[20,30) 0.000  
## poly(q.tlenkm, 2)1:f.espeed[30,40) -0.001  
## poly(q.tlenkm, 2)2:f.espeed[30,40) 1.374  
## poly(q.tlenkm, 2)1:f.espeed[40,50) 0.959  
## poly(q.tlenkm, 2)2:f.espeed[40,50) -0.744  
## poly(q.tlenkm, 2)1:f.espeed[50,55] 0.003  
## poly(q.tlenkm, 2)2:f.espeed[50,55] 0.003  
## q.extra:f.mta\_taxYes 0.000  
## q.extra:f.vendor\_idf.Vendor-VeriFone -1.585  
## q.extra:f.espeed[10,20) -1.746  
## q.extra:f.espeed[20,30) -1.492  
## q.extra:f.espeed[30,40) -2.178  
## q.extra:f.espeed[40,50) 0.811  
## q.extra:f.espeed[50,55] -0.001  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 0.000  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 0.000  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[10,20) -0.002  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[10,20) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[20,30) 0.000  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[20,30) 0.000  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[30,40) 0.001  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[30,40) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[40,50) NA  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[40,50) NA  
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[50,55] NA  
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[50,55] NA  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) -1.626  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) -1.663  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) -1.479  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) -1.580  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) -1.423  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) -1.433  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) -0.963  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 0.736  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] -0.001  
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 0.000  
## q.extra:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone NA  
## q.extra:f.mta\_taxYes:f.espeed[10,20) NA  
## q.extra:f.mta\_taxYes:f.espeed[20,30) NA  
## q.extra:f.mta\_taxYes:f.espeed[30,40) NA  
## q.extra:f.mta\_taxYes:f.espeed[40,50) NA  
## q.extra:f.mta\_taxYes:f.espeed[50,55] NA  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 1.855  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 1.355  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 1.547  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) -0.808  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 0.000  
## Pr(>|z|)   
## (Intercept) 1.0000   
## poly(q.tlenkm, 2)1 1.0000   
## poly(q.tlenkm, 2)2 1.0000   
## q.extra 1.0000   
## f.mta\_taxYes 1.0000   
## f.vendor\_idf.Vendor-VeriFone 1.0000   
## f.espeed[10,20) 0.9995   
## f.espeed[20,30) 1.0000   
## f.espeed[30,40) 0.9990   
## f.espeed[40,50) 0.9988   
## f.espeed[50,55] 0.9988   
## f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 1.0000   
## f.mta\_taxYes:f.espeed[10,20) 0.9995   
## f.mta\_taxYes:f.espeed[20,30) 1.0000   
## f.mta\_taxYes:f.espeed[30,40) 0.9990   
## f.mta\_taxYes:f.espeed[40,50) 0.9991   
## f.mta\_taxYes:f.espeed[50,55] NA   
## f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 0.0609 .  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 0.1140   
## f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 0.0983 .  
## f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 0.4134   
## f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 0.9993   
## poly(q.tlenkm, 2)1:f.mta\_taxYes 1.0000   
## poly(q.tlenkm, 2)2:f.mta\_taxYes 1.0000   
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone 1.0000   
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone 1.0000   
## poly(q.tlenkm, 2)1:f.espeed[10,20) 0.9983   
## poly(q.tlenkm, 2)2:f.espeed[10,20) 0.0952 .  
## poly(q.tlenkm, 2)1:f.espeed[20,30) 1.0000   
## poly(q.tlenkm, 2)2:f.espeed[20,30) 1.0000   
## poly(q.tlenkm, 2)1:f.espeed[30,40) 0.9991   
## poly(q.tlenkm, 2)2:f.espeed[30,40) 0.1693   
## poly(q.tlenkm, 2)1:f.espeed[40,50) 0.3376   
## poly(q.tlenkm, 2)2:f.espeed[40,50) 0.4571   
## poly(q.tlenkm, 2)1:f.espeed[50,55] 0.9974   
## poly(q.tlenkm, 2)2:f.espeed[50,55] 0.9973   
## q.extra:f.mta\_taxYes 1.0000   
## q.extra:f.vendor\_idf.Vendor-VeriFone 0.1129   
## q.extra:f.espeed[10,20) 0.0808 .  
## q.extra:f.espeed[20,30) 0.1358   
## q.extra:f.espeed[30,40) 0.0294 \*  
## q.extra:f.espeed[40,50) 0.4174   
## q.extra:f.espeed[50,55] 0.9991   
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 1.0000   
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone 1.0000   
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[10,20) 0.9984   
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[10,20) NA   
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[20,30) 1.0000   
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[20,30) 1.0000   
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[30,40) 0.9990   
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[30,40) NA   
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[40,50) NA   
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[40,50) NA   
## poly(q.tlenkm, 2)1:f.mta\_taxYes:f.espeed[50,55] NA   
## poly(q.tlenkm, 2)2:f.mta\_taxYes:f.espeed[50,55] NA   
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 0.1039   
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 0.0963 .  
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 0.1391   
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 0.1140   
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 0.1547   
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 0.1519   
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 0.3357   
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 0.4618   
## poly(q.tlenkm, 2)1:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 0.9989   
## poly(q.tlenkm, 2)2:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 0.9998   
## q.extra:f.mta\_taxYes:f.vendor\_idf.Vendor-VeriFone NA   
## q.extra:f.mta\_taxYes:f.espeed[10,20) NA   
## q.extra:f.mta\_taxYes:f.espeed[20,30) NA   
## q.extra:f.mta\_taxYes:f.espeed[30,40) NA   
## q.extra:f.mta\_taxYes:f.espeed[40,50) NA   
## q.extra:f.mta\_taxYes:f.espeed[50,55] NA   
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[10,20) 0.0636 .  
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[20,30) 0.1755   
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[30,40) 0.1219   
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[40,50) 0.4193   
## q.extra:f.vendor\_idf.Vendor-VeriFone:f.espeed[50,55] 0.9997   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1397.9 on 1481 degrees of freedom  
## Residual deviance: 1286.3 on 1419 degrees of freedom  
## AIC: 1412.3  
##   
## Number of Fisher Scoring iterations: 20

model\_factors\_3\_step <- step(model\_factors\_3, k=log(nrow(dffwork)))

## Start: AIC=1746.31  
## target.tip\_is\_given ~ (poly(q.tlenkm, 2) + q.extra) \* (f.mta\_tax +   
## f.vendor\_id + f.espeed)^2

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1746.31  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed +   
## f.vendor\_id:f.espeed + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + poly(q.tlenkm, 2):f.espeed + q.extra:f.mta\_tax +   
## q.extra:f.vendor\_id + q.extra:f.espeed + poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id +   
## poly(q.tlenkm, 2):f.mta\_tax:f.espeed + poly(q.tlenkm, 2):f.vendor\_id:f.espeed +   
## q.extra:f.mta\_tax:f.vendor\_id + q.extra:f.vendor\_id:f.espeed

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1746.31  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed +   
## f.vendor\_id:f.espeed + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + poly(q.tlenkm, 2):f.espeed + q.extra:f.mta\_tax +   
## q.extra:f.vendor\_id + q.extra:f.espeed + poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id +   
## poly(q.tlenkm, 2):f.mta\_tax:f.espeed + poly(q.tlenkm, 2):f.vendor\_id:f.espeed +   
## q.extra:f.vendor\_id:f.espeed

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.vendor\_id:f.espeed 10 1299.2 1686.2  
## - q.extra:f.vendor\_id:f.espeed 5 1297.3 1720.8  
## - poly(q.tlenkm, 2):f.mta\_tax:f.espeed 4 1294.8 1725.5  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1292.0 1737.3  
## - q.extra:f.mta\_tax 1 1286.3 1739.0  
## <none> 1286.3 1746.3

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1686.15  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed +   
## f.vendor\_id:f.espeed + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + poly(q.tlenkm, 2):f.espeed + q.extra:f.mta\_tax +   
## q.extra:f.vendor\_id + q.extra:f.espeed + poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id +   
## poly(q.tlenkm, 2):f.mta\_tax:f.espeed + q.extra:f.vendor\_id:f.espeed

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.mta\_tax:f.espeed 4 1307.6 1665.4  
## - q.extra:f.vendor\_id:f.espeed 5 1323.9 1674.4  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1305.2 1677.5  
## - q.extra:f.mta\_tax 1 1299.2 1678.8  
## <none> 1299.2 1686.2

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1665.4  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed +   
## f.vendor\_id:f.espeed + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + poly(q.tlenkm, 2):f.espeed + q.extra:f.mta\_tax +   
## q.extra:f.vendor\_id + q.extra:f.espeed + poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id +   
## q.extra:f.vendor\_id:f.espeed

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.espeed 10 1326.6 1611.3  
## - f.mta\_tax:f.espeed 4 1320.4 1649.0  
## - q.extra:f.vendor\_id:f.espeed 5 1332.3 1653.6  
## - q.extra:f.mta\_tax 1 1307.6 1658.1  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1318.1 1661.2  
## <none> 1307.6 1665.4

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1611.34  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed +   
## f.vendor\_id:f.espeed + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + q.extra:f.mta\_tax + q.extra:f.vendor\_id +   
## q.extra:f.espeed + poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id +   
## q.extra:f.vendor\_id:f.espeed

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - q.extra:f.vendor\_id:f.espeed 5 1343.9 1592.1  
## - f.mta\_tax:f.espeed 4 1338.7 1594.2  
## - q.extra:f.mta\_tax 1 1326.6 1604.0  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1336.6 1606.7  
## <none> 1326.6 1611.3

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1592.13  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed +   
## f.vendor\_id:f.espeed + poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm,   
## 2):f.vendor\_id + q.extra:f.mta\_tax + q.extra:f.vendor\_id +   
## q.extra:f.espeed + poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - f.vendor\_id:f.espeed 5 1344.9 1556.7  
## - q.extra:f.espeed 5 1349.8 1561.5  
## - f.mta\_tax:f.espeed 4 1355.8 1574.8  
## - q.extra:f.mta\_tax 1 1343.9 1584.8  
## - q.extra:f.vendor\_id 1 1344.0 1584.9  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1353.4 1587.0  
## <none> 1343.9 1592.1

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1556.67  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed +   
## poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm, 2):f.vendor\_id +   
## q.extra:f.mta\_tax + q.extra:f.vendor\_id + q.extra:f.espeed +   
## poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - q.extra:f.espeed 5 1350.8 1526.1  
## - f.mta\_tax:f.espeed 4 1356.8 1539.3  
## - q.extra:f.mta\_tax 1 1344.9 1549.4  
## - q.extra:f.vendor\_id 1 1345.0 1549.5  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1354.3 1551.4  
## <none> 1344.9 1556.7

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##   
## Step: AIC=1526.06  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + f.mta\_tax:f.espeed +   
## poly(q.tlenkm, 2):f.mta\_tax + poly(q.tlenkm, 2):f.vendor\_id +   
## q.extra:f.mta\_tax + q.extra:f.vendor\_id + poly(q.tlenkm,   
## 2):f.mta\_tax:f.vendor\_id

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Df Deviance AIC  
## - f.mta\_tax:f.espeed 4 1362.5 1508.5  
## - q.extra:f.mta\_tax 1 1350.8 1518.8  
## - q.extra:f.vendor\_id 1 1351.0 1518.9  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1360.2 1520.8  
## <none> 1350.8 1526.1  
##   
## Step: AIC=1508.51  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.espeed + f.mta\_tax:f.vendor\_id + poly(q.tlenkm,   
## 2):f.mta\_tax + poly(q.tlenkm, 2):f.vendor\_id + q.extra:f.mta\_tax +   
## q.extra:f.vendor\_id + poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id  
##   
## Df Deviance AIC  
## - f.espeed 5 1368.8 1478.3  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1363.9 1495.3  
## - q.extra:f.vendor\_id 1 1362.6 1501.3  
## - q.extra:f.mta\_tax 1 1363.1 1501.8  
## <none> 1362.5 1508.5  
##   
## Step: AIC=1478.29  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.mta\_tax:f.vendor\_id + poly(q.tlenkm, 2):f.mta\_tax +   
## poly(q.tlenkm, 2):f.vendor\_id + q.extra:f.mta\_tax + q.extra:f.vendor\_id +   
## poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.mta\_tax:f.vendor\_id 2 1370.1 1465.0  
## - q.extra:f.vendor\_id 1 1369.0 1471.2  
## - q.extra:f.mta\_tax 1 1369.6 1471.8  
## <none> 1368.8 1478.3  
##   
## Step: AIC=1465.05  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.mta\_tax:f.vendor\_id + poly(q.tlenkm, 2):f.mta\_tax +   
## poly(q.tlenkm, 2):f.vendor\_id + q.extra:f.mta\_tax + q.extra:f.vendor\_id  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.mta\_tax 2 1370.2 1450.5  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1371.4 1451.7  
## - q.extra:f.vendor\_id 1 1370.3 1457.9  
## - f.mta\_tax:f.vendor\_id 1 1370.3 1458.0  
## - q.extra:f.mta\_tax 1 1371.2 1458.8  
## <none> 1370.1 1465.0  
##   
## Step: AIC=1450.47  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.mta\_tax:f.vendor\_id + poly(q.tlenkm, 2):f.vendor\_id +   
## q.extra:f.mta\_tax + q.extra:f.vendor\_id  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2):f.vendor\_id 2 1371.5 1437.2  
## - q.extra:f.vendor\_id 1 1370.3 1443.3  
## - f.mta\_tax:f.vendor\_id 1 1370.4 1443.4  
## - q.extra:f.mta\_tax 1 1371.3 1444.3  
## <none> 1370.2 1450.5  
##   
## Step: AIC=1437.18  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.mta\_tax:f.vendor\_id + q.extra:f.mta\_tax +   
## q.extra:f.vendor\_id  
##   
## Df Deviance AIC  
## - q.extra:f.vendor\_id 1 1371.7 1430.2  
## - f.mta\_tax:f.vendor\_id 1 1371.8 1430.2  
## - q.extra:f.mta\_tax 1 1372.6 1431.0  
## - poly(q.tlenkm, 2) 2 1381.0 1432.2  
## <none> 1371.5 1437.2  
##   
## Step: AIC=1430.15  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + f.mta\_tax:f.vendor\_id + q.extra:f.mta\_tax  
##   
## Df Deviance AIC  
## - f.mta\_tax:f.vendor\_id 1 1372.1 1423.2  
## - q.extra:f.mta\_tax 1 1372.8 1424.0  
## - poly(q.tlenkm, 2) 2 1381.2 1425.0  
## <none> 1371.7 1430.2  
##   
## Step: AIC=1423.23  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id + q.extra:f.mta\_tax  
##   
## Df Deviance AIC  
## - q.extra:f.mta\_tax 1 1373.1 1416.9  
## - f.vendor\_id 1 1373.8 1417.6  
## - poly(q.tlenkm, 2) 2 1381.9 1418.4  
## <none> 1372.1 1423.2  
##   
## Step: AIC=1416.93  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + q.extra + f.mta\_tax +   
## f.vendor\_id  
##   
## Df Deviance AIC  
## - q.extra 1 1373.1 1409.6  
## - f.vendor\_id 1 1374.8 1411.3  
## - poly(q.tlenkm, 2) 2 1383.3 1412.5  
## <none> 1373.1 1416.9  
## - f.mta\_tax 1 1385.5 1422.0  
##   
## Step: AIC=1409.63  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax + f.vendor\_id  
##   
## Df Deviance AIC  
## - f.vendor\_id 1 1374.8 1404.0  
## - poly(q.tlenkm, 2) 2 1383.4 1405.3  
## <none> 1373.1 1409.6  
## - f.mta\_tax 1 1385.8 1415.0  
##   
## Step: AIC=1404.02  
## target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax  
##   
## Df Deviance AIC  
## - poly(q.tlenkm, 2) 2 1385.0 1399.6  
## <none> 1374.8 1404.0  
## - f.mta\_tax 1 1387.4 1409.3  
##   
## Step: AIC=1399.63  
## target.tip\_is\_given ~ f.mta\_tax  
##   
## Df Deviance AIC  
## <none> 1385.0 1399.6  
## - f.mta\_tax 1 1397.9 1405.2

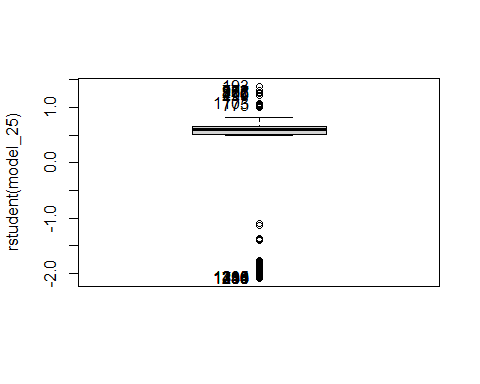
We stick with what we had.

Conclusion: we stick with the idea that the best model is model\_25.

Now, we are going to do some diagnosis:

# Diagnosis //Diagnosis of waste and observations. Lack of adjustment and / or influential.

Boxplot(rstudent(model\_25), id.n=15)



## [1] 796 1411 244 789 18 230 1344 1315 430 1216 193 178 772 922 891  
## [16] 257 416 290 1103 775

sout <- which(abs(rstudent(model\_25))>2); length(sout) # posem 2 en comptes de 2.5 perquè no tenim observacions en aquell rang

## [1] 32

llout <- which(row.names(dffwork) %in% names(rstudent(model\_25)[sout])); llout

## [1] 18 24 36 96 122 230 244 262 352 375 419 430 716 718 720  
## [16] 728 789 796 833 837 845 965 1071 1185 1216 1261 1290 1315 1344 1357  
## [31] 1368 1411

table(dffwork[llout,]$f.mta\_tax, dffwork[llout,]$target.tip\_is\_given)

##   
## No Yes  
## No 0 0  
## Yes 32 0

We see that they are samples that contain mta\_tax, but in the other hand they do not have tip.

We are going to determine which are the potencially influent observations:

quantile(hatvalues(model\_25), seq(0,1,0.1))

## 0% 10% 20% 30% 40% 50%   
## 0.0007271260 0.0007493487 0.0008151965 0.0009047853 0.0010084908 0.0011296173   
## 60% 70% 80% 90% 100%   
## 0.0012967311 0.0014734750 0.0016677898 0.0024633779 0.3957103629

mean(hatvalues(model\_25))

## [1] 0.002699055

hh <- 5\*mean(hatvalues(model\_25)); hh

## [1] 0.01349528

shat <- which(hatvalues(model\_25)>hh); length(shat); shat

## [1] 32

## 23958 64875 121215 125894 131915 196087 204903 273222 305082 309088   
## 21 65 114 119 125 178 193 257 286 290   
## 449320 529073 582626 666624 710390 720785 735280 761529 771658 773934   
## 416 503 554 645 694 706 725 757 772 775   
## 810930 825842 892761 916116 957220 1083301 1204489 1242754 1272045 1289351   
## 815 835 891 922 965 1103 1222 1268 1293 1317   
## 1342604 1377906   
## 1369 1401

summary(dffwork[shat,])

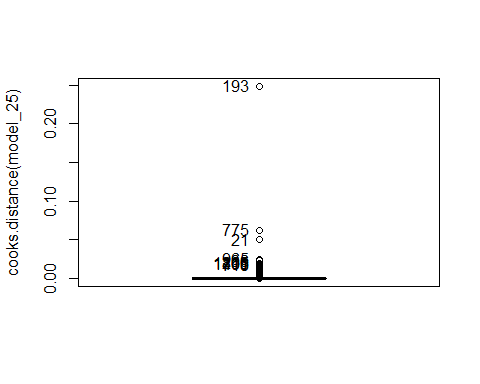
## f.vendor\_id f.code\_rate\_id q.pickup\_longitude q.pickup\_latitude  
## f.Vendor-Mobile : 7 Rate-1 : 6 Min. :-73.99 Min. :40.59   
## f.Vendor-VeriFone:25 Rate-Other:26 1st Qu.:-73.95 1st Qu.:40.70   
## Median :-73.93 Median :40.82   
## Mean :-73.92 Mean :40.77   
## 3rd Qu.:-73.90 3rd Qu.:40.82   
## Max. :-73.81 Max. :40.85   
##   
## q.dropoff\_longitude q.dropoff\_latitude q.passenger\_count q.trip\_distance   
## Min. :-73.99 Min. :40.58 Min. :1.000 Min. : 0.010   
## 1st Qu.:-73.95 1st Qu.:40.68 1st Qu.:1.000 1st Qu.: 1.330   
## Median :-73.94 Median :40.74 Median :1.000 Median : 5.472   
## Mean :-73.93 Mean :40.74 Mean :1.438 Mean : 7.809   
## 3rd Qu.:-73.90 3rd Qu.:40.82 3rd Qu.:2.000 3rd Qu.:12.455   
## Max. :-73.84 Max. :40.85 Max. :4.000 Max. :27.000   
##   
## q.fare\_amount q.extra f.mta\_tax q.tip\_amount q.tolls\_amount   
## Min. : 7.00 Min. :0.0000 No :24 Min. : 0.000 Min. :0.0000   
## 1st Qu.:12.00 1st Qu.:0.0000 Yes: 8 1st Qu.: 0.000 1st Qu.:0.0000   
## Median :23.50 Median :0.0000 Median : 1.000 Median :0.0000   
## Mean :30.56 Mean :0.0625 Mean : 3.528 Mean :0.4049   
## 3rd Qu.:47.75 3rd Qu.:0.0000 3rd Qu.: 5.100 3rd Qu.:0.0000   
## Max. :60.00 Max. :0.5000 Max. :14.350 Max. :5.5400   
##   
## f.improvement\_surcharge target.total\_amount f.payment\_type  
## No :23 Min. : 7.00 Credit card:32   
## Yes: 9 1st Qu.:14.05 Cash : 0   
## Median :26.20 No paid : 0   
## Mean :37.07   
## 3rd Qu.:55.73   
## Max. :97.05   
##   
## f.trip\_type q.hour f.period q.tlenkm   
## Street-Hail: 9 Min. : 0.00 Period night :15 Min. : 1.000   
## Dispatch :23 1st Qu.: 3.75 Period morning : 4 1st Qu.: 1.000   
## Median :10.50 Period valley : 6 Median : 5.681   
## Mean :10.53 Period afternoon: 7 Mean :11.871   
## 3rd Qu.:17.25 3rd Qu.:20.044   
## Max. :23.00 Max. :43.452   
##   
## q.traveltime q.espeed qual.pickup qual.dropoff   
## Min. : 0.01667 Min. : 7.242 Length:32 Length:32   
## 1st Qu.: 0.34167 1st Qu.:20.976 Class :character Class :character   
## Median : 9.24167 Median :26.850 Mode :character Mode :character   
## Mean :16.16198 Mean :30.175   
## 3rd Qu.:30.58750 3rd Qu.:37.008   
## Max. :57.71667 Max. :55.000   
##   
## f.trip\_distance\_range target.tip\_is\_given f.passenger\_groups f.paid\_tolls  
## Long\_dist :25 No :14 Couple: 8 No :28   
## Medium\_dist: 0 Yes:18 Group : 2 Yes: 4   
## Short\_dist : 7 Single:22   
##   
##   
##   
##   
## f.cost f.tt f.dist f.hour f.espeed f.extra   
## (11,18] : 4 (10,15]: 1 (0, 1.6] :10 17 : 2 [03,10): 1 0 :28   
## (18,30] : 6 (15,20]: 4 (1.6, 3] : 5 18 : 2 [10,20): 5 0.5: 4   
## (30,50] : 6 (20,60]:11 (3, 5.5] : 1 19 : 2 [20,30):13 1 : 0   
## (50,129):10 (5,10] : 1 (5.5, 30]:16 20 : 1 [30,40): 6   
## (8,11] : 4 [0,5] :15 21 : 1 [40,50): 3   
## [0,8] : 2 22 : 1 [50,55]: 4   
## other:23

They tend to:

* have rate=Rate-Other
* be in the same location (they have very similar latitudes and longitudes)
* have extra=0
* don’t have mta\_tax
* be at night
* be one passenger
* be long (distance) but short (time)

Now, to decide the influences, we are going to take a look at the cook distances:

Boxplot(cooks.distance(model\_25))



## [1] 193 775 21 965 36 815 1261 1401 706 718

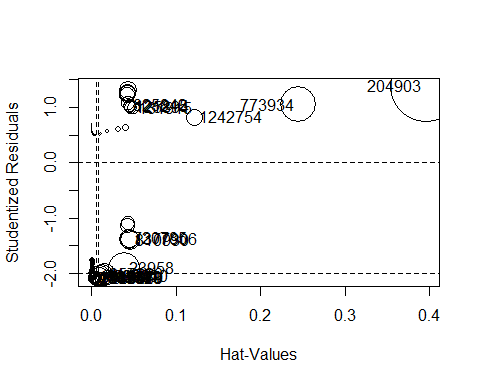
scoo <- which(cooks.distance(model\_25) > 0.02); length(scoo)

## [1] 5

llcoo <- which(row.names(dffwork) %in% names(cooks.distance(model\_25)[scoo])); llcoo

## [1] 21 36 193 775 965

llista<-influencePlot(model\_25, id=c(list="noteworthy", n=10))



summary(dffwork[llcoo,])

## f.vendor\_id f.code\_rate\_id q.pickup\_longitude q.pickup\_latitude  
## f.Vendor-Mobile :1 Rate-1 :5 Min. :-73.99 Min. :40.70   
## f.Vendor-VeriFone:4 Rate-Other:0 1st Qu.:-73.96 1st Qu.:40.70   
## Median :-73.95 Median :40.70   
## Mean :-73.93 Mean :40.74   
## 3rd Qu.:-73.93 3rd Qu.:40.76   
## Max. :-73.81 Max. :40.82   
##   
## q.dropoff\_longitude q.dropoff\_latitude q.passenger\_count q.trip\_distance  
## Min. :-73.99 Min. :40.58 Min. :1 Min. :15.47   
## 1st Qu.:-73.98 1st Qu.:40.61 1st Qu.:1 1st Qu.:15.92   
## Median :-73.98 Median :40.64 Median :1 Median :18.89   
## Mean :-73.97 Mean :40.66 Mean :1 Mean :20.44   
## 3rd Qu.:-73.96 3rd Qu.:40.72 3rd Qu.:1 3rd Qu.:24.92   
## Max. :-73.95 Max. :40.72 Max. :1 Max. :27.00   
##   
## q.fare\_amount q.extra f.mta\_tax q.tip\_amount q.tolls\_amount  
## Min. :44 Min. :0.0 No :0 Min. : 0.000 Min. :0   
## 1st Qu.:47 1st Qu.:0.0 Yes:5 1st Qu.: 0.000 1st Qu.:0   
## Median :54 Median :0.0 Median : 0.000 Median :0   
## Mean :53 Mean :0.2 Mean : 5.542 Mean :0   
## 3rd Qu.:60 3rd Qu.:0.5 3rd Qu.:13.360 3rd Qu.:0   
## Max. :60 Max. :0.5 Max. :14.350 Max. :0   
##   
## f.improvement\_surcharge target.total\_amount f.payment\_type  
## No :0 Min. :44.80 Credit card:5   
## Yes:5 1st Qu.:47.80 Cash :0   
## Median :55.30 No paid :0   
## Mean :62.84   
## 3rd Qu.:80.16   
## Max. :86.15   
##   
## f.trip\_type q.hour f.period q.tlenkm   
## Street-Hail:5 Min. : 0.0 Period night :3 Min. :24.90   
## Dispatch :0 1st Qu.: 3.0 Period morning :1 1st Qu.:25.62   
## Median : 7.0 Period valley :1 Median :30.40   
## Mean : 6.2 Period afternoon:0 Mean :32.89   
## 3rd Qu.: 8.0 3rd Qu.:40.10   
## Max. :13.0 Max. :43.45   
##   
## q.traveltime q.espeed qual.pickup qual.dropoff   
## Min. :30.15 Min. :34.87 Length:5 Length:5   
## 1st Qu.:36.73 1st Qu.:43.62 Class :character Class :character   
## Median :41.72 Median :49.55 Mode :character Mode :character   
## Mean :38.90 Mean :47.61   
## 3rd Qu.:41.82 3rd Qu.:55.00   
## Max. :44.08 Max. :55.00   
##   
## f.trip\_distance\_range target.tip\_is\_given f.passenger\_groups f.paid\_tolls  
## Long\_dist :5 No :3 Couple:0 No :5   
## Medium\_dist:0 Yes:2 Group :0 Yes:0   
## Short\_dist :0 Single:5   
##   
##   
##   
##   
## f.cost f.tt f.dist f.hour f.espeed f.extra  
## (11,18] :0 (10,15]:0 (0, 1.6] :0 17 :0 [03,10):0 0 :3   
## (18,30] :0 (15,20]:0 (1.6, 3] :0 18 :0 [10,20):0 0.5:2   
## (30,50] :2 (20,60]:5 (3, 5.5] :0 19 :0 [20,30):0 1 :0   
## (50,129):3 (5,10] :0 (5.5, 30]:5 20 :0 [30,40):1   
## (8,11] :0 [0,5] :0 21 :0 [40,50):2   
## [0,8] :0 22 :0 [50,55]:2   
## other:5

They tend to:

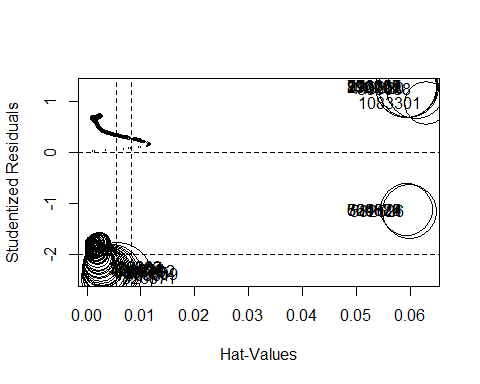
* have rate=Rate-1
* be in the same location (they have very similar latitudes and longitudes)
* be one passanger
* be between 20 and 60 min long
* have mta\_tax
* be long (distance) but short (time)

We redo the model now:

llout<-row.names(llista)  
ll<-which(row.names(dffwork)%in%llout);  
dffwork<-dffwork[-ll,]  
  
model\_25 <- glm(  
 target.tip\_is\_given~  
 poly(q.tlenkm, 2)+  
 f.mta\_tax  
 ,family = "binomial"  
 ,data=dffwork  
); summary(model\_25)

##   
## Call:  
## glm(formula = target.tip\_is\_given ~ poly(q.tlenkm, 2) + f.mta\_tax,   
## family = "binomial", data = dffwork)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4373 0.3971 0.6087 0.6739 1.2559   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.2325 0.4944 0.470 0.638172   
## poly(q.tlenkm, 2)1 25.7294 7.0098 3.671 0.000242 \*\*\*  
## poly(q.tlenkm, 2)2 8.6669 7.2389 1.197 0.231205   
## f.mta\_taxYes 1.4849 0.4978 2.983 0.002858 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1332.6 on 1456 degrees of freedom  
## Residual deviance: 1285.6 on 1453 degrees of freedom  
## AIC: 1293.6  
##   
## Number of Fisher Scoring iterations: 6

influencePlot(model\_25, id=c(list="noteworthy", n=10))



## StudRes Hat CookD  
## 64875 -1.1244847 0.059230834 0.013938544  
## 128263 -2.2493994 0.003010932 0.008586978  
## 196087 1.2856116 0.059230834 0.020082880  
## 273222 1.2720603 0.059197336 0.019479067  
## 305082 -1.1244847 0.059230834 0.013938544  
## 309088 1.2273741 0.061127527 0.018243499  
## 385367 -2.2547076 0.003058657 0.008833378  
## 404073 -2.2953956 0.003463803 0.011015409  
## 449320 1.2566957 0.059673111 0.018986601  
## 450785 -2.3535691 0.004149714 0.015146256  
## 529073 -1.1244847 0.059230834 0.013938544  
## 582626 -1.1543538 0.059754598 0.015081469  
## 666624 -1.1244847 0.059230834 0.013938544  
## 724424 -2.2802957 0.003305568 0.010142806  
## 728634 -2.3037213 0.003554806 0.011530018  
## 736861 -2.3905471 0.004637102 0.018472013  
## 761529 -1.1244847 0.059230834 0.013938544  
## 771658 1.2856116 0.059230834 0.020082880  
## 826871 -2.4585184 0.005599183 0.026179716  
## 892761 1.2724851 0.059190198 0.019494902  
## 916116 1.2856116 0.059230834 0.020082880  
## 1083301 0.9511501 0.062906442 0.009734763  
## 1159509 -2.3696961 0.004358162 0.016525816  
## 1332202 -2.3219791 0.003763192 0.012745456

!!!!! falta recalcular interaccions –> si n’hi ha millors, fer el model

# Confusion Table

fit.tip\_is\_given <- factor(ifelse(predict(model\_25, type="response")<0.5,0,1), labels=c("fit.no", "fit.yes"))  
tt <- table(fit.tip\_is\_given,dffwork$target.tip\_is\_given); tt

##   
## fit.tip\_is\_given No Yes  
## fit.no 9 7  
## fit.yes 240 1201

100\*sum(diag(tt)/sum(tt)) #accuracy

## [1] 83.04736

100\*(tt[2,2]/(tt[2,2] + tt[1,2])) # recall (sensitivity)

## [1] 99.42053

100\*(tt[1,1]/(tt[1,1] + tt[2,1])) # specificity

## [1] 3.614458

100\*(tt[2,2]/(tt[2,1]+ tt[2,2])) # precision

## [1] 83.3449

We have an accuracy of 83.05%. We have a recall of 99.42% which means that the positive results of this confusion table is very accurate. We can see that we have 1201 + 7 positive observations, from which 1201 of them have been correctly classified. Now, we are going to do the same, but for the negative results (specificity). We can see that only a 3.61% of specificity, which is a very bad result. Only 9 of the 240 + 9 negative observations have been classified as negative. To conclude, we see that the precision of this confusion table is 83.34%.