Estimación del modelo lineal ejercicios

Laura Sudupe Medinilla

1/3/2021

Ejercicios faraway

1.

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.

```
#ajustamos los datos al modelo
lm <- lm(gamble ~ sex + status + income + verbal, data=teengamb)
lmsum <- summary(lm)
lmsum
```

```
##
## Call:
## lm(formula = gamble ~ sex + status + income + verbal, data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.082 -11.320  -1.451   9.452  94.252
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.55565   17.19680   1.312   0.1968
## sex         -22.11833    8.21111  -2.694   0.0101 *
## status        0.05223    0.28111   0.186   0.8535
## income        4.96198    1.02539   4.839 1.79e-05 ***
## verbal       -2.95949    2.17215  -1.362   0.1803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.69 on 42 degrees of freedom
## Multiple R-squared:  0.5267, Adjusted R-squared:  0.4816
## F-statistic: 11.69 on 4 and 42 DF,  p-value: 1.815e-06
```

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

```
#El coeficiente de determinación R^2 es igual al cuadrado de la correlación del
#coeficiente. Cuando se expresa en porcentaje, R^2 representa el porcentaje de
#variación en la variable dependiente y puede ser explicado mediante la
```

```r
#variación en la variable independiente x mediante la linea de regreión.

(lmsum$r.squared)*100
```

```
## [1] 52.67234
```

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

number.

```r
residuos <- lmsum$residuals
max(residuos)      #valor residuo maximo
```

```
## [1] 94.25222
```

```r
which.max(residuos) #observacion con el maximo valor
```

```
## 24
## 24
```

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

```r
#Han de dar valores entorno al cero, el hecho de que no den es debido a errores
#de calculo
mean(residuos)
```

```
## [1] -3.065293e-17
```

```r
median(residuos)
```

```
## [1] -1.451392
```

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

```r
cor(fitted(lm), resid(lm))
```

```
## [1] -1.070659e-16
```

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

```r
cor(resid(lm), teengamb$income)
```

```
## [1] -7.242382e-17
```

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

```r
lmsum
```

```
##
## Call:
## lm(formula = gamble ~ sex + status + income + verbal, data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.082 -11.320  -1.451   9.452  94.252
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.55565   17.19680   1.312   0.1968
## sex         -22.11833    8.21111  -2.694   0.0101 *
## status        0.05223    0.28111   0.186   0.8535
## income        4.96198    1.02539   4.839 1.79e-05 ***
## verbal       -2.95949    2.17215  -1.362   0.1803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.69 on 42 degrees of freedom
## Multiple R-squared:  0.5267, Adjusted R-squared:  0.4816
## F-statistic: 11.69 on 4 and 42 DF,  p-value: 1.815e-06
```

```r
lm$coefficients["sex"]
```

```
##       sex
## -22.11833
```

Estos datos nos indican la pendiente de la recta, por lo tanto, la diferencia entre "expenditure on gambling in punds per year" es 22.12 "pounds" distinta para los hombres que para las mujeres. Menor para las mujeres (sex=1)

2.

The dataset uswages is drawn as a sample from the Current Population Survey in 1988. a) Fit a model with weekly wages as the response and years of education and experience as predictors.

```r
lm2 <- lm(wage ~ educ + exper, data = uswages)
summary(lm2)
```

```
##
## Call:
## lm(formula = wage ~ educ + exper, data = uswages)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1018.2  -237.9   -50.9   149.9  7228.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -242.7994    50.6816  -4.791 1.78e-06 ***
## educ          51.1753     3.3419  15.313  < 2e-16 ***
## exper          9.7748     0.7506  13.023  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 427.9 on 1997 degrees of freedom
## Multiple R-squared:  0.1351, Adjusted R-squared:  0.1343
## F-statistic:   156 on 2 and 1997 DF,  p-value: < 2.2e-16
```

b) Report and give a simple interpretation to the regression coefficient for years of education. Now fit the same model but with logged weekly wages.

```r
lm2$coefficients
```

```
## (Intercept)        educ       exper
## -242.799412   51.175268    9.774767
```

El ajuste no es muy bueno, tenemos un R^2 bajo y un error estandar elevado.

Los coeficientes nos indican que el sueldo (wage) depende de la educación recibida y la experiencia. La relación es de 51.18 unidades mas de educación por cada wage y 9.77 unidades mas de experiencia por cada wage. Esto quiere decir que hay un aumento de 51.18 unidades en el salario por cada año de educación y un aumento de 9.77 unidades en el salario por cada año de experiencia.

$$wage = -242.8 + 51.17educ + 9.77expe$$

```r
#modificamos los datos de waeg logaritmicamente
lmlog2 <- lm(log(wage) ~ educ + exper, data = uswages)
summary(lmlog2)
```

```
##
## Call:
## lm(formula = log(wage) ~ educ + exper, data = uswages)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7533 -0.3495  0.1068  0.4381  3.5699
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.650319   0.078354   59.35   <2e-16 ***
## educ        0.090506   0.005167   17.52   <2e-16 ***
## exper       0.018079   0.001160   15.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6615 on 1997 degrees of freedom
## Multiple R-squared:  0.1749, Adjusted R-squared:  0.174
## F-statistic: 211.6 on 2 and 1997 DF,  p-value: < 2.2e-16
```

```
lmlog2$coefficients
```

```
## (Intercept)        educ       exper
##  4.65031905  0.09050628  0.01807855
```

Vemos que R^2 ha aumentado ligeramente y el error estandar a bajado considerablemente.

La relación entre los coeficientes es la siguiente

$$\log(wage) = 4.65 + 0.09educ + 0.02expe$$

c)Give an interpretation to the regression coefficient for years of education. Which interpretation is more natural? No entiendo lo que se me pide

4.

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

```
lm4 <- lm(lpsa ~ lcavol, data=prostate)
summary(lm4)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.67625 -0.41648  0.09859  0.50709  1.89673
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50730    0.12194   12.36   <2e-16 ***
## lcavol       0.71932    0.06819   10.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7875 on 95 degrees of freedom
## Multiple R-squared:  0.5394, Adjusted R-squared:  0.5346
## F-statistic: 111.3 on 1 and 95 DF,  p-value: < 2.2e-16
```

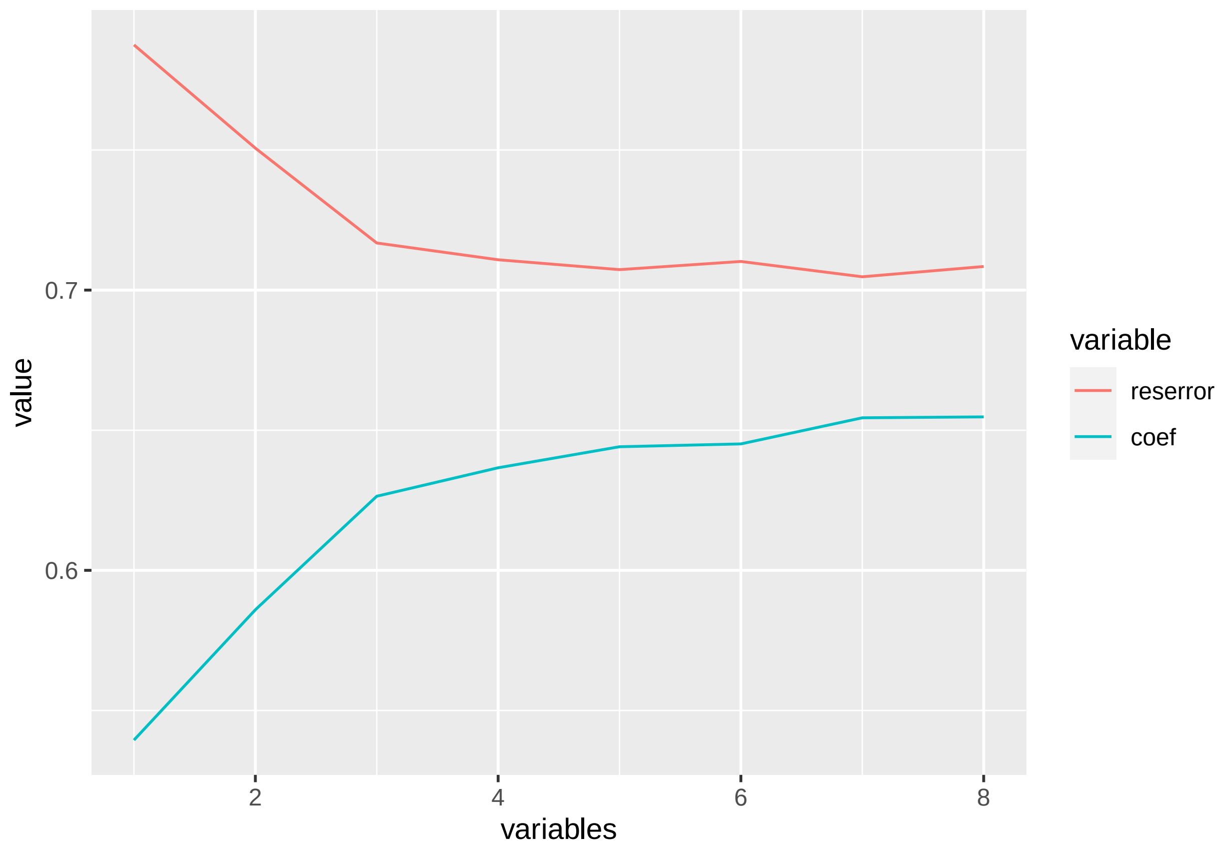
```
#Error estandar de los residups
res4.1 <- summary(lm4)$sigma

#R^2, coeficiente de determinación
coef4.1 <- summary(lm4)$r.squared
```

. 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

```r
lm4.2 <- update(lm4,. ~. + lweight)
res4.2 <- summary(lm4.2)$sigma
coef4.2 <- summary(lm4.2)$r.squared

lm4.3 <- update(lm4.2,. ~. + svi)
res4.3 <- summary(lm4.3)$sigma
coef4.3 <- summary(lm4.3)$r.squared

lm4.4 <- update(lm4.3,. ~. + lbph)
res4.4 <- summary(lm4.4)$sigma
coef4.4 <- summary(lm4.4)$r.squared

lm4.5 <- update(lm4.4,. ~. + age)
res4.5 <- summary(lm4.5)$sigma
coef4.5 <- summary(lm4.5)$r.squared

lm4.6 <- update(lm4.5,. ~. + lcp)
res4.6 <- summary(lm4.6)$sigma
coef4.6 <- summary(lm4.6)$r.squared

lm4.7 <- update(lm4.6,. ~. + pgg45)
res4.7 <- summary(lm4.7)$sigma
coef4.7 <- summary(lm4.7)$r.squared

lm4.8 <- update(lm4.7,. ~. + gleason)
res4.8 <- summary(lm4.8)$sigma
coef4.8 <- summary(lm4.8)$r.squared


reserror <- c(res4.1, res4.2, res4.3, res4.4, res4.5, res4.6, res4.7, res4.8)
coef <- c(coef4.1, coef4.2, coef4.3, coef4.4, coef4.5, coef4.6, coef4.7, coef4.8)
variables <- (1:8)
```

. Plot the trends in these two statistics.

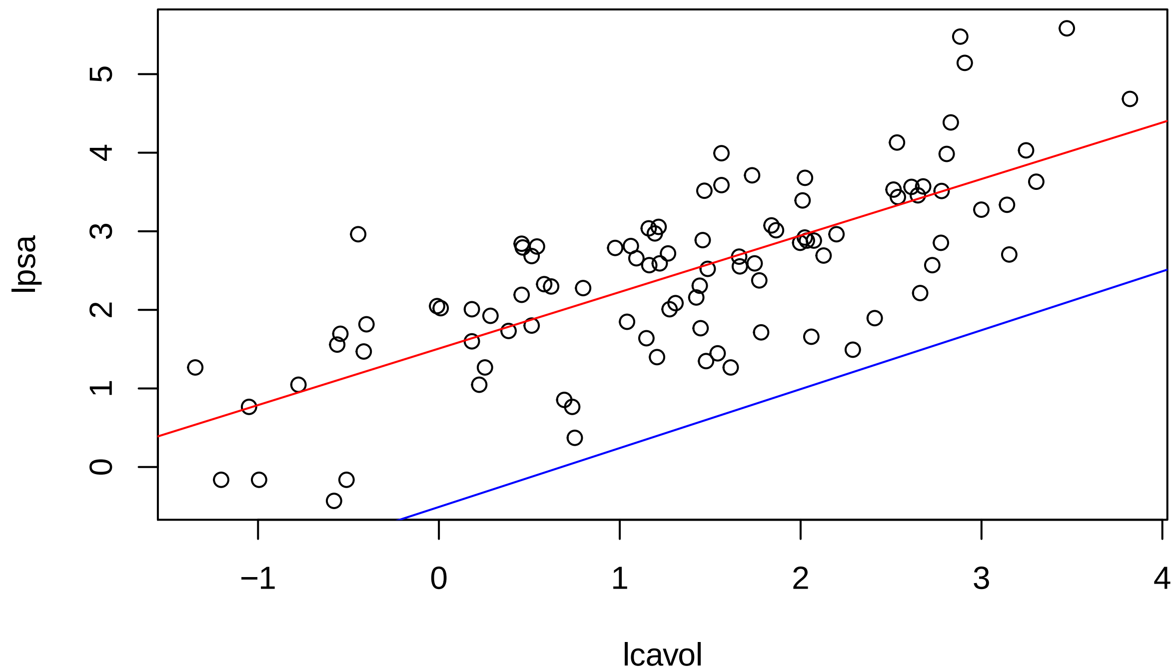
```r
data <- data.frame(variables, reserror, coef)

data <- melt(data, id="variables")

ggplot(data=data, aes(x=variables, y=value, colour=variable)) +
  geom_line()
```



5.

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?

```
lm5.1 <- lm(lpsa ~ lcavol, data=prostate)
lm5.2 <- lm(lcavol ~ lpsa, data=prostate)

plot(lpsa ~ lcavol, data=prostate)
abline(lm5.1, col="red")
abline(lm5.2, col="blue")
```



Las rectan son paralelas, pero cada recta tiene unos ejes distintos, aqui estamos representando las dos rectas respecto a los ejes lpsa against lcavol.

6.

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.

```
lm6 <- lm(taste ~ Acetic + H2S + Lactic, data=cheddar)
coefcheddar <- lm6$coefficients
coefcheddar
```

```
## (Intercept)      Acetic         H2S      Lactic
## -28.8767696   0.3277413   3.9118411  19.6705434
```

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

```
corr <- cor(cheddar$taste, lm6$fitted.values)^2
corr
```

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

```r
summary(lm6)
```

```
##
## Call:
## lm(formula = taste ~ Acetic + H2S + Lactic, data = cheddar)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.390  -6.612  -1.009   4.908  25.449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.8768    19.7354  -1.463  0.15540
## Acetic        0.3277     4.4598   0.073  0.94198
## H2S           3.9118     1.2484   3.133  0.00425 **
## Lactic       19.6705     8.6291   2.280  0.03108 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.13 on 26 degrees of freedom
## Multiple R-squared:  0.6518, Adjusted R-squared:  0.6116
## F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

Es el valor de `Multiple R-squared`

(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.

```r
lm6.2 <- lm(taste ~ 0 + Acetic + H2S + Lactic, data=cheddar)
summary(lm6.2)
```

```
##
## Call:
## lm(formula = taste ~ 0 + Acetic + H2S + Lactic, data = cheddar)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.4521  -6.5262  -0.6388   4.6811  28.4744
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## Acetic   -5.454      2.111  -2.583  0.01553 *
## H2S       4.576      1.187   3.854  0.00065 ***
## Lactic   19.127      8.801   2.173  0.03871 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.34 on 27 degrees of freedom
## Multiple R-squared:  0.8877, Adjusted R-squared:  0.8752
## F-statistic: 71.15 on 3 and 27 DF,  p-value: 6.099e-13
```

El valor de R-squared es `multiple R-squared = 0.8877`

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

7.

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

```
lm7 <- lm(resist ~ x1 + x2 + x3 + x4, data=wafer)
head(wafer, package="faraway")
```

```
##   x1 x2 x3 x4 resist
## 1  -  -  -  -  193.4
## 2  +  -  -  -  247.6
## 3  -  +  -  -  168.2
## 4  +  +  -  -  205.0
## 5  -  -  +  -  303.4
## 6  +  -  +  -  339.9
```

(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.

```
matrix7 <- model.matrix(lm7)
matrix7
```

```
##    (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"
```

Vemos que los valores minus se han asignado a `0`y los `+`a `1`.

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

```
cor(matrix7)
```

```
## Warning in cor(matrix7): 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
```

```
cor
```

```
## function (x, y = NULL, use = "everything", method = c("pearson",
##     "kendall", "spearman"))
## {
##     na.method <- pmatch(use, c("all.obs", "complete.obs", "pairwise.complete.obs",
##         "everything", "na.or.complete"))
##     if (is.na(na.method))
##         stop("invalid 'use' argument")
##     method <- match.arg(method)
##     if (is.data.frame(y))
##         y <- as.matrix(y)
##     if (is.data.frame(x))
##         x <- as.matrix(x)
##     if (!is.matrix(x) && is.null(y))
##         stop("supply both 'x' and 'y' or a matrix-like 'x'")
##     if (!(is.numeric(x) || is.logical(x)))
##         stop("'x' must be numeric")
##     stopifnot(is.atomic(x))
##     if (!is.null(y)) {
##         if (!(is.numeric(y) || is.logical(y)))
##             stop("'y' must be numeric")
##         stopifnot(is.atomic(y))
##     }
##     Rank <- function(u) {
##         if (length(u) == 0L)
##             u
##         else if (is.matrix(u)) {
```

```
##             if (nrow(u) > 1L)
##                 apply(u, 2L, rank, na.last = "keep")
##             else row(u)
##         }
##         else rank(u, na.last = "keep")
##     }
##     if (method == "pearson")
##         .Call(C_cor, x, y, na.method, FALSE)
##     else if (na.method %in% c(2L, 5L)) {
##         if (is.null(y)) {
##             .Call(C_cor, Rank(na.omit(x)), NULL, na.method, method ==
##                 "kendall")
##         }
##         else {
##             nas <- attr(na.omit(cbind(x, y)), "na.action")
##             dropNA <- function(x, nas) {
##                 if (length(nas)) {
##                   if (is.matrix(x))
##                     x[-nas, , drop = FALSE]
##                   else x[-nas]
##                 }
##                 else x
##             }
##             .Call(C_cor, Rank(dropNA(x, nas)), Rank(dropNA(y,
##                 nas)), na.method, method == "kendall")
##         }
##     }
##     else if (na.method != 3L) {
##         x <- Rank(x)
##         if (!is.null(y))
##             y <- Rank(y)
##         .Call(C_cor, x, y, na.method, method == "kendall")
##     }
##     else {
##         if (is.null(y)) {
##             ncy <- ncx <- ncol(x)
##             if (ncx == 0)
##                 stop("'x' is empty")
##             r <- matrix(0, nrow = ncx, ncol = ncy)
##             for (i in seq_len(ncx)) {
##                 for (j in seq_len(i)) {
##                   x2 <- x[, i]
##                   y2 <- x[, j]
##                   ok <- complete.cases(x2, y2)
##                   x2 <- rank(x2[ok])
##                   y2 <- rank(y2[ok])
##                   r[i, j] <- if (any(ok))
##                     .Call(C_cor, x2, y2, 1L, method == "kendall")
##                   else NA
##                 }
##             }
##             r <- r + t(r) - diag(diag(r))
##             rownames(r) <- colnames(x)
##             colnames(r) <- colnames(x)
```

```
##             r
##         }
##         else {
##             if (length(x) == 0L || length(y) == 0L)
##                 stop("both 'x' and 'y' must be non-empty")
##             matrix_result <- is.matrix(x) || is.matrix(y)
##             if (!is.matrix(x))
##                 x <- matrix(x, ncol = 1L)
##             if (!is.matrix(y))
##                 y <- matrix(y, ncol = 1L)
##             ncx <- ncol(x)
##             ncy <- ncol(y)
##             r <- matrix(0, nrow = ncx, ncol = ncy)
##             for (i in seq_len(ncx)) {
##                 for (j in seq_len(ncy)) {
##                   x2 <- x[, i]
##                   y2 <- y[, j]
##                   ok <- complete.cases(x2, y2)
##                   x2 <- rank(x2[ok])
##                   y2 <- rank(y2[ok])
##                   r[i, j] <- if (any(ok))
##                     .Call(C_cor, x2, y2, 1L, method == "kendall")
##                   else NA
##                 }
##             }
##             rownames(r) <- colnames(x)
##             colnames(r) <- colnames(y)
##             if (matrix_result)
##                 r
##             else drop(r)
##         }
##     }
## }
## <bytecode: 0x00000000193d67a8>
## <environment: namespace:stats>
```

Esto se puede deber a que los valores del ìntercept‘son constantes para todos

```
cor2 <- cor(matrix7[,2:5])
cor2
```

```
##     x1+ x2+ x3+ x4+
## x1+   1   0   0   0
## x2+   0   1   0   0
## x3+   0   0   1   0
## x4+   0   0   0   1
```

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

```
summary(lm7)
```

```
##
```

```
## Call:
## lm(formula = resist ~ x1 + x2 + x3 + x4, data = wafer)
##
## 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
```

Una diferencia de `25.76`unidades

(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?

```
lm7.2 <- lm(resist ~ x1 + x2 + x3, data=wafer)
summary(lm7.2)
```

```
##
## Call:
## lm(formula = resist ~ x1 + x2 + x3, data = wafer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.137 -20.550   3.575  18.462  41.013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   229.54      13.32  17.231 7.88e-10 ***
## x1+            25.76      13.32   1.934 0.077047 .
## x2+           -69.89      13.32  -5.246 0.000206 ***
## x3+            43.59      13.32   3.272 0.006677 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.64 on 12 degrees of freedom
## Multiple R-squared:  0.7777, Adjusted R-squared:  0.7221
## F-statistic: 13.99 on 3 and 12 DF,  p-value: 0.0003187
```

Los coeficientes de las variables `x1,x2 y x3` se mantienen iguales. Tenemos un `Multiple R-squared`algo menor. El `intercept` estimado tambien cambia ligeramente, pues ya no tiene en cuenta la dependencia de la variable `x4`

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

Ejercicios Carmona

##1. 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)

$$y_i = \beta_0 + \beta_1 x_i + \beta_2(x_i^2 - 1) + \epsilon_i$$

Es lineal

```
matriza <- matrix(c(1,1,(1^2 -1),
                    1,2,(2^2 -1),
                    1,3,(3^2 -1)), nrow=3, byrow=TRUE)
matriza
```

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

b)

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 e^{xi} + \epsilon_i$$

Es lineal

```
matrizb <- matrix(c(1,1,exp(1),
                    1,2,exp(2),
                    1,3,exp(3)), nrow=3, byrow=TRUE)
matrizb
```

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

c)

$$y_i = \beta_1 x_i(\beta_2 tan(xi)) + \epsilon_i$$

No es lineal

2.

Cuatro objetos cuyos pesos exactos son beta1, beta2, beta3 y beta4 han sido pesados en una balanza de platillos. Hallar las estimaciones de cada betai y de la varianza del error.

```r
#Creamos la matriz de diseño
X<- matrix(c(1,1,1,1,
             1,-1,1,1,
             1,0,0,1,
             1,0,0,-1,
             1,0,1,1,
             1,1,-1,1), ncol = 4, byrow=TRUE)

Y <- matrix(c(9.2, 8.3, 5.4, -1.6, 8.7, 3.5), ncol = 1)

X
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1   -1    1    1
## [3,]    1    0    0    1
## [4,]    1    0    0   -1
## [5,]    1    0    1    1
## [6,]    1    1   -1    1
```

```r
Y
```

```
##      [,1]
## [1,]  9.2
## [2,]  8.3
## [3,]  5.4
## [4,] -1.6
## [5,]  8.7
## [6,]  3.5
```

```r
#Multiplicamos ambas direcciones por la matriz traspuesta
XtX <- t(X)%*%X
XtY <- t(X)%*%Y
```

```r
#Con la funcion solve() calculamos las estimaciones de cada parametro
estim <- solve(XtX, XtY)
estim
```

```
##           [,1]
## [1,] 2.0685714
## [2,] 0.5342857
## [3,] 2.9400000
## [4,] 3.6685714
```

```r
n= length(Y)
p = length(estim)

#Mediante calculo matricial calculamos la varianza del error
(S2=t(Y-X%*%estim)%*%(Y-X%*%estim)/(n-p))
```

```
##            [,1]
## [1,] 0.08371429
```