Lab2ANOVA

#install.packages("tidyverse")  
#install.packages("plotmeans")  
#install.packages("gplots")  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.3 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

### cargamos datos

my\_data <- PlantGrowth  
my\_data

## weight group  
## 1 4.17 ctrl  
## 2 5.58 ctrl  
## 3 5.18 ctrl  
## 4 6.11 ctrl  
## 5 4.50 ctrl  
## 6 4.61 ctrl  
## 7 5.17 ctrl  
## 8 4.53 ctrl  
## 9 5.33 ctrl  
## 10 5.14 ctrl  
## 11 4.81 trt1  
## 12 4.17 trt1  
## 13 4.41 trt1  
## 14 3.59 trt1  
## 15 5.87 trt1  
## 16 3.83 trt1  
## 17 6.03 trt1  
## 18 4.89 trt1  
## 19 4.32 trt1  
## 20 4.69 trt1  
## 21 6.31 trt2  
## 22 5.12 trt2  
## 23 5.54 trt2  
## 24 5.50 trt2  
## 25 5.37 trt2  
## 26 5.29 trt2  
## 27 4.92 trt2  
## 28 6.15 trt2  
## 29 5.80 trt2  
## 30 5.26 trt2

### Mostrar una muestra aleatoria

set.seed (420)  
dplyr::sample\_n(my\_data, 10)

## weight group  
## 1 4.50 ctrl  
## 2 5.26 trt2  
## 3 5.14 ctrl  
## 4 3.59 trt1  
## 5 5.50 trt2  
## 6 5.33 ctrl  
## 7 5.80 trt2  
## 8 6.31 trt2  
## 9 5.37 trt2  
## 10 5.54 trt2

En la terminología R, la columna “grupo” se llama factor y las diferentes categorías (“ctr”, “trt1”, “trt2”) se denominan niveles de factor. Los niveles están ordenados alfabéticamente.

### Mostrar los niveles

levels(my\_data$group)

## [1] "ctrl" "trt1" "trt2"

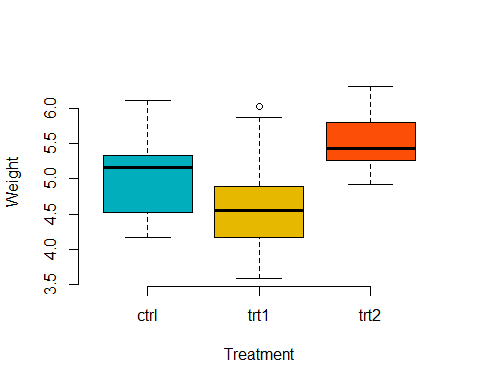
Si los niveles no están automáticamente en el orden correcto, vuelva a ordenarlos de la siguiente manera:

my\_data$group <- ordered(my\_data$group, levels = c("ctrl", "trt1", "trt2"))  
  
group\_by(my\_data, group) %>% summarise( count = n(), mean = mean(weight, na.rm = TRUE), sd = sd(weight, na.rm = TRUE) )

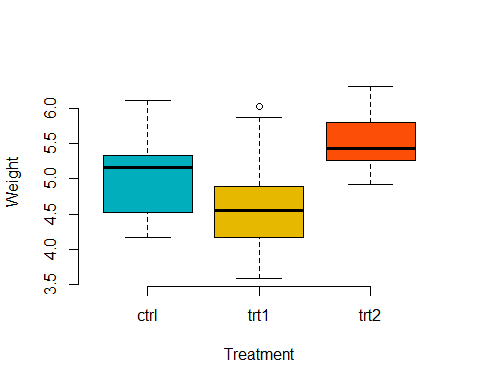
## # A tibble: 3 x 4  
## group count mean sd  
## <ord> <int> <dbl> <dbl>  
## 1 ctrl 10 5.03 0.583  
## 2 trt1 10 4.66 0.794  
## 3 trt2 10 5.53 0.443

### Gráfico Box plot

boxplot(weight ~ group, data = my\_data, xlab = "Treatment", ylab = "Weight", frame = FALSE, col = c("#00AFBB", "#E7B800", "#FC4E07"))



boxplot(weight ~ group, data = my\_data, xlab = "Treatment", ylab = "Weight", frame = FALSE, col = c("#00AFBB", "#E7B800", "#FC4E07"))



# plotmeans   
library("gplots")

##   
## Attaching package: 'gplots'

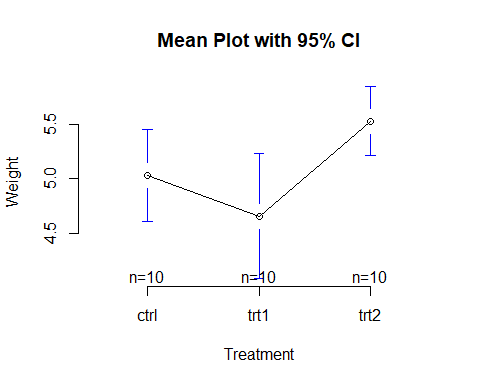
## The following object is masked from 'package:stats':  
##   
## lowess

plotmeans(weight ~ group, data = my\_data, frame = FALSE, xlab = "Treatment", ylab = "Weight", main="Mean Plot with 95% CI")

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "frame" is not a  
## graphical parameter

## Warning in axis(1, at = 1:length(means), labels = legends, ...): "frame" is not  
## a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "frame" is not a  
## graphical parameter



# Compute the analysis of variance  
res.aov <- aov(weight ~ group, data = my\_data)  
# Summary of the analysis  
summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## group 2 3.766 1.8832 4.846 0.0159 \*  
## Residuals 27 10.492 0.3886   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Las diferentes columnas en TukeyHSD representan el siguiente:

diff: diferencia entre medias de los dos grupos

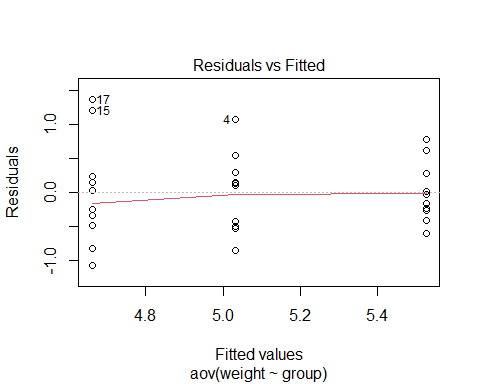
lwr, upr: el punto final inferior y superior del intervalo de confianza al 95% (predeterminado)

p adj: valor p después del ajuste para las comparaciones múltiples.

TukeyHSD (res.aov)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = weight ~ group, data = my\_data)  
##   
## $group  
## diff lwr upr p adj  
## trt1-ctrl -0.371 -1.0622161 0.3202161 0.3908711  
## trt2-ctrl 0.494 -0.1972161 1.1852161 0.1979960  
## trt2-trt1 0.865 0.1737839 1.5562161 0.0120064

plot(res.aov, 1)



# install.packages("tidyverse") # ya todos lo deberían tener  
library(tidyverse)  
# Almacenar los datos en la variable my\_data  
my\_data <- ToothGrowth  
# Mostrar una muestra aleatoria  
set.seed(667)  
dplyr::sample\_n (my\_data, 10)

## len supp dose  
## 1 20.0 OJ 1.0  
## 2 9.4 OJ 0.5  
## 3 14.5 VC 1.0  
## 4 27.3 OJ 2.0  
## 5 30.9 OJ 2.0  
## 6 5.2 VC 0.5  
## 7 21.2 OJ 1.0  
## 8 7.0 VC 0.5  
## 9 11.5 VC 0.5  
## 10 24.8 OJ 2.0

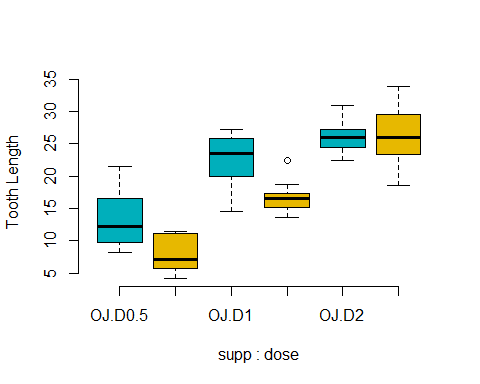
# Verificar la estructura  
str(my\_data)

## 'data.frame': 60 obs. of 3 variables:  
## $ len : num 4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...  
## $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...  
## $ dose: num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...

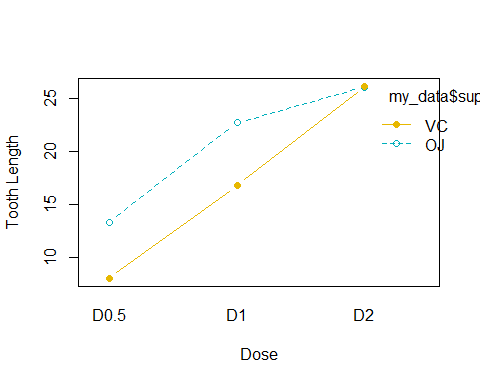
# Convertir dosis como factor y recodificar los niveles  
# como "D0.5", "D1", "D2"  
my\_data$dose <- factor (my\_data$dose,  
 levels = c(0.5, 1, 2),  
 labels = c("D0.5", "D1", "D2"))  
  
# Tablas de Frecuencias  
table(my\_data$supp, my\_data$dose)

##   
## D0.5 D1 D2  
## OJ 10 10 10  
## VC 10 10 10

# Box plot with two factor variables  
boxplot(len ~ supp \* dose, data=my\_data, frame = FALSE,   
 col = c("#00AFBB", "#E7B800"), ylab="Tooth Length")



# Two-way interaction plot  
interaction.plot(x.factor = my\_data$dose, trace.factor = my\_data$supp,   
 response = my\_data$len, fun = mean,   
 type = "b", legend = TRUE,   
 xlab = "Dose", ylab="Tooth Length",  
 pch=c(1,19), col = c("#00AFBB", "#E7B800"))



La función R aov () se puede usar para responder esta pregunta. La función summary.aov () se utiliza para resumir el modelo de análisis de varianza.

res.aov2 <- aov(len ~ supp + dose, data = my\_data)  
summary(res.aov2)

## Df Sum Sq Mean Sq F value Pr(>F)   
## supp 1 205.4 205.4 14.02 0.000429 \*\*\*  
## dose 2 2426.4 1213.2 82.81 < 2e-16 \*\*\*  
## Residuals 56 820.4 14.7   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

La salida incluye las columnas F value y Pr (> F) correspondientes al valor p de la prueba.

# ANOVA de 2 vías con efecto de interacción  
# Estas dos llamadas son equivalentes  
  
# metodo 1  
res.aov3 <- aov(len ~ supp \* dose, data = my\_data)  
# metodo 2  
res.aov3 <- aov(len ~ supp + dose + supp:dose, data = my\_data)  
# sumamry de resultados de ANOVA  
summary(res.aov3)

## Df Sum Sq Mean Sq F value Pr(>F)   
## supp 1 205.4 205.4 15.572 0.000231 \*\*\*  
## dose 2 2426.4 1213.2 92.000 < 2e-16 \*\*\*  
## supp:dose 2 108.3 54.2 4.107 0.021860 \*   
## Residuals 54 712.1 13.2   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Calcular algunas estadísticas resumidas

Calcule la media y la desviación estándar por grupos usando dplyr R dentro de tidyverse:

group\_by(my\_data, supp, dose) %>%  
 summarise(  
 count = n(),  
 mean = mean(len, na.rm = TRUE),  
 sd = sd(len, na.rm = TRUE)  
 )

## `summarise()` has grouped output by 'supp'. You can override using the `.groups` argument.

## # A tibble: 6 x 5  
## # Groups: supp [2]  
## supp dose count mean sd  
## <fct> <fct> <int> <dbl> <dbl>  
## 1 OJ D0.5 10 13.2 4.46  
## 2 OJ D1 10 22.7 3.91  
## 3 OJ D2 10 26.1 2.66  
## 4 VC D0.5 10 7.98 2.75  
## 5 VC D1 10 16.8 2.52  
## 6 VC D2 10 26.1 4.80

TukeyHSD(res.aov3, which = "dose")

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = len ~ supp + dose + supp:dose, data = my\_data)  
##   
## $dose  
## diff lwr upr p adj  
## D1-D0.5 9.130 6.362488 11.897512 0.0e+00  
## D2-D0.5 15.495 12.727488 18.262512 0.0e+00  
## D2-D1 6.365 3.597488 9.132512 2.7e-06

# Homogeneidad de variaciones  
plot(res.aov3, 1)

