Modelo linear

(linguagem R)

2025

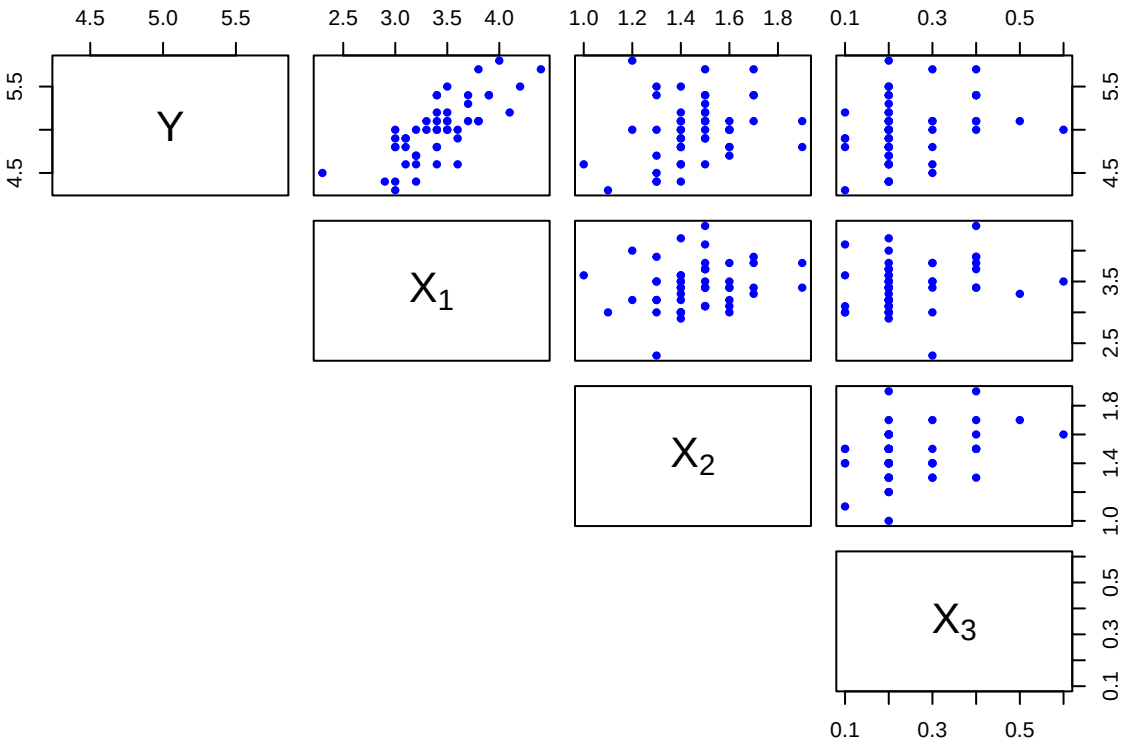
É apresentado um exemplo de ajuste e algumas ferramentas de diagnóstico aplicadas a um modelo linear normal. O conjunto de dados `iris`, espécie *Setosa* (50 primeiras linhas), é usado. A variável resposta é `Sepal.Length`, ao passo que as covariáveis são `Sepal.Width`, `Petal.Length` e `Petal.Width`, denotadas por `Y`, `X1`, `X2` e `X3`, respectivamente. Todas as variáveis são medidas em cm.

```
## Conjunto de dados iris
dados <- iris[1:50, 1:4]
colnames(dados) <- c("Y", "X1", "X2", "X3")
```

```
summary(dados)
```

```
##        Y               X1              X2              X3
##  Min.   :4.300   Min.   :2.300   Min.   :1.000   Min.   :0.100
##  1st Qu.:4.800   1st Qu.:3.200   1st Qu.:1.400   1st Qu.:0.200
##  Median :5.000   Median :3.400   Median :1.500   Median :0.200
##  Mean   :5.006   Mean   :3.428   Mean   :1.462   Mean   :0.246
##  3rd Qu.:5.200   3rd Qu.:3.675   3rd Qu.:1.575   3rd Qu.:0.300
##  Max.   :5.800   Max.   :4.400   Max.   :1.900   Max.   :0.600
```

```
pairs(dados, labels = c("Y", expression(X[1]), expression(X[2]),
   expression(X[3])), pch = 20, lower.panel = NULL, col = "blue")
```



Nota 1. Comente os gráficos acima.

Um modelo linear é ajustado com a função `lm`.

```
m1 <- lm(Y ~ X1 + X2 + X3, data = dados)
names(m1)
```

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

Nota 2. Explique cada um dos componentes do objeto `m1` acima.

Nota 3. Refaça o ajuste utilizando a função `glm`.

Resultados do ajuste são obtidos com a função `summary`.

```
summary(m1)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3, data = dados)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40662 -0.17721  0.01222  0.13388  0.49693
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.35189    0.39287   5.986 3.03e-07 ***
## X1           0.65483    0.09245   7.083 6.83e-09 ***
## X2           0.23756    0.20802   1.142    0.259
## X3           0.25213    0.34686   0.727    0.471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2371 on 46 degrees of freedom
## Multiple R-squared:  0.5751, Adjusted R-squared:  0.5474
## F-statistic: 20.76 on 3 and 46 DF,  p-value: 1.192e-08
```

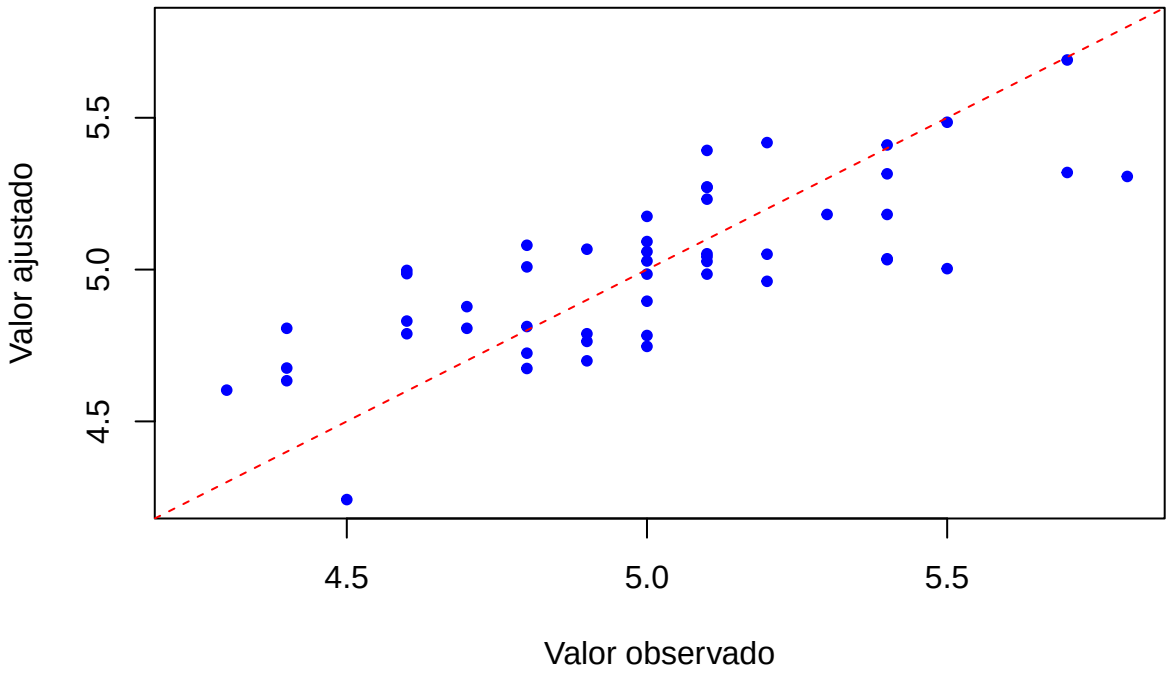
Nota 4. Comente os resultados acima.

Nota 5. No modelo nulo a média da variável resposta é constante. Pode ser ajustado com o comando `m0 <- lm(Y ~ 1, data = dados)`. Com os resultados do comando `anova(m0, m1)`, construa a tabela ANOVA do ajuste do modelo `m1`.

Nota 6. Apresente intervalos de confiança de 95% para os coeficientes de `X1`, `X2` e `X3`.

Os valores observados e ajustados pelo modelo são apresentados abaixo.

```
rm1 <- range(dados$Y, m1$fitted.values)
plot(dados$Y, m1$fitted.values, pch = 20, col = "blue", xlab = "Valor observado",
ylab = "Valor ajustado", xlim = rm1, ylim = rm1)
abline(0, 1, lty = 2, col = "red")
```



Nota 7. Comente o resultado do ajuste com base no gráfico acima.

A função `lm.influence` fornece resultados sobre diagnóstico e influência.

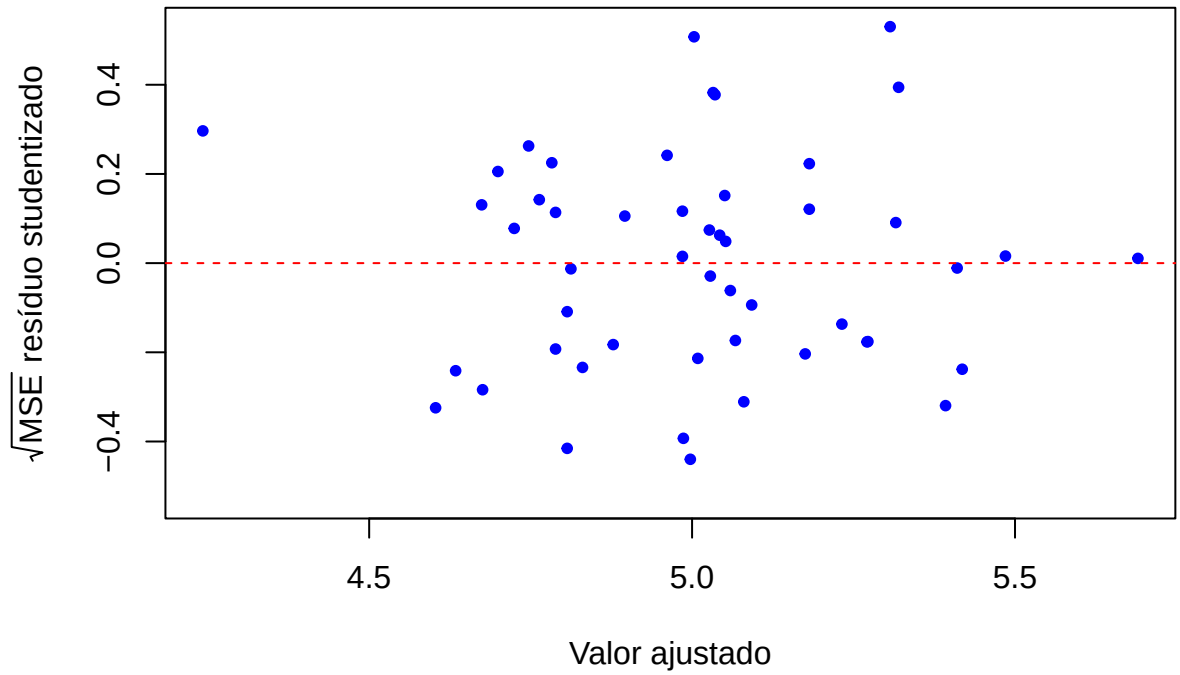
```
infm1 <- lm.influence(m1)
names(infm1)
```

```
## [1] "hat"          "coefficients" "sigma"        "wt.res"
```

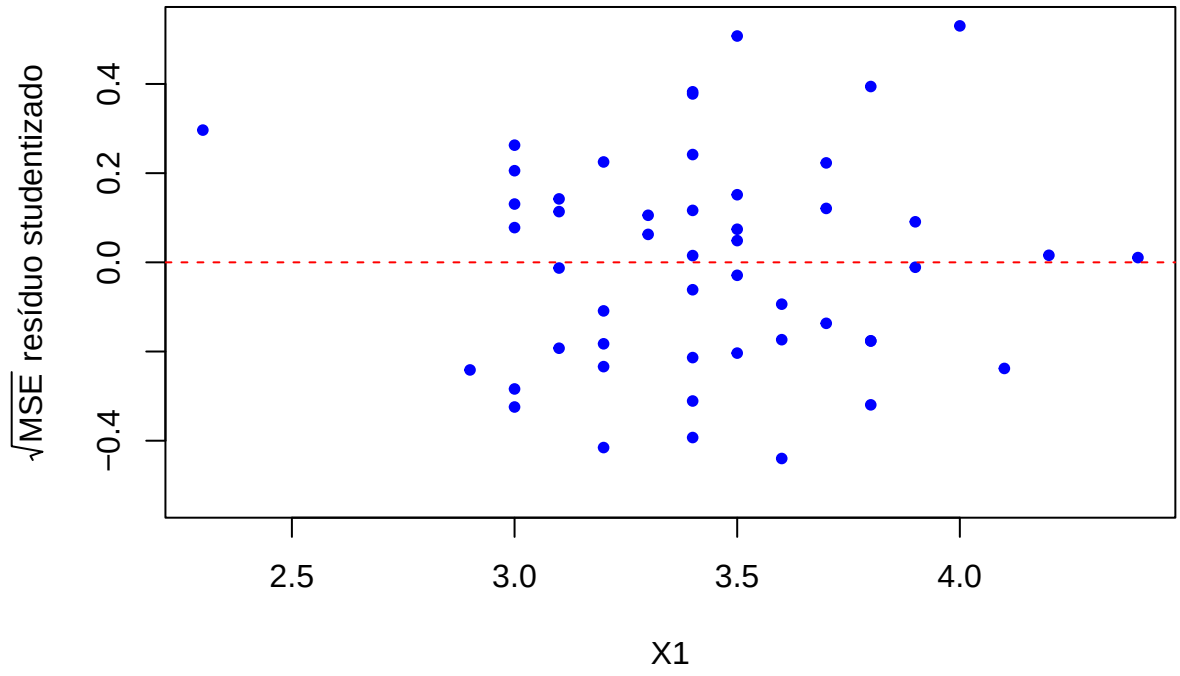
Nota 8. Explique cada um dos componentes do objeto `infm1` acima.

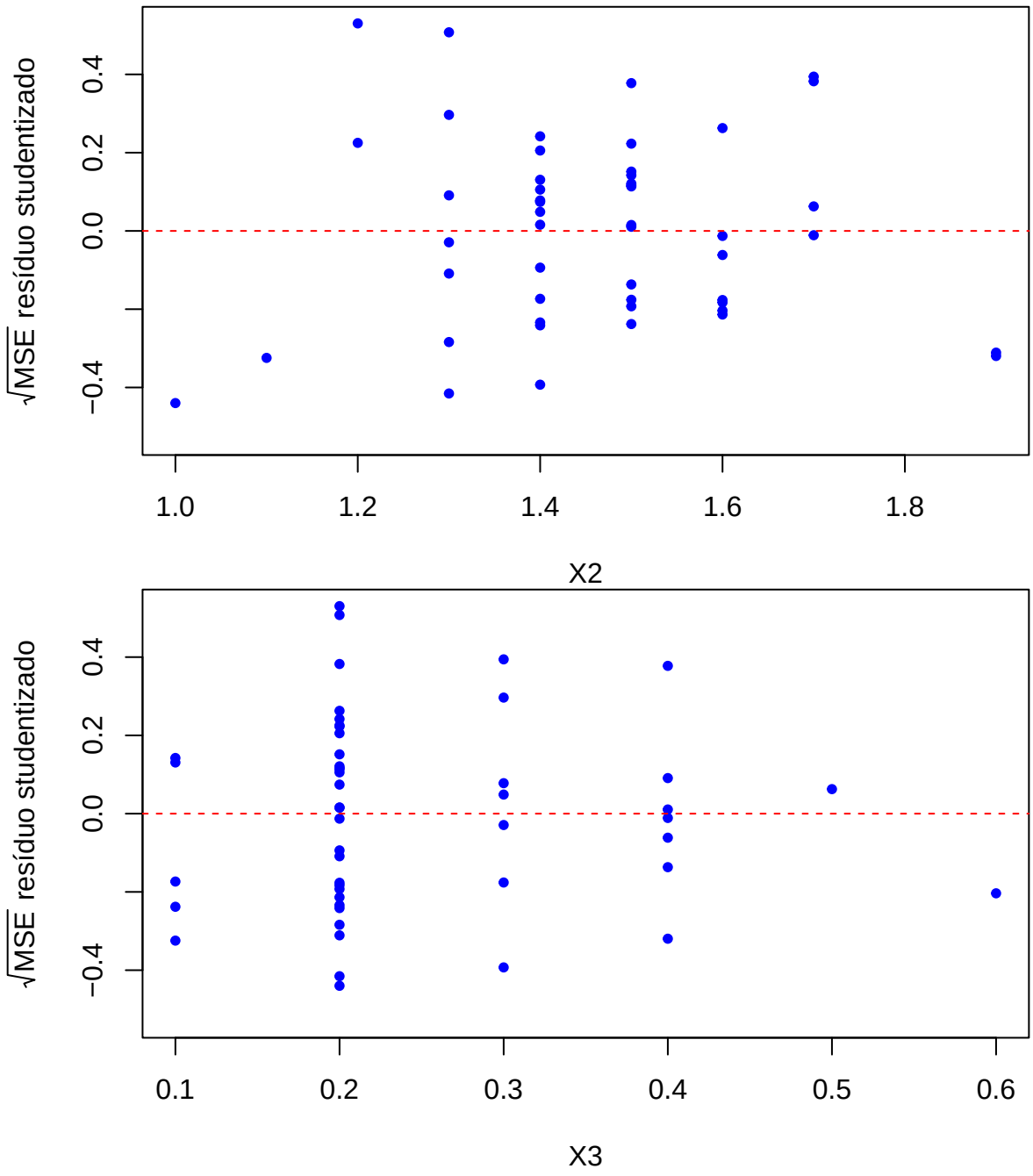
O resíduo studentizado é dado por $r_i = e_i/\sqrt{\text{MSE}(1-h_{ii})}$, de modo que $\text{var}(\sqrt{\text{MSE}}\, r_i) = \sigma^2$, $i = 1, \ldots, n$. Os gráficos de $\sqrt{\text{MSE}}\, r_i$ (que têm variância constante) *versus* valores preditos e também covariáveis são mostrados em seguida.

```
mri <- m1$resid / sqrt(1 - infm1$hat)
maxe <- max(abs(mri))
plot(m1$fitted.values, mri, pch = 20, xlab = "Valor ajustado",
     ylab = expression(paste(sqrt(MSE), " resíduo studentizado")),
     ylim = c(-maxe, maxe), col = "blue")
abline(h = 0, lty = 2, col = "red")
```




```
p <- length(m1$coefficients)
for (j in 2:p) {
plot(dados[, j], mri, pch = 20, xlab = names(dados)[j], ylim = c(-maxe, maxe),
     ylab = expression(paste(sqrt(MSE), " resíduo studentizado")), col = "blue")
abline(h = 0, lty = 2, col = "red")
}
```

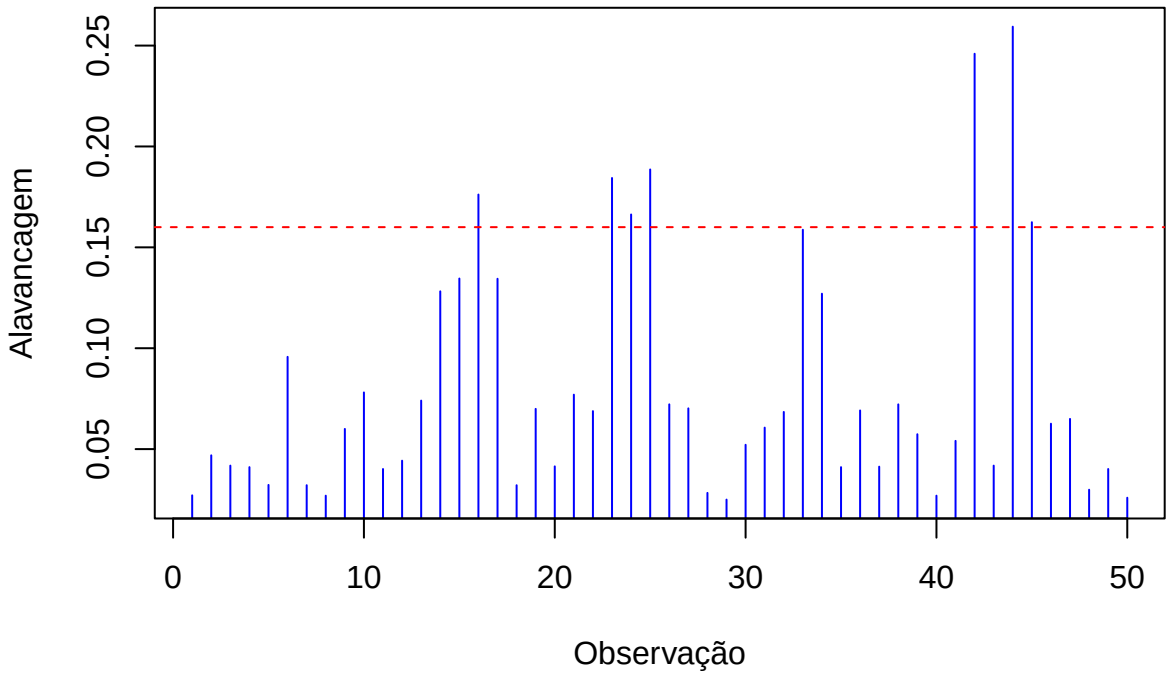




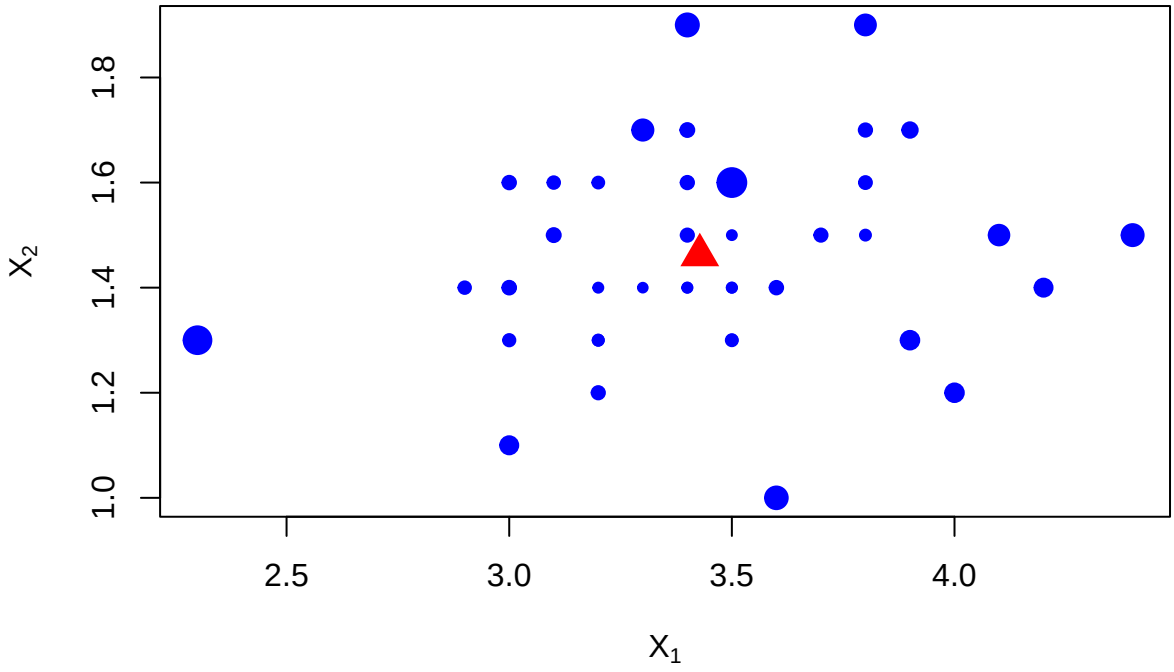
Nota 9. Comente os gráficos acima relacionando-os com alguma suposição do modelo.

Alguns gráficos representando a medida de alavancagem são mostrados abaixo.

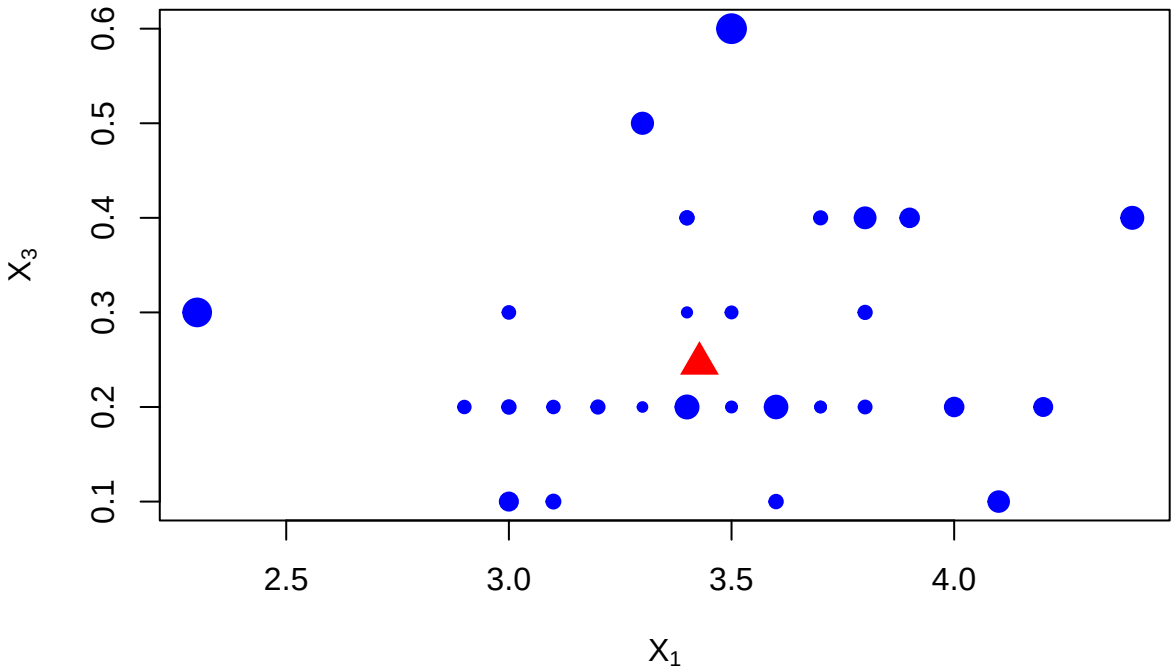
```
n <- nrow(dados)
minh <- min(infm1$hat)
cexh <- 2 * (infm1$hat - minh) / (max(infm1$h) - minh) + 1
plot(infm1$hat, type = "h", xlab = "Observação", ylab = "Alavancagem", col = "blue")
abline(h = 2 * p / n, lty = 2, col = "red")
```



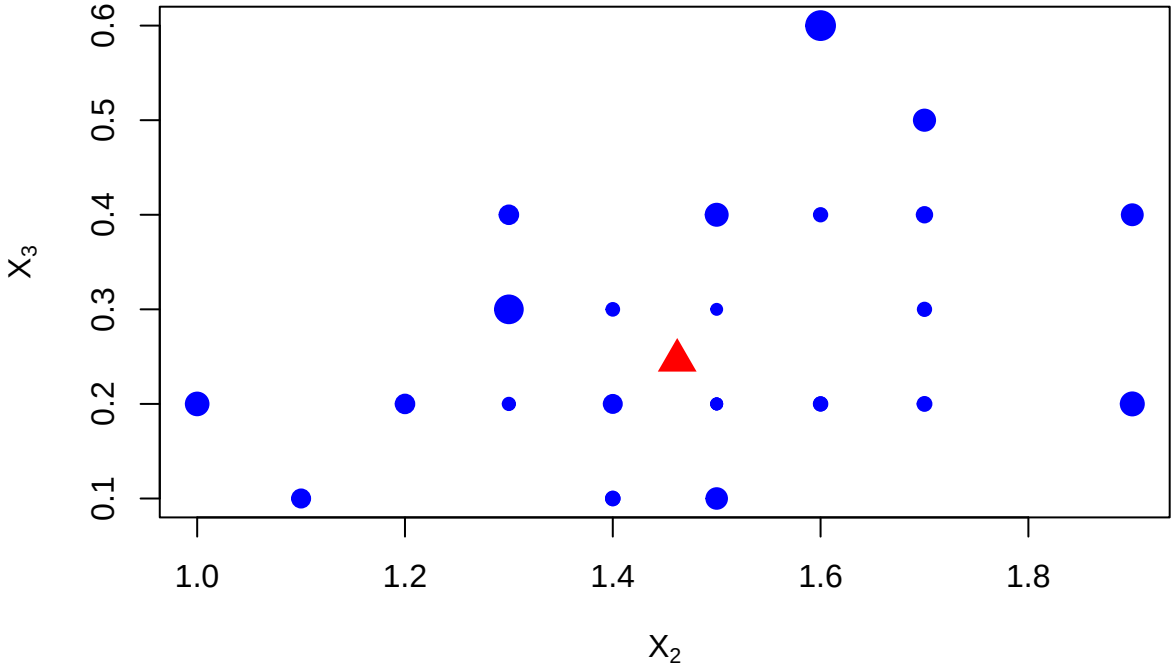

```
plot(dados[, 2], dados[, 3], pch = 20, cex = cexh, xlab = expression(X[1]),
     ylab =expression(X[2]), col = "blue")
points(mean(dados[, 2]), mean(dados[, 3]), pch = 17, col = "red", cex = 2)
```




```
plot(dados[, 2], dados[, 4], pch = 20, cex = cexh, xlab = expression(X[1]),
     ylab = expression(X[3]), col = "blue")
points(mean(dados[, 2]), mean(dados[, 4]), pch = 17, col = "red", cex = 2)
```




```
plot(dados[, 3], dados[, 4], pch = 20, cex = cexh, xlab = expression(X[2]),
     ylab = expression(X[3]), col = "blue")
points(mean(dados[, 3]), mean(dados[, 4]), pch = 17, col = "red", cex = 2)
```



Nota 10. Nos gráficos acima identifique as observações com alavancagem mais alta.

Nota 11. Calcule a matriz chapéu ($\boldsymbol{H}$). A função `model.matrix` fornece a matriz modelo. Para comparação, utilize as funções `hat` e `hatvalues`.

Dois gráficos com os resíduos studentizados deletados são apresentados a seguir.

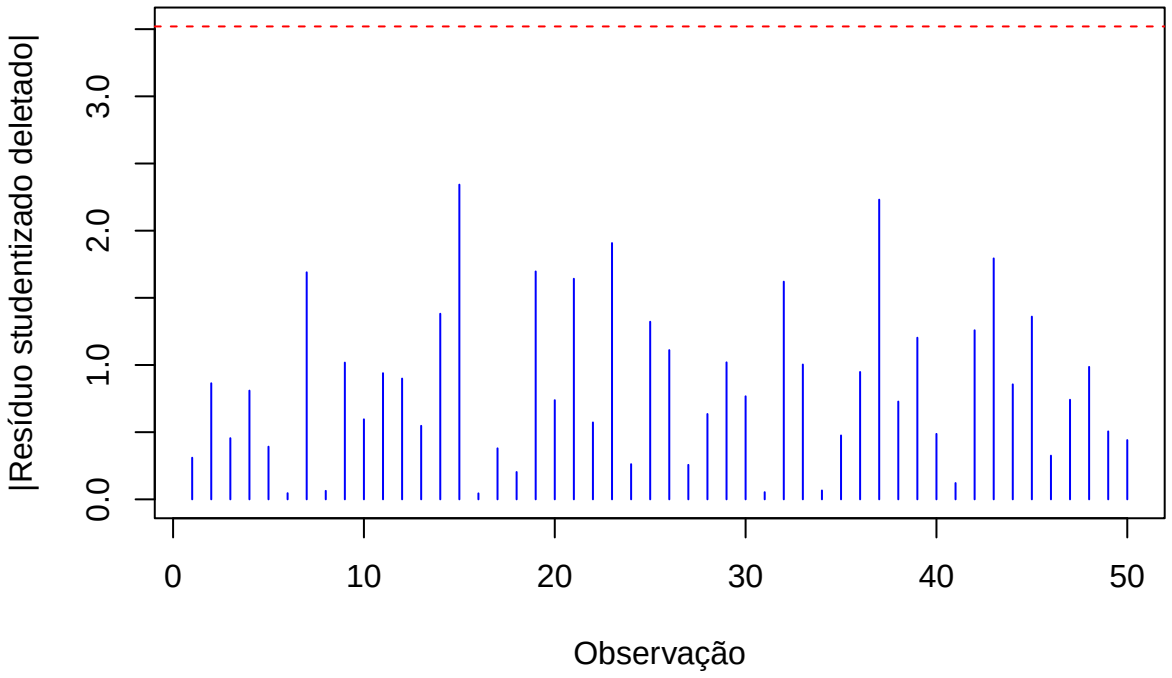
```
# Resíduo studentizado deletado
ti <- m1$residuals / (lm.influence(m1)$sigma * sqrt(1 - infm1$hat))
```

```r
summary(ti)
```

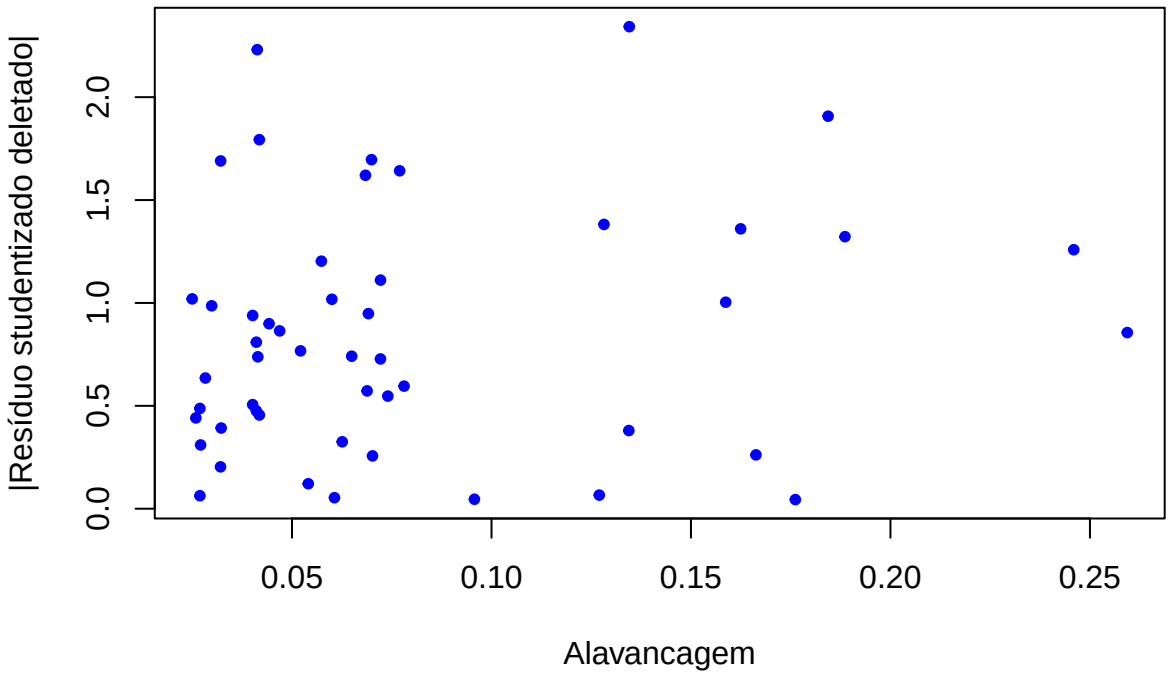
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.907550 -0.798755  0.053678 -0.001755  0.583479  2.342551
```

Nota 12. Verifique os resultados das funções `rstandard` e `rstudent`.

```r
tcrit <- qt(1 - 0.05 / (2 * n), df = n - p - 1) # alfa = 5%
plot(abs(ti), type = "h", xlab = "Observação", ylim = c(0, max(abs(ti), tcrit)),
     ylab = "|Resíduo studentizado deletado|", col = "blue")
abline(h = tcrit, lty = 2, col = "red")
```




```r
plot(infm1$hat, abs(ti), pch = 20, xlab = "Alavancagem", col = "blue",
     ylab = "|Resíduo studentizado deletado|")
```



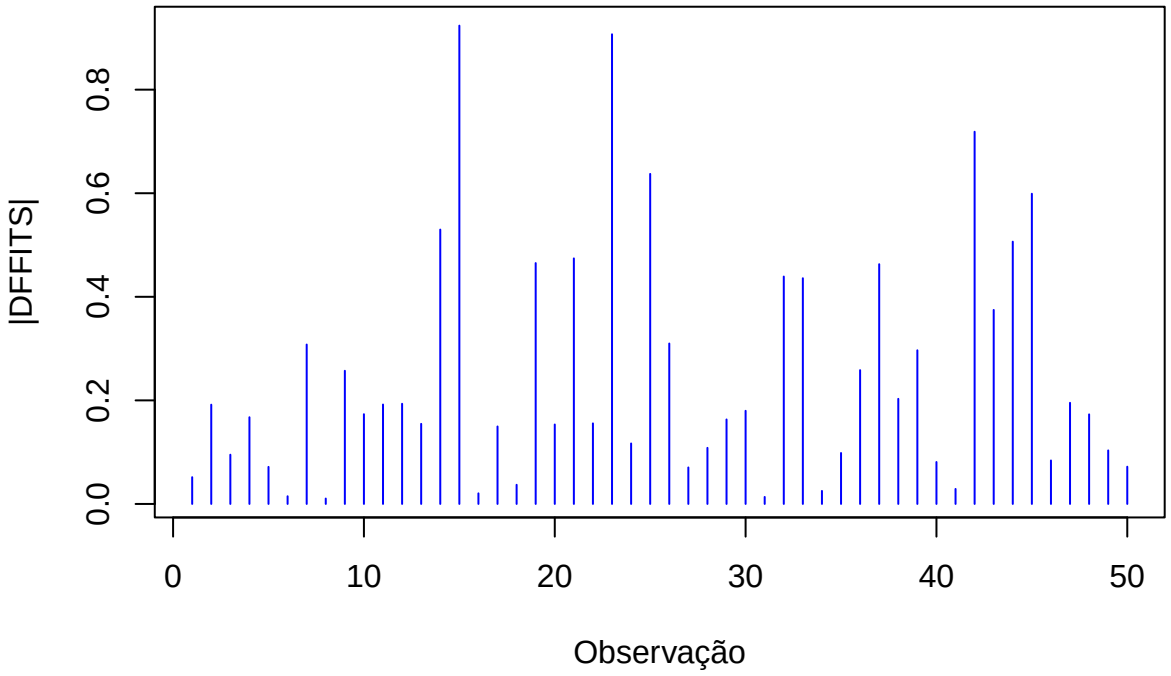
Nota 13. Comente os gráficos acima relacionando-os com suposições do modelo.

Para ajudar a identificar observações influentes, são apresentados gráficos de índices das medidas DFFITS e distância de Cook.

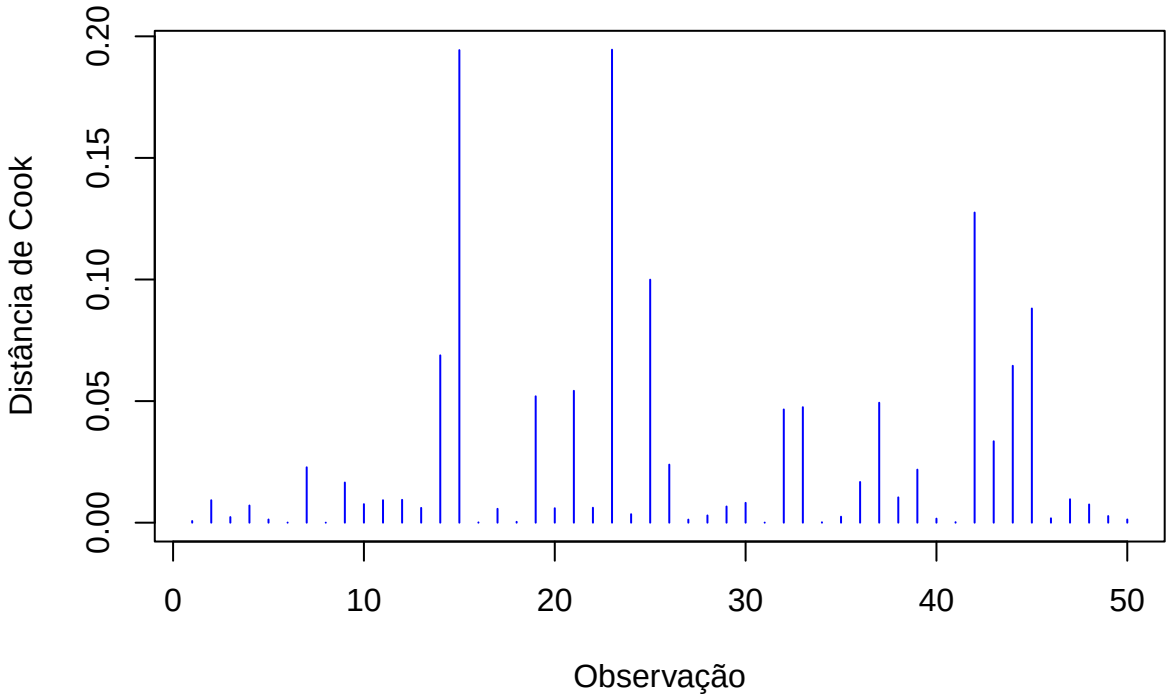
```
dffits <- ti * sqrt(infm1$hat / (1 - infm1$hat))
dcook <- m1$resid^2 * infm1$hat / (p * summary(m1)$s^2 * (1 - infm1$hat)^2)
```

Nota 14. Compare os resultados acima com os obtidos com as funções `dffits` e `cooks.distance`.

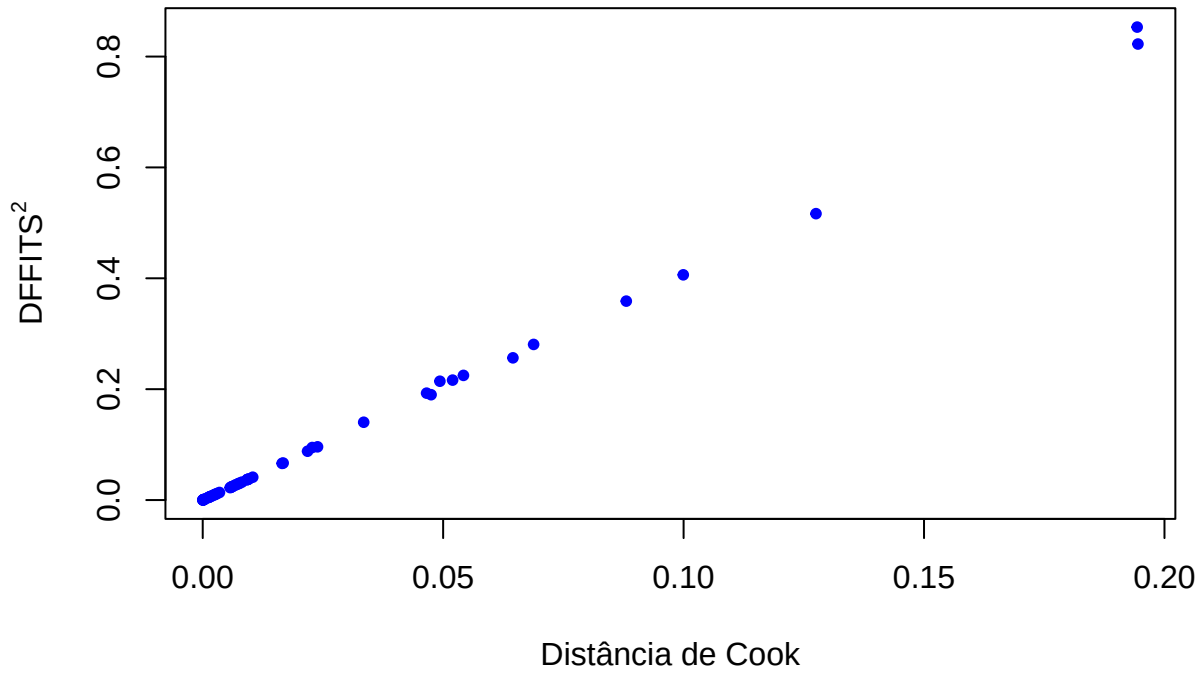
```
plot(abs(dffits), type = "h", xlab = "Observação", ylab = "|DFFITS|", col = "blue")
```



```
plot(dcook, type = "h", xlab = "Observação", ylab = "Distância de Cook",
     col = "blue")
```




```
plot(dcook, dffits^2, ylab = expression(DFFITS^2), xlab = "Distância de Cook",
     col = "blue", pch = 20)
```



Nota 15. Prove que

$$\text{DFFITS}_i^2 = p\frac{\text{MSE}}{\text{MSE}_{[i]}}D_i, \tag{1}$$

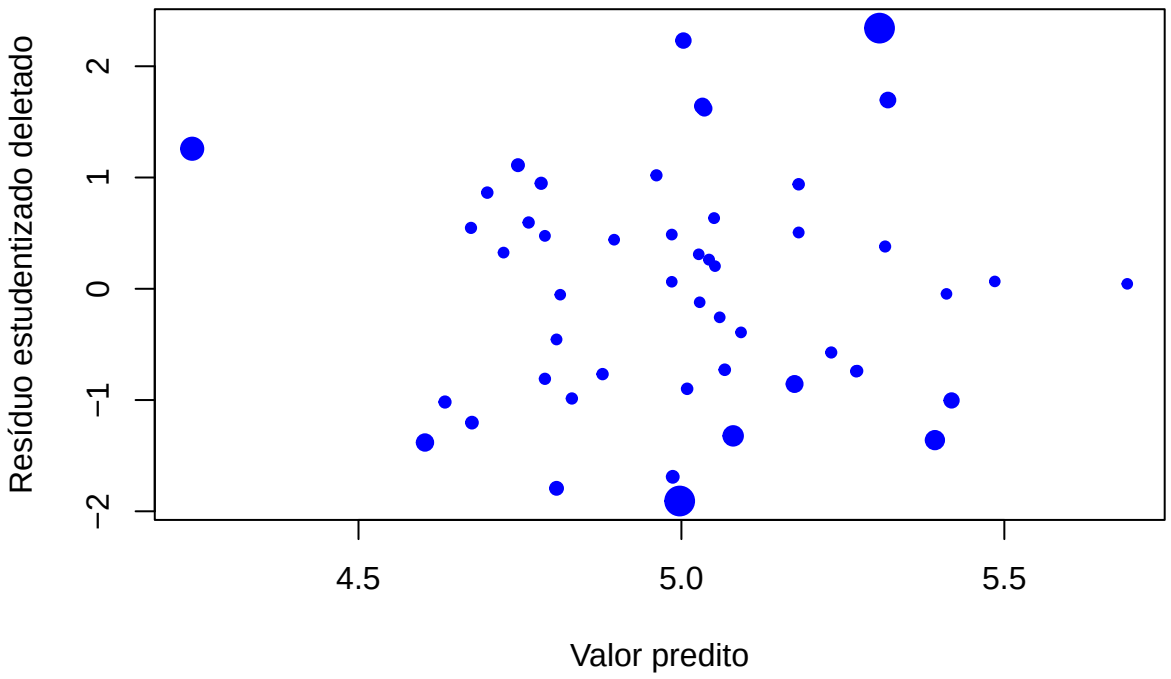
sendo que D_i denota a distância de Cook, $i = 1, \ldots, n$. Usando o resultado em (1), procure explicar o

comportamento do gráfico acima.

Nota 16. Identifique as observações mais influentes nos três gráficos acima. Tente apontar alguma diferença marcante delas em relação às demais.

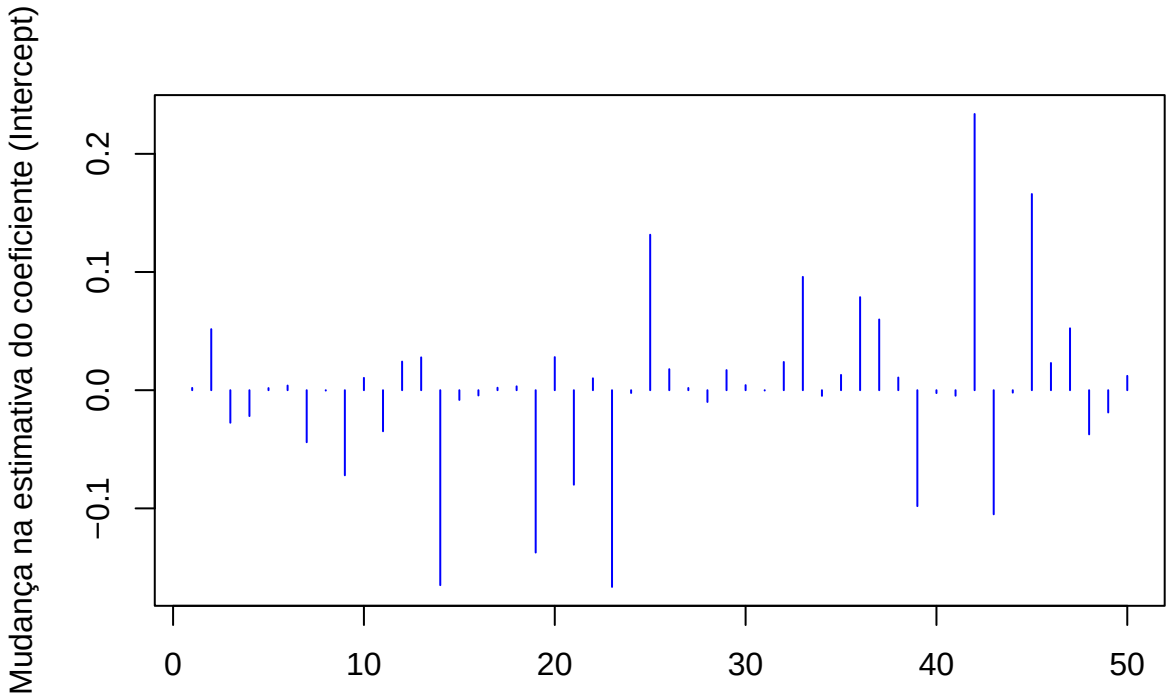
O gráfico abaixo é chamado de gráfico de influência proporcional. O diâmetro de cada círculo é proporcional ao valor da distância de Cook.

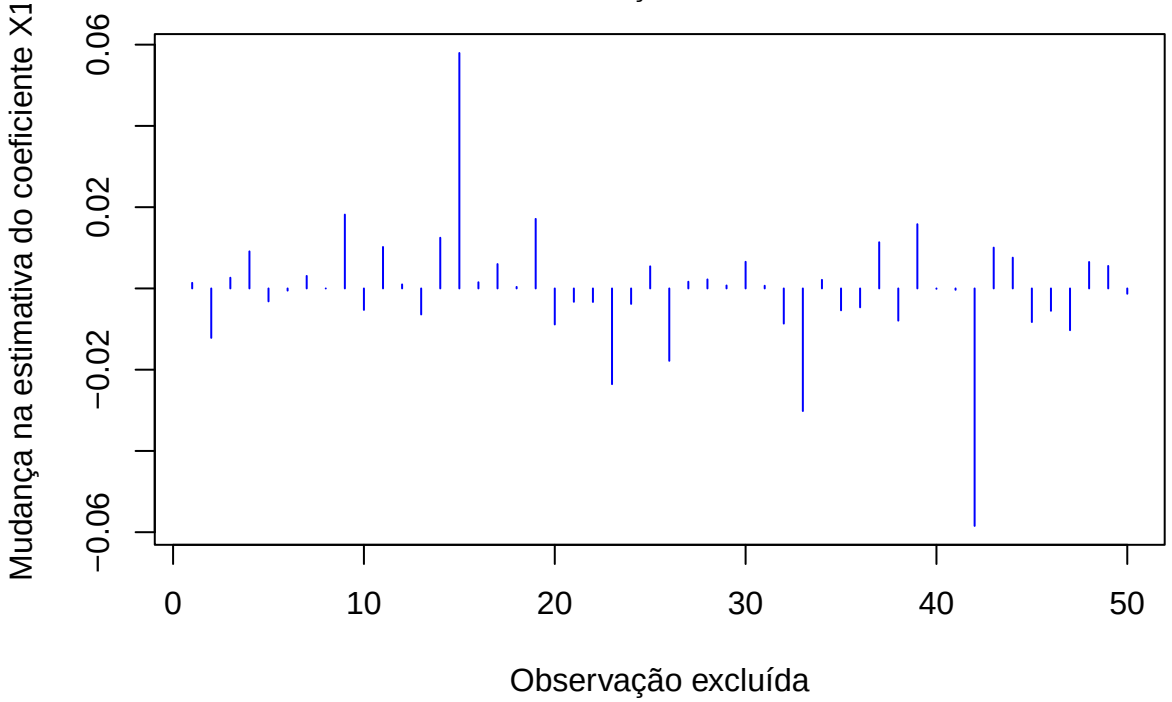
```
mind <- min(dcook)
cexd <- 2 * (dcook - mind) / (max(dcook) - mind) + 1
plot(m1$fitted.values, ti, pch = 20, cex = cexd, xlab = "Valor predito",
     ylab = "Resíduo estudentizado deletado", col = "blue")
```

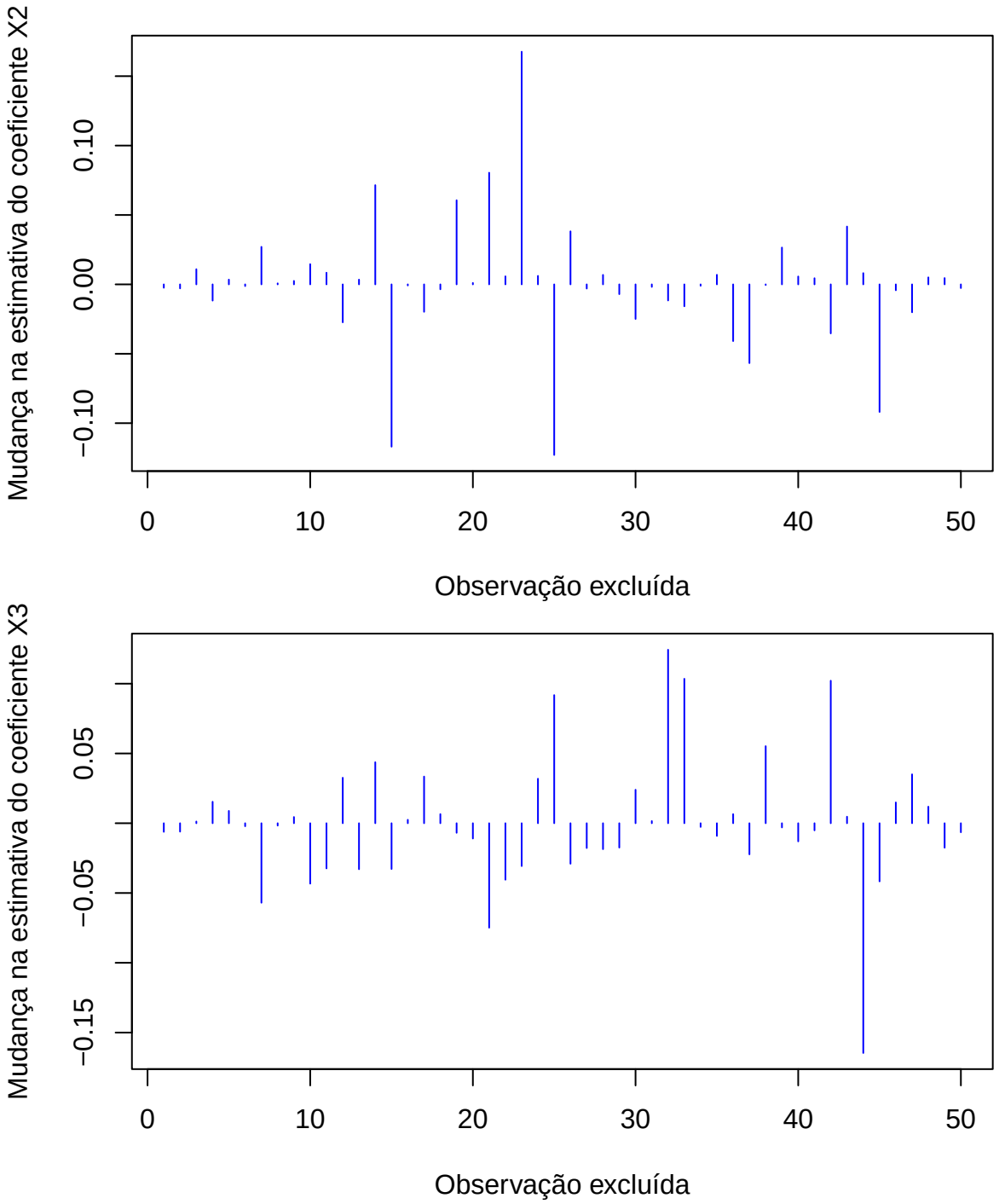


Em seguida, para cada coeficiente do modelo, é apresentada a mudança na estimativa decorrente da exclusão de uma observação de cada vez.

```
for (j in 1:p) {
   plot(lm.influence(m1)$coef[, j], type = "h", xlab = "Observação excluída",
        ylab = paste("Mudança na estimativa do coeficiente",
        names(coef(m1))[j]), col = "blue")
}
```


Mudança na estimativa do coeficiente (Intercept)
0.2
0.1
0.0
−0.1
0
10
20
30
40
50
Observação excluída


Mudança na estimativa do coeficiente X1
0.06
0.02
−0.02
−0.06
0
10
20
30
40
50
Observação excluída

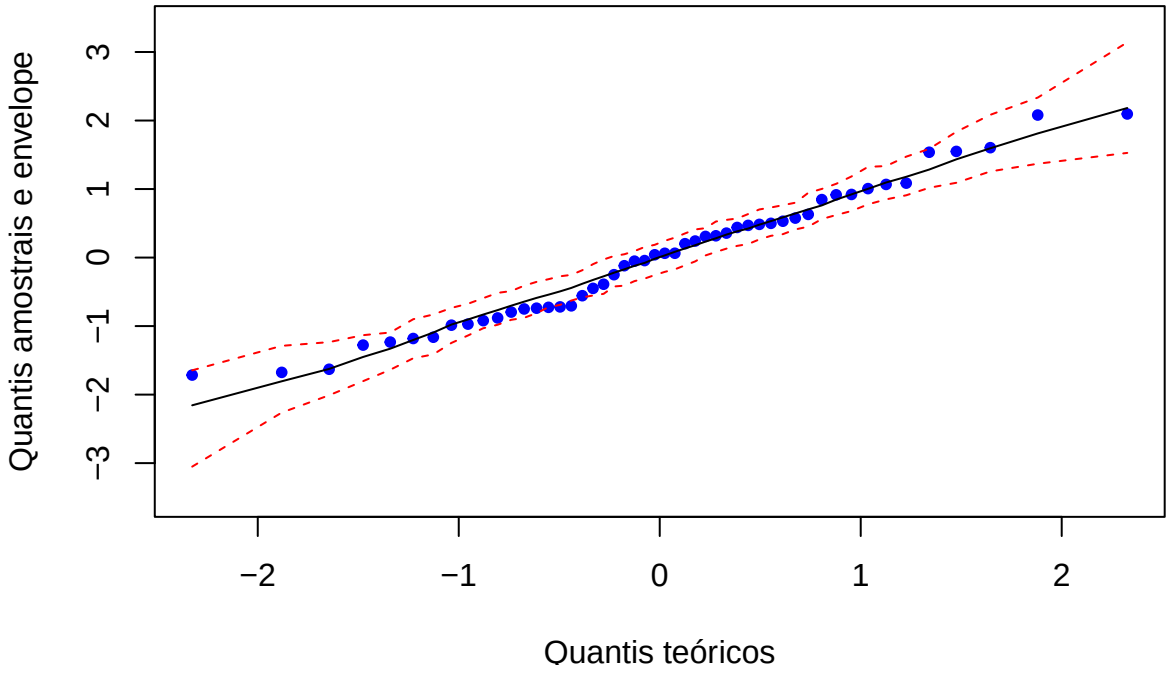


Nota 17. Calcule DFBETAS e represente graficamente. Compare com os resultados da função `dfbetas`.

Concluímos apresentando o gráfico de quantis (gráfico QQ) dos resíduos de quantis normalizados com base em $J = 100$ amostras simuladas.

```
## Envelope - resíduos de quantis
J <- 100 # Número de amostras
set.seed(184587)
sige <- summary(m1)$sigma
rqobs <- qnorm(pnorm(dados$Y, m1$fitted.values, sige))
```

```r
mrq <- matrix(NA, J, n)
for (j in 1:J) {
   Yj <- rnorm(n, m1$fitted.values, sige)
   mj <- lm(Yj ~ X1 + X2 + X3, data = dados)
   mrq[j,] <- qnorm(pnorm(Yj, mj$fitted.values, summary(mj)$sigma))
   mrq[j,] <- sort(mrq[j,])
}
# Limites da banda e média
conf <- 0.95
infsup <- apply(mrq, 2, quantile, probs = c((1 - conf) / 2,
          (1 + conf) / 2), type = 6)
media <- colMeans(mrq)
faixay <- range(mrq, rqobs)
qq0 <- qqnorm(rqobs, main = "", xlab = "Quantis teóricos", pch = 20,
        col = "blue", ylab = "Quantis amostrais e envelope", ylim = faixay)
eixox <- sort(qq0$x)
lines(eixox, media)
lines(eixox, infsup[1,], col = "red", lty = 2)
lines(eixox, infsup[2,], col = "red", lty = 2)
```



Nota 18. Efetue o teste da hipótese $H_0 : \beta_2 = \beta_3 = 0$ contra $H_1 : \beta_2 \neq 0$ e/ou $\beta_3 \neq 0$. Caso seja possível simplificar o modelo, refaça o exemplo com o modelo mais simples.

Nota 19. Ajuste um modelo para o cojunto de dados completo com $n = 150$ observações e considerando que `Species` é uma covariável (quinta coluna do conjunto de dados `iris`) com três níves, pois são três espécies.

Nota 20. Desenvolva o exemplo em linguagem Python.