

Instrucciones para resolver los ejercicios de estadística descriptiva de las clase 13 y 14 (Estadística (Q))

Ejercicio 1:

```
# para leer datos
```

```
gluco<- read.csv2(file.choose(),header=TRUE,sep="") #gluco es un data frame
```

```
#y buscar el archivo en la compu.
```

```
#O bien:
```

```
gluco<-read.table("C:\\Documents and Settings\\PC\\Escritorio\\datos\\estad  
descriptiva.txt", header = TRUE)
```

```
#o bien
```

```
gluco2<-read.table("C:/Documents and Settings/PC/Escritorio/datos/estad  
descriptiva.txt", header = TRUE)
```

```
#de este modo el objeto gluco es un dataframe.
```

```
#si el archivo que tiene los datos fuera una lista de números sin títulos ni  
nada se puede usar la función scan() de dos maneras:
```

```
    #primera posibilidad
```

```
    graduados<-scan()  
#luego, tipear los datos (o usar copy-paste)
```

```
    #segunda posibilidad  
    graduados<-scan("C:\\Documents and  
Settings\\PC\\Escritorio\\datos\\graduados.txt")
```

```
    attach(gluco) #esta instruccion le permite a R trabajar con dietaa, dietab,  
dietacc(las columnas del archivo original) como si fueran objetos (Vectores  
columna con nombre) por separado
```

```
#medidas de resumen:  
summary(gluco)
```

```
#media muestral  
mean(gluco)  
mean(dietaa)
```

```
#una de las ventajas del R es que trabaja con los objetos: mean(vector) devuelve  
un número, mean(dataframe) devuelve un vector.
```

```
#mediana muestral  
median(dietaa)  
sort(dietaaa) #ordena de menor a mayor al vector
```

```
dietaa[1:12]  
sort(dietaa)[1:12]
```

```

#media alfa-podada
mean(dietaa,trim=0.1) #poda el 10% de la muestra (ordenada) a cada lado, y
luego calcula la media de lo que resta. Debería coincidir con
a<-sort(dietaa)
mean(a[11:90])

#varianza muestral y desvío estándar muestral (divide por n-1)
var(dietaa)
cov(dietaa,dietab)

var(gluco) #en la diagonal aparecen las varianzas, en el lugar ij la
covarianza entre la columna i y la columna j de gluco

sqrt(var(dietaa))
sqrt(var(gluco)) #notar la facilidad de R para aplicar funciones numéricas
coordenada a coordenada

#el coeficiente de correlación muestral
cor(dietaa,dietab)
#que debería coincidir con
cov(dietaa,dietab)/sqrt(var(dietaa)*var(dietab))

#distancia intercuartil o intercuartos (ver más adelante)

#cuantiles o percentiles muestrales
quantile(dietaa) #da por default los cuantiles 0% 25% 50% 75% 100%
quantile(dietaa,0.6)
quantile(dietaa,0.9) #da igual a
0.9*sort(dietaa)[90]+ 0.1*sort(dietaa)[91]
#pues 0.9*101 = 90.9
quantile(dietaa, probs=c(1,2,5,10,50)/100)

#rango muestral
range(dietaa)
range(dietab)
range(dietac)

#mad (sin estandarizar)
mad(dietaa,constant=1)
#mad (estandarizada)
mad(dietaa)

#distancia intercuartil o intercuartos, ver más adelante

#histograma
hist(dietaa)
hist(dietab)
hist(dietac)

#para ver los tres histogramas en una misma pantalla, subdividimos el área
gráfica en 4 partes (sólo usamos 3 de ellas)

par(mfrow=c(2,2))
hist(dietaa)
hist(dietab)
hist(dietac)

```

```

#para obtener la tabla de frecuencias relativas
hist(dietaa,plot = FALSE)
#o bien, asignarla a alguna lista:
ha<-hist(dietaa)
#ha$counts guarda la cantidad total de observaciones en cada intervalo

#variantes: imponiendo los puntos de cortes
par(mfrow=c(1,1)) #para volver a ver un gráfico en la pantalla
hist(dietaa,breaks=c(65,75,85,95,105,125))
#variantes: imponiendo la cantidad de puntos de cortes
hist(dietaa,breaks=12)
hist(dietaa,breaks=20)
#variantes: no da las frecuencias relativas sino área total 1 para que distintos
histogramas (basados en muestras de distinto número de observaciones) sean
comparables
hist(dietaa,freq=FALSE)

#diagramas de tallo - hoja
stem(dietaa)
stem(dietab)
stem(dietac)

#boxplot
boxplot(dietaa)

ba<-boxplot(dietaa)
#ba guarda una lista con ba$stats los valores necesarios para hacer el dibujo:
mediana, cuartos, puntos hasta donde se extienden los "bigotes"
ba$stats

median(dietaa)
#cuarto inferior
median(sort(dietaa)[1:50])
#cuarto superior
median(sort(dietaa)[51:100])
#distancia intercuartos (que es una medida robusta de dispersión)
median(sort(dietaa)[51:100])-median(sort(dietaa)[1:50])

#boxplot de los tres conjuntos de datos en el mismo gráfico
boxplot(gluco)

#qqplot
qqnorm(dietaa)
qqline(dietaa) #superpone una recta
qqnorm(dietab)
qqnorm(dietaa)

```

## Ejercicio 2:

#a) primero generamos mil v.a. uniformes y luego hacemos el histograma correspondiente.

```

xbarra<-runif(1000)
hist(xbarra)

```

#b) primero generamos dos v.a. uniformes o un vector uniforme bivariado, con ambas coordenadas independientes

```
aa<-runif(2)
xbarra2<-mean(aa)
```

```
#ahora repetimos 1000 veces
#con un for:
xbarra2<-rep(0,1000)
for (i in 1:1000)
{
aa<-runif(2)
xbarra2[i]<-mean(aa)
}
```

```
hist(xbarra2)
```

```
#o vectorialmente;
a1<-runif(2000)
#convertimos al vector en matriz
matriz2<-matrix(a1,nrow=2, ncol=1000)
```

```
#aplicamos el promedio por columnas
xbarra2<-apply(matriz2,2,mean)
```

```
hist(xbarra2)
```

#observar que si uno repite todo el ítem b) con otros dos mil datos, obtiene un histograma ligeramente diferente

```
#b) ídem 5
matriz5<-matrix(runif(5000),nrow=5, ncol=1000)
```

```
#chequeamos
dim(matriz5)
```

```
xbarra5<-apply(matriz5,2,mean)
```

# Para comparar los 3 histogramas en un mismo gráfico, dividimos la pantalla en 6:

```
par(mfrow=c(2,3))
hist(xbarra,probability=T)
hist(xbarra2,probability=T)
hist(xbarra5,probability=T)
```

```
#d) ídem 30
matriz30<-matrix(runif(30000),nrow=30, ncol=1000)
xbarra30<-apply(matriz30,2,mean)
hist(xbarra30,probability=T)
```

```
#e) ídem 500
ídem 500
matriz500<-matrix(runif(500000),nrow=500, ncol=1000)
xbarra500<-apply(matriz500,2,mean)
hist(xbarra500,probability=T)
```

```

# idem 1000
matriz1000<-matrix(runif(1000000),nrow=1000, ncol=1000)
xbarra1000<-apply(matriz1000,2,mean)
hist(xbarra1000,probability=T)

hist(xbarra,probability=T)
hist(xbarra2,probability=T)
hist(xbarra5,probability=T)
hist(xbarra30,probability=T)
hist(xbarra500,probability=T)
hist(xbarra1000,probability=T)

#Mejor aún, pongamos la misma escala en x para comparar

hist(xbarra,probability=T,xlim=c(0,1))
hist(xbarra2,probability=T,xlim=c(0,1))
hist(xbarra5,probability=T,xlim=c(0,1))
hist(xbarra30,probability=T,xlim=c(0,1))
hist(xbarra500,probability=T,xlim=c(0,1))
hist(xbarra1000,probability=T,xlim=c(0,1))

#notemos que queda distorsionado si no obligamos a que la escala en el eje y sea
la misma

hist(xbarra,probability=T,xlim=c(0,1),ylim=c(0,45))
hist(xbarra2,probability=T,xlim=c(0,1),ylim=c(0,45))
hist(xbarra5,probability=T,xlim=c(0,1),ylim=c(0,45))
hist(xbarra30,probability=T,xlim=c(0,1),ylim=c(0,45))
hist(xbarra500,probability=T,xlim=c(0,1),ylim=c(0,45))
hist(xbarra1000,probability=T,xlim=c(0,1),ylim=c(0,45))

# o bien, hagamos los boxplot de los 6 grupos de datos en el mismo gráfico
par(mfrow=c(1,1))
boxplot(xbarra,xbarra2,xbarra5,xbarra30,xbarra500,xbarra1000)

#medias muestrales
mean(xbarra)
mean(xbarra2)
mean(xbarra5)
mean(xbarra30)
mean(xbarra500)
mean(xbarra1000)

var(xbarra)
var(xbarra2)
var(xbarra5)
var(xbarra30)
var(xbarra500)
var(xbarra1000)

e1<-(xbarra-0.5)*sqrt(1)/sqrt(1/12)
e2<-(xbarra2-0.5)*sqrt(2)/sqrt(1/12)
e5<-(xbarra5-0.5)*sqrt(5)/sqrt(1/12)
e30<-(xbarra30-0.5)*sqrt(30)/sqrt(1/12)
e500<-(xbarra500-0.5)*sqrt(500)/sqrt(1/12)
e1000<-(xbarra1000-0.5)*sqrt(1000)/sqrt(1/12)

```

```
boxplot(e1,e2,e5,e30,e500,e1000)

#Para generar observaciones con distribución Cauchy (no tiene esperanza ni
varianza finita, es simétrica alrededor del cero, su densidad es  $f(x) = 1 / (\pi * (1 + (x^2)))$  ), la instrucción es
#rcauchy(n)

matriz1<-matrix(rcauchy(1000),nrow=1, ncol=1000)
xbarra<-apply(matriz1,2,mean)
matriz2<-matrix(rcauchy(2000),nrow=2, ncol=1000)
xbarra2<-apply(matriz2,2,mean)
matriz5<-matrix(rcauchy(5000),nrow=5, ncol=1000)
xbarra5<-apply(matriz5,2,mean)
matriz30<-matrix(rcauchy(30000),nrow=30, ncol=1000)
xbarra30<-apply(matriz30,2,mean)
matriz500<-matrix(rcauchy(500000),nrow=500, ncol=1000)
xbarra500<-apply(matriz500,2,mean)
matriz1000<-matrix(rcauchy(1000000),nrow=1000, ncol=1000)
xbarra1000<-apply(matriz1000,2,mean)
```