

Plots

Julio

2023-03-12

```
options(repos = c(CRAN = "https://cran.rstudio.com"))
```

```
install.packages("carData")
```

```
## package 'carData' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\julio\AppData\Local\Temp\RtmpS81m2Y\downloaded_packages
```

```
install.packages("prettyR")
```

```
## package 'prettyR' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\julio\AppData\Local\Temp\RtmpS81m2Y\downloaded_packages
```

```
install.packages('latexpdf', repos= "http://cran.us.r-project.org")
```

```
## package 'latexpdf' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\julio\AppData\Local\Temp\RtmpS81m2Y\downloaded_packages
```

```
install.packages('tinytex', repos= "http://cran.us.r-project.org")
```

```
## package 'tinytex' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## C:\Users\julio\AppData\Local\Temp\RtmpS81m2Y\downloaded_packages
```

Queremos realizar una funcion que actue sobre un dataset que contenga variables categóricas y donde a cada variable categorica haya asociada una o varias variables numéricas . La funcion realizara la tarea de crear un boxplot recibiendo 5 inputs que seran dataset, variable numerica, variable categorica, color 1 y color 2

```
library(ggplot2)
```

```
mi_boxplot <- function(data, var_numerica, var_categorica, color1, color2) {
```

```

# Convertimos la variable categórica en un factor
data[[var_categorica]] <- factor(data[[var_categorica]])

# Definimos los colores para cada nivel de la variable categórica
n <- length(levels(data[[var_categorica]]))
col <- colorRampPalette(c(color1, color2))(n)

# Crear el boxplot
p <- ggplot(data, aes(x = !!sym(var_categorica), y = !!sym(var_numerica), fill = !!sym(var_categorica))) +
  geom_boxplot() +
  scale_fill_manual(values = col) +
  xlab(var_categorica) +
  ylab(var_numerica) +
  ggtitle(paste("Boxplot de ", var_numerica, " por ", var_categorica))

return(p)
}

```

Queremos realizar una función que actúe sobre un dataset que contenga variables categóricas y donde a cada variable categórica haya asociada una o varias variables numéricas. La función realizará la tarea de crear un swarmplot recibiendo 5 inputs que serán dataset, variable numérica, variable categórica, color 1 y color 2

```

library(ggplot2)

mi_violinplot <- function(data, var_numerica, var_categorica, color1, color2) {

  # Convertimos la variable categórica en un factor
  data[[var_categorica]] <- factor(data[[var_categorica]])

  # Definimos los colores para cada nivel de la variable categórica
  n <- length(levels(data[[var_categorica]]))
  col <- colorRampPalette(c(color1, color2))(n)
  relleno <- col

  # Crear el violinplot
  p <- ggplot(data, aes(x = !!sym(var_categorica), y = !!sym(var_numerica),
                        color = !!sym(var_categorica), fill = !!sym(var_categorica))) +
    geom_violin(size = 3, alpha = 0.7) +
    scale_color_manual(values = col) +
    scale_fill_manual(values = col) +
    xlab(var_categorica) +
    ylab(var_numerica) +
    ggtitle(paste("Swarmplot de ", var_numerica, " por ", var_categorica))

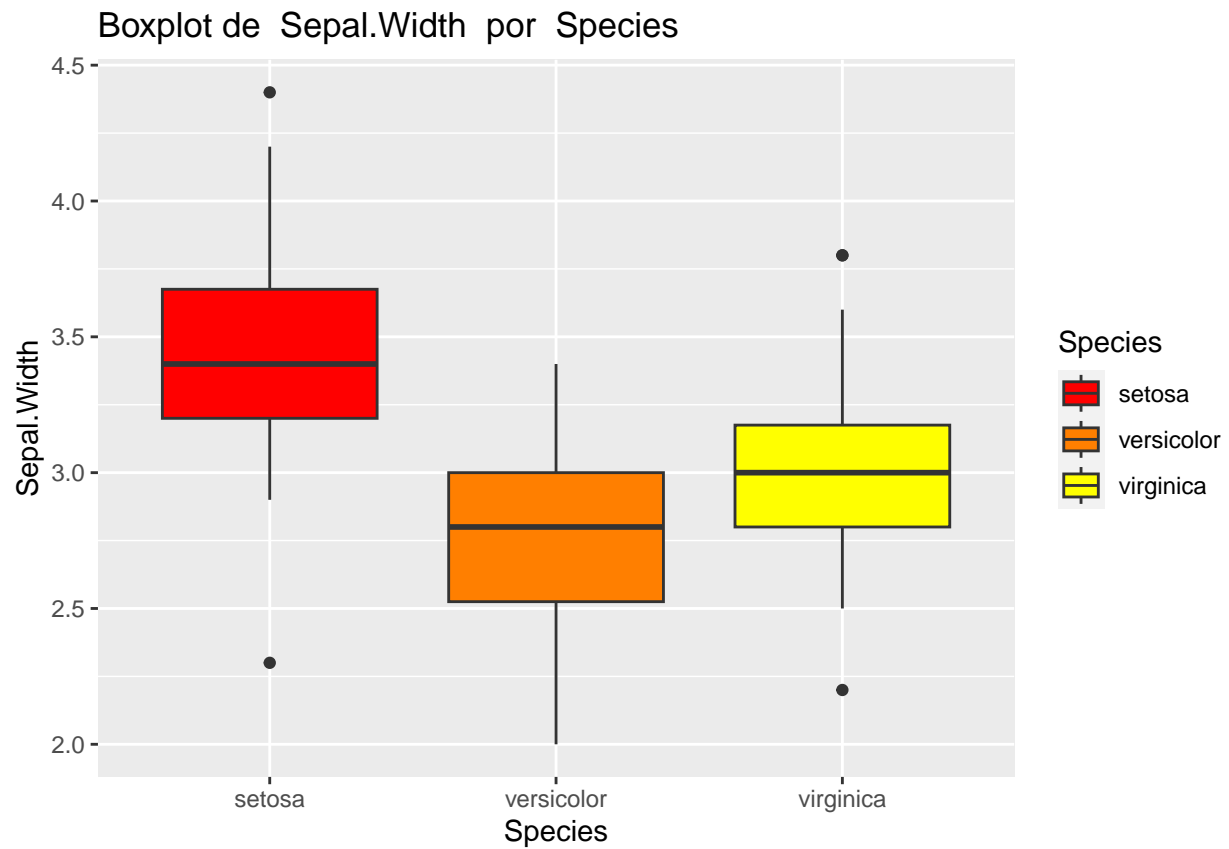
  return(p)
}

```

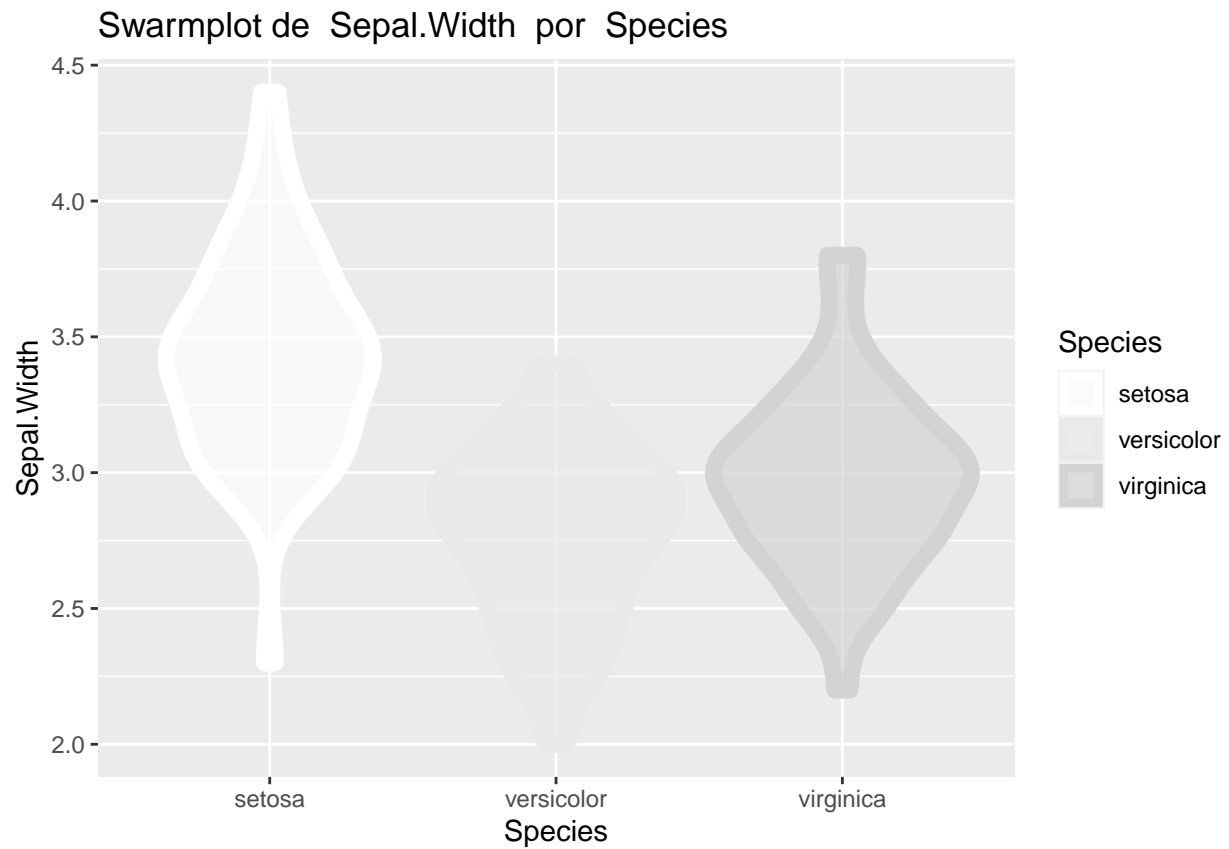
```

# Ejemplo 1: Dataset iris
mi_boxplot(iris, "Sepal.Width", "Species", "red", "yellow")

```



```
mi_violinplot(iris, "Sepal.Width", "Species", "white", "lightgrey")
```

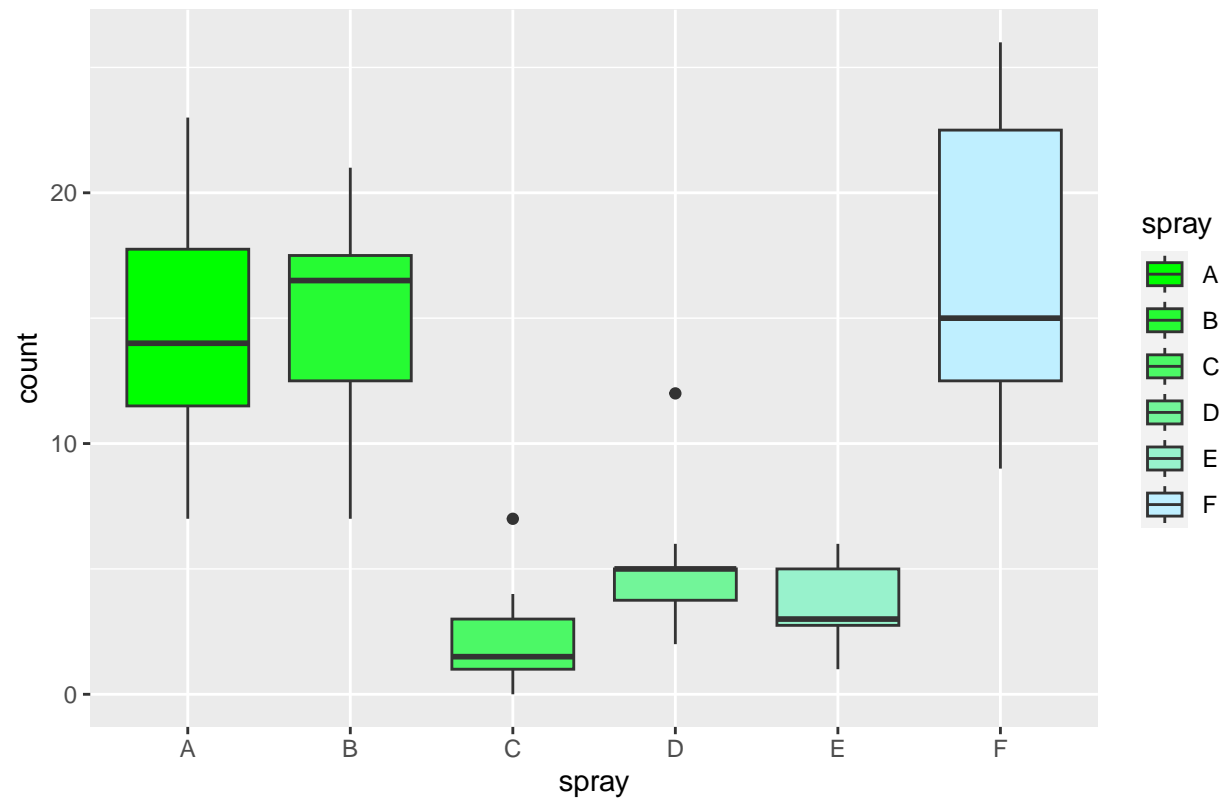


```
# Ejemplo 2: Dataset InsectSprays
head(InsectSprays)
```

```
##   count spray
## 1    10     A
## 2     7     A
## 3    20     A
## 4    14     A
## 5    14     A
## 6    12     A
```

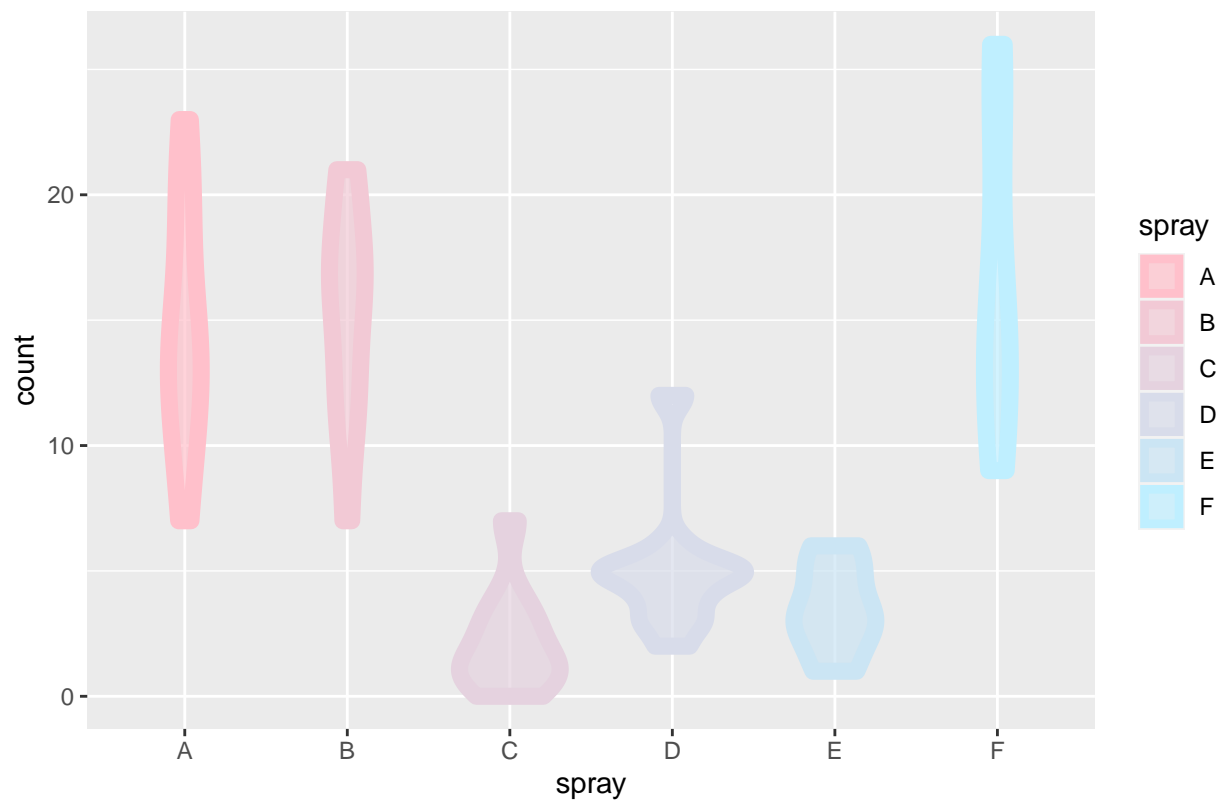
```
mi_boxplot(InsectSprays, "count", "spray", "green", "lightblue1")
```

Boxplot de count por spray



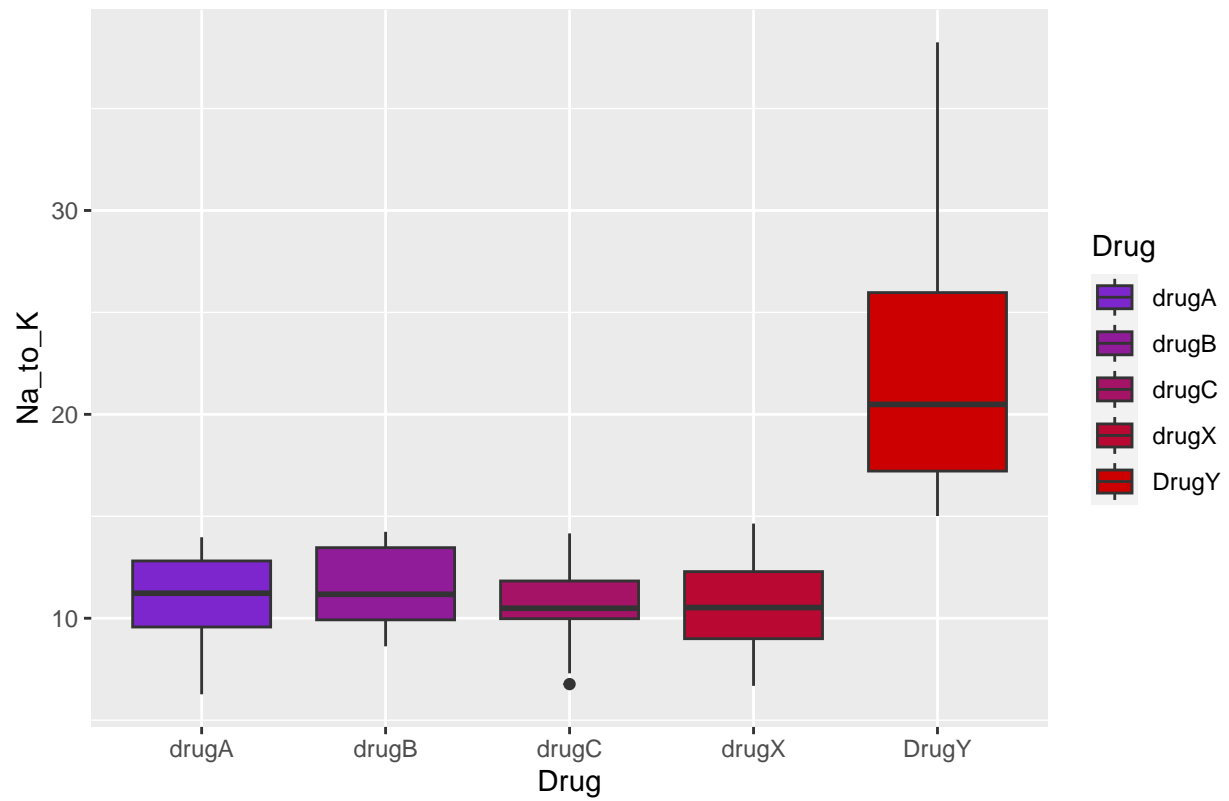
```
mi_violinplot(InsectSprays, "count", "spray", "pink", "lightblue1")
```

Swarmplot de count por spray



```
# Ejemplo 3: Drug200  
  
data = read.csv("Drug200.csv")  
#head(data)  
  
mi_boxplot(data, "Na_to_K", "Drug", "purple3", "red3")
```

Boxplot de Na_to_K por Drug



```
mi_violinplot(data, "Na_to_K", "Drug", "red3", "purple3" )
```

