Regressão - Valores Imóveis

Max Pereira

14/06/2020

Dataset

Informações sobre casas em Boston(EUA)

CRIM: Per capita crime rate by town

ZN: Proportion of residential land zoned for lots over 25,000 sq. ft

INDUS: Proportion of non-retail business acres per town

CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)

NOX: Nitric oxide concentration (parts per 10 million)

RM: Average number of rooms per dwelling

AGE: Proportion of owner-occupied units built prior to 1940

DIS: Weighted distances to five Boston employment centers

RAD: Index of accessibility to radial highways

TAX: Full-value property tax rate per $10,000

PTRATIO: Pupil-teacher ratio by town

B: $1000(Bk - 0.63)^2$, where Bk is the proportion of [people of African American descent] by town

LSTAT: Percentage of lower status of the population

MEDV: Median value of owner-occupied homes in $1000s

Objetivo

Prever os valores dos preços das casas usando as variáveis disponíveis

Carregando os pacotes

```
library(readr)
library(caret)
```

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

```
## Loading required package: ggplot2
```

```
library(ggpubr)
```

```
## Loading required package: magrittr
```

Carregando o dataset

```
casas <- read_csv('HousingData.csv')
```

```
## Parsed with column specification:
## cols(
##   CRIM = col_double(),
##   ZN = col_double(),
##   INDUS = col_double(),
##   CHAS = col_double(),
##   NOX = col_double(),
##   RM = col_double(),
##   AGE = col_double(),
##   DIS = col_double(),
##   RAD = col_double(),
##   TAX = col_double(),
##   PTRATIO = col_double(),
##   B = col_double(),
##   LSTAT = col_double(),
##   MEDV = col_double()
## )
```

```
dim(casas)
```

```
## [1] 506  14
```

```
View(casas)
```

Pré-processamento e análise exploratória dos dados

Verificando se há valores nulos (missing)

```r
sapply(casas, function(x) sum(is.na (x)))
```

```
##    CRIM      ZN   INDUS    CHAS     NOX      RM     AGE     DIS     RAD
##      20      20      20      20       0       0      20       0       0
##     TAX PTRATIO       B   LSTAT    MEDV
##       0       0       0      20       0
```

```r
casas <- na.omit(casas) # descartando os valores NA
dim(casas)
```

```
## [1] 394  14
```

Resumo estatístico das variáveis

```r
summary(casas)
```

```
##       CRIM                ZN              INDUS            CHAS
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08196   1st Qu.:  0.00   1st Qu.: 5.13   1st Qu.:0.00000
##  Median : 0.26888   Median :  0.00   Median : 8.56   Median :0.00000
##  Mean   : 3.69014   Mean   : 11.46   Mean   :11.00   Mean   :0.06853
##  3rd Qu.: 3.43597   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       NOX               RM             AGE              DIS
##  Min.   :0.3890   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4530   1st Qu.:5.879   1st Qu.: 45.48   1st Qu.: 2.110
##  Median :0.5380   Median :6.202   Median : 77.70   Median : 3.199
##  Mean   :0.5532   Mean   :6.280   Mean   : 68.93   Mean   : 3.805
##  3rd Qu.:0.6240   3rd Qu.:6.606   3rd Qu.: 94.25   3rd Qu.: 5.117
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       RAD              TAX           PTRATIO            B
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  2.6
##  1st Qu.: 4.000   1st Qu.:280.2   1st Qu.:17.40   1st Qu.:376.7
##  Median : 5.000   Median :330.0   Median :19.10   Median :392.2
##  Mean   : 9.404   Mean   :406.4   Mean   :18.54   Mean   :358.5
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.9
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.9
##      LSTAT             MEDV
##  Min.   : 1.730   Min.   : 5.00
##  1st Qu.: 7.125   1st Qu.:16.80
##  Median :11.300   Median :21.05
##  Mean   :12.769   Mean   :22.36
##  3rd Qu.:17.117   3rd Qu.:25.00
##  Max.   :37.970   Max.   :50.00
```

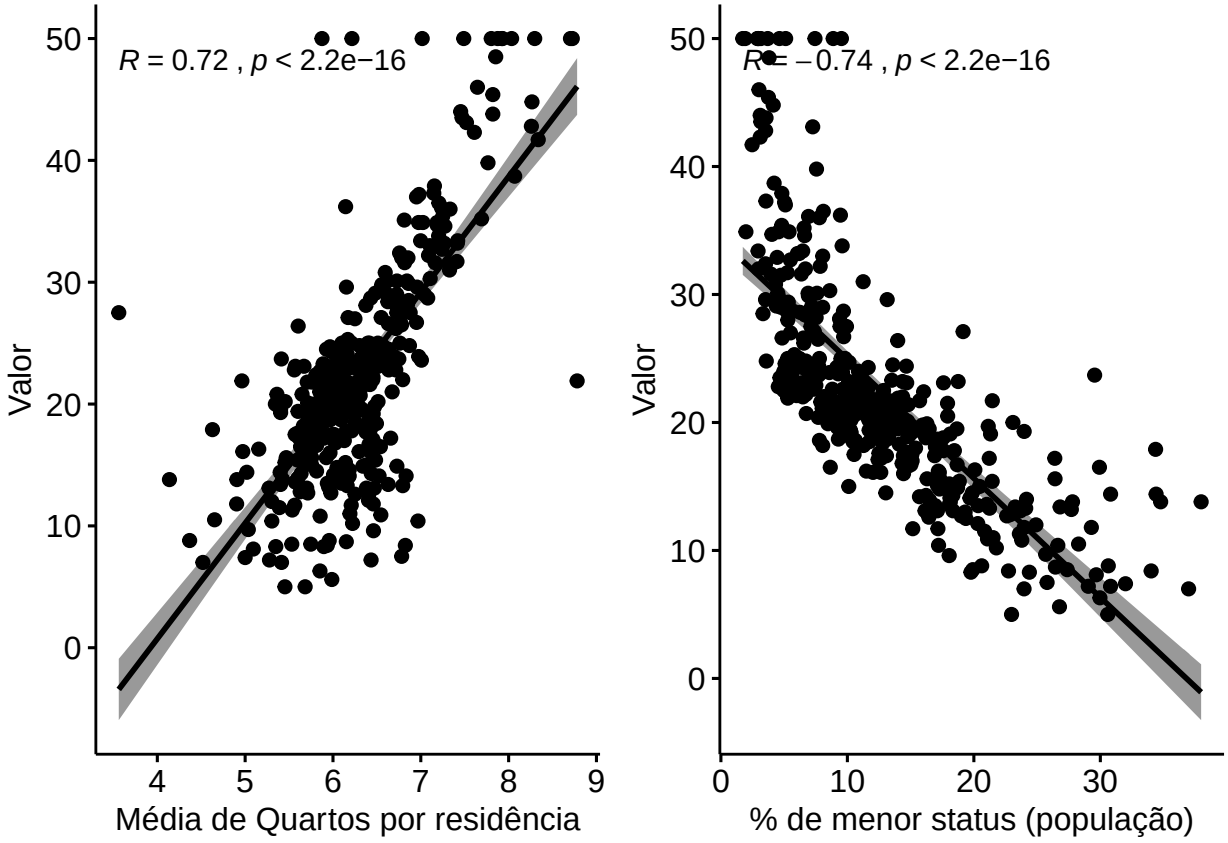
Tabela de correlação

```r
cor(casas)
```

```
##                CRIM          ZN       INDUS        CHAS         NOX
## CRIM     1.00000000 -0.18807507  0.39155182 -0.05196992  0.41615982
## ZN      -0.18807507  1.00000000 -0.52125603 -0.03335682 -0.51566046
## INDUS    0.39155182 -0.52125603  1.00000000  0.04981956  0.76273657
## CHAS    -0.05196992 -0.03335682  0.04981956  1.00000000  0.07666108
## NOX      0.41615982 -0.51566046  0.76273657  0.07666108  1.00000000
## RM      -0.22716991  0.34321034 -0.40306825  0.09530772 -0.31656347
## AGE      0.34131149 -0.56817376  0.64238703  0.07264446  0.73254019
## DIS     -0.36505178  0.64535889 -0.69656900 -0.09503705 -0.76813683
## RAD      0.60866672 -0.29877294  0.59194354  0.01410209  0.62817041
## TAX      0.56084114 -0.30576760  0.73420369 -0.02651313  0.67982405
## PTRATIO  0.26542768 -0.42216416  0.39569127 -0.10499480  0.21021622
## B       -0.38625382  0.16989420 -0.34478755  0.06891304 -0.38425662
## LSTAT    0.46190578 -0.41504110  0.59815590 -0.03711330  0.59365548
## MEDV    -0.39723006  0.40682152 -0.51082916  0.17370115 -0.45905433
##                  RM         AGE         DIS         RAD         TAX
## CRIM    -0.22716991  0.34131149 -0.36505178  0.60866672  0.56084114
## ZN       0.34321034 -0.56817376  0.64535889 -0.29877294 -0.30576760
## INDUS   -0.40306825  0.64238703 -0.69656900  0.59194354  0.73420369
## CHAS     0.09530772  0.07264446 -0.09503705  0.01410209 -0.02651313
## NOX     -0.31656347  0.73254019 -0.76813683  0.62817041  0.67982405
## RM       1.00000000 -0.24867008  0.21871341 -0.23605670 -0.32056056
## AGE     -0.24867008  1.00000000 -0.75354690  0.44358519  0.50447249
## DIS      0.21871341 -0.75354690  1.00000000 -0.47707545 -0.52960262
## RAD     -0.23605670  0.44358519 -0.47707545  1.00000000  0.89999984
## TAX     -0.32056056  0.50447249 -0.52960262  0.89999984  1.00000000
## PTRATIO -0.39068616  0.26496758 -0.22884007  0.44194918  0.44696148
## B        0.12331954 -0.28198984  0.28516841 -0.44413465 -0.43545656
## LSTAT   -0.63622618  0.60113652 -0.50503607  0.51086842  0.57221765
## MEDV     0.72395076 -0.40747050  0.27954693 -0.41663771 -0.50886427
##             PTRATIO           B      LSTAT       MEDV
## CRIM     0.2654277 -0.38625382  0.4619058 -0.3972301
## ZN      -0.4221642  0.16989420 -0.4150411  0.4068215
## INDUS    0.3956913 -0.34478755  0.5981559 -0.5108292
## CHAS    -0.1049948  0.06891304 -0.0371133  0.1737012
## NOX      0.2102162 -0.38425662  0.5936555 -0.4590543
## RM      -0.3906862  0.12331954 -0.6362262  0.7239508
## AGE      0.2649676 -0.28198984  0.6011365 -0.4074705
## DIS     -0.2288401  0.28516841 -0.5050361  0.2795469
## RAD      0.4419492 -0.44413465  0.5108684 -0.4166377
## TAX      0.4469615 -0.43545656  0.5722177 -0.5088643
## PTRATIO  1.0000000 -0.17981583  0.3950058 -0.5438090
## B       -0.1798158  1.00000000 -0.3837834  0.3472561
## LSTAT    0.3950058 -0.38378339  1.0000000 -0.7434496
## MEDV    -0.5438090  0.34725609 -0.7434496  1.0000000
```

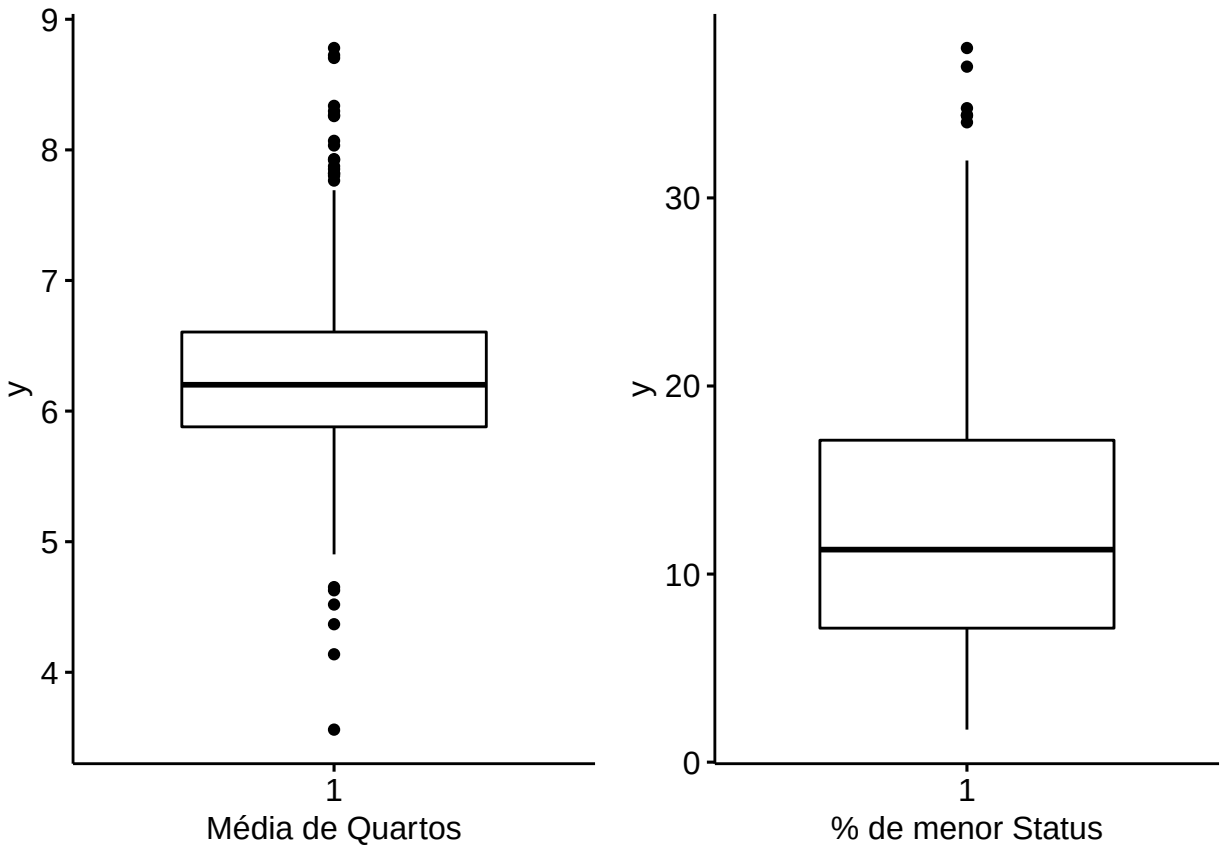
Gráfico de dispersão (scatterplot)

```
g1 <- ggscatter(casas, x = "RM", y = "MEDV",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Média de Quartos por residência", ylab = "Valor")

g2 <- ggscatter(casas, x = "LSTAT", y = "MEDV",
                add = "reg.line", conf.int = TRUE,
                cor.coef = TRUE, cor.method = "pearson",
                xlab = "% de menor status (população)", ylab = "Valor")

ggarrange(g1,g2)
```



Boxpplot das variáveis independentes (RM e LSTAT)

```
b1 <- ggboxplot(casas$RM, xlab="Média de Quartos")
b2 <- ggboxplot(casas$LSTAT, xlab="% de menor Status")
ggarrange(b1,b2)
```




```
boxplot.stats(casas$RM)$out
```

```
##  [1] 8.069 7.820 7.802 7.929 7.765 7.875 7.853 8.034 8.266 8.725 8.337
## [12] 8.259 8.704 8.297 7.820 7.923 8.780 3.561 4.138 4.368 4.652 4.628
## [23] 4.519
```

```
boxplot.stats(casas$LSTAT)$out
```

```
## [1] 34.41 34.77 37.97 34.37 36.98 34.02
```

Resumo estatístico da variável alvo (MEDV)

```
summary(casas$MEDV)
```

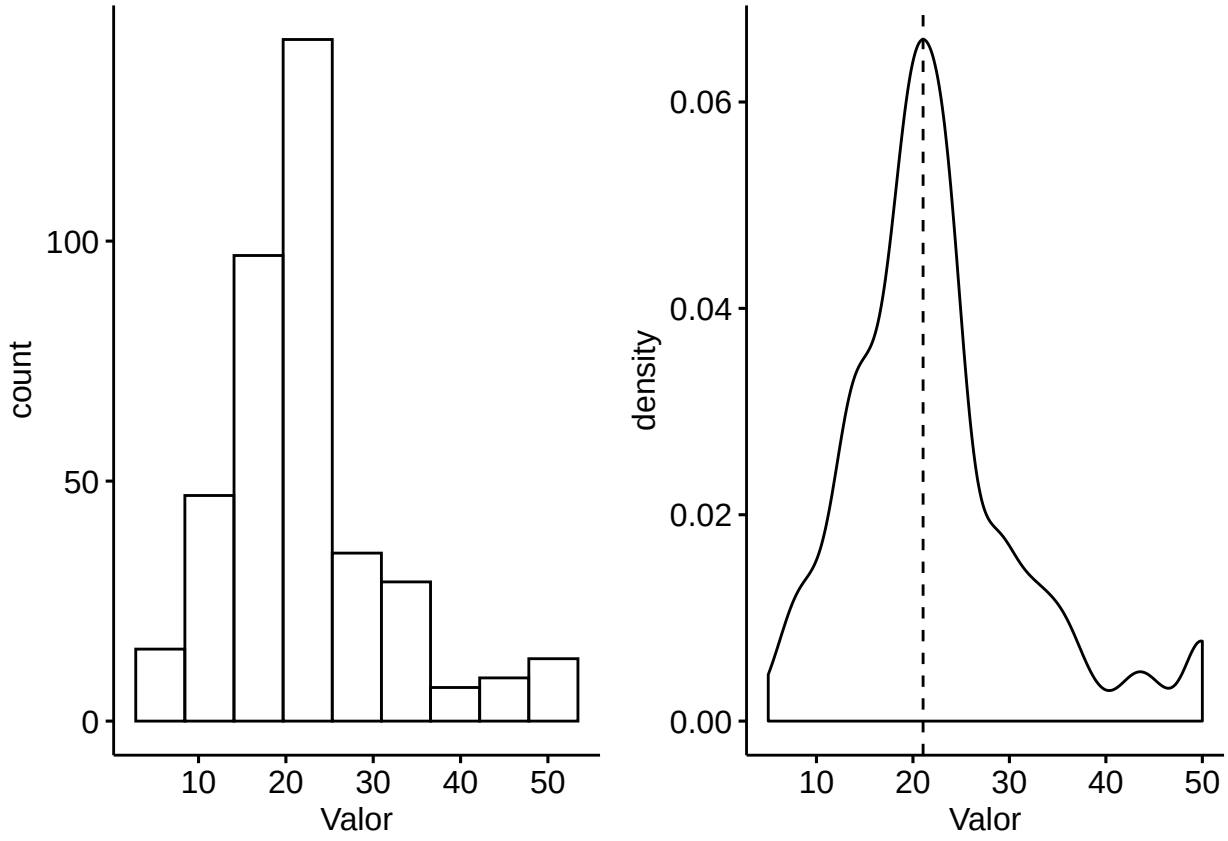
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   16.80   21.05   22.36   25.00   50.00
```

Histograma e densidade da variável alvo (MEDV)

```
ht <- gghistogram(casas, x = "MEDV", bins=9, xlab ="Valor")
ds <- ggdensity(casas, x = "MEDV", add = "median", xlab = "Valor")
```

```
## Warning in (function (mapping = NULL, data = NULL, ..., xintercept, na.rm = FALSE, : Using both `xin
```

```
ggarrange(ht, ds)
```



Criando os datasets de treino e teste

```
set.seed(50)
amostra <- sample(2, nrow(casas), replace=TRUE, prob=c(0.7,0.3))
amostra
```

```
##   [1] 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 2 1 1 1 2 1 1 1 2 2 1 2 2 2 2 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 2
##  [71] 1 2 1 1 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 1 1 1 2 1 1 1
## [106] 2 1 1 2 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
## [141] 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 2 1 1 1
## [176] 1 1 2 1 2 2 1 2 1 1 2 2 1 1 1 2 1 1 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 1
## [211] 1 1 1 1 2 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 2 1 1 1 1 1 1 1 1
## [246] 1 1 2 1 1 2 2 1 1 1 2 1 2 1 2 2 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1
```

```
## [281] 1 2 2 2 1 2 1 1 1 1 1 2 2 1 2 1 1 2 1 2 1 1 1 1 1 1 1 2 2 2 2 1 2 1 1
## [316] 1 1 2 1 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 1 2 1 2 1 1 1 1 1 1 2 1 2 1 1 1
## [351] 2 2 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 2 1 1 1 1
## [386] 2 2 2 1 1 1 2 2 1
```

```
treino <- casas[amostra==1,]
teste <- casas[amostra==2,]

dim(treino)
```

```
## [1] 293  14
```

```
dim(teste)
```

```
## [1] 101  14
```

———-Criando o modelo de regressão linear simples———-

Equação de Regressão y = a + bx (simples)

Treinando o modelo (dados de treino)

```
modelo_s <- lm(MEDV ~ RM, data = treino)
modelo_s
```

```
##
## Call:
## lm(formula = MEDV ~ RM, data = treino)
##
## Coefficients:
## (Intercept)           RM
##     -38.421        9.697
```

Resumo do modelo (métricas)

```
summary(modelo_s)
```

```
##
## Call:
## lm(formula = MEDV ~ RM, data = treino)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8455  -2.3469   0.2048   2.7668  31.4499
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -38.4214     3.5913   -10.7   <2e-16 ***
## RM            9.6973     0.5703    17.0   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.399 on 291 degrees of freedom
## Multiple R-squared:  0.4984, Adjusted R-squared:  0.4966
## F-statistic: 289.1 on 1 and 291 DF,  p-value: < 2.2e-16
```

Atributos do objeto modelo_s

```
attributes(modelo_s)
```

```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
##
## $class
## [1] "lm"
```

```
modelo_s$coefficients
```

```
## (Intercept)          RM
##  -38.421400    9.697272
```

Comparando os valores atuais e valores previstos

```
resultado <- data.frame(Valor_atual=treino$MEDV, Valor_previsto=predict(modelo_s))
head(resultado)
```

```
##   Valor_atual Valor_previsto
## 1        21.6       23.84478
## 2        34.7       31.25350
## 3        28.7       23.93206
## 4        27.1       21.43016
## 5        16.5       16.18394
## 6        15.0       23.41810
```

```
cor(resultado)
```

```
##                Valor_atual Valor_previsto
## Valor_atual      1.0000000      0.7059531
## Valor_previsto   0.7059531      1.0000000
```

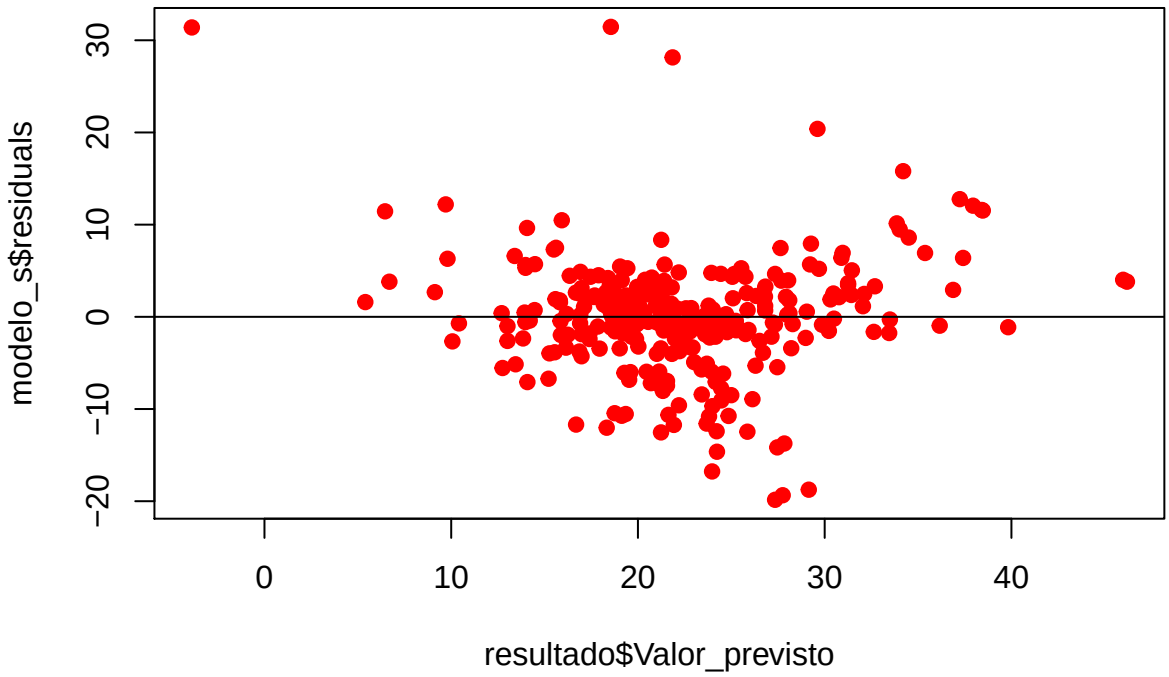
Mean absolute percentage error (MAPE)

```r
mape <- mean(abs(modelo_s$residuals)/resultado$Valor_atual)*100
mape
```

```
## [1] 25.37316
```

Gráfico - valores previstos e resíduos

```r
plot(resultado$Valor_previsto, modelo_s$residuals,pch=21,bg="red",col="red")
abline(0,0)
```



———Criando o modelo de regressão linear múltipla———-

Equação de Regressão y = a + b0x0 + b1x1 (múltipla)

Treinando o modelo (dados de treino)

```r
modelo_m <- lm(MEDV ~ ., data = treino)
modelo_m
```

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

```
## lm(formula = MEDV ~ ., data = treino)
##
## Coefficients:
## (Intercept)         CRIM           ZN        INDUS         CHAS
##   30.982217    -0.095854     0.047214     0.031715     3.517811
##         NOX           RM          AGE          DIS          RAD
##  -17.545239     4.177832     0.005161    -1.423337     0.296775
##         TAX      PTRATIO            B        LSTAT
##   -0.011945    -0.814019     0.008742    -0.539535
```

```
summary(modelo_m)
```

```
##
## Call:
## lm(formula = MEDV ~ ., data = treino)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2369 -2.6743 -0.5052  1.4325 24.7952
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.982217   6.542509   4.736 3.48e-06 ***
## CRIM         -0.095854   0.033856  -2.831 0.004975 **
## ZN            0.047214   0.016743   2.820 0.005147 **
## INDUS         0.031715   0.073246   0.433 0.665356
## CHAS          3.517811   0.999866   3.518 0.000507 ***
## NOX         -17.545239   4.788995  -3.664 0.000297 ***
## RM            4.177832   0.551500   7.575 5.29e-13 ***
## AGE           0.005161   0.016102   0.321 0.748801
## DIS          -1.423337   0.239625  -5.940 8.47e-09 ***
## RAD           0.296775   0.079016   3.756 0.000210 ***
## TAX          -0.011945   0.004485  -2.664 0.008179 **
## PTRATIO      -0.814019   0.157560  -5.166 4.55e-07 ***
## B             0.008742   0.003326   2.629 0.009051 **
## LSTAT        -0.539535   0.062421  -8.644 4.33e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.434 on 279 degrees of freedom
## Multiple R-squared:  0.769,  Adjusted R-squared:  0.7582
## F-statistic: 71.45 on 13 and 279 DF,  p-value: < 2.2e-16
```

Comparando os valores atuais e valores previstos

```
resultado <- data.frame(Valor_atual=treino$MEDV, Valor_previsto=predict(modelo_m))
head(resultado)
```

```
##   Valor_atual Valor_previsto
## 1        21.6       24.88991
## 2        34.7       30.71135
```

```
## 3        28.7       25.20136
## 4        27.1       18.95982
## 5        16.5       10.62007
## 6        15.0       18.49567
```

```
cor(resultado)
```

```
##                Valor_atual Valor_previsto
## Valor_atual      1.0000000      0.8769286
## Valor_previsto   0.8769286      1.0000000
```

Mean absolute percentage error (MAPE)

```
mape <- mean(abs(modelo_m$residuals)/resultado$Valor_atual)*100
mape
```

```
## [1] 15.01361
```

————————-Melhorando a performance do modelo————————————

Eliminando as variáveis que não são relevantes para o modelo

```
modelo_v2 <- lm(MEDV ~ .-INDUS-AGE, data = treino)
summary(modelo_v2)
```

```
##
## Call:
## lm(formula = MEDV ~ . - INDUS - AGE, data = treino)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2465 -2.6453 -0.5134  1.3628 24.9310
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.574187   6.470451   4.725 3.64e-06 ***
## CRIM         -0.095876   0.033746  -2.841 0.004825 **
## ZN            0.045409   0.016336   2.780 0.005808 **
## CHAS          3.575014   0.991217   3.607 0.000367 ***
## NOX         -16.549818   4.399535  -3.762 0.000205 ***
## RM            4.198644   0.529893   7.924 5.43e-14 ***
## DIS          -1.461503   0.226725  -6.446 4.98e-10 ***
## RAD           0.283024   0.074223   3.813 0.000169 ***
## TAX          -0.010911   0.003895  -2.801 0.005448 **
## PTRATIO      -0.804218   0.155894  -5.159 4.70e-07 ***
## B             0.008789   0.003305   2.659 0.008289 **
## LSTAT        -0.531588   0.057800  -9.197  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.421 on 281 degrees of freedom
## Multiple R-squared:  0.7688, Adjusted R-squared:  0.7597
## F-statistic: 84.93 on 11 and 281 DF,  p-value: < 2.2e-16
```

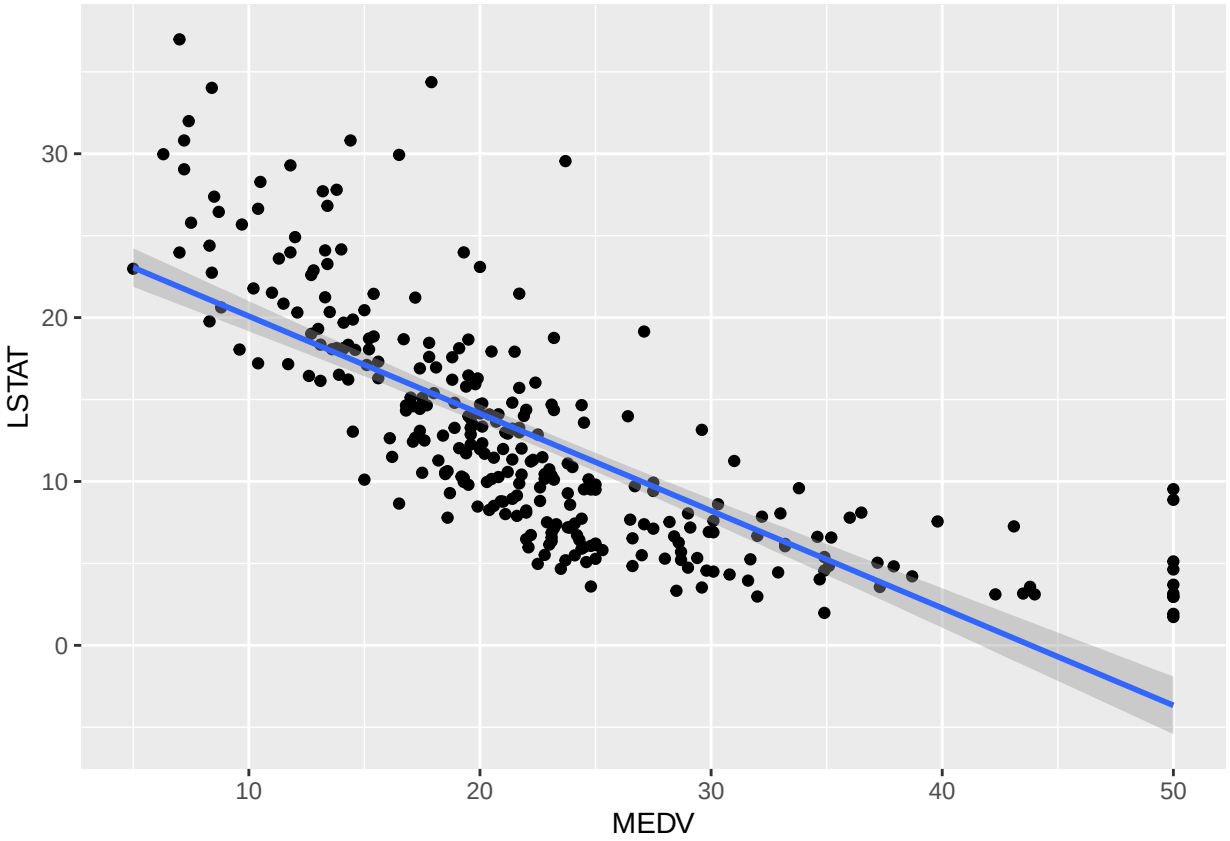
Aplicando uma transformação (log transformation) na variável alvo (MEDV)

```
modelo_v3 <- lm(log(MEDV) ~ .-INDUS-AGE, data = treino)
summary(modelo_v3)
```

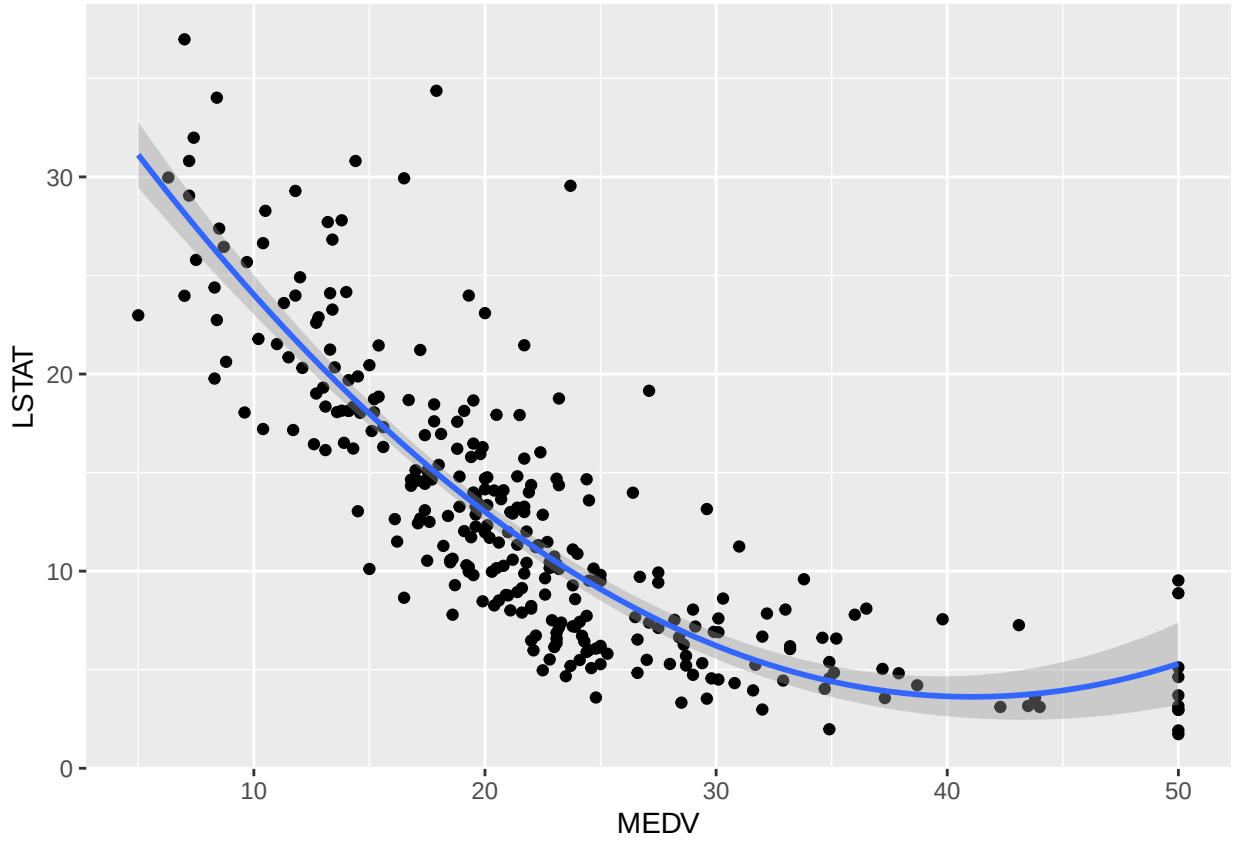
```
##
## Call:
## lm(formula = log(MEDV) ~ . - INDUS - AGE, data = treino)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54626 -0.09904 -0.01660  0.08871  0.89775
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.9119338  0.2601688  15.036  < 2e-16 ***
## CRIM        -0.0086487  0.0013569  -6.374 7.53e-10 ***
## ZN           0.0011044  0.0006568   1.681 0.093789 .
## CHAS         0.1529807  0.0398556   3.838 0.000153 ***
## NOX         -0.7303675  0.1768998  -4.129 4.81e-05 ***
## RM           0.1036971  0.0213063   4.867 1.89e-06 ***
## DIS         -0.0520327  0.0091163  -5.708 2.91e-08 ***
## RAD          0.0124713  0.0029844   4.179 3.92e-05 ***
## TAX         -0.0005502  0.0001566  -3.513 0.000517 ***
## PTRATIO     -0.0321417  0.0062683  -5.128 5.47e-07 ***
## B            0.0004283  0.0001329   3.223 0.001419 **
## LSTAT       -0.0297114  0.0023241 -12.784  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1778 on 281 degrees of freedom
## Multiple R-squared:  0.8208, Adjusted R-squared:  0.8138
## F-statistic:   117 on 11 and 281 DF,  p-value: < 2.2e-16
```

Verificando a não-linearidade do modelo

```
ggplot(treino, aes(MEDV, LSTAT)) +
  geom_point() +
  geom_smooth(method = "lm")
```




```
ggplot(treino, aes(MEDV, LSTAT)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2))
```



Criando uma variável quadrática

Construindo um outro modelo com a variável quadrática

```
treino$RM2 <- treino$RM ^ 2

modelo_v4 <- lm(log(MEDV) ~ .-INDUS-AGE, data = treino)
summary(modelo_v4)
```

```
##
## Call:
## lm(formula = log(MEDV) ~ . - INDUS - AGE, data = treino)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51930 -0.09056 -0.00663  0.08305  0.84723
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.6910086  0.5298184  12.629  < 2e-16 ***
## CRIM        -0.0091264  0.0012840  -7.108 9.82e-12 ***
## ZN           0.0007038  0.0006240   1.128 0.260348
## CHAS         0.1551004  0.0376418   4.120 4.98e-05 ***
## NOX         -0.7052312  0.1671204  -4.220 3.31e-05 ***
```

```
## RM          -0.8297607  0.1589444  -5.220 3.48e-07 ***
## DIS         -0.0423160  0.0087646  -4.828 2.27e-06 ***
## RAD          0.0109973  0.0028295   3.887 0.000127 ***
## TAX         -0.0004834  0.0001483  -3.259 0.001258 **
## PTRATIO     -0.0266282  0.0059927  -4.443 1.28e-05 ***
## B            0.0003931  0.0001257   3.128 0.001945 **
## LSTAT       -0.0297667  0.0021949 -13.562  < 2e-16 ***
## RM2          0.0736558  0.0124408   5.920 9.37e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1679 on 280 degrees of freedom
## Multiple R-squared:  0.8407, Adjusted R-squared:  0.8339
## F-statistic: 123.2 on 12 and 280 DF,  p-value: < 2.2e-16
```

Comparando os valores atuais e valores previstos

```
resultado <- data.frame(Valor_atual=treino$MEDV, Valor_previsto=exp(predict(modelo_v4)))
head(resultado)
```

```
##   Valor_atual Valor_previsto
## 1        21.6       23.89506
## 2        34.7       31.68658
## 3        28.7       25.75702
## 4        27.1       17.24556
## 5        16.5       12.12213
## 6        15.0       16.59363
```

```
cor(resultado)
```

```
##                Valor_atual Valor_previsto
## Valor_atual      1.0000000      0.9077325
## Valor_previsto   0.9077325      1.0000000
```

Mean absolute percentage error (MAPE)

```
mape <- mean(abs(resultado$Valor_atual-resultado$Valor_previsto)/resultado$Valor_atual)*100
mape
```

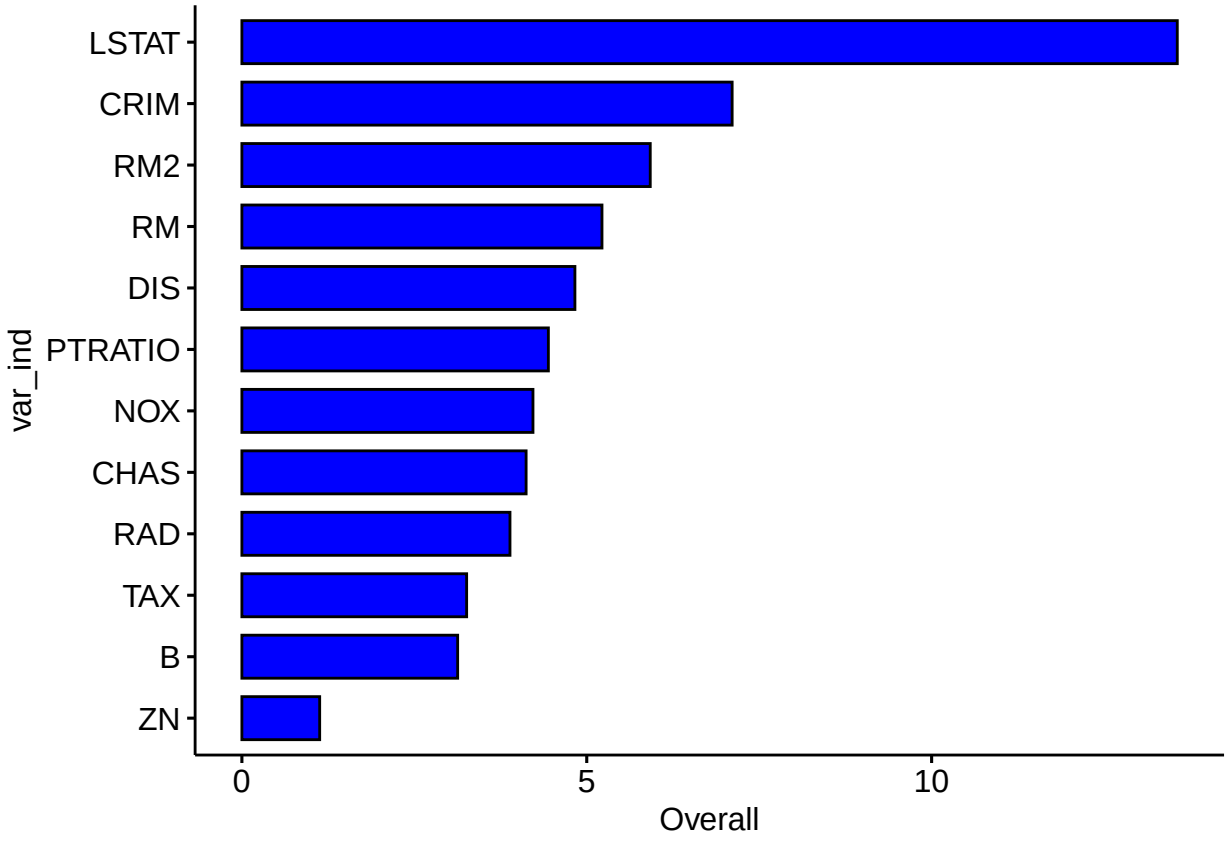
```
## [1] 11.42837
```

Analisando a importância das variáveis independentes

```
importancia <- varImp(modelo_v4, varImp.train=TRUE)
print(importancia)
```

```
##           Overall
## CRIM     7.107788
## ZN       1.127851
## CHAS     4.120426
## NOX      4.219899
## RM       5.220445
## DIS      4.828054
## RAD      3.886684
## TAX      3.258577
## PTRATIO  4.443463
## B        3.128161
## LSTAT   13.561696
## RM2      5.920491
```

```
importancia <- data.frame(var_ind = row.names(importancia), importancia)
ggbarplot(importancia, x='var_ind', y="Overall", sort.val="asc",
          orientation='horiz', fill='blue')
```



Testando o modelo

Acrescentando a variável RM2 no conjunto teste

```
teste$RM2 <- teste$RM ^ 2
```

Fazendo as previsões usando o modelo_v4

```
previsao <- predict(modelo_v4, teste)
View(previsao)
```

Mean absolute percentage error (MAPE)

```
mape <- mean(abs(teste$MEDV-exp(previsao))/teste$MEDV)*100
mape
```

```
## [1] 14.51062
```

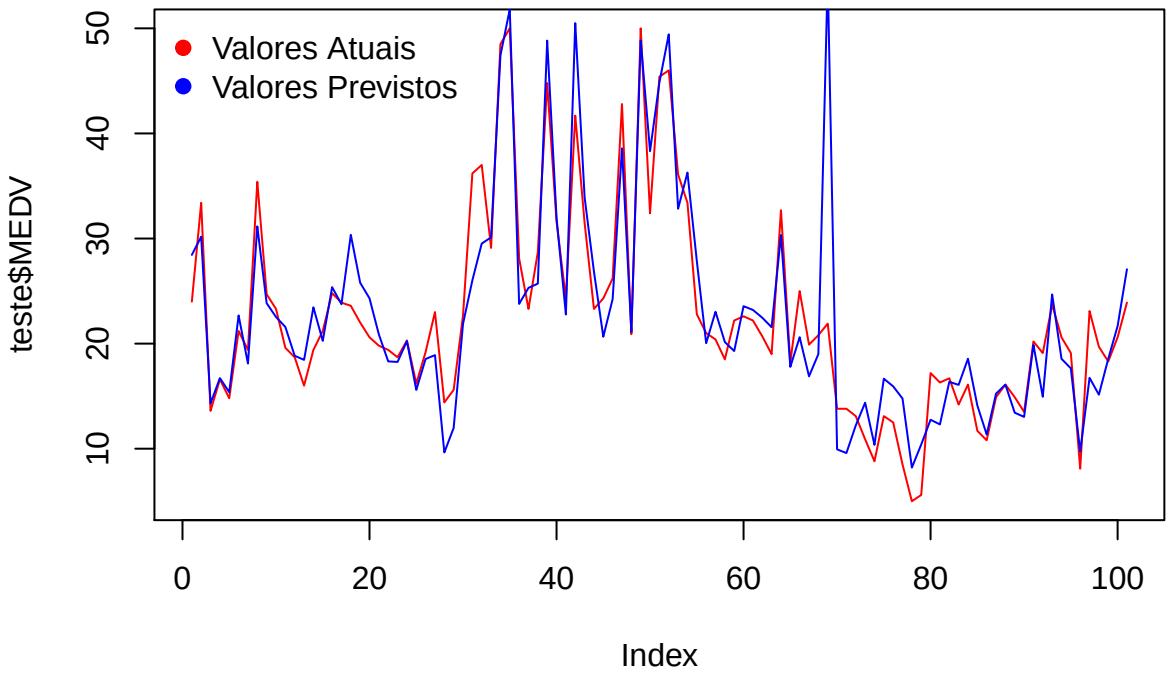
Root mean square error (RMSE)

```
rmse <- sqrt(sum((exp(previsao)-teste$MEDV)^2)/length(teste$MEDV))
rmse
```

```
## [1] 4.570381
```

Visualizando as diferenças entre valores atuais e previstos

```
plot(teste$MEDV,type="l",col="red")
lines(exp(previsao),col="blue")
legend("topleft",
       legend = c("Valores Atuais", "Valores Previstos"),
       col = c('red', 'blue'),
       pch = c(19,19),
       bty = "n",
       inset = c(0,0))
```



——SALVANDO O MODELO——

```
saveRDS(modelo_v4,"modelo_regressao.rds")
```

——CARREGANDO O MODELO——

```
regr <- readRDS("modelo_regressao.rds")
previsao_final <- predict(regr, teste)
```