Regressao_linear_notebook

June 8, 2021

1 Regressão Linear Simples

1.0.1 O Pacote `palmerpenguins`

Informações aqui: palmerpenguins

Caso não tenham instale o pacote com a função `install.packages("palmerpenguins")`

Carrega pacotes e define tamanho das figuras

[36]:
```
library(tidyverse)
library(palmerpenguins)
options(repr.plot.width=6, repr.plot.height=4)
```

1.0.2 Verificando o conjunto de dados

[2]:
```
data(penguins)
#penguins
```

[3]:
```
penguins %>%
  distinct(species, island)
```

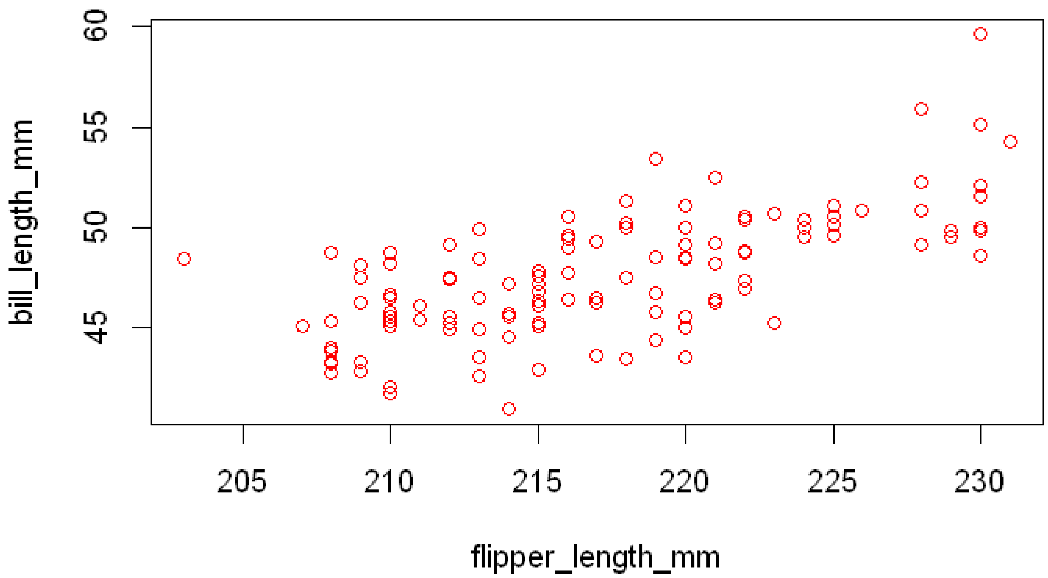
species	island
Adelie	Torgersen
Adelie	Biscoe
Adelie	Dream
Gentoo	Biscoe
Chinstrap	Dream

1.0.3 Selecionando a espécie Gentoo

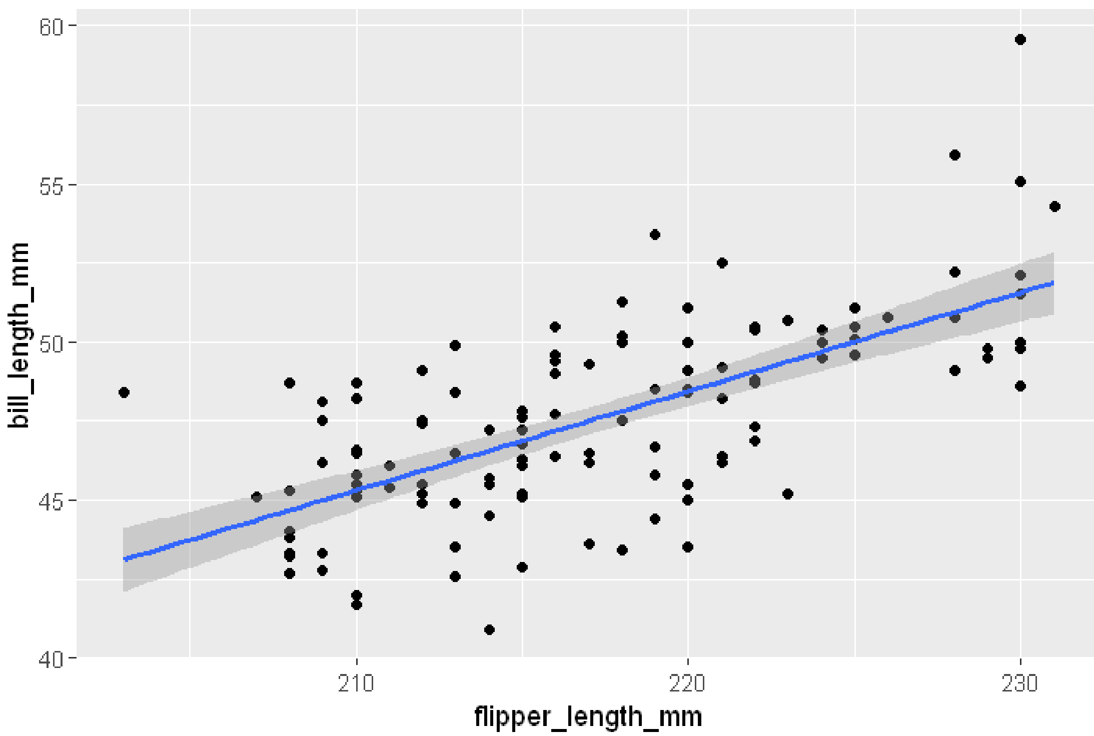
[4]:
```
Gentoo = penguins %>%
  filter(species == 'Gentoo') %>%
  na.omit()
#Gentoo
```

Análise exploratória

[5]:
```
plot(bill_length_mm ~ flipper_length_mm, data = Gentoo, col =2)
```



[6]:
```
plt1 = ggplot(data = Gentoo, mapping = aes(x = flipper_length_mm, y =␣
↪bill_length_mm)) +
    geom_point() +
    geom_smooth(method = 'lm', se = TRUE)
plt1
```



1.0.4 Regressão linear simples

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

Função lm

[7]:
```
m1 = lm(bill_length_mm ~ flipper_length_mm, data = Gentoo)
m1
```

```
Call:
lm(formula = bill_length_mm ~ flipper_length_mm, data = Gentoo)

Coefficients:
      (Intercept)  flipper_length_mm
         -20.4879             0.3133
```

[8]:
```
(beta0 = m1$coefficients[1])
(beta1 = m1$coefficients[2])
```

(Intercept): -20.4878879488712

flipper_length_mm: 0.313282219872178

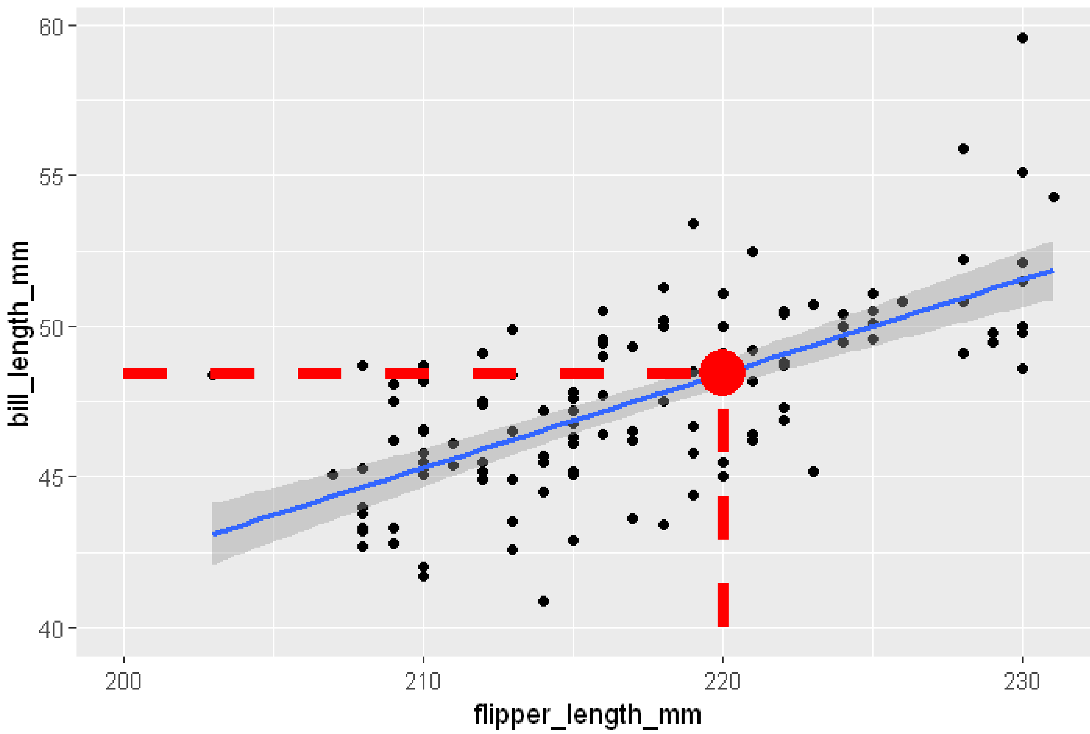
O valor predito

[9]:
```
Xextra = 210
(bico_predito = beta0 + beta1 * Xextra)
```

(Intercept): 45.3013782242862

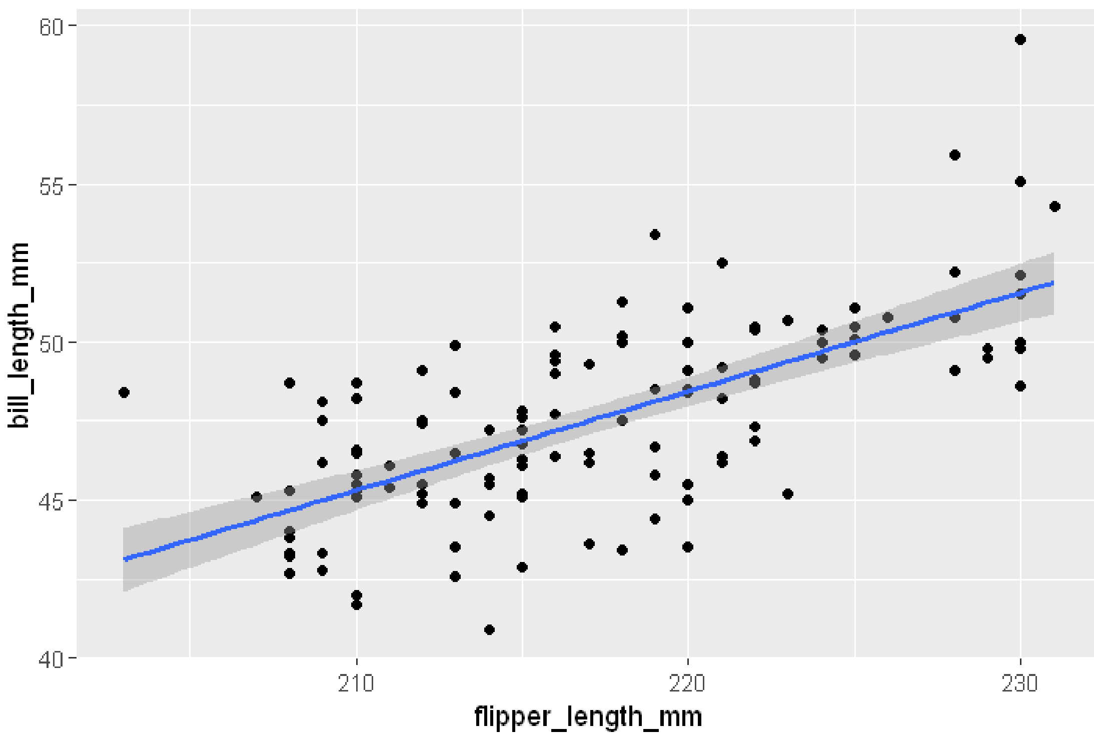
[10]:
```
Xextra = 220
(bico_predito = beta0 + beta1 * Xextra)
plt1 +
    geom_point(aes(x = Xextra, y = bico_predito), colour="red", size = 8) +
    geom_segment(aes(x = 200, xend = Xextra, y = bico_predito, yend =␣
↪bico_predito), linetype = 2, col = 2, size = 2) +
    geom_segment(aes(x = Xextra, xend = Xextra, y = 40, yend = bico_predito),␣
↪linetype = 2, col = 2, size = 2)
```

(Intercept): 48.4342004230079



1.0.5 Extrapolando além dos valores observados

[11]:
```
plt1# +
    #xlim(c(150, 290)) + ylim(c(30, 80)) +
    #geom_abline(intercept = beta0, slope = beta1, color = "red", linetype =␣
↪'dashed')
```



1.0.6 Teste de hipóteses sobre $\beta's$

$H_0 : \beta_1 = 0$

$H_a : \beta_1 \neq 0$

[12]:
```
summary(m1)
```

```
Call:
lm(formula = bill_length_mm ~ flipper_length_mm, data = Gentoo)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.6545 -1.5209 -0.0678  1.3123  8.0330 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -20.4879     7.0845  -2.892  0.00457 ** 
flipper_length_mm   0.3133     0.0326   9.611  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.332 on 117 degrees of freedom
Multiple R-squared:  0.4412,	Adjusted R-squared:  0.4364
```

```
F-statistic: 92.37 on 1 and 117 DF,  p-value: < 2.2e-16
```

Teste t sobre o coeficiente de regressão β_1

$$t_{calculado} = \frac{\hat{\beta}_1 - \beta_1}{s_{\hat{\beta}_1}}$$

[13]:
```
(beta1 = summary(m1)$coefficients[2,1])
(s_beta1 = summary(m1)$coefficients[2,2])
(BETA1 = 0)
(tc = (beta1 - BETA1) / s_beta1)
(gl = df.residual(m1))

pt(tc, df = gl, lower.tail = F) * 2
```

0.313282219872178

0.0325972466896251

0

9.61069573927813

117

1.80316577015333e-16

1.0.7 O que significa rejeitar $H_0 : \beta_1 = 0$? - um pouco de simulação

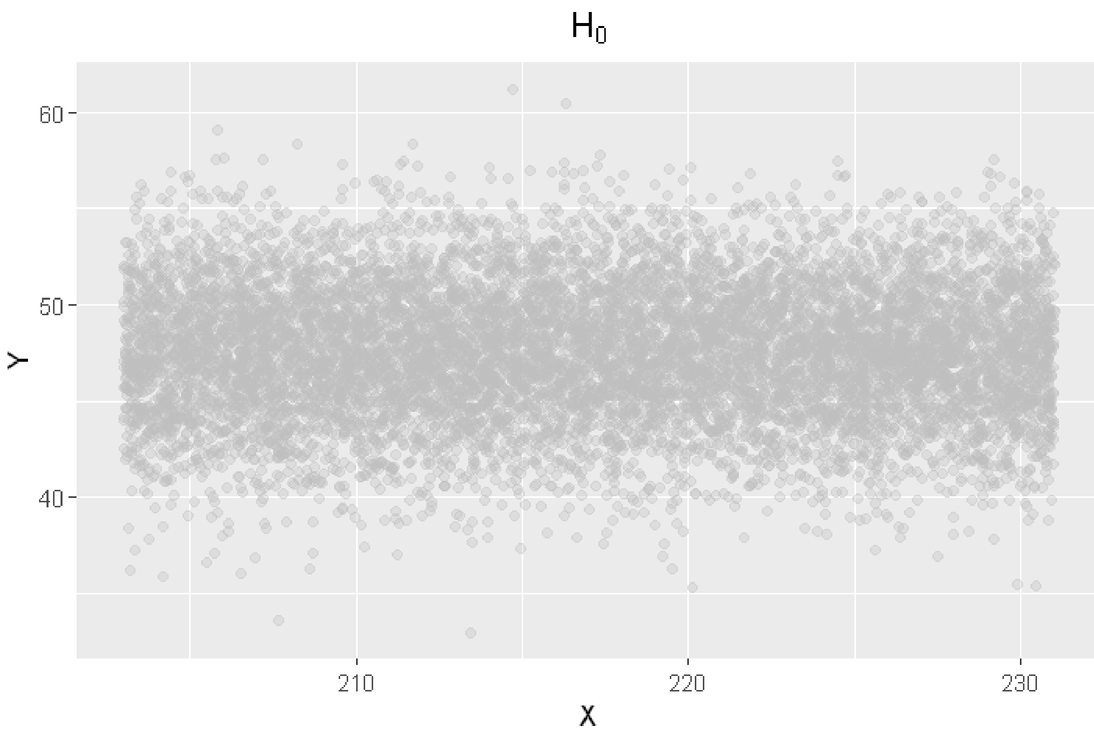
Suponha uma população estatística com as seguintes características:

[14]:
```
N = 10000
s_x = sd(Gentoo$flipper_length_mm)/1.96
BETA0 = mean(Gentoo$bill_length_mm)
BETA1 = 0
X = runif(N, min(Gentoo$flipper_length_mm), max(Gentoo$flipper_length_mm))
Y = rnorm(n = N, mean = BETA0 + BETA1 * X, sd = s_x)
df = data.frame(Y,X)
head(df)
dim(df)
```

Y	X
49.53358	216.8890
47.06076	226.5432
49.44390	225.7330
49.40035	215.8814
50.60243	206.2193
49.42463	207.2544

1. 10000 2. 2

[15]:
```r
plt2 = ggplot(df, mapping = aes(x = X, y = Y)) +
    geom_point(alpha = 0.3, color = 'grey') +
    ggtitle(label = expression(H[0])) +
    theme(plot.title = element_text(hjust = 0.5))

plt2
```



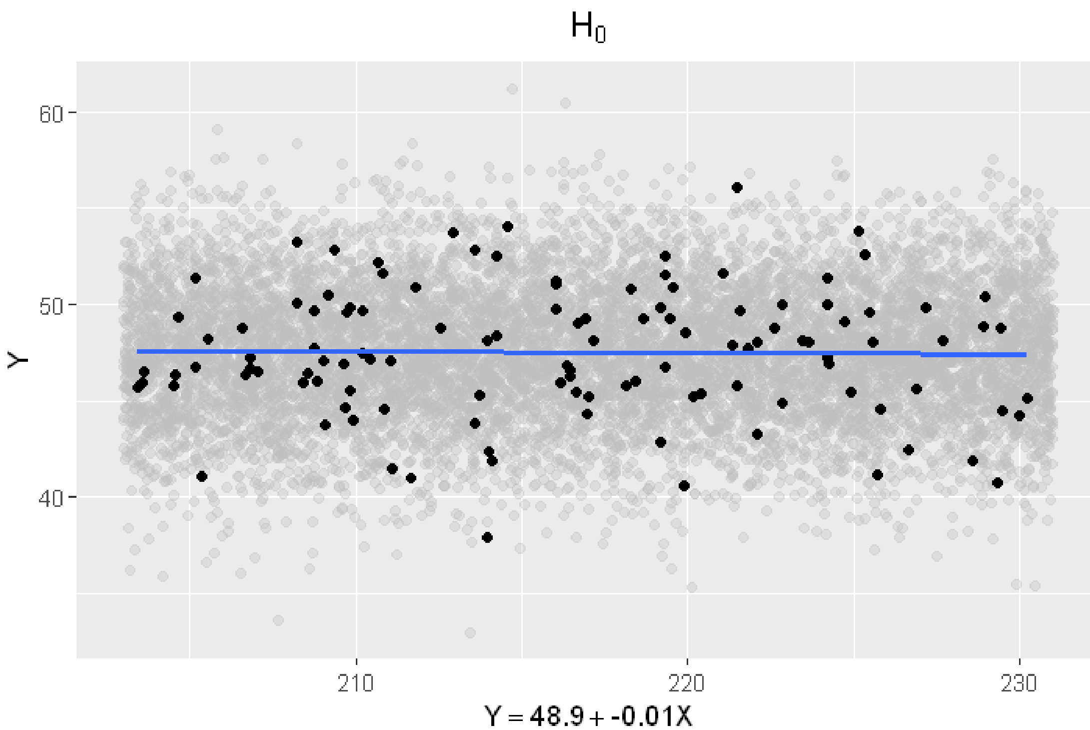
O que esperar se tirarmos uma amostra **aleatória** desta população e fizermos uma regressão linear?

[16]:
```r
amostra = df %>%
    sample_n(5)
amostra
```

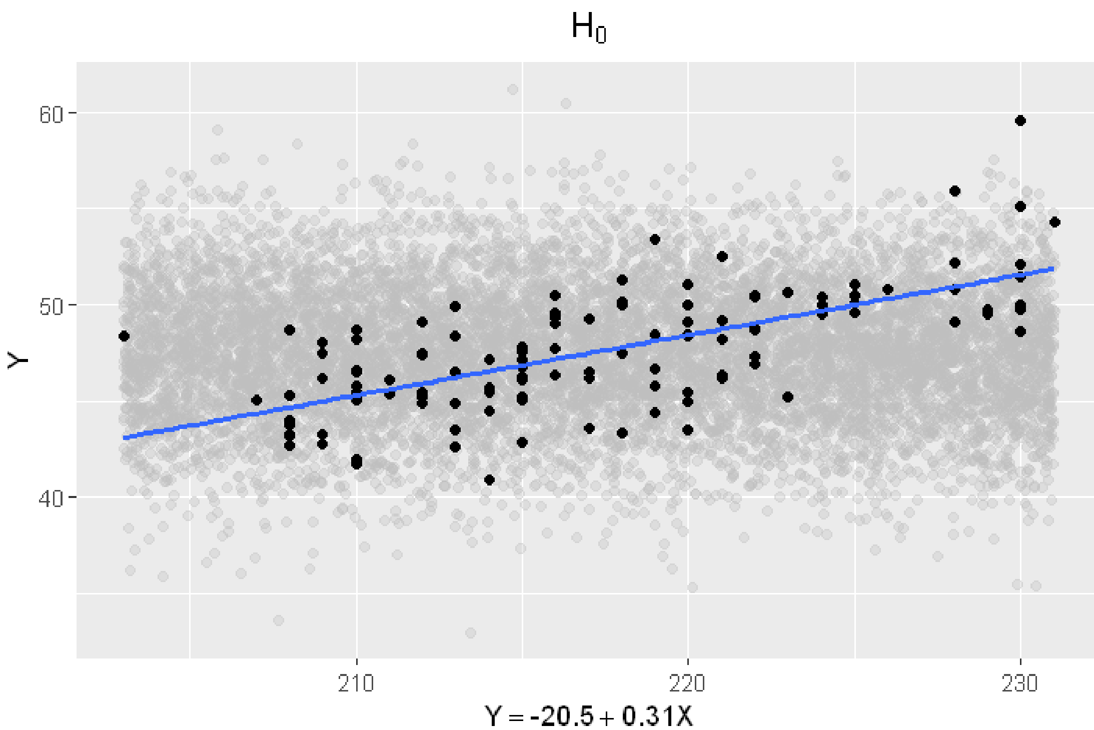
Y	X
46.43981	203.3922
48.19246	215.0912
49.10958	211.1593
44.65308	206.4462
48.79727	210.9419

[17]:
```r
amostra = df %>%
    sample_n(119)

m_sim = lm(Y ~ X, data = amostra)
```

```
b0a = as.numeric(summary(m_sim)$coefficients[1,1])
b1a = summary(m_sim)$coefficients[2,1]
texto = bquote(Y == .(round(b0a,1)) + .(round(b1a,2)) * X)

plt2 +
    geom_point(data = amostra, color = 1) +
    geom_smooth(data = amostra, method = 'lm', se = F) +
    xlab(texto)
```



[18]:
```
b0a = as.numeric(summary(m1)$coefficients[1,1])
b1a = summary(m1)$coefficients[2,1]
texto = bquote(Y == .(round(b0a,1)) + .(round(b1a,2)) * X)

plt2 +
    geom_point(data = Gentoo, mapping = aes(x = flipper_length_mm, y =␣
 ↪bill_length_mm)) +
    geom_smooth(data = Gentoo, mapping = aes(x = flipper_length_mm, y =␣
 ↪bill_length_mm),
                method = 'lm', se = F) +
    xlab(texto)
```



1.0.8 Partição da Soma dos Quadrados e coeficiente de determinação R^2

[19]:
```
anova(m1)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
flipper_length_mm	1	502.2522	502.252212	92.36547	1.803166e-16
Residuals	117	636.2064	5.437662	NA	NA

[20]:
```
summary(m1)
```

```
Call:
lm(formula = bill_length_mm ~ flipper_length_mm, data = Gentoo)

Residuals:
    Min      1Q  Median      3Q     Max
-5.6545 -1.5209 -0.0678  1.3123  8.0330

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -20.4879     7.0845  -2.892  0.00457 **
flipper_length_mm   0.3133     0.0326   9.611  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.332 on 117 degrees of freedom
Multiple R-squared:  0.4412,	Adjusted R-squared:  0.4364 
F-statistic: 92.37 on 1 and 117 DF,  p-value: < 2.2e-16
```

[21]:

```
9.611^2
```

92.371321

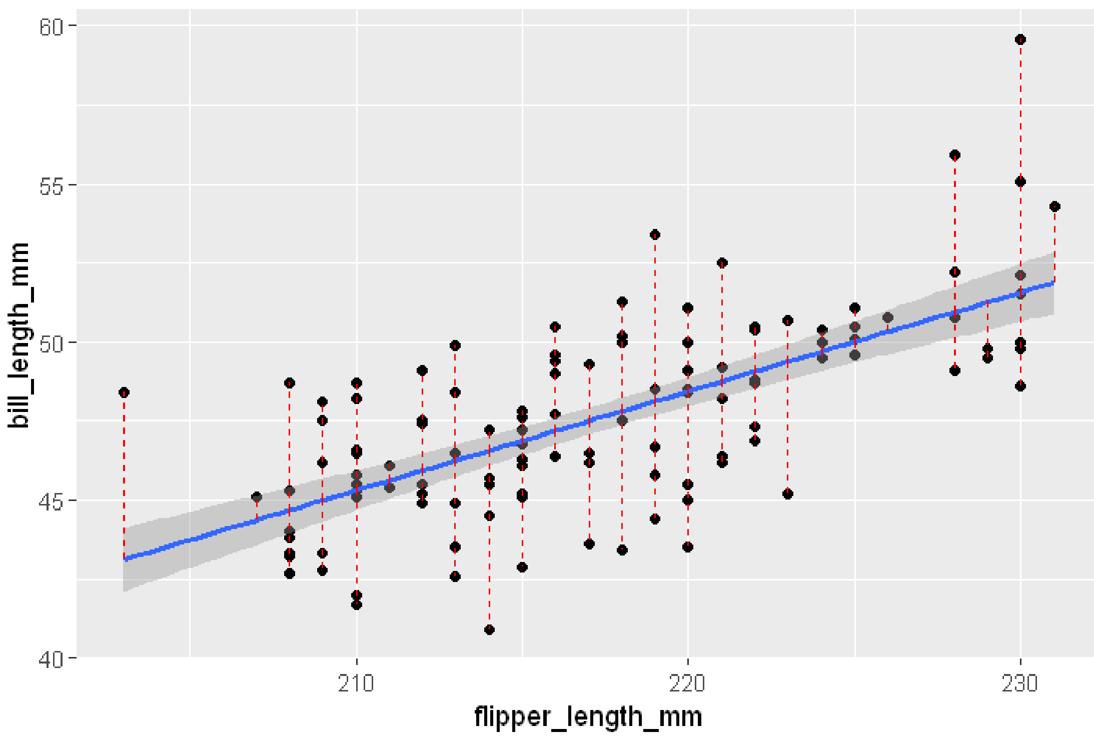
1.0.9 Pressupostos da regressão linear simples

1. Amostras independentes
2. O modelo linear descreve adequadamente a relação entre Y e X
3. Distribuição normal dos resíduos
4. Variância residual constante

[22]:

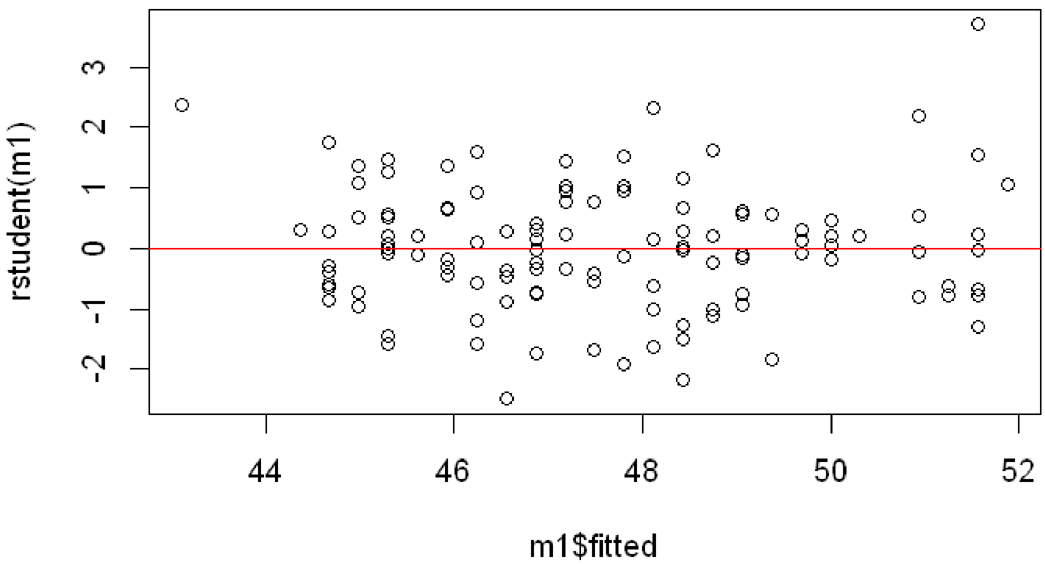
```
summary(m1)
plt1 +
    geom_segment(aes(x = flipper_length_mm,
                     xend = flipper_length_mm,
                     y = predict(m1),
                     yend = bill_length_mm), color = 2, linetype = 2)
```

```
Call:
lm(formula = bill_length_mm ~ flipper_length_mm, data = Gentoo)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.6545 -1.5209 -0.0678  1.3123  8.0330 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -20.4879     7.0845  -2.892  0.00457 ** 
flipper_length_mm   0.3133     0.0326   9.611  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.332 on 117 degrees of freedom
Multiple R-squared:  0.4412,	Adjusted R-squared:  0.4364 
F-statistic: 92.37 on 1 and 117 DF,  p-value: < 2.2e-16
```



Linearidade e Variância σ^2 constante

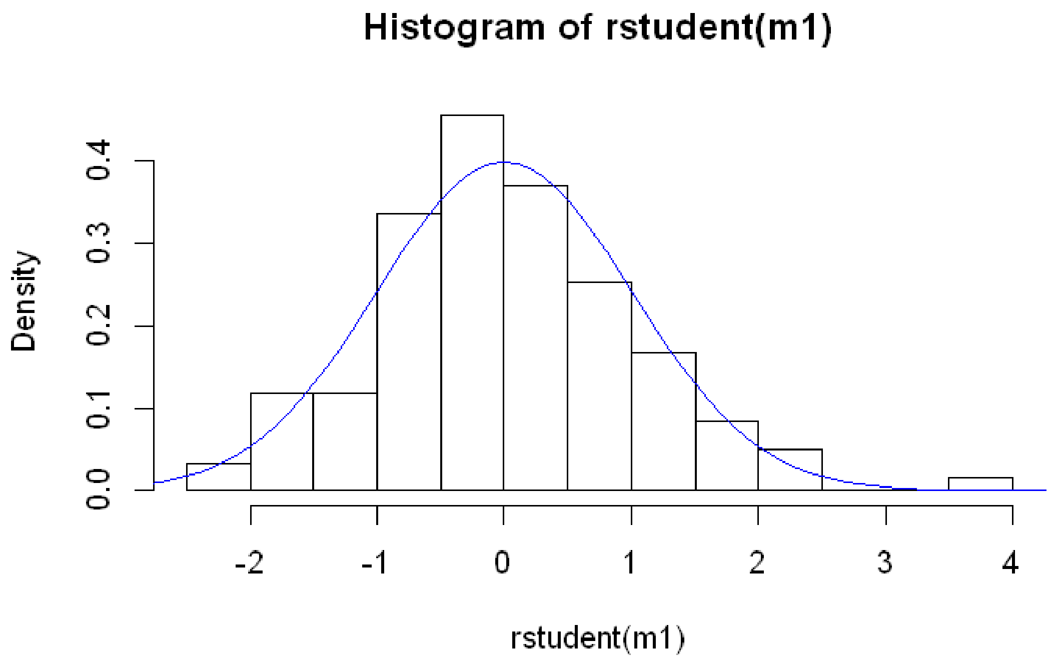
[23]:
```
plot(rstudent(m1) ~ m1$fitted)
abline(h = 0, col = 2)
```



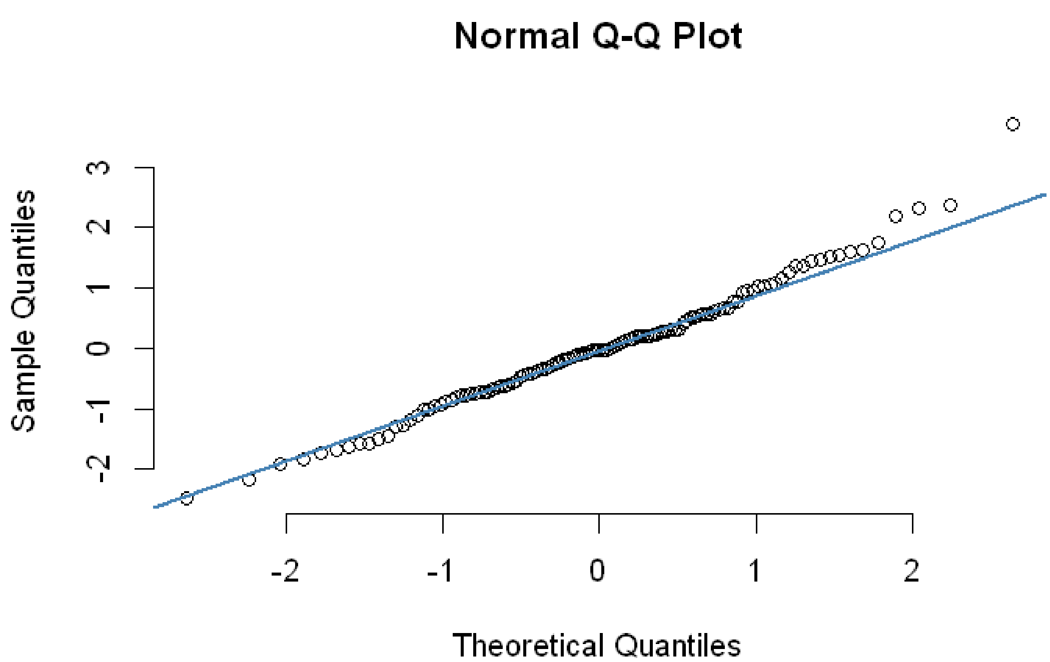
Normalidade dos resíduos $\epsilon \sim N(0, \sigma^2)$

[24]:
```
hist(rstudent(m1), freq = F, breaks = 15)
curve(dnorm(x),-5,5,add=T,col="blue")
shapiro.test(rstudent(m1))
```

```
        Shapiro-Wilk normality test

data:  rstudent(m1)
W = 0.98597, p-value = 0.2556
```

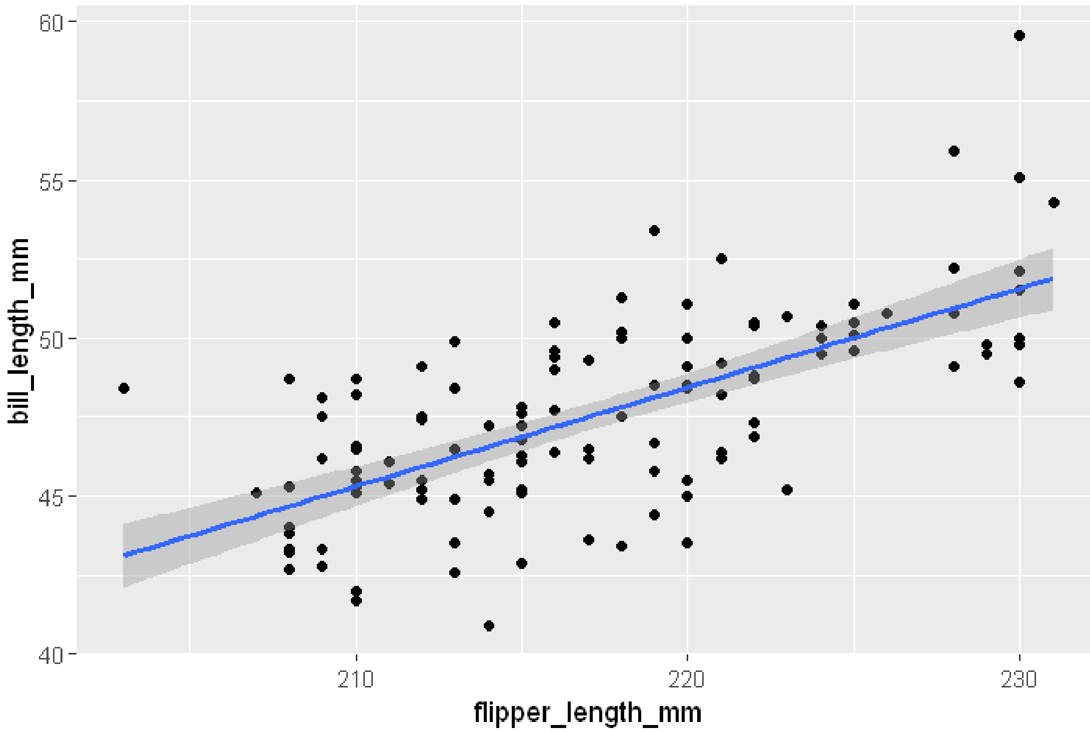


[25]:
```
qqnorm(rstudent(m1), pch = 1, frame = FALSE)
qqline(rstudent(m1), col = "steelblue", lwd = 2)
```



1.0.10 Resumo

[26]:
```
plt1
```

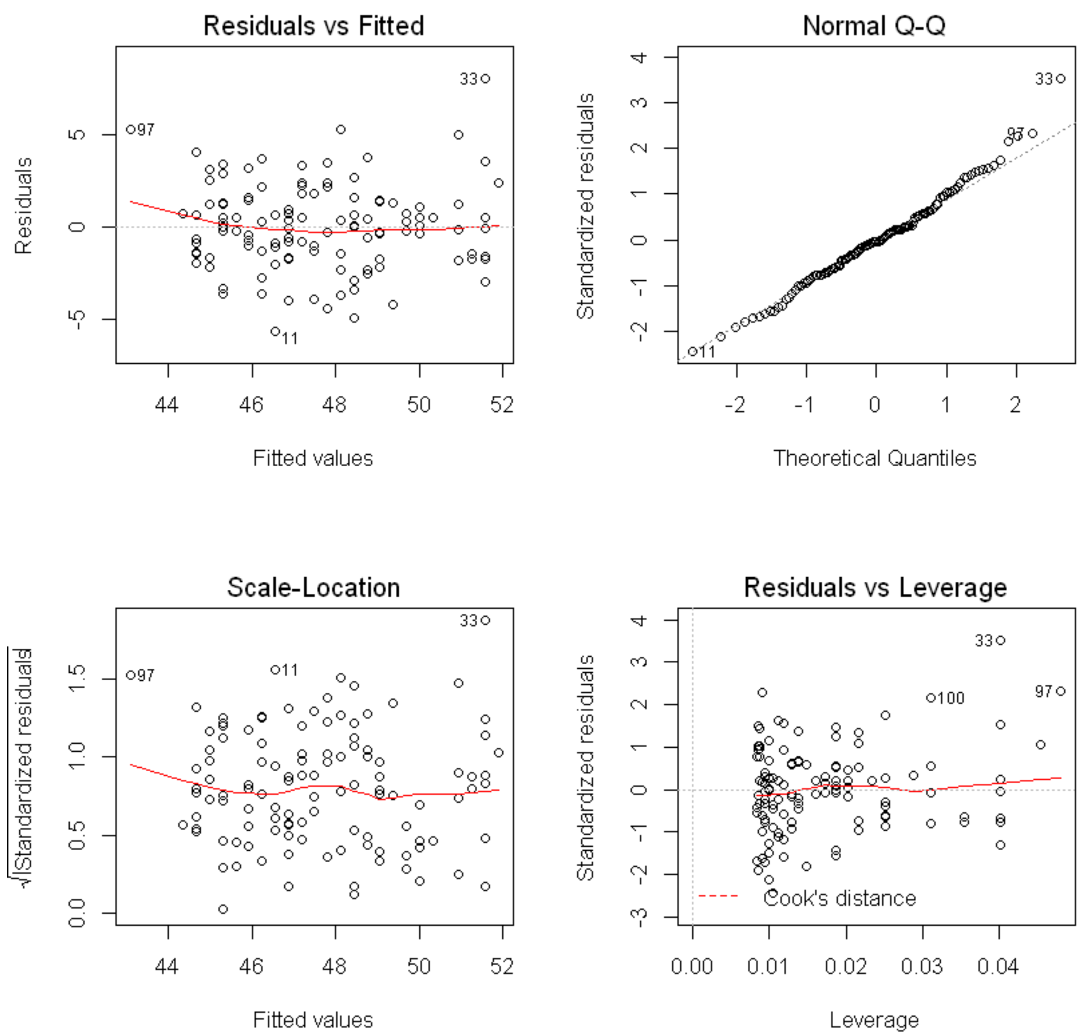


[27]:
```
m1 = lm(bill_length_mm ~ flipper_length_mm, data = Gentoo)
summary(m1)
```

```
Call:
lm(formula = bill_length_mm ~ flipper_length_mm, data = Gentoo)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.6545 -1.5209 -0.0678  1.3123  8.0330 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -20.4879     7.0845  -2.892  0.00457 ** 
flipper_length_mm   0.3133     0.0326   9.611  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.332 on 117 degrees of freedom
Multiple R-squared:  0.4412,	Adjusted R-squared:  0.4364
F-statistic: 92.37 on 1 and 117 DF,  p-value: < 2.2e-16
```

[28]:
```
options(repr.plot.width=7, repr.plot.height=7)
par(mfrow = c(2,2))
plot(m1)
```



1.1 Regressão Múltipla

Se temos mais de uma variável resposta X, o modelo torna-se uma **regressão linear múltipla**. Os coeficientes β_0,, β_0, $\cdots$, β_p, associados a cada variável X são conhecidos como coeficientes parciais de regressão. Assim como na regressão simples, β_0, se refere ao valor que Y assume se todas as demais variáveis $X's = 0$. Cada coeficiente β_i, mede a magnitude da mudança em Y para variação de uma unidade em X_i quando $TODAS$ as demais variáveis X permanecem constantes.

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon_i$$

Vamos utilizar o mesmo conjunto de dados `attitude` para ajustar um modelo de regressão múltipla.

[29]:
```
data(attitude)
attitude %>% head()
```

rating	complaints	privileges	learning	raises	critical	advance
43	51	30	39	61	92	45
63	64	51	54	63	73	47
71	70	68	69	76	86	48
61	63	45	47	54	84	35
81	78	56	66	71	83	47
43	55	49	44	54	49	34

Assim como na regressão simples, o primeiro passo envolve a análise gráfica entre a variável resposta e todas as preditoras. O comando pairs pode ser utilizado para vermos rapidamente o padrão de correlação entre as variáveis.

[30]:
```
#pairs(attitude)
```

[31]:
```
rm = lm(rating ~  complaints + privileges + learning + raises + critical +
 ↪advance, data  = attitude)
#rm
summary(rm)
```

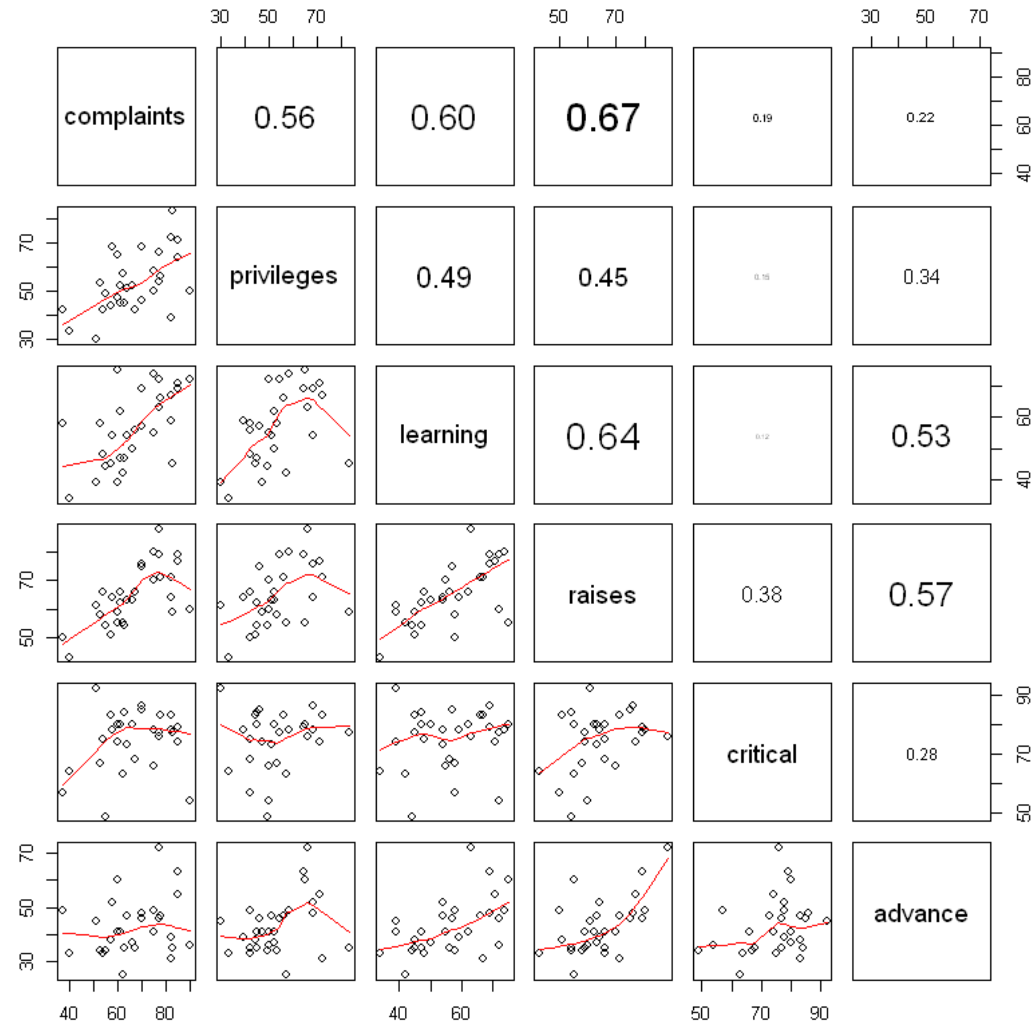
```
Call:
lm(formula = rating ~ complaints + privileges + learning + raises +
    critical + advance, data = attitude)

Residuals:
     Min       1Q   Median       3Q      Max
-10.9418  -4.3555   0.3158   5.5425  11.5990

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.78708   11.58926   0.931 0.361634
complaints   0.61319    0.16098   3.809 0.000903 ***
privileges  -0.07305    0.13572  -0.538 0.595594
learning     0.32033    0.16852   1.901 0.069925 .
raises       0.08173    0.22148   0.369 0.715480
critical     0.03838    0.14700   0.261 0.796334
advance     -0.21706    0.17821  -1.218 0.235577
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.068 on 23 degrees of freedom
Multiple R-squared:  0.7326,	Adjusted R-squared:  0.6628
F-statistic:  10.5 on 6 and 23 DF,  p-value: 1.24e-05
```

Pressupostos - o problema da colinearidade Os pressupostos da regressão múltipla são iguais aos da regressão simples com um adicional: **não deve haver colinearidade**. Isto significa que as variáveis preditoras X não podem ser correlacionadas umas às outras.

Uma forma de verificar este pressuposto passa por um gráfico de dispersão par-a-par entre as variáveis preditoras.

[32]:
```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}



pairs(attitude[,-1], lower.panel = panel.smooth, upper.panel = panel.cor)
```



Vejam que algumas variáveis têm correlações moderadas entre si. Por exemplo, a correlação entre complaints *vs raises é* $r = 0.67$ *e entre learning* vs* raises, $r = 0.64$. Isto pode ser um indício de colinearidade.

Uma forma de medir o grau de colinearidade de uma variável com as demais é pelo cálculo do **Fator de Inflação de Variância** (**VIF** - Variance Inflation Factor). O VIF mede o quanto a variâncias dos estimadores de regressão aumenta devido a multicolinearidade.

Seja X_i a variável preditora i, o VIF associado a esta variável tem relação ao R_i^2 quando esta é utilizada como resposta em um modelo linear com todas as demais variáveis X. Neste caso, VIF_i é calculado por:

$$VIF_i = \frac{1}{(1 - R_i^2}$$

Quanto maior R_i^2, maior o grau de multicolinearidade devido à X_i.

Vamos calcular o VIF associado a variável complaints.

[33]:
```
mcomp = summary(lm(complaints ~ learning + privileges + raises + critical +
 ↪advance, data  = attitude))
#mcomp
#mcomp$r.squared
Vif.comp = 1/ (1 - mcomp$r.squared)

Vif.comp
```

2.66706030825785

O valor de VIF para a variável complaints é 2,667. Veja portanto que o cálculo do VIF para uma determinada variável X_i é obtido encontrando o coeficiente de determinação R_i^2 quando esta variável é utilizada como resposta em um modelo com TODAS as demais preditoras.

Na presença de multicolinearidade, os estimadores da regressão tornam-se **instáveis**, i.e. a retirada de uma variável pode alterar os coeficientes das demais.

Vamos ver o que acontece se retirarmos `complaints` de nosso modelo.

[34]:
```
rm = lm(rating ~ privileges + critical + advance, data  = attitude)

summary(rm)
```

```
Call:
lm(formula = rating ~ privileges + critical + advance, data = attitude)

Residuals:
     Min       1Q   Median       3Q      Max 
-21.0229  -5.0810  -0.6476   5.0387  22.7583 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) 34.25069   17.97030   1.906   0.0678 .
privileges   0.41490    0.18716   2.217   0.0356 *
critical     0.12260    0.22665   0.541   0.5932  
advance     -0.01931    0.22955  -0.084   0.9336  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.56 on 26 degrees of freedom
Multiple R-squared:  0.1908,	Adjusted R-squared:  0.0974 
F-statistic: 2.043 on 3 and 26 DF,  p-value: 0.1324
```

Note que outras variáveis passaram a ser significativas como learning (p = 0,0172) e riases (0,0373). Estas estavam entre as variáveis com maior correlação entre si e com complaints.

Todos os VIF's podem ser calculados por uma única operação, calculando a inversa da matriz de correlação R^{-1}

[35]:
```
attitude %>%
    select(-rating) %>%
    cor() %>%
    solve() %>%
    diag() %>%
    round(2)
```

complaints 2.67 **privileges** 1.6 **learning** 2.27 **raises** 3.08 **critical** 1.23 **advance** 1.95