Deliverable 3

Numeric and Binary targets Forecasting Models

Júlia Gasull i Claudia Sánchez

Create factors needed for this deliverable

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

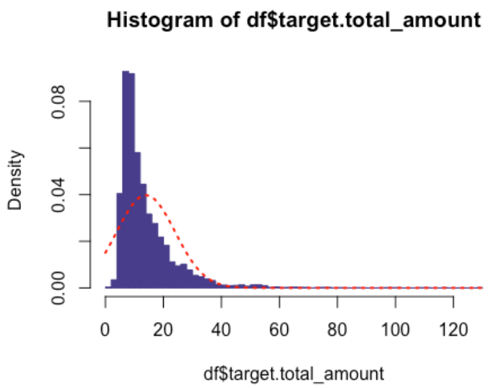
```
vars_con<-names(df)[c(3:10,12:13,15,18,20:22)];
vars_dis<-names(df)[c(1:2,16,19,27:32)];
vars_res<-names(df)[c(15,27)];
vars_cexp<-vars_con[c(5:10,12:15)];
```

Quantitative Logistics Regression

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

(0) 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

```
kewness(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.

(1) Numerical variables

Method 1: take the most correlated variables

We use spearman method since out target is not normally distributed

```
// Annex - Q1
```

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
```

As we have seen before, the most correlated variables are:

- 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: f 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)
## 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 ***
##
```

```
## 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)
## 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

```
// Annex - Q2
```
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)
## 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 ***
##
## 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)

## 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 ***
##
## 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)
## 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)

## 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
##
## 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)
## 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.tr
aveltime+q.espeed,data=df[,c("target.total_amount",vars_cexp)]);summary(model_4)

## 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 ***
##
## 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)
## 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:

```
// Annex - Q3
```

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)

## 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 ***
##
## 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
```

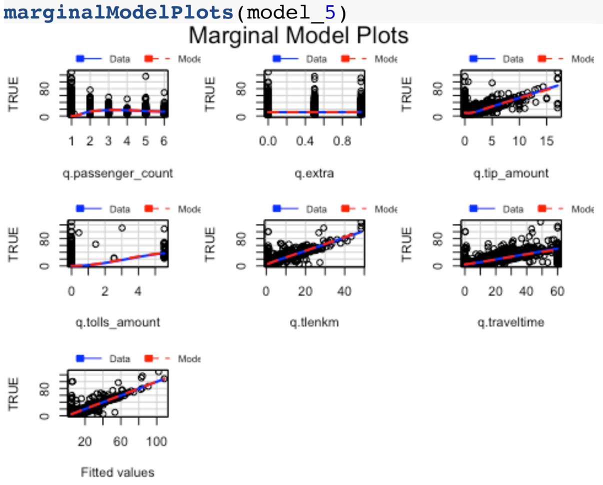
We see that the explainability is now 86.09%

```
vif(model_5)
## 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:

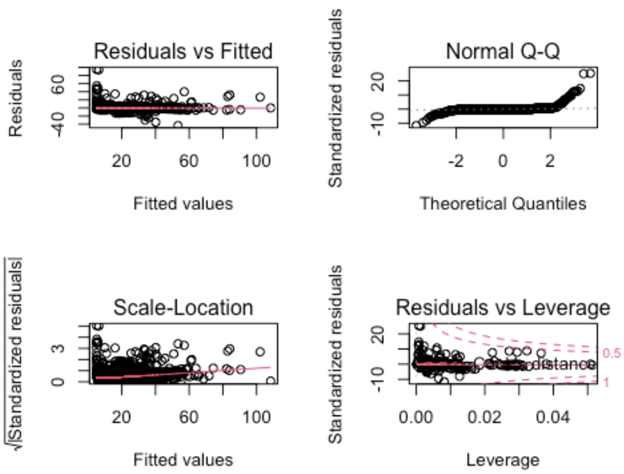
```
marginalModelPlots(model_5)
```



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)
##
## 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 ***
##
## 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:

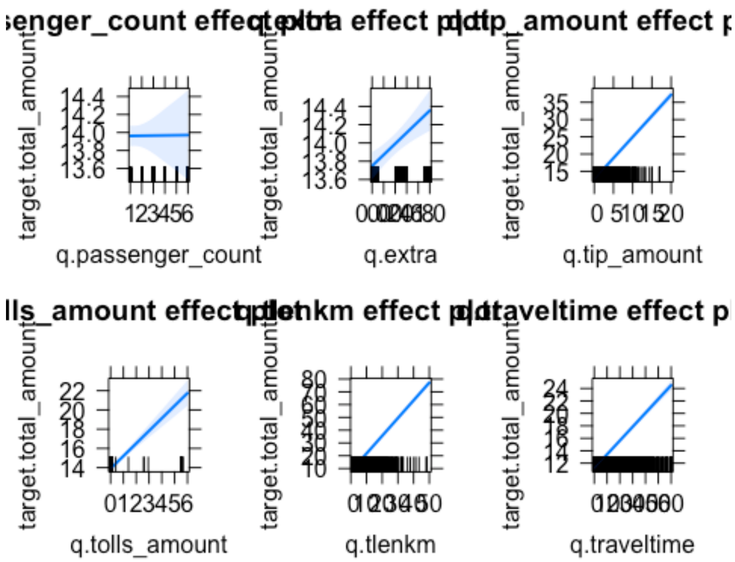
- Model 4
  - Coefficient of determination = 86,17%
  - > 5 VIFs:
    - q.tlenkm: 6.195063
- Model 5
  - Coefficient of determination = 86.09%
  - > 5 VIFs:
    - none
- Model 6
  - Coefficient of determination = 79.51%
  - > 5 VIFs:
    - none

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
```

We see that now the net effects are significant.

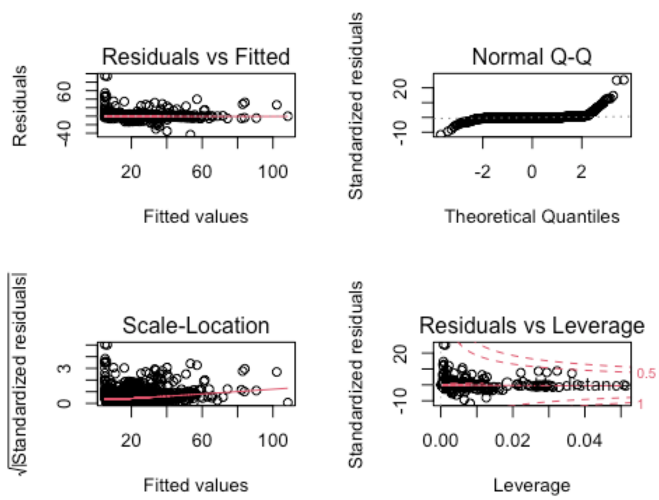
```
library(effects)
plot(allEffects(model_5))
```



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 )
```

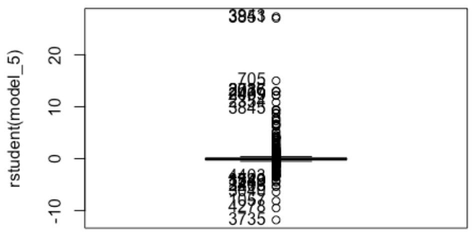



```
par(mfrow=c(1,1))
```

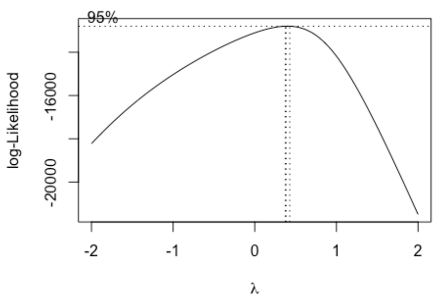
We see that the residues are not completely optimal.

Exhaustive

```
lll<-Boxplot(rstudent(model_5));lll
```




```
##  [1] 3735 4278 1057 3040 3216 2403 1249 3540 1723 4403 3943 3851  705 3026 2037
## [16] 2716 2439 2009 2354 3845

// Annex-Q4

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.trave
ltime,data=df)
```



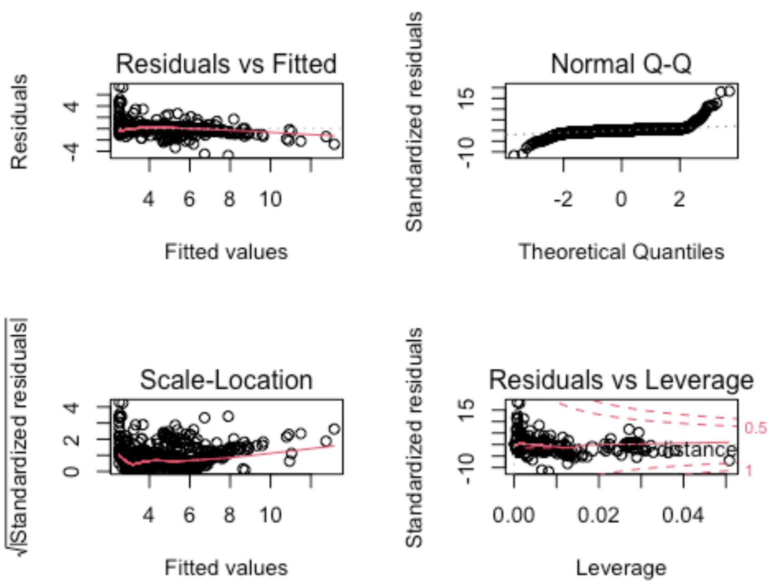
We see the lambda parameter estimation method in the boxcox method. This gives us an idea of the power to which we need to raise the target variable in order to improve the properties of the linear model.
It is worth trying a new model with a square root in the target variable:

```
model_7 <-
lm(sqrt(target.total_amount)~q.passenger_count+q.extra+q.tip_amount+q.tolls_amount+q.tlenkm+q.tra
veltime,data=df);summary(model_7)
##
## 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 ***
##
## 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.

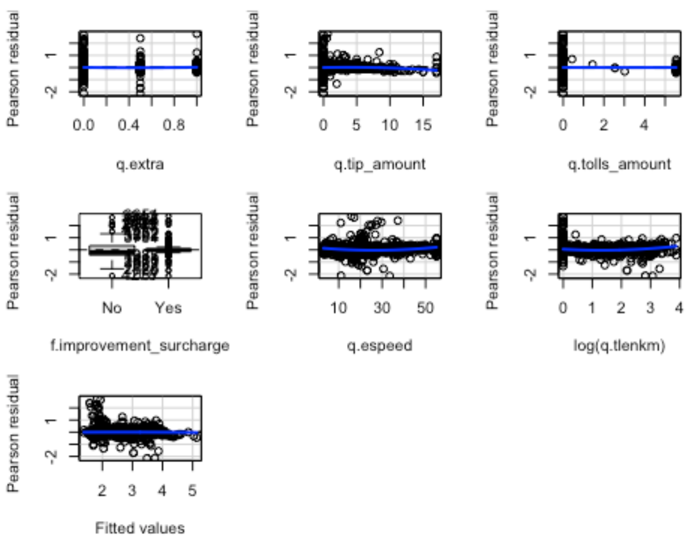
(2) Factors

```
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)
## 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 ***
## 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
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

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
```

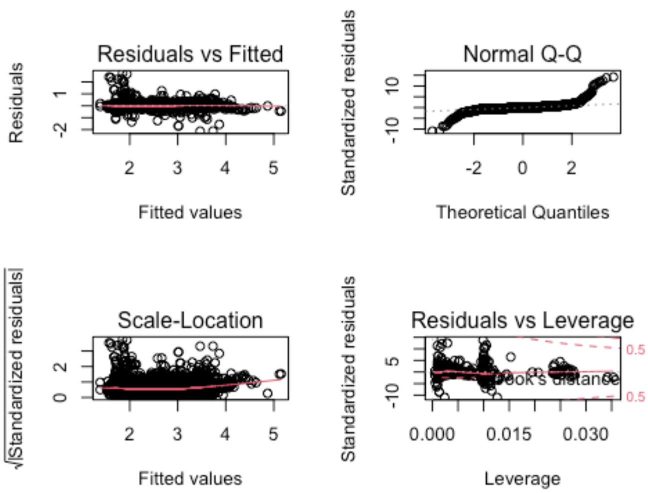
We can see that the best approach is the model_10, so we are going to stick to it for now.

```
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
## ---

summary(model_12)
##
## 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 ***
##
## 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.

```r
par(mfrow=c(2,2));plot( model_12, id.n=0 );par(mfrow=c(1,1))
```

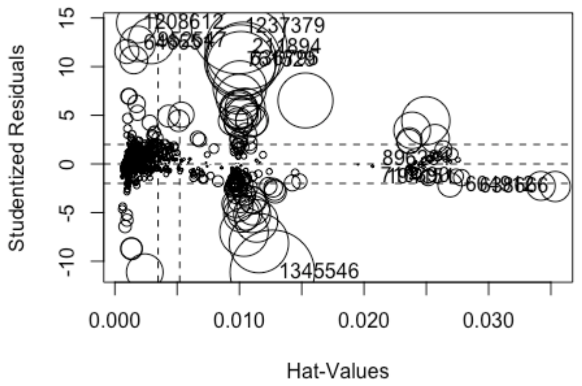


Looking at the results, we can say that:

- 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.

```r
influencePlot( model_12, id=c(list="noteworthy",n=5))
```



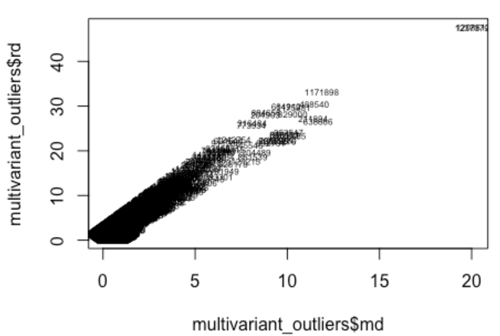
We see this model as a disaster. That is, we have a student waste of the order of 35. We can confirm that this is too much. We have to compare student waste with a normal standard. Therefore, we would say that the model we have so far is a model that has a serious waste problem.

Remove multivariant outliers to improve influence plot

Since we've realized that this should have been removed from the start, what we're going to do is put it at the beginning of the last deliverable in order to have a more consistent delivery. For now, however, we leave this section here so as not to have to change the entire delivery.

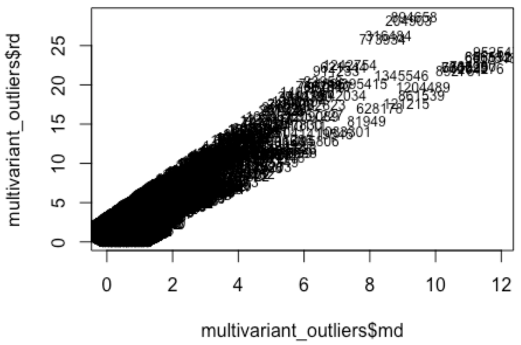
```
library(mvoutlier)
library(chemometrics)
multivariant_outliers <- Moutlier(df[, c(15,20)], quantile = 0.995)
```

```
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)
```



```r
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)
```

```r
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)
```

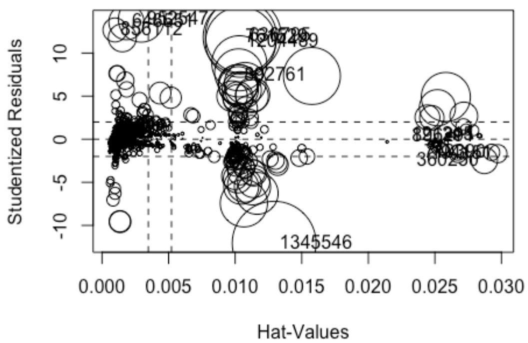


In order for this not to happen to us, we need to work on the variable q.tlenkm.

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

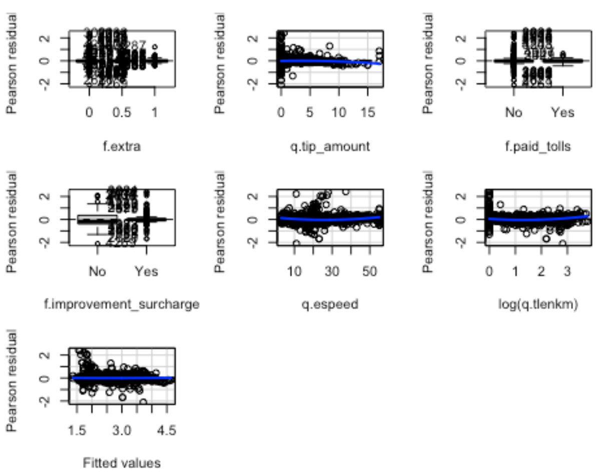
```r
model_13<-lm(log(target.total_amount)~ f.extra + q.tip_amount + f.paid_tolls +
f.improvement_surcharge + q.espeed + log(q.tlenkm),data=df); summary(model_13)
##
## Call:
## lm(formula = log(target.total_amount) ~ f.extra + q.tip_amount +
##     f.paid_tolls + f.improvement_surcharge + q.espeed + 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 ***
```

```
## ---
##
## 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))
```



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:

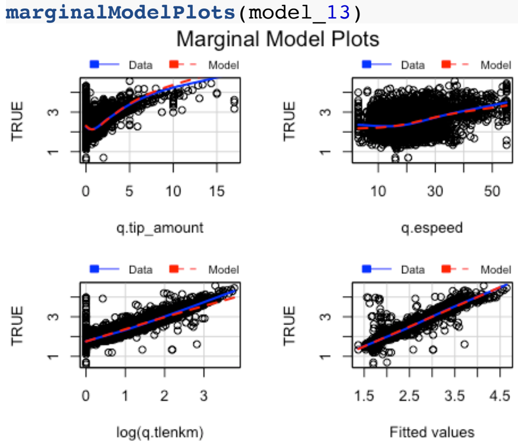
```
residualPlots(model_13)
```



```
##                          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
## ---
```

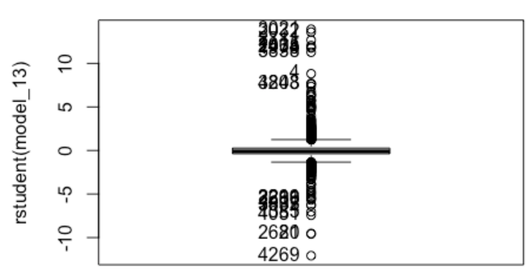
In the residualPlots, what we find is, for each factor, a boxplot of its categories and, for each quantitative variable, a pearson graph.
Let's use another tool to fully understand our model:

```
marginalModelPlots(model_13)
```



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)
// Annex - Q13
```

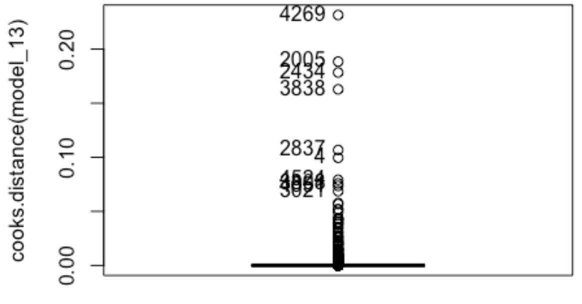
Let's see the strangest:

- 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)
// Annex - Q14
```

- 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)
##
## 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 ***
## 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
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.

(3) Add the main effects of factors and retain significant effects

```
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)
## 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
##
## 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
```

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 ***
## ---
```

We see that we haven't lost anything.

(4) Interactions

```
// Annex - Q5
```

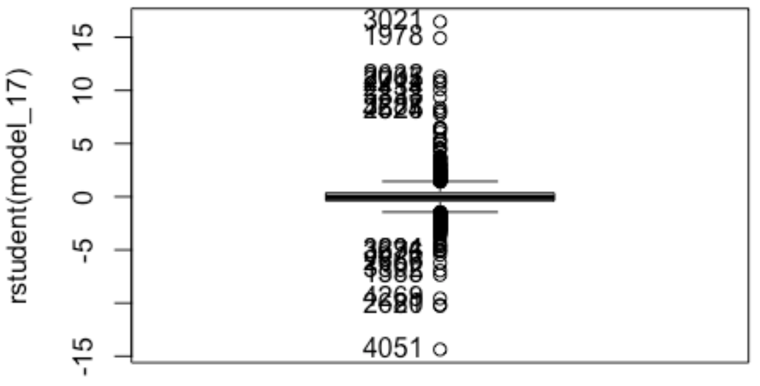
This method tells us that:

- 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

```
// Annex - Q6
```

Exhaustive

```
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
// Annex Q7
```

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.

(0) 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,]
```

(1) Numerical variables

Model 20

```
model_20 <- glm(target.tip_is_given~.,family =
"binomial",data=dffwork[,c("target.tip_is_given",vars_cexp)]);summary(model_20)
##
## 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
##
## (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
```

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_amoun
t+q.hour+q.espeed+q.tlenkm+q.traveltime ,family = "binomial",data=dffwork);summary(model_21)
## 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
## (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

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)
##
## 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
## (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
## ---
```

Now, we can do a step:

Model 23

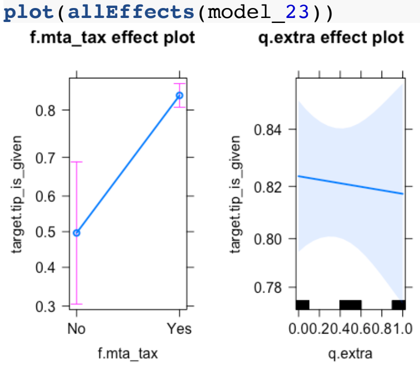
```
// Annex - Q8
```

Due to the fact that this model is really poor, 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)
##
## 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
##
## (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
```

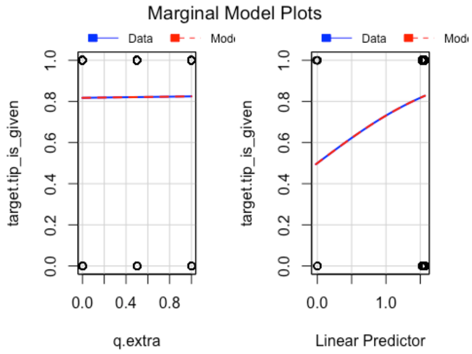
Understanding the model

```
plot(allEffects(model_23))
```



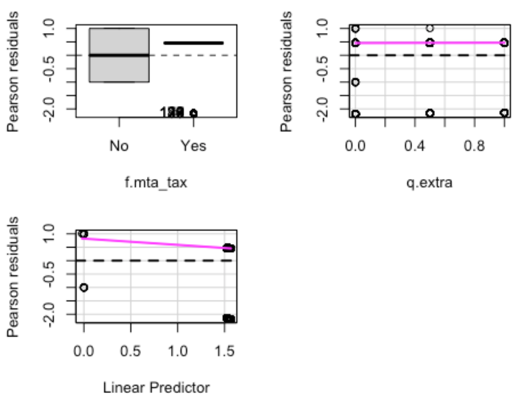
- For the f.mta_tax: we that if the value of the variable is “Yes”, it is more probable that the target.tip_is_given value will be “Yes” as well.
- For the q.extra: as we have said before, this variable does not really affect to the target, but we will include it in order to be able to do more plots.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

```
marginalModelPlots(model_23)
```



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

```
residualPlots(model_23)
```



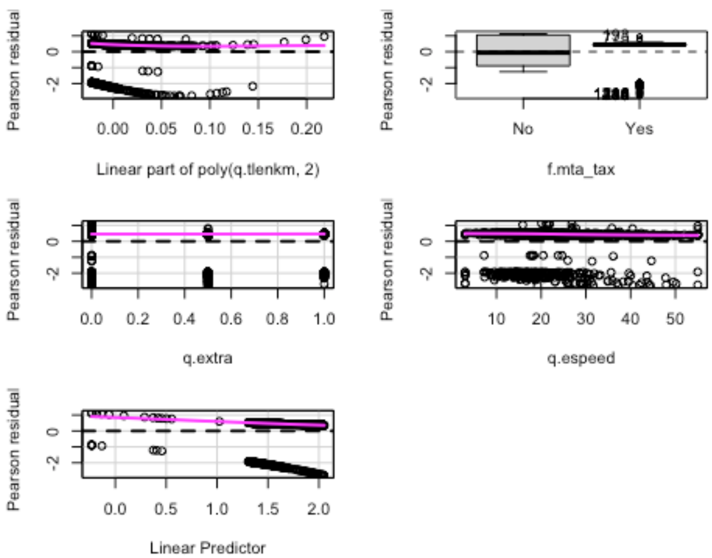

```
##           Test stat Pr(>|Test stat|)
## f.mta_tax
## q.extra      0.3308           0.5652
```

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)
##
## 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
##
## (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
```

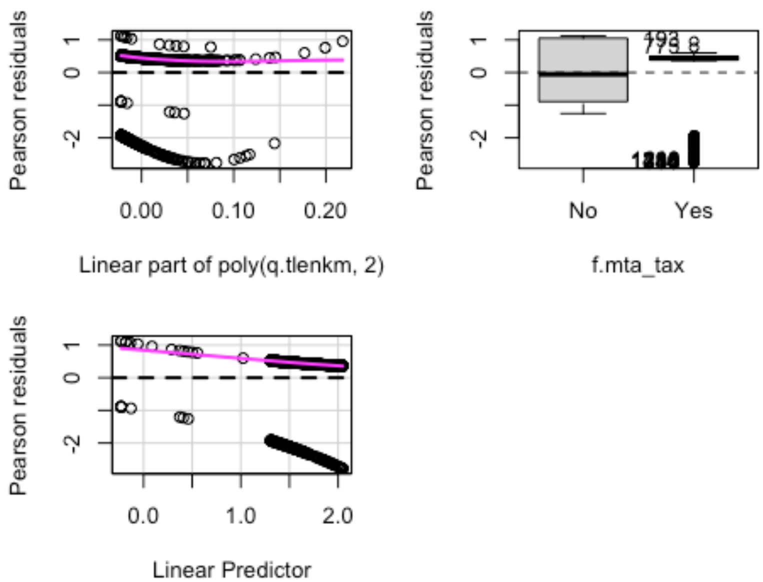
- q.tlenkm:
  - we see that the smoothe is plain, so it is ok.
  - the “weird” shapes that appear are because of the binery response model.
- q.extra:
  - we observe that the smoother is plain, so it is ok.
- q.espeed:
  - we see that the smoother is plain, so it is ok.
  - the “weird” shapes that appear 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
## ---
```

We have to ensure that we do not have any variable with a non significant net effect.
Thus, we are going to redo the model:

```
model_24 <- glm(target.tip_is_given~poly(q.tlenkm, 2)+f.mta_tax,family =
"binomial",data=dffwork); summary(model_24)
## 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 ***
## (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 ***
```

With Anova(model_24), we see that it is fulfilled.

(2) Factors

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)
##
## 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 ***
## (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.

(3) Add the main effects of factors and retain significant effects

We decide to keep with the model_25.

(4) Interactions

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:

```
// Annex - Q9
```

We remove the non significant variables:

```
// Annex - Q10
```

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)
## 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
## (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
## ---
```

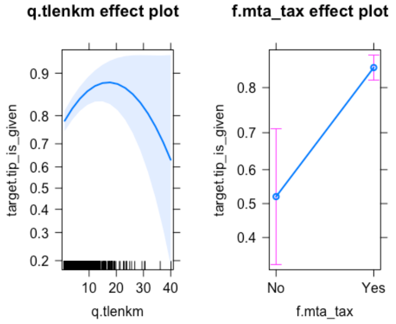
We do a comparison:

```
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.

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)+
model_factors_1_step <- step(model_factors_1, k=log(nrow(dffwork)))
## [. . .]
##              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="binom
ial",data=dffwork); summary(model_factors_2)
model_factors_2_step <- step(model_factors_2, k=log(nrow(dffwork)))
## [. . .]
##              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="bin
omial",data=dffwork); summary(model_factors_3)
model_factors_3_step <- step(model_factors_3, k=log(nrow(dffwork)))
## [. . .]
```

```
##              Df Deviance    AIC
## <none>            1385.0 1399.6
## - f.mta_tax  1   1397.9 1405.2
```

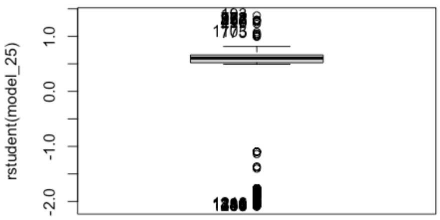
We stick with what we had.

Conclusion: we stick with the idea that the best model is model_25.

Now, we are going to do some diagnosis:

(5) Diagnosis

```
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)
// Annex - Q11
```

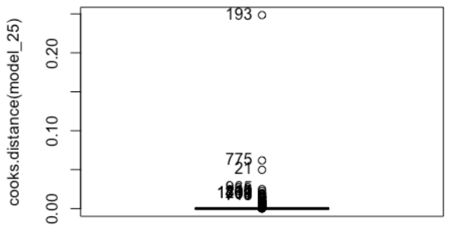
They tend to:

- have rate=Rate-Other
- be in the same location (they have very similar latitudes and longitudes)

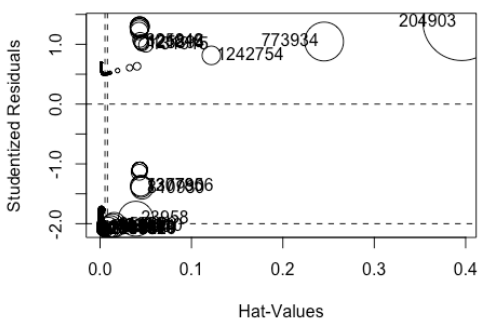
- have extra=0
- don’t have mta_tax
- be at night
- be one passenger
- be long (distance) but short (time)

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))
```



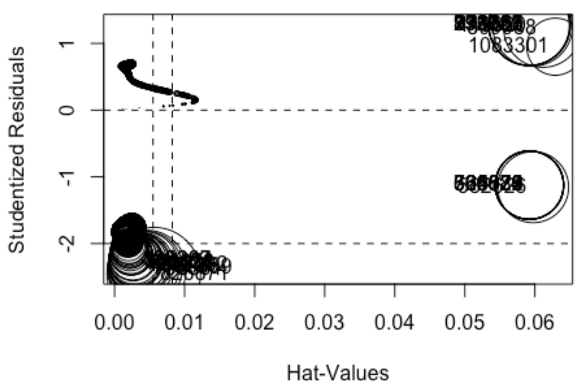

```
// Annex - Q12
```

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)
## 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 **
##
## (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))
```




```
# interaccions dobles entre factors
model_factors_5 <- 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_5)
model_factors_5_step <- step(model_factors_5, k=log(nrow(dffwork)))
## [. . .]
##                     Df Deviance    AIC
## <none>                   1285.6 1314.7
## - f.mta_tax          1   1294.0 1315.9
## - poly(q.tlenkm, 2)  2   1321.1 1335.7
```

We stick with what we had.

```r
# interaccions dobles entre factor-numèrica
model_factors_6 <-
glm(target.tip_is_given~(poly(q.tlenkm,2)+q.extra)*(f.mta_tax+f.vendor_id+f.espeed),family="binom
ial",data=dffwork); summary(model_factors_6)
model_factors_6_step <- step(model_factors_6, k=log(nrow(dffwork)))
## [. . .]
##                      Df Deviance    AIC
## <none>                    1285.6 1314.7
## - f.mta_tax           1   1294.0 1315.9
## - poly(q.tlenkm, 2)  2   1321.1 1335.7
```

We stick with what we had.

```r
# interaccions dobles entre factor-numèrica + dobles entre factors
model_factors_7 <-
glm(target.tip_is_given~(poly(q.tlenkm,2)+q.extra)*(f.mta_tax+f.vendor_id+f.espeed)^2,family="bin
omial",data=dffwork); summary(model_factors_7)
model_factors_7_step <- step(model_factors_7, k=log(nrow(dffwork)))
## [. . .]
##                      Df Deviance    AIC
## <none>                    1285.6 1314.7
## - f.mta_tax           1   1294.0 1315.9
## - poly(q.tlenkm, 2)  2   1321.1 1335.7
```

We stick with what we had.

Confusion Table

```r
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%.

Annex

Q1

```
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
```

Q2

```
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
```

Q3

```
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
```

Q4

```
sel2<-which(hatvalues(model_5)>5*length(model_5$coefficients)/nrow(df));sel2;length(sel2)
##    3060   14314   23958   36606   37238   41478   49078   71596   81949  101184
##      11      42      77     112     114     128     157     231     268     326
##  110353  194151  201926  202294  204903  216800  244971  250234  252056  267986
##     355     633     658     660     674     717     832     849     856     895
##  300524  316484  327762  329000  350170  360250  381123  394418  403814  404073
```

```
##     981    1025    1054    1057    1114    1152    1204    1244    1278    1280
##  415806  423307  444118  462782  486866  487457  488540  513170  529475  535352
##    1321    1352    1415    1479    1559    1562    1564    1626    1685    1700
##  560933  590161  604912  621420  621544  625503  638666  642379  644602  645141
##    1782    1853    1908    1962    1964    1972    2016    2025    2031    2033
##  659831  691705  710390  724424  731288  741591  751896  759052  771992  773934
##    2092    2211    2271    2325    2344    2383    2417    2435    2474    2480
##  777271  785532  821975  832751  861539  871576  881540  886530  894658  896291
##    2493    2515    2612    2646    2728    2756    2793    2816    2852    2858
##  911233  950707  965349  975103 1010826 1014307 1016299 1051194 1058632 1076485
##    2898    3020    3070    3101    3220    3231    3241    3358    3379    3426
## 1082823 1115959 1120203 1140092 1159509 1171898 1175981 1181893 1188969 1192516
##    3454    3558    3571    3638    3701    3735    3751    3783    3798    3812
## 1197687 1227019 1227021 1242754 1254924 1261276 1281722 1282165 1330280 1342604
##    3828    3910    3911    3960    3997    4016    4076    4080    4228    4273
## 1347654 1354552 1354822 1381927 1393691 1396114 1407546 1419545
##    4287    4308    4310    4403    4445    4458    4495    4533
## [1] 108
```

Q5

```
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 -17230
## - 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 -17182
## - 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 -17268
## - 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 -17294
## - 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
```

Q6

```
## 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
```

Q7

```r
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
```

Q8

```
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)  -6.327e-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
```

Q9

```
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
```

Q10

```
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
```

Q11

```
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
```

Q12

```
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
```

Q13

```
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
```

Q14

```
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
```