Lista 08

Matheus Cougias

15/02/2021

```
dados <- read.delim("boston_corrected.txt")
dados <- subset(dados, select = -c(OBS., TOWN, TOWN., TRACT, LON, LAT, CMEDV))
dadosLog <- data.frame(logMEDV=log(dados$MEDV),logCRIM=log(dados$CRIM),logZN=log(dados$ZN+1),logINDUS=l
                       logAGE=log(dados$AGE),logDIS=log(dados$DIS),logRAD=log(dados$RAD),logTAX=log(dad
```

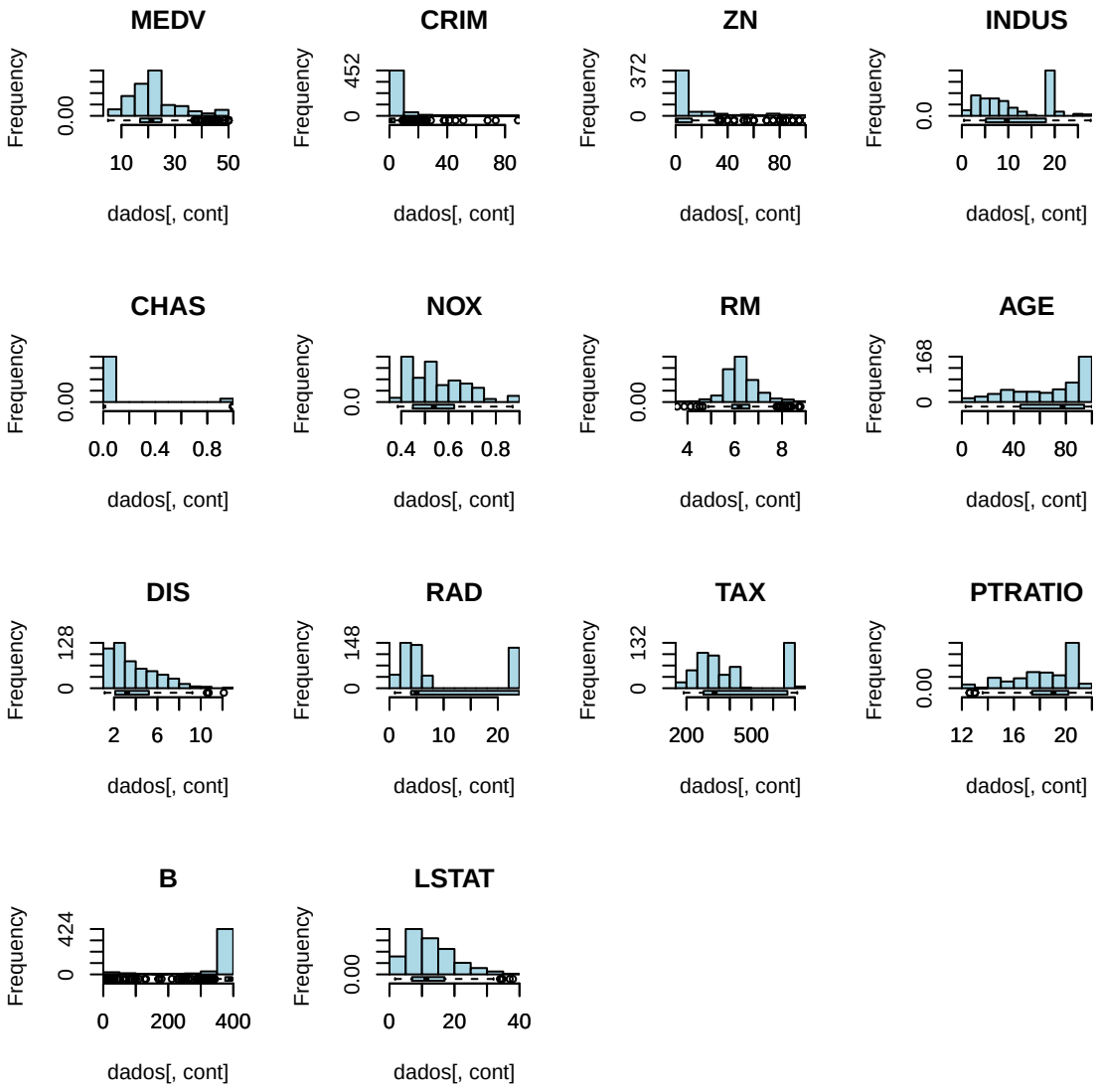
Análise inicial Na análise dos histogramas, é possível identificar que as variáveis, em sua maioria, não seguem uma distribuição normal. Em relação aos gráficos de dispersão, foi identificado que algumas variáveis seguem valores fixados. Já, no gráfico de correlação existe uma forte correlação entre as variáveis TAX e RAD, além da variável DIS com INDUS, NOX e AGE.

```
require(packHV)
```

```
## Loading required package: packHV
```

```
## Loading required package: survival
```

```
par(mfrow=c(4,4))
for (cont in 1:14)
{
  hist_boxplot(dados[,cont], main = names(dados)[cont],
               col = "light blue")
}
par(mfrow=c(4,4))
```

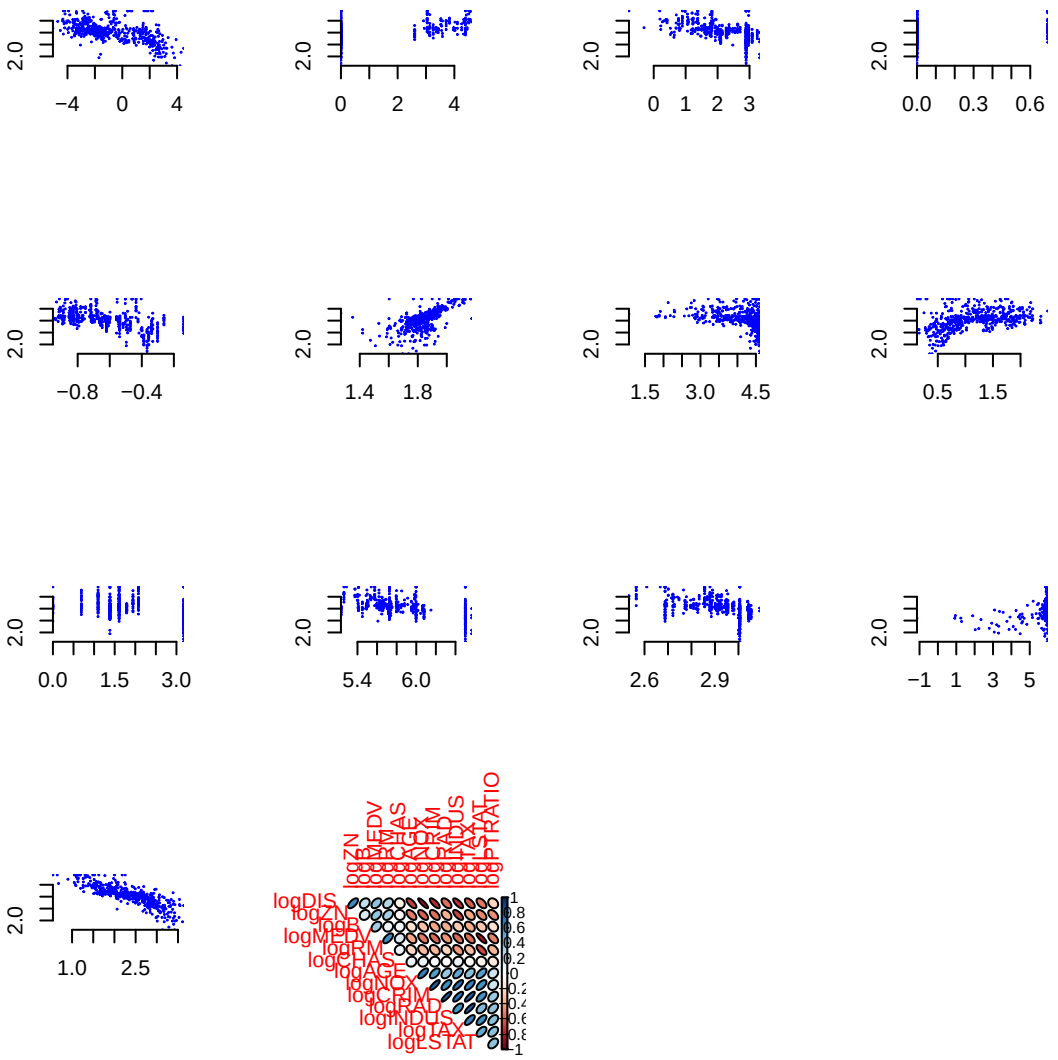



```
for (var in names(dadosLog)[2:14])
{
  equacao <- as.formula(paste("logMEDV", var, sep = " ~ "))
  plot(equacao, pch = 19, col = "blue", cex = 0.1, data = dadosLog)
}
require(corrplot)
```

```
## Loading required package: corrplot
```

```
## corrplot 0.84 loaded
```

```
corMat <- cor(dadosLog)
corrplot(corMat, method = "ellipse", type = "upper", order = "AOE", diag=FALSE, addgrid.col=NA, outline=
```



Ajuste do Modelo de Regressao Linear Multipla Para definir então qual base de dados será utilizada na regressão final, realizei duas regressões distintas para avaliar a aplicação do logaritmo. Ao aplicar a regressão linear múltipla com todas as variáveis da base, o modelo onde o logaritmo foi utilizad apresentou R^2 de 76,96%, resultado levemente superior ao modelo original, que teve o R^2 de 74,06%. Dessa maneira, decidi utilizar a base de dados com a transformação logaritmica.

```
modeloLog <- lm(logMEDV ~ logCRIM + logZN + logINDUS + logCHAS + logNOX + logRM + logAGE + logDIS + logR
                logTAX + logPTRATIO + logB + logLSTAT, data=dadosLog)
summary(modeloLog)
```

```
##
## Call:
## lm(formula = logMEDV ~ logCRIM + logZN + logINDUS + logCHAS +
##     logNOX + logRM + logAGE + logDIS + logRAD + logTAX + logPTRATIO +
##     logB + logLSTAT, data = dadosLog)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.95032 -0.10303 -0.00257  0.10887  0.82889
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.451574   0.437232  12.468  < 2e-16 ***
## logCRIM     -0.031427   0.011183  -2.810 0.005148 **
## logZN       -0.016429   0.008551  -1.921 0.055274 .
## logINDUS     0.007949   0.021823   0.364 0.715834
## logCHAS      0.142429   0.052235   2.727 0.006625 **
## logNOX      -0.367113   0.107536  -3.414 0.000693 ***
## logRM        0.389532   0.109522   3.557 0.000412 ***
## logAGE       0.042937   0.022258   1.929 0.054296 .
## logDIS      -0.138655   0.036043  -3.847 0.000135 ***
## logRAD       0.100878   0.022197   4.545 6.93e-06 ***
## logTAX      -0.192292   0.045509  -4.225 2.84e-05 ***
## logPTRATIO  -0.608403   0.094695  -6.425 3.12e-10 ***
## logB         0.053979   0.012647   4.268 2.37e-05 ***
## logLSTAT    -0.416449   0.025819 -16.130  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1988 on 492 degrees of freedom
## Multiple R-squared:  0.7696, Adjusted R-squared:  0.7635
## F-statistic: 126.4 on 13 and 492 DF,  p-value: < 2.2e-16
```

Diagnostico de Multicolinearidade - Estatistica VIF Com o logaritmo escolhido, através da análise do VIF percebe-se que as variáveis CRIM, NOX e DIS são as que possuem maior colinearidade, ou seja, carregam informações que podem ser explicadas pelas demais variáveis do modelo.

```
require(car)
```

```
## Loading required package: car
```

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

```
1 - 1/vif(modeloLog)
```

```
##    logCRIM      logZN   logINDUS    logCHAS     logNOX      logRM     logAGE
## 0.86613805 0.59263610 0.72781198 0.07468206 0.83330488 0.48292797 0.57651559
##     logDIS     logRAD     logTAX logPTRATIO       logB   logLSTAT
## 0.79307218 0.79246685 0.75949611 0.43987992 0.18226256 0.67487689
```

Ajuste de multicolinearidade

```
modeloLog <- lm(logMEDV ~ logCRIM+logZN+logINDUS+logCHAS+logNOX+logRM+logAGE+logDIS+logRAD+logTAX+logPTR
modeloFinalLog <- step(modeloLog)
```

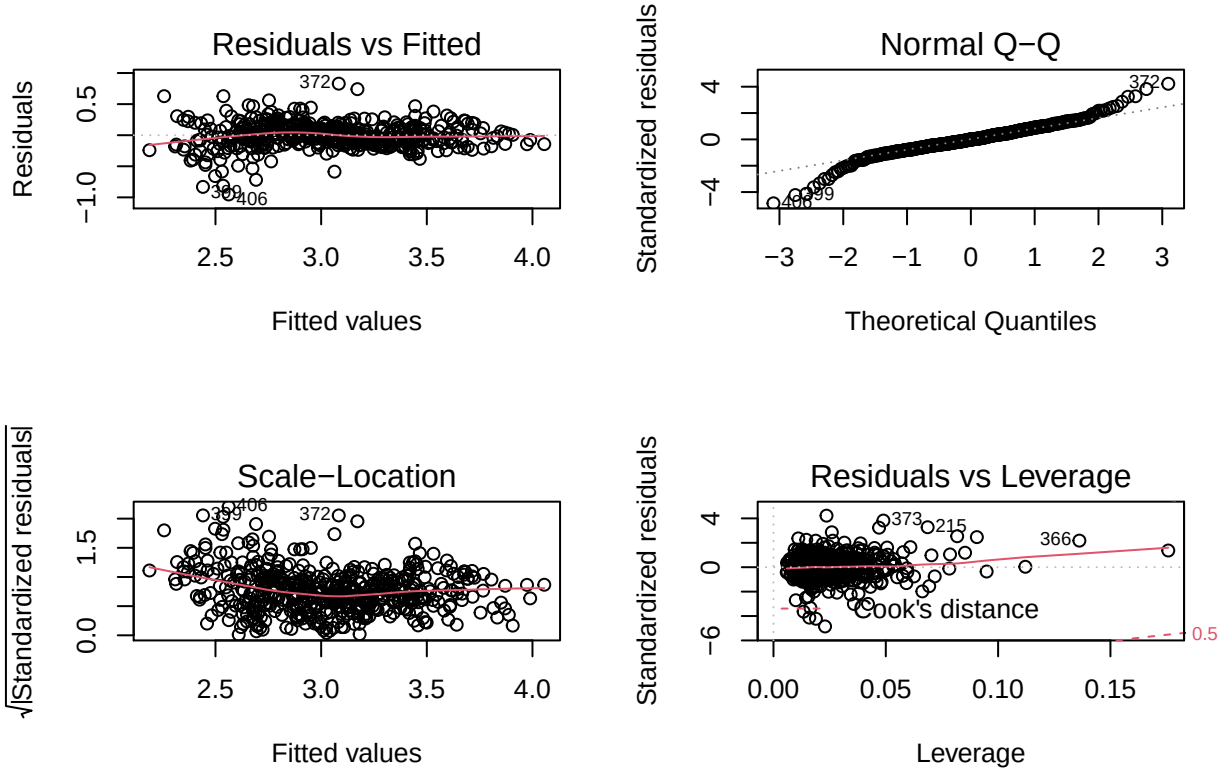
Análise do coeficiente de determinacao preditivo Através do método step, algumas variáveis puderam ser retiradas do modelo, de forma que o modelo final apresentou um R^2 de 76,95%, o que não pode ser classificado como um resultado realmente satisfatório. Pode-se perceber que o modelo não compreendeu

bem os dados através da visualização do gráfico Normal Q-Q, onde existem diversos pontos fora da reta normal. Por ultimo, busca-se analisar o coefiente de determinaçao preditivo, ou seja, quanto o modelo consegue prever futuras observações. O valor de R^2 preditivo encontrado foi de 75,48%.

```
summary(modeloFinalLog)
```

```
##
## Call:
## lm(formula = logMEDV ~ logCRIM + logZN + logCHAS + logNOX + logRM +
##     logAGE + logDIS + logRAD + logTAX + logPTRATIO + logB + logLSTAT,
##     data = dadosLog)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95294 -0.10198 -0.00164  0.11030  0.82871
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.454545   0.436772  12.488  < 2e-16 ***
## logCRIM     -0.030986   0.011108  -2.790 0.005481 **
## logZN       -0.017300   0.008203  -2.109 0.035453 *
## logCHAS      0.145037   0.051696   2.806 0.005221 **
## logNOX      -0.360863   0.106065  -3.402 0.000723 ***
## logRM        0.385928   0.108978   3.541 0.000436 ***
## logAGE       0.041863   0.022042   1.899 0.058119 .
## logDIS      -0.140854   0.035503  -3.967 8.34e-05 ***
## logRAD       0.100410   0.022140   4.535 7.23e-06 ***
## logTAX      -0.189414   0.044778  -4.230 2.79e-05 ***
## logPTRATIO  -0.604015   0.093842  -6.436 2.90e-10 ***
## logB         0.054122   0.012630   4.285 2.20e-05 ***
## logLSTAT    -0.415228   0.025578 -16.234  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1986 on 493 degrees of freedom
## Multiple R-squared:  0.7695, Adjusted R-squared:  0.7639
## F-statistic: 137.2 on 12 and 493 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(modeloFinalLog)
```




```
modeloLog <- lm(logMEDV ~ logCRIM + logZN + logCHAS + logNOX + logRM + logAGE + logDIS + logRAD +
                  logTAX + logPTRATIO + logB + logLSTAT, data=dadosLog)
X <- model.matrix(modeloLog)
H <- X%*%solve(t(X)%*%X)%*%t(X)
resDel <- residuals(modeloLog)/(1-diag(H))
SQT    <- sum( (dadosLog$logMEDV - mean(dadosLog$logMEDV))^2 )
SQRes  <- sum(resDel^2)
(R2Pred <- 1 - SQRes/SQT)
```

```
## [1] 0.7548254
```

Críticas e sugestões ao modelo final O modelo final encontrado foi o seguinte: logMEDV ~ logCRIM + logZN + logCHAS + logNOX + logRM + logAGE + logDIS + logRAD + logTAX + logPTRATIO + logB + logLSTAT

Como descrito durante a resolução do problema, não foi fácil a identificação entre utilizar ou não a transformação logaritmica na base de dados, então baseei minha decisão em um modelo inicial montado para cada caso. Provavelmente, pelos p-valores de quase todas as variáveis serem muito baixos, o modelo encontra uma certa dificuldade de cortar variáveis sem que o R^2 seja afetado. Dessa maneira, uma possibilidade de melhora nos resultados está na utilização de outra transformação que não seja o logaritmo. Outra opção está na utilização de regressões não lineares, que podem compreender de forma melhor os dados.