Ejemplo clase

```r
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
library(tidyverse)
library(tidymodels)
```

Sección 1

Introducción

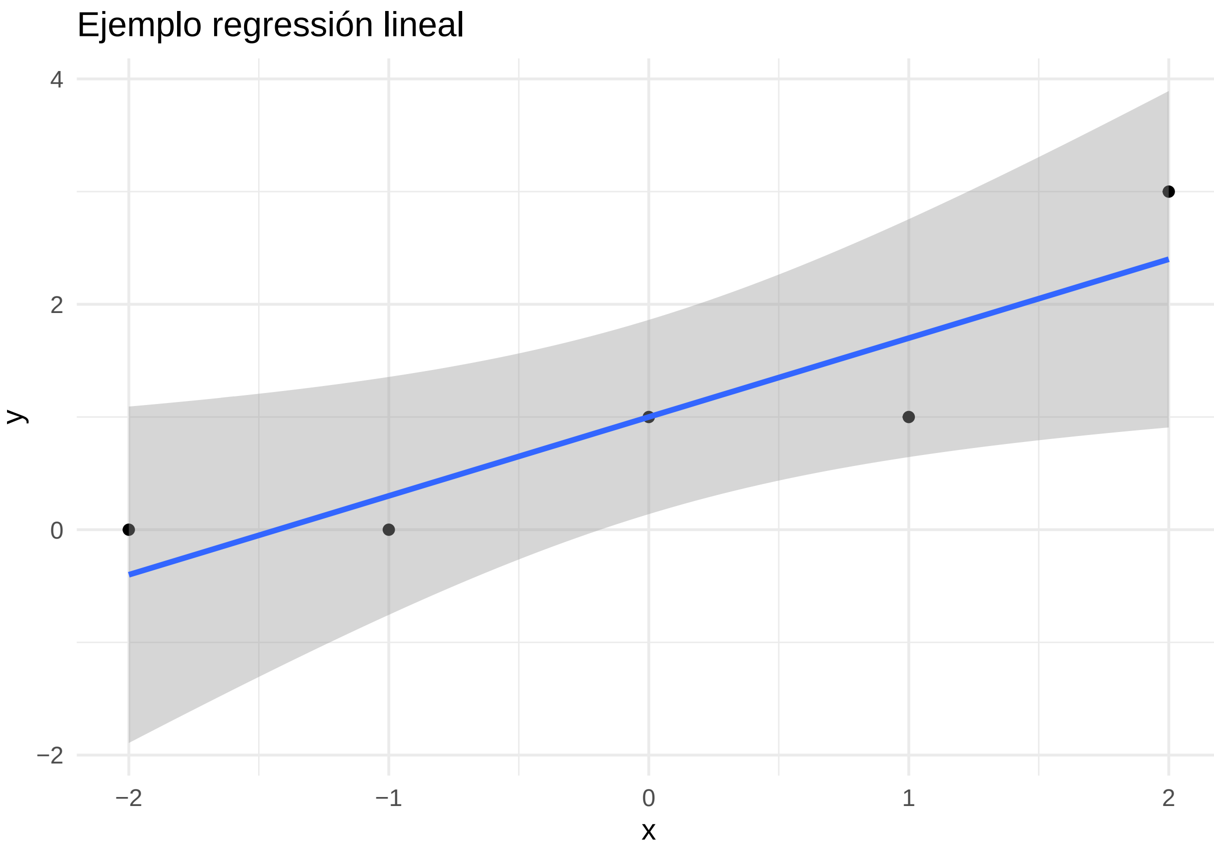
Este ejemplo es la versión en R del ejercicio visto en clase. Primero se guardan los datos en un data.frame llamado d:

```r
library(tidyverse)
library(tidymodels)

x <- c(-2, -1, 0, 1, 2)
y <- c(0, 0, 1, 1, 3)

d <- data.frame(x = x, y =y)
d
```

```
##    x y
## 1 -2 0
## 2 -1 0
## 3  0 1
## 4  1 1
## 5  2 3
```

Se realiza un exploratorio para observar los datos:

```r
d %>%
  ggplot(aes(x = x, y = y))+
  geom_point()+
  geom_smooth(method = "lm")+
  theme_minimal()+
  labs(title = "Ejemplo regressión lineal")
```



Se ajusta la regresión lineal:

```
lm_fit <-
  linear_reg() %>%
  fit(y ~ x, data = d)

tidy(lm_fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)      1       0.271      3.69  0.0345
## 2 x                0.7     0.191      3.66  0.0354
```

```
data.frame(glance(lm_fit))
```

```
##   r.squared adj.r.squared     sigma statistic    p.value df    logLik      AIC
## 1 0.8166667     0.7555556 0.6055301  13.36364 0.03535285  1 -3.309373 12.61875
##        BIC deviance df.residual nobs
## 1 11.44706      1.1           3    5
```

Inciso 1: Calcule $\bar{X}$, $\bar{Y}$, S_X^2, S_Y^2, S_{xy}

```
# 1)
xbar = mean(d$x)
ybar = mean(d$y)
sx2 = var(d$x)
sy2 = var(d$y)
```

```
sxy = cov(d$x, d$y)
```

Inciso 2: Calcule $\hat{\beta}_0$ y $\hat{\beta}_1$:

```
B1 <- sxy/sx2
B0 <- ybar - B1*xbar
```

Inciso 3: Calcule TSS, ESS, RSS, R^2 y $adjR^2$

```
yhat <- predict(lm_fit, new_data = d %>% select(x))
TSS <- sum((d$y-ybar)**2)
ESS <- sum((yhat-ybar)**2)
RSS <- sum((d$y-yhat)**2)
R2 <- ESS/TSS
adjR2 <- 1- ((1-R2)*(5-1))/(5-1-1)
```

Inciso 4: Calcule $\hat{\sigma}^2$, $V(\hat{\beta}_0)$, $V(\hat{\beta}_1)$

```
sigma2hat <- RSS/(5-2)
sigmahat <- sqrt(sigma2hat)

sigma2_B0 <- sigma2hat*(1/5 + 0/(sx2*(5-1)))
sigma2_B1 <- sigma2hat/(sx2*(5-1))

sigma_B0 <- sqrt(sigma2_B0)
sigma_B1 <- sqrt(sigma2_B1)
```

Inciso 5: Pruebas de hipótesis coeficientes:

```
T_B1 = (B1-0)/sqrt(sigma2_B1)
T_B0 = (B0-0)/sqrt(sigma2_B0)

2*(1-pt(T_B1, df = 5-2))
```

```
## [1] 0.03535285
```

```
2*(1-pt(T_B0, df = 5-2))
```

```
## [1] 0.03445085
```

Inciso 6: Pruebas de hipótesis correlación:

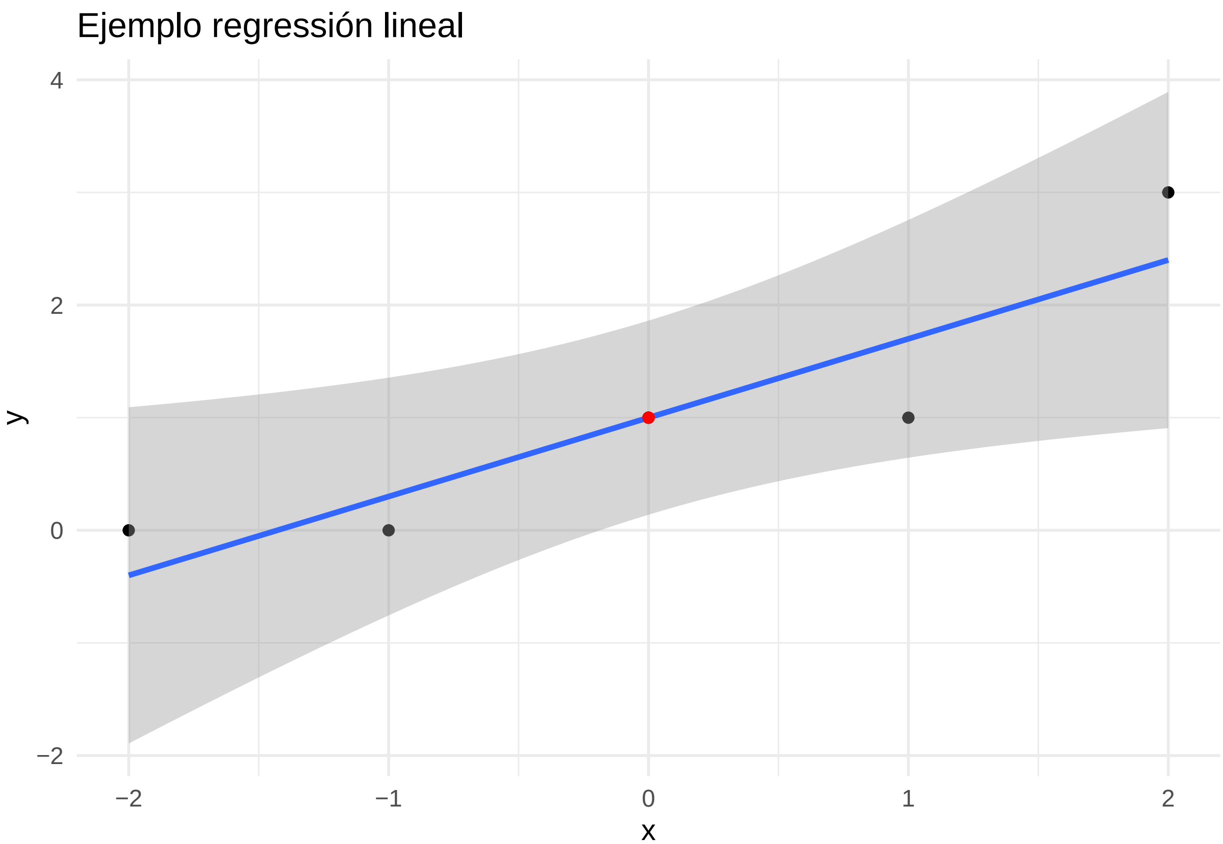
```
rxy = sxy/(sqrt(sx2)*sqrt(sy2))
T_corr = (rxy*sqrt(5-2))/sqrt(1-rxy**2)
2*(1-pt(T_corr, df = 5-2))
```

```
## [1] 0.03535285
```

Inciso 7: Comprobación de propiedades

```
# recta pasa por la media
d %>%
```

```r
ggplot(aes(x = x, y = y))+
geom_point()+
geom_smooth(method = "lm")+
theme_minimal()+
geom_point(x=xbar, y = ybar, color = "red")+
labs(title = "Ejemplo regressión lineal")
```




```r
# suma de residuos es 0
sum(d$y-yhat)
```

```
## [1] 0
```

```r
# residuos no están correlacionados con x
r_rx = cor(d$x, d$y-yhat)
# residuos no están correlacionados con yhat
r_ryhat = cor(yhat, d$y-yhat)
```

Inciso 8: Prueba F

```r
Fest = (ESS/1)/(RSS/3)
Fest
```

```
## [1] 13.36364
```

```r
pf(Fest, df1 = 1, df2 = 3, lower.tail = F)
```

```
## [1] 0.03535285
```

Inciso 9: Predicción a la media

```
mean_pred <- predict(lm_fit, new_data = data.frame(x = c(0,3)))
int_pred <- predict(lm_fit, new_data = data.frame(x = c(0,3)),
                    type = "conf_int", level = .95)
plot_data <-
  data.frame(x = c(0,3)) %>%
  bind_cols(mean_pred) %>%
  bind_cols(int_pred) %>%
  mutate(
    error_pred = sigmahat*sqrt((1/5)+(((x-xbar)**2)/(sx2*(5-1)))),
    errorind_pred = sigmahat*sqrt(1+(1/5)+(((x-xbar)**2)/(sx2*(5-1)))),
    pred_lower_formula = .pred-qt(.975, df = 3)*error_pred,
    pred_upper_formula = .pred+qt(.975, df = 3)*error_pred,
    pred_lowerind_formula = .pred-qt(.975, df = 3)*errorind_pred,
    pred_upperind_formula = .pred+qt(.975, df = 3)*errorind_pred)

plot_data
```

```
##   x .pred .pred_lower .pred_upper error_pred errorind_pred pred_lower_formula
## 1 0   1.0   0.1381895    1.861811  0.2708013     0.6633250          0.1381895
## 2 3   3.1   1.0788751    5.121125  0.6350853     0.8774964          1.0788751
##   pred_upper_formula pred_lowerind_formula pred_upperind_formula
## 1           1.861811            -1.1109961              3.110996
## 2           5.121125             0.3074147              5.892585
```

Inciso 9: Predicción individual

Sección 2) Ejercicio

Introducción

El problema presentado en este ejercicio es determinar si existe una relación lineal entre la edad de un niño y su altura. Se tiene la intuición que a mayor edad, más alto es. A continuación se presentan los datos de edad (meses) y altura (cm) en una muestra de 12 niños

```
data <- read_csv("https://raw.githubusercontent.com/savrgg/class_ITAM_metodos/main/notas_r/ageandheight
data
```

```
## # A tibble: 12 x 2
##      age height
##    <dbl>  <dbl>
##  1    18   76.1
##  2    19   77
##  3    20   78.1
##  4    21   78.2
##  5    22   78.8
##  6    23   79.7
##  7    24   79.9
##  8    25   81.1
##  9    26   81.2
## 10    27   81.8
## 11    28   82.8
## 12    29   83.5
```

1. Determine por medio de una gráfica si es razonable pensar en que existe una relación lineal entre las variables (variable independiente: edad) Al comparar gráficamente los datos, podemos observar que los datos si se ajustan por medio de una línea.