MAE0217 - Estatística Descritiva - Lista 4

Natalia Hitomi Koza[1]
Rafael Gonçalves Pereira da Silva[2]
Ricardo Geraldes Tolesano[3]
Rubens Kushimizo Rodrigues Xavier[4]
Rubens Gomes Neto[5]
Rubens Santos Andrade Filho[6]
Thamires dos Santos Matos[7]

June de 2021

Sumário

[1]Número USP: 10698432
[2]Número USP: 9009600
[3]Número USP: 10734557
[4]Número USP: 8626718
[5]Número USP: 9318484
[6]Número USP: 10370336
[7]Número USP: 9402940

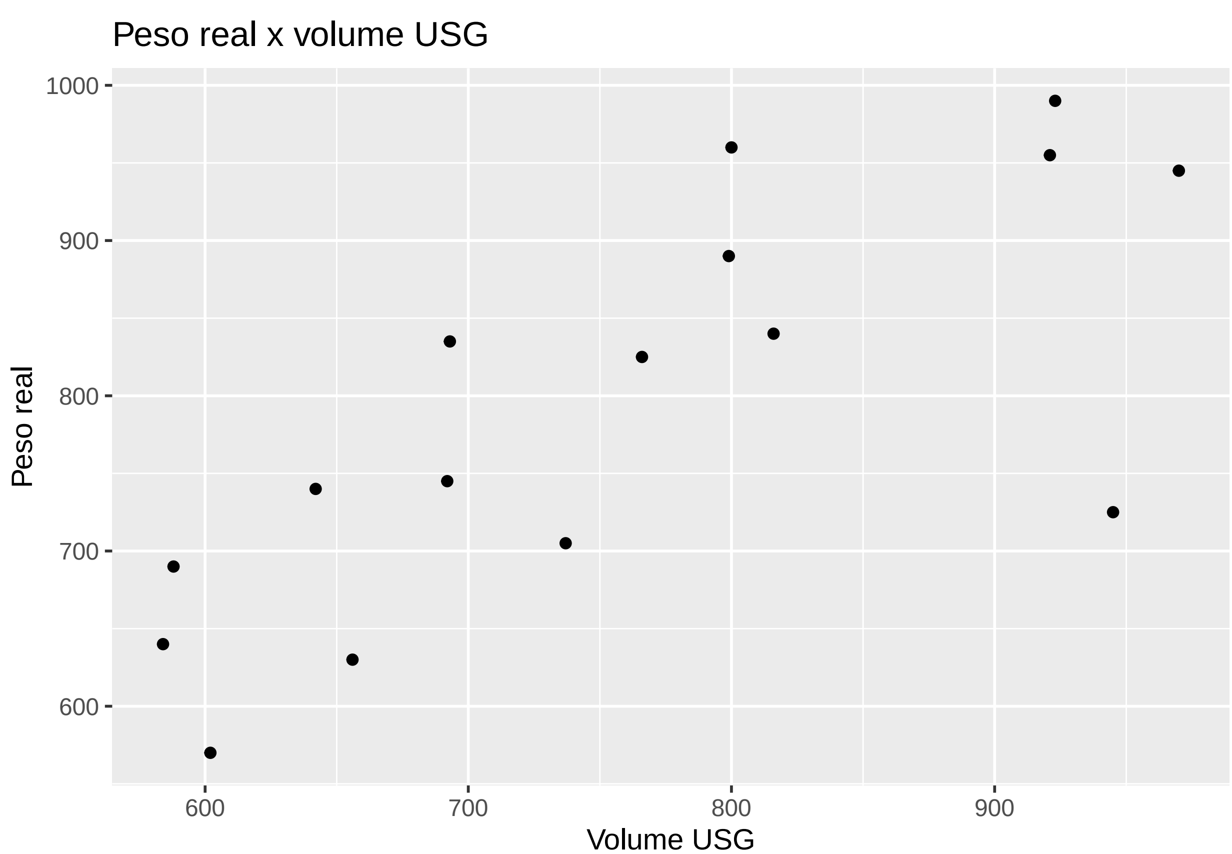
Exercício 1

i)

Tomaremos Volume USG como a variável explicativa x e Peso Real como a variável resposta y. Adotaremos o modelo de regressão linear simples $y_i = \alpha + \beta x_i + e_i$, onde α é o intercepto, *beta* é a inclinação da reta, e e_i são erros aleatórias não correlacionados.

ii)

```
scatter_title <- "Peso real x volume USG"
scatter_x <- "Volume USG"
scatter_y <- "Peso real"
fit_titles <- list("Resíduos vs observações x para o ajuste feito no modelo",
                   "Gráfico Q-Q normal para o ajuste feito no modelo",
                   "Resíduos normalizados vs observações x para o ajuste feito no modelo",
                   "Resíduos normalizados vs influência das observações para o ajuste feito no modelo")


dados1 <- read_excel("data/peso_volume_figado.xlsx")
dados1 <- dados1[order(dados1$volume_usg), ]
# ggplot(dados, aes(x=volume_usg, y=peso_real)) + geom_point() + geom_smooth(method=lm)
questao_i <- function(dados) {}
ggplot(dados1, aes(x=volume_usg, y=peso_real)) + geom_point() + labs(title=scatter_title, x=scatter_x, y
```

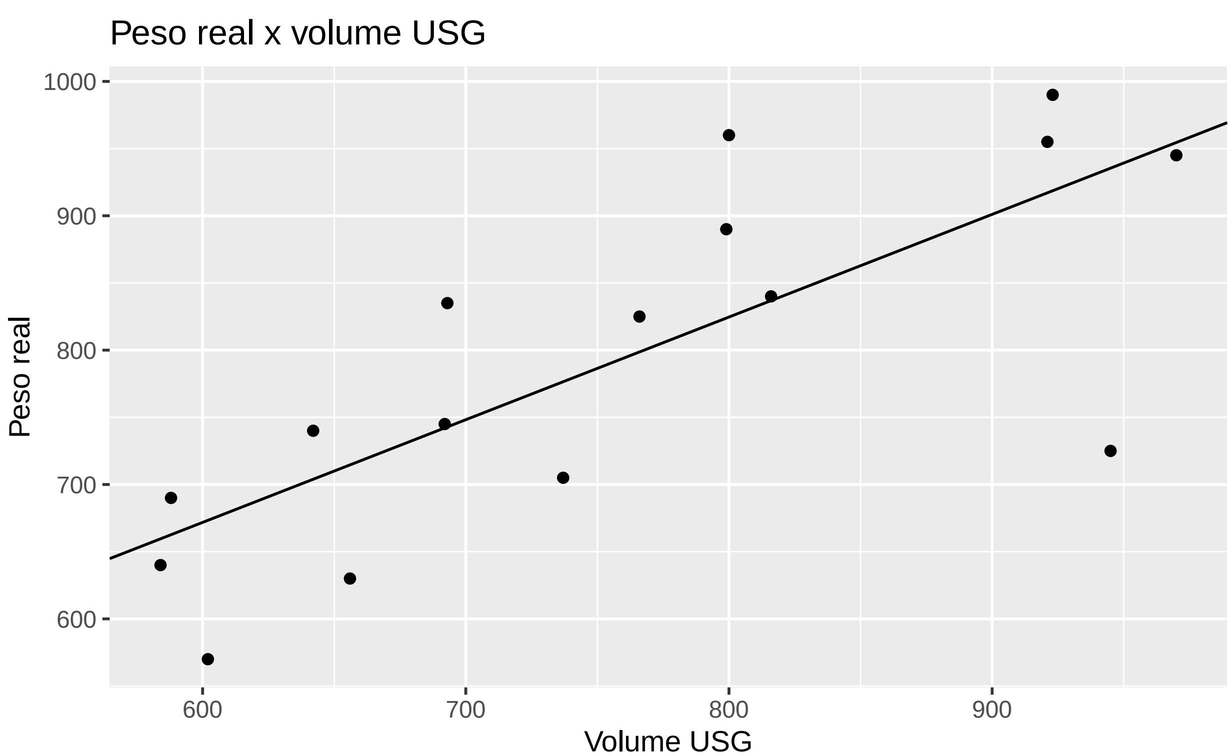


iii)

Realizaremos o ajuste do modelo e mostraremos algumas métricas de qualidade do modelo:

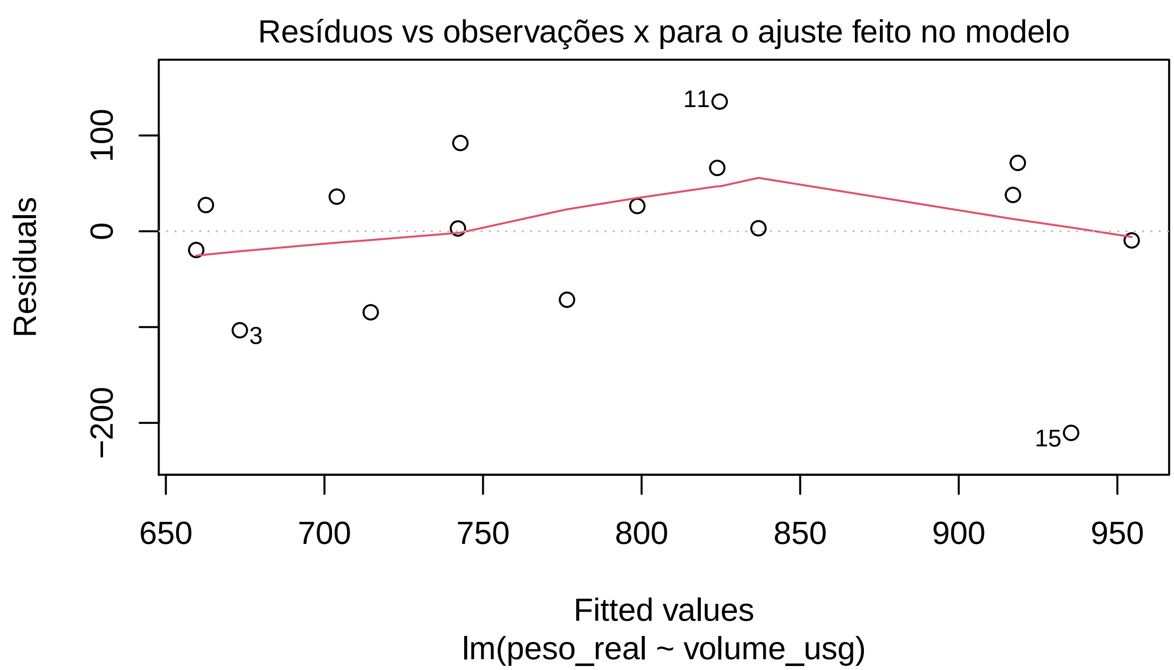
```
ajustarModelo <- function(dados) {
  ajuste <- lm(peso_real ~ volume_usg, data=dados)
  intercept <- ajuste$coefficients[1]
  slope <- ajuste$coefficients[2]
  print("O ajuste encontrou os coeficientes:")
  print(paste("Alpha:", intercept))
  print(paste("Beta:", slope))
  p <- ggplot(dados, aes(x=volume_usg, y=peso_real)) + geom_point() + geom_abline(intercept = intercept
  plot(p)
  print(summary(ajuste))
  plot(ajuste,
       caption=fit_titles)

  return(ajuste)
}

ajuste <- ajustarModelo(dados1)
```

```
## [1] "O ajuste encontrou os coeficientes:"
## [1] "Alpha: 213.276155355598"
## [1] "Beta: 0.764181763170465"
```




```
##
```

```
## Call:
## lm(formula = peso_real ~ volume_usg, data = dados)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -210.43  -32.54   14.76   44.97  135.38
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 213.2762   133.3334   1.600 0.132011
## volume_usg    0.7642     0.1734   4.407 0.000597 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87.91 on 14 degrees of freedom
## Multiple R-squared:  0.5811, Adjusted R-squared:  0.5512
## F-statistic: 19.42 on 1 and 14 DF,  p-value: 0.000597
```



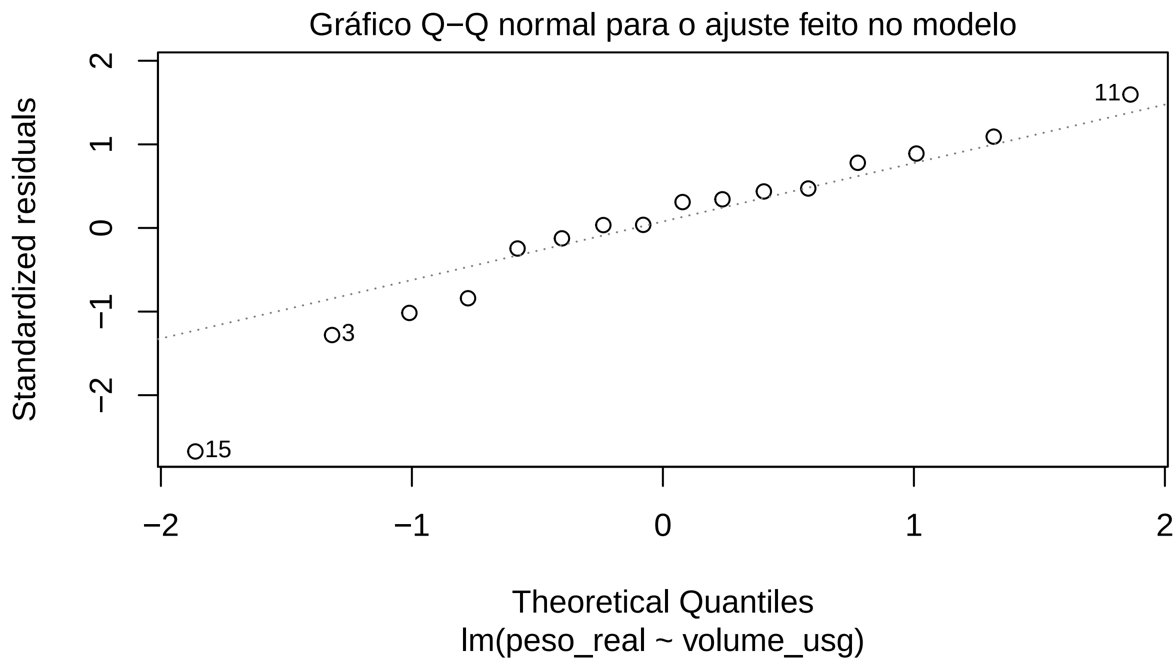
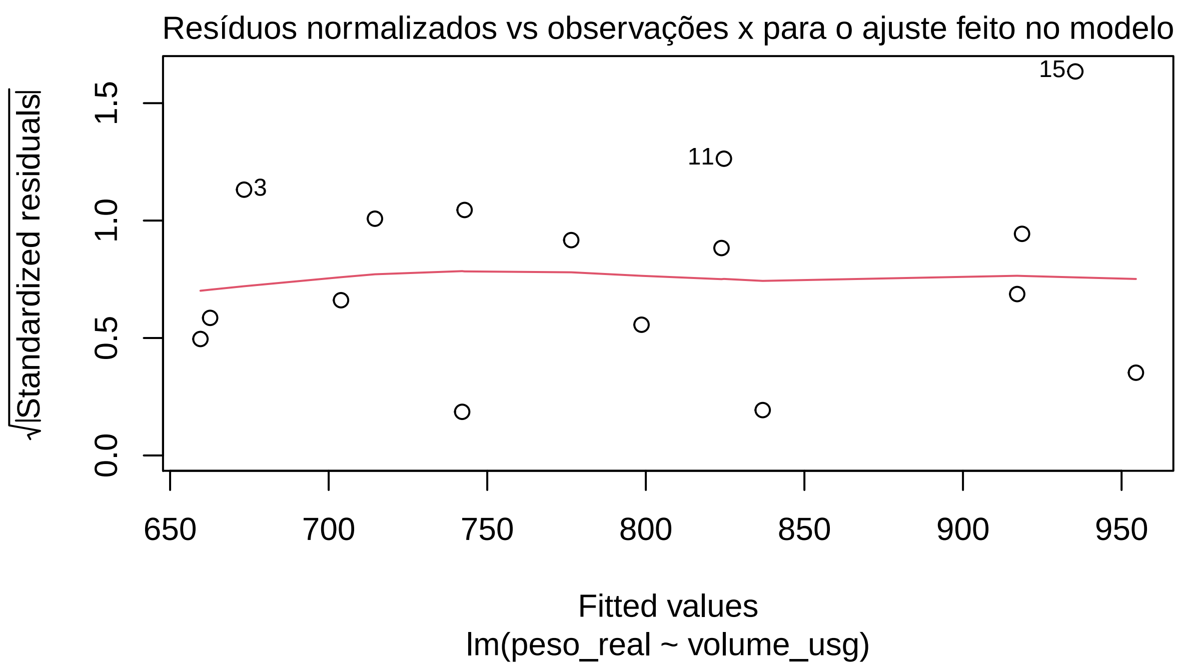
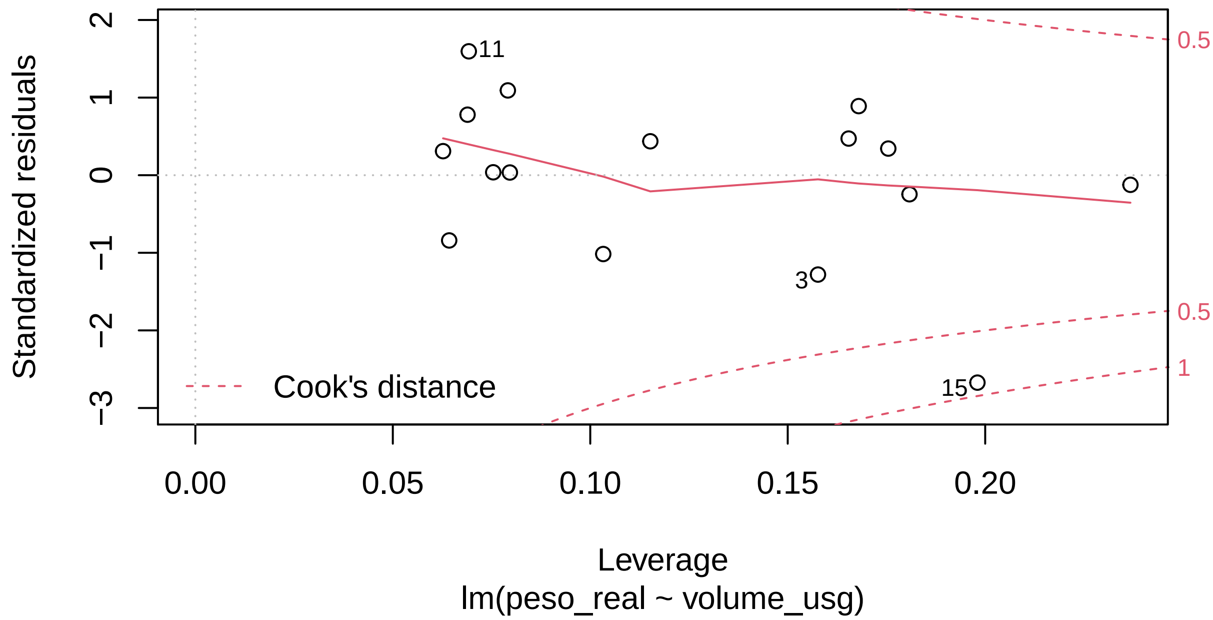


Gráfico Q−Q normal para o ajuste feito no modelo
Standardized residuals
Theoretical Quantiles
lm(peso_real ~ volume_usg)
11
3
15
−2
−1
0
1
2

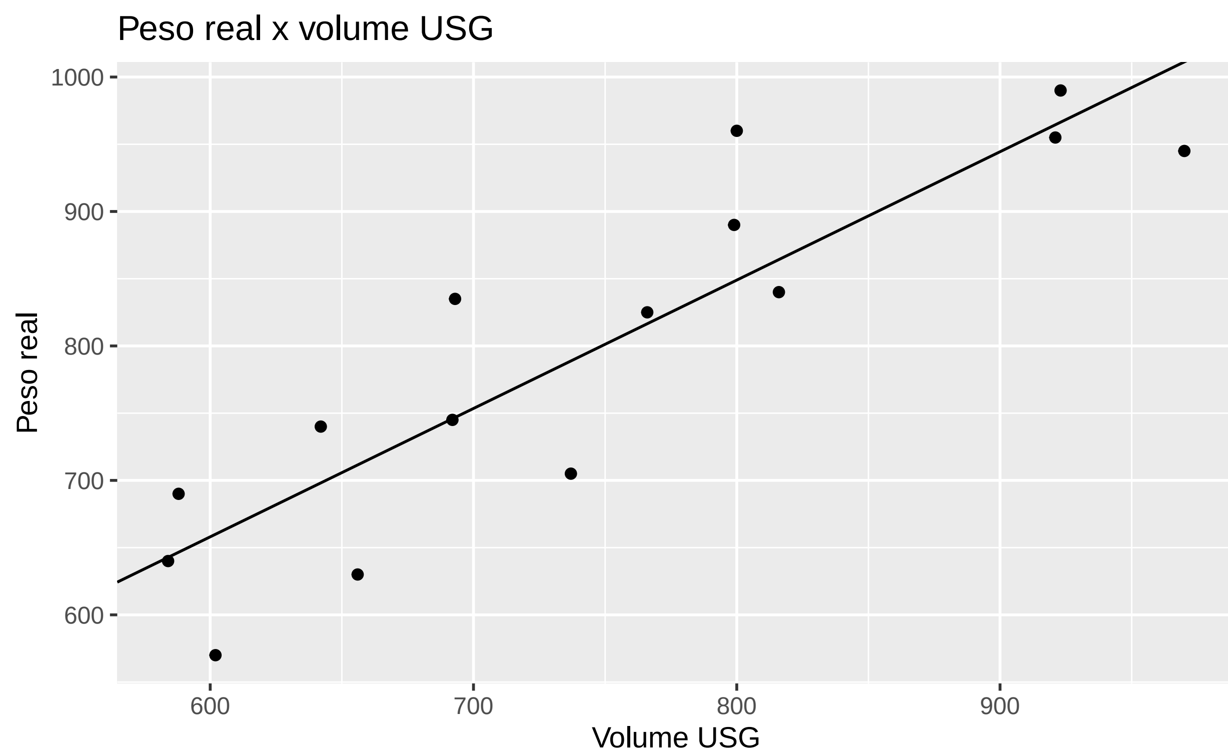

Resíduos normalizados vs observações x para o ajuste feito no modelo
√|Standardized residuals|
Fitted values
lm(peso_real ~ volume_usg)
15
11
3
0.0
0.5
1.0
1.5
650
700
750
800
850
900
950



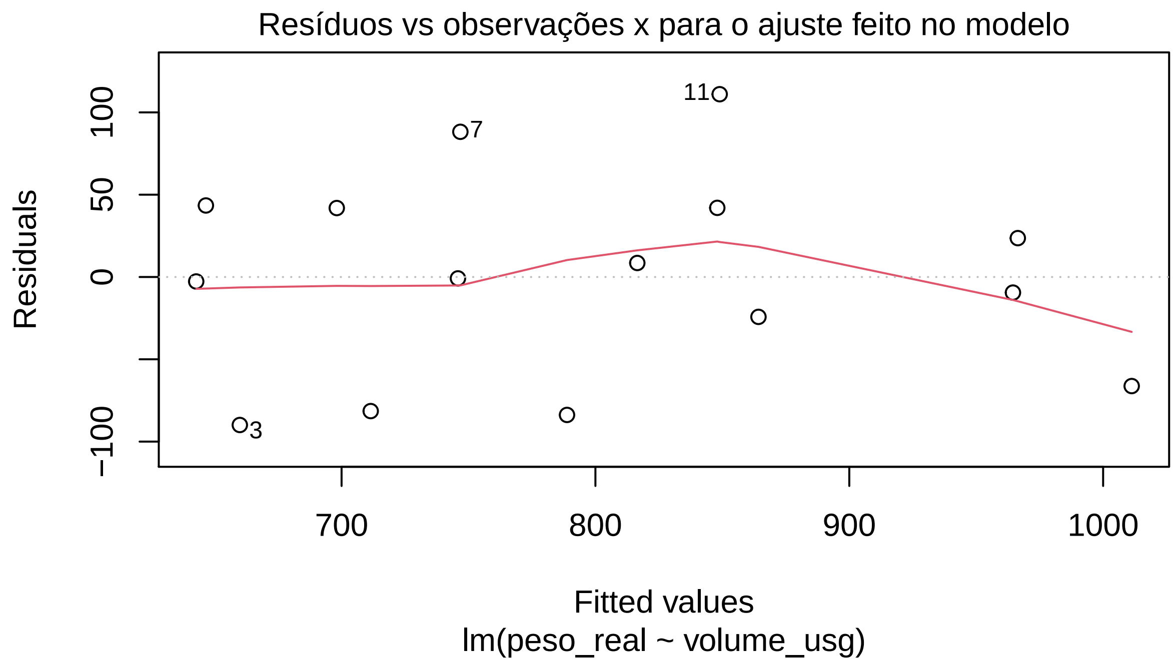
A análise do ajuste indicou que as observações 3, 11 e 15 são mais influentes no modelo. Em especial, a observação 15 se destaca como outlier em todos os gráficos mostrados. Realizaremos novamente o ajuste com essa observação removida. Não removeremos as observações 3 e 11 dado que possuímos poucas observações e elas não fogem do padrão na mesma intensidade elevada da observação 15.

```
dados2 <- dados1[-c(15), ]
ajuste <- ajustarModelo(dados2)
```

```
## [1] "O ajuste encontrou os coeficientes:"
## [1] "Alpha: 85.159261447348"
## [1] "Beta: 0.954742253846616"
```




```
##
## Call:
## lm(formula = peso_real ~ volume_usg, data = dados)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -89.914 -45.244  -0.841  41.949 111.047
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  85.1593   102.8848   0.828    0.423
## volume_usg    0.9547     0.1361   7.013 9.17e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.83 on 13 degrees of freedom
## Multiple R-squared:  0.7909, Adjusted R-squared:  0.7748
## F-statistic: 49.18 on 1 and 13 DF,  p-value: 9.167e-06
```


Resíduos vs observações x para o ajuste feito no modelo
Residuals
−100
0
50
100
700
800
900
1000
Fitted values
lm(peso_real ~ volume_usg)
11
7
3

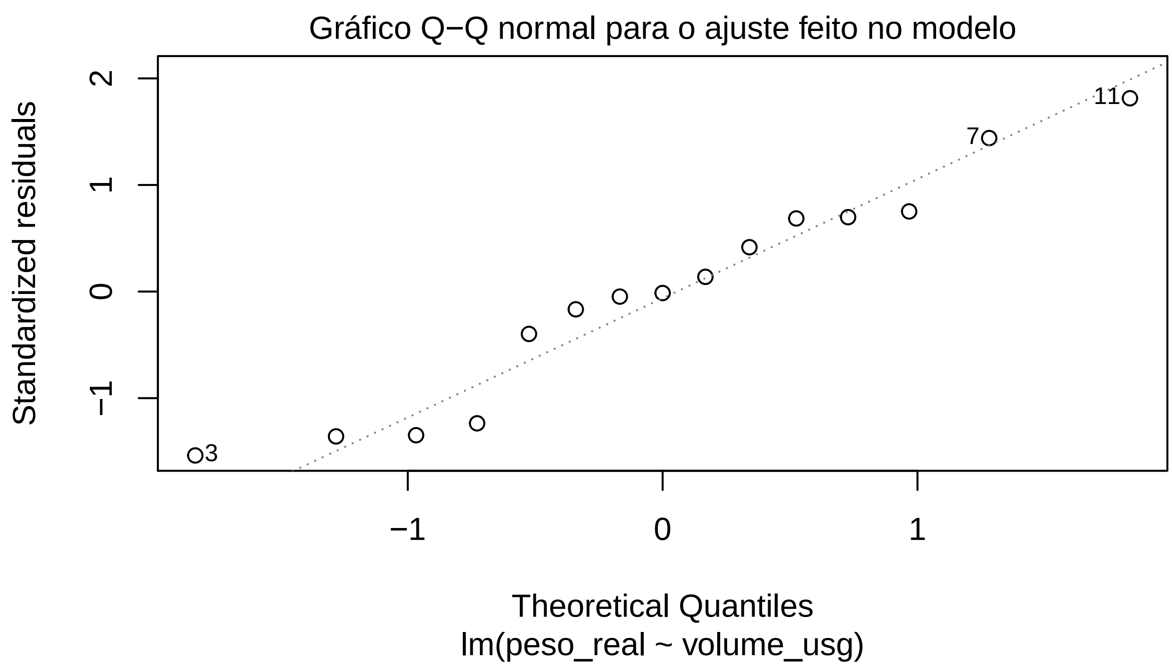
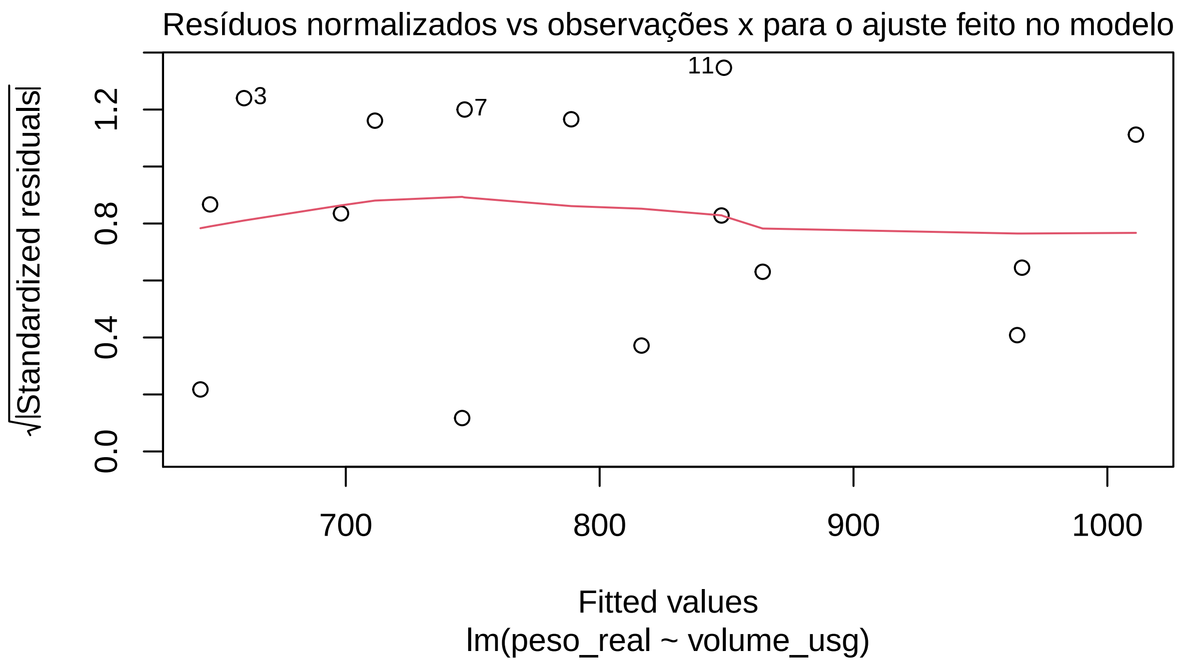
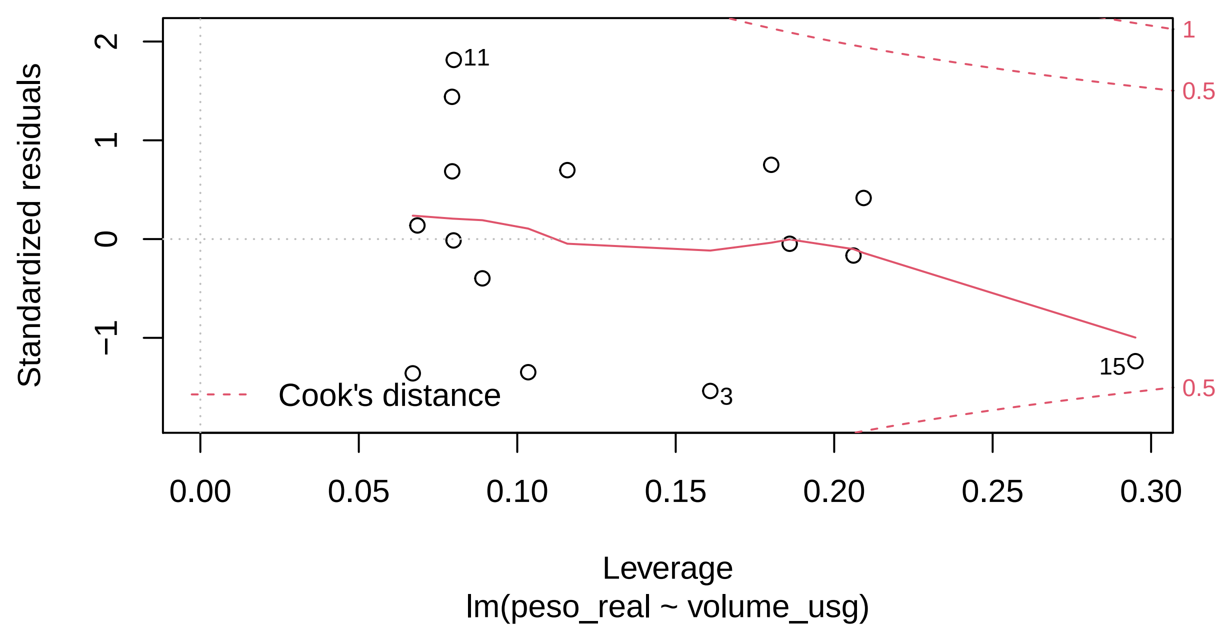


Gráfico Q−Q normal para o ajuste feito no modelo
Standardized residuals
−1
0
1
2
Theoretical Quantiles
lm(peso_real ~ volume_usg)
11
7
3





Observamos uma melhora significativa no valor Rˆ2 após a remoção da observação 15. Os gráficos indicam que os resíduos possuem os valores dentro do esperado. Idealmente, o Rˆ2 deveria estar próximo de 1, mas não está. Dessa forma, podemos concluir que o ajuste do modelo aproxima os dados, mas não estritamente. Assim, espera-se que o intervalo de confiança ao prever o peso real com base no volume seja grande.

iv)

Construindo intervalos de confiança dos parâmetros:

```
confidence_intervals <- confint(ajuste)
rownames(confidence_intervals) <- c("Alpha", "Beta")
kable(confidence_intervals, caption="Intervalos de confiança para o ajuste dos parâmetros do modelo")
```

Tabela 1: Intervalos de confiança para o ajuste dos parâmetros do modelo

	2.5 %	97.5 %
Alpha	-137,11	307,43
Beta	0,66	1,25

v)

A seguir, construiremos a tabela.

```
volumes <- c(600, 700, 800, 900, 1000)
df <- data.frame(volume_usg = volumes)
previsto <- predict(ajuste, df, interval='confidence')
previsto <- data.frame(previsto)
intervalo <- previsto$fit - previsto$lwr
previsto <- cbind(volume_usg = volumes, peso = previsto$fit, intervalo = intervalo)
colnames(previsto) <- c("Volume", "Peso previsto", "Intervalo de confiança de 95%")
kable(previsto, caption="Pesos previstos pelo modelo")
```

Tabela 2: Pesos previstos pelo modelo

Volume	Peso previsto	Intervalo de confiança de 95%
600	658,00	55,77
700	753,48	38,08
800	848,95	39,00
900	944,43	57,63
1.000	1.039,90	82,78

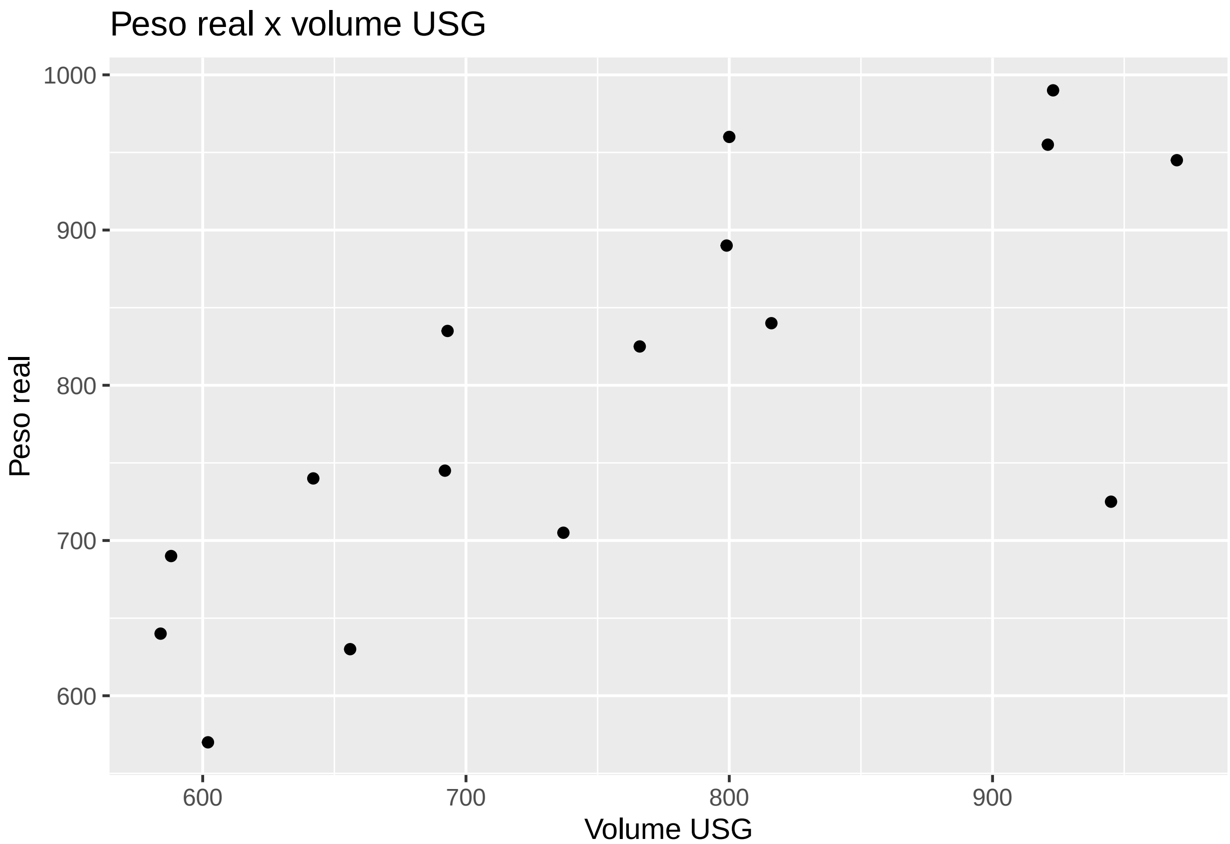
vi)

vi)i)

Novamente, tomaremos o Volume USG como a variável explicativa x e o Peso Real como a variável resposta y. Adotaremos o modelo de regressão linear simples $y_i = \beta x_i + e_i$, onde *beta* é a inclinação da reta e e_i são erros aleatórias não correlacionados.

vi)ii)

```r
dados3 <- data.frame(dados1)
ggplot(dados3, aes(x=volume_usg, y=peso_real)) + geom_point() + labs(title=scatter_title, x=scatter_x, 
```



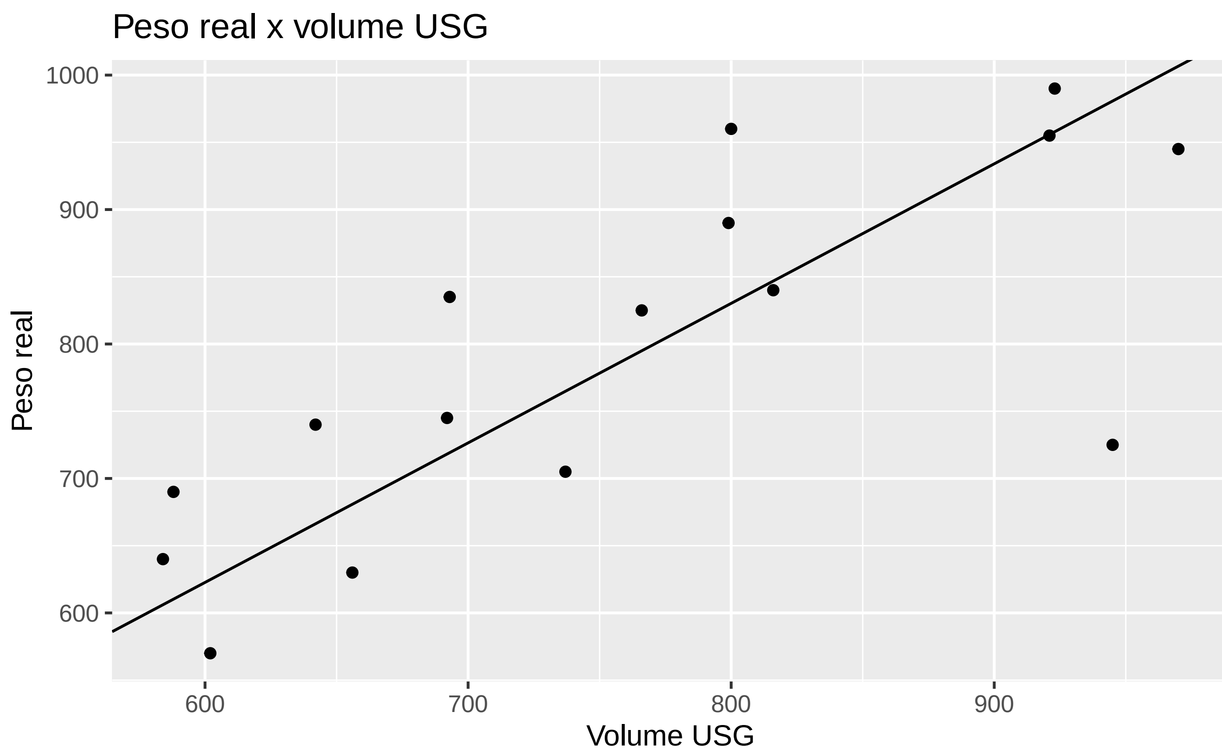
vi)iii)

Realizaremos o ajuste do modelo e mostraremos algumas métricas de qualidade do modelo:

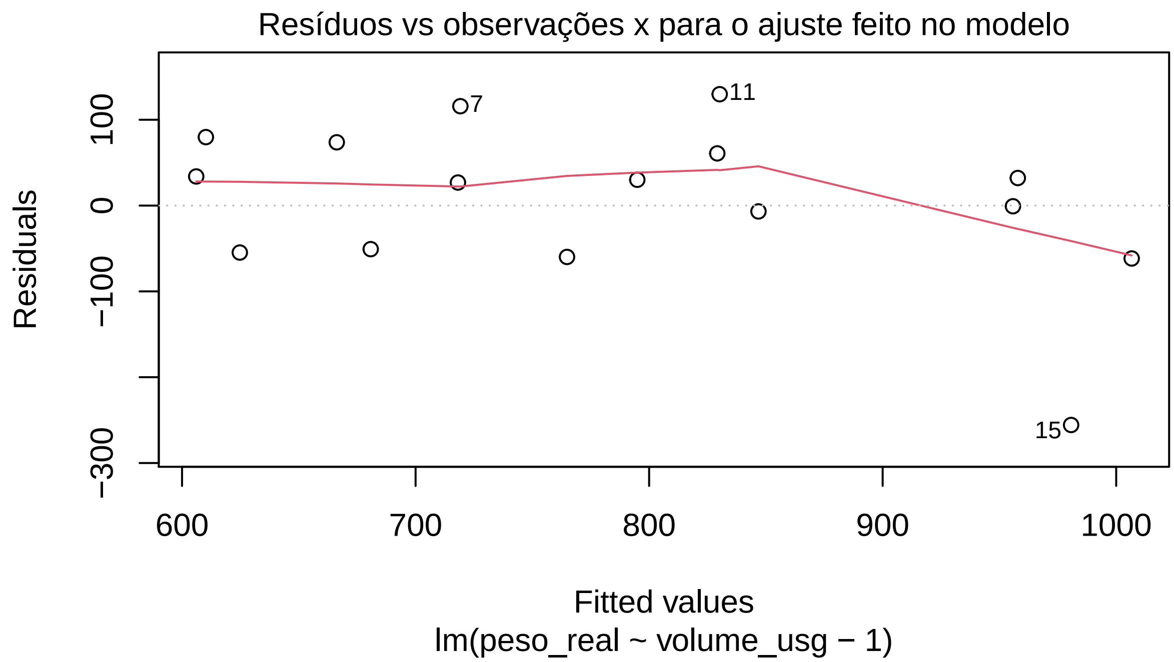
```r
ajustarModelo <- function(dados) {
  # - 1 omite o intercepto
  ajuste <- lm(peso_real ~ volume_usg - 1, data=dados)
  intercept <- 0
  slope <- ajuste$coefficients
  print("O ajuste encontrou o coeficiente:")
  print(paste("Beta:", slope))
  p <- ggplot(dados, aes(x=volume_usg, y=peso_real)) + geom_point() + geom_abline(intercept = intercept
  plot(p)
  print(summary(ajuste))
  plot(ajuste, caption=fit_titles)

  return(ajuste)
}

ajuste <- ajustarModelo(dados3)
```

```
## [1] "O ajuste encontrou o coeficiente:"
```

```
## [1] "Beta: 1.03776957920071"
```




```
##
## Call:
## lm(formula = peso_real ~ volume_usg - 1, data = dados)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -255.69  -51.77   28.47   64.06  129.78
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## volume_usg  1.03777    0.03003   34.56 1.03e-15 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.36 on 15 degrees of freedom
## Multiple R-squared:  0.9876, Adjusted R-squared:  0.9868
## F-statistic:  1194 on 1 and 15 DF,  p-value: 1.026e-15
```


Resíduos vs observações x para o ajuste feito no modelo
Residuals
100
0
−100
−300
600
700
800
900
1000
Fitted values
lm(peso_real ~ volume_usg − 1)
7
11
15

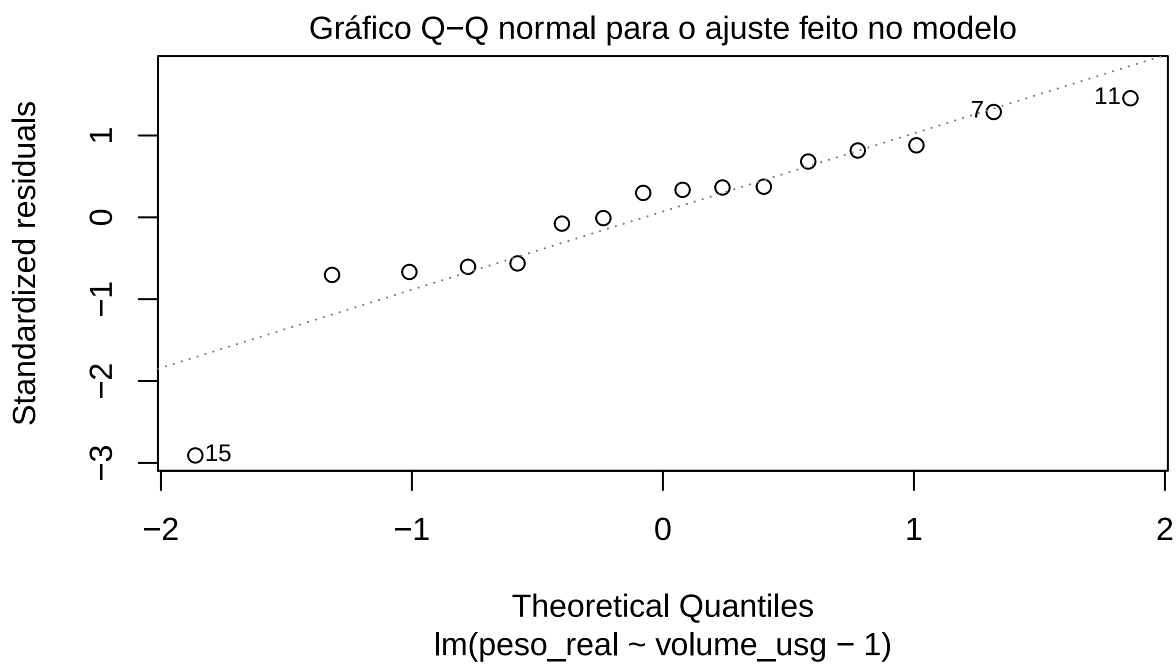
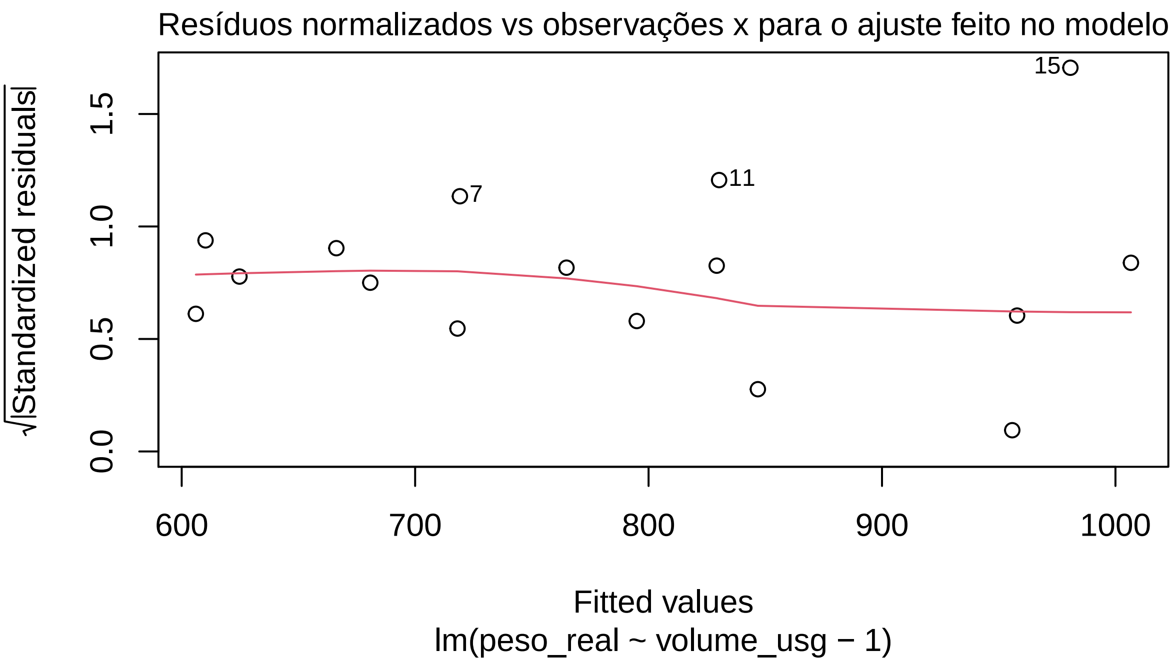
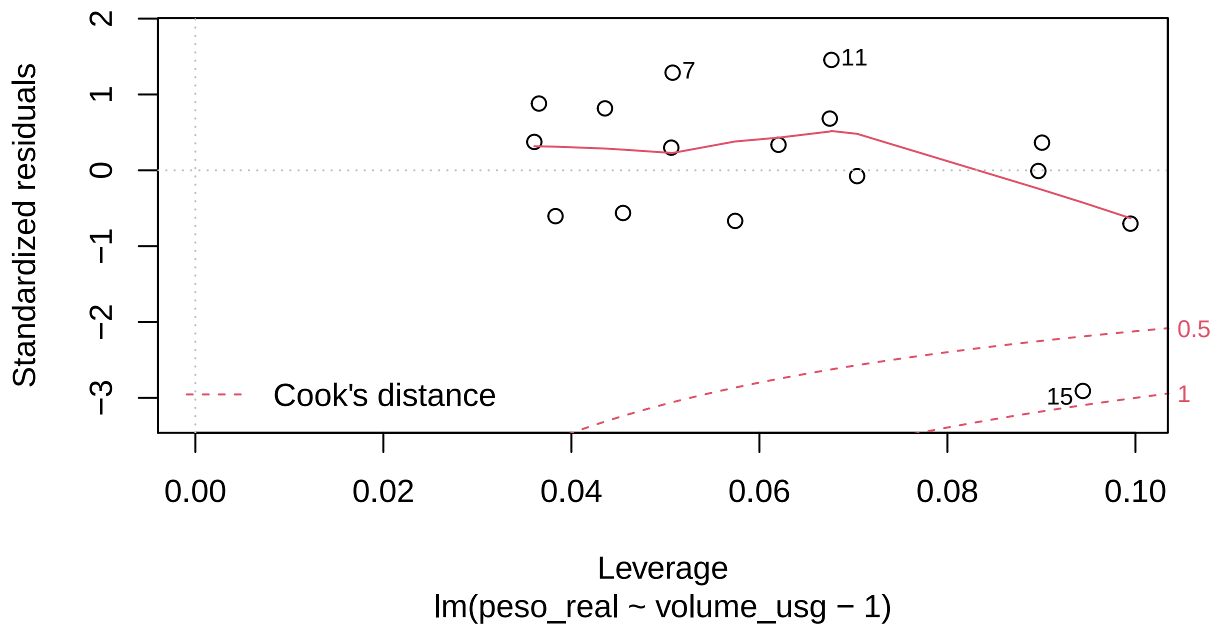


Gráfico Q−Q normal para o ajuste feito no modelo
Standardized residuals
1
0
−1
−2
−3
−2
−1
0
1
2
Theoretical Quantiles
lm(peso_real ~ volume_usg − 1)
15
7
11

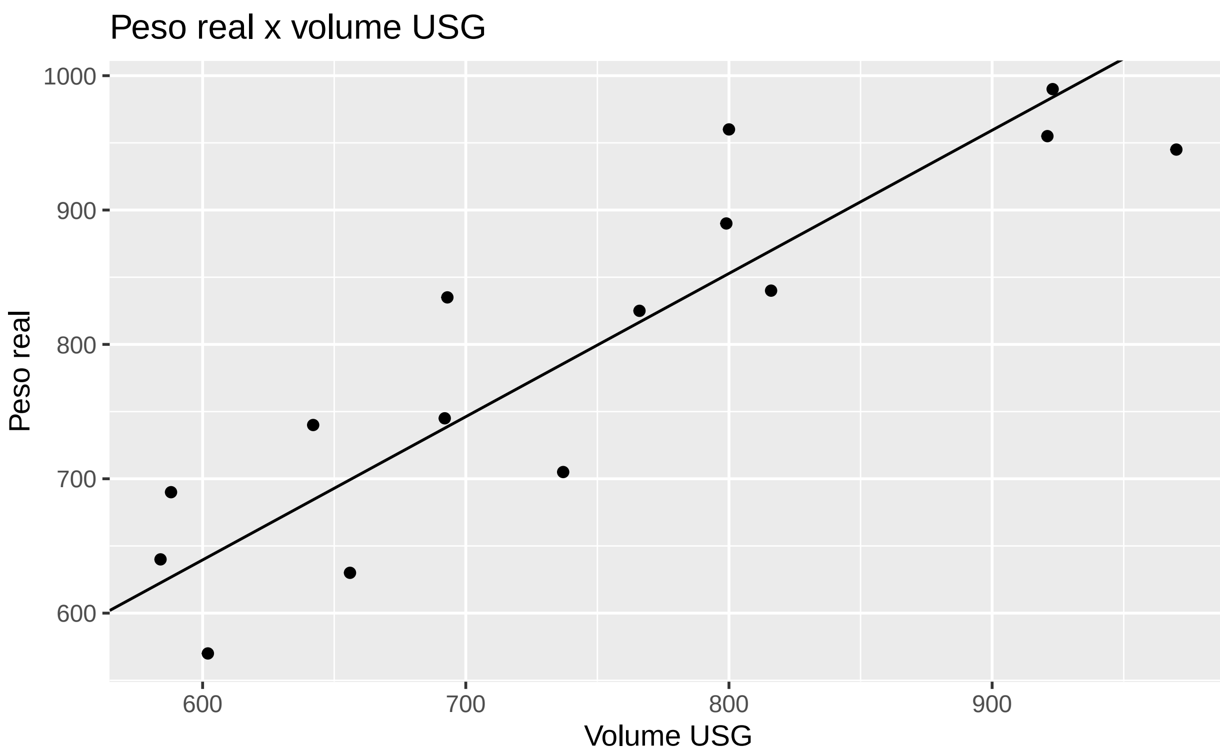




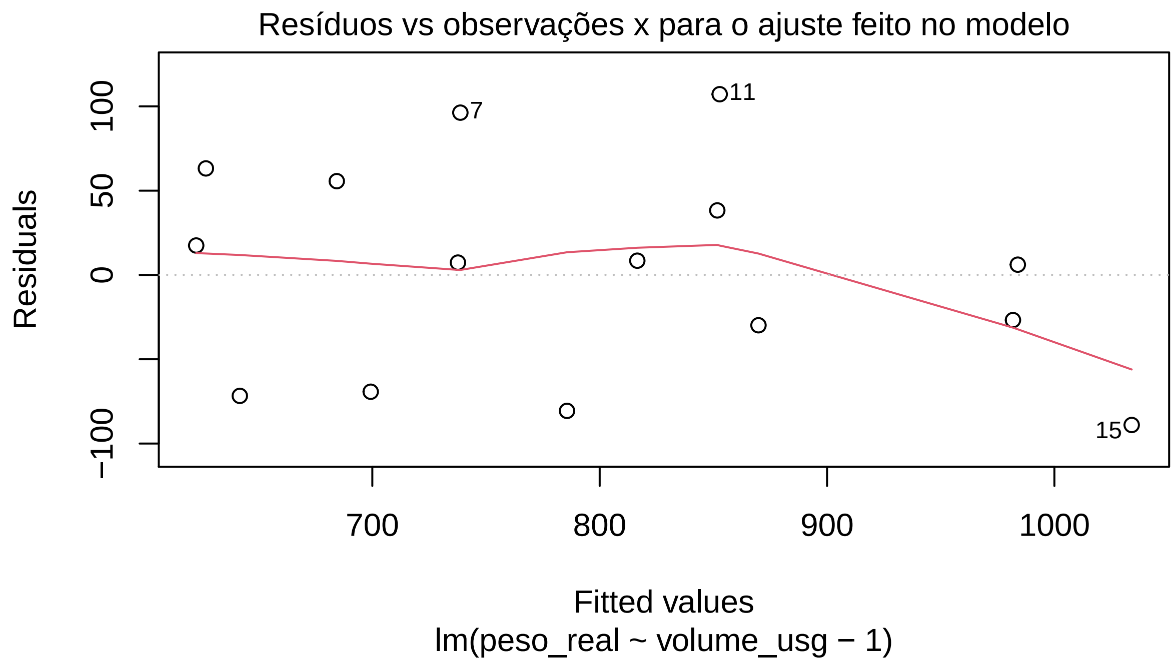
Novamente, os gráficos indicam que a observação 15 é um outlier. Refaremos o ajuste removendo a observação 15.

```r
dados4 <- dados1[-c(15), ]
ajuste <- ajustarModelo(dados4)
```

```
## [1] "O ajuste encontrou o coeficiente:"
## [1] "Beta: 1.06597728783179"
```




```
##
## Call:
## lm(formula = peso_real ~ volume_usg - 1, data = dados)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -88.998 -49.559   7.344  46.963 107.218
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## volume_usg  1.06598    0.02156   49.44   <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.11 on 14 degrees of freedom
## Multiple R-squared:  0.9943, Adjusted R-squared:  0.9939
## F-statistic:  2444 on 1 and 14 DF,  p-value: < 2.2e-16
```


Resíduos vs observações x para o ajuste feito no modelo
Residuals
100
50
0
−100
7
11
15
700
800
900
1000
Fitted values
lm(peso_real ~ volume_usg − 1)

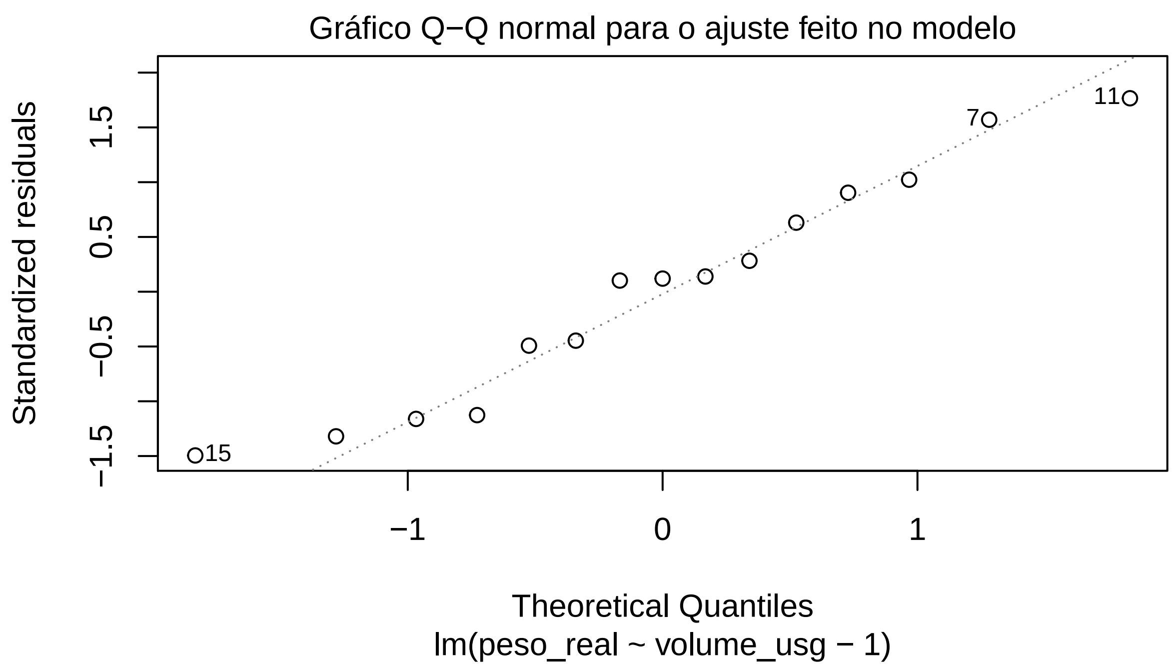
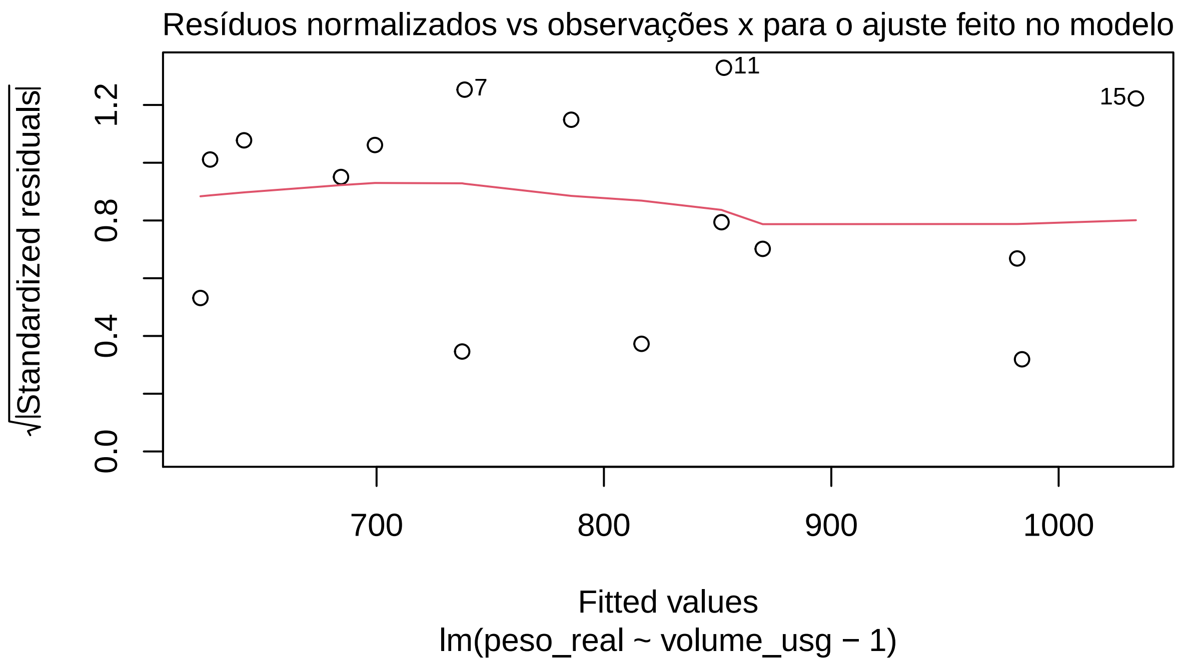
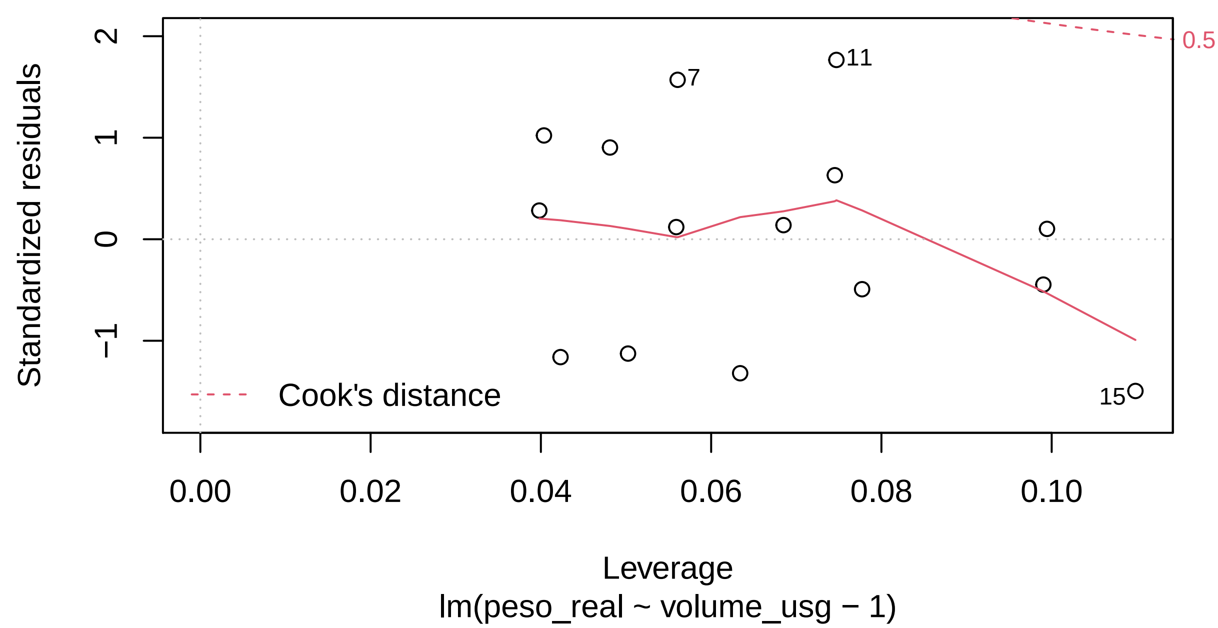


Gráfico Q−Q normal para o ajuste feito no modelo
Standardized residuals
1.5
0.5
−0.5
−1.5
11
7
15
−1
0
1
Theoretical Quantiles
lm(peso_real ~ volume_usg − 1)





As mesmas observações sobre a qualidade do modelo se aplicam. Os gráficos indicam que os resíduos possuem os valores dentro do esperado. Idealmente, o R^2 deveria estar próximo de 1, mas não está. Dessa forma, podemos concluir que o ajuste do modelo aproxima os dados, mas não estritamente. Assim, espera-se que o intervalo de confiança ao prever o peso real com base no volume seja grande.

vi)iv)

Construindo intervalos de confiança dos parâmetros:

```
confidence_intervals <- confint(ajuste)
rownames(confidence_intervals) <- c("Beta")
kable(confidence_intervals, caption="Intervalos de confiança para o ajuste dos parâmetros do modelo")
```

Tabela 3: Intervalos de confiança para o ajuste dos parâmetros do modelo

	2.5 %	97.5 %
Beta	1,02	1,11

vi)v)

A seguir, construiremos a tabela.

```
volumes <- c(600, 700, 800, 900, 1000)
df <- data.frame(volume_usg = volumes)
previsto <- predict(ajuste, df, interval='confidence')
previsto <- data.frame(previsto)
intervalo <- previsto$fit - previsto$lwr
previsto <- cbind(volume_usg = volumes, intervalo=previsto$fit, intervalo = intervalo)
colnames(previsto) <- c("Volume", "Peso previsto", "Intervalo de confiança de 95%")
kable(previsto, caption="Pesos previstos pelo modelo")
```

Tabela 4: Pesos previstos pelo modelo

Volume	Peso previsto	Intervalo de confiança de 95%
600	639,59	27,75
700	746,18	32,37
800	852,78	37,00
900	959,38	41,62
1.000	1.065,98	46,25

vi)vi)

Ambos os modelos satisfazem de forma similar as métricas mostradas na etapa (iii). Entretanto, observa-se na etapa (v) que o segundo modelo apresenta intervalos de confiança menores para suas predições de peso real. Dessa forma, o modelo sem intersecto demonstrou-se mais conveniente. Destacamos que o intervalo de confiança de 97,5% do parâmetro α no primeiro modelo era consideravelmente alto, o que poderia indicar que ele não possuia muita importância no modelo.

Exercício 2

Exercício 3

Exercício 4

Exercício 15

Exercício 16