Funciones examen (Normality, trimmed)

No copiarla tal cual, que nos joden en el examen

Normality

→Calcular si un vector (o conjunto de datos) es normal, luego valora qué opción es mejor para calcularlo en relación al número de variables que tiene el vector

💡 If n>3 and n<5000 → Shapiro-Wilks
• else Lilliefor's correction for Kolmogorov-Smirnoff
REMEMBER: Install nortest package using install.packages("nortest")

```
install.packages("nortest")
library(nortest)
nortest::lillie.test()



normality <- function(x, na.rm=TRUE) {

  n <- sum(!is.na(x))

  if (n > 3 & n < 5000) {
    p1 <- shapiro.test(x)
    if (p1$p > 0.05) {
      message("Es normal")
      cat("Shapiro-Wilk Test", "\n p-value:", p1$p, "\n Mean:",
    } else {
      message("No es normal")
```

```
      cat("Shapiro-Wilk Test", "\n p-value:", p1$p, "\n Median:"
    }
  } else {
    p2 <- nortest::lillie.test(x)
    if (p2$p > 0.05) {
      message("Es normal")
      cat("Lilliefor's correction K-S","\n p-value:", p2$p, "\n
    } else {
      message("No es normal")
      cat("Lilliefor's correction K-S", "\n p-value:", p2$p, "\n
    }
  }
}
```

Para ejecutarlo creamos una variable con los datos a procesar y lo metemos dentro de la función

```
mis_datos <- c(10, 15, 20, 25, 30)
normality(mis_datos)
#Es normal
#Saphiro-Wilks Test
#p: 0.9671739
#Mean 20
#Standard deviation: 7.905694
```

PROFESOR

```
normality <- function(valores, alpha = 0.5) {
norm <- shapiro.test(valores)
result <- data.frame(p.value = norm$p.value)
result$is.normal <- result$p.value > alpha
result$statistic <- norm$statistic
result$statistic_name <- names(norm$statistic)
result
```

```
}
## normality(valores, 0.05)
##      p.value is.normal statistic statistic_name
## 1 0.02080657     FALSE 0.9200127              W
```

```
desc_num <- function(valores, alpha = 0.05, na.rm = TRUE){
 result <- data.frame(
     mean = mean(valores, na.rm = na.rm),
     sd = sd(valores, na.rm = na.rm),
     median = median(valores, na.rm = na.rm),
     IQR = IQR(valores, na.rm = na.rm),
     min = min(valores, na.rm = na.rm),
     max = max(valores, na.rm = na.rm)
     )
     result <- cbind(result, normality(valores, alpha))
     texto <- "es normal usamos media y sd"
     if (!result$is.normal) texto = "NO es normal por lo que usa
     cat("\n", texto, "\n")
     result
     }
     ##desc_num(valores, 0.05, TRUE)
##NO es normal por lo que usamos la mediana y el rango intercuar
##      mean       sd median     IQR  min max    p.value is.norm
##1 230.7219 123.9387  196.3 205.175 71.1 472 0.02080657     FAL
##   statistic_name
##1              W
```

Trimmed

```
trimmed <- function(values, trim = 0.1) {
trim.p <- trim
trim.n <- ceiling(length(values)*trim.p) #... number of items th
final.values <- sort(values)[-(1:trim.n)] #... remove the lower
```

```
final.values <- sort(values,decreasing=TRUE)[-(1:trim.n)] #... r
trim_mean <- mean(final.values)
names(trim_mean) <- paste("Mean trimmed ",trim.p*100,"%", sep =
return(trim_mean)
}
trimmed(DATA$qsec, trim = 0.2)

#Mean trimmed 20%
#         17.1688
```