MAE0399 – Análise de Dados e Simulação: Introdução ao R para análise exploratória de dados

Professora: Márcia D'Elia Branco Monitor PAE: Rafael O. Silva

01/04/2020

Introdução

Iremos analisar o conjunto de dados Boston do pacote MASS(Modern Applied Statistics with S) e seguiremos os passos do capítulo 3 do livro An Introduction to Statistical Learning with Applications in R. Assim, precisamos carregar a biblioteca

```
library(MASS)
```

- Podemos verificar os nomes contidos no conjunto de dados usando a função *names()*

```
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

- A função *str()* retorna informações detalhadas sobre os objetos no R:

```
str(Boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

Para saber mais sobre o conjunto de dados use o comando *?Boston*.

Regressão linear simples no R

Para ajustar um modelo de regressão no R usamos a função *lm()*. A sintaxe básica dessa função é $lm(y \sim x, data)$, em que y é chamada de variável resposta, x é chamada de variável explicativa e $data$ refere-se aos dados em que as duas variáveis estão.

Exemplo usando os dados de Boston:

- Considere o seguinte modelo $medv_i = \alpha + \beta lstat_i + \epsilon_i$:

```
lm.fit = lm(medv ~ lstat, data = Boston )
#medv valor médio das casas ocupadas pelos proprietários em 1000 dolares.
# menor status da população em %

#A função attach nos permite acessar cada variável do conjunto de dados
attach(Boston)
lm.fit = lm(medv~lstat)
```

- Algumas informações sobre o modelo são fornecidas pelo objeto $lm.fit$,

```
lm.fit
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

- Podemos usar a função *names()* para acessar informações dos objetos dentro do $lm.fit$:

```
names(lm.fit)
```

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

- Para extrair os coeficientes estimados usamos:

```
lm.fit$coefficients
```

```
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

Função *summary()*

A função *summary()* fornece informações mais detalhadas do objeto $lm.fit$,

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

A função *summary()* apresenta um resumo dos resíduos, as estimativas, os erros padrão (*Std.Error*) das estimativas dos coeficientes de regressão ($EP(\widehat{\alpha}) = \sqrt{\frac{QMRes \sum x_i^2}{nS_{xx}}}$ e $EP(\widehat{\beta}) = \sqrt{\frac{QMRes}{S_{xx}}}$) , os valores das estatísticas dos testes $H_0 : \alpha = 0$ versus $H_1 : \alpha \neq 0$

$$t = \frac{\widehat{\alpha} - 0}{EP(\widehat{\alpha})} \sim t(n-2)$$

$H_0 : \beta = 0$ versus $H_1 : \beta \neq 0$

$$t = \frac{\widehat{\beta} - 0}{EP(\widehat{\beta})} \sim t(n-2)$$

e seus respectivos valores-p. Além disso, apresenta o erro padrão dos resíduos, o $R^2 = \frac{SQRes}{SQTot}$, o R^2-ajustado $= 1 - \left(\frac{n-1}{n-(p+1)}\right)(1 - R^2)$ e o valor da estatística $F = \frac{QMReg}{QMRes} \sim F_{(1,n-2)}$.

- A função *anova()* apresenta a tabela de análise de variância:

```
anova(lm.fit)
```

```
## Analysis of Variance Table
##
## Response: medv
##            Df Sum Sq Mean Sq F value    Pr(>F)
## lstat       1  23244 23243.9  601.62 < 2.2e-16 ***
## Residuals 504  19472    38.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observações:

- Estimador : É uma função de uma estatística que fornece uma informação sobre uma característica (parâmetro) da população. Por exemplo, a média $\bar{Y}$ é um estimador para média μ populacional.
- Estimativa : É o valor assumido pelo estimador para uma amostra observada. Por exemplo, se observamos $y_1 = 2$ e $y_2 = 4$, a média $\bar{y} = \frac{2+4}{2} = 3$ é uma estimativa para a média populacional.
- Erro padrão: Sabendo que um estimador é uma função de uma estatística e que, portanto, possui uma distribuição, podemos calcular a expressão do devio padrão do estimador e este é definido como erro padrão. Por exemplo, considere $Y_1, Y_2, \cdots, Y_n$ variáveis aleatórias com distribuição $Y \sim N(\mu, \sigma^2)$. Um estimador para a média populacional μ é $\hat{\mu} = \bar{Y}$ e sua variância é

$$var(\bar{Y}) \quad = \quad var\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) = \frac{1}{n^2}\sum_{i=1}^{n} var(Y_i) = \frac{1}{n^2} n\sigma^2 = \frac{1}{n}\sigma^2.$$

  Sabendo que $E(\bar{Y}) = \mu$ e usando as propriedades da distribuição normal, teremos que $\hat{\mu} = \bar{Y} \sim N(\mu, \frac{1}{n}\sigma^2)$. O erro padrão do estimador é, $EP(\hat{\mu}) = EP(\bar{Y}) = \sqrt{var(\bar{Y})} = \sqrt{\frac{\sigma^2}{n}}$. Agora, se temos $n = 100, \bar{y} = 20$ e $\widehat{\sigma}^2 = s^2 = 36$, a estimativa do erro padrão é $\widehat{EP(\hat{\mu})} = \sqrt{\frac{s^2}{n}} = \frac{6}{10}$.

- Calculando as estimativas dos erros padrão $\widehat{EP(\widehat{\alpha})}$ e $\widehat{EP(\widehat{\beta})}$

```
n <- length(Boston$lstat)
Sxx <- (n-1)*var(Boston$lstat)
Sumx2 <- sum(Boston$lstat^2)
QMRes <- 38.6
```

```
SEalpha <- sqrt(QMRes*Sumx2/(n*Sxx)); SEalpha
```

```
## [1] 0.5623675
```

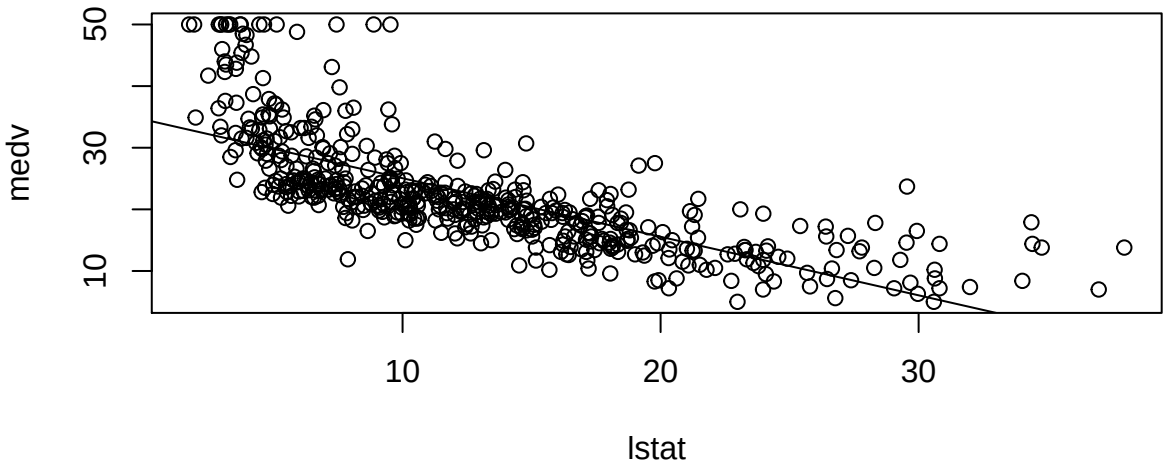
```
SEbeta <- sqrt(QMRes/Sxx); SEbeta
```

```
## [1] 0.03871553
```

Análise Gráfica

Nós podemos fazer o gráfico dos dados com a reta ajustada usando as funções plot() e abline():

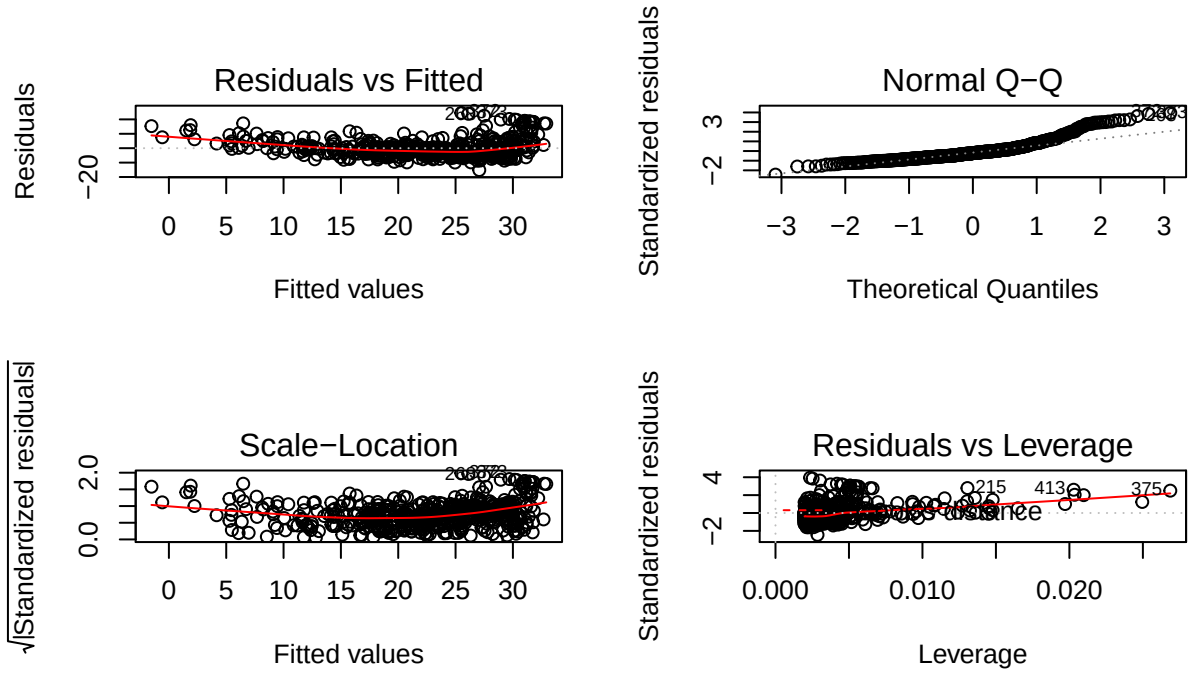
```
plot(lstat, medv ); abline(lm.fit )
```



Note que pelo gráfico existe evidência de não lineraridade da relação entre *lstat* e *medv*. Isso será explorado mais a frente.

Apresentamos agora alguns gráficos para fazer o diagnóstico do modelo ajustado:

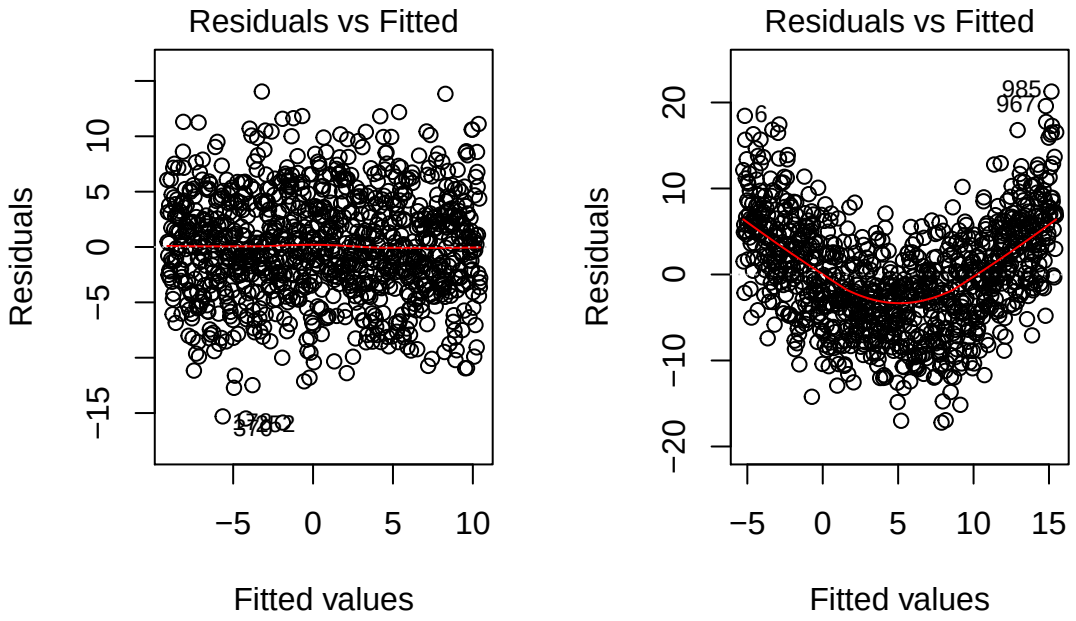
```
par(mfrow = c(2,2)); plot (lm.fit)
```



Os gráficos de diagnóticos dos resíduos são:

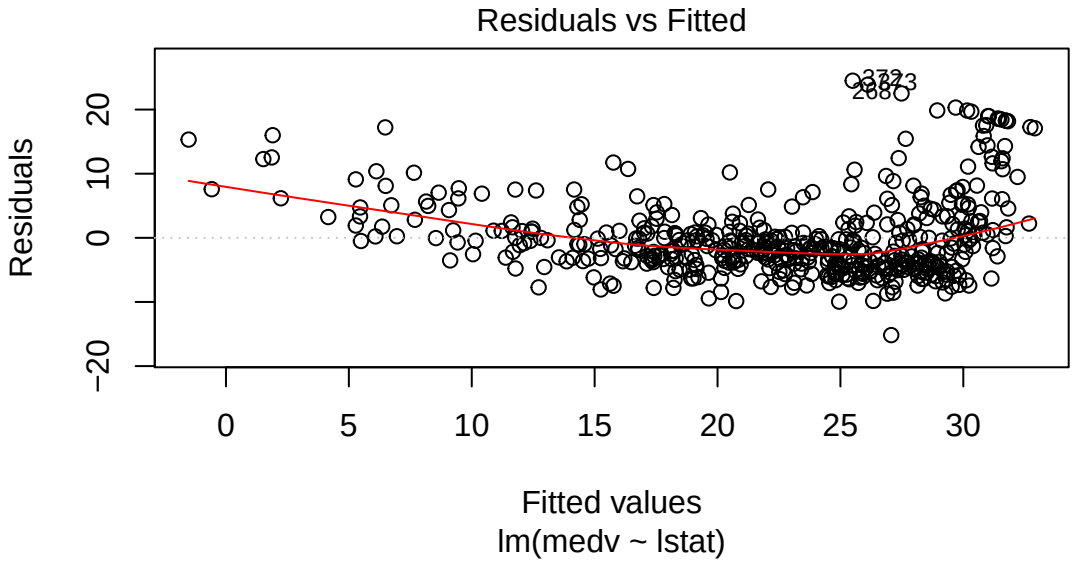
Residuals vs Fitted:

- É usado para checar a suposição de linearidade.
  - Uma linha horizontal, sem padrões distintos, é uma indicação para uma relação linear.
  - A linha vermelha é a curva polinomial suavizada de alta ordem para nos dar uma ideia do padrão de movimento residual.
- Exemplo de um ajuste bom e ruim, respectivamente:



- Residuals vs Fitted dos nossos dados:

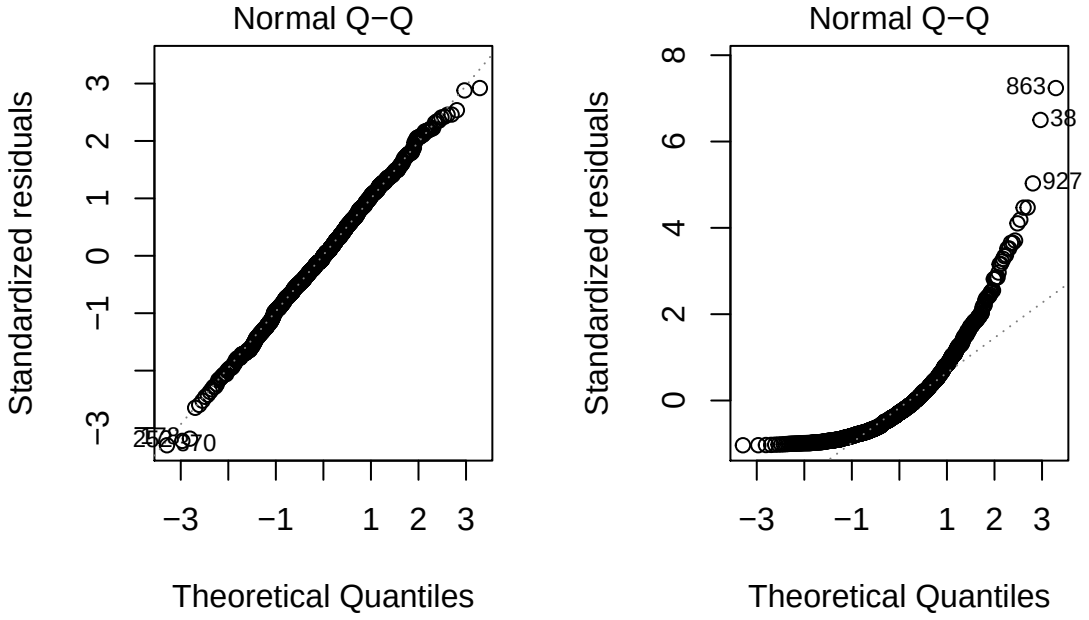
```
plot (lm.fit,1)
```



- No nosso conjunto de dados podemos ver que nossos resíduos têm um padrão curvo. Isso pode significar que podemos obter um modelo melhor se tentarmos um modelo com um grau maior.

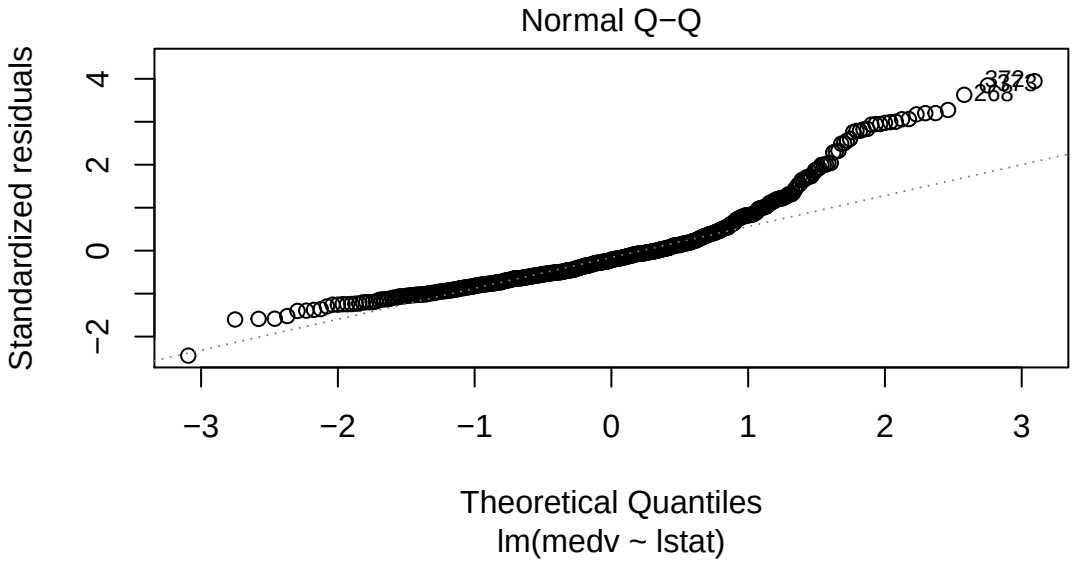
Normal Q-Q:

- É usado para examinar a suposição de normalidade dos resíduos.
  - Pontos bem próximos da reta é um indicativo a favor dessa suposição.
  - Exemplo de um ajuste bom e ruim, respectivamente:



- Normal Q-Q dos nossos dados:

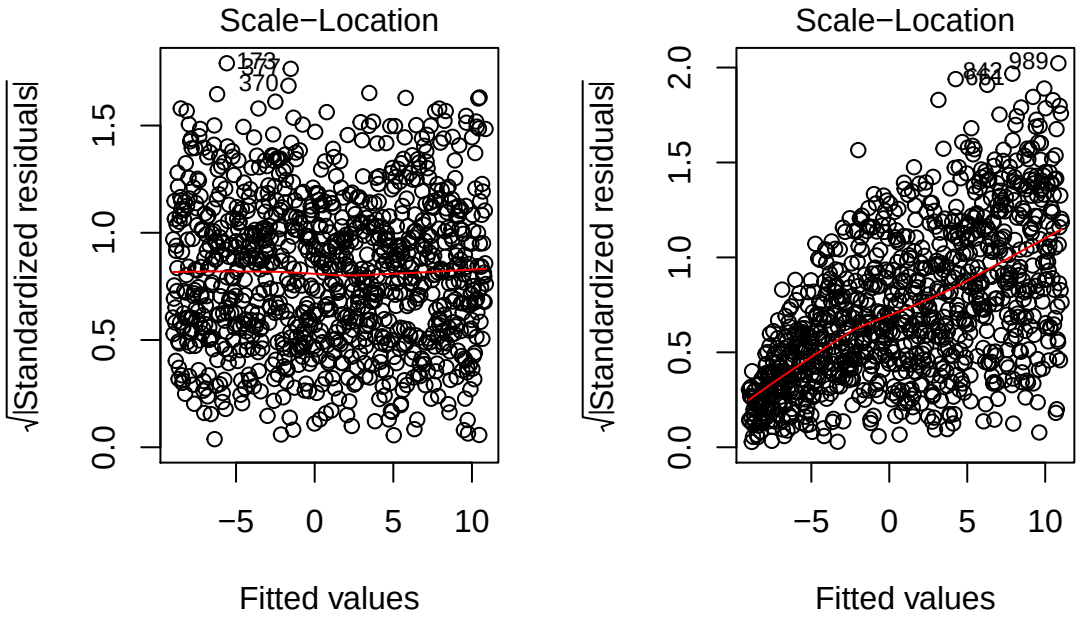
```
plot (lm.fit,2)
```



- No nosso caso vemos um indicativo de fuga da normalidade.

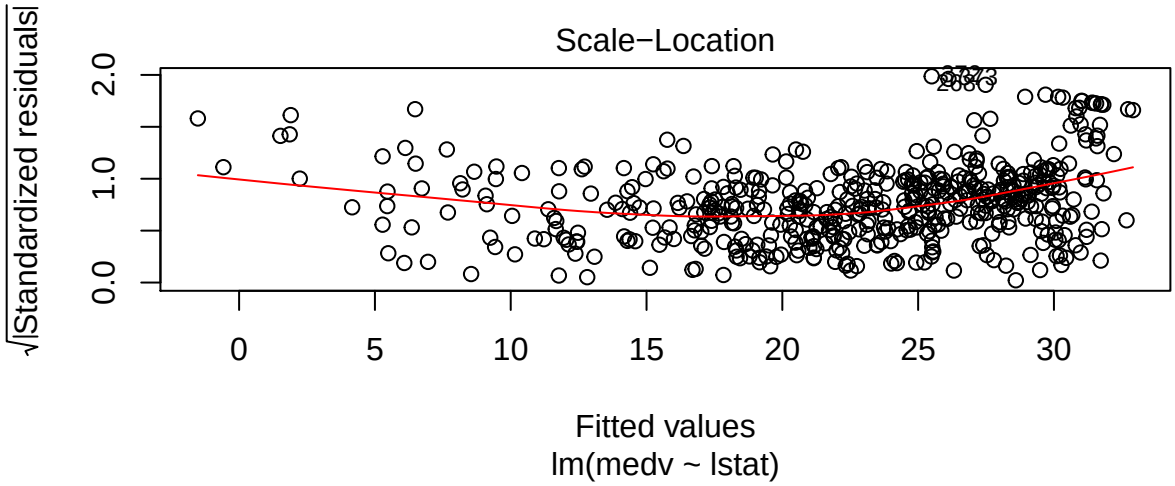
Scale-Location:

- É usado para checar a suposição de homocedasticidade da variância dos resíduos.
  - A linha horizontal com pontos igualmente dispersos é uma boa indicação de homoscedasticidade.
  - Exemplo de um ajuste bom e ruim, respectivamente:



- Normal Q-Q dos nossos dados:

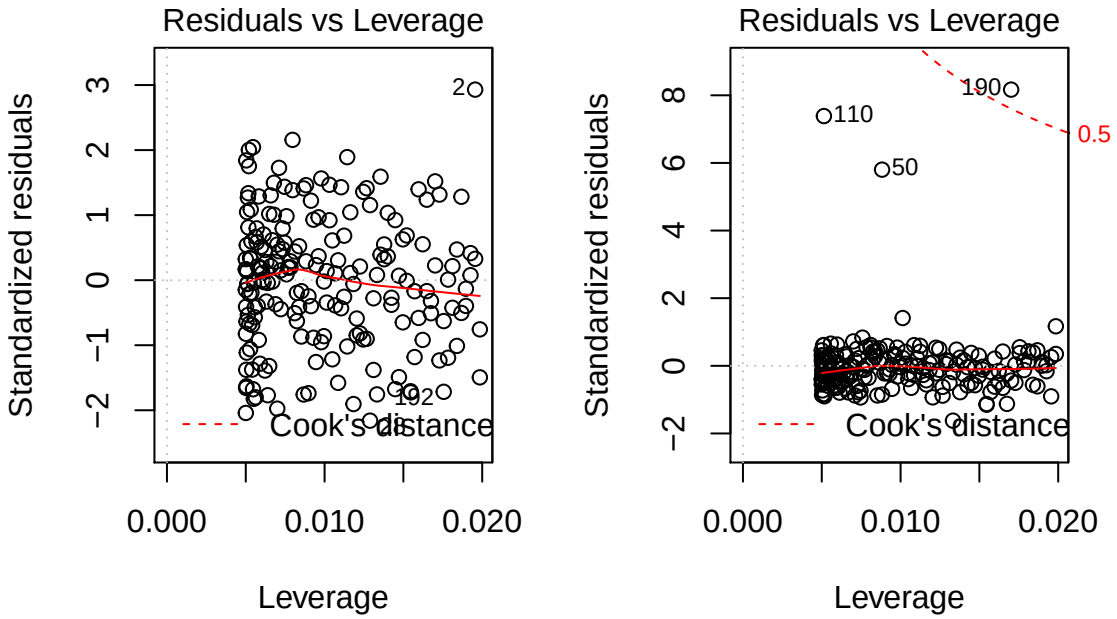
```
plot (lm.fit,3)
```



- No nosso caso vemos que os nossos dados corroboram para a suposição de homocedasticidade.

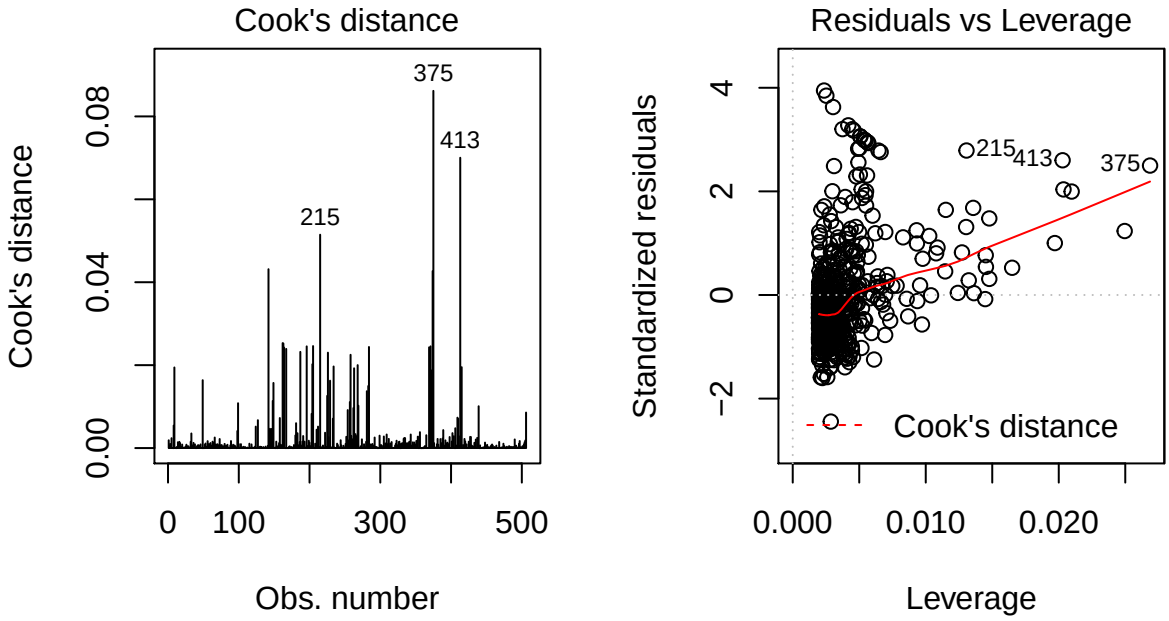
Residuals vs Leverage:

- Residuals vs Leverage: É usado para identificar pontos influentes.
  - Nem todos os outliers são influentes na análise de regressão linear.
  - Usa a distância de Cook.
  - Identificando pontos de influência:



- Residuals vs Leverage dos nossos dados:

```
par(mfrow = c(1,2))
plot (lm.fit,4)
plot (lm.fit,5)
```



- No nosso caso vemos a presença de 3 pontos influentes.

Intervalos de Confiança e de Predição

Das aulas anteriores sabemos que $\widehat{\sigma}^2 = QMRes$,

$$(\widehat{\beta} - \beta)\sqrt{\frac{S_{xx}}{QMRes}} \sim t(n-2)$$

e

$$(\widehat{\alpha} - \alpha)\sqrt{\frac{nS_{xx}}{QMRes\sum x_i^2}} \sim t(n-2)$$

em que $\sum x_i^2 = S_{xx} + n\bar{x}^2$. Os intervalos de confiança para α e β ao nível γ são, respectivamente:

- Intervalo de confiança de α:

$$IC(\alpha, \gamma) = \left(\widehat{\alpha} \pm t_\gamma(n-2)\sqrt{QMRes\left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right]}\right)$$

- Intervalo de confiança de β:

$$IC(\beta, \gamma) = \left(\widehat{\beta} \pm t_\gamma(n-2)\sqrt{\frac{QMRes}{S_{xx}}}\right).$$

Podemos obter intervalos de confiança para as estimativas dos coeficientes usando o comando

```
confint(lm.fit, level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

Tendo o estimador pontual $\widehat{\mu(x_i)} = \widehat{y}_i = \widehat{\alpha} + \widehat{\beta}x_i$ e sabendo que,

$$T = \frac{\widehat{\mu(x_i)} - \mu(x_i)}{\sqrt{QMRes\left[\frac{1}{n} + \frac{(x_i - \bar{x})^2}{S_{xx}}\right]}} \sim t(n-2).$$

Em regressão, calculamos o intervalo de confiança para um valor de Y dado um x, em que x pertence aos dados. Já no intervalo de predição, calculamos o intervalo para um valor de Y dado um x, em que x não pertence aos dados, porém pertence ao intervalo de variação estudado.

- O intervalo com $\gamma\%$ de confiança de $\mu(x_i)$:

$$IC(\mu(x_i), \gamma) = \left(\widehat{y}_i \pm t_\gamma(n-2)\sqrt{QMRes\left[\frac{1}{n} + \frac{(x_i - \bar{x})^2}{S_{xx}}\right]}\right)$$

- O intervalo de predição com $\gamma\%$ de confiança de $\mu(x_i)$:

$$IP(\mu(x_i), \gamma) = \left(\widehat{y}_i \pm t_\gamma(n-2)\sqrt{QMRes\left[1 + \frac{1}{n} + \frac{(x_i - \bar{x})^2}{S_{xx}}\right]}\right).$$

A função predict() pode ser usada para produzir intervalos de confiança e de predição para a predição de *medv* para um dado valor de *lstat*.

```
predict(lm.fit, data.frame( lstat =( c(5, 10, 15))), level = 0.95,
          interval ="confidence")
```

```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

```
predict( lm.fit , data.frame( lstat = ( c(5, 10,15))), level = 0.95,
          interval ="prediction")
```

```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

Regressão Linear Múltipla

Para ajustar uma regressão linear múltipla por mínimos quadrados usamos a função lm(), cujo a sintaxe agora é lm(y∼x1+x2+x3) em que agora as variáveis explicativas são x_1 , x_2 e x_3. Usando a função summary() obtemos o resumo das estimativas do seguinte modelo:

```
lm.fit <- lm( medv ~ lstat + age, data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

- Ajustanto o modelo com todas as 13 variáveis explicativas:

```
lm.fit = lm( medv~., data = Boston )
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

- Retirando uma das covariváveis do modelo:

```
lm.fit1 = lm( medv~. -age, data = Boston )
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6054  -2.7313  -0.5188   1.7601  26.2243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
## crim         -0.108006   0.032832  -3.290 0.001075 **
## zn            0.046334   0.013613   3.404 0.000719 ***
## indus         0.020562   0.061433   0.335 0.737989
## chas          2.689026   0.859598   3.128 0.001863 **
## nox         -17.713540   3.679308  -4.814 1.97e-06 ***
## rm            3.814394   0.408480   9.338  < 2e-16 ***
## dis          -1.478612   0.190611  -7.757 5.03e-14 ***
## rad           0.305786   0.066089   4.627 4.75e-06 ***
## tax          -0.012329   0.003755  -3.283 0.001099 **
## ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
## black         0.009321   0.002678   3.481 0.000544 ***
## lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 4.74 on 493 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7343
## F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

Interações

Para incluir interação entre duas variáveis usamos * . Como exemplo, considere o modelo $medv_i = \alpha + \beta_1 lstat_i + \beta_2 age_i + \beta_3 lstat * age_i + \epsilon_i$, com interação entre $lstat$ e age.

```
summary(lm(medv~lstat*age, data = Boston ) )
```

```
##
## Call:
## lm(formula = medv ~ lstat * age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
summary(lm(medv~lstat + age + lstat:age, data = Boston ) )
```

```
##
## Call:
## lm(formula = medv ~ lstat + age + lstat:age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Transformação Não Linear dos Preditores

Com a função lm() podemos fazer transormações não lineares dos preditores. Por exemplo, dado um preditor X, nós podemos criar o preditor X^2 usando $I(X^2)$. Considere o exemplo: $medv_i = \alpha + \beta_1 lstat_i + \beta_2 lstat_i^2 + \epsilon_i$.

```
lm.fit2 = lm(medv~lstat + I(lstat^2))
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

Note que o valo-p associado ao termo quadrático sugere que esse termo é significativo para o modelo. Podemos usar a função poly() para criar um modelo polinomial dentro da função lm(). Veja o exemplo com um modelo polinomial de grau 5:

```
lm.fit5 = lm(medv~poly(lstat,5))
summary(lm.fit5)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 5))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5433  -3.1039  -0.7052   2.0844  27.1153
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       22.5328     0.2318  97.197  < 2e-16 ***
## poly(lstat, 5)1 -152.4595     5.2148 -29.236  < 2e-16 ***
## poly(lstat, 5)2   64.2272     5.2148  12.316  < 2e-16 ***
## poly(lstat, 5)3  -27.0511     5.2148  -5.187 3.10e-07 ***
## poly(lstat, 5)4   25.4517     5.2148   4.881 1.42e-06 ***
## poly(lstat, 5)5  -19.2524     5.2148  -3.692 0.000247 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.215 on 500 degrees of freedom
## Multiple R-squared:  0.6817, Adjusted R-squared:  0.6785
```

```
## F-statistic: 214.2 on 5 and 500 DF,  p-value: < 2.2e-16
```

Uma outra transformação dos preditores que pode ser utilizada é a transformação logarítmica, como:

```
summary(lm(medv~log(rm), data = Boston ) )
```

```
##
## Call:
## lm(formula = medv ~ log(rm), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.487  -2.875  -0.104   2.837  39.816
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -76.488      5.028  -15.21   <2e-16 ***
## log(rm)       54.055      2.739   19.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.915 on 504 degrees of freedom
## Multiple R-squared:  0.4358, Adjusted R-squared:  0.4347
## F-statistic: 389.3 on 1 and 504 DF,  p-value: < 2.2e-16
```

Preditores Qualitativos

Examinaremos agora o conjunto de dados Carseats, que faz parte da biblioteca ISLR.

```
library(ISLR)
# Conjunto de dados sobre vendas de cadeiras de carro para criança
names(Carseats)
```

```
##  [1] "Sales"       "CompPrice"   "Income"      "Advertising" "Population"
##  [6] "Price"       "ShelveLoc"   "Age"         "Education"   "Urban"
## [11] "US"
```

```
str(Carseats)
```

```
## 'data.frame':    400 obs. of  11 variables:
##  $ Sales      : num  9.5 11.22 10.06 7.4 4.15 ...
##  $ CompPrice  : num  138 111 113 117 141 124 115 136 132 132 ...
##  $ Income     : num  73 48 35 100 64 113 105 81 110 113 ...
##  $ Advertising: num  11 16 10 4 3 13 0 15 0 0 ...
##  $ Population : num  276 260 269 466 340 501 45 425 108 131 ...
##  $ Price      : num  120 83 80 97 128 72 108 120 124 124 ...
##  $ ShelveLoc  : Factor w/ 3 levels "Bad","Good","Medium": 1 2 3 3 1 1 3 2 3 3 ...
##  $ Age        : num  42 65 59 55 38 78 71 67 76 76 ...
##  $ Education  : num  17 10 12 14 13 16 15 10 10 17 ...
##  $ Urban      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 2 2 1 1 ...
##  $ US         : Factor w/ 2 levels "No","Yes": 2 2 2 2 1 2 1 2 1 2 ...
```

Dada uma variável qualitativa como Shelveloc(Bad, Medium e Good), essa variável é um indicador da qualidade do espaço dentro da loja na qual o assento do carro é exibido. O R gera variáveis dummy automaticamente. Abaixo, ajustamos o seguinte modelo $Sales_i = \alpha + \beta_1 Age_i + \beta_2 ShelveLoc\ I_{Good} + \beta_3 ShelveLoc\ I_{Medium} + \epsilon_i$

- I_{Good} é a variável indicadora para "Good", isto é, ela tem valor 1 se a qualidade for "Good" e valor 0

(zero) se a qualidade não for Good;

- I_{Medium} é a variável indicadora para a quallidade "Medium".

```
lm.fit = lm( Sales~ Age + Price + ShelveLoc, data = Carseats )
summary(lm.fit )
```

```
##
## Call:
## lm(formula = Sales ~ Age + Price + ShelveLoc, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9685 -1.1598 -0.0671  1.2015  5.1970
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     15.005120   0.561275  26.734   <2e-16 ***
## Age             -0.049989   0.005418  -9.226   <2e-16 ***
## Price           -0.060209   0.003706 -16.245   <2e-16 ***
## ShelveLocGood    4.936509   0.259707  19.008   <2e-16 ***
## ShelveLocMedium  1.972011   0.213528   9.235   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.741 on 395 degrees of freedom
## Multiple R-squared:  0.6237, Adjusted R-squared:  0.6199
## F-statistic: 163.7 on 4 and 395 DF,  p-value: < 2.2e-16
```