Trabajo foro John cabrera - Asignatura 3

2024-03-30

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(visreg)
library(caret)
```

```
## Loading required package: lattice
```

Solucion al reto propuesto en foro ed la Asignatura 3 para el Master en BigData y BI

A continuacion se detallan los pasos de mi solucion al reto

Carga de datos

Datos de entrenamiento

```
tablon_train = read.csv("TABLON_ENTRENAMIENTO.csv",header=TRUE, sep = ";")
head(tablon_train)
```

```
##     TARGET VAR1 VAR2 VAR3 VAR4
## 1 0.127681    1  0.5    1  300
## 2 0.137929    2  0.5    1  300
## 3 0.166577    3  0.5    1  300
## 4 0.276655    4  0.5    1  300
## 5 0.375599    5  0.5    1  300
## 6 0.537873    6  0.5    1  300
```

Datos de test

```
tablon_test = read.csv("TABLON_VALIDACION.csv",header=TRUE, sep = ";")
head(tablon_test)
```

```
##     TARGET VAR1 VAR2 VAR3 VAR4
## 1 0.153415    1  0.8    1  300
## 2 0.164236    2  0.8    1  300
## 3 0.182893    3  0.8    1  300
## 4 0.290766    4  0.8    1  300
## 5 0.402376    5  0.8    1  300
## 6 0.544483    6  0.8    1  300
```

Exploracion de datos

Verificar los tipos de datos para cada variable en test

```
str(tablon_test)
```

```
## 'data.frame':    2250 obs. of  5 variables:
##  $ TARGET: num  0.153 0.164 0.183 0.291 0.402 ...
##  $ VAR1  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ VAR2  : num  0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 0.8 ...
##  $ VAR3  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ VAR4  : int  300 300 300 300 300 300 300 300 300 300 ...
```

Verificar los tipos de datos para cada variable en train

```
str(tablon_train)
```

```
## 'data.frame':    3150 obs. of  5 variables:
##  $ TARGET: num  0.128 0.138 0.167 0.277 0.376 ...
##  $ VAR1  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ VAR2  : num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
##  $ VAR3  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ VAR4  : int  300 300 300 300 300 300 300 300 300 300 ...
```

Verificar falta de datos para cada variable en train

```
colSums(is.na(tablon_train))
```

```
## TARGET   VAR1   VAR2   VAR3   VAR4
##      0      0      0      0      0
```

Verificar falta de datos para cada variable en test

```
colSums(is.na(tablon_test))
```

```
## TARGET   VAR1   VAR2   VAR3   VAR4
##      0      0      0      0      0
```

verificar detalles de los datasets

```
summary(tablon_train)
```

```
##      TARGET            VAR1            VAR2             VAR3            VAR4
##  Min.   :0.1277   Min.   : 1.0   Min.   : 0.500   Min.   :1.000   Min.   :300
##  1st Qu.:2.2131   1st Qu.:13.0   1st Qu.: 0.750   1st Qu.:1.000   1st Qu.:300
##  Median :3.5503   Median :25.5   Median : 2.500   Median :3.000   Median :400
##  Mean   :3.0593   Mean   :25.5   Mean   : 3.893   Mean   :3.333   Mean   :400
##  3rd Qu.:4.1773   3rd Qu.:38.0   3rd Qu.: 7.500   3rd Qu.:6.000   3rd Qu.:500
##  Max.   :4.6191   Max.   :50.0   Max.   :10.000   Max.   :6.000   Max.   :500
```

```
summary(tablon_test)
```

```
##      TARGET            VAR1            VAR2            VAR3            VAR4
##  Min.   :0.1486   Min.   : 1.0   Min.   :0.80   Min.   :1.000   Min.   :300
##  1st Qu.:2.3024   1st Qu.:13.0   1st Qu.:2.00   1st Qu.:1.000   1st Qu.:300
##  Median :3.7150   Median :25.5   Median :4.00   Median :3.000   Median :400
##  Mean   :3.1526   Mean   :25.5   Mean   :4.16   Mean   :3.333   Mean   :400
##  3rd Qu.:4.2959   3rd Qu.:38.0   3rd Qu.:6.00   3rd Qu.:6.000   3rd Qu.:500
##  Max.   :4.6055   Max.   :50.0   Max.   :8.00   Max.   :6.000   Max.   :500
```

Al parecer los datasets estan preprocesados y sin datos faltantes

verificar los valores de VAR3

```
unique(tablon_train$VAR3)
```

```
## [1] 1 3 6
```

verificar los valores de VAR4

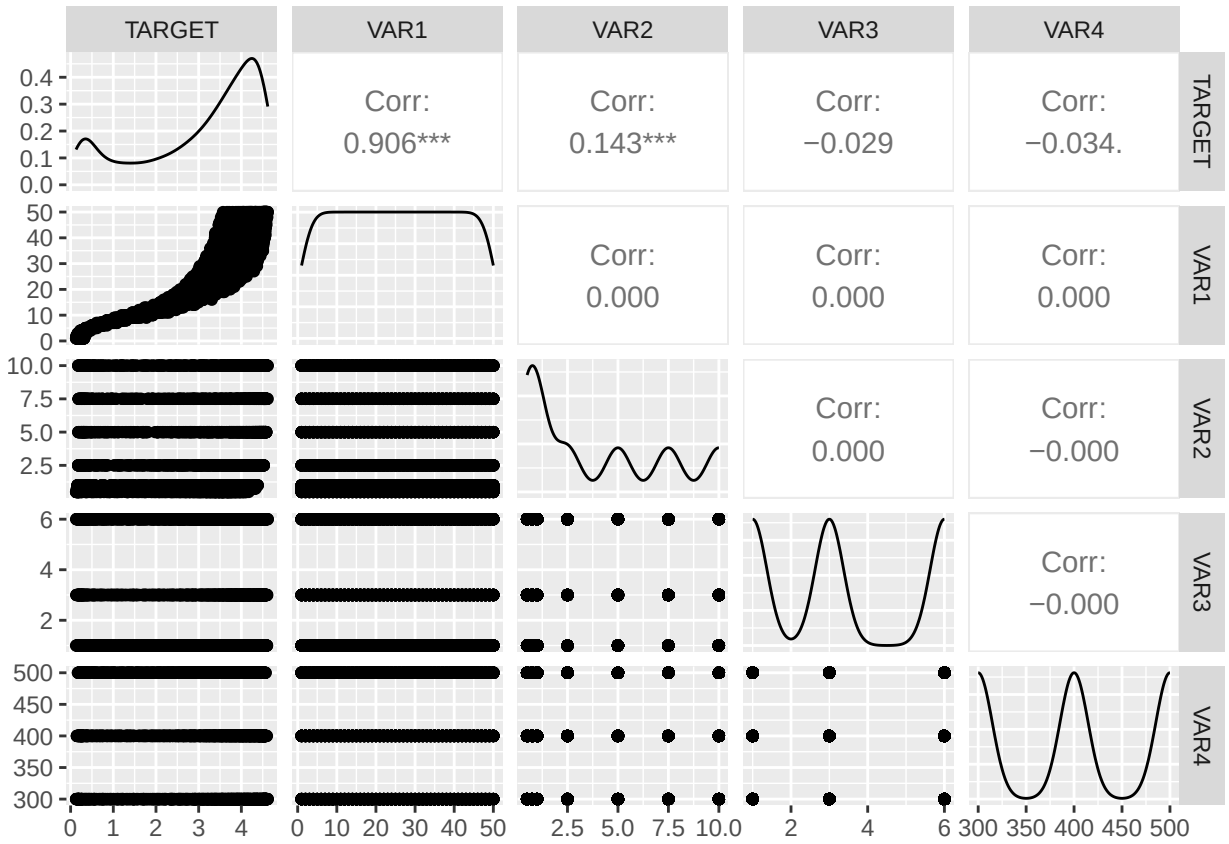
```
unique(tablon_train$VAR4)
```

```
## [1] 300 400 500
```

Al parecer las variables VAR3 y VAR4 podrian ser factores pero se necesitaria mas contexto sobre las variables del dataset y el problema a resolver, por ahora se podría continuar sin realizar transformaciones y dependiendo de los resultados de un analisis predictivo se podrían hacer transformaciones.

grafico de distribuciones de las variables en test

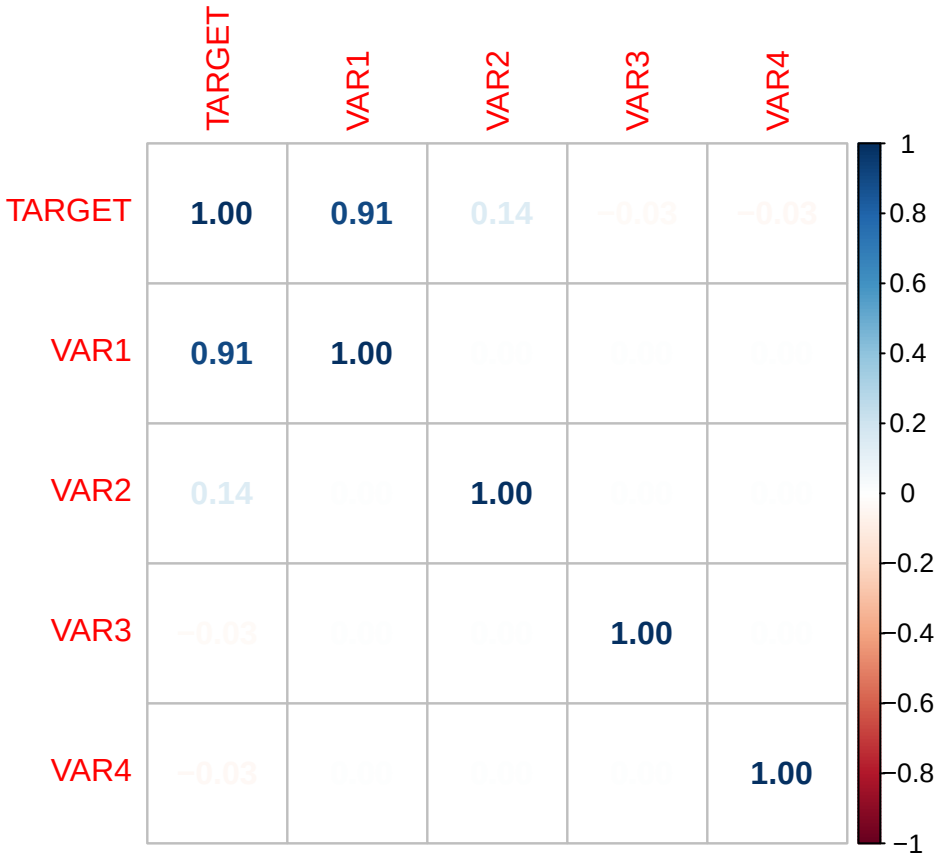
```
ggpairs(tablon_train, columns = 1:5)
```



segun el grafico anterior las distribuciones de las diferentes variables no son normales y la unica que guarda una correlacion significativa con la variable target es VAR1

verificar correlaciones

```
corrplot(cor(tablon_train), method = "number")
```



la variable VAR1 es la que mejor correlacion tiene con la variable target

Reto

A través de un conjunto de datos (TABLON_ENTRENAMIENTO.csv) se debe entrenar un modelo que realice la predicción de la variable TARGET teniendo en cuenta los inputs dados por las variables (VAR1, VAR2, VAR3, VAR4).

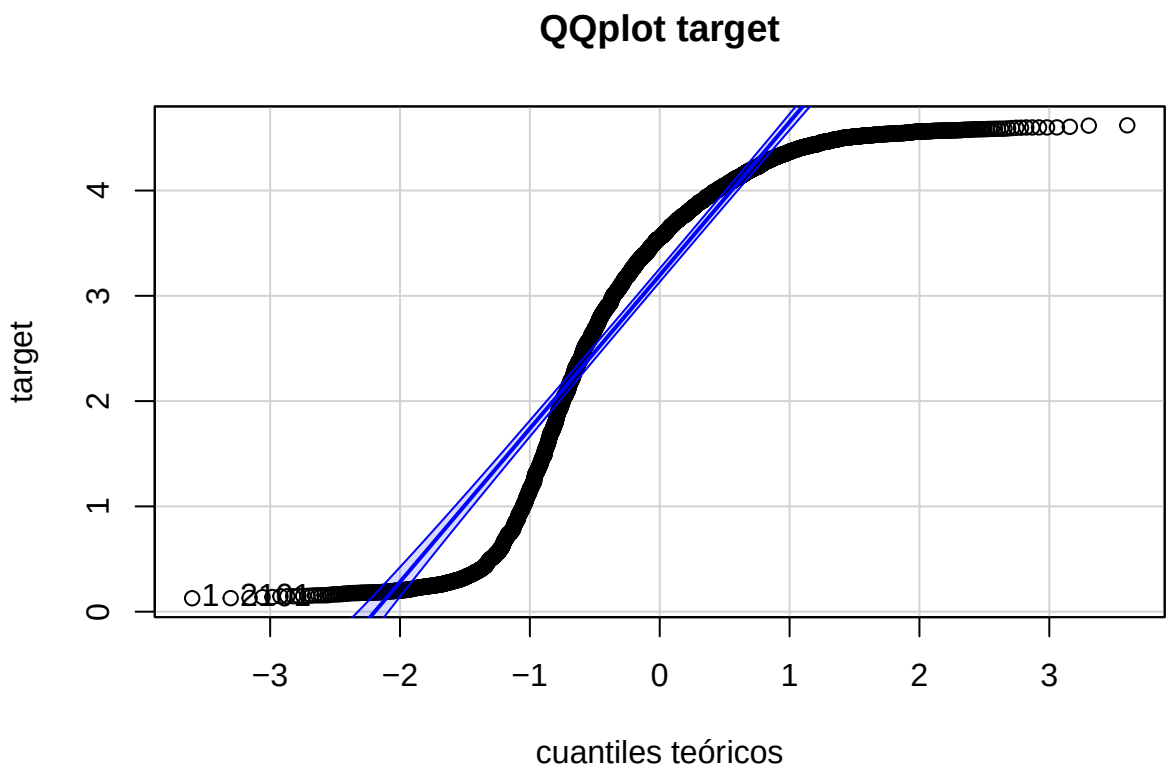
Se debe realizar el test del modelo utilizando los datos del fichero TABLON_VALIDACION.csv. Para ello, elija la métrica que mejor se adapte a la naturaleza de los datos. Pista: la variable TARGET es cuantitativa continua.

Modelo opcion 1

Creacion del Modelo

verificar la forma del la variable TARGET

```
qqPlot(tablon_train$TARGET, ylab="target", xlab="cuantiles teóricos", main="QQplot target")
```




```
## [1]    1 2101
```

Debido a que la variable TARGET es cuantitativa continua y se pide tener en cuenta las variables (VAR1, VAR2, VAR3, VAR4), el modelo de prediccion será una regresion lineal multiple

```
model <- lm(TARGET~VAR1 + VAR2 + VAR3 + VAR4,data = tablon_train)
```

estadisticas del modelo

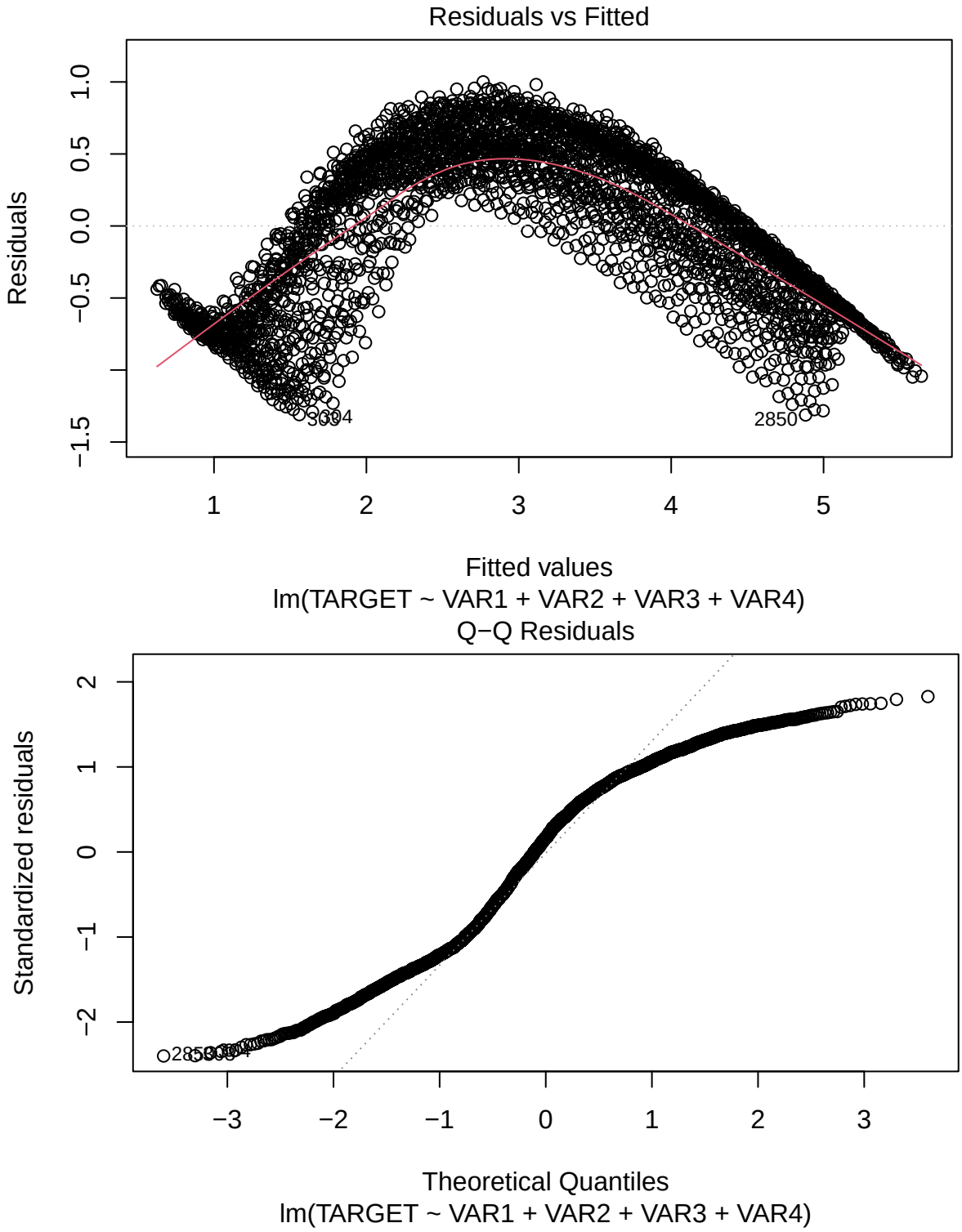
```
summary(model)
```

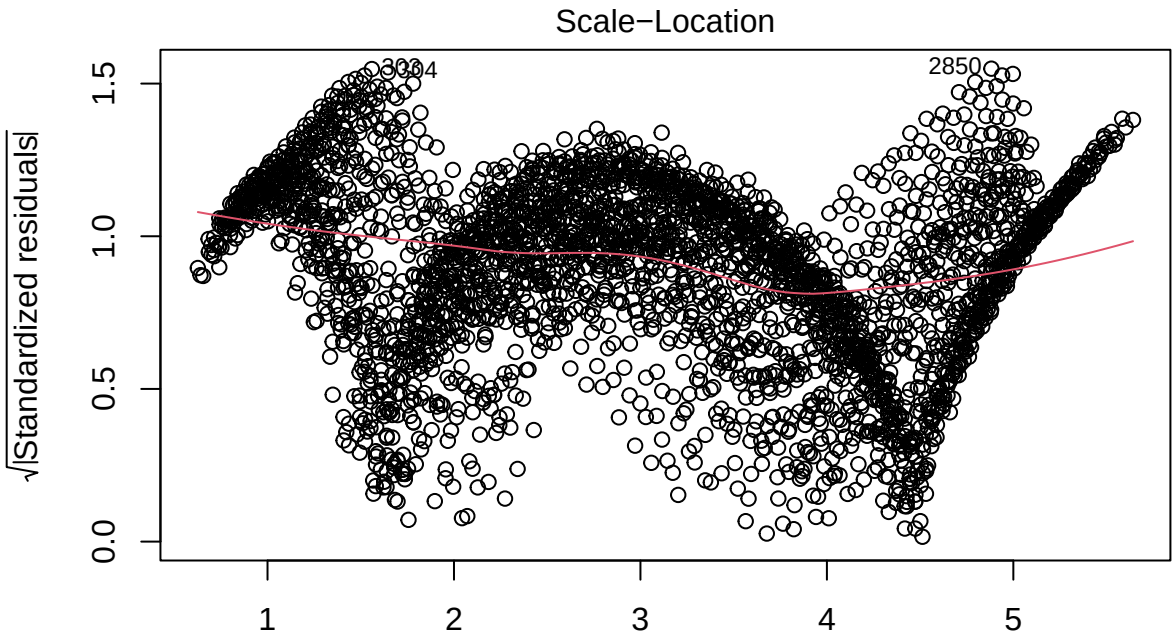
```
##
## Call:
## lm(formula = TARGET ~ VAR1 + VAR2 + VAR3 + VAR4, data = tablon_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.31196 -0.49367  0.09087  0.47884  1.00021
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.9167535  0.0552213  16.601  < 2e-16 ***
## VAR1         0.0868552  0.0006762 128.455  < 2e-16 ***
## VAR2         0.0574772  0.0028297  20.312  < 2e-16 ***
## VAR3        -0.0196577  0.0047486  -4.140 3.57e-05 ***
## VAR4        -0.0005762  0.0001195  -4.822 1.49e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5476 on 3145 degrees of freedom
## Multiple R-squared:  0.8435, Adjusted R-squared:  0.8433
```

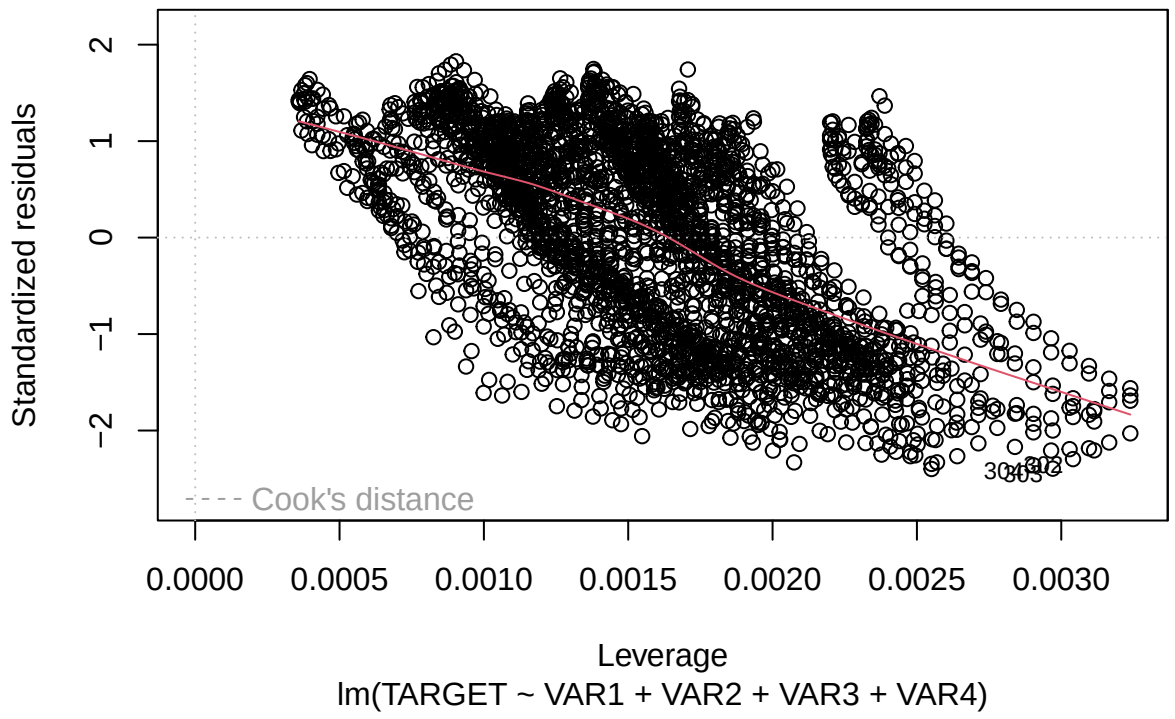
```
## F-statistic:  4238 on 4 and 3145 DF,  p-value: < 2.2e-16
```

El modelo presenta buenas metricas iniciales de acertividad del 84% y un error bajo para ser el primer modelo entrenado

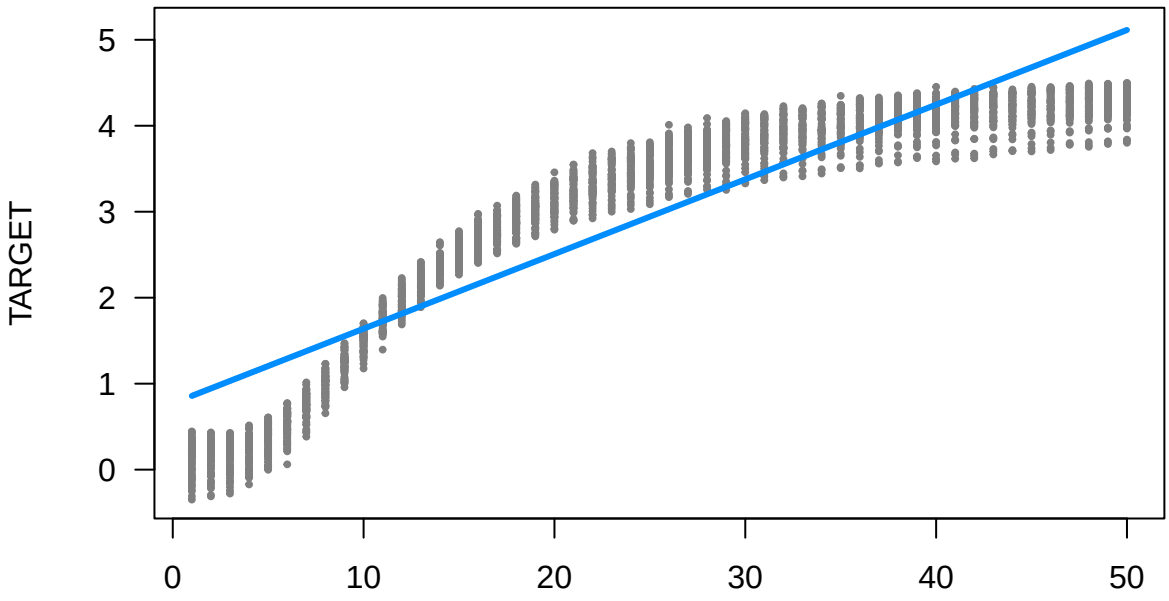
```
plot(model)
```

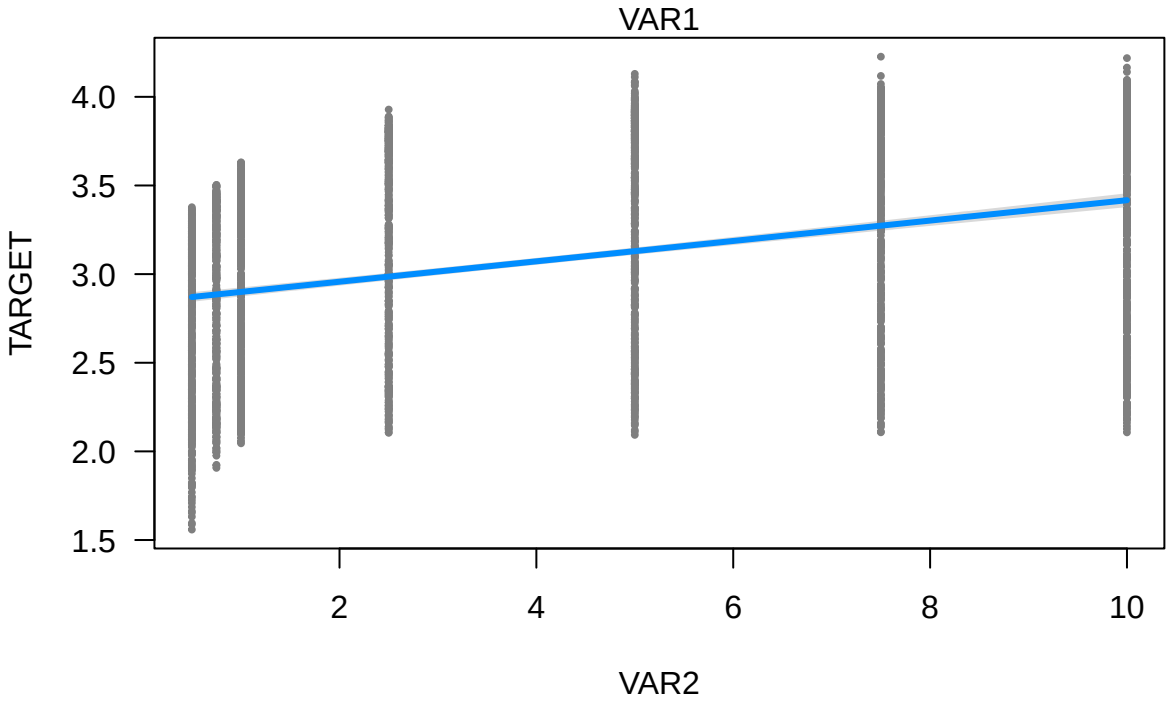


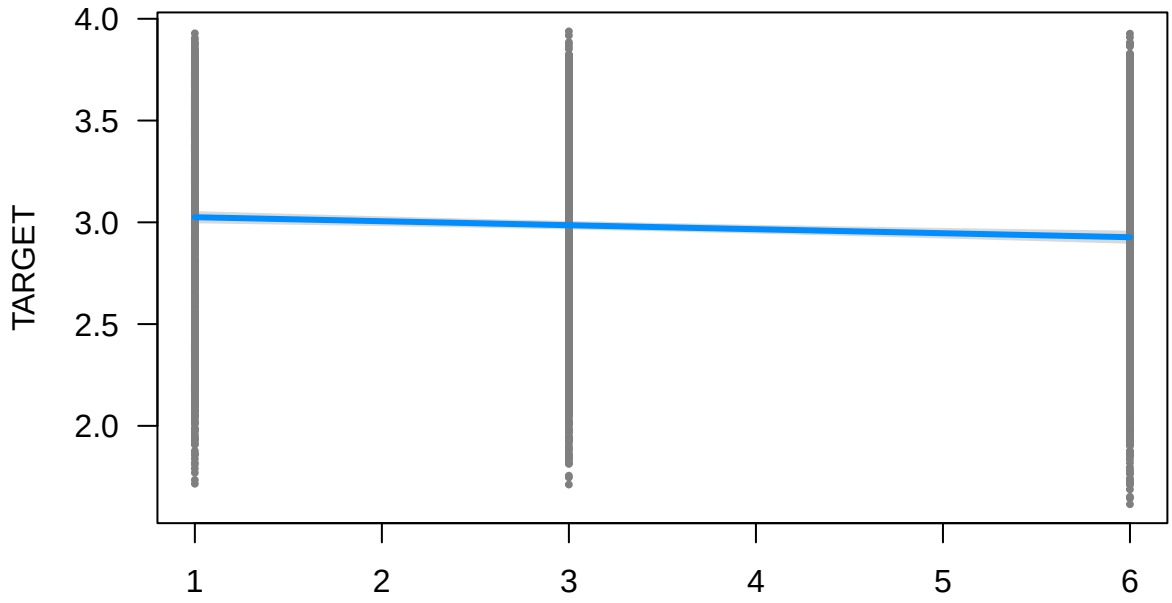


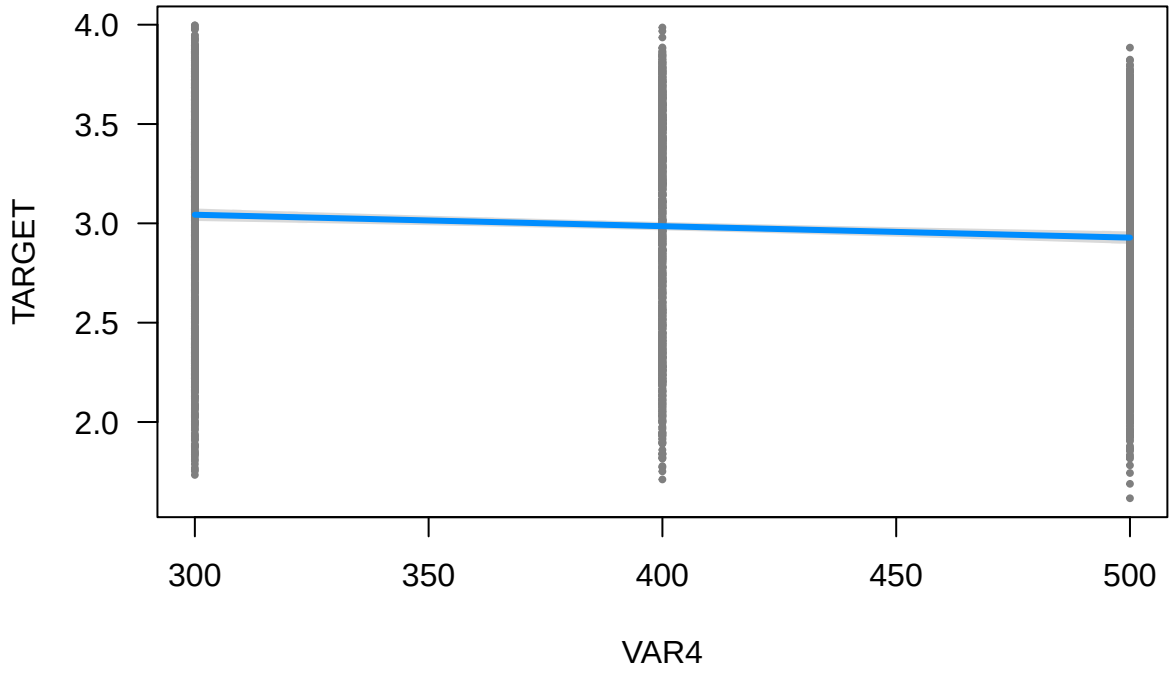



```
visreg(model)
```


TARGET
5
4
3
2
1
0
0
10
20
30
40
50
VAR1


TARGET
4.0
3.5
3.0
2.5
2.0
1.5
2
4
6
8
10
VAR2





###

evaluacion del modelo

```
predictions <- predict(model, newdata = tablon_test)
# MSE
print(sprintf("MSE: %.3f", sqrt(mean((predictions - tablon_test$TARGET)^2))))
```

```
## [1] "MSE: 0.560"
```

```
# R2 score
print(sprintf("R2 score: %.3f", R2(predictions, tablon_test$TARGET)))
```

```
## [1] "R2 score: 0.844"
```

se evidencia una buena metrica del 84% de acertividad del modelo pero no es el mejor. si tomamos en cuenta

que la unica variable que tiene una buena correlacion con TARGET es VAR1.

Modelo opcion 2

observando la forma de la distribucion de los valores en TARGET y tomando en cuenta unicamente la variable VAR1 que tiene mas correlacion con TARGET podriamos generar un modelo de regresion polinomica con mayor precision.

Creacion del Modelo

```
model2 <- lm(TARGET~poly(VAR1,3,raw = TRUE) + VAR2+VAR3+VAR4,data = tablon_train)
```

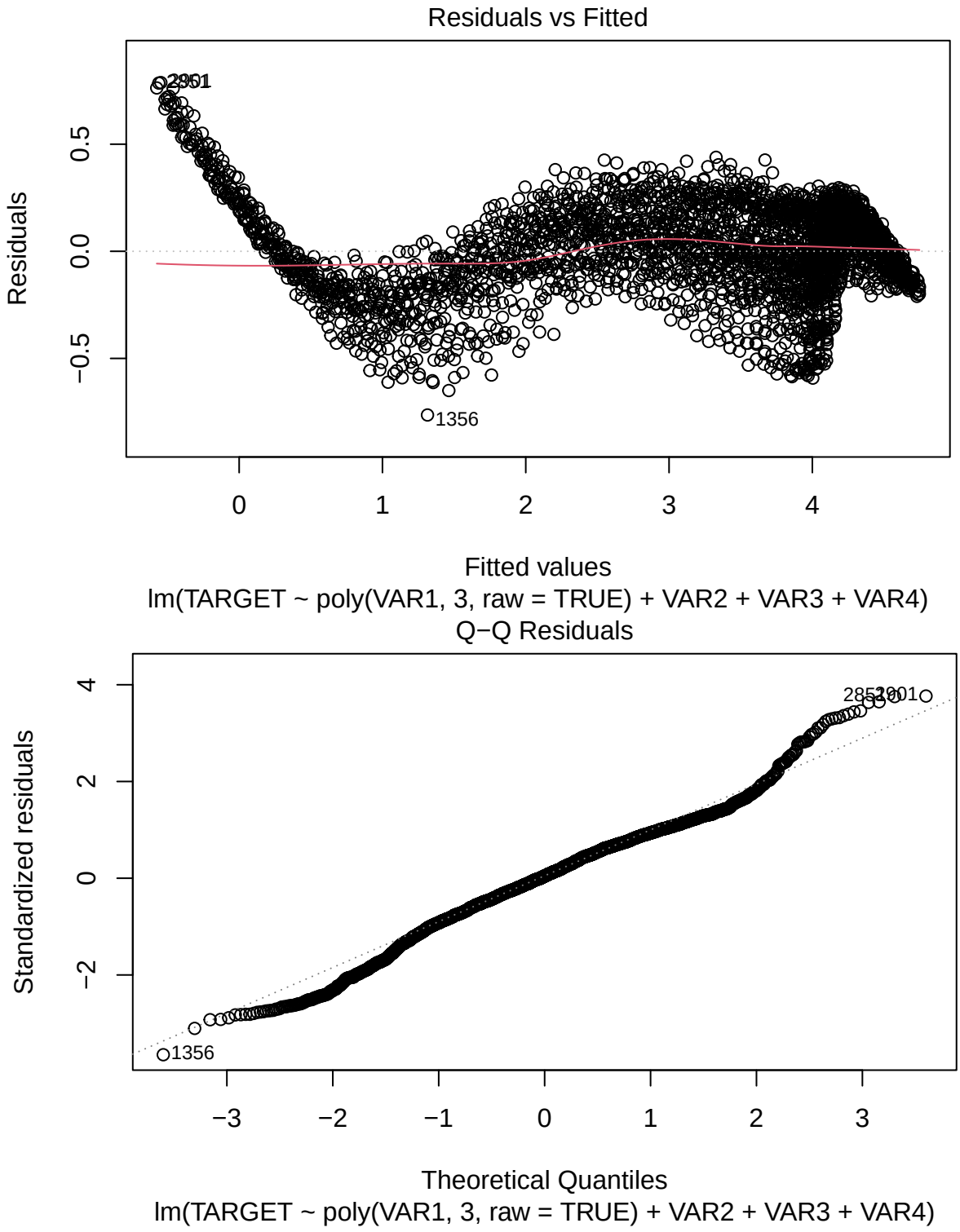
estadisticas del modelo 2

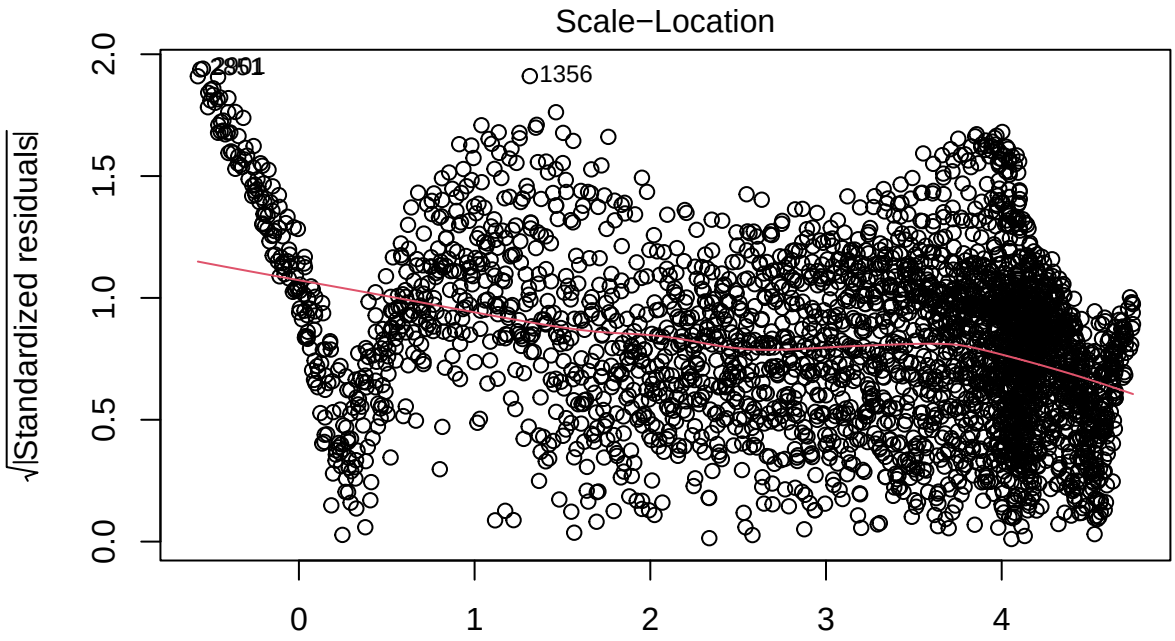
```
summary(model2)
```

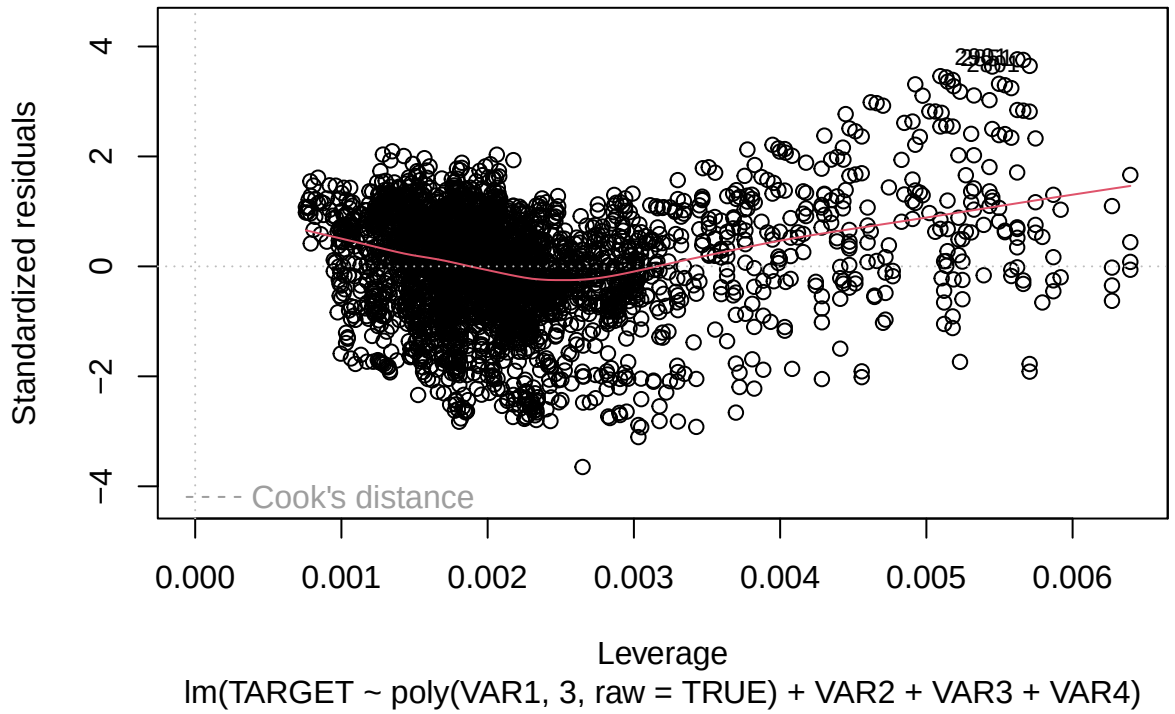
```
##
## Call:
## lm(formula = TARGET ~ poly(VAR1, 3, raw = TRUE) + VAR2 + VAR3 +
##     VAR4, data = tablon_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76405 -0.12323  0.00892  0.14474  0.78747
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -4.589e-01  2.549e-02  -18.00   <2e-16 ***
## poly(VAR1, 3, raw = TRUE)1  2.656e-01  2.713e-03   97.90   <2e-16 ***
## poly(VAR1, 3, raw = TRUE)2 -4.701e-03  1.230e-04  -38.23   <2e-16 ***
## poly(VAR1, 3, raw = TRUE)3  2.622e-05  1.586e-06   16.53   <2e-16 ***
## VAR2                         5.748e-02  1.083e-03   53.05   <2e-16 ***
## VAR3                        -1.966e-02  1.818e-03  -10.81   <2e-16 ***
## VAR4                        -5.762e-04  4.575e-05  -12.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2097 on 3143 degrees of freedom
## Multiple R-squared:  0.9771, Adjusted R-squared:  0.977
## F-statistic: 2.233e+04 on 6 and 3143 DF,  p-value: < 2.2e-16
```

Este modelo muestra una mejora significativa con solo agregar un polinomio base 3 a la variable con mayor correlacion, he decidio la base 3 por la forma en s que toma la variable TARGET

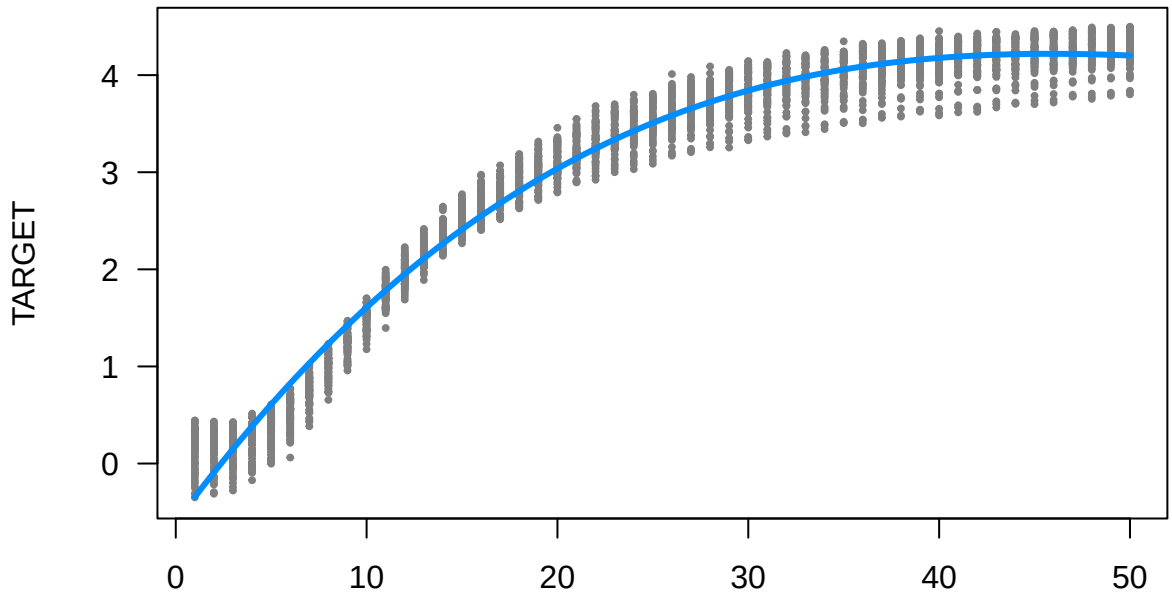
```
plot(model2)
```

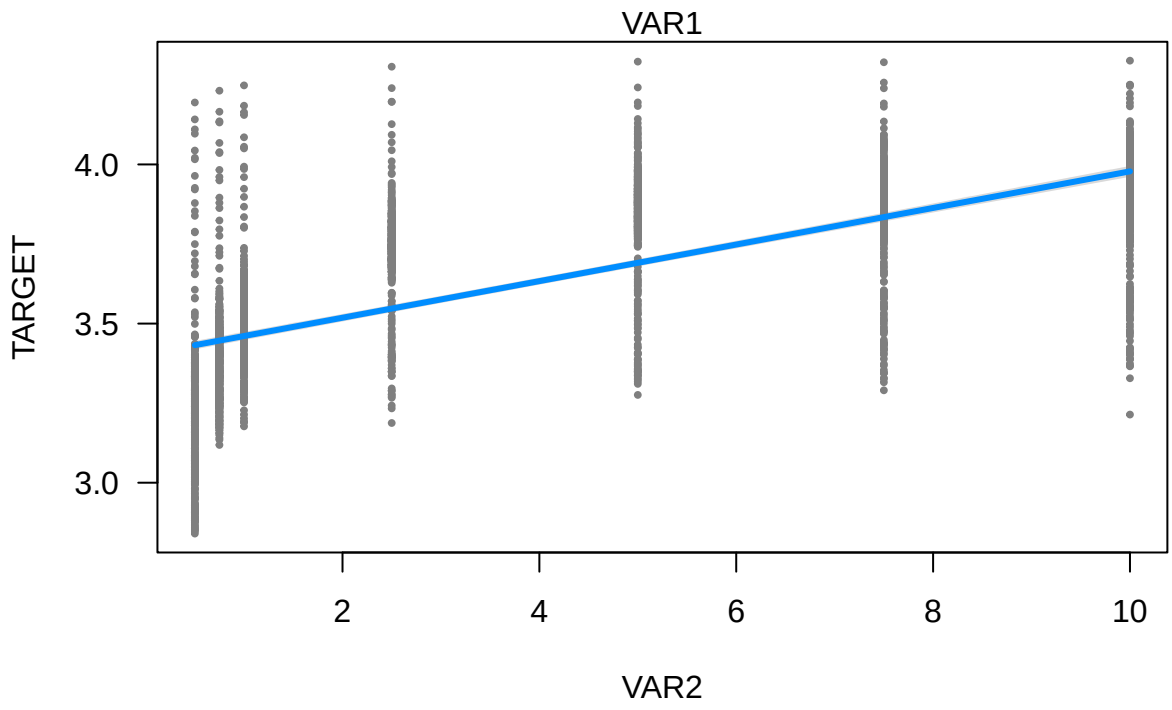

Residuals vs Fitted
Residuals
0.5
0.0
−0.5
1356
0
1
2
3
4
Fitted values
lm(TARGET ~ poly(VAR1, 3, raw = TRUE) + VAR2 + VAR3 + VAR4)
Q−Q Residuals
Standardized residuals
4
2
0
−2
1356
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(TARGET ~ poly(VAR1, 3, raw = TRUE) + VAR2 + VAR3 + VAR4)

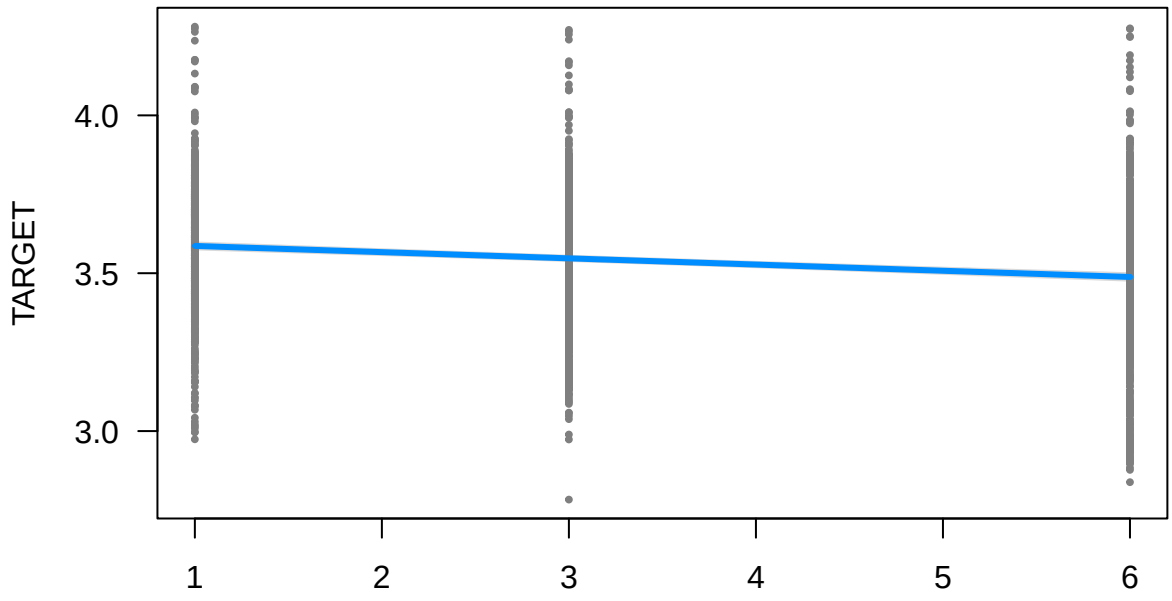


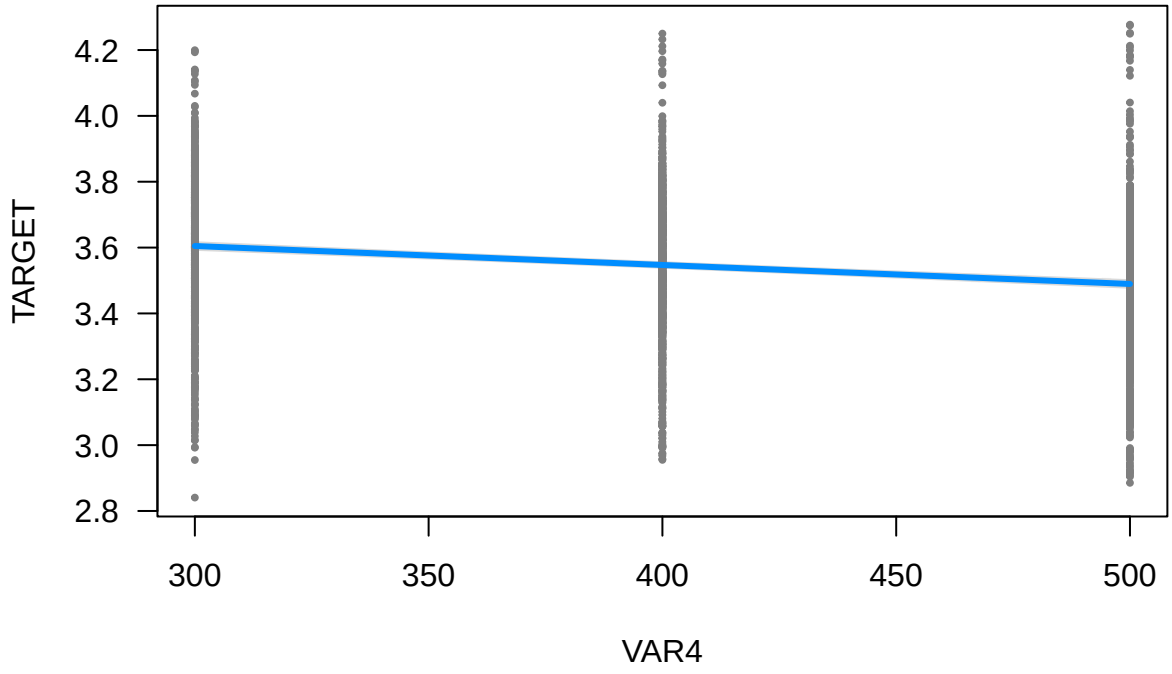



```
visreg(model2)
```


TARGET
0
1
2
3
4
0
10
20
30
40
50
VAR1


TARGET
3.0
3.5
4.0
2
4
6
8
10
VAR2





###

evaluacion del modelo

```
predictions2 <- predict(model2, newdata = tablon_test)
# MSE
print(sprintf("MSE: %.3f", sqrt(mean((predictions2 - tablon_test$TARGET)^2))))
```

```
## [1] "MSE: 0.194"
```

```
# R2 score
print(sprintf("R2 score: %.3f", R2(predictions2, tablon_test$TARGET)))
```

```
## [1] "R2 score: 0.984"
```

Conclusion

Una observacion previa del dataset nos puede dar luz de las variables que debemos tener en cuenta y tal vez como tratar cada una de ellas. El contexto de los datos y del problema es importante para determinar la efectividad de un modelo, en este caso se logra una acertividad de 98% de acierto y un error bajo lo cual es muy bueno para un modelo predictivo tan sencillo, sin embargo al tener mas informacion del dataset o del problema es posible que sea necesario aplicar tecnicas de tranformacion o incluso optar por otro tipo de modelos.