Inferencia Estadística

Profesor(es): Jarnishs Beltran
Ayudante: Pablo Rivera

Pauta Ayudantía Nº6
Otoño 2020

Ejercicios

Distr. Normal

i) La función pnorm nos permite calcular la proporción (probabilidad) de valores de una muestra que se encuentran antes o después de un valor Xi, siempre que conozcamos la media y desviación estándar de la muestra (¡asumiendo que los valores se distribuyen normalmente!). La función tiene la siguiente sintáxis:
pnorm(xi, mean = Xbarra, sd = s, lower.tail = TRUE o FALSE)
xi: valor (cuantil) que divide los datos
Xbarra: media de la muestra
s: desviación estándar de la muestra lower.tail TRUE o FALSE: si queremos proporción antes o después del valor xi.

a) Calcular la proporción (o probabilidad) de valores menores de 6.6 mm en una distribución normal de tamaños de semillas, con media = 6.0 mm y desviación estándar = 1.1 mm.

b) y la proporción de valores mayores de 5.4 mm.

c) Calcular la proporción de valores entre 5.4 mm y 6.6 mm

```
#Calcular la proporción (o probabilidad) de valores menores de 6.6 mm en una distribución normal de tamaños de semillas, con
media = 6.0 mm y desviación estándar = 1.1 mm:
pnorm(6.6, mean = 6.0, sd = 1.1, lower.tail = TRUE)
```

```
## [1] 0.7072795
```

```
#y la proporción de valores mayores de 5.4 mm:
pnorm(5.4, mean = 6.0, sd = 1.1, lower.tail = FALSE)
```

```
## [1] 0.7072795
```

```
#también podemos calcular la proporción de valores entre dos valores (o fuera de esos dos valores)
props <- pnorm(c(5.4,6.6), mean = 6.0, sd = 1.1, lower.tail = TRUE)
props
```

```
## [1] 0.2927205 0.7072795
```

```
#proporción de valores entre 5.4 mm y 6.6 mm
props[2] - props[1]
```

```
## [1] 0.4145591
```

ii) La función qnorm en R, realiza el proceso contrario a pnorm, esto es: calcula los valores de Xi que delimitan una proporción en la curva de densidad normal.

a) Valor de Xi antes del cual se encuentra el 95 % de los valores en la curva de densidad normal con media = 0 y desviación estándar de 1.

b) Con media 6.0 y desviación estándar de 1.1.

c) Valor de Xi sobre el cual se encuentra el 5 % de los valores.

```r
#Valor de Xi antes del cual se encuentra el 95 % de los valores en la curva de densidad normal con media = 0 y desviación es
tándar de 1:
qnorm(0.95)
```

```
## [1] 1.644854
```

```r
#Con media 6.0 y desviación estándar de 1.1:
qnorm(0.95, mean = 6.0, sd = 1.1)
```

```
## [1] 7.809339
```

```r
#Valor de Xi sobre el cual se encuentra el 5 % de los valores:
qnorm(0.05, mean = 6.0, sd = 1.1, lower.tail = FALSE)
```

```
## [1] 7.809339
```

iii) Una aplicación práctica (asumiendo que tenemos una muestra de una población con valores distribuidos normalmente) es calcular la proporción de nuestros valores dentro de los límites de la desviación estándar (o un múltiplo de ella).
Primero vamos a usar valores generados usando la función rnorm de R, que produce n valores aleatorios de una distribución con media y desviación estándar dada. Usaremos n = 10000 y media = 0, y desviación estándar = 1.

a) Calcule el porcentaje de datos entre 1 SD.

b) Calcule el porcentaje de datos entre 2 SD.

Ahora usaremos una muestra más pequeña: los datos de altura total de individuos Melocactus intortus.

c) Calcule el porcentaje de datos entre 1 SD.

d) Calcule el porcentaje de datos entre 2 SD.

```r
#valores generados
propg <- rnorm(10000)
#estadísticos
n = length(propg)
mean = mean(propg)
sd = sd(propg)
#función para calcular proporción
proporcion <- function(nSD){
  lo = mean - nSD*sd
  hi = mean + nSD*sd
  percent = sum(propg>=lo
              & propg<=hi)/n *100
}
#cálculo de la proporción (en %) para 1 y 2 desviaciones estándares:
print(paste("Porcentaje de datos entre 1 SD es ",proporcion(1),"%", sep=""))
```

```
## [1] "Porcentaje de datos entre 1 SD es 68.52%"
```

```r
print(paste("Porcentaje de datos entre 2 SD es ",proporcion(2),"%", sep=""))
```

```
## [1] "Porcentaje de datos entre 2 SD es 95.63%"
```

```r
#lectura de los datos
library(readr)
melocactus <- read_csv("melocactus.csv")
#estadísticos
n = length(melocactus$alturatotal)
mean = mean(melocactus$alturatotal)
sd = sd(melocactus$alturatotal)
#función para calcular proporción
proporcion <- function(nSD){
  lo = mean - nSD*sd
  hi = mean + nSD*sd
  percent = sum(melocactus$alturatotal>=lo
              & melocactus$alturatotal<=hi)/n *100
}
#cálculo de la proporción (en %) para 1 y 2 desviaciones estándares:
print(paste("Porcentaje de datos entre 1 SD es ",proporcion(1),"%", sep=""))
```

```
## [1] "Porcentaje de datos entre 1 SD es 71.0344827586207%"
```

```r
print(paste("Porcentaje de datos entre 2 SD es ",proporcion(2),"%", sep=""))
```

```
## [1] "Porcentaje de datos entre 2 SD es 96.551724137931%"
```

En estadística, se llama intervalo de confianza a un par o varios pares de números entre los cuales se estima que estará cierto valor desconocido con una determinada probabilidad de acierto. Podemos estimarlos utilizando las herramientas de cálculo de las probabilidades o proporciones que hemos utilizado, siempre asumiendo que nuestra muestra proviene de una población con distribución normal, y que la muestra es suficientemente grande.

a) calcule el límite inferior del intervalo de confianza 95 %
b) calcule el límite superior del intervalo de confianza 95 %

```r
#estadísticos
media <- mean(melocactus$alturatotal)
ds <- sd(melocactus$alturatotal)
n <- length(melocactus$alturatotal)
#cálculo de los límites del intervalo para 95 % de los valores:
error <- qnorm(0.975)*ds/sqrt(n)
izq <- media-error
der <- media+error
print(paste("Límite inferior del intervalo de confianza 95 %:  ",izq, sep=""))
```

```
## [1] "Límite inferior del intervalo de confianza 95 %:  19.6228158026175"
```

```r
print(paste("Límite superior del intervalo de confianza 95 %: ",der, sep=""))
```

```
## [1] "Límite superior del intervalo de confianza 95 %: 24.2392531628997"
```

Distr. Chi-Cuadrado

Cargue paquete MASS y use la base de datos ”survey”.

a) Realice un cuadro que contenga la una matriz de fumadores (Smoke) vrs ejercicios (Exer).
b) Obtenga el test de independencia. Explore las múltiples formas de desarrollarlo.
c) Realice un grafico de mosaico despliega informacion para examinar la relacion entre dos o más variables categoricas.
d) Obtenga el mosaico para los datos del vector “cuadro”.

```
library(MASS)       # Carga el paquete MASS
data(survey)
head(survey) # se refiere a la base de datos que estamos utilizando y que contiene los datos
```

```
##      Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse    Clap Exer Smoke Height
## 1 Female   18.5   18.0 Right  R on L    92    Left Some Never    173
## 2   Male   19.5   20.5  Left  R on L   104    Left None Regul    178
## 3   Male   18.0   13.3 Right  L on R    87 Neither None Occas     NA
## 4   Male   18.8   18.9 Right  R on L    NA Neither None Never    160
## 5   Male   20.0   20.0 Right Neither    35   Right Some Never    165
## 6 Female   18.0   17.7 Right  L on R    64   Right Some Never    173
##        M.I  Age
## 1   Metric 18.2
## 2 Imperial 17.6
## 3     <NA> 16.9
## 4   Metric 20.3
## 5   Metric 23.7
## 6 Imperial 21.0
```

```
names(survey)# visualiza títulos de las columnas de los datos
```

```
##  [1] "Sex"    "Wr.Hnd" "NW.Hnd" "W.Hnd"  "Fold"   "Pulse"  "Clap"
##  [8] "Exer"   "Smoke"  "Height" "M.I"    "Age"
```

```
cuadro<- table(survey$Smoke, survey$Sex)
cuadro
```

```
##
##         Female Male
##   Heavy      5    6
##   Never     99   89
##   Occas      9   10
##   Regul      5   12
```

```
cuadro2<-table(survey$Smoke, survey$Exer)
cuadro2
```

```
##
##         Freq None Some
##   Heavy    7    1    3
##   Never   87   18   84
##   Occas   12    3    4
##   Regul    9    1    7
```

```
chisq.test(cuadro2)
```

```
## Warning in chisq.test(cuadro2): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  cuadro2
## X-squared = 5, df = 6, p-value = 0.5
```

```
fisher.test(cuadro2,simulate.p.value=TRUE)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based
##  on 2000 replicates)
##
## data:  cuadro2
## p-value = 0.4
## alternative hypothesis: two.sided
```

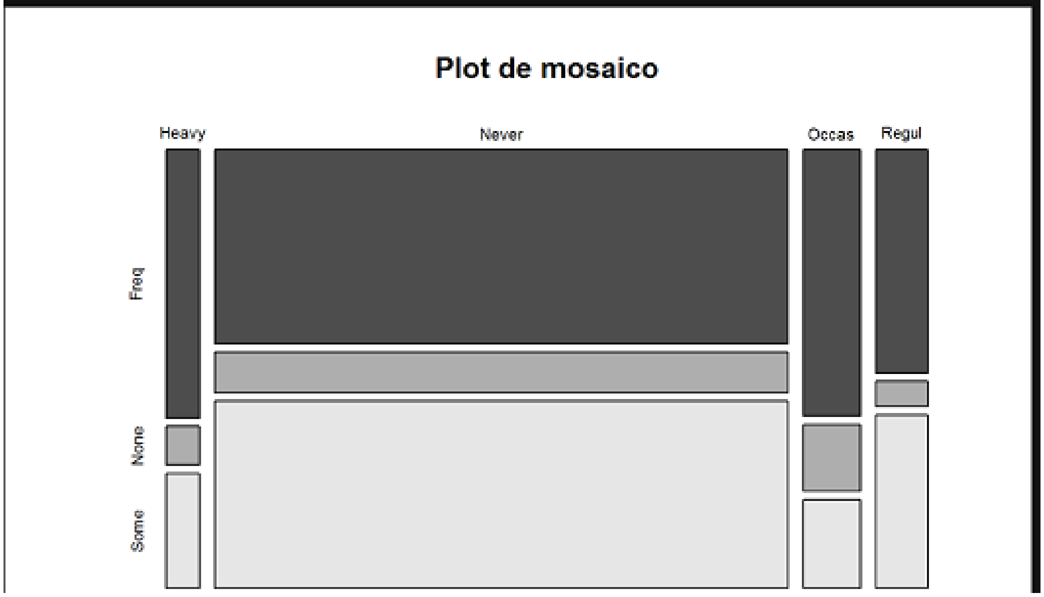
```
fisher.test(cuadro2,simulate.p.value=TRUE,B=5000)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based
##  on 5000 replicates)
##
## data:  cuadro2
## p-value = 0.4
## alternative hypothesis: two.sided
```

```
chisq.test(cuadro2,simulate.p.value=T, B=5000)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 5000
##  replicates)
##
## data:  cuadro2
## X-squared = 5, df = NA, p-value = 0.5
```

```
mosaicplot(cuadro2, color=TRUE, main="Plot de mosaico")
```



Formulas

Z para distribución normal

$$Z = \frac{X_i - \mu}{\sigma}$$

Chi-Cuadrado

$$x^2 = \sum \frac{(O - E)^2}{E}$$

Donde: O= se refiere a las frecuencias observadas E= frecuencias esperadas