# Ejemplo:

Para estudiar el examen de ingreso a la UES, se selecciona aleatoriamente una muestra de 60 alumnos, las notas de estos alumnos son las siguientes:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4.47 | 4.47 | 3.48 | 5.0 | 3.42 | 3.78 | 3.1 | 3.57 | 4.2 | 4.5 |
| 3.6 | 3.75 | 4.5 | 2.85 | 3.7 | 4.2 | 3.2 | 4.05 | 4.9 | 5.1 |
| 5.3 | 4.16 | 4.56 | 3.54 | 3.5 | 5.2 | 4.71 | 3.7 | 4.78 | 4.14 |
| 4.14 | 4.8 | 4.1 | 3.83 | 3.6 | 2.98 | 4.32 | 5.1 | 4.3 | 3.9 |
| 3.96 | 3.54 | 4.8 | 4.3 | 3.39 | 4.47 | 3.19 | 3.75 | 3.1 | 4.7 |
| 3.69 | 3.3 | 2.85 | 5.25 | 4.68 | 4.04 | 4.44 | 5.43 | 3.04 | 2.95 |

# ANÁLISIS ESTADÍSTICO DE LOS DATOS.

1º) Visualiza el directorio por defecto y activa su directorio de trabajo getwd()

setwd("C:/Curso R2012")

2º) Crea un nuevo Script y llámale "Script08-DatosContinuos" 3º) Crea el vector que contendrá los datos.

Notas <- c(4.47, 4.47); Notas

data.entry(Notas) Notas length(Notas)

4º) Guarda el vector de datos en un archivo. write(Notas, "Notas.txt")

5º) Limpia el área de trabajo (Workspace) ls()

rm(list=ls(all=TRUE)) ls()

6º) Lee o recupera el vector de datos desde el archivo de texto.

X <- scan("Notas.txt", what = double(0), na.strings = "NA", flush=FALSE) ls()

# Si el vector contiene valores reales se ocupa: what = double(0)

7º) Crea la tabla de frecuencias.

# Define el número k de los intervalos o clases.

# Usa el Método de Herbert A. Sturges para determinar dicho número. n <- length(X); n

k <- 1+3.322\*logb(n, 10); k k <- round(k); k

# Calcula el ancho o amplitud a de cada intervalo a=rango/k rango <- max(X)-min(X); rango

a=rango/k; a

a <- round(a, 3); a

# Define los límites y puntos medios de cada uno de los k intervalos limites <- seq(from=min(X)-0.01/2, to=max(X)+0.01/2, by=a); limites options(digits=4)

ci <- cbind(1:k); ci

for(i in 2:length(limites)) ci[i-1, 1] <- (limites[i] + limites[i-1])/2 ci

# Encuentra las frecuencias absolutas fi para cada intervalo. options(digits=2)

fi <- cbind(table(cut(X, breaks = limites, labels=NULL, include.lowest=FALSE, right=FALSE, dig.lab=4))); fi

# **breaks** es un vector o secuencia de cortes 1:6, o el número de clases.

# **labels** indica que no hay nombres para los intervalos o clases, por defecto las etiquetas tienen la notación (a, b].

# **include.lowest** indica que si un X[i] es igual al corte inferior (0 superior, para right=FALSE) el valor debe ser incluido.

# **right** indica que sí el intervalo debe ser cerrado a la derecha y abierto a la izquierda, o viceversa.

# **dig.lab** es un entero el cual es usado cuando las etiquetas no son dadas, determina el número de dígitos usado en el formato de números de cortes.

# Encuentra las frecuencias relativas o proporciones fri. options(digits=4)

fri <- fi/n; fri

# Encuentra las frecuencias acumuladas ascendentes Fi options(digits=2)

Fi <- cumsum(fi); Fi

# Encuentra las frecuencias relativas acumuladas Fri options(digits=4)

Fri <- Fi/n; Fri

# Completa la tabla de frecuencias.

tablaFrec <- data.frame(ci=ci, fi=fi, fri=fri, Fi=Fi, Fri=Fri); tablaFrec

# Nuevamente puede usar el comando xtable para importar a código LATEX.

8º) Crea el histograma de frecuencias

h <- hist(X, breaks=c(limites[1]-a, limites, limites[k+1]+a), freq = TRUE, probability = FALSE, include.lowest = FALSE,right = TRUE, main = "Histograma de frecuencias", col="lightyellow", lty=1, border="purple", xlab=" Notas de aspirantes", ylab="Frecuencia (fi)", axes=TRUE, labels=FALSE)

text(h$mids, h$density, h$counts, adj=c(0.5, -0.5), col="red") rug(jitter(X)) # adiciona marcas de los datos

# h es un objeto del tipo lista que contiene atributos del histograma is.list(h); h

9º) Aproxima al histograma la función de densidad normal

h <- hist(X, breaks=c(limites[1]-a, limites, limites[k+1]+a), freq = FALSE, probability = TRUE, include.lowest = FALSE, right = TRUE,

main="Aproximación a una Normal\n", col="lightyellow",lty=1,border="purple", xlab="Notas de aspirantes\n", ylab="Frecuencia relativa (fri)",

axes=TRUE, labels=FALSE)

text(h$mids, h$density, h$counts, adj=c(0.5, 0.2), col="red") rug(jitter(X)) # adiciona marcas de los datos

curve(dnorm(x, mean=mean(X), sd=sd(X)), col = 2, lty = 2,lwd = 2, add = TRUE)

10º) Crea el polígono de frecuencias

h <- hist(X, breaks=c(limites[1]-a, limites, limites[k+1]+a), freq = TRUE, probability=FALSE, include.lowest=FALSE,right=TRUE,

main = "Polígono de frecuencias",col="lightyellow", lty=1, border="purple", xlab=" Notas de aspirantes", ylab="Frecuencia (fi)", axes=TRUE, labels=FALSE)

text(h$mids, h$density, h$counts, adj=c(0.5, -0.5), col="red") rug(jitter(X)) # adiciona marcas de los datos

vCi <- c(h$mids[1]-a, h$mids, h$mids[k+1]+a); vCi vfi <- c(0, h$counts, 0); vfi

lines(vCi, vfi, col="blue", type="l")

11º) Crea la Ojiva ascendente o polígono de frecuencias acumuladas ascendentes Fia <- c(0, Fi); Fia

plot(limites, Fia, type = "p", pch=1, col = "blue", main="Ojiva ascendente", xlab="Notas de aspirantes", ylab="Frecuencia acumulada (Fi)")

text(limites, h$density, Fia, adj=c(0.5, -0.5), col="red") lines(limites, Fia, col="black", type="l")

12º) Calcula los principales estadísticos descriptivos de la variable

# Calcula la moda, ya que el R no proporciona una función para eso. options(digits=4)

for(i in 1:k) if (fi[i] == max(fi)) break()

if(i > 1) moda <- limites[i]+((fi[i]-fi[i-1])/((fi[i]-fi[i-1])+(fi[i]-fi[i+1]) ))\*a else moda <- limites[i]+(fi[i]/(fi[i]+(fi[i]-fi[i+1])))\*a

moda

# Calcula los cuartiles: Q1, Q2, Q3 Q <- 1:3

for(v in 1:3) for(i in 1:k) if (Fi[i] > (v\*25\*n)/100)

{

Q[v] <- limites[i]+(((25\*v\*n/100)-Fi[i-1])/fi[i])\*a break

} Q

# Calcula los principales estadísticos.

estadisticos <- rbind(media=sum(tabEstad$cifi)/n, moda=moda, Q1=Q[1], Q2=Q[2], Q3=Q[3], rango=max(X)-min(X), varianza=sum(tabEstad$ciMedia2fi)/n,

Desviacion=sqrt(sum(tabEstad$ciMedia2fi)/n), CoeficienteVariacion=sqrt(sum(tabEstad$ciMedia2fi)/n)/(sum(tabEstad$cifi)/n),

CAfisher=(sum(tabEstad$ciMedia3fi)/n)/sqrt(sum(tabEstad$ciMedia2fi)/n)^3, CoeficienteCurtosis=((sum(tabEstad$ciMedia4fi)/n)/sqrt(sum(tabEstad$ciMedia2fi)/n)^4)-3)

estadisticos

13º) Otros gráficos: # Gráfico de cajas

boxplot(X, main="Gráfico de caja", xlab="Notas", notch=FALSE, data=parent.frame(), plot=TRUE, border="red", col="yellow",horizontal=TRUE)

# Observación: en la función boxplot(), sí plot es FALSE se produce un resumen de los valores (los cinco números).

# Una variante del boxplot, es el notched boxplot de McGill, Larsen y Tukey, el cual adiciona intervalos de confianza para la mediana, representados con un par de cuñas a los lados de la caja:

windows()

boxplot(X, main="Gráfico de caja", xlab="X = Notas", notch=TRUE, data=parent.frame(), plot=TRUE, border="red", col="yellow",horizontal=TRUE)

# Varios gráficos en una misma ventana

par(mfrow=c(1,2)) # Divide la ventana gráfica en dos partes (1 fila, 2 columnas) mtext(side=3, line=0, cex=2, outer=T, "Titulo para Toda la Página")

hist(X); boxplot(X)

# Ejemplo:

Se selecciona aleatoriamente una muestra de 18 personas adultas, para estudiar si existe relación entre su estado civil y su ocupación.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Estado | casado | soltero | soltero | casado | acompañado | soltero | casado |
| Ocupación | desocupado | estudia | trabaja | estudia | trabaja | desocupado | trabaja |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| casado | acompañado | acompañado | casado | soltero | acompañado | casado | soltero |
| estudia | desocupado | estudia | trabaja | estudia | desocupado | desocupado | estudia |

|  |  |  |
| --- | --- | --- |
| soltero | casado | soltero |
| trabaja | desocupado | trabaja |

# REALICE UN ANÁLISIS ESTADÍSTICO DE LOS DATOS.

1º) Activa tu directorio de trabajo. getwd()

setwd("C:/Curso R2020")

2º) Limpia de objetos el área de trabajo (Workspace). ls()

rm(list=ls(all=TRUE)) ls()

3º) Crea un nuevo Script y llámale "Script09-DatosBivariados1". 4º) Crea en Excel una hoja de datos con dos columnas o variables

# Recuerda que al guardar la hoja, el tipo de archivo es de extensión .csv(delimitado por comas).

# Llámale al archivo: HojaCat

# Otra forma de crear la hoja de datos es la siguiente (Vea la Práctica 04):

# Primero crear las dos variables categóricas en un editor de texto como NotePad o WordPad, colocando nombre a cada columna, y llamándole "HojaCat.txt".

# Luego puede leer o recuperar este archivo con la función read.table() HojaCat <- read.table("HojaCat.txt", header=TRUE)

HojaCat

5º) Recupera desde el entorno de R la hoja de datos de Excel.

HojaCat <- read.csv("HojaCat.csv", strip.white=TRUE);HojaCat

6º) Conecta la hoja de datos a la segunda ruta o lista de búsqueda. attach(HojaCat, pos=2) # pos especifica la posición donde buscar la conexión search()

7º) Crea una tabla de contigencia o de doble entrada tablaCont <- table(HojaCat); tablaCont length(HojaCat)

# Note que esta instrucción no devuelve el número de elementos, sino más bien el número de variables o columnas consideradas en el conjunto de datos.

# Encuentra la suma de cada fila de la tabla de contingencia # Distribución marginal de X=Estado civil

suma.filas <- apply(tablaCont, 1, sum); suma.filas # El 1 indica que son totales por fila

# Encuentra la suma de cada fila de la tabla de contingencia # distribución marginal de Y=Ocupación

suma.columnas <- apply(tablaCont,2,sum); suma.columnas # 2 indica que son totales por columna

# Gráficos de barras para tabla de contingencia. # Barras apiladas

barplot(t(tablaCont), main="Gráfico de barras (Estado, Ocupación)", xlab="Estado civil", ylab="Ocupación", legend.text=TRUE)

# Note que t(tablaCont) indica que las barras representan el Estado civil de los encuestados y que éstas se subdividen en cada una de las diferentes ocupaciones consideradas.

# En caso de usar únicamente tablaCont; las barras representarán las diferentes ocupaciones y éstas estarán subdividas en cada uno de los estados civiles.

# Barras agrupadas

barplot(t(tablaCont), main="Gráfico de barras (Estado, Ocupación)", xlab="Estado civil", ylab="Ocupación", beside=TRUE, legend.text=TRUE)

# Note que la instrucción beside =TRUE, indica que por cada una de las diferentes ocupaciones se creará una barra para cada estado civil. Note que al usar beside =FALSE se obtiene el mismo gráfico de la instrucción anterior.

barplot(tablaCont, main="Gráfico de barras (Ocupación, Estado)", xlab="Ocupación\n", ylab="Estado civil", beside=TRUE, legend.text=TRUE)

8º) Calcula tablas de proporciones o de probabilidades.

# Guardar las todas las opciones iniciales y modificar número de decimales op <- options()

options(digits=3) # sólo imprime 3 lugares decimales options('digits')

# Proporciones basadas en el total de la muestra, la suma de filas y columnas suman 1. propTotal <- prop.table(tablaCont); propTotal

barplot(t(propTotal), main="Gráfico de barras (Estado, Ocupación)", xlab="Estado civil\n", ylab="Ocupación", beside=TRUE, legend.text=TRUE)

# Proporciones basadas en el total por fila, cada fila suma 1. propFila <- prop.table(tablaCont, 1); propFila

# Total por fila se indica en 1

barplot(t(propFila), main="Gráfico de barras (Estado, Ocupación)", xlab="Estado civil\n", ylab="Ocupación", beside=TRUE, legend.text=TRUE)

# Proporciones basadas en el total por columna, cada columna suma 1. propColum <- prop.table(tablaCont, 2); propColum

# Total por columna se indica en 2

barplot(propColum, main="Gráfico de barras (Ocupación, Estado)", xlab="Ocupación\n", ylab="Estado civil", beside=TRUE, legend.text=TRUE)

9º) Otra forma de elaborar los gráficos de barras para el vector bidimensional categórico. # Gráfico de barras no apiladas y colocación de leyenda

barplot(table(Ocupacion, Estado), main="Gráfico de barras (Estado, Ocupación)", xlab = "Estado civil", ylab="Ocupación", beside=TRUE, legend.text=T)

barplot(table(Estado, Ocupacion), main="Gráfico de barras (Ocupación, Estado)", xlab = "Ocupación", ylab="Estado civil", beside=TRUE, legend.text=TRUE)

barplot(table(Estado, Ocupacion), main="Gráfico de barras (Ocupación, Estado)", xlab="Ocupación", ylab="Estado civil", beside=TRUE, legend.text=c("menor que 2", "2-3", "mayor que 3"))

# Note que se puede definir a conveniencia la leyenda que se desea incorporar en el gráfico con la instrucción legend.text

10º) Realizar la prueba o contraste Chi-cuadrado de independencia prueba <- chisq.test(tablaCont); prueba

# Tenga en cuenta que las frecuencias esperadas deben ser todas mayores a 5

# Frecuencias absolutas esperadas para la prueba Chi-cuadrada prueba$expected # fij = fi./No. column

# Ejemplo 1:

Se están estudiando tres procesos (A, B, C) para fabricar pilas o baterías. Se sospecha que el proceso incide en la duración (en semanas) de las baterías, es decir, que la duración (en semanas) de los procesos es diferente. Se seleccionan aleatoriamente cinco baterías de cada proceso y al medirles aleatoriamente su duración los datos que se obtienen, son los siguientes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Proceso | Duración (en semanas) | | | | |
| A | 100 | 96 | 92 | 96 | 92 |
| B | 76 | 80 | 75 | 84 | 82 |
| C | 108 | 100 | 96 | 98 | 100 |

Realice un análisis estadístico de los datos.

# Nota: Cuando los datos bivariados se obtiene de una variable cualitativa y otra cuantitativa, los valores cuantitativos de cada categoría o nivel de la variable cualitativa se consideran como muestras o grupos diferentes. Cada muestra se describe aplicando la representación y medidas de resumen de una variable univariada pero de manera conjunta.

1º) Activa tu directorio de trabajo. getwd()

setwd("C:/Curso R2012")

2º) Crea un nuevo script y llámale "Script10-DatosBivariados2"

3º) Crea un vector de datos para cada proceso descrito en el problema.

A <- c(100,96,92,96,92); A

B <- c(76,80,75,84,82); B

C <- c(108,100,96,98,100); C

4º) Crea una hoja de datos teniendo como componentes (columnas) los tres vectores (se puede hacer pues el número de datos en cada proceso es igual, de lo contrario se debería de crear dos variables una para la duración de cada proceso y otra para identificar a qué proceso corresponde).

Baterias <- data.frame(procesoA=A, procesoB=B, procesoC=C); Baterias # Para editar los datos puede utilizar la función fix()

fix(Baterias)

5º) Guarda la hoja de datos en un archivo.

write.table(Baterias, file="Baterias.txt", append=FALSE, quote=TRUE, sep=" ", na="NA", col.names=TRUE)

6º) Elimina todos objetos que existen en el espacio de trabajo (Workspace) ls(); rm(list=ls(all=TRUE)); ls()

7º) Recupera la hoja de datos, para probar si fue guardada.

Baterias <- read.table("Baterias.txt", header=TRUE); Baterias

8º) Conecta o adjunta la hoja de datos a la segunda ruta o lista de búsqueda. attach(Baterias, pos=2)

search()

9º) Dibuja un gráfico horizontal de puntos para los tres procesos.

stripchart(Baterias, main="Gráfico de puntos para los tres procesos", method = "stack", vertical = FALSE, col="blue", pch=1, xlab="Duración (semanas)", ylab="Proceso")

# Note que con ayuda de este gráfico podemos observar sí los tres procesos se comportan de manera distinta o parecida en cuanto a duración en semanas de las baterías.

10º) Muestra un resumen estadístico para los tres procesos. summary(Baterias)

11º) Dibuja un gráfico de cajas (box-plot) para los tres procesos. # Horizontal

boxplot(Baterias, width=NULL, varwidth=TRUE, names, add= FALSE, horizontal = TRUE, main="Gráfico de caja por proceso", border=par("fg"), col=c("yellow", "cyan", "red"), xlab = "Duración (semanas)", ylab="Proceso")

# Vertical

boxplot(Baterias, width=NULL, varwidth=TRUE, names, add= FALSE, horizontal = FALSE, main="Gráfico de caja por proceso", border=par("fg"), col=c("yellow", "cyan", "red"), xlab = "Duración (semanas)", ylab="Proceso")

12º) Presenta la matriz de covarianzas muestral. options(digits=3) # sólo imprime 3 lugares decimales S <- var(Baterias); S

13º) Presenta la desviación estándar de cada proceso. desv <- sd(Baterias); desv

14º) Realiza un análisis de varianza de una vía, para probar la hipótesis nula de que el proceso no influye en la duración de las baterías, es decir, que no hay diferencias entre los tres procesos.

*H*0 : *A*  *B*  *C*

, no existe diferencias entre los tres procesos.

*H*1 : *i*   *j* , por lo menos un par *i* 

*j* , de procesos difieren en la duración de las baterías.

# Concatena los tres vectores dentro de un vector simple, junto con un vector factor indicador de la categoría o tratamiento (A, B, C) que origina cada observación. El resultado es un data.frame que tiene como componentes los dos vectores anteriores.

Baterias <- stack(Baterias); Baterias

names(Baterias) # Muestra los encabezados de los vectores

# Prueba de igualdad de medias por descomposición de la varianza en dos fuentes de variación: la variabilidad que hay entre los grupos (debida a la variable independiente o los tratamientos), y la variabilidad que existe dentro de cada grupo (variabilidad no explicada por los tratamientos).

aov.Baterias <- aov(values~ind, data=Baterias)

# values~ind relaciona los valores muestrales con los respectivos grupos summary(aov.Baterias)

# Note que es necesario la instrucción anterior para poder visualizar la tabla ANOVA

# Decisión: ya que α = 0.05 > p-value obtenido, entonces se rechaza Ho

# Prueba de igualdad de medias en un diseño de una vía (o unifactorial) asumiendo que las varianzas de los grupos son iguales

oneway.test(values~ind, data=Baterias, var.equal = TRUE)

15º) Deshace la concatenación del vector de valores y el vector indicador de categoría.

Baterias = unstack(Baterias);Baterias

16º) Desconecta la hoja de datos de la segunda ruta o lista de búsqueda. detach(Baterias, pos=2); search()

# Ejemplo 2:

Suponga que un estudiante hace una encuesta para evaluar sí los estudiantes que fuman estudian menos que los que no fuman. Los datos registrados son:

|  |  |  |  |
| --- | --- | --- | --- |
| Persona | Fuma | Cantidad  (horas estudiando) | Código para el  intervalo |
| 1 | Si | menos de 5 horas | 1 |
| 2 | No | 5-10 horas | 2 |
| 3 | No | 5-10 horas | 2 |
| 4 | Si | más de 10 horas | 3 |
| 5 | No | más de 10 horas | 3 |
| 6 | Si | menos de 5 horas | 1 |
| 7 | Si | 5-10 horas | 2 |
| 8 | Si | menos de 5 horas | 1 |
| 9 | Si | más de 10 horas | 3 |
| 10 | Si | 5-10 horas | 2 |

# REALICE UN ANÁLISIS ESTADÍSTICO DE LOS DATOS.

1º) Activa tu directorio de trabajo. getwd()

setwd("C:/Curso R2020")

2º) Crea un nuevo script y llámale "Script11-DatosBivariados3" 3º) Crea dos vectores con los datos.

Fuma = c("Si","No","No","Si","No","Si","Si","Si","No","Si"); Fuma

Cantidad = c(1,2,2,3,3,1,2,1,3,2); Cantidad

4º) Crea una hoja de datos que tenga como componentes o columnas los dos vectores.

Estudia <- data.frame(Fuma=Fuma, Cantidad=Cantidad); Estudia

# Puedes editar los datos utilizando fix(Estudia)

5º) Guarda la hoja de datos en un archivo.

write.table(Estudia, file="Estudia.txt", append=FALSE, quote=TRUE, sep=" ", na="NA", col.names=TRUE)

6º) Elimina los objetos almacenados en el área de trabajo (Workspace). ls()

rm(list=ls(all=TRUE)) ls()

7º) Recupera desde el archivo la hoja de datos. Estudia <- read.table("Estudia.txt", header=TRUE) Estudia

8º) Conecta la hoja de datos a la segunda ruta o lista de búsqueda, attach(Estudia, pos=2)

search()

9º) Crea una tabla de contigencia o de doble entrada. tablaCont <- table(Estudia)

tablaCont

10º) Calcula las tablas de proporciones o de probabilidades. options(digits=3) # sólo imprime 3 lugares decimales

# Proporciones basadas en el total de la muestra, la suma de filas y columnas suman 1 propTotal <- prop.table(tablaCont); propTotal

# Proporciones basadas en el total por fila, cada fila suma 1 propFila <- prop.table(tablaCont, 1)

propFila

# Proporciones basadas en el total por columna, cada columna suma 1 propCol <- prop.table(tablaCont, 2)

propCol

11º) Construya los gráficos de barras de la variable bidimensional.

# Gráfico de barras apiladas con la frecuencia de Cantidad como altura barplot(table(Estudia$Cantidad, Estudia$Fuma), beside = FALSE, horizontal=FALSE, main="Gráfico de barras (Fuma, Cantidad de horas de estudio)", legend.text =T, xlab="Fuma", ylab="Cantidad de horas-estudio")

# Gráfico de barras apiladas con la frecuencia de Fuma como altura

barplot(table(Estudia$Fuma, Estudia$Cantidad), beside = FALSE, horizontal=FALSE,main="Gráfico de barras (Cantidad de horas de estudio,Fuma)", legend.text =T, xlab="Cantidad de horas-estudio", ylab="Fuma")

# Gráfico de barras no apiladas y colocación de leyenda # Crear un factor para los nombres en la leyenda Fuma=factor(Estudia$Fuma); Fuma

barplot(table(Estudia$Cantidad, Estudia$Fuma), main="Gráfico de barras (Fuma, Cantidad de horas de estudio)", xlab="Fuma", ylab="Cantidad de horas-estudio", beside=TRUE, legend.text=T)

barplot(table(Estudia$Cantidad, Estudia$Fuma), main="Gráfico de barras (Fuma, Cantidad de horas de estudio)", xlab="Fuma", ylab="Cantidad de horas-estudio", beside=TRUE, legend.text=c("menor que 5", "5-10", "mayor que 10"))

12º) Realiza la prueba o contraste Chi-cuadrado para las probabilidades dadas chisq.test(tablaCont)

# Sí p-value >  aceptar *H*0 : Las variables son independientes

# Recuerde que las frecuencias esperadas deben ser mayores a 5 para poder utilizarlas.

# Probabilidades esperadas para la prueba Chi-cuadrada chisq.test(tablaCont) $expected

# Ejemplo:

El tiempo que tarda un sistema informático en red en ejecutar una instrucción depende del número de usuarios conectados a él. Sí no hay usuarios el tiempo es cero. Se tienen registrados los siguientes datos:

|  |  |
| --- | --- |
| No. usuarios | Tiempo de ejecución |
| 10 | 1.0 |
| 15 | 1.2 |
| 20 | 2.0 |
| 20 | 2.1 |
| 25 | 2.2 |
| 30 | 2.0 |
| 30 | 1.9 |

# REALICE UN ANÁLISIS ESTADÍSTICO.

1º) Activa tu directorio de trabajo getwd()

setwd("C:/Curso R2020")

2º) Crea un nuevo script y llámale "Script11-DatosBivariados4" 3º) Crea los dos vectores para las dos variables

# Número de usuarios = Variable explicativa o independiente usuarios <- c(10, 15, 20, 20, 25, 30, 30); usuarios

tiempo = c(1.0, 1.2, 2.0, 2.1, 2.2, 2.0, 1.9); tiempo

4º) Crea una hoja de datos que tenga como componentes o columnas los dos vectores.

Sistema <- data.frame(Usuarios=usuarios, Tiempo=tiempo);Sistema # Para editar o ampliar los datos puede utilizar la función fix() fix(Sistema)

5º) Guarda la hoja de datos en un archivo.

write.table(Sistema, file="Sistema.txt", append=FALSE, quote=TRUE, sep=" ", na="NA", col.names = TRUE)

6º) Elimina los objetos almacenados en el área de trabajo (Workspace). ls(); rm(list=ls(all=TRUE)); ls()

7º) Recupera la hoja de datos.

Sistema <- read.table("Sistema.txt", header=TRUE); Sistema

8º) Conecta la hoja de datos a la segunda ruta o lista de búsqueda. attach(Sistema, pos=2); search()

9º) Muestra un resumen de principales estadísticos de las variables. summary(Sistema)

cov(Sistema) # Matriz de covarianzas

cor(Sistema, use = "all.obs", method="pearson") # Matriz de correlaciones

10º) Elabora un gráfico de dispersión para analizar alguna relación entre las variables.

plot(Usuarios, Tiempo, xlim= c(5, 35), ylim= c(0.0, 2.5), type = "p", pch=1, col = "blue", main = "Gráfico de dispersión (Usuarios, Tiempo)", xlab="Número de usuarios", ylab="Tiempo de ejecución")

11º) Para identificar un punto arbitrario, se procede de la siguiente manera: #Sin cerrar la ventana del gráfico anterior, ejecuta la siguiente instrucción

identify(Usuarios, Tiempo, n=1) # n=1 indica que solamente será un punto seleccionado

# Y luego selecciona un punto en el gráfico haciendo clic con el ratón. Esto es útil para identificar puntos que podrían ser atípicos.

# Deberá aparecer en la R-Console el índice que corresponde a este punto.

12º) Aplica la función lm() para encontrar el modelo lineal que se ajusta a los datos.

reg.Y.X <- lm(Tiempo ~ -1 + Usuarios, Sistema, na.action=NULL, method="qr", model=TRUE) #-1 indica que no se toma en cuenta la constante en el modelo.

summary(reg.Y.X)

# Note que es necesaria la instrucción anterior para poder visualizar los resultados más sobresalientes de la regresión encontrada. Nos muestra la estimación de los parámetros junto con su significancia, el coeficiente de determinación.

13º) Agrega la recta de regresión al gráfico de dispersión. abline (reg.Y.X)

# Observación: Alternativamente si quiere una recta más "exacta" use:

lines(Usuarios, 0.079437\*Usuarios)

14º) Efectúa una análisis de variabilidad del modelo o descomposición de la varianza. reg.anova <- anova(reg.Y.X); reg.anova