Taller Estadística en R: Sesión 2

Abril de 2022

Estadística Descriptiva y Distribuciones

Selección y modificación de datos

Lo primero que hay que hacer es cargar y seleccionar los datos que nos interesan.

```
lombrices<-read.delim("Bd_Lombrices_Tulenapa_todo.txt")
```

Cuando tenemos datos lo primero que nos interesa es ver qué tipo de datos tenemos, qué es lo más común, qué tan dispersos son, etc, etc y para esto es muy útil R

Variables

Nuestros sets de datos tienen registros para múltiples variables y es importante ser consciente de cuales son esas variables y qué naturaleza tienen. Para ver que variables hay en un set de datos usamos el comando **names()**

```
names(lombrices)
```

```
##  [1] "Punto"    "X"        "Y"        "Abu_Lomb" "HZ_A"     "HS"
##  [7] "DA"       "pH"       "BH"       "BDF"      "MO"       "estacion"
```

Las variables pueden ser numéricas o categóricas. Para saber si una variable de un set de datos es númerica podemos utilizar el comando **mode()**

```
mode(lombrices$estacion)
```

```
## [1] "character"
```

```
#Con el operador $ seleccionamos una variable dentro del set de datos
mode(lombrices$pH)
```

```
## [1] "numeric"
```

Seleccionar los datos

Muchas veces solo nos interesan ciertas variables de un set de datos o ciertos registros, por lo que podemos recortar nuestros sets para trabajar con sets más pequeños y manejables. Esto lo podemos hacer indexando o con el comando **subset()**

Por ejemplo si queremos solo ciertas columnas o ciertas filas podemos guardar un objeto indexando nuestro set de datos

```
lombrices_modificado1<-lombrices[1:5,1:4]
#lombrices modificado 1 va a ser solo las filas de la 1 a la 5 y las columnas de la 1 a la 4

lombrices_modificado2<-lombrices[,c(-2,-3)]
#lombrices modificado 2 va a ser todas las filas pero sin las columnas de la 2 y 3
```

```
lombrices_modificado3<-lombrices[c(1:100),]
#lombrices modificado 3 va a ser los datos de lombrices pero solo las 100 primeras filas
```

Si lo que necesitamos es que se nos seleccionen registros que cumplan una característica específica usamos **subset()**

```
estacion_seca<-subset(lombrices, estacion=="seca")
head(estacion_seca)
```

```
##   Punto       X       Y Abu_Lomb HZ_A   HS  DA  pH    BH  BDF   MO estacion
## 1     1 1045191 1352343        3  0.0 17.7 1.4 6.7  75.3 16.1 14.4     seca
## 2     3 1044695 1351377        0  3.0 29.4 0.8 6.5 139.2 54.2 16.2     seca
## 3     4 1044751 1351363        0  0.1 31.4 1.1 7.0  47.6 21.7 15.0     seca
## 4     5 1044812 1351357        1  2.5 30.6 0.8 6.5  94.0 62.0 18.5     seca
## 5     6 1044875 1351351        0  1.6 23.7 1.1 6.6 109.0 51.2 16.8     seca
## 6     7 1044930 1351345        0  1.7 17.5 1.5 6.7 141.9 75.1 12.2     seca
```

```
#estacion_seca son únicamente los datos para la temporada seca#
ph_basico<-subset(lombrices,pH>6)
#ph básico son los registros en donde el ph es mayor a 6
basico_lluvia<-subset(lombrices,estacion=="Lluvia"& pH>6)
#basico_lluvia filtra los datos en donde la estación sea lluvia Y el pH mayor a 6
```

Añadir datos

Si por ejemplo quiero añadirle una variable más a mi set de datos, puedo usar el operador $ con la siguiente estructura:

set_de_datos$nombre_de_la_variable<-objeto con la variable

```
sitio<-rep("Tulenapa",194)
lombrices$Sitio<-sitio
head(lombrices)
```

```
##   Punto       X       Y Abu_Lomb HZ_A   HS  DA  pH    BH  BDF   MO estacion
## 1     1 1045191 1352343        3  0.0 17.7 1.4 6.7  75.3 16.1 14.4     seca
## 2     3 1044695 1351377        0  3.0 29.4 0.8 6.5 139.2 54.2 16.2     seca
## 3     4 1044751 1351363        0  0.1 31.4 1.1 7.0  47.6 21.7 15.0     seca
## 4     5 1044812 1351357        1  2.5 30.6 0.8 6.5  94.0 62.0 18.5     seca
## 5     6 1044875 1351351        0  1.6 23.7 1.1 6.6 109.0 51.2 16.8     seca
## 6     7 1044930 1351345        0  1.7 17.5 1.5 6.7 141.9 75.1 12.2     seca
##      Sitio
## 1 Tulenapa
## 2 Tulenapa
## 3 Tulenapa
## 4 Tulenapa
## 5 Tulenapa
## 6 Tulenapa
```

Medidas de tendencia central

Los estadísticos de tendencia central más usados son la media aritmética, la mediana y la moda.

Las funciones **mean()** y **median()** nos permiten hallar la media y la mediana

```
mean(lombrices$pH,na.rm = TRUE)
```

```
## [1] 6.869948
```

```r
#Utilizamos el argumento na.rm para quitar los datos faltantes
median(lombrices$DA)
```

```
## [1] 0.8
```

Medidas de dispersión o envergadura

En R también podemos hallar las medida básicas para mirar qué tan dispersos están nuestros datos. Las funciones **sd()**,**var()** y **IQR()** nos permiten hallar la desviación estándar, la varianza y el rango intercuartil respectivamente

```r
sd(lombrices$DA)
```

```
## [1] 0.2627793
```

```r
var(lombrices$BDF,na.rm = TRUE)
```

```
## [1] 1941.78
```

```r
IQR(lombrices$MO)
```

```
## [1] 5.725
```

Si queremos sacar todos estos datos, podemos usar el comando **summary()** que los arroja todos de una vez

```r
summary(lombrices$pH)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    5.70    6.30    7.10    6.87    7.40    8.00       1
```

Gráficos

Representar gráficamente los datos es importante. Para hacer una buena gráfica es necesario:

- Mostrar todos los datos
- Representar las magnitudes con precisión
- Minimizar el desorden
- Que sea fácil de interpretar
- Que los ejes estén correctamente identificados

Comandos básicos para gráficos

Los comandos más utilizados son:

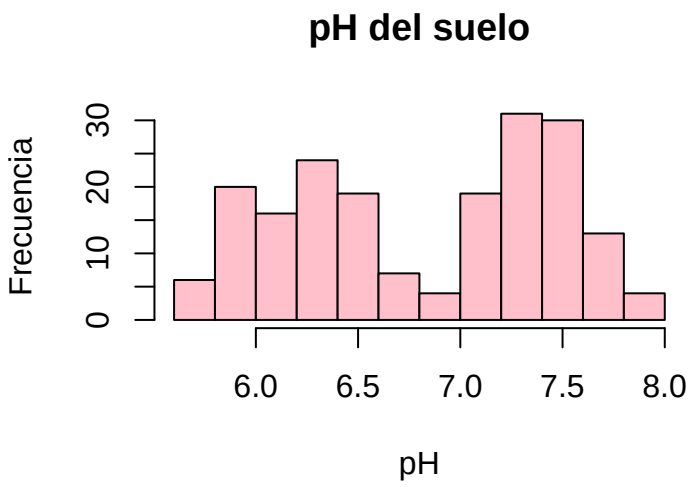
- *plot()*
- *hist()*
- *boxplot()*

y en todos estos comandos hay argumentos básicos para cambiar parámetros gráficos como:

- *main=:* Para poner el título del gráfico
- *ylab=/xlab=:* para poner los nombres de los ejes
- *col=:* para elegir el color de la gráfica
- *xlim/ylim=:* para elegir la escala de los ejes (donde empieza y donde termina)

Histograma Se usa para graficar variables numéricas (en el eje x) contra su frecuencia (eje y)

```r
hist(lombrices$pH, main="pH del suelo", ylab="Frecuencia",xlab = "pH",col="pink")
```



Boxplot Se usa para graficar variables categóricas (en el eje x) contra variables numéricas (en el eje y)

```
boxplot(lombrices$pH~lombrices$estacion,
        main= "Estación vs pH",xlab="Estacion",ylab="pH",
        col=c("pink","lightblue"))
```

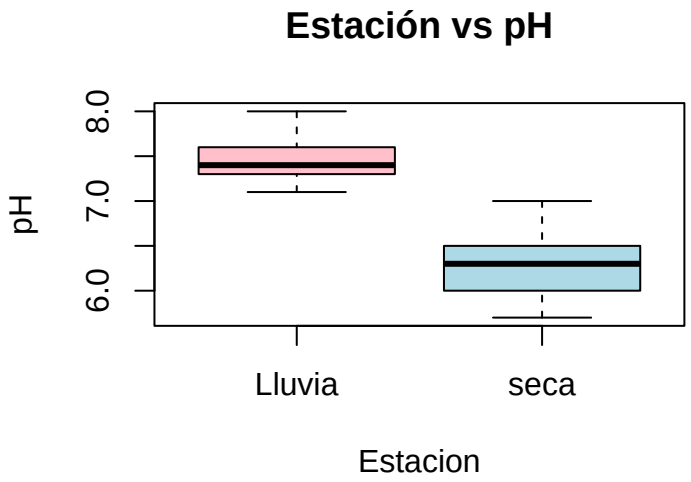


Gráfico de dispersión Se usa para graficar variables numéricas, variable independiente en el eje x, y variable dependiente en el eje y

```
plot(lombrices$MO,lombrices$HS,
     main = "MO vs HS",xlab = "MO",
     ylab = "HS",col="lightgreen")
```

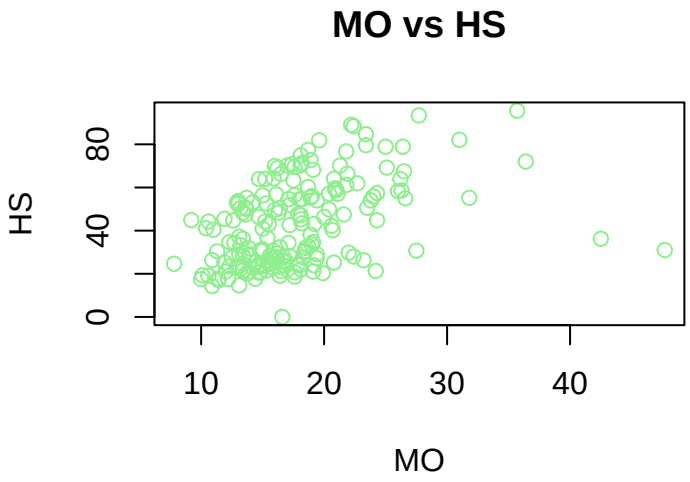
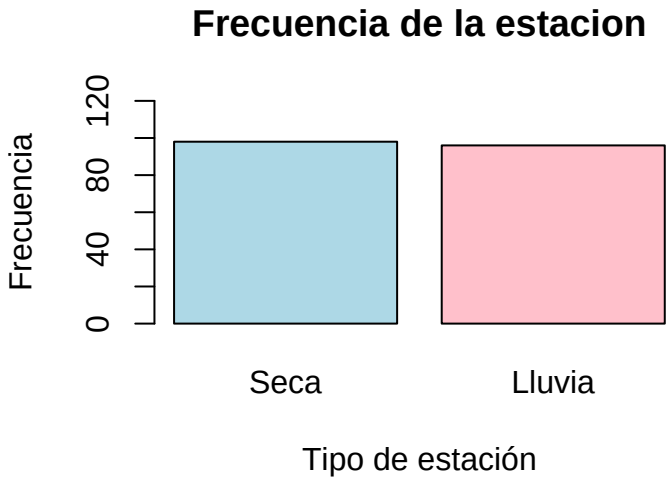


Grafico de barras Sirve para gráficar variables categóricas vs frecuencias.

```
estacion_tabla<-table(lombrices$estacion)
barplot(estacion_tabla,main="Frecuencia de la estacion", ylim = c(0,120),
  names.arg = c("Seca","Lluvia"),
  col=c("lightblue","pink"),
  ylab = "Frecuencia",xlab = "Tipo de estación")
```



Distribuciones de datos

Los datos se distribuyen en diferentes valores. Hablamos de *distribución de frecuencias* si trabajamos con la muestra y *distribución de probabilidades* si trabajamos la población. Se puede usar las frecuencias absolutas, en donde se utiliza el número de observaciones o se puede usar la frecuencia relativa, que es la proporción de observaciones teniendo en cuenta el total de observaciones.

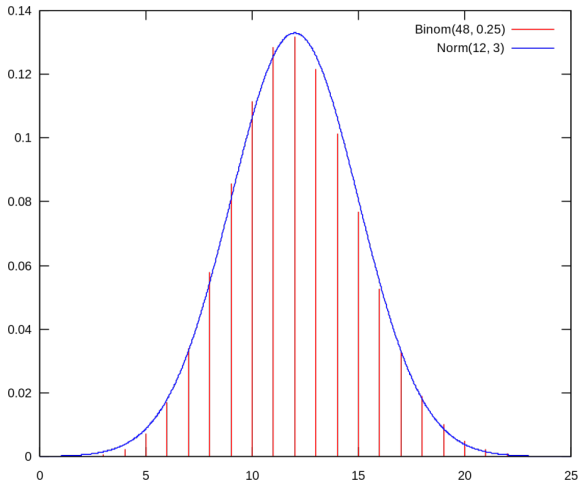
Hay dos distribuciones muy importantes: la distribución normal y la distribución binomial

La distribución normal

La distribución normal (también denominada gaussiana) es una distribución de probabilidad simétrica con una característica forma de campana. La distribución normal es la distribución de probabilidad más importante para el análisis de datos; Los procedimientos estadísticos más utilizados en biología (por ejemplo, regresión lineal, análisis de varianza) asumen que las variables que se analizan (o las desviaciones de un modelo ajustado) siguen una distribución normal.

La distribución binomial

Es una distribución discreta de probabilidades para el número de éxitos en un número fijo de pruebas independientes en donde la probabilidad de éxito siempre es la misma



Comprobar normalidad y homocedasticidad

Para muchas de las pruebas, se supone que las variables tienen una distribución normal y que las varianzas de los grupos son homogéneas, por lo que es necesario comprobar la normalidad y la homocedasticidad.

Para eso podemos usar diferentes pruebas. Para normalidad la más común es la prueba de Shapiro-Wilks y para la homocedasticidad la prueba F o la prueba de Levene si tengo más de 2 grupos.

Prueba de Shapiro-Wilks

se usa para contrastar la normalidad de un conjunto de datos. Se plantea como hipótesis nula que una muestra proviene de una población normalmente distribuida.

Para hacerla en R se usa el comando **shapiro.test()**

Si p>0.05 fallo en rechazar la hipótesis nula y concluyo que los datos son normales

Si p<0.05 rechazo la hipótesis nula y concluyo que los datos no son normales

```
seca<-subset(lombrices,estacion=="seca")
shapiro.test(seca$pH)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  seca$pH
## W = 0.97139, p-value = 0.03384
```

```r
#los datos de pH para la estación seca NO son normales
lluvia<-subset(lombrices,estacion=="Lluvia")
shapiro.test(lluvia$pH)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lluvia$pH
## W = 0.95983, p-value = 0.004678
```

```r
#los datos de pH para la estación lluvia NO son normales
```

Prueba F

Sirve para mirar si hay homogeneidad de varianzas entre 2 grupos. La hipótesis nula es que no hay diferencias significativas entre las varianzas de los dos grupos

En R se puede hacer con el comando **var.test()**

```r
var.test(seca$pH,lluvia$pH) #las varianzas del pH no son parecidas
```

```
##
##  F test to compare two variances
##
## data:  seca$pH and lluvia$pH
## F = 1.9186, num df = 95, denom df = 96, p-value = 0.001632
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.281799 2.872918
## sample estimates:
## ratio of variances
##           1.918571
```

Test de Levene

El test de Levene se puede hacer cuando se requiere comparar las varianzas de más de dos grupos. La hipótesis nula es igual a la del test F.

En R se puede hacer instalando el paquete *car* y usando el comando **leveneTest()**

El comando se llena de la siguiente manera: leveneTest(variable numerica, variable categorica, si el centro es la media o la mediana)

```r
install.packages("car")
```

```r
library(car)
```

```
## Loading required package: carData
```

```r
data(iris)
data("iris")
leveneTest(iris$Petal.Length,iris$Species,center = "mean")
```

```
## Levene's Test for Homogeneity of Variance (center = "mean")
##        Df F value    Pr(>F)
## group   2  20.683 1.216e-08 ***
##       147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# No hay homocedasticidad entre especies
```