Análise do conjunto de Dados Boston Housing

```
> library(dplyr)
> library(ggcorrplot)
```

```
> ### Reading the Boston Housing data ###
> Boston = read.csv("http://stat.wharton.upenn.edu/~khyuns/stat431/BostonHousing.txt")
> attach(Boston)
> head(Boston)
```

Estes dados estão contidos também no pacote MASS. Eles podem ser lidos da seguinte forma:

```
> require(MASS)
> data(Boston)
> # Digitando
> help(Boston)
> # vai mostrar a descrição dos dados
> head(Boston)
```

	crim <dbl>	zn <dbl>	indus <dbl>	chas <int>	nox <dbl>	rm <dbl>	age <dbl>	dis <dbl>	rad <int>
1	0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3
6	0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3

6 rows | 1-10 of 15 columns

Descrição dos Dados

1. CRIM: per capita crime rate by town
2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS: proportion of non-retail business acres per town
4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX: nitric oxides concentration (parts per 10 million)
6. RM: average number of rooms per dwelling
7. AGE: proportion of owner-occupied units built prior to 1940
8. DIS: weighted distances to five Boston employment centres
9. RAD: index of accessibility to radial highways
10. TAX: full-value property-tax rate per $10,000
11. PTRATIO: pupil-teacher ratio by town
12. B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT: % lower status of the population
14. MEDV: Median value of owner-occupied homes in $1000's

Sugestão de Análise

Exploração dos Dados

- Sumários estatísticos (summary, dim, cor, etc)
- Gráficos exploratórios

Ajuste Modelo Linear

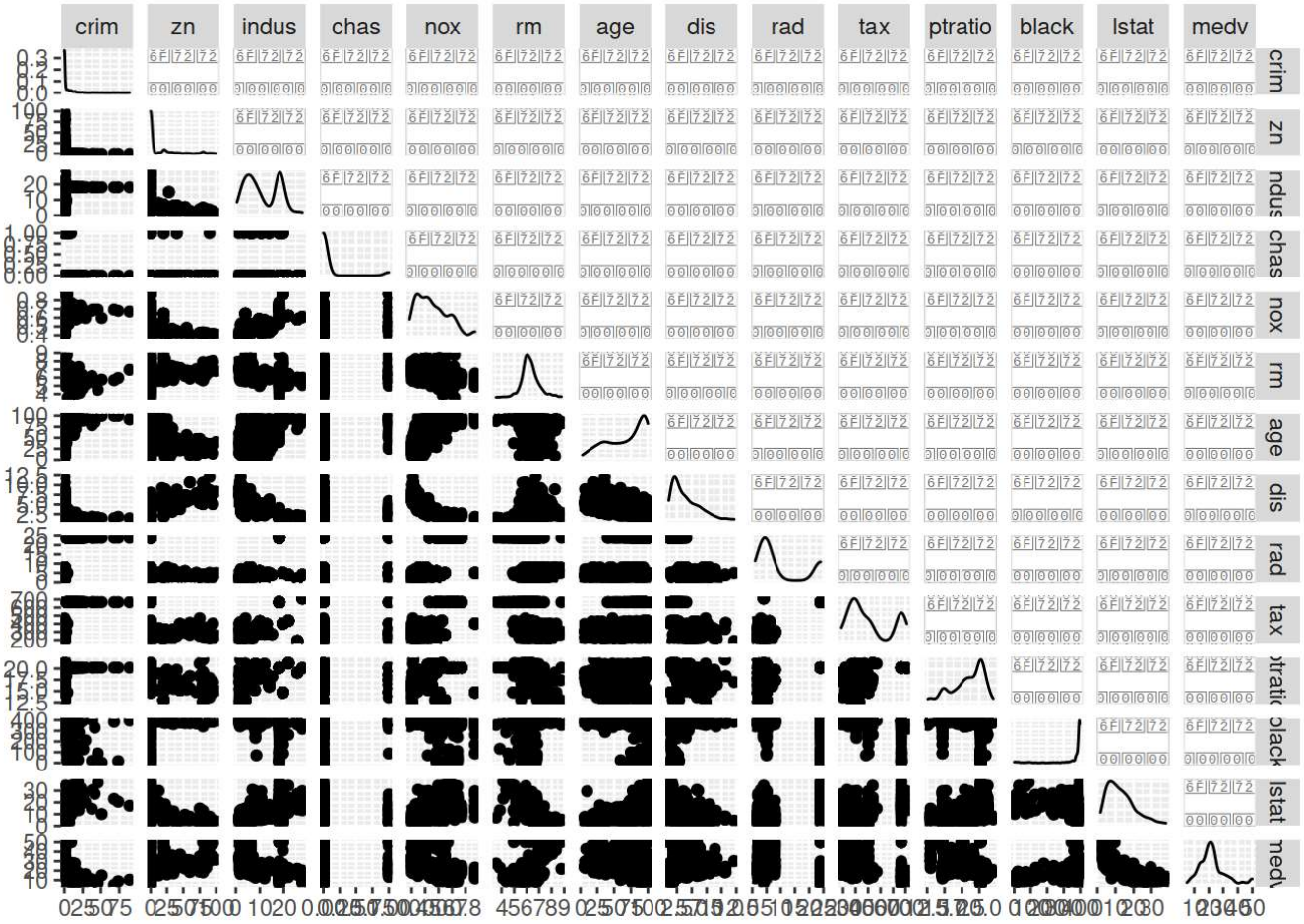
- Crie dois subconjuntos dos dados: `train` e `test` de modo a poder avaliar a capacidade preditora de seu modelo.

```
> set.seed(1234)
> train <- sample_frac(Boston, 0.7)
> sid <- as.numeric(rownames(train))  # because rownames() returns character
> test <- Boston[-sid, ]
```

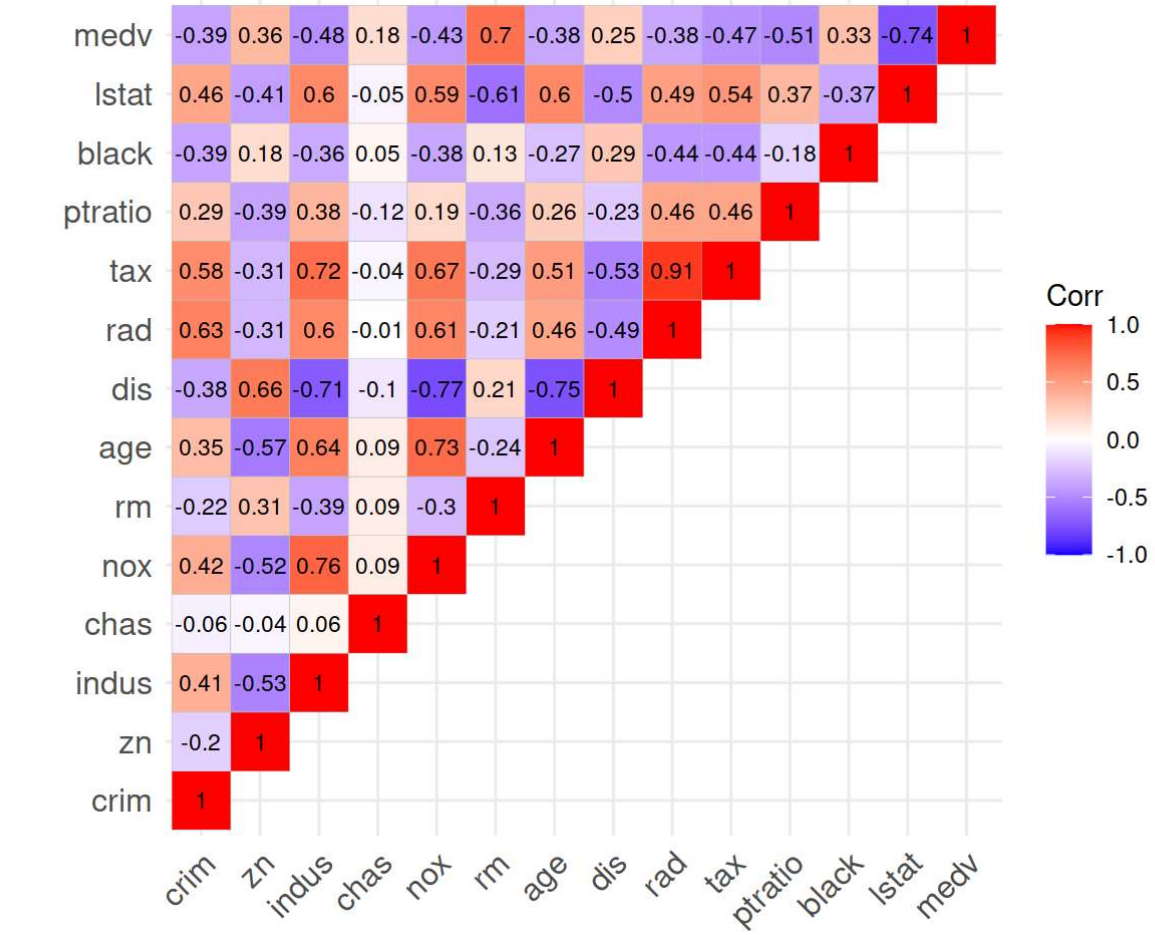
```
> str(train)
'data.frame':   354 obs. of  14 variables:
 $ crim   : num  0.015 0.0396 67.9208 0.1487 0.1057 ...
 $ zn     : num  90 0 0 0 0 0 0 0 0 0 ...
 $ indus  : num  1.21 5.19 18.1 8.56 27.74 ...
 $ chas   : int  1 0 0 0 0 0 0 0 0 0 ...
 $ nox    : num  0.401 0.515 0.693 0.52 0.609 0.52 0.7 0.624 0.718 0.693 ...
 $ rm     : num  7.92 6.04 5.68 6.73 5.98 ...
 $ age    : num  24.8 34.5 100 79.9 98.8 54.4 97 97.9 95.3 77.8 ...
 $ dis    : num  5.88 5.99 1.43 2.78 1.87 ...
 $ rad    : int  1 5 24 5 4 5 24 4 24 24 ...
 $ tax    : num  198 224 666 384 711 384 666 437 666 666 ...
 $ ptratio: num  13.6 20.2 20.2 20.9 20.1 20.9 20.2 21.2 20.2 20.2 ...
 $ black  : num  396 397 385 395 390 ...
 $ lstat  : num  3.16 8.01 22.98 9.42 18.07 ...
 $ medv   : num  50 21.1 5 27.5 13.6 21.7 9.7 23 14.2 6.3 ...
> str(test)
'data.frame':   152 obs. of  14 variables:
 $ crim   : num  0.043 0.107 8.983 3.85 5.202 ...
 $ zn     : num  80 80 0 0 0 0 0 0 0 0 ...
 $ indus  : num  1.91 1.91 18.1 18.1 18.1 18.1 18.1 18.1 18.1 18.1 ...
 $ chas   : int  0 0 1 1 1 0 0 0 0 1 ...
 $ nox    : num  0.413 0.413 0.77 0.77 0.77 0.77 0.77 0.77 0.77 0.77 ...
 $ rm     : num  5.66 5.94 6.21 6.39 6.13 ...
 $ age    : num  21.9 19.5 97.4 91 83.4 81.3 88 91.1 96.2 89 ...
 $ dis    : num  10.59 10.59 2.12 2.51 2.72 ...
 $ rad    : int  4 4 24 24 24 24 24 24 24 24 ...
 $ tax    : num  334 334 666 666 666 666 666 666 666 666 ...
 $ ptratio: num  22 22 20.2 20.2 20.2 20.2 20.2 20.2 20.2 20.2 ...
 $ black  : num  383 376 378 391 395 ...
 $ lstat  : num  8.05 5.57 17.6 13.27 11.48 ...
 $ medv   : num  18.2 20.6 17.8 21.7 22.7 22.6 25 19.9 20.8 16.8 ...
> # write.csv(test,file='datasets/boston_test_prof.csv',row.names
> # = FALSE,quote = FALSE)
> # write.csv(train,file='datasets/boston_train_prof.csv',row.names=FALSE,quote=FALSE)
```

Inspecionando correlações entre as variáveis

```
> library(GGally)
> ggpairs(Boston)
```




```
> library(ggcorrplot)
> ggcorrplot(cor(Boston), show.diag = TRUE, lab = TRUE, lab_size = 3,
    type = "upper")
```



Percebemos correlações fortes entre variáveis explicativas; isso não é bom e vai atrapalhar o modelo.

- Ajuste de Modelo Linear (`lm`) — queremos prever o valor mediano das residências `MEDV` .

- Análise do ajuste:
    - `summary(modelo)`
    - gráficos diagnósticos

Seleção de Modelos (Variáveis Explicativas)

- Seleção de Modelos (variáveis explicativas)

```
> model.1 <- lm(medv ~ ., data = train)
> summary(model.1)

Call:
lm(formula = medv ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-16.2210  -2.5130  -0.7122   1.8295  26.9382

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  26.399485   6.347070   4.159 4.05e-05 ***
crim         -0.111639   0.035323  -3.160 0.001717 **
zn            0.044388   0.015841   2.802 0.005367 **
indus        -0.008549   0.071304  -0.120 0.904641
chas          2.793031   0.972463   2.872 0.004333 **
nox         -17.494132   4.314041  -4.055 6.22e-05 ***
rm            4.907947   0.542364   9.049  < 2e-16 ***
age          -0.016715   0.015386  -1.086 0.278074
dis          -1.441577   0.229553  -6.280 1.03e-09 ***
rad           0.254051   0.075422   3.368 0.000843 ***
tax          -0.011540   0.004309  -2.678 0.007765 **
ptratio      -0.838551   0.154130  -5.441 1.02e-07 ***
black         0.009602   0.003200   3.000 0.002896 **
lstat        -0.341352   0.062357  -5.474 8.55e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.656 on 340 degrees of freedom
Multiple R-squared:  0.7508,    Adjusted R-squared:  0.7413
F-statistic: 78.79 on 13 and 340 DF,  p-value: < 2.2e-16
```

VIF

Para uma dada variável preditora (p), podemos aferir a multicolineraridade calculando um índice chamado “fator de inflação de variância” (VIF), que mede o quanto a variância do coeficiente de regressão é inflacionado devido a multicolineraridade no modelo.

O menor valor possível de VIF é um (ausência de multicolinearidade). Como regra geral, um valor de VIF que excede 5 ou 10 indica uma quantidade problemática de multicolinearidade.

Quando temos multicolinearidade, as variáveis envolvidas devem ser removidas, já que a presença de multicolinearidade implica que a informação que esta variável provê sobre a resposta é redundante na presença das outras variáveis.

```
> car::vif(model.1)
    crim       zn    indus     chas      nox       rm      age      dis
1.770761 2.164574 3.969857 1.087845 4.259040 2.274295 3.055292 3.746143
     rad      tax  ptratio    black    lstat
7.094406 8.823177 1.807068 1.361045 3.463498
```

Inicialmente detectamos a variável *tax*

```
> model.2 <- update(model.1, medv ~ . - tax)
> summary(model.2)

Call:
lm(formula = medv ~ crim + zn + indus + chas + nox + rm + age +
    dis + rad + ptratio + black + lstat, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-16.6866  -2.7722  -0.6172   1.8859  26.8975

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  24.657384   6.370527   3.871  0.00013 ***
crim         -0.110312   0.035638  -3.095  0.00213 **
zn            0.036322   0.015692   2.315  0.02122 *
indus        -0.099739   0.063212  -1.578  0.11553
chas          3.065232   0.975849   3.141  0.00183 **
nox         -18.168004   4.345495  -4.181 3.70e-05 ***
rm            4.991936   0.546335   9.137  < 2e-16 ***
age          -0.017086   0.015524  -1.101  0.27184
dis          -1.435859   0.231611  -6.199 1.64e-09 ***
rad           0.098852   0.048705   2.030  0.04317 *
ptratio      -0.874388   0.154931  -5.644 3.50e-08 ***
black         0.009902   0.003227   3.068  0.00233 **
lstat        -0.340533   0.062918  -5.412 1.17e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.698 on 341 degrees of freedom
Multiple R-squared:  0.7455,    Adjusted R-squared:  0.7366
F-statistic: 83.25 on 12 and 341 DF,  p-value: < 2.2e-16
```

```
> car::vif(model.2)
    crim       zn    indus     chas      nox       rm      age      dis
1.770412 2.086330 3.064516 1.075961 4.244550 2.266690 3.055045 3.745819
     rad  ptratio    black    lstat
2.905852 1.793446 1.359381 3.463415
```

```
> model.3 <- update(model.2, medv ~ . - age)
> summary(model.3)

Call:
lm(formula = medv ~ crim + zn + indus + chas + nox + rm + dis +
    rad + ptratio + black + lstat, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-16.3648  -2.8158  -0.5752   1.8242  26.5875

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  25.203891   6.353108   3.967 8.86e-05 ***
crim         -0.110334   0.035649  -3.095  0.00213 **
zn            0.038708   0.015546   2.490  0.01325 *
indus        -0.101751   0.063205  -1.610  0.10835
chas          2.998826   0.974283   3.078  0.00225 **
nox         -19.480487   4.179960  -4.660 4.53e-06 ***
rm            4.874112   0.535909   9.095  < 2e-16 ***
dis          -1.362138   0.221782  -6.142 2.27e-09 ***
rad           0.102206   0.048624   2.102  0.03629 *
ptratio      -0.885261   0.154663  -5.724 2.28e-08 ***
black         0.009622   0.003218   2.990  0.00299 **
lstat        -0.360704   0.060208  -5.991 5.29e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.7 on 342 degrees of freedom
Multiple R-squared:  0.7446,    Adjusted R-squared:  0.7364
F-statistic: 90.65 on 11 and 342 DF,  p-value: < 2.2e-16
```

```
> car::vif(model.3)
    crim       zn    indus     chas      nox       rm      dis      rad
1.770412 2.046523 3.061952 1.071848 3.924904 2.179660 3.432532 2.894475
 ptratio    black    lstat
1.786154 1.350901 3.169559
```

Avaliando a habilidade preditora do nosso modelo

```
> fitted.values <- predict(model.3, newdata = test, type = "response")
> # fitted.values
> diferenca <- fitted.values - test$medv
> (rmse <- sqrt(mean(diferenca^2)))
[1] 6.573464
```