Ecología de comunidades en R- clase 2

Ph D.Stephanie Hereira-Pacheco
CTBC
UATx

14 - 03 - 2022

Estadística descriptiva en datos ecológicos

Inicialmente veremos algunas estadísticos básicos de R que son aplicables en la mayoría de nuestros datos incluyendo los datos ecológicos.

Trabajaremos con el dataset conocido como **iris**, que es un dataset ejemplo muy útil para explorar todas estas funciones básicas.

Probando normalidad

Como ya vimos en el módulo anterior hay diversas estrategias para explorar la normalidad de nuestros datos o mediciones tales como los qqplots, la prueba de shapiro y los histogramas de frecuencia. Usando el dataset **iris**:

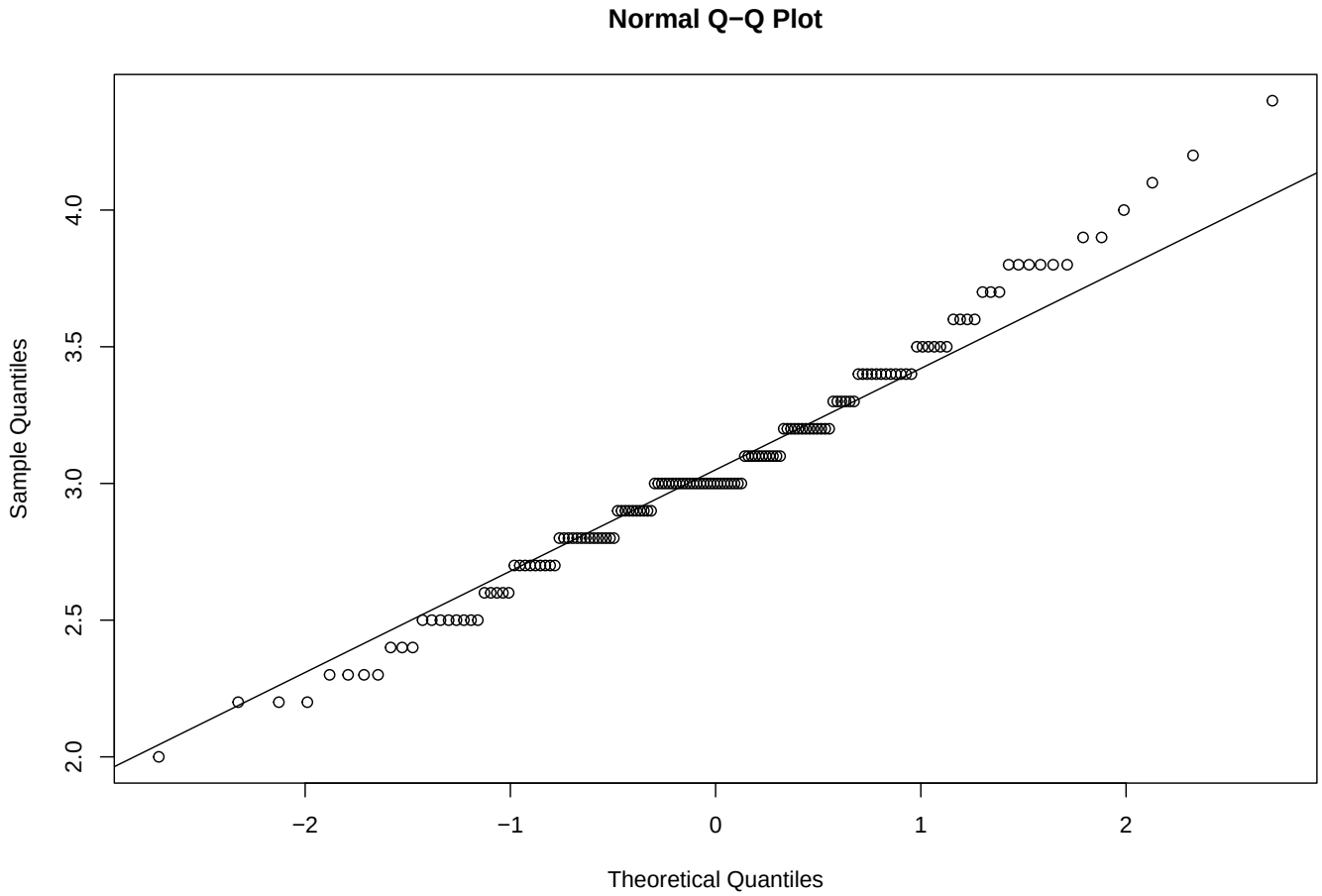
```
data("iris")
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1
```

Probando normalidad

Ahora si, exploremos la normalidad de la variable "Ancho del Sépalo":

```
qqnorm(iris$Sepal.Width)
qqline(iris$Sepal.Width)
```



Probando normalidad

Y numéricamente:

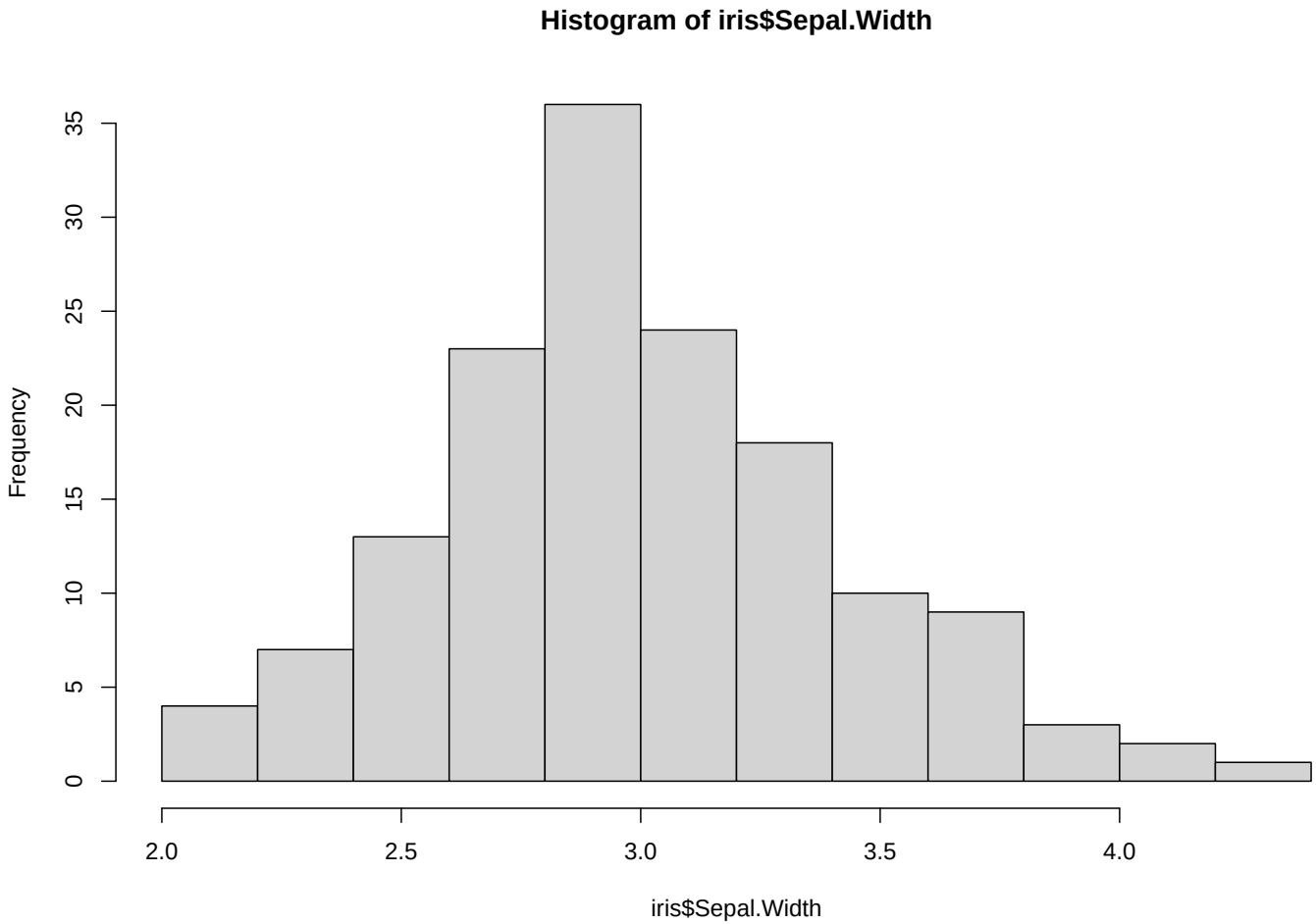
```
shapiro.test(iris$Sepal.Width)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  iris$Sepal.Width
## W = 0.98492, p-value = 0.1012
```

Probando normalidad

Otra gráfica que nos permite ver cómo es la distribución de nuestros datos es un histograma de frecuencias:

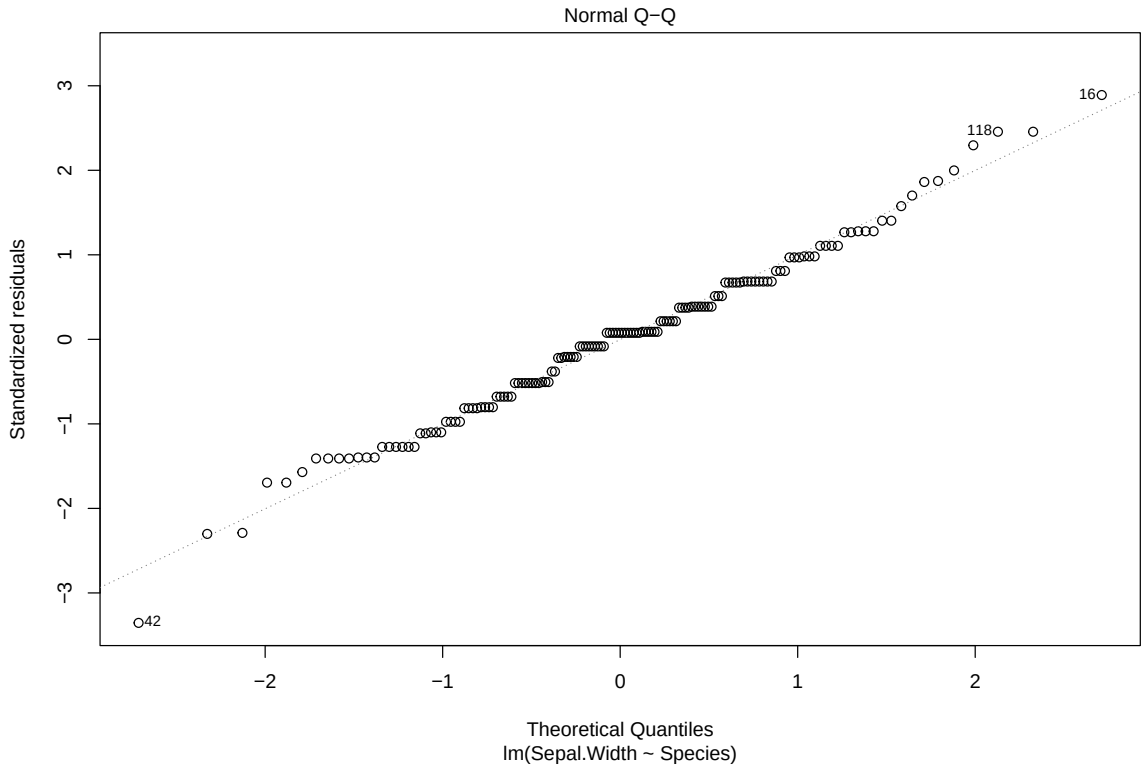
```
hist(iris$Sepal.Width)
```



Probando normalidad

Ahora exploremos la normalidad en un modelo lineal simple declarado:

```
modelo <- lm(Sepal.Width ~ Species, data = iris)
plot(modelo, which = 2)
```



Son solo pocos puntos que se salen de la gráfica, así que asumimos normalidad de nuestro modelo.

Probando homocedasticidad

Para probar la homocedasticidad o la homogeneidad de varianzas, podemos aplicar la prueba de Barlett:

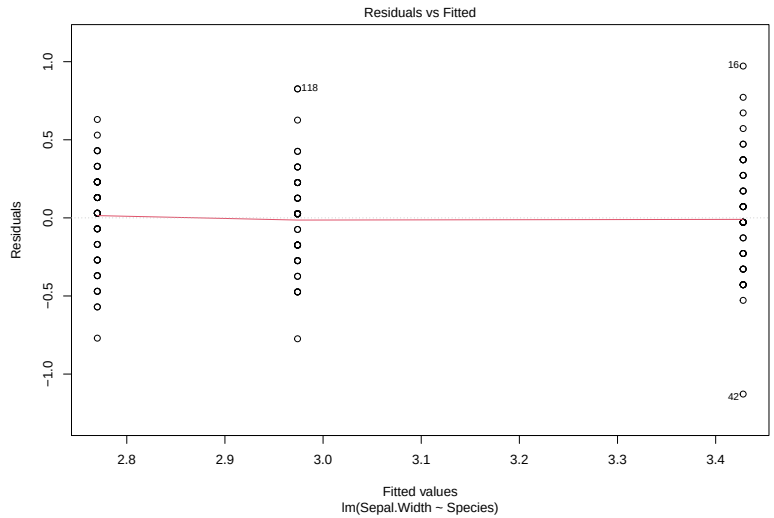
```
bartlett.test(iris$Sepal.Width, iris$Species)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  iris$Sepal.Width and iris$Species
## Bartlett's K-squared = 2.0911, df = 2, p-value = 0.3515
```

Probando homocedasticidad

No hay diferencias significativas entre las varianzas de los grupos. Esto lo podemos ver gráficamente también:

```
plot(modelo, which = 1)
```




```
aggregate(Sepal.Width ~Species, data = iris, FUN = var)
```

```
##      Species Sepal.Width
## 1     setosa  0.14368980
## 2 versicolor  0.09846939
## 3  virginica  0.10400408
```

Prueba de Chi-cuadrado y de Fisher (tablas de contingencia)

Primero categorizaremos la variable *Largo del Sépalo* de **iris**, haciendo 'pequeño' las valores que estén por debajo de la mediana y 'grande' los que estén por encima de la mediana.

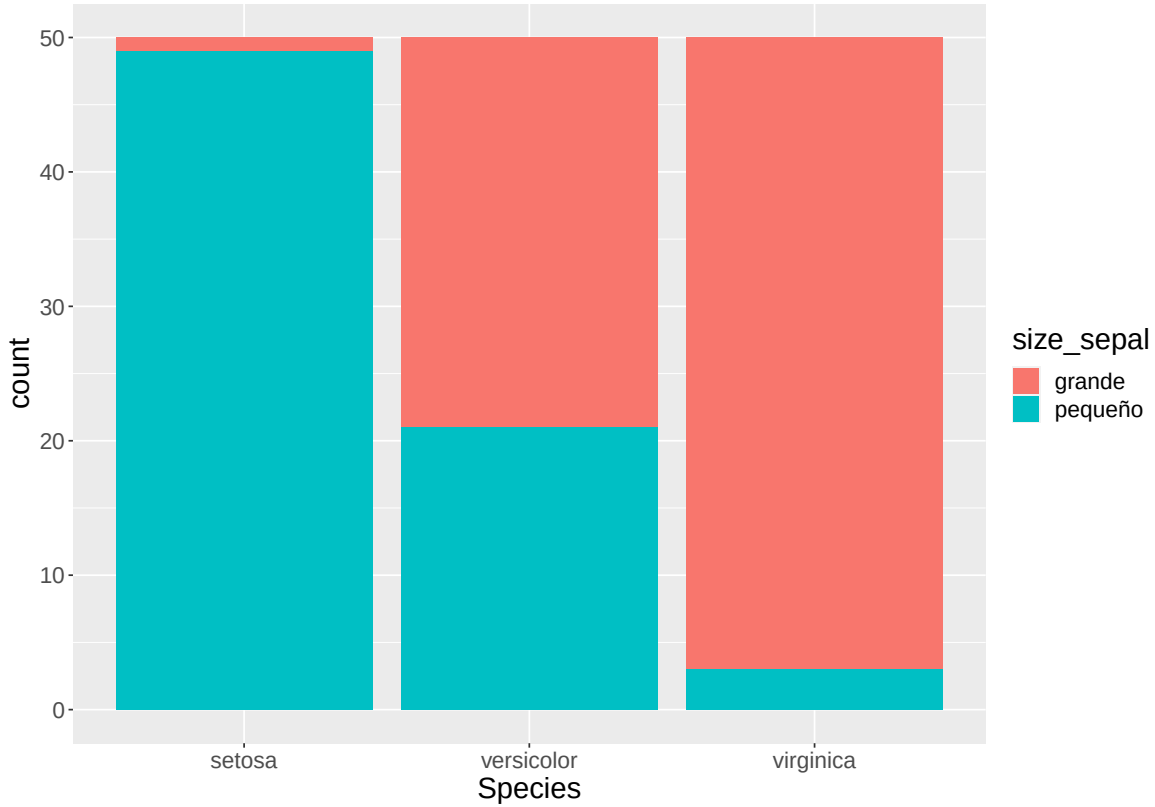
```
iris$size_sepal <- ifelse(iris$Sepal.Length < median(iris$Sepal.Length),
                          "pequeño", "grande")
```

Prueba de Chi-cuadrado y de Fisher (tablas de contingencia)

Gráficamente se ve así:

```
library(ggplot2)

ggplot(iris) +
  aes(x = Species, fill = size_sepal) +
  geom_bar()+ theme(text = element_text(size = 18))
```



Prueba de Chi-cuadrado y de Fisher (tablas de contingencia)

Luego construiremos una tabla de contingencia a partir de esto:

```
tabla_contingencia<-table(iris$Species, iris$size_sepal)
tabla_contingencia
```

```
##
##                grande pequeño
##    setosa           1      49
##    versicolor      29      21
##    virginica       47       3
```

Prueba de Chi-cuadrado y de Fisher (tablas de contingencia)

Y aplicaremos las pruebas:

```
chisq.test(tabla_contingencia)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tabla_contingencia
## X-squared = 86.035, df = 2, p-value < 2.2e-16
```

Prueba de Chi-cuadrado y de Fisher (tablas de contingencia)

```
fisher.test(tabla_contingencia)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tabla_contingencia
## p-value < 2.2e-16
## alternative hypothesis: two.sided
```

Así que rechazamos la hipótesis nula para la prueba de independencia Chi-cuadrado y de Fisher, esto significa que existe una relación significativa entre la especie y el tamaño del sépalo.

Análisis 1 variable cuantitativa y 2 grupos independientes

```
library(dplyr)
iris_dos<- iris %>% filter(!Species == "versicolor")
unique(iris_dos$Species)
```

```
## [1] setosa    virginica
## Levels: setosa versicolor virginica
```

Análisis 1 variable cuantitativa y 2 grupos independientes

Paramétrica

```
t.test(iris_dos$Sepal.Width ~ iris_dos$Species)
```

```
##
##  Welch Two Sample t-test
##
## data:  iris_dos$Sepal.Width by iris_dos$Species
## t = 6.4503, df = 95.547, p-value = 4.571e-09
## alternative hypothesis: true difference in means between group setosa an
## 95 percent confidence interval:
##  0.3142808 0.5937192
## sample estimates:
##    mean in group setosa mean in group virginica
##                   3.428                   2.974
```

Análisis 1 variable cuantitativa y 2 grupos independientes

No paramétrica:

```
wilcox.test(iris_dos$Sepal.Length ~ iris_dos$Species)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  iris_dos$Sepal.Length by iris_dos$Species
## W = 38.5, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Análisis 1 variable cuantitativa y 2 grupos dependientes

Tengamos una data de ejemplo de unos ratones a los que se pesaron al comienzo del experimento (aplicación del tratamiento) y al final, y deseamos saber si hay diferencias significativas en su peso:

```
#individuos
ratones<- paste0("raton_", 1:10)
#Peso antes
pa<-c(200.1, 190.9, 192.7, 213, 241.4, 196.9, 172.2, 185.5, 205.2, 193.7)
#peso después
pd<-c(392.9, 393.2, 345.1, 393, 434, 427.9, 422, 383.9, 392.3, 352.2)

data_ratones<- data.frame(ratones=ratones, peso_antes=pa, peso_despues=pd)
```

Análisis 1 variable cuantitativa y 2 grupos independientes

Versión paramétrica:

```
t.test(data_ratones$peso_antes, data_ratones$peso_despues, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  data_ratones$peso_antes and data_ratones$peso_despues
## t = -20.883, df = 9, p-value = 6.2e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -215.5581 -173.4219
## sample estimates:
## mean of the differences
##                  -194.49
```

Análisis 1 variable cuantitativa y 2 grupos independientes

Versión no paramétrica:

```
wilcox.test(data_ratones$peso_antes, data_ratones$peso_despues, paired = TR
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  data_ratones$peso_antes and data_ratones$peso_despues
## V = 0, p-value = 0.001953
## alternative hypothesis: true location shift is not equal to 0
```

Análisis 1 variable cuantitativa y 2 o más grupos independientes:

Paramétrico:

```
modelo<- lm(data = iris, Sepal.Width ~ Species)
anova_modelo<- aov(modelo)
summary(anova_modelo)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Species       2  11.35   5.672   49.16 <2e-16 ***
## Residuals   147  16.96   0.115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Análisis 1 variable cuantitativa y 2 o más grupos independientes:

No paramétrico:

```
kruskal.test(iris$Petal.Length, iris$Species)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  iris$Petal.Length and iris$Species
## Kruskal-Wallis chi-squared = 130.41, df = 2, p-value < 2.2e-16
```

Análisis 1 varible cuantitva y 2 o más grupos dependientes

Un estudio pretende determinar si existe diferencia en como de bueno consideran los consumidores que es un vino dependiendo de la hora del día en la que lo toman. Para ello se selecciona a un grupo de 7 sujetos a los que se les da a probar un vino por la mañana, por la tarde y por la noche. En cada degustación se valora del 1 al 11 el vino (los degustadores no saben que es el mismo vino).

Análisis 1 varible cuantitva y 2 o más grupos dependientes

```
valoracion <- c( 9, 5, 2, 6, 3, 1, 5, 5, 5, 11, 5,
                 1, 8, 4, 3, 10, 4, 1, 7, 3, 4 )
hora <- factor( rep( c( "mañana", "tarde", "noche" ), 7 ) )
sujeto <- factor( rep( 1:7, each = 3 ) )
datos <- data.frame( valoracion, hora, sujeto )
head(datos)
```

```
##   valoracion   hora sujeto
## 1          9 mañana      1
## 2          5  tarde      1
## 3          2  noche      1
## 4          6 mañana      2
## 5          3  tarde      2
## 6          1  noche      2
```

Análisis 1 varible cuantitva y 2 o más grupos dependientes

Versión no paramétrica:

```
friedman.test(valoracion, hora, sujeto)
```

```
##
##  Friedman rank sum test
##
## data:  valoracion, hora and sujeto
## Friedman chi-squared = 10.333, df = 2, p-value = 0.005704
```

```
#valor medido, grupos, bloques
```

Análisis 1 varible cuantitva y 2 o más grupos dependientes

Versión paramétrica (anova por bloques):

```
anova_bloques<- aov(lm(valoracion ~ hora+sujeto))
summary(anova_bloques)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## hora         2 114.00   57.00  17.690 0.000264 ***
## sujeto       6   9.90    1.65   0.512 0.788220
## Residuals   12  38.67    3.22
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Análisis de medidas repetidas

Queremos realizar un análisis de medidas repetidas para los datos de rendimiento académico de seis alumnos. En cada alumno se ha medido a cinco tiempos diferentes su rendimiento, por tanto las muestras tomadas no son independientes entre si. Para poder analizar estos datos debemos considerar las muestras como relacionadas, es decir debemos realizar un ANOVA de medidas repetidas.

```
individuos<- factor(c(rep(1,5), rep(2,5), rep(3, 5),
                      rep(4, 5), rep(5,5), rep(6,5)))
tiempo<- factor(rep(1:5, 6))
rendimiento<- c(8.5, 8.2,8.9, 7.7, 7.4,
                9.8,8.9,8.9,8.8,8.1,
                9.6,9.0, 9.3, 7.5, 7.1,
                7.5, 7.8, 7.8, 4.5, 4.6,
                5.8, 5.8, 5.9, 2.6, 1.2,
                9.9, 9.8, 9.6, 8.6, 8.7)
data_rendimiento<- data.frame(individuos=individuos,
                              tiempo=tiempo, rendimiento=rendimiento)
```

Análisis de medidas repetidas

```
str(data_rendimiento)
```

```
## 'data.frame':    30 obs. of  3 variables:
##  $ individuos : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 2 2 2 2
##  $ tiempo     : Factor w/ 5 levels "1","2","3","4",..: 1 2 3 4 5 1 2 3 4
##  $ rendimiento: num  8.5 8.2 8.9 7.7 7.4 9.8 8.9 8.9 8.8 8.1 ...
```

Análisis de medidas repetidas

Vamos a realizar el análisis de medidas repetidas ANOVA paramétrico por medio del paquete ez. Para ello hay que indicar nuestros datos (data), nuestra variable (dv), nuestros individuos (wid) y el tiempo (within):

Análisis de medidas repetidas

Versión paramétrica:

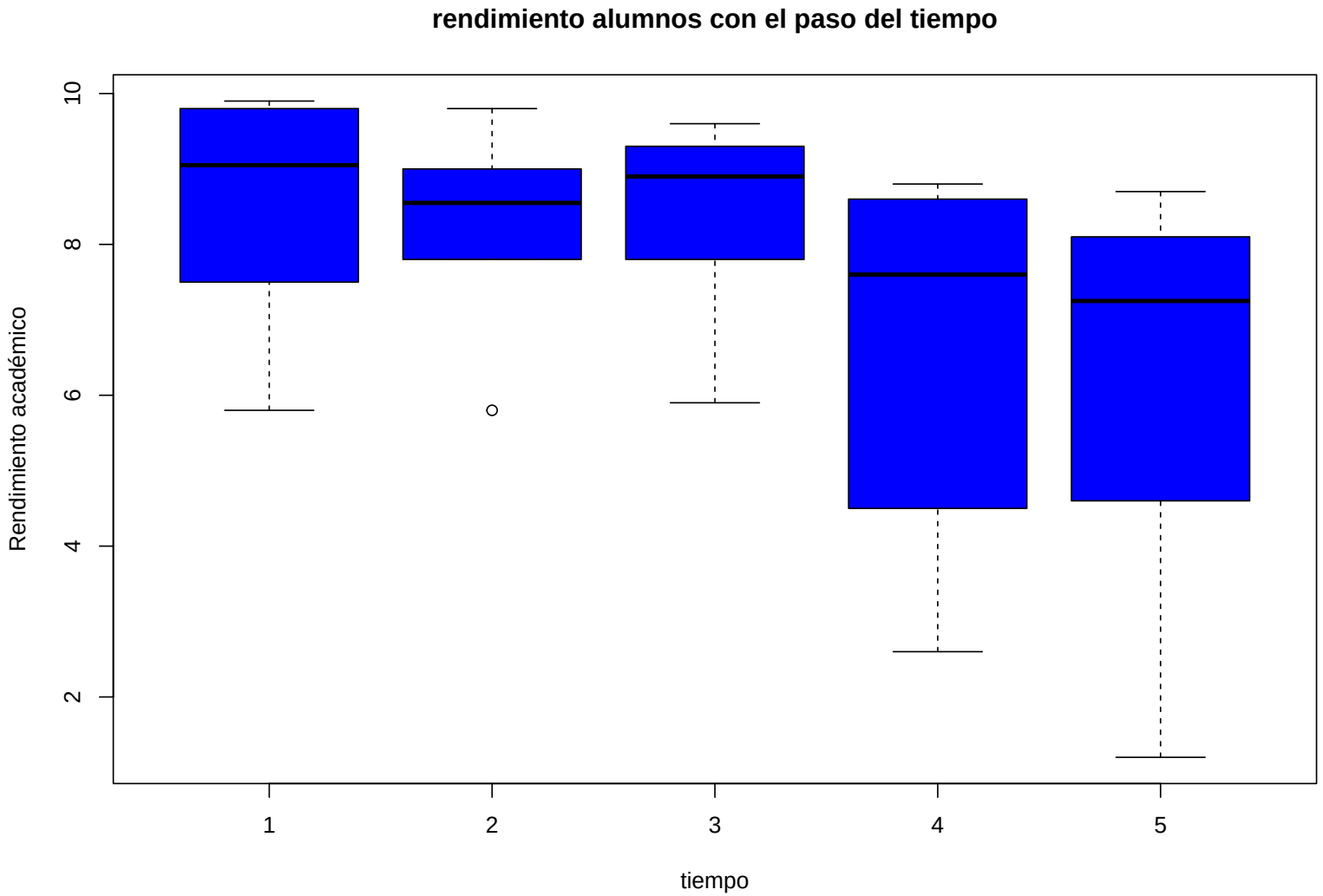
```
library(ez)
ezANOVA(data=data_rendimiento, dv=rendimiento, wid=individuos, within=tiemp
```

```
## $ANOVA
##   Effect DFn DFd        F            p p<.05       ges
## 2 tiempo   4  20 12.41317 3.096979e-05     * 0.2211562
##
## $`Mauchly's Test for Sphericity`
##   Effect           W          p p<.05
## 2 tiempo 0.003212948 0.03413309     *
##
## $`Sphericity Corrections`
##   Effect       GGe       p[GG] p[GG]<.05       HFe       p[HF] p[HF]<.05
## 2 tiempo 0.3126606 0.009730034         * 0.3669864 0.006076078         *
```

Podemos ver en nuestros resultados que el tiempo es significativo, es decir, el rendimiento escolar cambia con el tiempo. Pero en este caso nuestros datos violan uno de los requisitos que es la esfericidad (test de Mauchly significativo), debemos fiarnos de la p de la **Sphericity corrections** que nos confirma lo que hemos deducido al principio.

Análisis de medidas repetidas

```
boxplot(rendimiento~tiempo, xlab="tiempo",
        ylab="Rendimiento académico",
        main="rendimiento alumnos con el paso del tiempo",
        col="blue", data=data_rendimiento)
```



Análisis de medidas repetidas

Versión no paramétrica:

```
library(tidyverse)
library(jmv)
#cambiamos el formato de la data
data_rendimiento_notidy<-data_rendimiento %>% mutate(tiempo =case_when(
  tiempo== 1 ~ "T1",
  tiempo== 2 ~ "T2",
  tiempo== 3 ~ "T3",
  tiempo== 4 ~ "T4",
  tiempo== 5 ~ "T5")) %>% pivot_wider(names_from =tiempo,
                                      values_from = rendimiento )

jmv::anovaRMNP(data_rendimiento_notidy, measures=vars(T1,T2,T3,T4,T5))
```

Correlación lineal simple

```
cor(iris$Petal.Length, iris$Petal.Width)
```

```
## [1] 0.9628654
```

```
cor(iris$Petal.Length, iris$Petal.Width, method = "spearman")
```

```
## [1] 0.9376668
```

Regresión lineal simple lm y glm

```
modelo_lm <- lm(Petal.Width ~ Petal.Length, data = iris)
summary(modelo)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Species, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.128 -0.228  0.026  0.226  0.972
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.42800    0.04804  71.359  < 2e-16 ***
## Speciesversicolor -0.65800    0.06794  -9.685  < 2e-16 ***
## Speciesvirginica  -0.45400    0.06794  -6.683 4.54e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3397 on 147 degrees of freedom
## Multiple R-squared:  0.4008, Adjusted R-squared:  0.3926
## F-statistic: 49.16 on 2 and 147 DF,  p-value: < 2.2e-16
```

Regresión lineal simple lm y glm

Con summary() podemos ver los coeficientes de la ecuación, en este caso son: para el intercepto -0.36 y para la pendiente es 0.41. De nuevo los valores p están por debajo de 0.05. Los coeficientes son la pendiente y el intercepto. Así que la ecuación queda -> Ancho_Petalo = Largo_Petalo*0.4157 - 0.3630

Otro resultado importante es el R cuadrado que nos dice la bondad del ajuste del modelo, esto es la fracción de mis datos que es explicado por el modelo en este caso si miramos el valor ajustado, el modelo explica el 92% de mis datos.

glm() se utiliza con otras distribuciones que no sean la distribución normal. Porque **lm()** asume la distribución normal de los datos.

Regresión lineal simple lm y glm

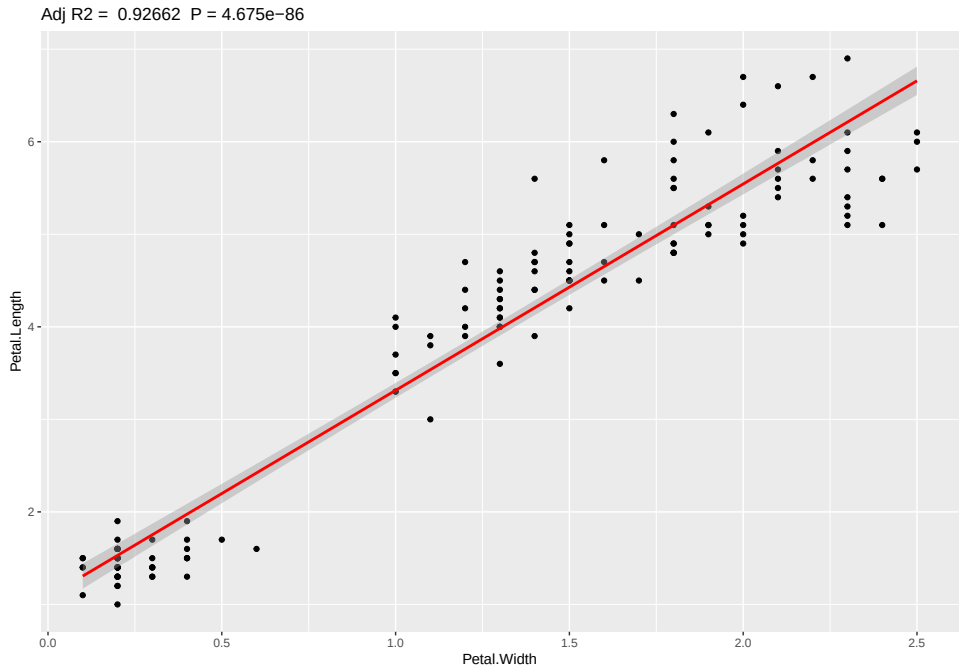
```
modelo_glm <- glm(Petal.Width ~ Petal.Length, data = iris)
summary(modelo)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Species, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.128 -0.228  0.026  0.226  0.972
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.42800    0.04804  71.359  < 2e-16 ***
## Speciesversicolor -0.65800    0.06794  -9.685  < 2e-16 ***
## Speciesvirginica  -0.45400    0.06794  -6.683 4.54e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3397 on 147 degrees of freedom
## Multiple R-squared:  0.4008, Adjusted R-squared:  0.3926
## F-statistic: 49.16 on 2 and 147 DF,  p-value: < 2.2e-16
```

Regresión lineal simple lm y glm

Si queremos visualizar el modelo lineal simple:

```
library(ggplot2)
ggplot(iris, aes(x = Petal.Width, y = Petal.Length)) +
  geom_point() +
  stat_smooth(method = "lm", col = "red") +
  labs(title = paste("Adj R2 = ",signif(
    summary(modelo_lm)$adj.r.squared, 5),
    " P =",signif(summary(modelo_lm)$coef[2,4], 5)))
```



Modelos lineales mixtos

```
library(readr)
rabbit<- read_tsv("https://raw.githubusercontent.com/Steph0522/Curso_R_basi
head(rabbit)
```

```
## # A tibble: 6 x 3
##   treat  gain block
##   <chr> <dbl> <chr>
## 1 f      42.2 b1
## 2 b      32.6 b1
## 3 c      35.2 b1
## 4 c      40.9 b2
## 5 a      40.1 b2
## 6 b      38.1 b2
```

Modelos lineales mixtos

Paramétrica:

```
library(lme4)
library(nlme)
lme.rabbit1 <- lmer(gain~ treat +(1|block), data=rabbit)
lme.rabbit2 <- lme(gain~ treat, random = ~1|block, data=rabbit)

anova(lme.rabbit1)
```

```
## Analysis of Variance Table
##       npar Sum Sq Mean Sq F value
## treat    5 165.47  33.093  3.2818
```

```
anova(lme.rabbit2)
```

```
##             numDF denDF  F-value p-value
## (Intercept)     1    15 590.5297  <.0001
## treat           5    15   3.2818  0.0336
```

Modelos lineales mixtos

No paramétrica:

```
lme.rabbit1 <- glmer(gain~ treat +(1|block), data=rabbit, family = "poisson
```

No correré este porque toma tiempo en correr pero sólo para que conozcan la función y puedan aplicarla si es de su interés.

Transformación de datos

Dependiendo de nuestros tipos de datos y de los análisis a realizar algunas veces es necesario filtrar nuestros datos (datos NA's o ceros) o también puede ser requerido transformar los datos. Por ejemplo con los datos de expresión de genes.

En R podemos usar varias funciones para transformar datos:

```
log() # aplicar el logartimo a nuestros datos
scale() # escala o centra tus datos
sqrt() #aplica la raiz cuadrada a nuestros datos
```

Transformación de datos

Ejemplo:

```
data("pressure")
str(pressure)
```

```
## 'data.frame':    19 obs. of  2 variables:
##  $ temperature: num  0 20 40 60 80 100 120 140 160 180 ...
##  $ pressure   : num  0.0002 0.0012 0.006 0.03 0.09 0.27 0.75 1.85 4.2 8.
```

```
cor(pressure$temperature, pressure$pressure)
```

```
## [1] 0.7577923
```

```
model<- lm(pressure ~ temperature, data = pressure)
summary(model)
```

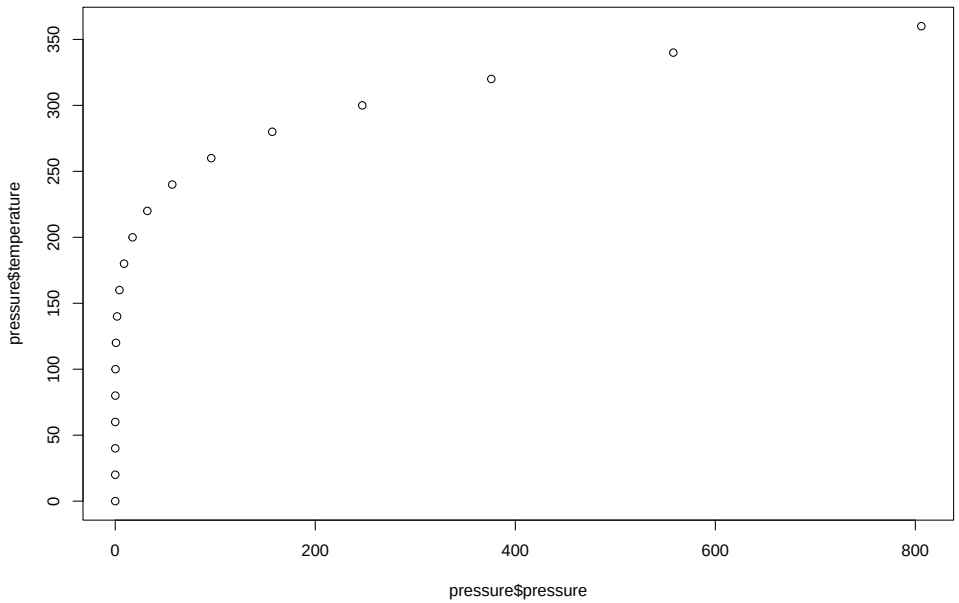
```
##
## Call:
## lm(formula = pressure ~ temperature, data = pressure)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -158.08 -117.06  -32.84   72.30  409.43
##
```

Transformación de datos

```
shapiro.test(pressure$pressure)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  pressure$pressure
## W = 0.63666, p-value = 1.071e-05
```

```
plot(pressure$pressure, pressure$temperature)
```



Transformación de datos

```
library(dplyr)
pressure_log<- pressure %>% mutate(pressure_log=log(pressure))
model_log<-lm(pressure_log~ temperature, data=pressure_log)
```

Transformación de datos

```
summary(model_log)
```

```
##
## Call:
## lm(formula = pressure_log ~ temperature, data = pressure_log)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4491 -0.6876  0.2866  0.8716  1.1365
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.068144   0.483831  -12.54 5.10e-10 ***
## temperature  0.039792   0.002296   17.33 3.07e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.096 on 17 degrees of freedom
## Multiple R-squared:  0.9464, Adjusted R-squared:  0.9433
## F-statistic: 300.3 on 1 and 17 DF,  p-value: 3.07e-12
```

Transformación de datos

Visualizando:

```
ggplot(pressure_log, aes(x = pressure_log, y = temperature)) +
  geom_point() +
  stat_smooth(method = "lm", col = "red")+theme(text = element_text(size =
```