Lab04 - Graphs and Comparisons

```
#1.Data frame manipulation
  #Wide to long
    DATA <- data.frame(ID = 1, SEX = "M", ER_REASON_V1 = "COVID"
    DATA <- rbind(DATA, c(2, "F", "Heartattack", 15, "COVID", 10
    DATA <- rbind(DATA, c(3, "F", "CarAccident", 53, "Flu", 8))
    DATA <- rbind(DATA, c(4, "M", "Flu", 7, "COVID", 10))
    DATA <- rbind(DATA, c(5, "M", "COVID", 12, "Fracture", 4))
    DATA <- rbind(DATA, c(6, "F", "Fracture", 2, NA, NA))
    #DATA_LONG <- reshape(DATA, direction = finaldirectionofdata
    "", "", ""), timevar = "time indicator",
    times = how visitswillbeindicated1:2, v.names = c("finalname

    #Change id name
    names(DATA_LONG)[1] <- "ID_SUBJ"
    DATA_LONG$ID <- 1:nrow(DATA_LONG)
    DATA_LONG <- DATA_LONG[, c(ncol(DATA_LONG), 1:(ncol(DATA_LON

    #Variable class can change
    class(DATA_LONG$ER_TIME)
    DATA_LONG$ER_TIME <- as.numeric(DATA_LONG$ER_TIME)

    #Describe reasons people have gone to ER
    t1 <- table(unlist(DATA_LONG[, "ER_REASON"]))
    t2 <- data.frame(cbind(t1, prop.table(t1) * 100))
    names(t2) <- c("n", "%")
    t2

  #Long to wide: now we want to get people and not ER
  DATA_WIDE <- reshape(DATA_LONG[, -1], direction = "wide", idva
```

```
#3.R Markdown: Decia que en el examen habra que entregar un pdf
                              #tampoco tenia mucho misterio
  #YAML header, regular text, code chunks
  #Code chunks
  ```{r chunk_name, chunk_options}
    #here you put R code
  ```
    #Chunk options: eval(Markdown if you wanna run code)
                    #echo(Show code)
                    #Results='asis' (Will print code output as i
                    #Results='default'(Will show output as in co

#4.Graphics: we are going to use them now to compare variables
  #Numerical VS Categorical
    #Statistical view: comparing age of 2 independent populatio
    #Analysis view: compare age variable against gender
    ggplot(DATA, aes(x = am, y = mpg, fill = am)) + geom_boxplo
    ggplot(DATA, aes(x = am, y = mpg, fill = am)) + geom_violin
    #Remove legenda theme(legend.position =“none”)
      #Hay un monton de cosas mas
  #Categorical VS Categorical
    #To compare 2 categorical variables we use Pearson’s chi² te
    t1 <- table(unlist(DATA[, "am"]))
    t2 <- data.frame(rbind(prop.table(t1, margin = 1) * 100))
    t1 <- data.frame(rbind(t1))
    names(t1) <- paste0(names(t1), "_n")
    names(t2) <- paste0(names(t2), "_%")
    t.final <- cbind(t1, t2)[, order(c(2 * (seq_along(t1) - 1) +
                                       1, 2 * (seq_along(t2))
    t.final

    There are 0 cells with expected values < 5
    Pearson's Chi-squared test with Yates' continuity correctio
    data: DATA$vs and DATA$am
```

```
    X-squared = 0.34754, df = 1, p-value = 0.5555

    #Now is when we are supossed to use those tests, since p is
    ggplot(DATA, aes(x = am, fill = vs)) + geom_bar(position = '
    #Now with porcentages
    ggplot(DATA, aes(x = am, fill = vs)) + geom_bar(aes(y =((..c

  #Numeric VS Numeric: We can use paired means comparison(parame
    #Paired means: We will use datawide for all statistics and c
      shapiro.test(DATA_WIDE$ER_TIME_V1)
      comparison <- wilcox.test(DATA_WIDE$ER_TIME_V1, DATA_WIDES
      #Ahora los gráficos
      ggplot(DATA_LONG, aes(x = factor(VISIT), y = ER_TIME, fill
      factor(VISIT))) +
      geom_violin(alpha = 0.3) + geom_boxplot(alpha = 0.5, width
      geom_jitter(alpha = 0.7, width = 0.1) + theme(legend.posit
      round(comparison$statistic, digits = 3)), parse = TRUE, hj

    #Non paired
      #Correlations. Recomienda el corrplot package. Ejemplo de
      shapiro.test(mtcars$hp)
      shapiro.test(mtcars$mpg)
      #Sale que no es normal, asi que usa Spearmans
      cor(mtcars$hp, mtcars$mpg, method = "spearman")
      cor.test(mtcars$hp, mtcars$mpg, method = "spearman")
      cor <- cor(mtcars, method = "spearman")
      corrplot::corrplot(cor, method = "color/pie",type="lower",
      #Todo lo que esta arriba entre comas se lo puedes añadir c

      #Point diagram: Common to show relation between 2 numeric
      ggplot(mtcars, aes(x = hp, y = mpg)) + geom_point()+geom_s
```