```{r, echo = FALSE}

knitr::opts\_chunk$set(echo = TRUE)

```

```{r, echo = TRUE}

library(readxl)

data <- as.data.frame(read\_excel("C:/correlacion\_regresion\_lineal/información correlacion/data.xls"))

View(data)

print(data)

```

Actividad 1: teórica

Actividad 2: teórica

Actividad 3:

Actividad 3. Calcula la correlación entre las variables almacenadas en la tabla 'data'.

```{r, echo = TRUE}

correlacion\_datos <- cor(data)

print(correlacion\_datos)

### Actividad 4. Calcula los coeficientes de correlación de las variables junto con el nivel de significancia (p-value) en 1 solo gráfico. Interpreta los resultados.

```{r, echo = TRUE}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {

usr <- par("usr")

on.exit(par(usr))

par(usr = c(0, 1, 0 ,1))

Cor <- abs(cor(x, y))

txt <- paste0(prefix, format(c(Cor, 0.123456789), digits = digits)[1])

if(missing(cex.cor)) {

cex.cor <- 0.4 / strwidth(txt)

}

text(0.5, 0.5, txt,

cex = 1 + cex.cor\*Cor)

}

```

#Aquí dibujamos la matriz de correlación

```{r, echo = TRUE}

pairs(data,

upper.panel = panel.cor, # Este es el panel de correlación

lower.panel = panel.smooth)

```

Actividad 5.

### Ejercicio 5. Emplea una función para obtener en una matriz de correlación lineal, IC 95% y p-value de todas las variables en el data frame 'data'.

```{r, echo = TRUE}

library(correlation)

matriz <- correlation(data)

matriz

```

### Ejercicio 6. Visualiza gráficamente la correlación lineal existente entre las variables 'longitud' y 'peso'.

```{r, echo = TRUE}

library(ggpubr)

library(ggplot2)

ggscatter(data, x = "altura", y = "peso",

add = "reg.line", conf.int = TRUE,

cor.coef = TRUE, cor.method = "pearson",

xlab = "altura piezas (mm)", ylab = "peso piezas (mg)")

### Ejercicio 7. Emplea la librería 'corrplot()' para visualizar la correlación entre variables.

```{r, echo = TRUE}

library(corrplot)

corrplot(cor(data))

```

### Ejercicio 8. A partir de la siguiente secuncia de valores numéricos:

#a) Creamos 2 vectores: 'distancia' y 'n\_piezas' para almacenarlos en un data frame:

```{r, echo = TRUE}

distancia <- c( 1.1,100.2,90.3,5.4,57.5,6.6,34.7,65.8,57.9,86.1)

n\_piezas <- c(110,2,6,98,40,94,31,5,8,10)

datos\_2 <- data.frame(distancia, n\_piezas)

print(datos\_2)

```

#b) Calcula el coeficiente de correlación:

```{r, echo = TRUE}

correlacion\_datos\_2 <- cor(datos\_2)

print(correlacion\_datos\_2)

```

#c) Calcula el nivel de significancia:

```{r, echo = TRUE}

significancia\_datos\_2 <- cor.test(datos\_2$distancia, datos\_2$n\_piezas)$p.value

print(significancia\_datos\_2)

```

#d) Calcula el Intervalo de confianza al 95% en relación con el coeficiente de correlación:

```{r, echo = TRUE}

intervaloconfianza\_datos\_2 <- cor.test(datos\_2$distancia, datos\_2$n\_piezas)$conf.int

print(intervaloconfianza\_datos\_2)

```