Ejemplo clase

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:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.0.0 --
## v broom        1.0.0     v rsample      1.1.0
## v dials        1.0.0     v tune         1.0.0
## v infer        1.0.2     v workflows    1.0.0
## v modeldata    1.0.0     v workflowsets 1.0.0
## v parsnip      1.0.0     v yardstick    1.0.0
## v recipes      1.0.1
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

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

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

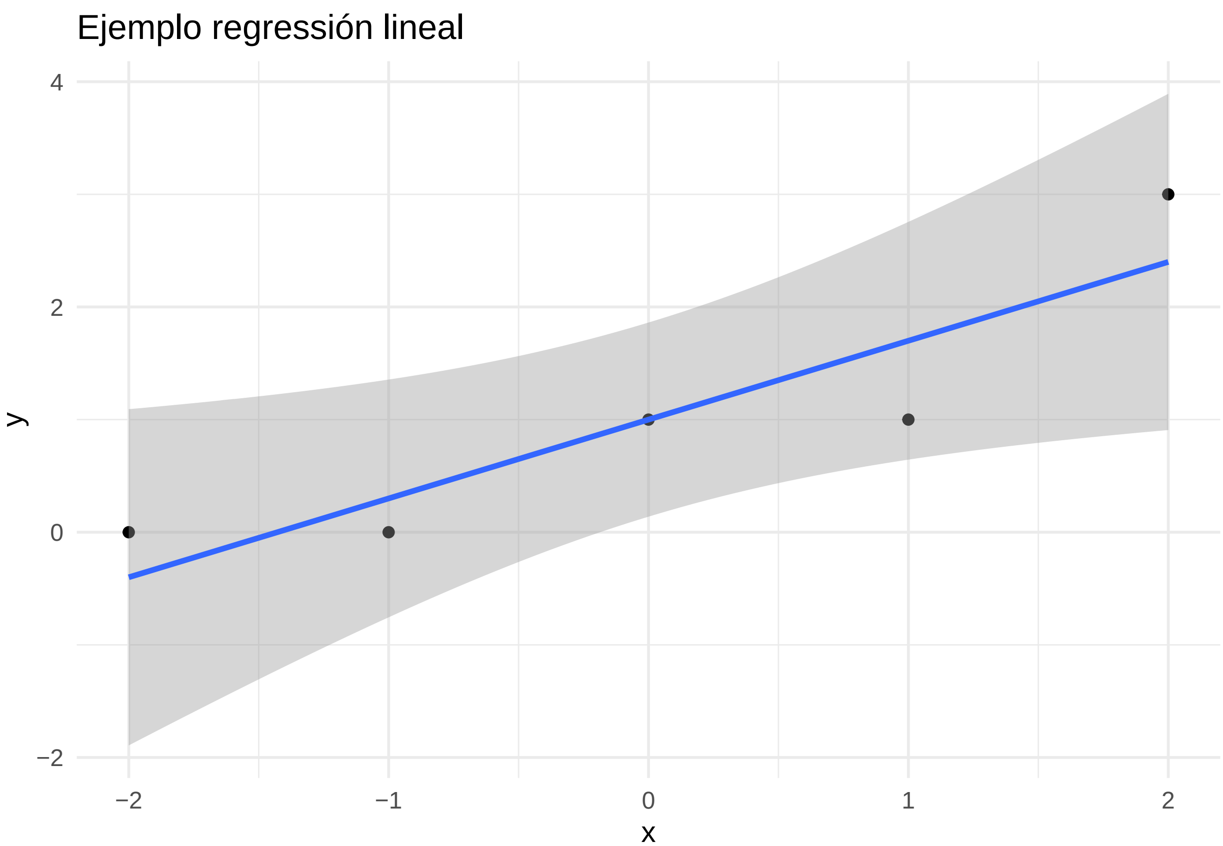
```
##    x y
## 1 -2 0
## 2 -1 0
## 3  0 1
## 4  1 1
```

```
## 5  2 3
```

Se realiza un exploratorio para observar los datos:

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

```
## `geom_smooth()` using formula 'y ~ x'
```



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 $\hat{\sigma}^2$, $\hat{V}(\hat{\beta}_0)$, $\hat{V}(\hat{\beta}_1)$

```
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:

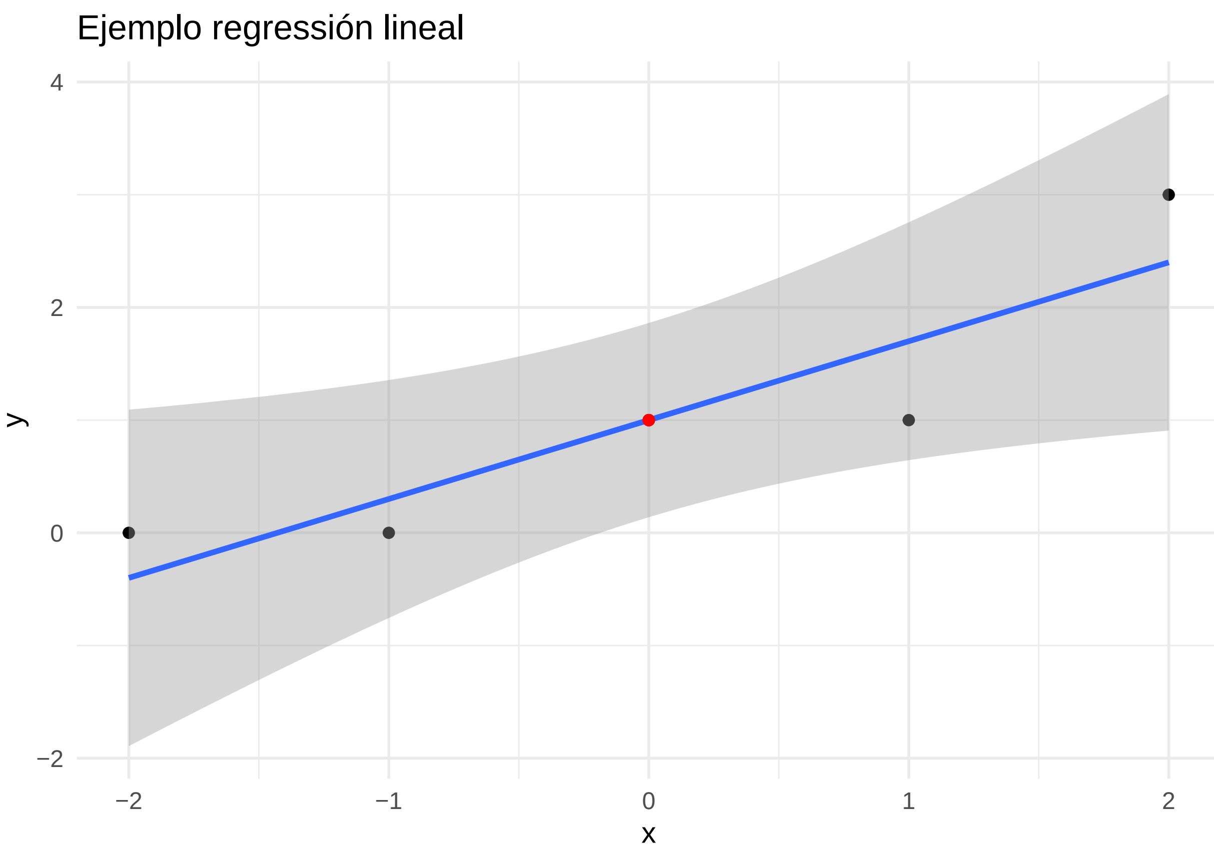
```
rxy = sxy/(sqrt(sx2)*sqrt(sy2))
T_corr = (rxy*sqrt(5-2))/sqrt(1-rxy**2)
2*(1-pt(T_corr, df = 5-2))
```

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

Inciso 7: Comprobación de propiedades

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

```
## `geom_smooth()` using formula 'y ~ x'
```




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

```
## [1] -3.330669e-16
```

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

```r
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

```r
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

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

```
## Rows: 12 Columns: 2
## -- Column specification ------------------------------------------------------
## Delimiter: ","
## dbl (2): age, height
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

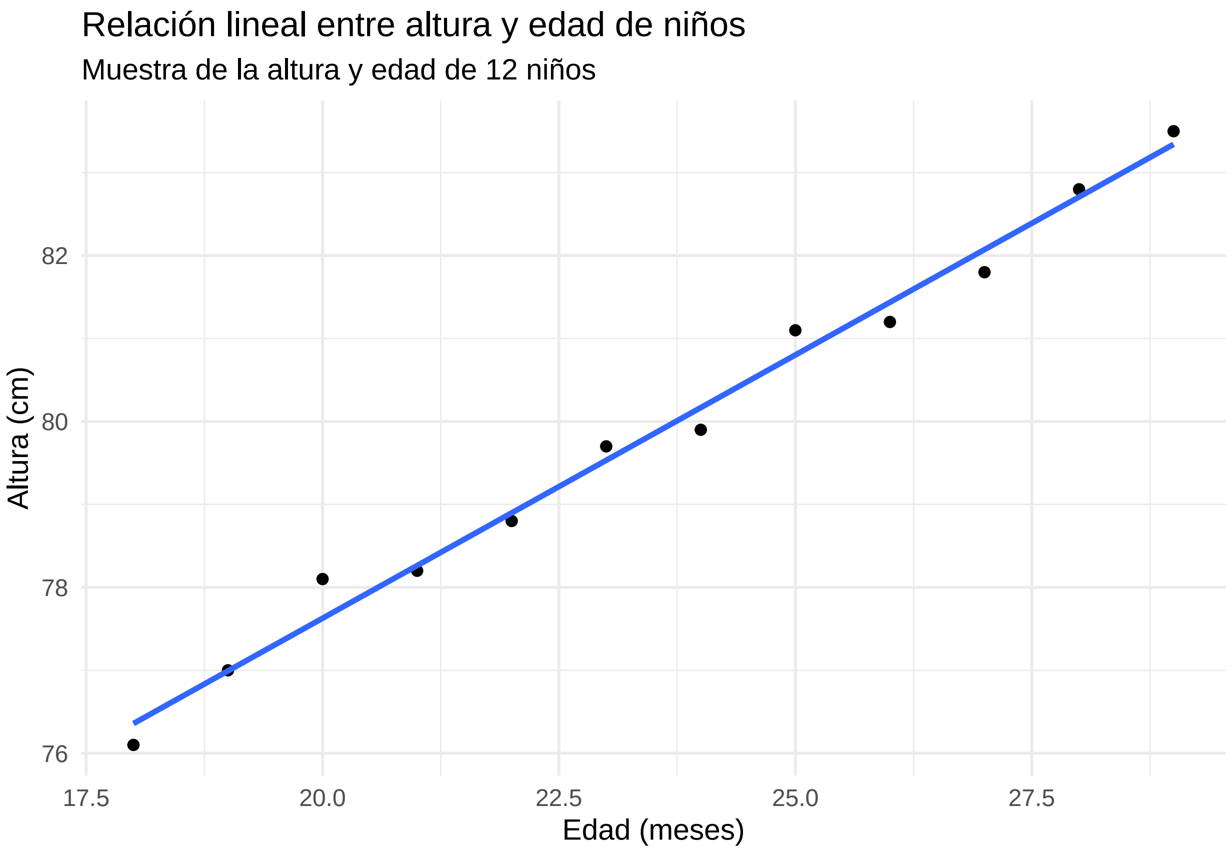
```r
data
```

```
## # 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
```

```r
data %>%
  ggplot(aes(x = age, y = height))+
  geom_point()+
  geom_smooth(method = "lm", se = F)+
  theme_minimal()+
  labs(x = "Edad (meses)",
       y = "Altura (cm)",
       title = "Relación lineal entre altura y edad de niños",
       subtitle = "Muestra de la altura y edad de 12 niños")
```

1. Determine por medio de una gráfica si es razonable pensar en que existe una relación lineal entre las variables (variable independiente: edad)

```
## `geom_smooth()` using formula 'y ~ x'
```



Al comparar gráficamente los datos, podemos observar que los datos si se ajustan por medio de una línea.

2. Determine con una Prueba de Hipótesis si existe una relación lineal entre las variables. Para determinar si existe una relación lineal, se construye una Prueba de Hipótesis del coeficiente de correlación. Recordando, el coeficiente de correlación mide solamente la asociación lineal, mas no la pendiente de la linea

Sea $H_0 : \rho = 0$ y $H_1 : \rho \neq 0$, entonces:

```
n = nrow(data)
ybar = mean(data$height)
xbar = mean(data$age)
sxy = cov(data$age, data$height)
sx2 = var(data$age)
sy2 = var(data$height)
rxy = sxy/(sqrt(sx2)*sqrt(sy2))
T_corr = (rxy*sqrt(n-2))/sqrt(1-rxy**2)
T_corr
```

```
## [1] 29.66465
```

```
2*(1-pt(T_corr, df = n-2))
```

```
## [1] 4.428058e-11
```

El $valor - p$ es cercano a 0, entonces rechazamos $H_0 : \rho = 0$, por lo que rechazamos que la correlación sea cero.

```
lm_fit <-
```

```
  linear_reg() %>%
  fit(height ~ age, data = data)

tidy(lm_fit)
```

3. Utilicé tidymodels (lm) para realizar la regresión lineal, analice los resultados obtenidos

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   64.9      0.508      128.  2.13e-17
## 2 age            0.635    0.0214      29.7 4.43e-11
```

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

```
##   r.squared adj.r.squared     sigma statistic      p.value df    logLik
## 1 0.9887639     0.9876403 0.2559638  879.9915 4.428071e-11  1 0.4192965
##        AIC      BIC  deviance df.residual nobs
## 1 5.161407 6.616127 0.6551748          10   12
```

```

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

4. Determine analíticamente los coeficientes $\hat{\beta}_0$ y $\hat{\beta}_1$

```
## [1] 0.634965
```

```
B0
```

```
## [1] 64.92832
```

5. Calcule analíticamente TSS, ESS, RSS, R^2 y $adjR^2$

```
yhat <- predict(lm_fit, new_data = data %>% select(age))
TSS <- sum((data$height-ybar)**2)
ESS <- sum((yhat-ybar)**2)
RSS <- sum((data$height-yhat)**2)
R2 <- ESS/TSS
adjR2 <- 1- ((1-R2)*(n-1))/(n-1-1)
```

6. Calcule analíticametne $\hat{\sigma}^2$, $V(\hat{\beta}_0)$, $V(\hat{\beta}_1)$

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

sigma2_B0 <- sigma2hat*(1/n + (xbar*xbar)/(sx2*(n-1)))
sigma2_B1 <- sigma2hat/(sx2*(n-1))

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

sigma_B0
```

```
## [1] 0.5084102
```

```r
sigma_B1
```

```
## [1] 0.02140477
```

7. Determine si el valor de los coeficientes es significativo (realice una prueba para cada uno)

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

T_B1
```

```
## [1] 29.66465
```

```r
T_B0
```

```
## [1] 127.7085
```

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

```
## [1] 4.428058e-11
```

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

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

El valor-p de ambas pruebas es cercano a 0, por lo que podemos rechazar que $H_0 : \beta_1 = 0$ y $H_0 : \beta_0 = 0$

8. Comprueba que:

a) Suma de residuales es igual a 0

```r
sum(data$height-yhat)
```

```
## [1] -3.126388e-13
```

b) Residuos no están correlacionados a X

```r
cor(data$age, data$height-yhat)
```

```
##             .pred
## [1,] 2.202239e-14
```

c) Residuos no están correlacionados a $\hat{Y}$

```r
cor(yhat, data$height-yhat)
```

```
##              .pred
## .pred 2.144355e-14
```

9. Determine si el de R^2 es significativo

```r
Fest = (ESS/1)/(RSS/(n-2))
Fest
```

```
## [1] 879.9915
```

```r
pf(Fest, df1 = 1, df2 = (n-2), lower.tail = F)
```

```
## [1] 4.428071e-11
```

El $valor - p$ es cercano a 0,por lo que podemos rechazar $H_0 : \rho^2 = 0$ vs $H_1 : \rho^2 > 0$.

Calcula la predicción a la media para una edad 25 años. Calcule el IC al 95%.

```
yhat_25 <- B0+B1*25
quantil_t = qt(.975, df = n-2)
lim_inf = yhat_25 - quantil_t*sigmahat*sqrt(1/n + ((25-xbar)^2)/(sx2*(n-1)))
lim_sup = yhat_25 + quantil_t*sigmahat*sqrt(1/n + ((25-xbar)^2)/(sx2*(n-1)))
```

IC para prediccion media de edad 25: (80.62294, 80.98196)

Calcula la predicción individual para un valor de 30 años. Calcule el IC al 95

```
yhat_30 <- B0+B1*30
quantil_t = qt(.975, df = n-2)
lim_inf = yhat_30 - quantil_t*sigmahat*sqrt(1/n + ((30-xbar)^2)/(sx2*(n-1)))
lim_sup = yhat_30 + quantil_t*sigmahat*sqrt(1/n + ((30-xbar)^2)/(sx2*(n-1)))
```

IC para prediccion individual de edad 30: (83.62626, 84.32828)