Construindo Gráficos Diagnósticos Manualmente

Mário Olímpio de Menezes

26/04/2020

```
library(tidyverse)
library(ggplot2)
library(ggpubr)
library(ggfortify)
library(GGally)
```

Análise dos dados `state.x77`

```
states <- as.data.frame(state.x77[, c("Murder", "Population",
    "Illiteracy", "Income", "Frost")])
```

```
ggcorr(states, palette = "RdYlGn", name = bquote(rho),
    label = TRUE, label_color = "black") +
    labs(caption = "Fonte: Elaborado pelo autor") +
    theme(plot.caption = element_text(hjust = 0,
        size = 8))
```


Correlação entre variáveis da base state.x77

Ajustando um Modelo de Regressão Linear Multivariada

Vamos utilizar a função `lm()` para fazer o ajuste multivariado

```
fit <- lm(Murder ~ Population + Illiteracy + Income + Frost, data = states)
summary(fit)
```

```
##
## Call:
## lm(formula = Murder ~ Population + Illiteracy + Income + Frost,
##     data = states)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7960 -1.6495 -0.0811  1.4815  7.6210
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.235e+00  3.866e+00   0.319   0.7510
## Population  2.237e-04  9.052e-05   2.471   0.0173 *
## Illiteracy  4.143e+00  8.744e-01   4.738 2.19e-05 ***
## Income      6.442e-05  6.837e-04   0.094   0.9253
## Frost       5.813e-04  1.005e-02   0.058   0.9541
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.535 on 45 degrees of freedom
## Multiple R-squared:  0.567,  Adjusted R-squared:  0.5285
## F-statistic: 14.73 on 4 and 45 DF,  p-value: 9.133e-08
```

```
fit <- update(fit, . ~ . - Frost)
summary(fit)
```

```
##
## Call:
## lm(formula = Murder ~ Population + Illiteracy + Income, data = states)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7846 -1.6768 -0.0839  1.4783  7.6417
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.3402721  3.3694210   0.398   0.6926
## Population  0.0002219  0.0000842   2.635   0.0114 *
## Illiteracy  4.1109188  0.6706786   6.129 1.85e-07 ***
## Income      0.0000644  0.0006762   0.095   0.9245
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.507 on 46 degrees of freedom
## Multiple R-squared:  0.5669, Adjusted R-squared:  0.5387
## F-statistic: 20.07 on 3 and 46 DF,  p-value: 1.84e-08
```

Continuamos nossa **atualização** do modelo, removendo as variáveis que não tiveram significado estatístico. `Income` é a proxima candidata a remoção.

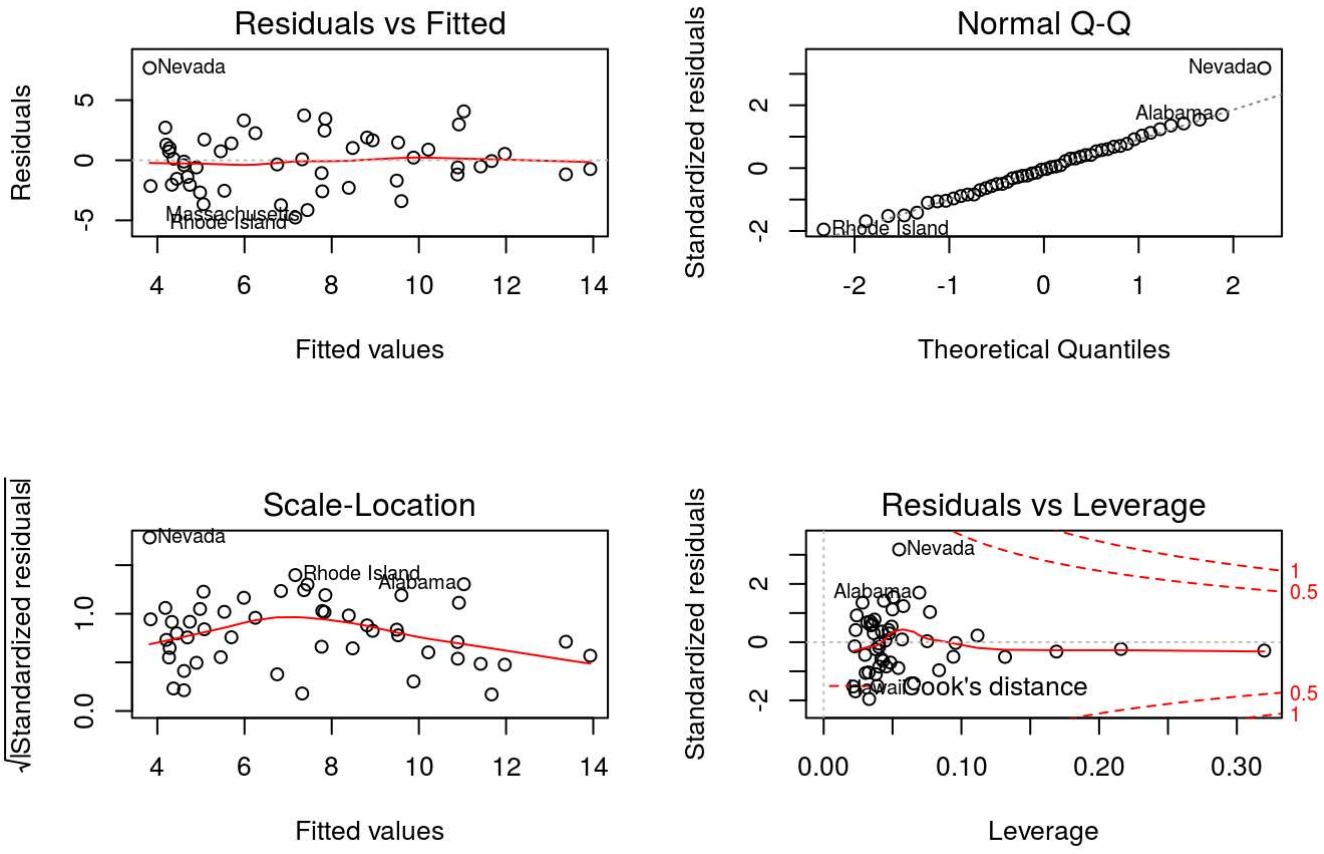
```
fit <- update(fit, . ~ . - Income)
summary(fit)
```

```
##
## Call:
## lm(formula = Murder ~ Population + Illiteracy, data = states)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7652 -1.6561 -0.0898  1.4570  7.6758
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.652e+00  8.101e-01   2.039  0.04713 *
## Population  2.242e-04  7.984e-05   2.808  0.00724 **
## Illiteracy  4.081e+00  5.848e-01   6.978 8.83e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.481 on 47 degrees of freedom
## Multiple R-squared:  0.5668, Adjusted R-squared:  0.5484
## F-statistic: 30.75 on 2 and 47 DF,  p-value: 2.893e-09
```

Todas os parâmetros agora têm significância estatística.

Gráficos Diagnósticos

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



Analisando os gráficos diagnósticos, vemos que:

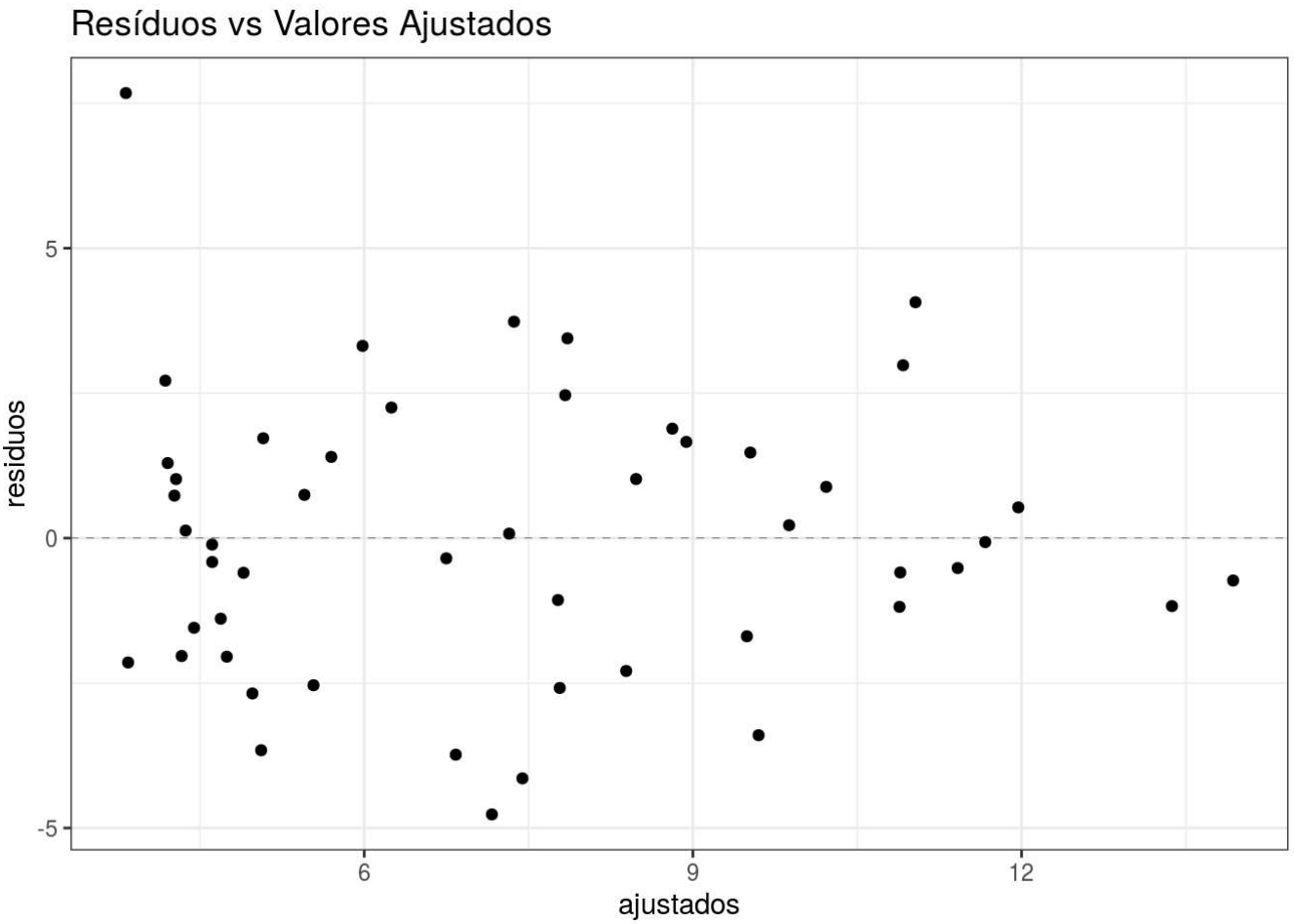
- Não observamos padrão claro no gráfico *Residuals x Fitted-values*, o que indica uma independência dos resíduos com relação aos valores ajustados.
- O gráfico QQ-Plot mostra que os resíduos tem distribuição normal.
- E no gráfico *Scale-Location*, que mostra o valor absoluto dos resíduos, observamos que a amplitude dos resíduos não depende dos valores ajustados, o que indica homocedasticidade.

Construindo manualmente os gráficos diagnósticos

Resíduos vs Valores Ajustados

```
residuos <- fit$residuals
ajustados <- fit$fitted.values
df <- data.frame(ajustados = ajustados, residuos = residuos)
```

```
df %>%
  ggplot() + geom_point(aes(x = ajustados, y = residuos)) + geom_hline(yintercept = 0, linetype = "dashe
d", size = 0.1) + labs(title = "Resíduos vs Valores Ajustados") + theme_bw()
```



QQ-Plot

Para fazer o QQ-Plot precisamos:

- número de pontos no conjunto de dados, que corresponde ao número de resíduos que temos.
- padronizar os resíduos (média zero e desvio padrão 1);
- ordenar os resíduos padronizados;
- gerar uma distribuição normal de mesmo tamanho (mesmo número de pontos que temos de resíduos);
- ordenar os resíduos teóricos.

```
n <- length(residuos)
residuos_std <- scale(residuos)
residuos_std <- sort(residuos_std)
residuos_teorico <- rnorm(n)
residuos_teorico <- sort(residuos_teorico)
```

```
head(residuos_std)
```

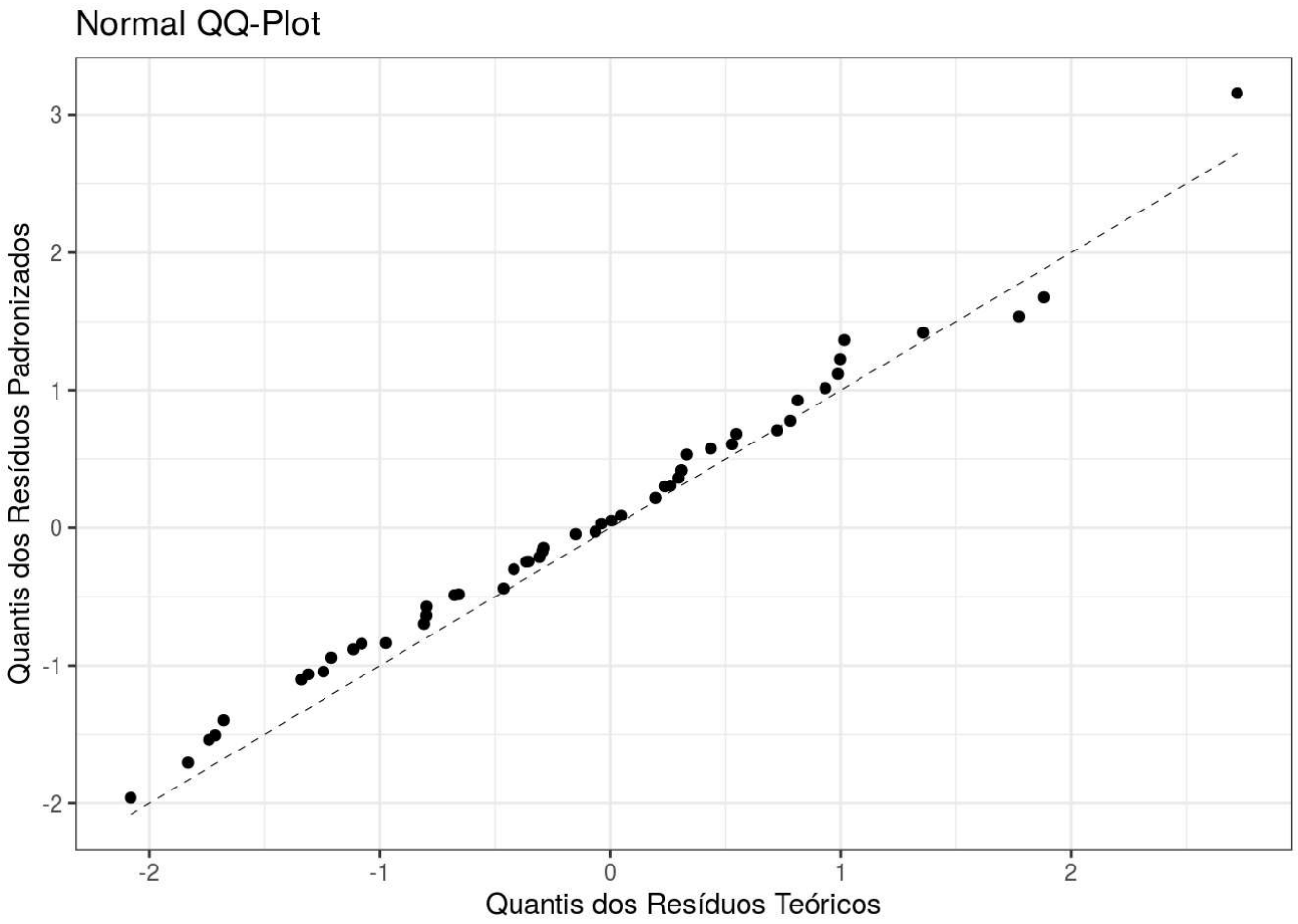
```
## [1] -1.961318 -1.705537 -1.537427 -1.505985 -1.399217 -1.102660
```

```
head(residuos_teorico)
```

```
## [1] -2.081439 -1.831983 -1.741645 -1.713715 -1.677354 -1.339945
```

```
df2 <- data.frame(residstd = residuos_std, residteo = residuos_teorico)
```

```
df2 %>%
  ggplot() + geom_point(aes(x = residteo , y = residstd)) + geom_line(aes(x = residteo, y = residteo), li
netype = "dashed", size = 0.2) + labs(x = "Quantis dos Resíduos Teóricos", y = "Quantis dos Resíduos Padr
onizados", title = "Normal QQ-Plot") + theme_bw()
```



Os pontos no gráfico QQ-Plot são obtidos fazendo **x = quantis dos resíduos teóricos** e **y = quantis dos resíduos padronizados**. A linha tracejada é obtida utilizando em **x** e em **y** os quantis teóricos