Estimación del modelo lineal - Ejercicios

Alejandro Keymer

Ejercicios del libro de Faraway

1. (Ejercicio 1 cap. 2 pág. 30)

The dataset teengamb concerns a study of teenage gambling in Britain. Fit a regression model with the expenditure on gambling as the response and the sex, status, income and verbal score as predictors. Present the output.

(a) What percentage of variation in the response is explained by these predictors?

(b) Which observation has the largest (positive) residual? Give the case number.

(c) Compute the mean and median of the residuals.

(d) Compute the correlation of the residuals with the fitted values.

(e) Compute the correlation of the residuals with the income.

(f) For all other predictors held constant, what would be the difference in predicted expenditure on gambling for a male compared to a female?

```
# me he acostumbrado a la gramática de tidyverse, por lo que utilizo esta biblioteca
pacman::p_load(faraway, broom, tidyverse)
```

```
# a) El porcentaje corresponde al valor de R2. en este caso utilizo la función
# glance para mirarlo
model_1 <-
    faraway::teengamb %>%
    lm(gamble ~ sex + status + income + verbal, data = .)

glance(model_1)$adj.r.squared
```

```
## [1] 0.4816495
```

```
# b) El caso cuyo residual es el mas elevado (positivo) es:
augment(model_1) %>%
    rowid_to_column() %>%
    top_n(1, .resid) %>%
    pull(rowid)
```

```
## [1] 24
```

```
# c) la media y mediana son
augment(model_1) %>%
    summarise(
        media = mean(.resid),
        mediana = median(.resid))
```

```
## # A tibble: 1 x 2
##      media mediana
##      <dbl>   <dbl>
## 1 3.62e-16   -1.45
```

```
# d) correlación residual v/s fitted.values
augment(model_1) %>%
```

```
    summarise(
        correlation = cor(.resid, .fitted)
    )
```

```
## # A tibble: 1 x 1
##   correlation
##         <dbl>
## 1    9.25e-17
```

```
# e) correlación residuales v/s income
augment(model_1) %>%
    summarise(
        correlation = cor(.resid, income)
    )
```

```
## # A tibble: 1 x 1
##   correlation
##         <dbl>
## 1    3.90e-17
```

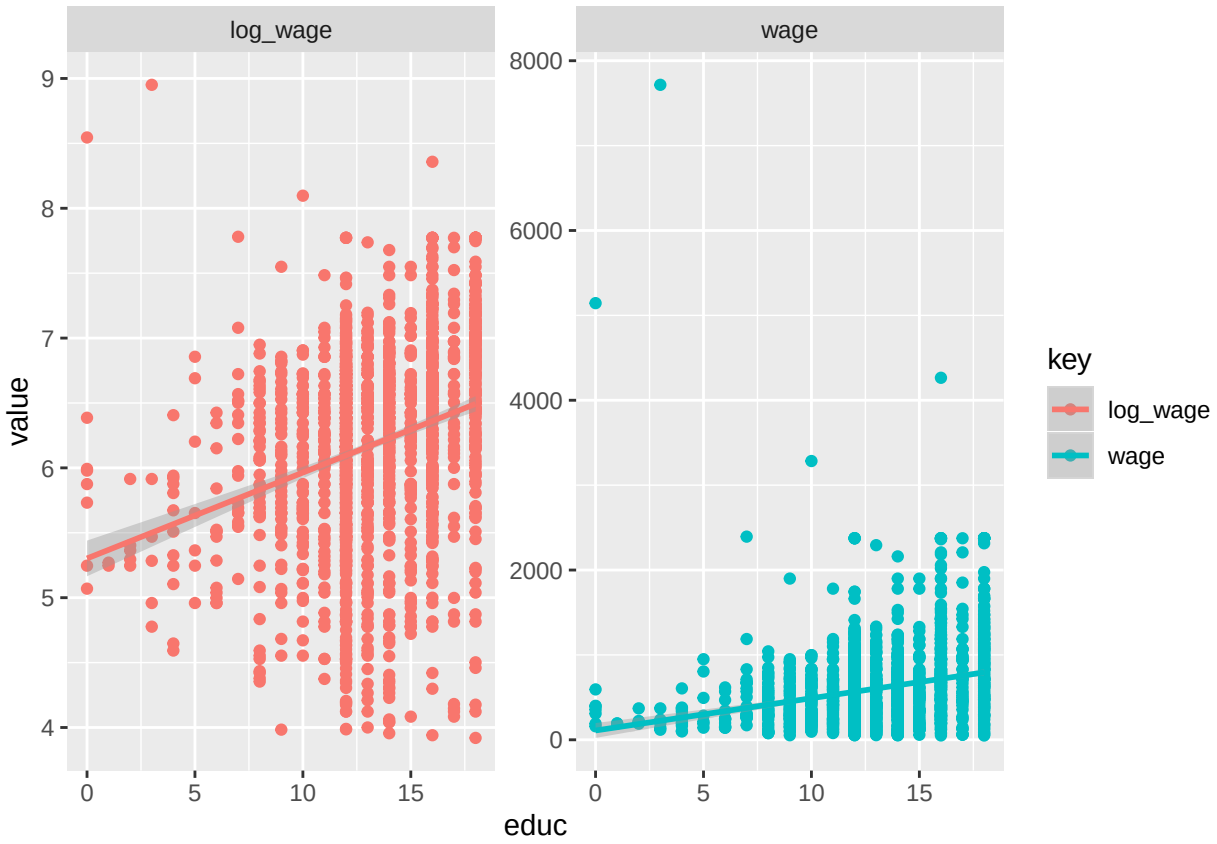
```
# f) Si todos los predictores quedan igual, se pide el poder explicativo de la
# variable género. En este caso el sexo femenino (que se codifica como 1) se
# asocia a una disminución en la media de gastos en apuestas de 25.9.

faraway::teengamb %>%
    lm(gamble ~ sex, data = .) %>%
    tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic    p.value
##   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept)     29.8      5.50      5.42 0.00000228
## 2 sex            -25.9      8.65     -3.00 0.00444
```

2. (Ejercicio 2 cap. 2 pág. 30)

The dataset uswages is drawn as a sample from the Current Population Survey in 1988. Fit a model with weekly wages as the response and years of education and experience as predictors. Report and give a simple interpretation to the regression coefficient for years of education. Now fit the same model but with logged weekly wages. Give an interpretation to the regression coefficient for years of education. Which interpretation is more natural?

```
faraway::uswages %>%
    select(educ,  wage) %>%
    mutate(log_wage = log(wage)) %>%
    gather(key, value, -educ) %>%
    ggplot(aes(educ, value, color = key)) +
    geom_point() +
    geom_smooth(method = "lm") +
    facet_wrap(~ key, scales = "free_y")
```




```
model_2.1 <-
    faraway::uswages %>%
    lm(wage ~ educ + exper, data = .)

# Por cada año de educacion, aumenta el salario en 51 dolares
tidy(model_2.1)
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  -243.      50.7       -4.79 1.78e- 6
## 2 educ           51.2      3.34      15.3  3.93e-50
## 3 exper           9.77     0.751     13.0  2.86e-37
```

```
model_2.2 <-
    faraway::uswages %>%
    lm(log(wage) ~ educ + exper, data = .)

# Por cada año de educación, aumenta el salario en un 9% (beta x 100 %). La
# interpretación del modelo logarítmico es mucho mas natural y comprensible,
# que la estimación en cifras absolutas.

tidy(model_2.2)
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   4.65     0.0784       59.4 0.
```

```
## 2 educ          0.0905   0.00517      17.5 5.04e-64
## 3 exper         0.0181   0.00116      15.6 9.83e-52
```

3. (∗) (Ejercicio 3 cap. 2 pág. 30)

In this question, we investigate the relative merits of methods for computing the coefficients. Generate some artificial data by:

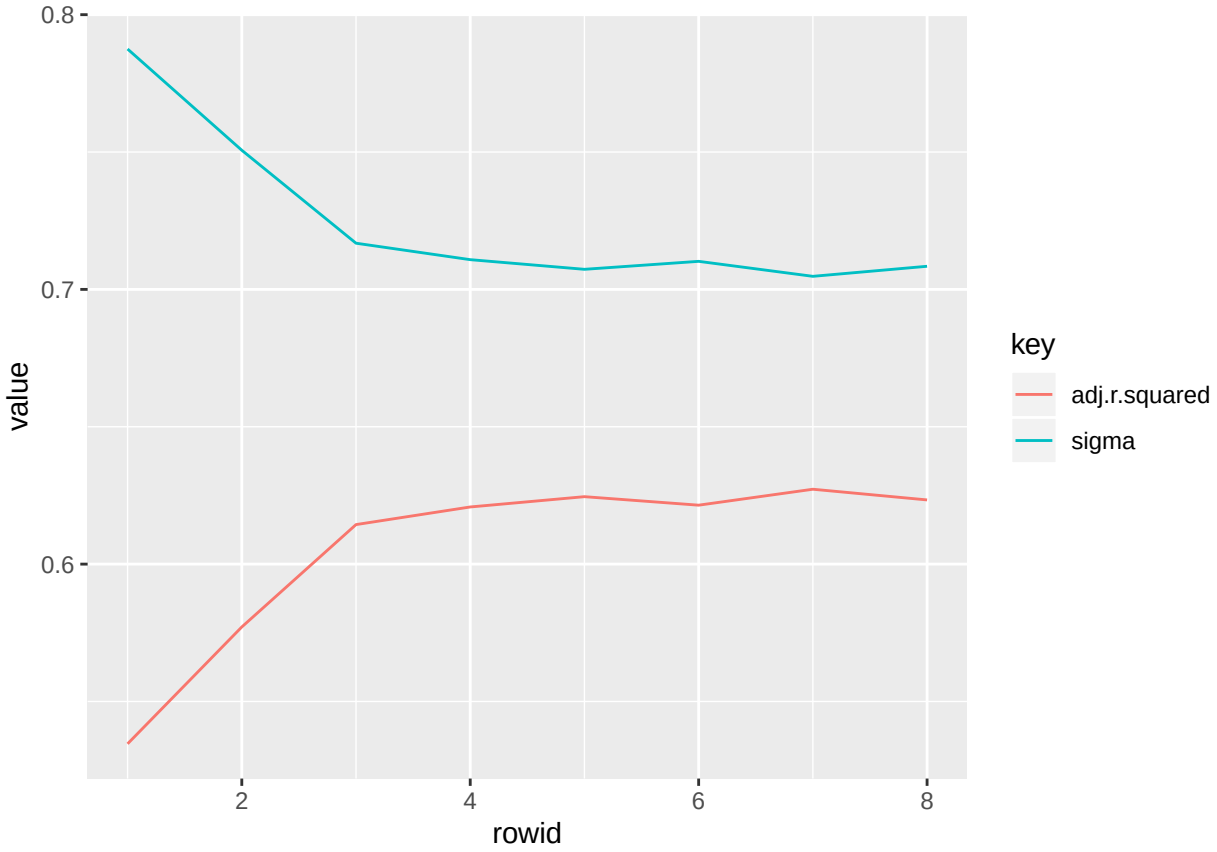
```
> x <- 1:20
> y <- x+rnorm(20)
```

Fit a polynomial in x for predicting y. Compute ˆ in two ways — by lm() and by using the direct calculation described in the chapter. At what degree of polynomial does the direct calculation method fail? (Note the need for the I() function in fitting the polynomial, that is, lm(y ~ x + I(xˆ2)).

4. (Ejercicio 4 cap. 2 pág. 30)

The dataset prostate comes from a study on 97 men with prostate cancer who were due to receive a radical prostatectomy. Fit a model with lpsa as the response and lcavol as the predictor. Record the residual standard error and the R2. Now add lweight, svi, lbph, age, lcp, pgg45 and gleason to the model one at a time. For each model record the residual standard error and the R2. Plot the trends in these two statistics.

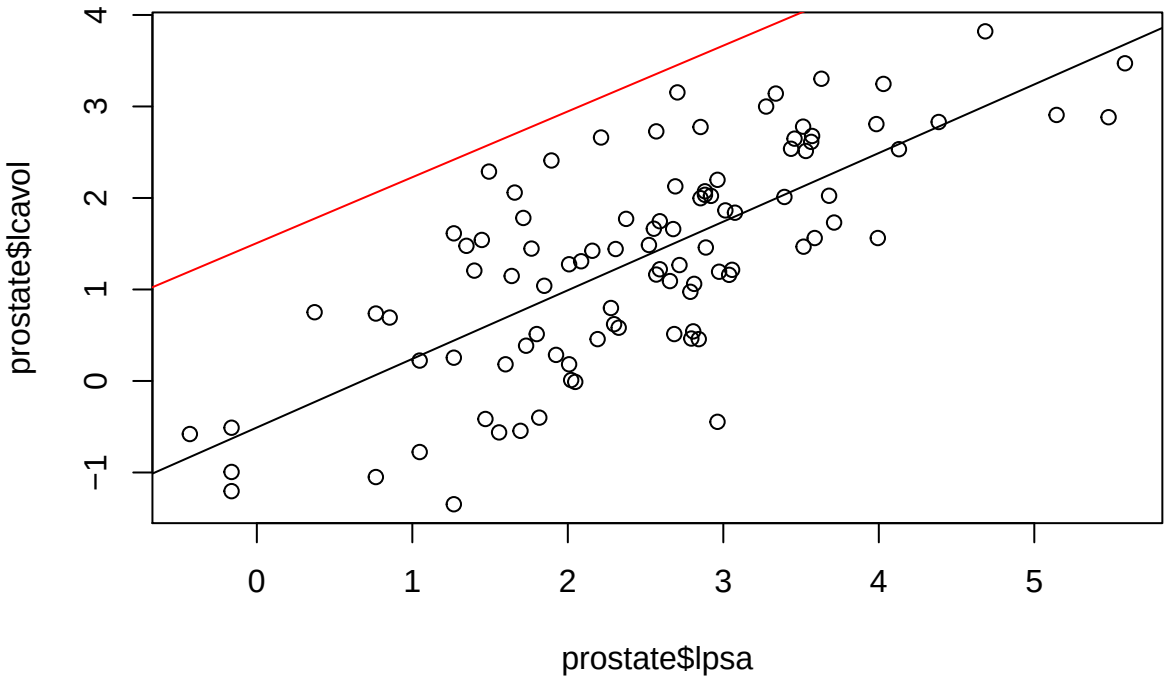
```
# esto se puede hacer como una iteracion con purrr...?
model_1 <- prostate %>%
    lm(lpsa ~ lcavol, data = .) %>% glance
model_2 <- prostate %>%
    lm(lpsa ~ lcavol + lweight, data = .) %>% glance
model_3 <- prostate %>%
    lm(lpsa ~ lcavol + lweight + svi, data = .) %>% glance
model_4 <- prostate %>%
    lm(lpsa ~ lcavol + lweight + svi + lbph, data = .) %>% glance
model_5 <- prostate %>%
    lm(lpsa ~ lcavol + lweight + svi + lbph + age, data = .) %>% glance
model_6 <- prostate %>%
    lm(lpsa ~ lcavol + lweight + svi + lbph + age + lcp, data = .) %>% glance
model_7 <- prostate %>%
    lm(lpsa ~ lcavol + lweight + svi + lbph + age + lcp + pgg45, data = .) %>% glance
model_8 <- prostate %>%
    lm(lpsa ~ lcavol + lweight + svi + lbph + age + lcp + pgg45 + gleason, data = .) %>%
    glance

# plot
bind_rows(model_1, model_2, model_3, model_4, model_5, model_6, model_7, model_8) %>%
    select(adj.r.squared, sigma) %>%
    rowid_to_column() %>%
    gather(key, value, -rowid) %>%
    ggplot(aes(rowid, value, color = key)) +
    geom_line()
```



5. (Ejercicio 5 cap. 2 pág. 30)

Using the prostate data, plot lpsa against lcavol. Fit the regressions of lpsa on lcavol and lcavol on lpsa. Display both regression lines on the plot. At what point do the two lines intersect?

```
model_1 <- lm(lcavol ~ lpsa, data = prostate)
model_2 <- lm(lpsa ~ lcavol, data = prostate)

plot(prostate$lpsa, prostate$lcavol)
abline(model_1)
abline(model_2, col =  "red")
```




```
# calcular la intersección:
# para calcular la intersección utilizo la
# aproximación de solución de sistemas de variables con matrices que se propone en:
# https://stackoverflow.com/questions/7114703/finding-where-two-linear-fits-intersect-in-r

cm <- rbind(coef(model_1),coef(model_2)) # matriz de coeficientes

# intersección
c(-solve(cbind(cm[,2],-1)) %*% cm[,1])
```

```
## [1] 65.88061 48.89655
```

6. (Ejercicio 6 cap. 2 pág. 30)

Thirty samples of cheddar cheese were analyzed for their content of acetic acid, hydrogen sulfide and lactic acid. Each sample was tasted and scored by a panel of judges and the average taste score produced. Use the cheddar data to answer the following:

(a) Fit a regression model with taste as the response and the three chemical contents as predictors. Report the values of the regression coefficients.

(b) Compute the correlation between the fitted values and the response. Square it. Identify where this value appears in the regression output.

(c) Fit the same regression model but without an intercept term. What is the value of R2 reported in the output? Compute a more reasonable measure of the good- ness of fit for this example.

(d) Compute the regression coefficients from the original fit using the QR decomposition showing your R code.

```
mod <-
   cheddar %>%
    lm(taste ~ Acetic + H2S + Lactic, data = .)

# a)
```

```r
tidy(mod)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)  -28.9       19.7    -1.46   0.155
## 2 Acetic         0.328      4.46    0.0735 0.942
## 3 H2S            3.91       1.25    3.13   0.00425
## 4 Lactic        19.7        8.63    2.28   0.0311
```

```r
# b) el valor calculado corresponde a R2
augment(mod) %>%
    summarise(cor_2 = cor(.fitted, taste)^2) %>%
    pull(cor_2)
```

```
## [1] 0.6517747
```

```r
# c) En este caso podemos utilizar el test de Likelihood ratio, para valorar el
# ajuste de los diferentes modelos.

mod_0 <-
  lm(taste ~ 1, data = cheddar)
mod <-
  lm(taste ~ Acetic + H2S + Lactic, data = cheddar)
mod_alt <-
  lm(taste ~ Acetic + H2S + Lactic -1, data =cheddar)


lmtest::lrtest(mod_0, mod, mod_alt)
```

```
## Likelihood ratio test
##
## Model 1: taste ~ 1
## Model 2: taste ~ Acetic + H2S + Lactic
## Model 3: taste ~ Acetic + H2S + Lactic - 1
##   #Df  LogLik Df   Chisq Pr(>Chisq)
## 1   2 -125.71
## 2   5 -109.89  3 31.6472  6.211e-07 ***
## 3   4 -111.08 -1  2.3739     0.1234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# d) Esta permitido utilizar `qr` ?
qr_x <-
cheddar %>%
  mutate(intercept = 1) %>%
  select(intercept, Acetic, H2S, Lactic) %>%
  as.matrix() %>%
  qr()

vec_y <-
  cheddar %>%
  pull(taste)

# coeficientes
backsolve(qr_x$qr, qr.qty(qr_x, vec_y))
```

```
## [1] -28.8767696   0.3277413   3.9118411  19.6705434
```

7. (Ejercicio 7 cap. 2 pág. 31)

An experiment was conducted to determine the effect of four factors on the resistivity of a semiconductor wafer. The data is found in wafer where each of the four factors is coded as − or + depending on whether the low or the high setting for that factor was used. Fit the linear model resist ~ x1 + x2 + x3 + x4.

(a) Extract the X matrix using the model.matrix function. Examine this to determine how the low and high levels have been coded in the model.

(b) Compute the correlation in the X matrix. Why are there some missing values in the matrix?

(c) What difference in resistance is expected when moving from the low to the high level of x1?

(d) Refit the model without x4 and examine the regression coefficients and standard errors? What stayed the the same as the original fit and what changed?

(e) Explain how the change in the regression coefficients is related to the correlation matrix of X.

```
mod <-
wafer %>%
    lm(resist ~ x1 + x2 + x3 + x4, data =.)

# a) Los niveles se han codificado de manera binaria.
model.matrix(mod)
```

```
##    (Intercept) x1+ x2+ x3+ x4+
## 1            1   0   0   0   0
## 2            1   1   0   0   0
## 3            1   0   1   0   0
## 4            1   1   1   0   0
## 5            1   0   0   1   0
## 6            1   1   0   1   0
## 7            1   0   1   1   0
## 8            1   1   1   1   0
## 9            1   0   0   0   1
## 10           1   1   0   0   1
## 11           1   0   1   0   1
## 12           1   1   1   0   1
## 13           1   0   0   1   1
## 14           1   1   0   1   1
## 15           1   0   1   1   1
## 16           1   1   1   1   1
## attr(,"assign")
## [1] 0 1 2 3 4
## attr(,"contrasts")
## attr(,"contrasts")$x1
## [1] "contr.treatment"
##
## attr(,"contrasts")$x2
## [1] "contr.treatment"
##
## attr(,"contrasts")$x3
## [1] "contr.treatment"
##
```

```
## attr(,"contrasts")$x4
## [1] "contr.treatment"
```

```
# b) Si la varianza es 0 (como aparece en el mensaje) la correlación es NA, como es el caso del interce
model.matrix(mod) %>%
    cor()
```

```
## Warning in cor(.): the standard deviation is zero
```

```
##             (Intercept) x1+ x2+ x3+ x4+
## (Intercept)           1  NA  NA  NA  NA
## x1+                  NA   1   0   0   0
## x2+                  NA   0   1   0   0
## x3+                  NA   0   0   1   0
## x4+                  NA   0   0   0   1
```

```
# c) Un aumento en la resistencia de 25.76
summary(mod)
```

```
##
## Call:
## lm(formula = resist ~ x1 + x2 + x3 + x4, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.381 -17.119   4.825  16.644  33.769
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   236.78      14.77  16.032 5.65e-09 ***
## x1+            25.76      13.21   1.950 0.077085 .
## x2+           -69.89      13.21  -5.291 0.000256 ***
## x3+            43.59      13.21   3.300 0.007083 **
## x4+           -14.49      13.21  -1.097 0.296193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.42 on 11 degrees of freedom
## Multiple R-squared:  0.7996, Adjusted R-squared:  0.7267
## F-statistic: 10.97 on 4 and 11 DF,  p-value: 0.0007815
```

```
# d) Los coeficientes quedan igual (salvo el intercepto). El error estándar
# aumenta un poco y cambian los valores de las t y las p y el valor de r2
mod_2 <- wafer %>%
    lm(resist ~ x1 + x2 + x3, data = .)

tidy(mod)
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic       p.value
##   <chr>          <dbl>     <dbl>     <dbl>         <dbl>
## 1 (Intercept)    237.       14.8     16.0  0.00000000565
## 2 x1+             25.8      13.2      1.95 0.0771
## 3 x2+            -69.9      13.2     -5.29 0.000256
## 4 x3+             43.6      13.2      3.30 0.00708
## 5 x4+            -14.5      13.2     -1.10 0.296
```

```r
tidy(mod_2)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    230.       13.3     17.2  7.88e-10
## 2 x1+             25.8      13.3      1.93 7.70e- 2
## 3 x2+            -69.9      13.3     -5.25 2.06e- 4
## 4 x3+             43.6      13.3      3.27 6.68e- 3
```

```r
glance(mod)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>  <dbl> <dbl> <dbl>
## 1     0.800         0.727  26.4      11.0 7.81e-4     5  -72.1  156.  161.
## # ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```r
glance(mod_2)
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>  <dbl> <dbl> <dbl>
## 1     0.778         0.722  26.6      14.0 3.19e-4     4  -72.9  156.  160.
## # ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```r
# e) La correlación 0 entre coeficientes los hace independientes.
```

8. (∗) (Ejercicio 8 cap. 2 pág. 31)

Ejercicios del libro de Carmona

1. (Ejercicio 2.1 del Capítulo 2 página 41)

Una variable Y toma los valores y1, y2 y y3 en función de otra variable X con los valores x1, x2 y x3. Determinar cuales de los siguientes modelos son lineales y encontrar, en su caso, la matriz de diseño para x1 = 1, x2 = 2 y x3 = 3.

a) yi = 0 + 1xi + 2(x^2i − 1) + ei

b) yi = 0 + 1xi + 2e^xi + ei

c) yi = 1xi(2tang(xi)) + ei

```r
x_vec <- c(1,2,3)

# a)
a_mat <- function(x) c(1, x, x^2 -1)
sapply(x_vec, a_mat) %>% t()
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    0
## [2,]    1    2    3
## [3,]    1    3    8
```

```r
# b)
b_mat <- function(x) c(1, x, exp(1)^x)
sapply(x_vec, b_mat) %>% t()
```

```
##      [,1] [,2]      [,3]
## [1,]    1    1  2.718282
## [2,]    1    2  7.389056
## [3,]    1    3 20.085537
```

```
# c) No lineal
```

2. (Ejercicio 2.4 del Capítulo 2 página 42)

Cuatro objetos cuyos pesos exactos son 1, 2, 3 y 4 han sido pesados en una balanza de platillos de acuerdo con el siguiente esquema: Hallar las estimaciones de cada i y de la varianza del error.

```
dis_mat <-
  matrix(c(
    1,1,1,1,1,1,
    1,-1,0,0,0,1,
    1,1,0,0,1,-1,
    1,1,1,-1,1,1,
    9.2,8.3,5.4,-1.6,8.7,3.5), ncol = 5)

y_vec <- dis_mat[,5]
qr_x <- dis_mat[,1:4] %>% qr()


# coeficientes con formula
backsolve(qr_x$qr, qr.qty(qr_x, y_vec))
```

```
## [1] 2.0685714 0.5342857 2.9400000 3.6685714
```

```
# modelo con lm
as.tibble(dis_mat) %>%
  select(-V1) %>%
  lm(V5 ~ V2+V3+V4, data = .) %>%
  summary()
```

```
## Warning: `as.tibble()` is deprecated, use `as_tibble()` (but mind the new semantics).
## This warning is displayed once per session.
```

```
##
## Call:
## lm(formula = V5 ~ V2 + V3 + V4, data = .)
##
## Residuals:
##          1          2          3          4          5          6
## -1.143e-02  1.571e-01 -3.371e-01  1.344e-17  2.286e-02  1.686e-01
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0686     0.1658  12.473  0.00637 **
## V2            0.5343     0.1956   2.731  0.11199
## V3            2.9400     0.1830  16.066  0.00385 **
## V4            3.6686     0.1658  22.120  0.00204 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2893 on 2 degrees of freedom
## Multiple R-squared:  0.9981, Adjusted R-squared:  0.9951
```

```
## F-statistic: 342.4 on 3 and 2 DF,  p-value: 0.002914
```

3. (∗) (Ejercicio 3.2 del Capítulo 3 página 54)

4. (∗) (Ejercicio 3.7 del Capítulo 3 página 55)

5. (∗) (Ejercicio 3.8 del Capítulo 3 página 56)

6. (∗) (Ejercicio 3.10 del Capítulo 3 página 56)