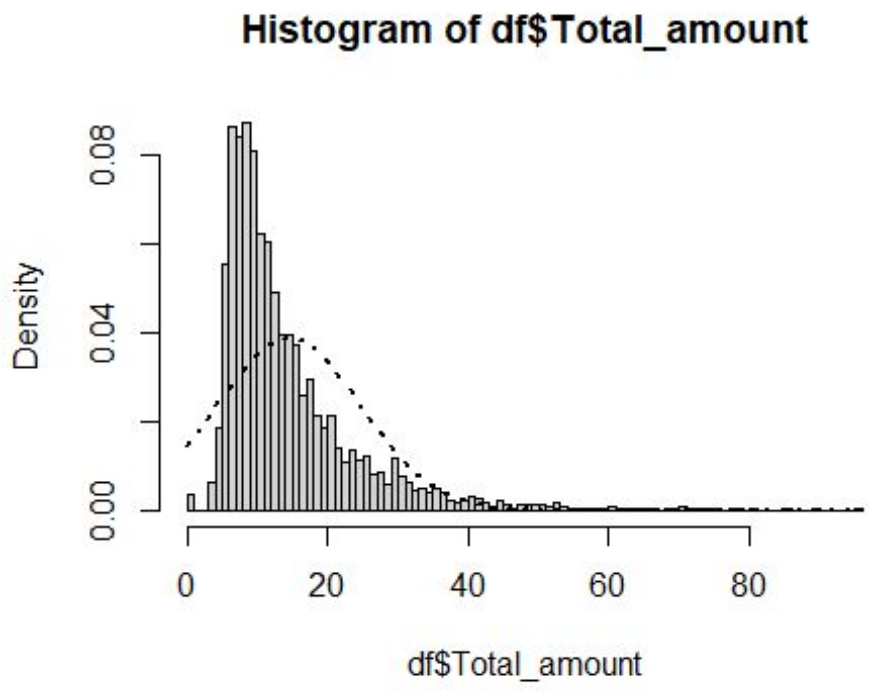
Deliverable III

Carles Capilla Cànovas
Jesús Molina Roldán

Multiple Linear Regression Model

En esta sección se construye un modelo lineal para una variable target numérica, Total_amount.

Multivariate Analysis



Antes de poder hacer la modelización hay que mirar si la variable respuesta sigue una distribución normal. Para realizar la comprobación utilizamos diferentes indicadores. Si vemos el histograma del Total amount junto a la correcta distribución normal, vemos que el los histogramas no se solapan, lo que significa que tenemos que normalizar la variable.

```
shapiro.test(df$Total_amount)
## W = 0.76853, p-value < 2.2e-16
```

Si realizamos el test de normalidad Shapiro-Wilk, observamos que la H0 puede ser rechazada al mostrar un p-value muy inferior a 0.05. Lo que significa que los datos no siguen una distribución normal.

```
skewness(df$Total_amount)

## [1] 2.485124
```

Si realizamos un test de simetría como Skewness, vemos que nos devuelve un valor diferente a 0. Por lo tanto, los datos son asimétricos y como consecuencia, no siguen una distribución normal. También podemos ver que el valor es superior a 0 por lo tanto los

datos son right-skewed lo que significa que las observaciones presentan una larga cola de observaciones por la derecha.

```
kurtosis(df$Total_amount)
## [1] 11.95862
```

Si computamos la curtosis de la variable Total amount, observamos que es superior a 3, lo que significa que no sigue una distribución normal. Tras ver todos estos argumentos, vemos que los datos no siguen una distribución normal.

Es por eso que el método más apropiado para calcular la correlación deba ser a partir de Spearman.

```
> round(cor(df[,c("Total_amount",vars_cexp)], method="pearson"),dig=2)
```

	Total_amount	Passenger_count	tlenkm	Fare_amount	espeed	Tip_amount	Tolls_amount	lpep_pickup_time	traveltime	distHaversine
Total_amount	1.00	0.02	0.89	0.96	0.43	0.62	0.31	-0.04	0.46	0.81
Passenger_count	0.02	1.00	0.02	0.02	0.01	-0.01	0.02	-0.02	0.00	0.01
tlenkm	0.89	0.02	1.00	0.90	0.58	0.45	0.27	-0.06	0.45	0.90
Fare_amount	0.96	0.02	0.90	1.00	0.44	0.48	0.24	-0.05	0.47	0.83
espeed	0.43	0.01	0.58	0.44	1.00	0.23	0.20	-0.12	-0.01	0.56
Tip_amount	0.62	-0.01	0.45	0.48	0.23	1.00	0.19	-0.02	0.23	0.44
Tolls_amount	0.31	0.02	0.27	0.24	0.20	0.19	1.00	-0.03	0.08	0.27
lpep_pickup_time	-0.04	-0.02	-0.06	-0.05	-0.12	-0.02	-0.03	1.00	0.40	-0.06
traveltime	0.46	0.00	0.45	0.47	-0.01	0.23	0.08	0.40	1.00	0.41
distHaversine	0.81	0.01	0.90	0.83	0.56	0.44	0.27	-0.06	0.41	1.00

Si vemos los resultados, observamos que la distancia recorrida, la tarifa abonada, la duración del viaje y la distancia Haversine son las variables más correlacionadas con el target numérico, Total amount.

Use explanatory numeric variables

Para el modelo inicial podríamos elegir aquellas variables más correlacionadas. A pesar de todo, como tenemos pocas variables explicativas, decidimos coger todas las variables explicativas.

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.383185   0.102835  13.451  < 2e-16 ***
## Passenger_count   0.031776   0.025303   1.256  0.20923
## tlenkm            0.224812   0.017958  12.519  < 2e-16 ***
## Fare_amount       0.951982   0.007468 127.483  < 2e-16 ***
## espeed           -0.015837   0.003893  -4.069 4.80e-05 ***
## Tip_amount        1.013289   0.015632  64.821  < 2e-16 ***
## Tolls_amount      1.009723   0.039853  25.336  < 2e-16 ***
## lpep_pickup_time  0.011693   0.003936   2.971  0.00299 **
## traveltime       -0.002567   0.002172  -1.182  0.23744
## distHaversine    -0.147172   0.020475  -7.188 7.54e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.893 on 4990 degrees of freedom
## Multiple R-squared:  0.9663, Adjusted R-squared:  0.9663
## F-statistic: 1.591e+04 on 9 and 4990 DF,  p-value: < 2.2e-16
```

Como podemos ver el modelo tiene una explicatividad del 96’63% de la variabilidad del target. A pesar de todo, hay variables como el passenger count y el traveltime que no son significativas ya que, como podemos ver en el test Anova tienen una proporción de la distribución t superior a 0.05.

```
> vif(m)  # Check association between explanatory vars
```

Passenger_count	tlenkm	Fare_amount	espeed	Tip_amount	Tolls_amount	lpep_pickup_time	traveltime	distHaversine
1.001878	9.657089	5.946645	1.839423	1.322309	1.094175	1.320458	1.905585	5.344283

A partir de la variance inflation factors vemos la asociación entre las variables explicativas. Podemos observar que las variables tlenkm, Fare amount y la distancia Haversine están muy correlacionadas. También podemos ver que la distancia fare amount y la distancia Haversine tienen un valor similar de inflación lo que nos hace creer que hay una gran relación entre ellas.

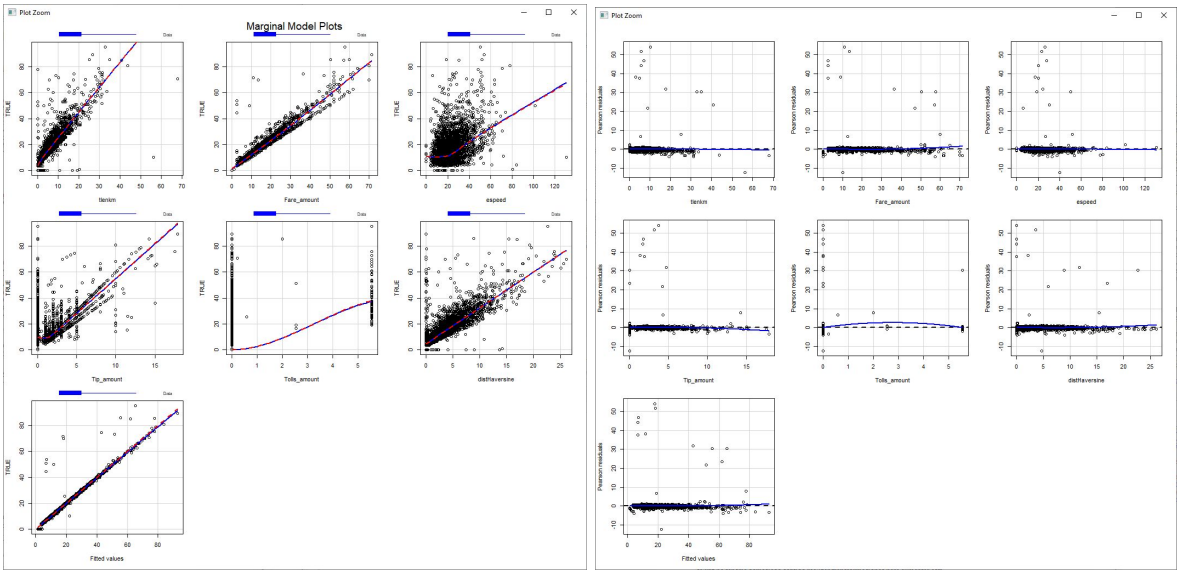
Tras haber analizado todas las variables que no aportan mucho en el modelo decidimos eliminar las variables passenger count y el traveltime de este ya que eran las dos primeras variables que elimina el vif y las rechazadas por la hipótesis nula. Podríamos eliminar la variable Fare_amount, ya que es prácticamente la misma que nuestro target, sin embargo decidimos no eliminarla. Aún así esperaremos a ver los resultados que nos muestra el step para acabar de concretar si eliminamos del modelo alguna otra variable.

```
m1 <- step( m, k=log(nrow(df)) )

## Step:  AIC=6439.54
## Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount +
##     distHaversine
##
##                 Df Sum of Sq   RSS     AIC
## <none>                       17911  6439.5
## - espeed         1        63 17974  6448.7
## - distHaversine  1       192 18103  6484.3
## - tlenkm         1       565 18476  6586.3
## - Tolls_amount   1      2307 20218  7036.7
## - Tip_amount     1     15047 32958  9480.1
## - Fare_amount    1     58799 76710 13704.1

summary(m1)
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.574884   0.079479  19.815  < 2e-16 ***
## tlenkm         0.222321   0.017717  12.548  < 2e-16 ***
## Fare_amount    0.950798   0.007427 128.027  < 2e-16 ***
## espeed        -0.015307   0.003644  -4.200 2.71e-05 ***
## Tip_amount     1.012951   0.015640  64.765  < 2e-16 ***
## Tolls_amount   1.010725   0.039858  25.358  < 2e-16 ***
## distHaversine -0.149401   0.020423  -7.315 2.98e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.894 on 4993 degrees of freedom
## Multiple R-squared:  0.9662, Adjusted R-squared:  0.9662
## F-statistic: 2.382e+04 on 6 and 4993 DF,  p-value: < 2.2e-16
```

Como podemos ver que si aplicamos el Stepwise Algorithm Akaike según el Akaike information criterion (AIC), vemos que nos elimina las variables explicativas traveltime, Passenger_count y lpep_pickup_time. La calidad del criterio de AIC se nos queda en 6439.54.

Al realizar un summary del modelo resultante vemos como la explicación de la variabilidad del target, una vez eliminadas las variables del anterior modelo, se mantiene casi igual pasando de 96'63 a 96'62. Consideramos el nuevo modelo m1 ya que obtenemos la misma explicación prácticamente, simplificando en 3 variables el nuevo modelo.

El modelo resultante tendría la siguiente predicción:
Y=1.57+0.22*tlenkm+0.95*Fare_amount-0.015*espeed+1.01*Tip_amount+1.01*Tolls_amount-0.15*distHaversine

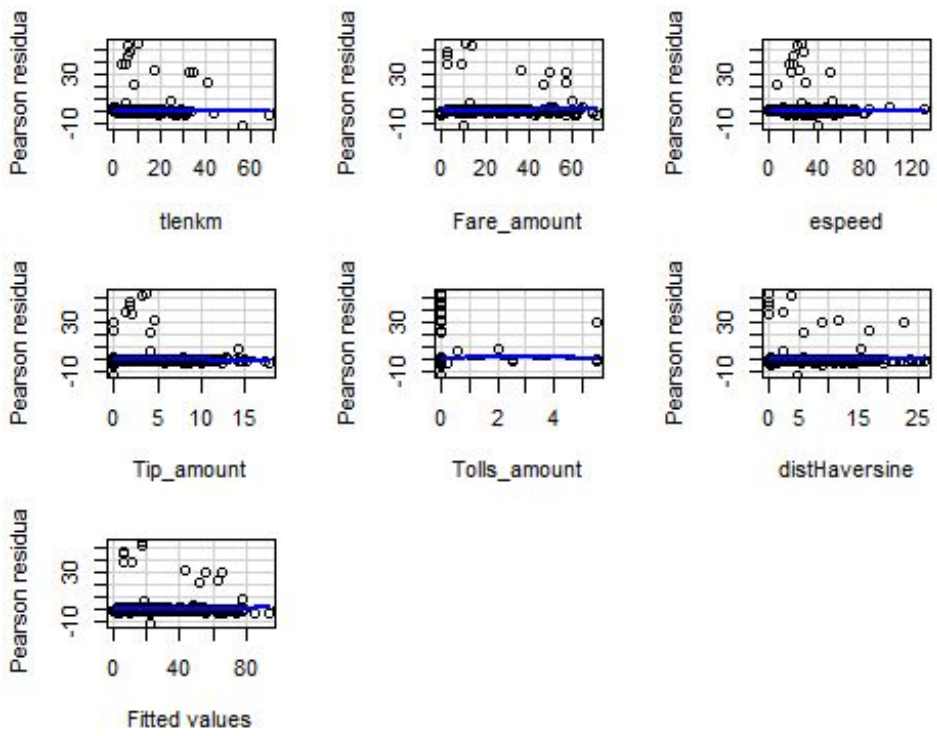


En la imagen superior podemos ver los resultados que obtenemos del modelo de regresión. Como vemos la distancia recorrida en km, el importe de la tarifa y la distancia de Haversine siguen una regresión lineal perfecta. Aun así podemos ver como las observaciones no se distribuyen de una manera homogénea por todo el rango de valores posibles sin llegar, por eso, a mostrar patrones que puedan hacernos considerar tratarlas. A destacar sobretodo el Tolls amount que al ser una variable con valores enteros y más cercana a ser considerada un factor, vemos como no se ajustan sus observaciones al modelo de regresión pero es perfectamente normal debido a sus propiedades.

Al disponer de una explicación de la variabilidad del target por el modelo tan alta y una igualdad entre la predicción de las variables y las observaciones que define el modelo, talvez no tendría demasiado sentido realizar transformaciones para aumentar este Multiple R-squared pero si para reducir el Residual Standard error del modelo.

Si observamos los residuos encontramos algún valor negativo y varios valores fuera de la linea residual. Además encontramos que los residuos no muestran una tendencia normalizada debido a las observaciones de la derecha del gráfico, muy alejadas de lo que sería considerada la normal del target. Una de las modificaciones que hicimos es modificar la escala de la variable target a logarítmica pero no observamos mejoras al respecto.

Transformations




```
m2 <-lm(Total_amount ~  tlenkm + Fare_amount + espeed + Tip_amount + poly(Tolls_amount,2) +
distHaversine,data=df)
summary(m2)
```

```
##
## Call:
## lm(formula = Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount +
##     poly(Tolls_amount, 2) + distHaversine, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.163  -0.394  -0.050   0.184  54.145
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.676645   0.079923  20.978  < 2e-16 ***
## tlenkm                   0.220981   0.017712  12.476  < 2e-16 ***
## Fare_amount              0.950796   0.007422 128.111  < 2e-16 ***
## espeed                  -0.015660   0.003644  -4.297 1.76e-05 ***
## Tip_amount               1.011293   0.015642  64.653  < 2e-16 ***
## poly(Tolls_amount, 2)1 50.297952   1.978935  25.417  < 2e-16 ***
## poly(Tolls_amount, 2)2 -5.222973   1.898610  -2.751  0.00596 **
## distHaversine          -0.147326   0.020424  -7.213 6.27e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.893 on 4992 degrees of freedom
## Multiple R-squared:  0.9663, Adjusted R-squared:  0.9663
## F-statistic: 2.045e+04 on 7 and 4992 DF,  p-value: < 2.2e-16
```

```
##                      Test stat Pr(>|Test stat|)
## tlenkm                 -0.6848        0.4934854
```

```
## Fare_amount              3.8600        0.0001148 ***
## espeed                  -0.3472        0.7284631
## Tip_amount              -3.4021        0.0006740 ***
## poly(Tolls_amount, 2)
## distHaversine            2.9020        0.0037236 **
## Tukey test               1.7407        0.0817347 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

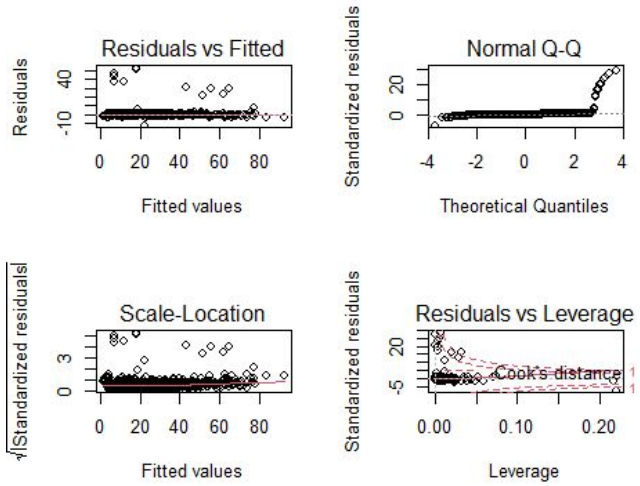
```
anova(m1,m2)
```

```
## Analysis of Variance Table
##
## Model 1: Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount +
##     distHaversine
## Model 2: Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + poly(Tolls_amount,
##     2) + distHaversine
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1   4993 17911
## 2   4992 17884  1    27.112 7.5677 0.005964 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Al ver los gráficos obtenidos por el residualplots vemos como la única variable que sufre un patrón respecto el smoother sería la de tolls amount así que probaremos a aplicarle alguna transformación.

Una vez usado un polinomio de grado dos para esta transformación vemos como la línea de los residuos se ajusta más al smoother pero,al analizar el summary vemos como la explicación de la variabilidad no varía apenas y al usar el anova(m1,m2) vemos como no debemos considerar esta transformación.

Diagnostics Linear Regression using explanatory numeric variables



Si observamos los residuos del m1 encontramos algún valor negativo y varios valores fuera de la línea residual. Además encontramos que los residuos no muestran una tendencia normalizada debido a las observaciones de la derecha del gráfico, muy alejadas de lo que sería considerada la normal del target. Una de las modificaciones que hicimos es modificar la escala de la variable target a logarítmica pero no observamos mejoras al respecto.

```r
l <- which(df$Total_amount == 0 )
df[l,'Total_amount'] <- 0.0001
m3 <- lm( log(Total_amount) ~ tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount+ distHaversine,
data=df)
summary(m3)
```

```
##
## Call:
## lm(formula = log(Total_amount) ~ tlenkm + Fare_amount + espeed +
##     Tip_amount + Tolls_amount + distHaversine, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.8307  -0.0533   0.0871   0.1761   3.3071
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.562062   0.027989  55.810   <2e-16 ***
## tlenkm        -0.064713   0.006239 -10.372   <2e-16 ***
## Fare_amount    0.083073   0.002615  31.764   <2e-16 ***
## espeed         0.003798   0.001283   2.959   0.0031 **
## Tip_amount     0.052982   0.005508   9.619   <2e-16 ***
## Tolls_amount   0.017401   0.014036   1.240   0.2151
## distHaversine  0.016996   0.007192   2.363   0.0182 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.667 on 4993 degrees of freedom
## Multiple R-squared:  0.4343, Adjusted R-squared:  0.4336
## F-statistic: 638.8 on 6 and 4993 DF,  p-value: < 2.2e-16
```

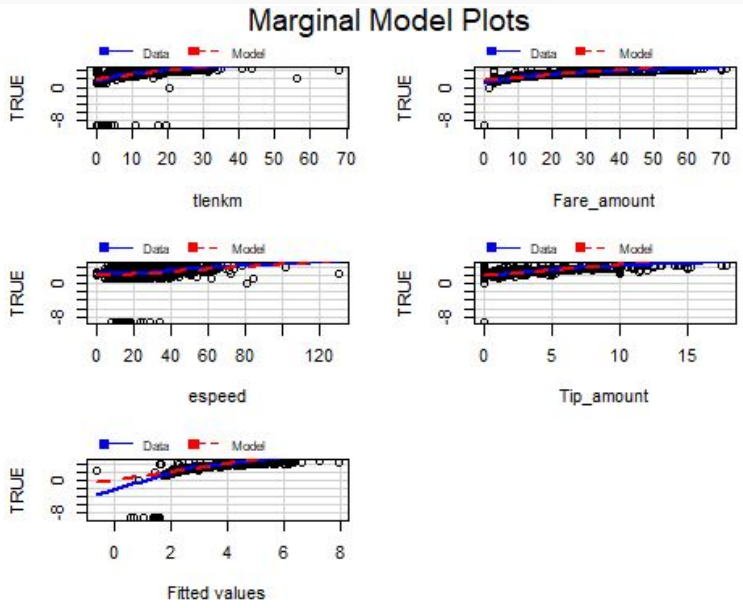
Con el objetivo de mejorar el modelo y buscar la normalidad de la variable target Total amount, le aplicamos el logaritmo. Aunque antes de aplicarlo debemos eliminar las observaciones que contengan un 0 y es por eso que les asignamos un valor de 0.0001.

Como podemos ver en el summary las variables espeed, Tolls_amount y dist_haversine, debido a su p_value deberían ser eliminadas del modelo ya que este es muy superior a 0.05, es decir que no aportan información significativa al mismo. Aún así, vemos que el modelo con el logaritmo del importe total solo explica una variabilidad de cerca del 43'43%. Probaremos a eliminar del modelo las variables mencionadas pero esto como mucho mantendrá la explicación de la variabilidad del target.

```r
m4 <- lm( log(Total_amount) ~  tlenkm + Fare_amount + espeed + Tip_amount, data=df)
summary(m4)
```

```
##
## Call:
## lm(formula = log(Total_amount) ~ tlenkm + Fare_amount + espeed +
##     Tip_amount, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.8336  -0.0544   0.0861   0.1790   2.8938
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.556781   0.027866  55.866  < 2e-16 ***
## tlenkm      -0.056365   0.005284 -10.667  < 2e-16 ***
## Fare_amount  0.083872   0.002594  32.331  < 2e-16 ***
## espeed       0.004358   0.001267   3.439 0.000588 ***
## Tip_amount   0.054492   0.005479   9.946  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.6673 on 4995 degrees of freedom
## Multiple R-squared:  0.4334, Adjusted R-squared:  0.433
## F-statistic: 955.3 on 4 and 4995 DF,  p-value: < 2.2e-16

Anova(m4)

## Anova Table (Type II tests)
##
## Response: log(Total_amount)
##              Sum Sq   Df  F value    Pr(>F)
## tlenkm        50.68    1  113.793 < 2.2e-16 ***
## Fare_amount  465.54    1 1045.325 < 2.2e-16 ***
## espeed         5.27    1   11.828 0.0005882 ***
## Tip_amount    44.05    1   98.918 < 2.2e-16 ***
## Residuals   2224.53 4995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

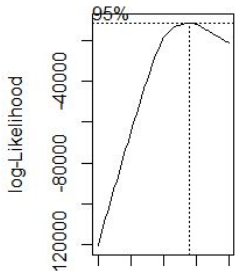


Tras realizar las mejoras vemos que el modelo reduce el residual standard error respecto a m1. Aun así seguimos teniendo en el modelo m1 una mejor explicación de la variabilidad del target, lo que resulta en un mejor modelo del mismo.

Además vemos como gracias al plot del marginalModel como todas las predicciones de color azul ya no siguen la distribución del modelo lo que nos hace descartar estas transformaciones.

```
BIC(m1,m2,m4)

##    df      BIC
## m1  8 20637.44
## m2  9 20638.39
## m4  6 10191.02
```

Si calculamos Akaike en los modelos, vemos también que m1 es mejor en comparación a los otros.



Si aplicamos una Box-Cox power transformation a nuestros datos, vemos que con el primer modelo tiene un parámetro inferior a 1 pero

casi igual, lo que sugiere que no deberíamos aplicar ninguna transformación, o como mucho probar con una raíz cuadrada ya que ya hemos descartado el logaritmo.

```
m4<-lm(sqrt(Total_amount) ~  tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount+
distHaversine,data=df)
summary(m4)
```

```
##
## Call:
## lm(formula = sqrt(Total_amount) ~ tlenkm + Fare_amount + espeed +
##     Tip_amount + Tolls_amount + distHaversine, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2810 -0.1128  0.0402  0.1506  4.8103
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.1663437  0.0124569 173.908  < 2e-16 ***
## tlenkm        -0.0079799  0.0027769  -2.874  0.00407 **
## Fare_amount    0.1105970  0.0011640  95.018  < 2e-16 ***
## espeed        -0.0015027  0.0005712  -2.631  0.00854 **
## Tip_amount     0.1083534  0.0024513  44.202  < 2e-16 ***
## Tolls_amount   0.0703380  0.0062470  11.259  < 2e-16 ***
## distHaversine  0.0216390  0.0032009   6.760 1.54e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2968 on 4993 degrees of freedom
## Multiple R-squared:  0.9339, Adjusted R-squared:  0.9338
## F-statistic: 1.175e+04 on 6 and 4993 DF,  p-value: < 2.2e-16
```

```
summary(m1)
```

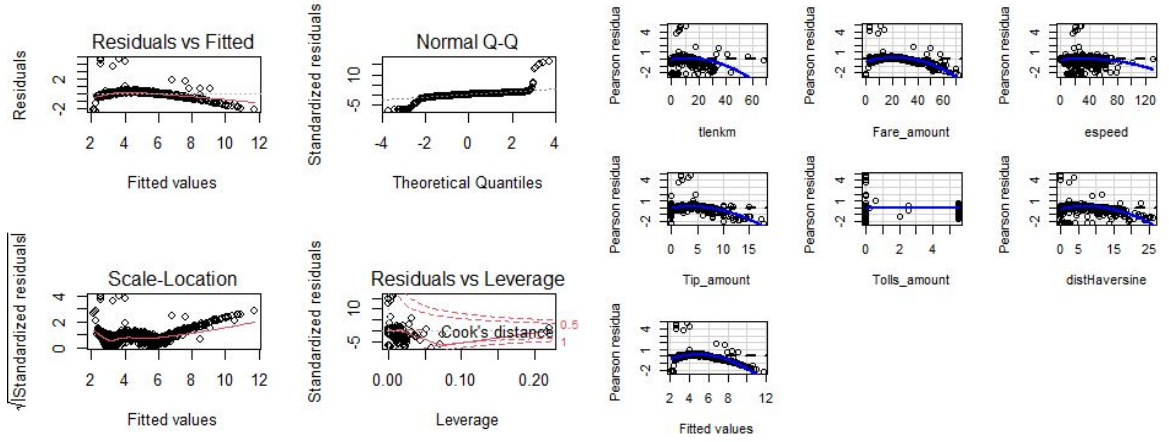
```
##
## Call:
## lm(formula = Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount +
##     Tolls_amount + distHaversine, data = df[, c("Total_amount",
##     vars_cexp)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.235  -0.393  -0.055   0.183  54.123
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.574884   0.079479  19.815  < 2e-16 ***
## tlenkm         0.222321   0.017717  12.548  < 2e-16 ***
## Fare_amount    0.950798   0.007427 128.027  < 2e-16 ***
## espeed        -0.015307   0.003644  -4.200 2.71e-05 ***
## Tip_amount     1.012951   0.015640  64.765  < 2e-16 ***
## Tolls_amount   1.010725   0.039858  25.358  < 2e-16 ***
## distHaversine -0.149401   0.020423  -7.315 2.98e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.894 on 4993 degrees of freedom
## Multiple R-squared:  0.9662, Adjusted R-squared:  0.9662
## F-statistic: 2.382e+04 on 6 and 4993 DF,  p-value: < 2.2e-16
```

```
Anova(m4)
```

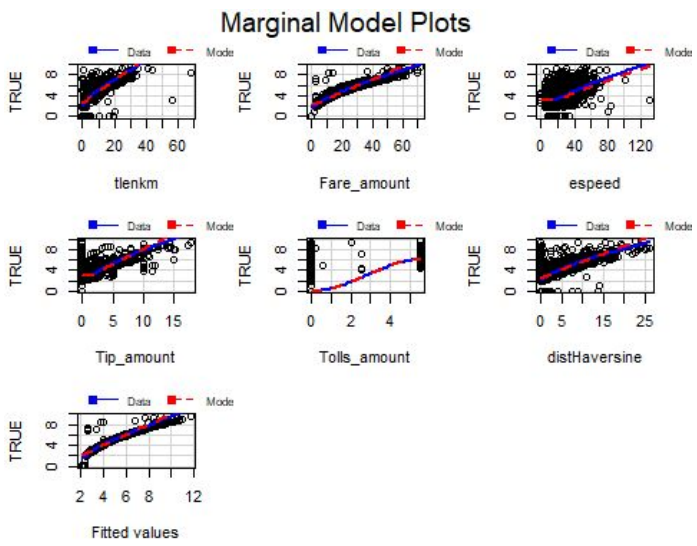
```
## Anova Table (Type II tests)
##
## Response: sqrt(Total_amount)
##               Sum Sq   Df   F value    Pr(>F)
## tlenkm          0.73    1    8.2583  0.004074 **
## Fare_amount   795.57    1 9028.3324 < 2.2e-16 ***
## espeed          0.61    1    6.9212  0.008544 **
## Tip_amount    172.17    1 1953.7944 < 2.2e-16 ***
## Tolls_amount   11.17    1  126.7751 < 2.2e-16 ***
## distHaversine   4.03    1   45.7002 1.536e-11 ***
## Residuals     439.98 4993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vif(m4)
```

```
##        tlenkm   Fare_amount        espeed    Tip_amount  Tolls_amount
##      9.385882      5.872915      1.609990      1.321788      1.092833
## distHaversine
##      5.309666
```




```
##               Test stat Pr(>|Test stat|)
## tlenkm          -30.7479          <2e-16 ***
## Fare_amount     -63.0933          <2e-16 ***
## espeed           -8.6098          <2e-16 ***
## Tip_amount      -33.4372          <2e-16 ***
## Tolls_amount      0.8415          0.4001
## distHaversine   -40.1881          <2e-16 ***
## Tukey test      -67.4183          <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

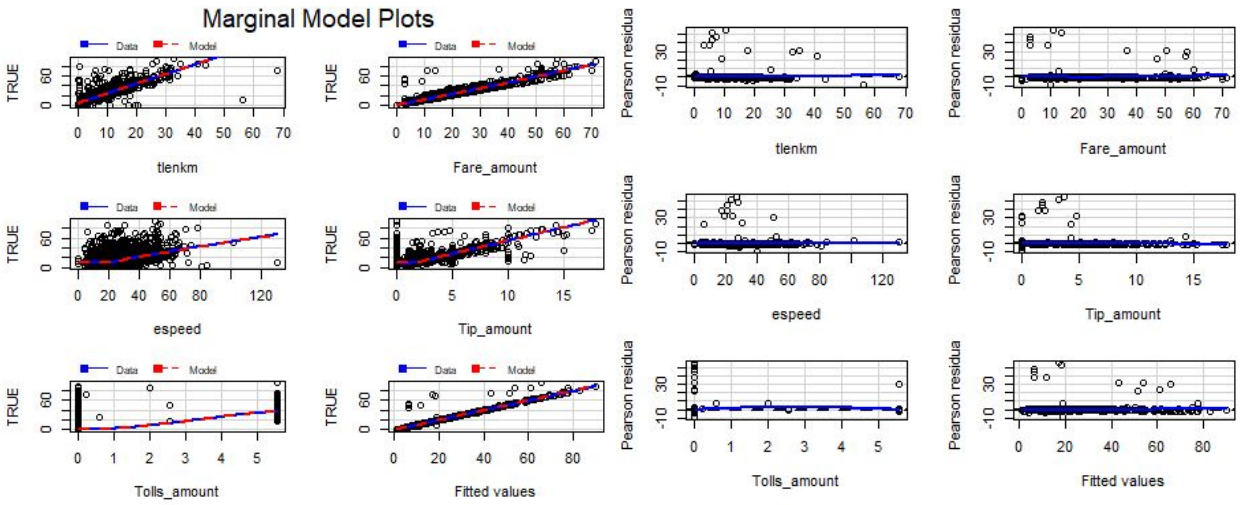


Al probar de aplicar la raíz cuadrada a la variable Total_amount vemos como la explicación de la variabilidad se reduce a 93.39%, lo que al haber complicado el modelo, nos indicaría no escogerlo como óptimo. Además vemos que para el residualplots las predicciones dejan de ajustarse a sus smoother creando patrones de términos cuadráticos, consecuencia de haber aplicado la raíz en el target.

Una vez aplicado en vif, por eso, vemos cómo Fare Amount y distHaversine están claramente correlacionadas así que probaremos a descartar una de ellas en el siguiente modelo.

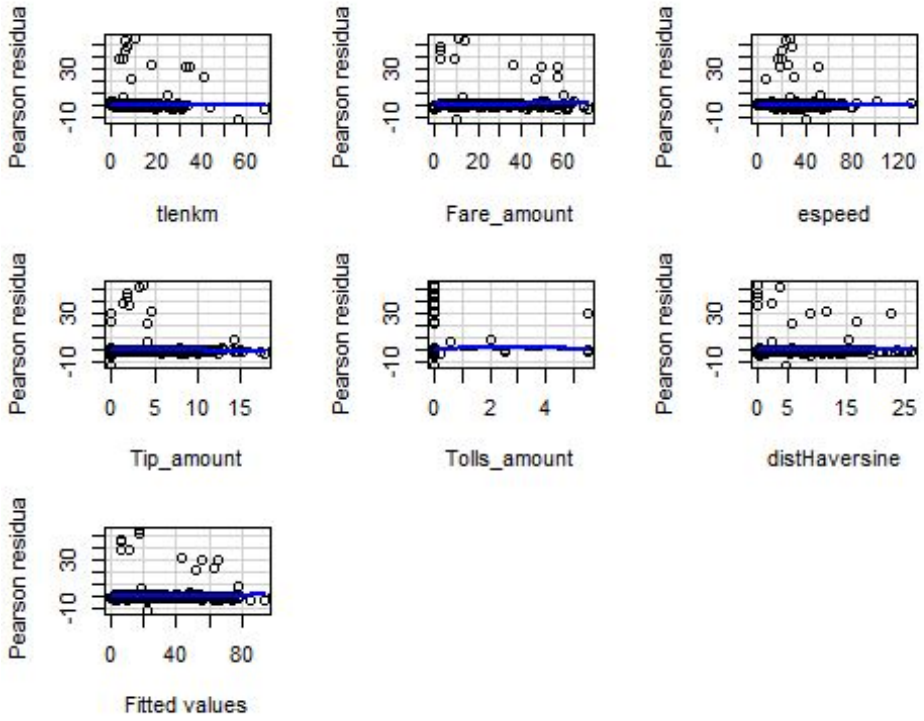
```
m5<-lm(Total_amount ~  tlenkm + espeed + Tip_amount + Tolls_amount+ distHaversine,data=df)
summary(m5)

##
## Call:
## lm(formula = Total_amount ~ tlenkm + espeed + Tip_amount + Tolls_amount +
##     distHaversine, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -88.120  -1.031  -0.427   0.343  59.080
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.90624    0.14008  49.301  < 2e-16 ***
## tlenkm         1.69563    0.02788  60.827  < 2e-16 ***
## espeed        -0.11978    0.00735 -16.297  < 2e-16 ***
## Tip_amount     1.35734    0.03188  42.574  < 2e-16 ***
## Tolls_amount   0.96001    0.08247  11.640  < 2e-16 ***
## distHaversine  0.19326    0.04190   4.613 4.07e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.919 on 4994 degrees of freedom
## Multiple R-squared:  0.8555, Adjusted R-squared:  0.8553
## F-statistic:  5911 on 5 and 4994 DF,  p-value: < 2.2e-16

m6<-lm(Total_amount ~  tlenkm +Fare_amount + espeed + Tip_amount + Tolls_amount,data=df)
summary(m6)

##
## Call:
```

```
## lm(formula = Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount +
##     Tolls_amount, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.880 -0.404 -0.038  0.162 55.026
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.588970   0.079873  19.894  < 2e-16 ***
## tlenkm        0.153818   0.015119  10.174  < 2e-16 ***
## Fare_amount   0.943678   0.007401 127.505  < 2e-16 ***
## espeed       -0.019291   0.003622  -5.326 1.05e-07 ***
## Tip_amount    1.005206   0.015686  64.081  < 2e-16 ***
## Tolls_amount  0.995197   0.040010  24.873  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.904 on 4994 degrees of freedom
## Multiple R-squared:  0.9659, Adjusted R-squared:  0.9659
## F-statistic: 2.828e+04 on 5 and 4994 DF,  p-value: < 2.2e-16

summary(m1)

##
## Call:
## lm(formula = Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount +
##     Tolls_amount + distHaversine, data = df[, c("Total_amount",
##     vars_cexp)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.235  -0.393  -0.055   0.183  54.123
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.574884   0.079479  19.815  < 2e-16 ***
## tlenkm         0.222321   0.017717  12.548  < 2e-16 ***
## Fare_amount    0.950798   0.007427 128.027  < 2e-16 ***
## espeed        -0.015307   0.003644  -4.200 2.71e-05 ***
## Tip_amount     1.012951   0.015640  64.765  < 2e-16 ***
## Tolls_amount   1.010725   0.039858  25.358  < 2e-16 ***
## distHaversine -0.149401   0.020423  -7.315 2.98e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.894 on 4993 degrees of freedom
## Multiple R-squared:  0.9662, Adjusted R-squared:  0.9662
## F-statistic: 2.382e+04 on 6 and 4993 DF,  p-value: < 2.2e-16

anova(m6,m1)

## Analysis of Variance Table
##
## Model 1: Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount
## Model 2: Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount +
##     distHaversine
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1   4994 18103
## 2   4993 17911  1    191.96 53.511 2.98e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

marginalModelPlots(m6)
```




```
##              Test stat Pr(>|Test stat|)
## tlenkm          2.7892        0.0053033 **
## Fare_amount     6.0739         1.34e-09 ***
## espeed         -0.0586        0.9532750
## Tip_amount     -2.8502        0.0043864 **
## Tolls_amount   -3.0041        0.0026766 **
## Tukey test      3.6569        0.0002553 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residualPlots(m1)
```



```
##               Test stat Pr(>|Test stat|)
## tlenkm          -0.5294         0.596537
## Fare_amount      3.9819        6.933e-05 ***
## espeed          -0.3207         0.748461
## Tip_amount      -3.1162         0.001842 **
## Tolls_amount    -2.7509         0.005964 **
## distHaversine    2.8748         0.004059 **
## Tukey test       1.8956         0.058014 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Probamos primero a descartar el Fare_amount debido a que estará muy relacionada con nuestro target, pero vemos como sufrimos un descenso considerable hasta 85.55% de la explicación de la variabilidad por el modelo. En cambio, al quitar el distHaversine solo vemos un descenso de 96.62% a 96.59% reduciendo el uso de una variable respecto al modelo m1. Aplicando anova vemos como és buena opción escoger este nuevo modelo como óptimo.

```
m7<-lm(Total_amount ~  f.tlenkm +Fare_amount + espeed + Tip_amount + Tolls_amount,data=df)
summary(m7)
```

```
##
## Call:
## lm(formula = Total_amount ~ f.tlenkm + Fare_amount + espeed +
##     Tip_amount + Tolls_amount, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.591 -0.387 -0.037  0.150 55.772
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.212557   0.076090  15.936   <2e-16 ***
## f.tlenkmf.tlenkm-(5,67.9]  0.132659   0.089314   1.485    0.138
## f.tlenkmf.tlenkm-[0,1]    -0.043945   0.101516  -0.433    0.665
## Fare_amount                1.003235   0.004947 202.795   <2e-16 ***
## espeed                    -0.004032   0.003369  -1.197    0.231
## Tip_amount                 1.011523   0.015836  63.875   <2e-16 ***
## Tolls_amount               1.027439   0.040313  25.487   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.923 on 4993 degrees of freedom
## Multiple R-squared:  0.9652, Adjusted R-squared:  0.9652
## F-statistic: 2.308e+04 on 6 and 4993 DF,  p-value: < 2.2e-16
```

```
m8<-lm(Total_amount ~  tlenkm +f.Fare_amount + espeed + Tip_amount + Tolls_amount,data=df)
summary(m8)
```

```
##
## Call:
## lm(formula = Total_amount ~ tlenkm + f.Fare_amount + espeed +
##     Tip_amount + Tolls_amount, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -72.442  -0.787  -0.118   0.474  53.447
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       12.618792   0.227247   55.53   <2e-16 ***
## tlenkm                             1.386418   0.019843   69.87   <2e-16 ***
## f.Fare_amountf.Fare_amount-(6.5,9] -5.367044   0.204256  -26.28   <2e-16 ***
## f.Fare_amountf.Fare_amount-(9,14.5] -4.022798   0.178615  -22.52   <2e-16 ***
## f.Fare_amountf.Fare_amount-[0,6.5]  -6.515156   0.209623  -31.08   <2e-16 ***
```

```
## espeed                              -0.097683   0.006716  -14.54   <2e-16 ***
## Tip_amount                           1.272224   0.029240   43.51   <2e-16 ***
## Tolls_amount                         1.066612   0.075498   14.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.589 on 4992 degrees of freedom
## Multiple R-squared:  0.8788, Adjusted R-squared:  0.8787
## F-statistic:  5173 on 7 and 4992 DF,  p-value: < 2.2e-16
```

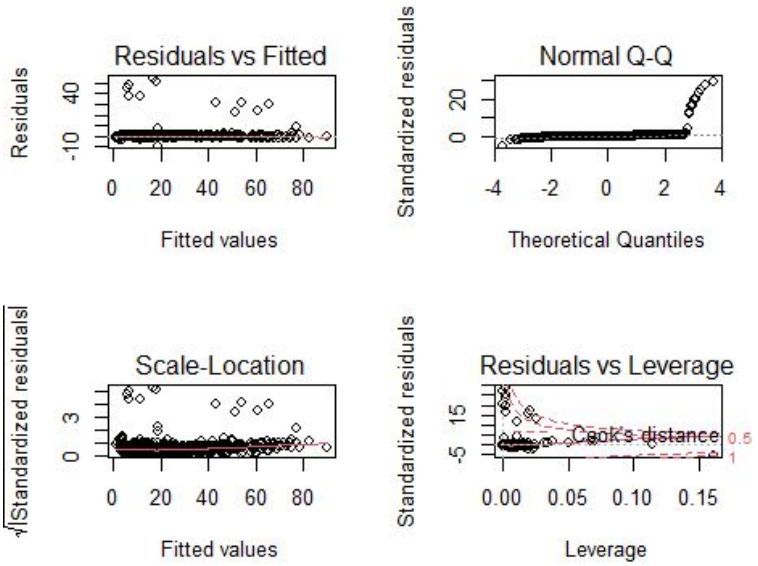
```
m9<-lm(Total_amount ~  f.tlenkm +Fare_amount + f.espeed + Tip_amount + Tolls_amount,data=df)
summary(m9)
```

```
##
## Call:
## lm(formula = Total_amount ~ f.tlenkm + Fare_amount + f.espeed +
##     Tip_amount + Tolls_amount, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.620 -0.380 -0.039  0.137 55.793
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.148572   0.051685  22.223   <2e-16 ***
## f.tlenkmf.tlenkm-(5,67.9]  0.123461   0.090119   1.370    0.171
## f.tlenkmf.tlenkm-[0,1]    -0.040152   0.101498  -0.396    0.692
## Fare_amount                 1.002671   0.004916 203.946   <2e-16 ***
## f.espeedf.espeed-(25,130] -0.048258   0.073715  -0.655    0.513
## f.espeedf.espeed-[0,1]    -0.178280   0.786093  -0.227    0.821
## Tip_amount                  1.011469   0.015840  63.856   <2e-16 ***
## Tolls_amount                1.024093   0.040189  25.482   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.924 on 4992 degrees of freedom
## Multiple R-squared:  0.9652, Adjusted R-squared:  0.9651
## F-statistic: 1.977e+04 on 7 and 4992 DF,  p-value: < 2.2e-16
```

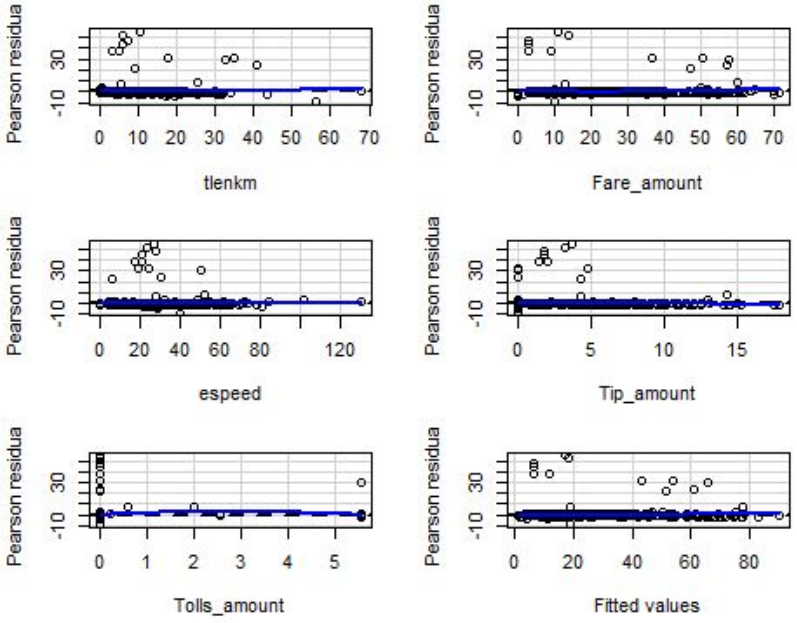
```
BIC(m6,m7,m8,m9)
```

```
##    df      BIC
## m6  7 20682.23
## m7  8 20790.88
## m8  9 27036.18
## m9  9 20800.36
```

Probamos a cambiar la variable tlenkm por su factor obteniendo una explicación de la variabilidad muy similar, aún así, el método BIC nos indica que sigue siendo mejor el modelo anterior siendo su BIC menor. Y la misma argumentación podemos aplicar para el uso del factor de Fare_amount aunque este sí que muestra un descenso significativo de la explicación de la variabilidad del target.




```
residualPlots(m6)
```

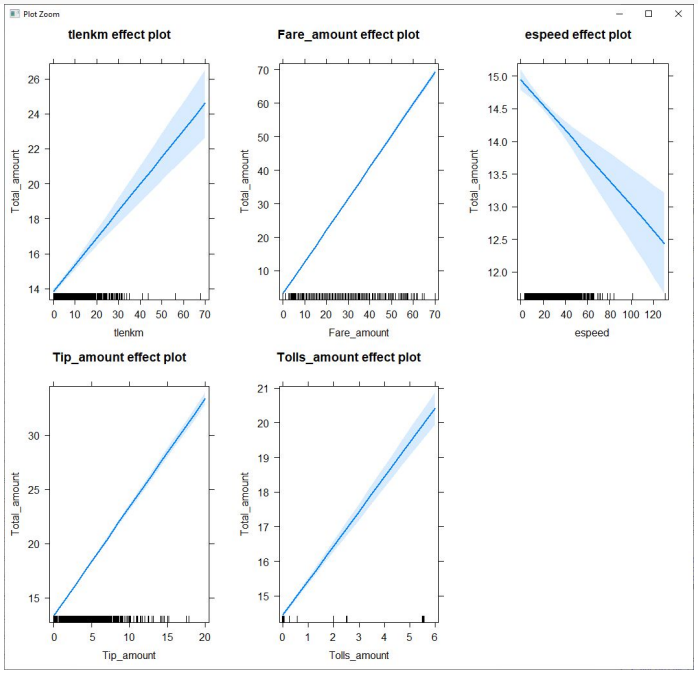



```
##              Test stat Pr(>|Test stat|)
## tlenkm          2.7892        0.0053033 **
## Fare_amount     6.0739         1.34e-09 ***
## espeed         -0.0586        0.9532750
## Tip_amount     -2.8502        0.0043864 **
## Tolls_amount   -3.0041        0.0026766 **
## Tukey test      3.6569        0.0002553 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

Seguimos sin disponer de unos residuos normales, aún así, a excepción de los extremos que distan significativamente, la mayoría de residuos siguen esta distribución.Estos residuos no normales pueden deberse al porcentaje de explicación que le falta a nuestro modelo de la variación de todo el target.

Es cierto que podemos observar un pequeño desajuste respecto a los smoothers del residualPlots pero no son lo suficientemente significativos.

```
library(effects)
plot(allEffects(m6))
```

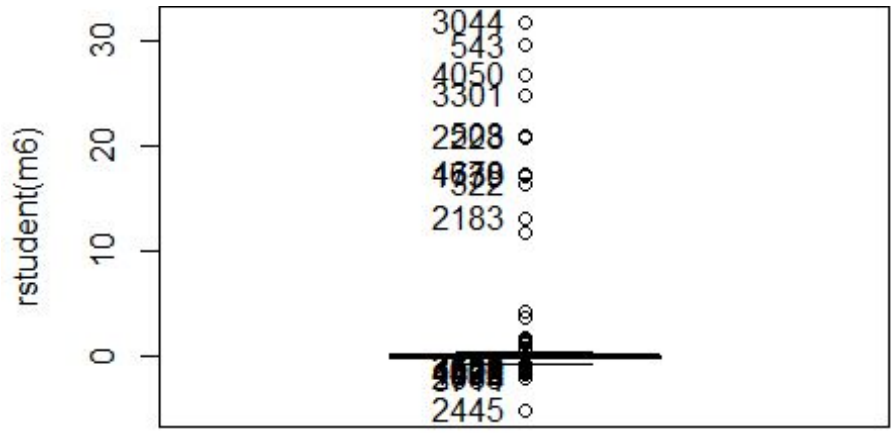


Analizamos como funcionan las variables de nuestro modelo respecto a nuestro target.

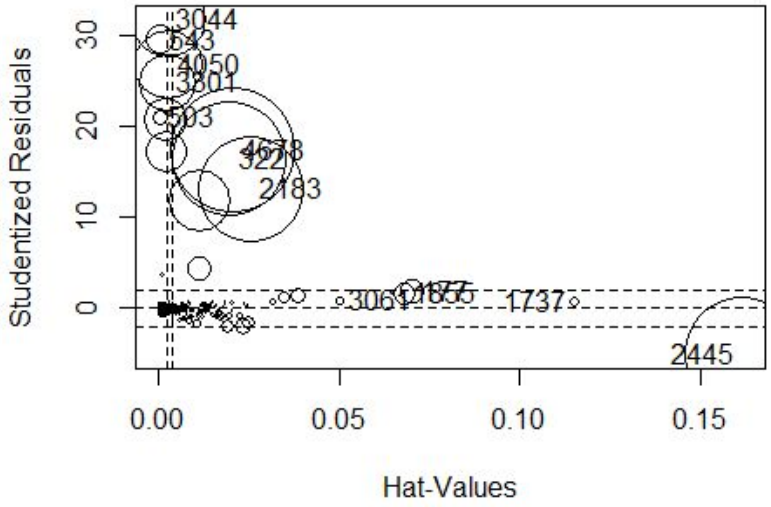
Que a medida que aumenten tanto la distancia en km como el número de peajes así como la tarifa aumente la cuantía total del servicio parece ser coherente por cómo funciona la composición de esta. También podemos llegar a entender que cuanta más propina ofrezca un cliente, más acabará pagando.

La explicación no tan trivial es la de la velocidad efectiva. Se puede justificar esta relación inversa con el hecho de que cuanto más lento vaya el taxi, más tiempo estará produciendo el servicio y por tanto más acabará cobrando. Esto sucede sobretodo en los núcleos urbanos, donde el tiempo predomina frente a la distancia en cuanto al cómputo del precio del transporte.

```
sel1<-Boxplot(rstudent(m6));sel1
```




```
#  [1] 2445 3714 2065  492 1674 4492 1026 3547 2789  634 3044  543 4050 3301  503
## [16] 2228 4678 1739  322 2183

influencePlot(m6,id=list(method="noteworthy", n=5))
```




```
##          StudRes          Hat       CookD
## 177   1.7249933 0.0705226674 0.037613327
## 322  16.3094533 0.0196117938 0.842155534
## 503  20.9400558 0.0002536419 0.017047686
## 543  29.5050653 0.0004487365 0.055477416
## 1737  0.5825831 0.1153369866 0.007375861
## 1855  1.5905816 0.0679028642 0.030708213
## 2183 13.0405729 0.0256565037 0.721886847
## 2445 -5.1061262 0.1614370452 0.832385348
## 3044 31.7182308 0.0026453769 0.370229441
## 3061  0.7904553 0.0502540170 0.005510605
## 3301 24.8888840 0.0022600322 0.208090575
## 4050 26.7275138 0.0029229175 0.305398546
## 4678 17.3590485 0.0205975996 0.996306640
```

En el primer boxplot podemos observar las observaciones que consideraríamos outlier con una distribución rstudent, que debido a nuestro número de observaciones es equivalente a considerar una normal. Con el segundo plot podemos observar los individuos inusuales y ver si son o no influyentes. Podríamos destacar sobretodo el 2445 al tener un residuo

estandarizado negativo y por otro lado, al parecer bastante influyentes, el 3044,543, 2183 y 4678 así como 322.

Using factors as explanatory variables

```
m6<-lm(Total_amount ~  tlenkm +Fare_amount + espeed + Tip_amount + Tolls_amount,data=df)

m6a <-lm(Total_amount ~  poly(tlenkm,2) +Fare_amount + espeed + Tip_amount + Tolls_amount,data=df)
m6b <-lm(Total_amount ~  tlenkm +poly(Fare_amount,2) + espeed + Tip_amount + Tolls_amount,data=df)
m6c <-lm(Total_amount ~  tlenkm+ Fare_amount +poly(espeed,2) + Tip_amount + Tolls_amount,data=df)
m6d <-lm(Total_amount ~  tlenkm+ Fare_amount +espeed + poly(Tip_amount,2) + Tolls_amount,data=df)
m6e <-lm(Total_amount ~  tlenkm +Fare_amount +espeed + Tip_amount + poly(Tolls_amount,2),data=df)

m6f<-lm(Total_amount ~  poly(tlenkm,2) +poly(Fare_amount,2) +poly(espeed,2) + poly(Tip_amount,2) +
poly(Tolls_amount,2),data=df)


BIC(m6,m6a,m6b,m6c,m6d,m6e,m6f)

##     df      BIC
## m6   7 20682.23
## m6a  8 20682.96
## m6b  8 20653.94
## m6c  8 20690.74
## m6d  8 20682.62
## m6e  8 20681.72
## m6f 12 20633.85
```

Si intentamos realizar transformaciones en todas aquellas variables que en el modelo del apartado anterior nos da resultado. Observamos que el único modelo que da mejor resultado es aquel con la variable Fare_amount en la que se aplica un polinomio de segundo grado.

A pesar de todo hemos considerado no oportuno utilizar este modelo debido a la mínima diferencia de BIC así como en cuanto a la explicación de la variabilidad que, si bien aumenta, no consideramos suficiente como para justificar la complejidad añadida al modelo. Al no observar ninguna transformación que mejore sustancialmente el modelo tampoco consideramos hacer una combinación de las mismas.

```
vars_cexp_cat <- c("f.Improvement_surcharge","f.MTA_tax", "Trip_type", "RateCodeID","lpep_pickup_period",
"VendorID", "lpep_pickup_date",  "f.Extra")

m10<-lm(Total_amount~. ,family="binomial",data=df[,c("Total_amount", vars_cexp_cat)])

## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'family' will be disregarded

vif(m10)

##                              GVIF Df GVIF^(1/(2*Df))
## f.Improvement_surcharge 19.079653  1        4.368026
## f.MTA_tax               22.004858  1        4.690934
## Trip_type               14.431207  1        3.798843
## RateCodeID               5.788352  1        2.405899
## lpep_pickup_period       8.756254  3        1.435665
## VendorID                 1.012459  1        1.006210
## lpep_pickup_date         1.339936 30        1.004889
## f.Extra                  9.758503  2        1.767445
```

```r
Anova(m10)
```

```
## Anova Table (Type II tests)
##
## Response: Total_amount
##                          Sum Sq   Df  F value    Pr(>F)
## f.Improvement_surcharge    137    1   1.4086    0.2353
## f.MTA_tax                  144    1   1.4736    0.2248
## Trip_type                 4383    1  44.9457 2.252e-11 ***
## RateCodeID               36962    1 379.0241 < 2.2e-16 ***
## lpep_pickup_period          56    3   0.1918    0.9020
## VendorID                    83    1   0.8461    0.3577
## lpep_pickup_date          2727   30   0.9321    0.5724
## f.Extra                     73    2   0.3756    0.6869
## Residuals               483597 4959
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
step(m10, k=log(nrow(df)))
```

```
## Step:  AIC=22925.57
## Total_amount ~ f.MTA_tax + Trip_type + RateCodeID
##
##              Df Sum of Sq    RSS   AIC
## <none>                    486744 22926
## - f.MTA_tax   1       936 487681 22927
## - Trip_type   1      5467 492211 22973
## - RateCodeID  1     39683 526427 23309

##
## Call:
## lm(formula = Total_amount ~ f.MTA_tax + Trip_type + RateCodeID,
##     data = df[, c("Total_amount", vars_cexp_cat)], family = "binomial")
##
## Coefficients:
##                     (Intercept)           f.MTA_taxf.MTA_tax_YES
##                           21.25                             8.61
## Trip_typef.TripType-Street-Hail          RateCodeIDStandard rate
##                           24.08                           -39.71
```

```r
m11<- lm(Total_amount ~  tlenkm +Fare_amount + espeed + Tip_amount + Tolls_amount + Trip_type +RateCodeID
,data=df)
summary(m11)
```

```
##
## Call:
## lm(formula = Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount +
##     Tolls_amount + Trip_type + RateCodeID, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.830 -0.414 -0.059  0.146 55.043
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.703558   0.204604   3.439 0.000589 ***
## tlenkm                         0.149000   0.015358   9.702  < 2e-16 ***
## Fare_amount                    0.947452   0.007693 123.163  < 2e-16 ***
## espeed                        -0.017603   0.003656  -4.815 1.52e-06 ***
## Tip_amount                     0.998823   0.015710  63.579  < 2e-16 ***
## Tolls_amount                   0.990151   0.040145  24.664  < 2e-16 ***
## Trip_typef.TripType-Street-Hail  1.043899   0.415641   2.512 0.012052 *
## RateCodeIDStandard rate       -0.188495   0.397269  -0.474 0.635180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.9 on 4992 degrees of freedom
```

```
## Multiple R-squared:  0.9661, Adjusted R-squared:  0.966
## F-statistic: 2.029e+04 on 7 and 4992 DF,  p-value: < 2.2e-16
```

```
vif(m11)
```

```
##       tlenkm  Fare_amount       espeed   Tip_amount Tolls_amount    Trip_type
##     7.010194     6.263560     1.610679     1.325573     1.101979     5.743132
##   RateCodeID
##     6.074742
```

```
step(m11, k=log(nrow(df)))
## Step:  AIC=6468.86
## Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount +
##     Trip_type
##
##                Df Sum of Sq   RSS     AIC
## <none>                      18016  6468.9
## - espeed        1        83 18099  6483.3
## - Trip_type     1        87 18103  6484.3
## - tlenkm        1       344 18360  6554.9
## - Tolls_amount  1      2228 20245  7043.4
## - Tip_amount    1     14589 32605  9426.2
## - Fare_amount   1     58537 76553 13693.9

##
## Call:
## lm(formula = Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount +
##     Tolls_amount + Trip_type, data = df)
##
## Coefficients:
##                      (Intercept)                           tlenkm
##                          0.68526                          0.14777
##                      Fare_amount                           espeed
##                          0.94837                         -0.01742
##                       Tip_amount                     Tolls_amount
##                          0.99870                          0.99214
## Trip_typef.TripType-Street-Hail
##                          0.86538
```

También intentamos añadir algunas variables categóricas en el modelo. Como resultado nos dio que las variables categóricas Trip_Type y RateCodeID funcionan bien en el modelo de predicción. Si unimos nuestro mejor modelo del apartado anterior junto con dichas variables nos da que la explicación de la variabilidad del target es del 96.61%.

A pesar de todo hemos considerado no oportuno utilizar este modelo debido a la mínima diferencia de BIC así como en cuanto a la explicación de la variabilidad que, si bien aumenta, no consideramos lo suficiente como para justificar la complejidad añadida al modelo. Al no observar ninguna transformación que mejore sustancialmente el modelo tampoco consideramos hacer una combinación de las mismas.

Clear effects

```
anova(m6a,m6f)
```

```
## Analysis of Variance Table
##
## Model 1: Total_amount ~ poly(tlenkm, 2) + Fare_amount + espeed + Tip_amount +
##     Tolls_amount
## Model 2: Total_amount ~ poly(tlenkm, 2) + poly(Fare_amount, 2) + poly(espeed,
##     2) + poly(Tip_amount, 2) + poly(Tolls_amount, 2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   4993 18075
## 2   4989 17777  4     298.2 20.922 < 2.2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m6b,m6f)
```

```
## Analysis of Variance Table
##
## Model 1: Total_amount ~ tlenkm + poly(Fare_amount, 2) + espeed + Tip_amount +
##     Tolls_amount
## Model 2: Total_amount ~ poly(tlenkm, 2) + poly(Fare_amount, 2) + poly(espeed,
##     2) + poly(Tip_amount, 2) + poly(Tolls_amount, 2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   4993 17970
## 2   4989 17777  4    193.58 13.582 5.151e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m6c,m6f)
```

```
## Analysis of Variance Table
##
## Model 1: Total_amount ~ tlenkm + Fare_amount + poly(espeed, 2) + Tip_amount +
##     Tolls_amount
## Model 2: Total_amount ~ poly(tlenkm, 2) + poly(Fare_amount, 2) + poly(espeed,
##     2) + poly(Tip_amount, 2) + poly(Tolls_amount, 2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   4993 18103
## 2   4989 17777  4    326.35 22.897 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m6d,m6f)
```

```
## Analysis of Variance Table
##
## Model 1: Total_amount ~ tlenkm + Fare_amount + espeed + poly(Tip_amount,
##     2) + Tolls_amount
## Model 2: Total_amount ~ poly(tlenkm, 2) + poly(Fare_amount, 2) + poly(espeed,
##     2) + poly(Tip_amount, 2) + poly(Tolls_amount, 2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   4993 18074
## 2   4989 17777  4    296.95 20.835 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m6e,m6f)
```

```
## Analysis of Variance Table
##
## Model 1: Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + poly(Tolls_amount,
##     2)
## Model 2: Total_amount ~ poly(tlenkm, 2) + poly(Fare_amount, 2) + poly(espeed,
##     2) + poly(Tip_amount, 2) + poly(Tolls_amount, 2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   4993 18070
## 2   4989 17777  4     293.7 20.606 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(m6f)
```

```
## Anova Table (Type II tests)
##
## Response: Total_amount
##                       Sum Sq   Df  F value    Pr(>F)
## poly(tlenkm, 2)          400    2   56.073 < 2.2e-16 ***
## poly(Fare_amount, 2)   45256    2 6350.436 < 2.2e-16 ***
## poly(espeed, 2)          138    2   19.407 4.019e-09 ***
```

```
## poly(Tip_amount, 2)    14771    2 2072.675 < 2.2e-16 ***
## poly(Tolls_amount, 2)   2280    2  319.925 < 2.2e-16 ***
## Residuals              17777 4989
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Aplicando el método anova para comprobar los efectos limpios producidos por el nuevo modelo, obtenemos que el uso de polinomios de grado dos para todas las variables que forman el modelo són útiles para su construcción. Esto podemos comprobarlo con el método Anova.

Aún así, vemos como las transformaciones que más mejoras aportan, aún siendo estas ínfimas, són las de aplicar el polinomio a las variables Fare_amount y Tolls_amount.

Dirty effects

```
m0<-lm(Total_amount ~ 1,data=df)
anova(m0,m6b)

## Analysis of Variance Table
##
## Model 1: Total_amount ~ 1
## Model 2: Total_amount ~ tlenkm + poly(Fare_amount, 2) + espeed + Tip_amount +
##     Tolls_amount
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1   4999 530682
## 2   4993  17970  6    512712 23743 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m0,m6c)

## Analysis of Variance Table
##
## Model 1: Total_amount ~ 1
## Model 2: Total_amount ~ tlenkm + Fare_amount + poly(espeed, 2) + Tip_amount +
##     Tolls_amount
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1   4999 530682
## 2   4993  18103  6    512579 23562 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m0,m6d)

## Analysis of Variance Table
##
## Model 1: Total_amount ~ 1
## Model 2: Total_amount ~ tlenkm + Fare_amount + espeed + poly(Tip_amount,
##     2) + Tolls_amount
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1   4999 530682
## 2   4993  18074  6    512609 23602 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(m0,m6e)

## Analysis of Variance Table
##
## Model 1: Total_amount ~ 1
## Model 2: Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + poly(Tolls_amount,
##     2)
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
```

```
## 1   4999 530682
## 2   4993  18070  6    512612 23606 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m0,m6f)
```

```
## Analysis of Variance Table
##
## Model 1: Total_amount ~ 1
## Model 2: Total_amount ~ poly(tlenkm, 2) + poly(Fare_amount, 2) + poly(espeed,
##     2) + poly(Tip_amount, 2) + poly(Tolls_amount, 2)
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1   4999 530682
## 2   4989  17777 10    512906 14395 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como podemos ver debido al uso de los efectos sucios, rechazamos la hipótesis nula para todos los modelos que incluían distintos usos del polinomio, por tanto, podemos confirmar que no se trata de modelos equivalentes y necesitamos la aplicación de estos modelos.

Como bién llevamos diciendo durante todo el tratamiento de modelos, este aumento en la complejidad del modelo respecto a las ventajas que nos ofrece no nos parece justificable así que no llegaríamos a adoptar como modelo del target el m6f sinó el m6, el cual no contiene ninguno de los polinomios.

Interactions

```
m12<- lm(Total_amount ~  (tlenkm +Fare_amount + espeed + Tip_amount + Tolls_amount) + (Trip_type
+RateCodeID)^2 ,data=df) # Interaccions dobles en factors
m12<-step(m12, k=log(nrow(df)))
```

```
## Step:  AIC=6468.86
## Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount +
##     Trip_type
##
##                Df Sum of Sq   RSS     AIC
## <none>                      18016  6468.9
## - espeed        1        83 18099  6483.3
## - Trip_type     1        87 18103  6484.3
## - tlenkm        1       344 18360  6554.9
## - Tolls_amount  1      2228 20245  7043.4
## - Tip_amount    1     14589 32605  9426.2
## - Fare_amount   1     58537 76553 13693.9
```

```
m13<- lm(Total_amount ~  (tlenkm +Fare_amount + espeed + Tip_amount + Tolls_amount) * (Trip_type
+RateCodeID)  ,data=df) # Interaccions dobles en factor-numèrica
m13<-step(m13, k=log(nrow(df)))
## Step:  AIC=6260.07
## Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount + Tolls_amount +
##     Trip_type + RateCodeID + tlenkm:RateCodeID + Fare_amount:RateCodeID +
##     espeed:RateCodeID + Tolls_amount:Trip_type
##
##                           Df Sum of Sq   RSS    AIC
## <none>                                  17133 6260.1
## - espeed:RateCodeID        1      50.5 17184 6266.3
## - Tolls_amount:Trip_type   1     285.2 17418 6334.1
## - Fare_amount:RateCodeID   1     450.7 17584 6381.4
## - tlenkm:RateCodeID        1     474.8 17608 6388.2
## - Tip_amount               1   15180.1 32313 9423.9
```

```
BIC(m6,m11,m12,m13)
```

```
##     df      BIC
## m6   7 20682.23
## m11  9 20675.05
## m12  8 20666.76
## m13 13 20457.97
```

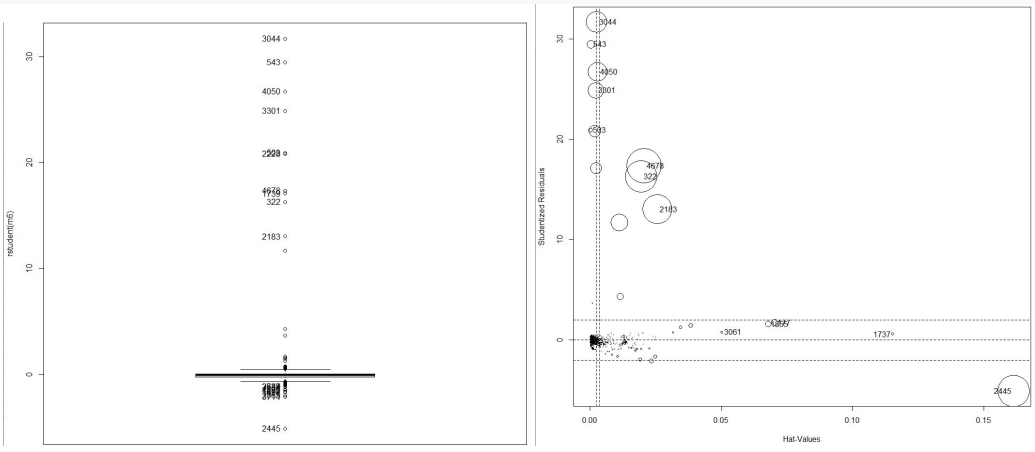
```
summary(m12)
```

```
##
## Call:
## lm(formula = Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount +
##     Tolls_amount + Trip_type, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.759 -0.413 -0.061  0.147 55.050
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.685261   0.200921   3.411 0.000653 ***
## tlenkm                         0.147766   0.015135   9.763  < 2e-16 ***
## Fare_amount                    0.948368   0.007446 127.368  < 2e-16 ***
## espeed                        -0.017415   0.003634  -4.792  1.7e-06 ***
## Tip_amount                     0.998699   0.015707  63.585  < 2e-16 ***
## Tolls_amount                   0.992137   0.039923  24.851  < 2e-16 ***
## Trip_typef.TripType-Street-Hail  0.865381   0.176620   4.900  9.9e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.9 on 4993 degrees of freedom
## Multiple R-squared:  0.9661, Adjusted R-squared:  0.966
## F-statistic: 2.368e+04 on 6 and 4993 DF,  p-value: < 2.2e-16
```

```
summary(m13)
```

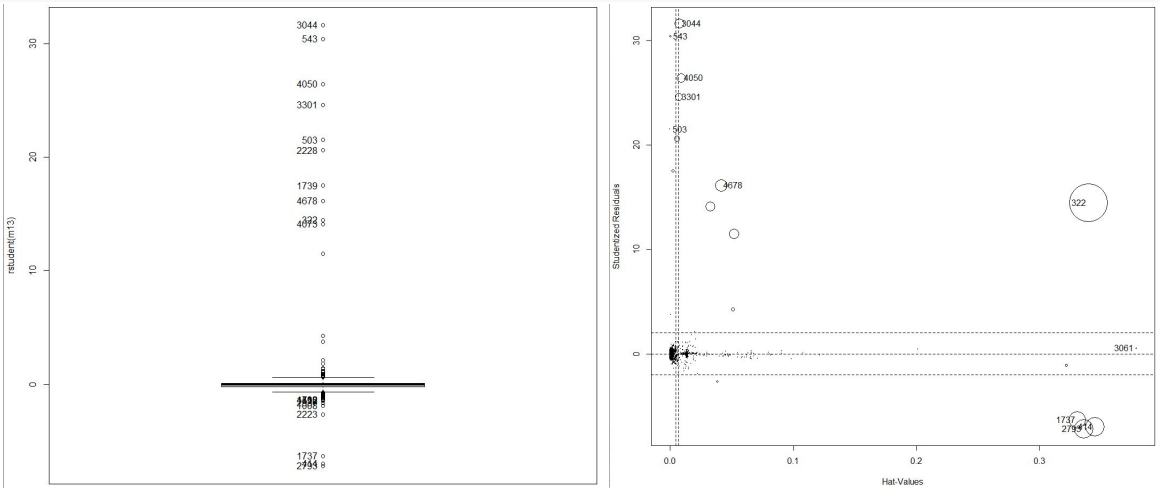
```
##
## Call:
## lm(formula = Total_amount ~ tlenkm + Fare_amount + espeed + Tip_amount +
##     Tolls_amount + Trip_type + RateCodeID + tlenkm:RateCodeID +
##     Fare_amount:RateCodeID + espeed:RateCodeID + Tolls_amount:Trip_type,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.841  -0.435  -0.053   0.184  53.321
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                -0.034628   0.312090  -0.111
## tlenkm                                      0.034592   0.021215   1.631
## Fare_amount                                 0.997337   0.011025  90.460
## espeed                                     -0.005694   0.010186  -0.559
## Tip_amount                                  1.027393   0.015454  66.479
## Tolls_amount                                2.871734   0.209986  13.676
## Trip_typef.TripType-Street-Hail             0.952772   0.476165   2.001
## RateCodeIDStandard rate                     1.676628   0.616624   2.719
## tlenkm:RateCodeIDStandard rate              0.409302   0.034811  11.758
## Fare_amount:RateCodeIDStandard rate        -0.199409   0.017408 -11.455
## espeed:RateCodeIDStandard rate             -0.042593   0.011106  -3.835
## Tolls_amount:Trip_typef.TripType-Street-Hail -1.939430   0.212852  -9.112
##                                            Pr(>|t|)
## (Intercept)                                0.911655
## tlenkm                                     0.103052
## Fare_amount                                 < 2e-16 ***
## espeed                                     0.576172
## Tip_amount                                  < 2e-16 ***
## Tolls_amount                                < 2e-16 ***
## Trip_typef.TripType-Street-Hail            0.045454 *
```

```
## RateCodeIDStandard rate                       0.006570 **
## tlenkm:RateCodeIDStandard rate                 < 2e-16 ***
## Fare_amount:RateCodeIDStandard rate            < 2e-16 ***
## espeed:RateCodeIDStandard rate                0.000127 ***
## Tolls_amount:Trip_typef.TripType-Street-Hail   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.853 on 4988 degrees of freedom
## Multiple R-squared:  0.9677, Adjusted R-squared:  0.9676
## F-statistic: 1.359e+04 on 11 and 4988 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
influencePlot(m6)

##         StudRes          Hat       CookD
## 322  16.3094533 0.0196117938 0.842155534
## 543  29.5050653 0.0004487365 0.055477416
## 1737  0.5825831 0.1153369866 0.007375861
## 2445 -5.1061262 0.1614370452 0.832385348
## 3044 31.7182308 0.0026453769 0.370229441
## 4678 17.3590485 0.0205975996 0.996306640

influencePlot(m12)

##         StudRes          Hat       CookD
## 322  16.8428511 0.0261692343 1.030679619
## 543  29.5763724 0.0004508224 0.047968509
## 1737  0.6351762 0.1154336530 0.007522191
## 2445 -4.5083574 0.1759470935 0.617572508
## 3044 31.8231469 0.0026518042 0.319854291
## 4678 17.3889190 0.0206030687 0.856973131

influencePlot(m13)

##         StudRes          Hat      CookD
## 322  14.4448459 0.3400346495 8.60068176
## 414  -7.0136487 0.3451641260 2.14005133
## 543  30.4108580 0.0004609966 0.02999020
## 2793 -7.2154961 0.3360629530 2.17380963
## 3044 31.6474445 0.0077899326 0.54579570
## 3061  0.5465404 0.3785873287 0.01516737

ll1<-Boxplot(rstudent(m6));ll1
```




```
##  [1] 2445 3714 2065  492 1674 4492 1026 3547 2789  634 3044  543 4050 3301  503
## [16] 2228 4678 1739  322 2183

sel2<-which(hatvalues(m6)>5*length(m6$coefficients)/nrow(df));length(sel2)
```

```
## [1] 172

ll2<-which(row.names(df) %in% names(hatvalues(m6)[sel2]));

sel3<-which(cooks.distance(m6)> 0.5 );sel3;length(sel3)

##  322 2183 2445 4678
##  322 2183 2445 4678

## [1] 4

ll3<-which(row.names(df) %in% names(cooks.distance(m6)[sel3]));ll3

## [1]  322 2183 2445 4678

ll1<-Boxplot(rstudent(m13));

##  [1] 2793  414 1737 2223 1008 2937 4415  592 1738 1712 3044  543 4050 3301  503
## [16] 2228 1739 4678  322 4073

sel2<-which(hatvalues(m13)>5*length(m13$coefficients)/nrow(df));length(sel2)

## [1] 215

ll2<-which(row.names(df) %in% names(hatvalues(m13)[sel2]));

sel3<-which(cooks.distance(m13)> 0.5 );sel3;length(sel3)

##  322  414 1737 2183 2793 3044 4073 4678
##  322  414 1737 2183 2793 3044 4073 4678

## [1] 8

ll3<-which(row.names(df) %in% names(cooks.distance(m13)[sel3]));ll3

## [1]  322  414 1737 2183 2793 3044 4073 4678
```



Para analizar las interacciones que pueden hacer mejorar nuestro modelo, consideramos el modelo m6 y m11 al disponer este último de las variables categóricas necesarias.

Después de aplicar una serie de interacciones entre factores así como entre numéricas y factores, vemos como aparecen los dos modelos con un BIC inferior a nuestro modelo sin interacciones m11, donde el m13 presenta una mejora sustancial en cuanto a su AIC.

Aún así la mejora que obtenemos al aplicar estas interacciones no es lo suficientemente significativa como para justificar el aumento de complejidad del modelo, así que como modelo principal seguiremos usando el m6 ya que como dijimos, el m11 lo descartamos por el mismo motivo producido por las variables extra.

Aquí podemos ver una comparación de las observaciones inusuales tanto de nuestro modelo como del que hemos escogido con interacciones(m13) y vemos como el modelo m13 presenta tal vez menos individuos inusuales aún así acaba presentando el doble(8) de individuos significativos que en el modelo m6 destacando : 322 414 1737 2183 2793 3044 4073 4678. Sobre todo vemos cómo el individuo 322 ha pasado a ser mucho más influyente en este nuevo modelo, aún así no hay una gran diferencia entre los demás individuos influyentes como para considerarlo mejor o peor por ello.

Binary Regression Model

En esta sección se construye un modelo lineal para una variable target numérica, AnyTip.

Para la realización del modelo, consideramos que la variable Tip_amount no estuviese como variable explicativa ya que la variable target AnyTip se calculaba a partir del Tip_amount. Decidimos seleccionar aquellos individuos que no pagaban en efectivo, ya que eran los únicos que tenían registrada la propina. Como consecuencia la variable Payment_type se elimina como variable explicativa. Para poder realizar el estudio, un 70% de nuestro dataset formó parte del train dataset y el 30% formó parte del test dataset. Inicialmente consideramos como variables explicativas: Passenger_count, tlenkm, Fare_amount, espeed, Tolls_amount, lpep_pickup_time, traveltime y distHaversine.

Use explanatory numeric variables

```
m<-glm(AnyTip~.,family="binomial",data=train_dataset[,c("AnyTip",vars_cexp)])
summary(m)

## Analysis of Deviance Table (Type II tests)
##
## Response: AnyTip
##                  Df  Chisq Pr(>Chisq)
## Passenger_count   1 0.3979    0.52816
## tlenkm            1 4.5918    0.03213 *
## Fare_amount       1 0.1895    0.66333
## espeed            1 2.5285    0.11181
## Tolls_amount      1 0.0071    0.93297
## lpep_pickup_time  1 0.0553    0.81411
## traveltime        1 2.6192    0.10558
## distHaversine     1 3.8074    0.05103 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

vif(m)

##  Passenger_count           tlenkm      Fare_amount           espeed
##         1.004448        10.557730         7.221147         3.564562
##     Tolls_amount lpep_pickup_time       traveltime    distHaversine
##         1.057881         1.008695         6.207482         3.806424
```

Como podemos ver con el Anova las variables tlenkm, es la única variable que rechaza la hipotesi nula y por lo tanto son las variables que tienen más asociación. Podríamos

considerar que la variable distHaversine también está asociada a la variable target ya que su Chisq es muy próximo al nivel de significancia 0.05. Hemos considerado que no hay problemas de multicolinealidad si el valor de VIF es inferior a 11.

Como vemos que hay pocas variables explicativas asociadas con la variable target, intentaremos ampliar la lista de variables explicativas.

```
##                          Eta2      P-value
## Tip_amount        0.212639269 1.354923e-92
## Pickup_longitude  0.018844441 8.532558e-09
## Dropoff_longitude 0.010593539 1.648365e-05
## Total_amount      0.008553588 1.093070e-04
## Pickup_latitude   0.005779308 1.482968e-03
## Dropoff_latitude  0.005663466 1.655806e-03
```

Si realizamos un categorical description, vemos que las variables más correlacionadas con la variable target son el Pickup_longitude , el Dropoff_longitude y el total amount. Como el Total amount es una variable target, hemos decidido no incluirla como variable explicativa, a pesar de todo, podría incluirse.

```
m2<-glm(AnyTip~tlenkm+Pickup_longitude+ Dropoff_longitude + distHaversine
,family="binomial",data=train_dataset)
summary(m2)
```

```
##
## Call:
## glm(formula = AnyTip ~ tlenkm + Pickup_longitude + Dropoff_longitude +
##     distHaversine, family = "binomial", data = train_dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3142   0.4502   0.5073   0.5477   1.4798
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -616.92335  112.61024  -5.478 4.29e-08 ***
## tlenkm              -0.03843    0.02278  -1.687 0.091644 .
## Pickup_longitude    -7.77988    2.32850  -3.341 0.000834 ***
## Dropoff_longitude   -0.58682    1.98806  -0.295 0.767861
## distHaversine        0.09409    0.03898   2.414 0.015783 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1408.6  on 1744  degrees of freedom
## Residual deviance: 1372.2  on 1740  degrees of freedom
## AIC: 1382.2
##
## Number of Fisher Scoring iterations: 4
```

Si cogemos aquellas variables numéricas que no fueron eliminadas por hipótesis del modelo anterior más aquellas variables correlacionadas, vemos que eliminaríamos todas las variables explicativas excepto el Pickup_longitude y distHaversine. A pesar de todo decidimos quedarnos con un modelo cuyas variables explicativas son Pickup_longitude, distHaversine y tlenkm.

```
m3<-glm(AnyTip~tlenkm+Pickup_longitude + distHaversine ,family="binomial",data=train_dataset)
summary(m3)
```

```
##
## Call:
```

```
## glm(formula = AnyTip ~ tlenkm + Pickup_longitude + distHaversine,
##     family = "binomial", data = train_dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3445   0.4500   0.5091   0.5485   1.4777
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -612.21900  111.60934  -5.485 4.13e-08 ***
## tlenkm             -0.03869    0.02279  -1.698   0.0895 .
## Pickup_longitude   -8.30310    1.50954  -5.500 3.79e-08 ***
## distHaversine       0.09451    0.03901   2.423   0.0154 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1408.6  on 1744  degrees of freedom
## Residual deviance: 1372.3  on 1741  degrees of freedom
## AIC: 1380.3
##
## Number of Fisher Scoring iterations: 4
```

Consider factors and interactions

```
m41<-glm(AnyTip~f.tlenkm+Pickup_longitude + distHaversine ,family="binomial",data=train_dataset)
m42<-glm(AnyTip~tlenkm+Pickup_longitude + f.distHaversine ,family="binomial",data=train_dataset)
summary(m41)

BIC(m3,m41,m42)

##     df      BIC
## m3   4 1402.171
## m41  5 1411.438
## m42  5 1414.632
```

Si intentamos categorizar todas aquellas variables del modelo del apartado anterior. Observamos que el modelo empeora por lo tanto, no utilizaremos la categorización de las variables numéricas.

```
res.cat <- catdes(df,num.var=which(names(df)=="AnyTip"))

##                               p.value df
## Payment_type             0.000000e+00  2
## f.Total_amount          1.923205e-104  7
## f.Fare_amount            6.129444e-24  3
## f.tlenkm                 1.087294e-19  2
## f.traveltime             2.107456e-18  4
## f.distHaversine          7.945740e-11  2
## f.Improvement_surcharge  3.570885e-09  1
## f.MTA_tax                4.542332e-09  1
## Trip_type                5.576360e-08  1
## RateCodeID               3.777248e-06  1
## lpep_pickup_period       5.610059e-06  3
## AnyToll                  3.502094e-05  1
## f.espeed                 1.425728e-04  2
## VendorID                 5.925000e-03  1
## lpep_pickup_date         1.450254e-02 30
## f.Extra                  3.273592e-02  2


m5<-glm(AnyTip~. ,family="binomial",data=train_dataset[,c("AnyTip", vars_cexp_cat)])

##                                 GVIF Df GVIF^(1/(2*Df))
## f.Improvement_surcharge 2.027487e+07  1     4502.762320
## f.MTA_tax               2.803930e+07  1     5295.215279
```

```
## Trip_type               1.445557e+07  1     3802.048400
## RateCodeID              6.691136e+06  1     2586.722988
## lpep_pickup_period      9.945849e+00  3        1.466472
## VendorID                1.018046e+00  1        1.008983
## lpep_pickup_date        1.375646e+00 30        1.005330
## f.Extra                 1.095316e+01  2        1.819218
```

Si vemos la multicolinealidad de las variables más correlacionadas con el target, nos encontramos que las variables f.Improvement_surcharge, f.MTA_tax, Trip_Type y RateCodeID tienen un valor GVIF muy alto. Es por eso que de todas esas variables vamos a quedarnos solo con f.MTA_tax.

```
m6<-glm(AnyTip~(f.MTA_tax+lpep_pickup_period+VendorID+lpep_pickup_date+f.Extra),family="binomial",data=tra
in_dataset)
vif(m6)

##                         GVIF Df GVIF^(1/(2*Df))
## f.MTA_tax           1.246385  1        1.116416
## lpep_pickup_period  9.311071  3        1.450441
## VendorID            1.017422  1        1.008674
## lpep_pickup_date    1.393596 30        1.005547
## f.Extra            10.340142  2        1.793212

Anova(m6,test="Wald")

## Analysis of Deviance Table (Type II tests)
##
## Response: AnyTip
##                     Df   Chisq Pr(>Chisq)
## f.MTA_tax            1 34.4323  4.413e-09 ***
## lpep_pickup_period   3  9.2760    0.02584 *
## VendorID             1  0.0736    0.78615
## lpep_pickup_date    30 29.6474    0.48381
## f.Extra              2  5.7404    0.05669 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como podemos observar, las únicas variables categóricas que han pasado el test de la Chisq son f.MTA_tax y lpep_pickup_period. A pesar de todo, la variable f.Extra está cerca del 0.05 por lo tanto la consideraremos para el estudio.

```
m6<-glm(AnyTip~f.MTA_tax+lpep_pickup_period+f.Extra,family="binomial",data=train_dataset)
step(m6, k=log(nrow(df)))

## Step:  AIC=1395.63
## AnyTip ~ f.MTA_tax
##
##             Df Deviance    AIC
## <none>           1378.6 1395.6
## - f.MTA_tax  1   1408.6 1417.1

##
## Call:  glm(formula = AnyTip ~ f.MTA_tax, family = "binomial", data = train_dataset)
##
## Coefficients:
##            (Intercept)  f.MTA_taxf.MTA_tax_YES
##               -0.05407                1.94492
##
## Degrees of Freedom: 1744 Total (i.e. Null);  1743 Residual
## Null Deviance:       1409
## Residual Deviance: 1379  AIC: 1383

summary(m6)
```

```
## AIC: 1380.3
```

Si utilizamos la función step de R nos menciona que deberíamos eliminar la variable f.Extra y lpep_pickup_period. A pesar de todo, vamos a mantener todas las variables explicativas. El resultado que nos da es de AIC 1380.3

```
m7<-glm(AnyTip~ (tlenkm+Pickup_longitude+distHaversine)+(f.MTA_tax+lpep_pickup_period+f.Extra),
family="binomial", data=train_dataset)
m71<-glm(AnyTip~ (poly(tlenkm,2)+Pickup_longitude+distHaversine)+(f.MTA_tax+lpep_pickup_period+f.Extra),
family="binomial", data=train_dataset)

Anova(m7,test="Wald")

## Analysis of Deviance Table (Type II tests)
##
## Response: AnyTip
##                     Df   Chisq Pr(>Chisq)
## tlenkm               1  0.9072    0.34085
## Pickup_longitude     1 20.3001  6.620e-06 ***
## distHaversine        1  2.8814    0.08961 .
## f.MTA_tax            1 21.8609  2.931e-06 ***
## lpep_pickup_period   3  8.7453    0.03288 *
## f.Extra              2  5.3584    0.06862 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova(m71,test="Wald")

## Analysis of Deviance Table (Type II tests)
##
## Response: AnyTip
##                     Df   Chisq Pr(>Chisq)
## poly(tlenkm, 2)      2  7.6596    0.02171 *
## Pickup_longitude     1 20.4303  6.184e-06 ***
## distHaversine        1  0.0260    0.87181
## f.MTA_tax            1 22.4679  2.137e-06 ***
## lpep_pickup_period   3  8.1314    0.04337 *
## f.Extra              2  4.9681    0.08340 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

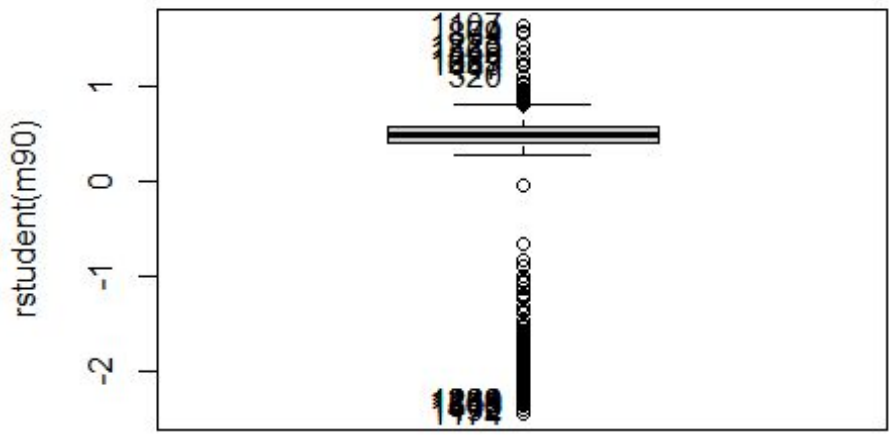
Si intentamos unir las variables numéricas explicativas y las variables categóricas explicativas, nos sale que las variables numéricas tlenkm y distHaversine deben eliminarse. Pero si intentamos hacer el polinomio ortogonal de base 2 respecto a la variable tlenkm, nos sale que solo debemos eliminar la variable distHaversine.

```
m8<-glm(AnyTip~ (poly(tlenkm,2)+Pickup_longitude)+(f.MTA_tax+lpep_pickup_period+f.Extra),
family="binomial", data=train_dataset)
m82<-glm(AnyTip ~ (poly(tlenkm,2)+Pickup_longitude)+(f.MTA_tax+lpep_pickup_period+f.Extra)^2,
family="binomial", data=train_dataset) # Interaccions dobles en factors
m83<-glm(AnyTip ~ (poly(tlenkm,2)+Pickup_longitude)*(f.MTA_tax+lpep_pickup_period+f.Extra),
family="binomial", data=train_dataset)  # Interaccions dobles en factor-numèrica

BIC(m8,m82,m83)

##     df      BIC
## m8  10 1409.216
## m82 18 1453.547
## m83 28 1510.916

m90 <- m8
```

Si intentamos comparar el modelo normal con el modelo con interacción doble en factor y el modelo con interacción doble en factor-numèrica, observamos que los modelos empeoran. Por lo tanto, seguiremos con el modelo normal.

Final Diagnostics

Residus




```
##  [1]  141  545  683  813 1019 1139 1174 1177 1238 1239 1406 1464 1492
```

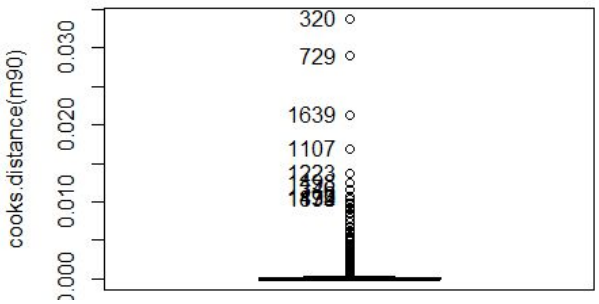
Como podemos ver, en total encontramos 13 individuos que tienen como residuo un valor superior a 2.25.

Observacions potencialment influents

```
##  [1]   47   49   60   75  101  106  139  186  312  320  340  346  363  447  485
## [16]  519  569  577  614  692  699  729  744  757  758  781  785  840  855  869
## [31]  874  961 1064 1079 1107 1132 1194 1210 1223 1367 1436 1442 1503 1569 1589
## [46] 1639 1653 1726
```

En total encontramos que 48 individuos son potencialmente influyentes.

Influent data

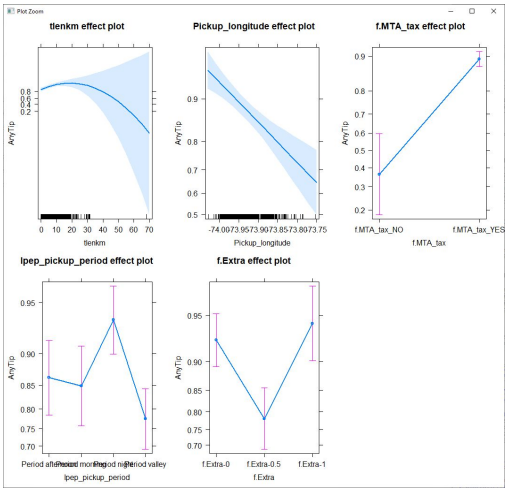



```
##  [1]  141  320  346  363  447  545  614  729  813  874  961 1107 1139 1174 1223
## [16] 1238 1239 1406 1464 1492 1498 1639
```

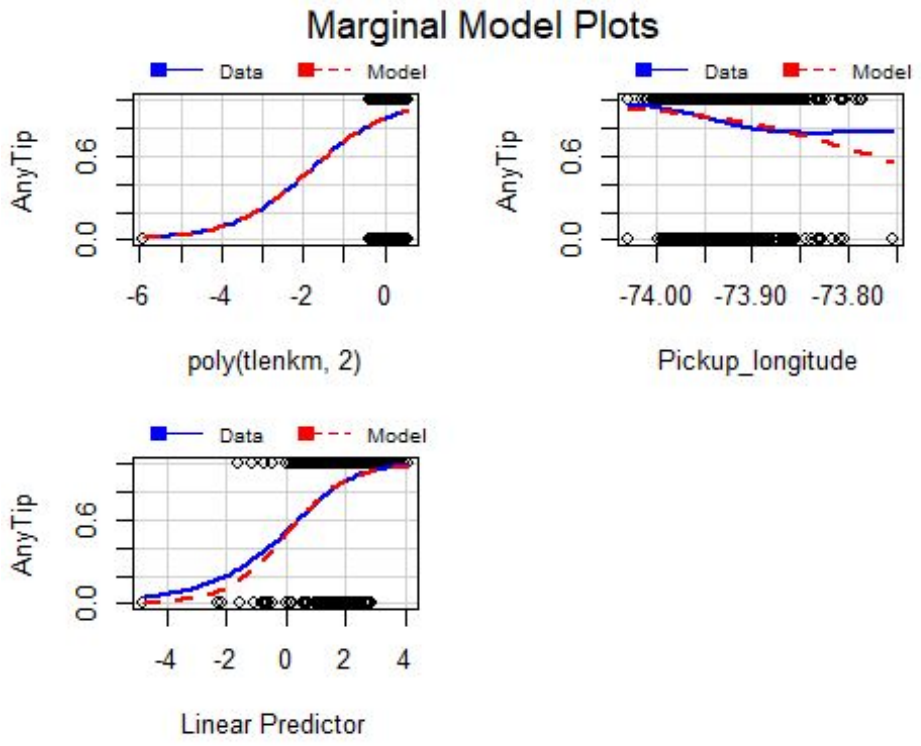
En total encontramos que 22 individuos son influyentes.

```
m10<-glm(AnyTip~ (poly(tlenkm,2)+Pickup_longitude)+(f.MTA_tax+lpep_pickup_period+f.Extra),
family="binomial", data=df1)
m101<-glm(AnyTip ~ (poly(tlenkm,2)+Pickup_longitude)+(f.MTA_tax+lpep_pickup_period+f.Extra)^2,
family="binomial", data=df1) # Interaccions dobles en factors
m102<-glm(AnyTip ~ (poly(tlenkm,2)+Pickup_longitude)*(f.MTA_tax+lpep_pickup_period+f.Extra),
family="binomial", data=df1)  # Interaccions dobles en factor-numèrica

BIC(m10,m101,m102)

##      df      BIC
## m10  10 1328.457
## m101 17 1367.295
## m102 28 1429.212
```

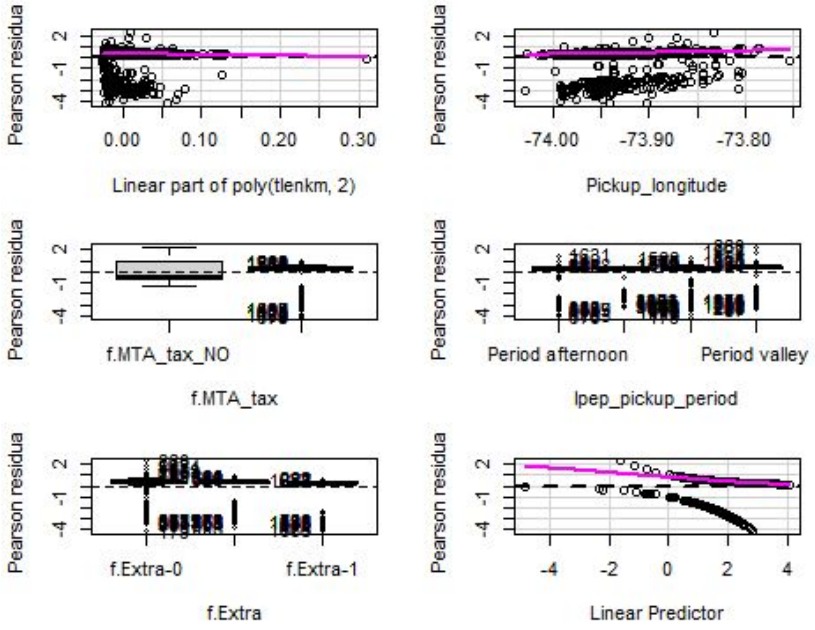
Si volvemos a recalcular el modelo observamos que el modelo cuya fórmula es un sumatorio, sigue siendo el mejor modelo para nuestros datos.



Si intentamos entender el modelo, vemos que si aumenta la distancia recorrida en km o aumenta el valor Pickup_longitude disminuye la probabilidad de dar propina. Si se paga taxa entonces la probabilidad de dar propina es muy alta y si no se paga taxas entonces la probabilidad de no dar propinas es muy alta.



Si vemos el marginal modelo plot, vemos que la línea de los modelos se ajusta más o menos a la línea de los datos, por cada una de las variables.



Si vemos los residuos nos encontramos que en el plot de Linear Predictor - Pearson residuals vemos como la línea de smoother es inclinado por lo tanto tenemos desajuste en el modelo.

Confusion Table

```
##  AnyTip No AnyTip Yes
##  0.1323273  0.8676727

##
## fit.AnyTip              AnyTip No AnyTip Yes
##   Prediction-AnyTip No          13           5
##   Prediction-AnyTip Yes       215        1490

100*sum(diag(tt))/sum(tt)

## [1] 87.23157
```

Como podemos ver nuestro modelo tiene una precisión del 87.23157. A pesar de todo, no es un modelo totalmente correcto ya que podemos podemos ver como en nuestro dataset hay más individuos con AnyTip Yes que con AnyTip No, por lo tanto nuestro dataset está desbalanceado. Además podemos ver que nuestro modelo tiene una tendencia a predecir siempre que el individuo da propinas, probablemente causado por el desbalance del dataset.