

# Gráficos básicos

Cynthia Lizbeth Diaz Pale

2022-09-23

## Reconocimiento de la matriz de datos.

1.- Indicar la ruta de trabajo

session/Set working Directory/ Choose Directory/Cloud/project/choose.

2.- Lectura de la matriz de datos. Se copia la ruta de acceso que proporciona R y se le agrega el nombre del archivo.

```
library("readxl")  
BD<-read_excel("/cloud/project/penguins.xlsx")
```

## Exploración de la matriz

1.- Visualización de los nombres de las variables.

```
colnames(BD)  
  
## [1] "ID"           "especie"      "isla"         "largo_pico_mm"  
## [5] "grosor_pico_mm" "largo_aleta_mm" "masa_corporal_g" "genero"  
## [9] "año"
```

2.- Visualización del tipo de variable

```
str(BD)  
  
## tibble [344 x 9] (S3: tbl_df/tbl/data.frame)  
## $ ID           : chr [1:344] "i1" "i2" "i3" "i4" ...  
## $ especie      : chr [1:344] "Adelie" "Adelie" "Adelie" "Adelie" ...  
## $ isla         : chr [1:344] "Torgersen" "Torgersen" "Torgersen" "Torgersen" ...  
## $ largo_pico_mm : num [1:344] 39.1 39.5 40.3 37.8 36.7 39.3 38.9 39.2 34.1 42 ...  
## $ grosor_pico_mm : num [1:344] 18.7 17.4 18 18.1 19.3 20.6 17.8 19.6 18.1 20.2 ...  
## $ largo_aleta_mm : num [1:344] 181 186 195 190 193 190 181 195 193 190 ...  
## $ masa_corporal_g: num [1:344] 3750 3800 3250 3700 3450 ...  
## $ genero       : chr [1:344] "male" "female" "female" "female" ...  
## $ año          : num [1:344] 2007 2007 2007 2007 2007 ...
```

## Instalación de librerías

Se va a instalar el paquete *ggplot2*. No olvides declarar los argumentos `warning=FALSE` y `message=FALSE`

```
library(ggplot2)
```

# Generación de gráficos

## Boxplot

Se utiliza para visualizar variables cualitativas y cuantitativas simultáneamente. En el eje de la  $x$  se muestra la variable cualitativa *codificada en factor*. En el eje  $y$  se muestra la variable cuantitativa.

1.- Codificación de la variable cualitativa a factor.

```
BD$genero<-factor(BD$genero,  
                  levels=c("male", "female"))
```

2.- Creación de un vector de color

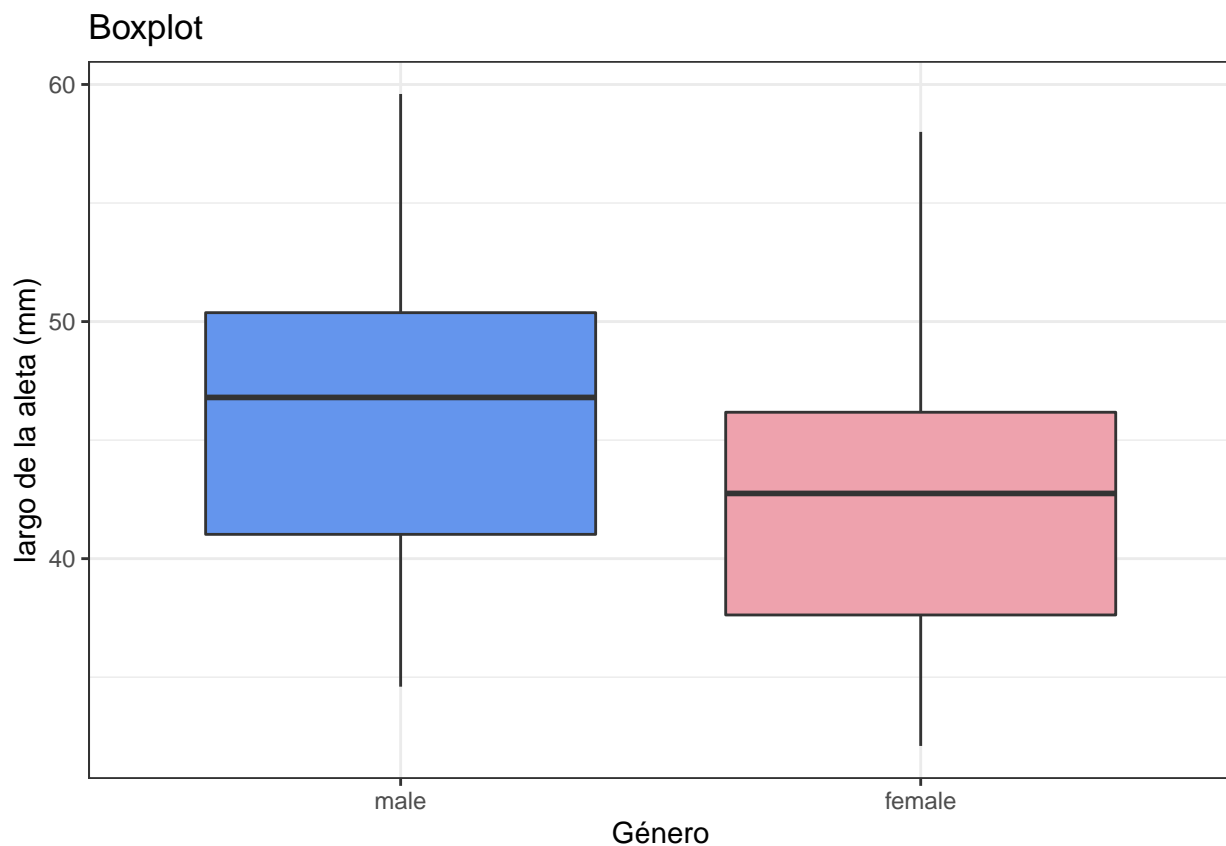
```
color=c("cornflowerblue","lightpink2")
```

3.- Creación del gráfico

```
BX<-ggplot(BD, aes(x=genero, y=largo_pico_mm))+  
  geom_boxplot(fill=color)+  
  ggtitle("Boxplot")+  
  xlab("Género")+  
  ylab("largo de la aleta (mm)")+  
  theme_bw()
```

4.- Visualización del boxplot

BX



## Gráfico de barras

Se tiene que tener en cuenta que las variables a graficar tienen que ser cualitativas y deben estar configuradas en factores.

1.- Codificación de la variable en factores.

```
BD$especie<-factor(BD$especie,  
  levels=c("Adelie", "Gentoo", "Chinstrap"))
```

2.- Creación de un vector de colores. Dependiendo del número de factores que tenga nuestra variable, es el número de colores que vamos a crear. Los colores los puedes seleccionar desde el catálogo de RColors.

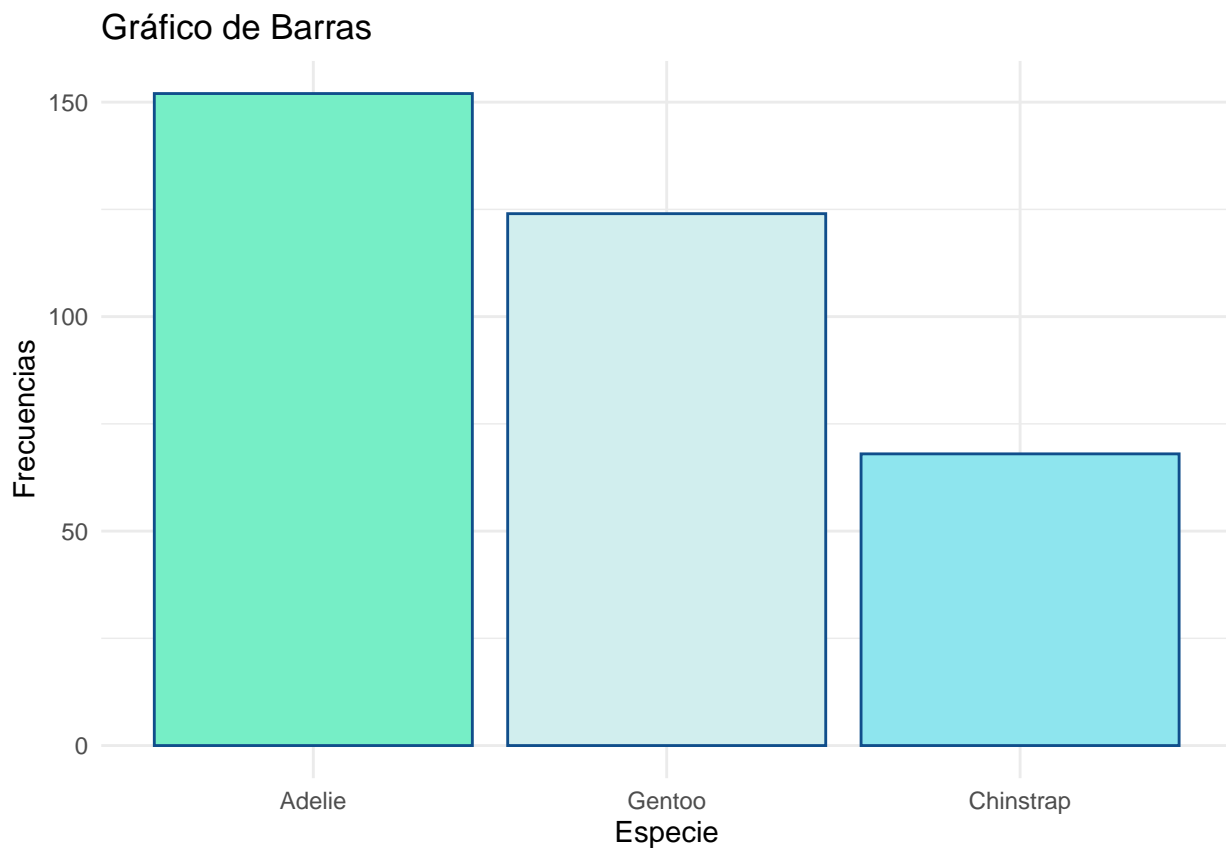
```
color=c("aquamarine2", "lightcyan2", "cadetblue2")
```

3.- Creación del gráfico.

```
GB1<-ggplot(BD, aes(x=especie))+  
  geom_bar(colour= "dodgerblue4", fill=color)+  
  ggtitle("Gráfico de Barras")+  
  xlab("Especie")+  
  ylab("Frecuencias")+  
  theme_minimal()
```

4.- Visualización del Gráfico.

GB1



## Histograma

Se tiene que cuenta que las variables que se van a graficar deben ser cuantitativas y se recomienda para más de 100 observaciones.

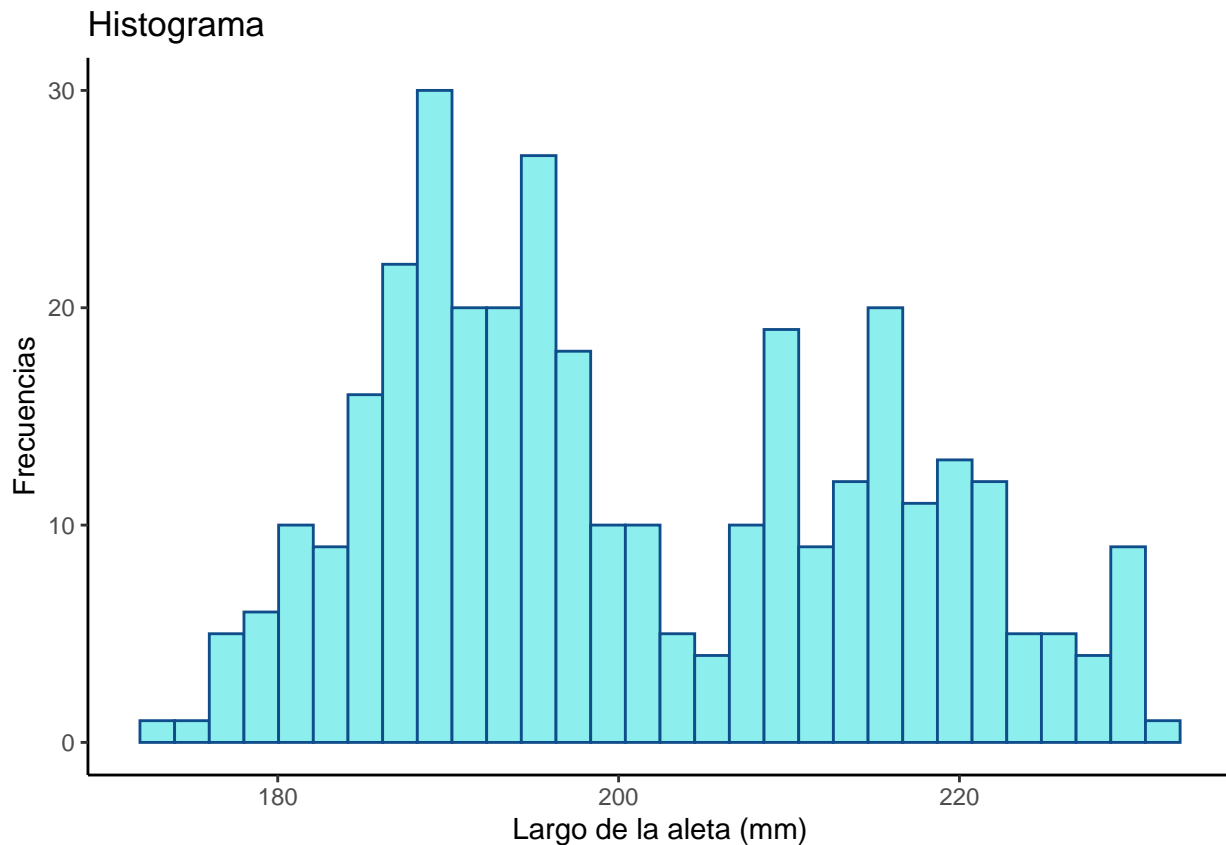
1.- Creación del gráfico.

```
HG<-ggplot(BD, aes(x=largo_aleta_mm))+  
  geom_histogram(col="dodgerblue4", fill="darkslategray2")+  
  ggtitle("Histograma")+  
  xlab("Largo de la aleta (mm)")+  
  ylab("Frecuencias")+  
  theme_classic()
```

2.- Visualización del gráfico.

HG

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



## Gráfico de dispersión.

Se utiliza para visualizar variables cuantitativas.

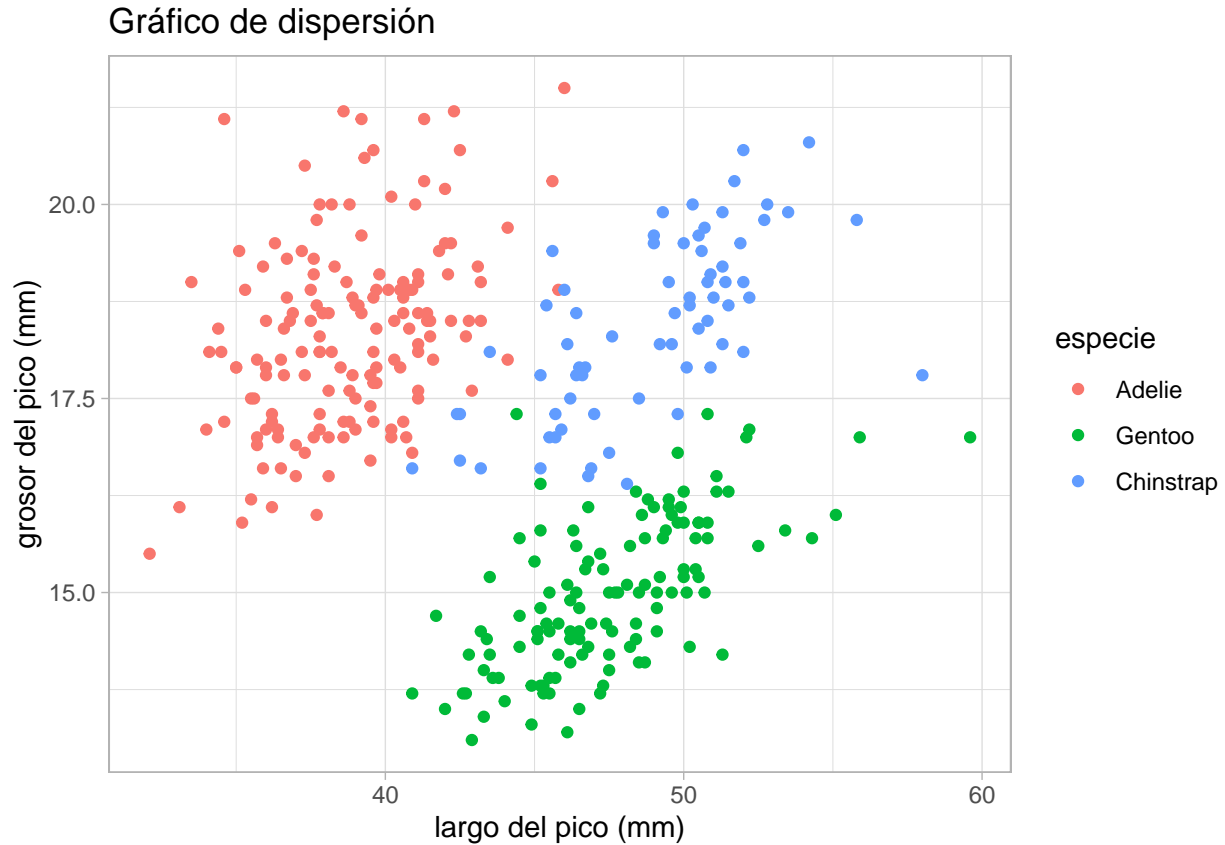
1.- Construcción del gráfico.

```
GD<-ggplot(BD, aes(x=largo_pico_mm, y=grosor_pico_mm))+  
  geom_point(aes(color=especie))+  
  ggtitle("Gráfico de dispersión")+  
  xlab("largo del pico (mm)")
```

```
ylab("grosor del pico (mm)") +  
theme_light()
```

2.- Visualización del gráfico.

GD



## Organización de gráficos

Se pueden organizar los gráficos de tal forma que se puedan mostrar dos objetos en una sola fila y como mejor se aprecie la información.

1.- Instalación de paquetes. No olvides declarar los argumentos `warning=FALSE` y `message=FALSE`

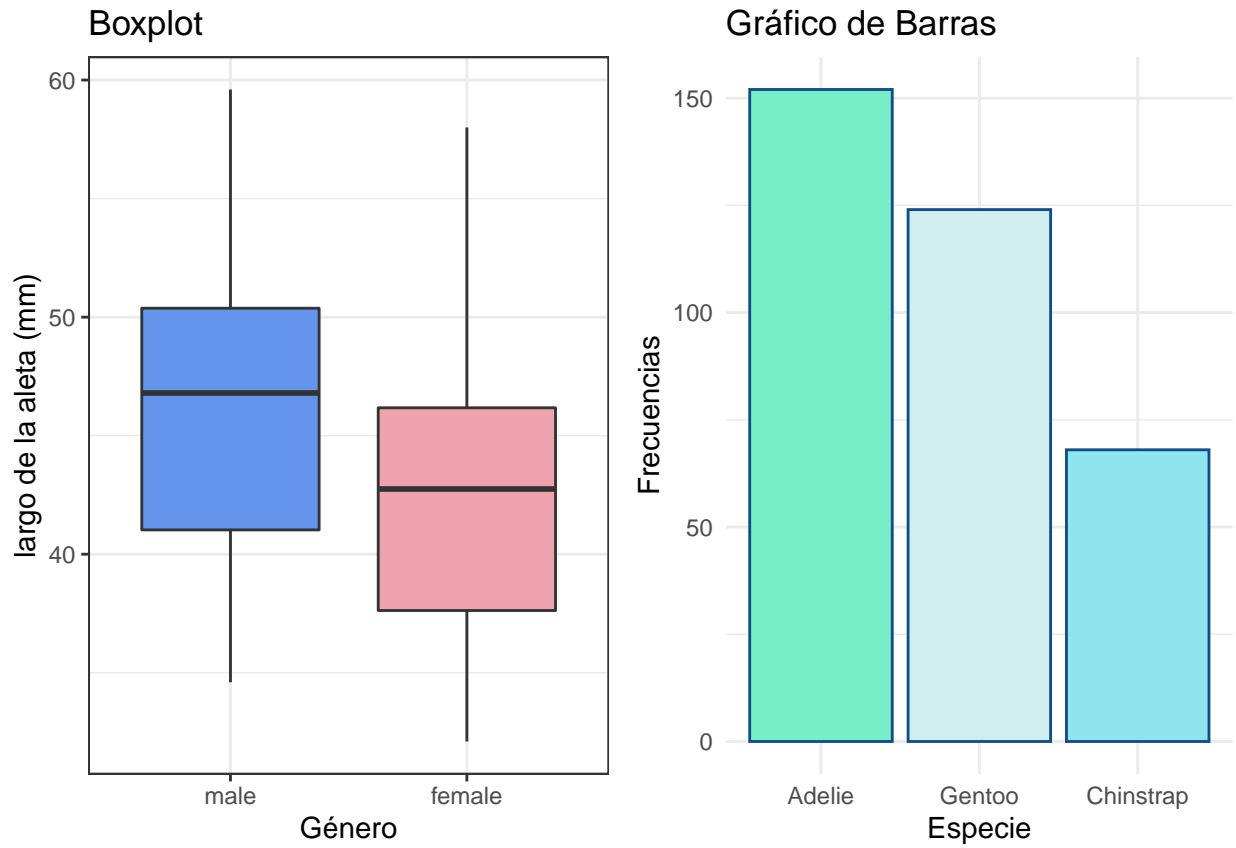
```
install.packages("gridExtra")
```

2.- Abrir librería.

```
library(gridExtra)
```

3.- Organización 2 gráficos en una fila y dos columnas.

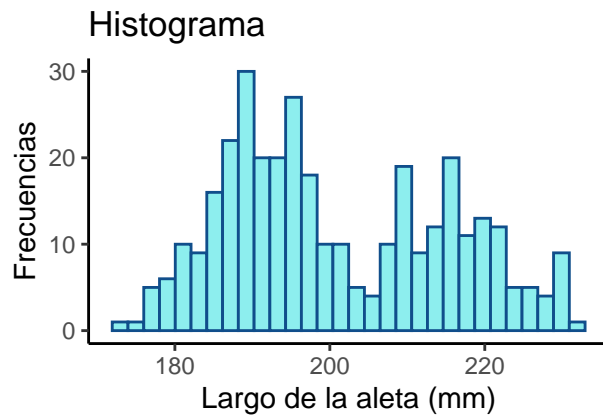
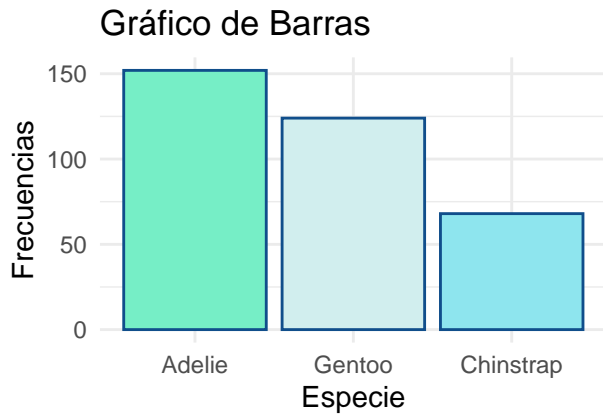
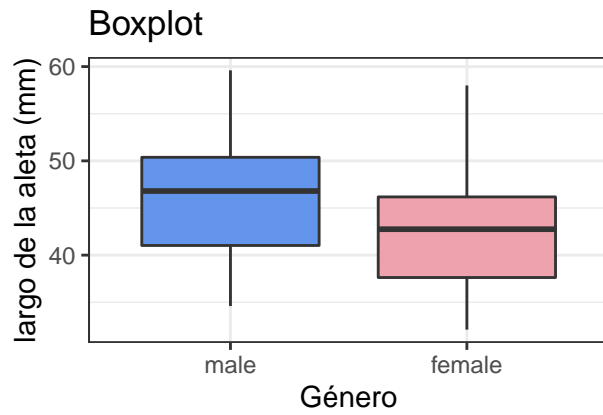
```
grid.arrange(BX,GB1, nrow=1, ncol=2)
```



4.- Organización 3 graficos en dos filas y dos columnas.

```
grid.arrange(BX,GB1,HG, nrow=2, ncol=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



5.- Organizacion 4 graficos en dos filas y dos columnas.

```
grid.arrange(BX,GB1,HG,GD, nrow=2, ncol=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

