WRS2

Julio Ladron de Guevara

2023-03-13

```
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\RtmpsdugpL\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\RtmpsdugpL\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\RtmpsdugpL\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\RtmpsdugpL\downloaded_packages
```

```
library(WRS2)
library(ggplot2)
```

Explorando el paquete WRS2 de R para obtener métodos estadísticos robustos

Robust Statistical Methods in R Using the WRS2 Package

Esta es una actividad realizada para analizar en profundidad los métodos estadísticos que aporta el paquete de R WRS2 disponible en CRAN. Esta actividad se basa en la publicación con doi: 10.18637/jss.v000.i00,

llamada “Robust Statistical Methods in R Using the WRS2 Package” de Patrick Mair y Rand Wilcox.

1) Introducción

La mayoría de los veces los datos sufren desviaciones en la normalidad. distribuciones sesgadas o asimétricas, datos atípicos o valores extremos, y distribuciones con colas pesadas o anchas. Es trivial que la media puede verse afectada por outliers o asimetrías. En estos casos han de usarse medidas robustas como la mediana o la media truncada (trimmed mean) y realizar los tests basandonos en las distribuciones muestrales de estas medidas rubustas.

Otra estrtegia que se realiza sobretodo cuando hay valores asimétricos es aplicar transformaciones como la logarítmica o la más sofisticada Box-Cox. Por ejemplo, al comparar las medias de 2 grupos con los datos con asimetría a la derecha, podemos pensar en aplicar transformaciones logaritmicas y comparar los dos grupos con un t-test, que es una prueba estadística paramétrica. El problema es que las distribuciones pueden mantenerse lo suficientemente asimétricas y dar resultados inexactos. Además se estarían comparando los logaritmos de las medias y no estas. Por lo que el resultado podría dejar de estar alineado con las hipótesis originales.

En general tenemos las siguientes opciones cuando hacemos inferencia con bases o colecciones de datos pequeñas:

- Nos podemos quedar con el marco de trabajo parametrico y establecer la distribución de muestreo bajo la hipótesis nula a través de estrategias de permutación. Una opción es utilizar el paquete “coin” en R. Sin embargo, las pruebas de permutación básicas no son satisfactorias para comparar medias o medianas y tienen limitaciones teóricas.
- Otra forma es recurrir a las pruebas estadísticas no paramétricas como el Mann-Whitney U-test, el Wilcoxon signed-rank y rank-sum test y Kruskall Wallis ANOVA. Sin embargo hay limitaciones para estas pruebas también. Por ejemplo cuando las distribuciones difieren, el Mann-Whithney U-test usa una estimación incorrecta del error estándar
- Los métodos robustos para la estimación y prueba estadística son una buena opción para tratar datos que no se comportan bien. La base matemática de estos métodos hace que no se hagan suposiciones acerca de la forma funcional de la distribución de probabilidad. En lugar de eso, se ven los parámetros como funcionales, y se utilizan expresiones para la función de influencia para encontrar las medidas de error estándar. Estos métodos tienen el potencial de aumentar la potencia estadística incluso en situaciones donde los métodos clásicos basados en la media y la varianza no funcionan bien, y proporcionan una comprensión más precisa y profunda de los datos en comparación con las técnicas clásicas. *La **función de influencia** es un concepto clave en la teoría de la robustez. Se refiere a una función matemática que describe cómo un cambio pequeño en un punto de datos afecta la estimación de un parámetro de interés*

A continuación estudiaremos los métodos robustos del paquete WRS2

2) Medidas robustas de posición

Una alternativa robusta a la media aritmética es la **media truncada**, que descarta cierto porcentaje a ambas colas de la distribución. Por ejemplo una media truncada al 20%, elimina el 20% de los valores a derecha y el 20% de los valores a izquierda de la distribución. En R, esto puede obtenerse añadiendo el argumento trim a la función *mean*. Por ejemplo mean(vector, 0.1). Nótese que si el argumento trim = 0.5, el valor resultante es la mediana, que es otra medida de posición rubusta.

Otra alternativa robusta a la media es la **media winsorizada** o Winsorized mean. El proceso consistente en dar menor peso a las observaciones en las colas y mayor a las observaciones centrales, se llama winsorizar. El nivel de winsorizacion ha de ser elegido a priori. La función se llama *winmean*.

M-estimadores son una familia general de medidas robustas de ubicación. La idea detrás de los M-estimadores es encontrar una medida de ubicación que sea robusta a valores atípicos y que sea calculada a partir de una función de pérdida a minimizar. La función de pérdida se define como una función que mide la discrepancia entre un estimador y el verdadero valor de la ubicación. En este caso, se busca una función de pérdida que sea adecuada para la distribución de los datos y que tenga una solución única. La solución se encuentra minimizando la función de pérdida, lo que lleva a la estimación de la ubicación que mejor se ajusta a los datos. En el caso más simple, la función de pérdida puede ser la suma de los cuadrados de las desviaciones de los datos a la ubicación estimada. Los M-estimadores son muy flexibles y pueden adaptarse a diferentes situaciones de datos.

K es un parámetro que se utiliza en la función de propuesta por Huber para su estimador M-estimador. Esta constante influye en la forma en que la función de distancia penaliza las observaciones que están lejos de la ubicación central estimada. Un valor de K más grande indica una mayor tolerancia a las observaciones atípicas (outliers), mientras que un valor más pequeño indica una mayor sensibilidad a las mismas. En el caso de Huber, el valor propuesto para K es de 1.28. Para la media se usa la funcion *mest*

Ejemplos:

- Media y error estandar con la distribucion truncada al 10%

```
timevec <- c(77, 87, 88, 114, 151, 210, 219, 246, 253, 262, 296, 299, 306, 376,
             428, 515, 666, 1310, 2611)

mean(timevec, 0.1)
```

```
## [1] 342.7059
```

```
trimse(timevec, 0.1)
```

```
## [1] 103.2686
```

```
median(timevec)
```

```
## [1] 262
```

```
mean(timevec, 0.5)
```

```
## [1] 262
```

- Media y error estandar con la distribucion Winsorizada al 10%

```
winmean(timevec, 0.1)
```

```
## [1] 380.1579
```

```
winse(timevec, 0.1)
```

```
## [1] 92.9417
```

- M-Estimador Huber con constante k=1.28

```
mest(timevec)
```

```
## [1] 285.1576
```

```
mestse(timevec)
```

```
## [1] 52.59286
```

Estrategias para t-test y ANOVA robustos

Tests de medidas de posicion de dos grupos independientes. En 1974, Yuen propuso un estadístico de prueba para realizar una comparación de medias truncadas entre dos muestras que permite tener en cuenta varianzas desiguales. El estadístico de prueba propuesto por Yuen para comparar medias recortadas de dos muestras sigue una distribución t bajo la hipótesis nula H0: $\mu_{t1} = \mu_{t2}$

Usaremos el test de Yuen para ver si hay una diferencia significativa entre los goles marcados en la Liga Española y la Bundesliga. Haremos esto porque los goles marcados por equipos alemanes siguen una distribucion más simétrica que la española, que se ve afectada por la mayor cantidad de goles de Real Madrid, Barcelona y Atlético. La función es *yuen* y el trim por defecto es 0.2.

```
data = eurosoccer
# Seleccionamos las ligas que nos interesan
spain_germany = data [which (data$League == "Spain" | data$League == "Germany"), ]
#Eliminamos los niveles que no utilizaremos
spain_germany$League = droplevels(spain_germany$League)
levels (spain_germany$League)
```

```
## [1] "Spain"   "Germany"
```

```
yuen(GoalsGame ~ League, data = spain_germany)
```

```
## Call:
## yuen(formula = GoalsGame ~ League, data = spain_germany)
##
## Test statistic: 0.8394 (df = 16.17), p-value = 0.4135
##
## Trimmed mean difference:  -0.11494
## 95 percent confidence interval:
## -0.405     0.1751
##
## Explanatory measure of effect size: 0.14
```

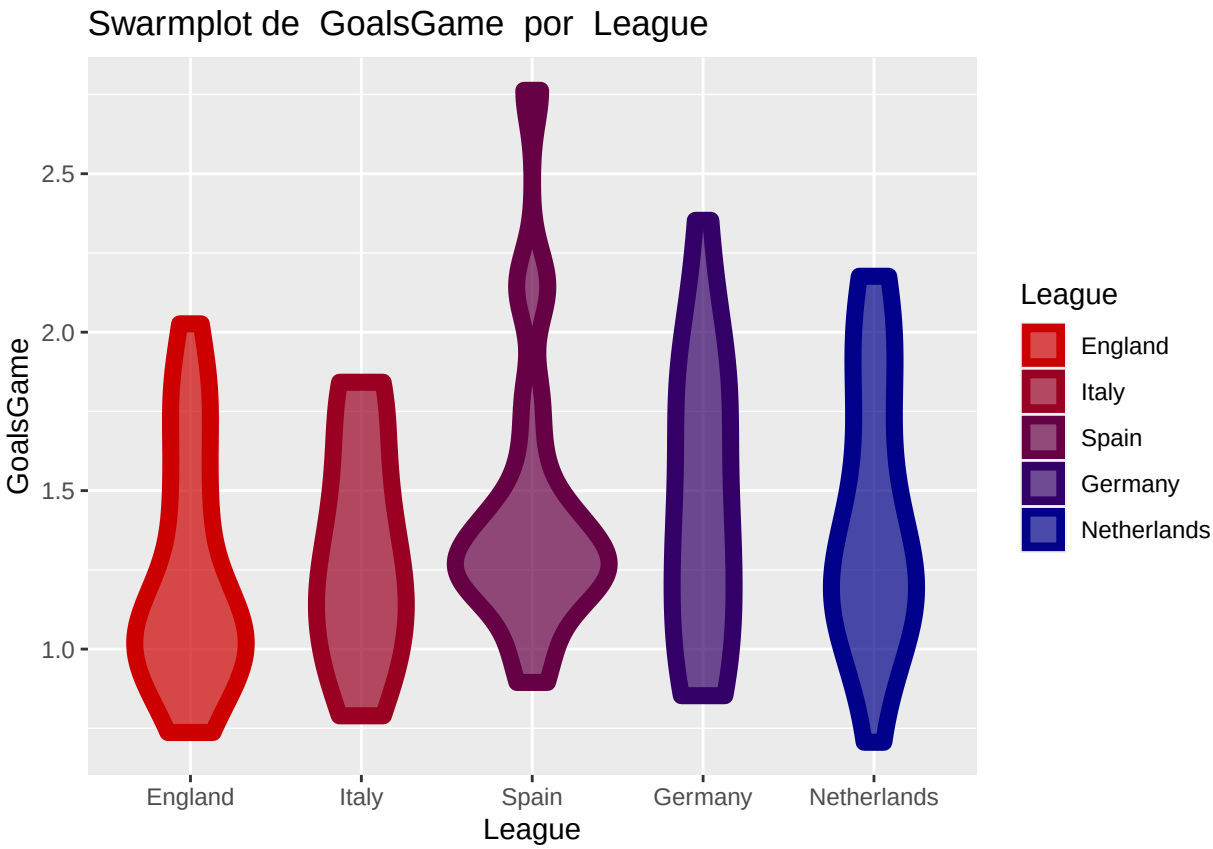
El resultado del test sugiere que no hay diferencias significativas entre las medias truncadas de estas dos ligas.

Para hacer los boxplots usaré las funciones que he creado y están en mi cuenta de GitHub en el siguiente enlace: https://github.com/julioldg/My-functions-in-R/tree/main/Boxplot_violinplot

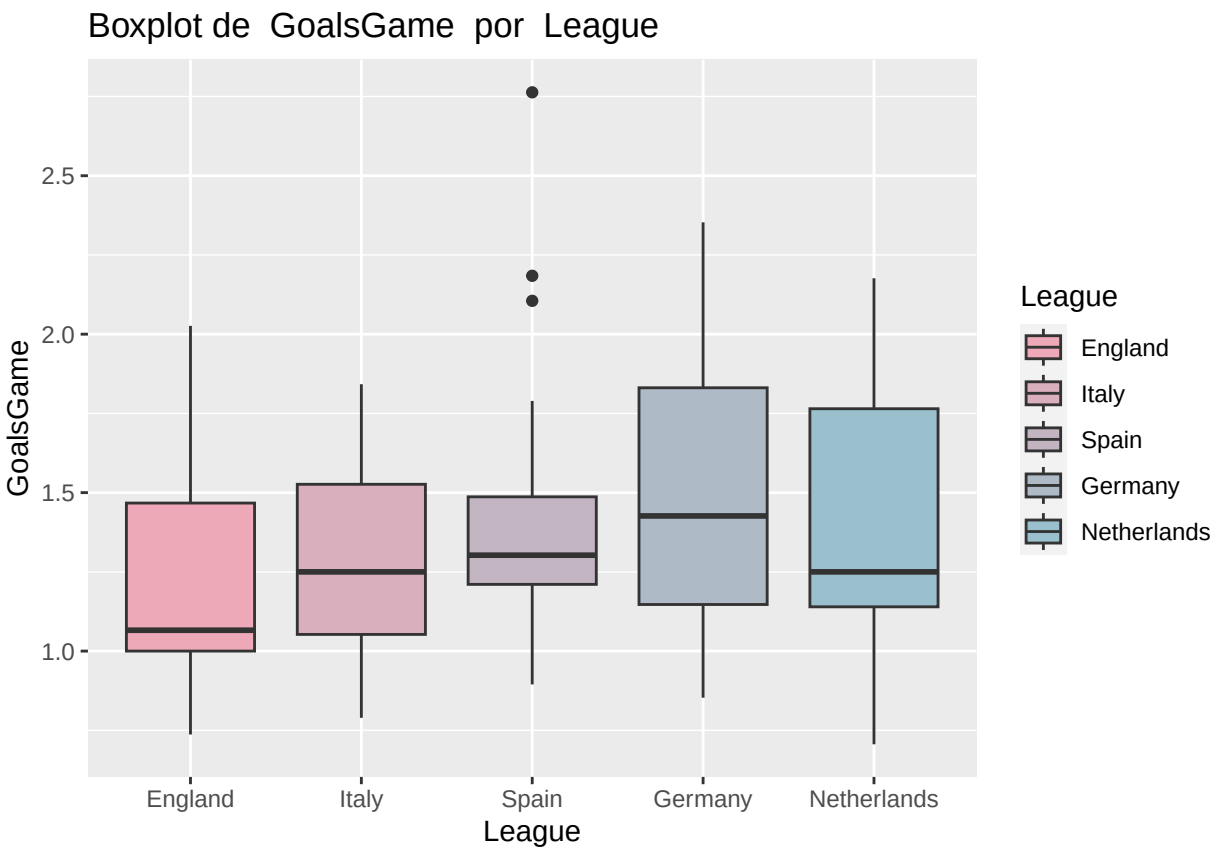
```r
mi_boxplot <- function(data, var_numerica, var_categorica, color1, color2) {

  # 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)
}
```

```r
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)
}
```

```r
mi_violinplot(data, "GoalsGame", "League", "red3", "blue4")
```




```
mi_boxplot(data, "GoalsGame", "League", "pink2", "lightblue3")
```



Si queremos hacer este tipo de análisis, pero basándonos en métodos con M-estimadores utilizaremos la función *pbgen*. Cuando est = "median" Es relativo a diferencias de mediana y cuando est = "onestep" Es relativo al estimador de Huber

```
data = eurosoccer
# Seleccionamos las ligas que nos interesan
spain_germany = data [which (data$League == "Spain" | data$League == "Germany"), ]
#Eliminamos los niveles que no utilizaremos
spain_germany$League = droplevels(spain_germany$League)
levels (spain_germany$League)
```

```
## [1] "Spain"   "Germany"
```

```
pb2gen(GoalsGame ~ League, data = spain_germany, est = "median" )
```

```
## Call:
## pb2gen(formula = GoalsGame ~ League, data = spain_germany, est = "median")
##
## Test statistic: -0.1238, p-value = 0.46745
## 95% confidence interval:
## -0.5325    0.2214
```

```
pb2gen(GoalsGame ~ League, data = spain_germany, est = "onestep")
```

```
## Call:
```

```
## pb2gen(formula = GoalsGame ~ League, data = spain_germany, est = "onestep")
##
## Test statistic: -0.1181, p-value = 0.48414
## 95% confidence interval:
## -0.3509    0.1723
```

Debido a que los p-value no son lo suficientemente pequeños, la diferencia de medias no es estadistiacamente significativa.

Comparación unidireccional entre varios grupos A menudo se dice que las pruebas F son bastante robustas contra las violaciones de normalidad. Sin embargo, esto no siempre es así. De hecho, escenarios elaborados en Games (1984), Tan (1982), Wilcox (1996) y Cressie y Whitford (1986) muestran que las cosas pueden salir mal al aplicar ANOVA en situaciones donde tenemos distribuciones con colas pesadas, tamaños de muestra desiguales y cuando las distribuciones difieren en sesgo. La transformación de los datos tampoco es una alternativa muy atractiva porque, en condiciones generales, esto no resuelve efectivamente los problemas de sesgo o valores atípicos.

La primera alternativa robusta al ANOVA presentada aquí es una comparación unidireccional de múltiples medias de grupo recortadas, implementada en la función *t1way*. Sea J el número de grupos. La hipótesis nula correspondiente es:

$$H_0 : \mu_{t1} = \mu_{t2} = \cdots = \mu_{tJ}$$

Algo similar hace la función *med1way*, pero comparando las medianas

Los aplicamos al dataset eurosoccer, sabiendo que España posee outliers, Inglaterra y Holanda distribucion hacia la derecha y Alemania e Italia, simétrica.

```
t1way(GoalsGame ~ League, data = eurosoccer)
```

```
## Call:
## t1way(formula = GoalsGame ~ League, data = eurosoccer)
##
## Test statistic: F = 1.1178
## Degrees of freedom 1: 4
## Degrees of freedom 2: 26.95
## p-value: 0.36875
##
## Explanatory measure of effect size: 0.3
## Bootstrap CI: [0.14; 0.51]
```

```
med1way(GoalsGame ~ League, data = eurosoccer)
```

```
## Call:
## med1way(formula = GoalsGame ~ League, data = eurosoccer)
##
## Test statistic F: 1.2335
## Critical value: 2.2442
## p-value: 0.217
```

Ninguno de los métodos sugiere que haya diferencias significativas.

La funcion *lincon* ilustra una comparativa entre todas las variables, con un valor trim predeterminado de 0.1, lo podemos cambiar

```
lincon(GoalsGame ~ League, data = eurosoccer)
```

```
## Call:
## lincon(formula = GoalsGame ~ League, data = eurosoccer)
##
##                            psihat ci.lower ci.upper p.value
## England vs. Italy       -0.11184 -0.51061  0.28692 0.72607
## England vs. Spain       -0.17105 -0.50367  0.16157 0.72607
## England vs. Germany     -0.28599 -0.75439  0.18241 0.72027
## England vs. Netherlands -0.22472 -0.69088  0.24145 0.72607
## Italy vs. Spain         -0.05921 -0.41380  0.29538 0.72607
## Italy vs. Germany       -0.17415 -0.65496  0.30666 0.72607
## Italy vs. Netherlands   -0.11287 -0.59157  0.36583 0.72607
## Spain vs. Germany       -0.11494 -0.55124  0.32136 0.72607
## Spain vs. Netherlands   -0.05366 -0.48748  0.38015 0.72607
## Germany vs. Netherlands  0.06127 -0.47101  0.59356 0.72607
```

```
## lincon(GoalsGame ~ League, data = eurosoccer, trim = 0.2)
```

Comparaciones robustas para diseños factoriales de orden superior En el análisis de diseños factoriales de orden superior con dos factores, en los cuales el primer factor presenta J categorías y el segundo factor presenta K categorías, es posible realizar comparaciones de medias recortadas y de medianas mediante el uso de las funciones *t2way* y *med2way*, respectivamente. Estas funciones permiten generalizar las comparaciones unidireccionales de medias y medianas a diseños factoriales de dos vías de orden superior. Es importante destacar que la utilización de estas técnicas robustas permitirá obtener resultados precisos y confiables, aún en presencia de posibles violaciones de los supuestos de normalidad y homogeneidad de varianzas en los datos.

Para diseños factoriales de dos vías que involucran M-estimadores más generales, se puede utilizar la función *pbad2way*.

El conjunto de datos utilizado como ejemplo es el "beer goggles dataset" de Field, Miles y Field (2012). Este conjunto de datos se centra en los efectos del alcohol en la selección de parejas en clubs nocturnos. La hipótesis es que después de consumir alcohol, las percepciones subjetivas de la atracción física se volverán más inexactas (efecto de las gafas de cerveza). En este conjunto de datos, se tienen en cuenta dos factores: género (24 estudiantes masculinos y 24 estudiantes femeninos) y cantidad de alcohol consumido (ninguno, 2 pintas, 4 pintas). Al final de la noche, el investigador tomó una fotografía de la persona con la que el participante estaba hablando. El atractivo de la persona en la fotografía fue evaluada por jueces independientes en una escala del 0 al 100 (variable de respuesta).

```
data = goggles

t2way(attractiveness ~ gender*alcohol, data = data)
```

```
## Call:
## t2way(formula = attractiveness ~ gender * alcohol, data = data)
##
##                   value p.value
## gender           1.6667   0.209
## alcohol         48.2845   0.001
## gender:alcohol 26.2572   0.001
```

```
med2way(attractiveness ~ gender*alcohol, data = data)
```

```
## Call:
## med2way(formula = attractiveness ~ gender * alcohol, data = data)
##
##    value       df p.value
## 1 0.5587 F(1,Inf)  0.4548
## 2 9.1983 F(2,Inf)  0.0001
## 3 9.6640 Chisq(2)  0.0080
```

```
pbad2way(attractiveness ~ gender*alcohol, data = data, est = "onestep")
```

```
## Call:
## pbad2way(formula = attractiveness ~ gender * alcohol, data = data,
##     est = "onestep")
##
##                p.value
## gender          0.1686
## alcohol         0.0000
## gender:alcohol  0.0000
```

En la fila gender no ha salido resultado significativo, por lo que no podemos rechazar la hipótesis nula de que el atractivo se percibe de la misma manera según el género

En la fila alcohol, ha salido resultado significativo para todas, por lo tanto rechazamos la hipótesis nula de que las personas que han bebido alcohol perciben a las personas menos atractivas igual que las que no han bebido.

En la fila gender:alcohol, ha salido resultado significativo, por lo que la forma en la cual cambia la percepcion es diferente entre hombres y mujeres a mas alcohol consuman

Diseños de medidas repetidas En este apartado se explica el diseño más simple de medidas repetidas, el cual corresponde a un escenario de prueba de muestras emparejadas. El test de la media recortada de Yuen puede ser generalizado para ajustarse a diseños dependientes (es decir, diseños de sujetos relacionados). La función correspondiente en R se llama *yuend*

Se presenta un conjunto de datos de prueba del paquete MASS que incluye pares de datos de peso de niñas antes y después del tratamiento para la anorexia. Se utiliza un subconjunto de 17 niñas sometidas a tratamiento familiar. Se muestra una gráfica con las trayectorias individuales de peso. Se observa que para cuatro niñas el tratamiento no pareció ser efectivo, mientras que para las restantes se registró un aumento de peso. La prueba de muestras emparejadas sobre las diferencias de medias recortadas muestra un efecto significativo del tratamiento, lo que indica que, en general, el tratamiento fue efectivo (y el tamaño del efecto puede considerarse “grande”).

```
library(MASS)

data("anorexia")

data = anorexia

anorexia_FT = subset (anorexia,
                      subset = Treat == "FT")
yuend(anorexia_FT$Prewt,
      anorexia_FT$Postwt)
```

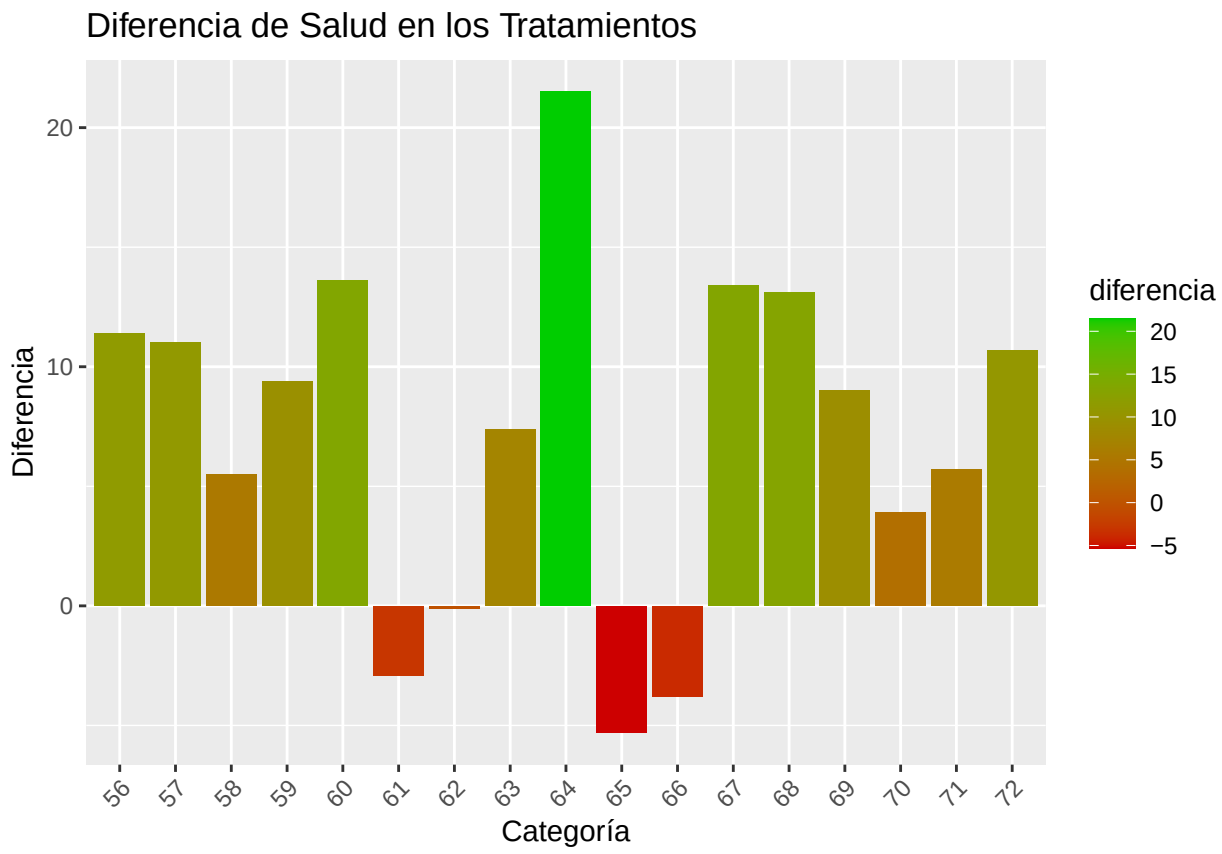
```
## Call:
## yuend(x = anorexia_FT$Prewt, y = anorexia_FT$Postwt)
##
## Test statistic: -3.829 (df = 10), p-value = 0.00332
##
## Trimmed mean difference:  -8.56364
## 95 percent confidence interval:
## -13.5469     -3.5804
##
## Explanatory measure of effect size: 0.6
```

```
with(anorexia_FT, yuend(Prewt, Postwt))
```

```
## Call:
## yuend(x = Prewt, y = Postwt)
##
## Test statistic: -3.829 (df = 10), p-value = 0.00332
##
## Trimmed mean difference:  -8.56364
## 95 percent confidence interval:
## -13.5469     -3.5804
##
## Explanatory measure of effect size: 0.6
```

Es decir el p-valor es $0.00332 < 0.05$, por lo tanto el tratamiento se considera significativo y se rechaza la hipótesis nula de que la ganacia de peso es igual con o sin el tratamiento

```
library(ggplot2)

tratamiento <- function(dataset, antes, despues, color1, color2){
  # Calcular la diferencia entre las variables "antes" y "después"
  dataset$diferencia <- dataset[[despues]] - dataset[[antes]]

  # Crear el gráfico de barras con escala de color
  ggplot(data = dataset, aes(x = rownames(dataset),
                             y = diferencia, fill = diferencia)) +
    geom_bar(stat = "identity") +
    xlab("Categoría") +
    ylab("Diferencia") +
    ggtitle("Diferencia de Salud en los Tratamientos") +
    scale_fill_gradient(low = color1, high = color2) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

tratamiento(anorexia_FT, "Prewt", "Postwt","red3","green3")
```


Diferencia de Salud en los Tratamientos
Diferencia
20
10
0
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
Categoría
diferencia
20
15
10
5
0
−5