Graduado en Informática

Prácticas de Estadística

Práctica 1: Estadística Descriptiva 1 variable.

Ejercicio 1:

El fichero 'datospr1.mat' contiene los valores medidos para la variable estadística X. Queremos detectar aquellos valores anómalos ('outliers') de X. Una de las técnicas empleadas es calcular Q_1, Me y Q_3 y considerar anómalos aquellos que sean menores de la mediana en $3(Me - Q_1)$, o mayores que esta en $3(Q_3 - Me)$.

1. *Encontrar los valores anómalos de X según la regla antes indicada.*
2. *Crear una nueva variable Y donde no se encuentren los valores anómalos.*
3. *Calcular media y varianza de los datos X y de los Y.*

Solución:

Debemos cargar la variable unidimensional X en Matlab (Scilab o similar), para ello debemos copiar el fichero en el directorio de trabajo de Matlab y ejecutar:

clear all, format long % Limpia de variables preexistentes y muestra todas las cifras.
load datospr1 % Carga los datos desde el fichero.
X % Comprobamos que ya existe la variable.
N=length(X) % Vemos su dimension.

Para calcular Q_1, Me y Q_3 hacemos:

XX=sort(X) % Ordenamos la variable, creando la variable ordenada XX.
n1=N*1/4, n2=N*1/2, n3=N*3/4 % Calculamos los lugares que deben ocupar los 3 cuartiles.
nn1=floor(n1), nn2=floor(n2), nn3=floor(n3) % Calculamos su parte entera.
% Si es entero saco la media, si no lo es tomo el siguiente.
% a==floor(a), es 'cierto' si a es entero y 'falso' si no lo es.
if n1==nn1, Q1=(XX(nn1)+XX(nn1+1))/2, else Q1=XX(nn1+1), end
if n2==nn2, Me=(XX(nn2)+XX(nn2+1))/2, else Me=XX(nn2+1), end
if n3==nn3, Q3=(XX(nn3)+XX(nn3+1))/2, else Q3=XX(nn3+1), end

Hallamos los límites:

Lmin=Me-3*(Me-Q1) % Limite inferior
Lmax=Me+3*(Q3-Me) % Limite superior

Hallamos los elementos fuera de [Lmin, Lmax]. Usamos los símbolos '|' y '&' para el 'or' y 'and' lógicos en Matlab.

La = **find**((**X** > **Lmax**)|(**X** < **Lmin**)) **% Lista elementos extraños**
L = **find**((**X** <= **Lmax**)&(**X** >= **Lmin**)) **% Lista elementos no extraños**
Y=X(L) % En Y están los no extraños

Calculamos media y varianza de X:

Sx=sum(X), mediaX=Sx/N
Sx2=sum(X.^2), m2X=Sx2/N
VarX=m2X-mediaX^2

Calculamos media y varianza de Y:

Ny=length(Y)
Sy=sum(Y), mediaY=Sy/Ny
S2Y=sum(Y.^2), m2Y=S2Y/Ny
VarY=m2Y-mediaY^2

Ejercicio 2:

El fichero 'TempMalaga.mat' obtenido a partir del suministrado por la Agencia Estatal de Meteorología (AEMET), muestra las temperaturas máximas, mínimas y media obtenidas diariamente en Málaga desde el año 2000, cada fila (registro del fichero consta de 6 columnas que indican:

Columnas 1, 2 y 3: Año, Mes y Día, respectivamente

Columnas 4, 5 y 6: Temperatura máxima, mínima y media en grados centígrados, respectivamente.

Calcular:

1. *La tabla de frecuencias absolutas de las temperaturas mínimas del mes de Febrero con las clases: 'Menor o igual a 3 grados', (3,6], (6,8], (8,10], (10,13], 'Más de 13 grados'.*

2. *A partir de la tabla, hallar:*

   (a) Q_1, Me, Q_3 *y Moda.*

   (b) *Media, varianza, sesgo y curtosis.*

   (c) *Desviación media y la media cuadrática de las desviaciones.*

3. *Repetir los cálculos del apartado 2 (excepto la moda) con los datos iniciales (sin agrupar).*

Solución:

Apartado 1: Cargamos el fichero en Matlab mediante:

clear all % Limpia de variables preexistentes.
load TempMalaga % Carga los datos desde el fichero.
whos % Muestra los nombres de las variables existentes.

Observamos que solo existe una variable llamada 'TMA' con 4383 filas y 6 columnas a la que le vamos a filtrar los elementos correspondientes al mes de febrero (segunda columna con un '2':

L=find(TMA(:,2)==2) % Nos da las filas (posiciones) de los registros de febrero.
X=TMA(L,5) % Crea la variable X con las temperaturas mínimas de los registros de febrero.
N=length(X) % Número de elementos tras el filtro.

Calculamos las frecuencias absolutas de cada clase:

```
n=zeros(1,6); % Pues hay 6 modalidades
for k=1:N,
   if X(k)<= 3, n(1)=n(1)+1;
   elseif X(k)<= 6, n(2)=n(2)+1;
   elseif X(k)<= 8, n(3)=n(3)+1;
   elseif X(k)<= 10, n(4)=n(4)+1;
   elseif X(k)<= 13, n(5)=n(5)+1;
   else n(6)=n(6)+1;
   end
end, n % Mostramos al final las frecuencias absolutas.
```

Apartado 2-a: Tomo las marcas de clase 'x' y las alturas del histograma 'h':

```
Li=[0,3,6,8,10,13,16], x=[1.5 4.5 7 9 11.5 14.5]
a=diff(Li), h=n./a % Amplitud y altura.
```

Calculamos la moda:

```
[val,ind]=max(h) % ind nos indica el intervalo modal.
h1=h(ind)-h(ind-1), h2=h(ind)-h(ind+1) % NOTA: val=h(ind)
Mo=Li(ind)+a(ind)*h1/(h1+h2)
```

NOTA: El método no funcionaría si ind=1 ó ind=6(última clase) pues no existe h(0) ni h(7).

Calculamos Q_1, Mediana y Q_3:

```
n1=N/4, n2=N/2, n3=3*N/4
nac=cumsum(n) % frecuencias acumuladas
L=find(n1<nac),ind=L(1);
Q1=Li(ind)+a(ind)*(n1-nac(ind-1))/n(ind)
L=find(n2<nac),ind=L(1);
Me=Li(ind)+a(ind)*(n2-nac(ind-1))/n(ind)
L=find(n3<nac),ind=L(1);
Q3=Li(ind)+a(ind)*(n3-nac(ind-1))/n(ind)
```

Apartado 2-b: Para hallar la media, varianza, sesgo y curtosis:

```
m1=sum(n.*x)/N, m2=sum(n.*x.^2)/N, m3=sum(n.*x.^3)/N, m4=sum(n.*x.^4)/N
mu3=m3-3*m2*m1+2*m1^3, mu4=m4-4*m3*m1+6*m2*m1^2-3*m1^4
```

media=m1, var=m2-m1^2, s=sqrt(var), sesgo=mu3/s^3, curtosis=mu4/s^4-3

Apartado 2-c: Para hallar la desviación media y la media cuadrática de las desviaciones:
d=x-m1, DM=sum(n.*abs(d))/N, MCdesv=sqrt(sum(n.*d.^2)/N)

Apartado 3: Repetimos para los datos sin agrupar:
Calculamos Q_1, Mediana y Q_3:
nn1=floor(n1), nn2=floor(n2), nn3=floor(n3) % n1, n2, n3 fueron calculados antes.
XX=sort(X); % Ordenamos los datos
if n1==nn1, Q1X=(XX(nn1)+XX(nn1+1))/2, else Q1X=XX(nn1+1),end
if n2==nn2, MeX=(XX(nn2)+XX(nn2+1))/2, else MeX=XX(nn2+1),end
if n3==nn3, Q3X=(XX(nn3)+XX(nn3+1))/2, else Q3X=XX(nn3+1),end
Para hallar la media, varianza, sesgo y curtosis:
m1X=sum(X)/N, m2X=sum(X.^2)/N, m3X=sum(X.^3)/N, m4X=sum(X.^4)/N
mu3X=m3X-3*m2X*m1X+2*m1X^3
mu4X=m4X-4*m3X*m1X+6*m2X*m1X^2-3*m1X^4
mediaX=m1X, varX=m2X-m1X^2
sX=sqrt(varX), sesgo=mu3X/sX^3, curtosis=mu4X/sX^4-3
Para hallar la desviación media y la media cuadrática de las desviaciones:
dX=X-m1X, DMX=sum(abs(dX))/N, MCdesvX=sqrt(sum(dX.^2)/N)

Alternativa al apartado 3:
Matlab cuenta con las funciones estadísticas: **mean, median, std, min, max, moment, skewness, kurtosis, mode y cov** que nos permiten simplificar los cálculos:
MeX=median(X) % Sin necesidad de ordenar.
m1X=moment(X,1), m2X=moment(X,2), m3X=moment(X,4), m4X=moment(X,4)
mediaX=mean(X), varX=var(X), sesgoX=skewness(X), curtosisX=kurtosis(X)-3

Prácticas de Estadística

Práctica 2: Estadística Descriptiva 2 variables.

Ejercicio 3:

El fichero 'Hamb.m' contiene las medidas obtenidas en un estudio sobre hamburguesas. Cada análisis (registro) tiene 3 campos:

- Tipo: Con los valores { Ternera, Resto de carnes, Ave} codificadas respectivamente como 1, 2 y 3.
- Calorías de la hamburguesa.
- Sodio: Miligramos de sodio en la hamburguesa.

Hallar:

1. Las calorías medias y varianzas para cada tipo de hamburguesa.
2. Lo mismo pero para el contenido en sodio.
3. Representar los resultados medios gráficamente.
4. Ajustar la recta de regresión de Y=calorías respecto a X=sodio.
5. Hallar varianza residual y coeficiente de correlación lineal del ajuste Y/X.
6. Ajustar la recta de regresión de X=sodio respecto a Y=calorías.
7. Hallar varianza residual y coeficiente de correlación lineal del ajuste X/Y.
8. Representar conjuntamente la nube de puntos (con 'x') y ambas rectas.
9. Representar conjuntamente la nube de puntos (con 'x') y las rectas, pero ahora los puntos correspondientes a hamburguesas de ternera en rojo (r), las de carne en general en azul (b) y las de ave en magenta (m)'.
10. Ajustar un polinomio de tercer grado Y/X. Hallar la varianza residual, suma de los cuadrados de los errores y coeficiente de determinación.
11. Representar conjuntamente los puntos (con 'x'), la recta de Y/X y el polinomio anterior.

SOLUCIÓN:
El programa MATLAB puede quedar como:

```
clear all, format compact,clf          %  1 Borro variables y graficas.
Hamb,whos                              %  2 Cargo los datos y miro que variables hay
CC                                     %  3 Visualizo la unica variable existente
Y=CC(:,2);                             %  4 La segunda columna es la Y (sodio)
X=CC(:,3);N=length(X)                  %  5 La tercera es la X y N (numero de registros).
L1=find(CC(:,1)==1);N1=length(L1)      %  6 Lugares clase 1 y cuantos hay
L2=find(CC(:,1)==2);N2=length(L2)      %  7 Lo mismo para clase 2
L3=find(CC(:,1)==3);N3=length(L3)      %  8 Lo mismo para clase 3
Y1=Y(L1),Y2=Y(L2),Y3=Y(L3)             %  9 Variable Y para cada clase
X1=X(L1),X2=X(L2),X3=X(L3)             % 10 Variable X para cada clase

mY1=sum(Y1)/N1,%mY1b=mean(Y1)          % 11 media de 1 (ternera) de dos formas. (sodio)
mY2=sum(Y2)/N2,%mY2b=mean(Y3)          % 12 media de 2 (otra carne).
mY3=sum(Y3)/N3,%mY3b=mean(Y3)          % 13 media de 3 (ave).
mX1=sum(X1)/N1,%mX1b=mean(X1)          % 14 media de 1 (ternera) de dos formas. (calorias)
mX2=sum(X2)/N2,%mX2b=mean(X2)          % 15 media de 2 (otra carne) (calorias)
mX3=sum(X3)/N3,%mX3b=mean(X3)          % 16 media de 3 (ave) (calorias)
vY1=sum(Y1.^2)/N1-mY1^2,%vY1b=var(Y1)*(N1-1)/N1 % 17 VARIANZAS sodio de 2 formas
vY2=sum(Y2.^2)/N2-mY2^2,%vY2b=var(Y2)*(N2-1)/N2 % 18    "
vY3=sum(Y3.^2)/N3-mY3^2,%vY3b=var(Y3)*(N3-1)/N3 % 19    "
vX1=sum(X1.^2)/N1-mX1^2,%vX1b=var(X1)*(N1-1)/N1 % 20    "   ahora para calorias
vX2=sum(X2.^2)/N2-mX2^2,%vX2b=var(X2)*(N2-1)/N2 % 21    "
```

```
vX3=sum(X3.^2)/N3-mX3^2,%vX3b=var(X3)*(N3-1)/N3 % 22    "

disp('Apartado 3')                                % 23 Saca el mensaje
line([0.6,3.4],[0,0])                             % 24 Dibuja una recta entre (0.6,0) y (3.4,0)
line([1,1],[0,mY1],'Linewidth',8,'Color','r')     % 25 Linea en rojo y ancho 8 de (1,0) a (1,mY1)
line([2,2],[0,mY2],'Linewidth',8,'Color','g')     % 26 Linea en verde, ancho 8 de (2,0) a (2,mY2)
line([3,3],[0,mY3],'Linewidth',8,'Color','b')     % 27 Linea en azul y ancho 8 de (3,0) a (3,mY3)
pause,clf                                         % 28 Pausa para ver el grafico y lo borra
line([0.6,3.4],[0,0])                             % 29 Lo mismo para el sodio
line([1,1],[0,mX1],'Linewidth',8,'Color','r')     % 30
line([2,2],[0,mX2],'Linewidth',8,'Color','g')     % 31
line([3,3],[0,mX3],'Linewidth',8,'Color','b')     % 32
pause,clf                                         % 33 Para salir del pause pulsar tecla

disp('Apartados 4 y 5')                           % 34
p=polyfit(X,Y,1)                                  % 35 Ajuste lineal por la via rapida
a=p(2), b=p(1)                                    % 36 Coeficientes de la recta Y/X
mX=mean(X),mY=mean(Y),                            % 37 Medias
vX=sum(X.^2)/N-mX^2,vY=sum(Y.^2)/N-mY^2           % 38 Varianzas
covar=sum(X.*Y)/N-mX*mY,                          % 39 Covarianza
r=covar/sqrt(vX*vY)                               % 40 Coeficiente correlación lineal
Vr=(1-r^2)*vY                                     % 41 Varianza residual caso lineal

disp('Apartados 6 y 7')                           % 42
p1=polyfit(Y,X,1),b1=p1(1),a1=p1(2)               % 43 Ajuste de la recta X/Y: X=a1+b1*Y
Vr1=(1-r^2)*vX                                    % 44 Varianza residual ajuste X/Y

disp('Apartado 8')                                % 45
xx=100:700;yy1=polyval(p,xx);                     % 46 Auxiliar para dibujar recta Y/X
yy=80:200;xx1=polyval(p1,yy);                     % 47    "       "     "      "   X/Y
plot(X,Y,'x',xx,yy1,xx1,yy),grid                  % 48 Dibuja todo. Grid pone la cuadrícula
pause,clf                                         % 49

disp('Apartado 9')                                % 50
plot(X1,Y1,'xr',X2,Y2,'xb',X3,Y3,'xm',xx,yy1,xx1,yy),grid % 51 Dibuja lo pedido
pause,clf                                         % 52

disp('Apartado 10')                               % 53
pp=polyfit(X,Y,3)                                 % 54 Ajusta el polinomio y=a+bx+cx^2+dx^3
ye=polyval(pp,X)                                  % 55 Calcula los y estimados
re=Y-ye                                           % 56 Calcula vector residuo o error.
Vre=var(re)*(N-1)/N                               % 57 Varianza residual.
SSE=sum(re.^2)                                    % 58 Suma de errores al cuadrado
R2=1-Vre/vY                                       % 59 Coeficiente de determinación

disp('Apartado 11')                               % 60
yy2=polyval(pp,xx);                               % 61 Auxiliar para dibujar polinomio
plot(X,Y,'X',xx,yy1,xx,yy2),grid                  % 62 Dibuja todo conjuntamente
```

Resultados obtenidos: Comentaremos los principales: Número de elementos totales N =54, del tipo 1 (ternera) hay N1=20, del tipo2 hay N2=17 y de ave hay N3=17.

Aptdo. 1: Las calorías medias para 'ternera' es $\mathbf{mY1 = 156.85}$ (significado 1.5685e+002), para 'otra carne' $\mathbf{mY2 = 158.7059}$ y para 'ave' vale $\mathbf{mY3 = 122.4706}$.

Las varianza para 'ternera' es $vY1 = 487.0275$, para 'otra carne' $vY2 = 599.3841$ y para 'ave' de $vY3 = 611.1903$

Aptdo. 2: El contenido medio de sodio es para 'ternera' de $\mathbf{mX1 = 401.15}$, para 'carne general' de $\mathbf{mX2 = 418.5294}$ y para 'ave' de $\mathbf{mX3 = 459}$.

Las varianzas son respectivamente $\mathbf{vX1 = 9968.2275}$, $\mathbf{vX2 = 8293.6609}$ y $\mathbf{vX3 = 6758.3529}$.

Aptdo. 4: La recta obtenida es $\mathbf{Y = 0.1565X + 80.1073}$.

Aptdo. 5 La media de X (sodio) es $\mathbf{mX = 424.8333}$, la de Y (calorías) es $\mathbf{mY = 146.6111}$, la varianza de X vale $\mathbf{vX = 9018.2870}$ y la de Y $\mathbf{vY = 8298.3025}$, la covarianza es $\mathbf{cov = 1411.7315}$ y por último el coeficiente de correlación lineal $\mathbf{r = 0.5161}$ y la varianza residual $\mathbf{Vr = 608.8364}$.

Aptdo. 6: La recta obtenida es: $\mathbf{X = 1.7012Y + 175.4142}$. La varianza residual del ajuste X/Y es $Vr1 = 6616.6082$.

Notese que el coeficiente de correlación lineal es el mismo en el ajuste Y/X y X/Y, pero la varianza residual vale $(1 - r^2)V_Y$ en un caso y $(1 - r^2)V_X$ en el otro.

Aptdo. 10: El polinomio obtenido es (obtenido en pp con 'format long'):

$$\mathbf{Y = -0.0000011327908X^3 + 0.0013465122356X^2 - 0.3429711494881X + 136.9271160836836}$$

La varianza residual es: $\mathbf{Vre = 603.3198}$, la suma del cuadrado de los errores $\mathbf{SSE = 32579.2684}$ y el coeficiente de determinación vale $\mathbf{R2 = 0.2730}$.

Ejercicio 4:

El fichero 'car.m' contiene características de diversos modelos de coche. una de las variables contiene la marca, mientras la otra contiene una matriz con 5 columnas donde:

- Columna 1: **VOL:** Volumen interior en pies cúbicos. (ft^3)
- Columna 2: **HP:** Potencia del motor en CV. (Hp)
- Columna 3: **MPG:** Consumo medio en millas por galón. (mpg)
- Columna 4: **VM:** Velocidad máxima en millas por hora. (mph)
- Columna 5: **WT:** Peso del vehículo en libras dividido por 100. (100 lb)

Hallar:

1. Ajustar una recta que nos de el consumo en función del peso. Hallar r y Vr.
2. Ajustar un hiperplano que nos de el consumo en función de las restantes. Hallar el coeficiente de determinación.
3. ¿Existe alguna variable poco significativa? Es decir, que variables influyen mucho en el consumo y cuales no.
4. ¿Qué modelo presenta el consumo más bajo para sus características? ¿Cuál lo tiene más alto?
5. Imputación de datos: Supongamos que del modelo de coche XXXX conocemos los valores de $VOL = 70$, $HP = 84$, $VM = 100$ y $WT = 30$, pero el dato de MPG es desconocido (missing) y necesitamos usar un valor para él. Una primera aproximación es darle el de una medida de tendencia central de los valores MPG conocidos. Otra posibilidad es imputar su valor a partir del ajuste.
   (a) Imputar el valor medio.
   (b) Imputar el valor mediano.
   (c) Imputar el valor ajustado por el hiperplano.

Solución:

El programa MATLAB puede ser:

```
clear all,clf,clc,format compact                                  %  1 Preparacion
car                                                               %  2 Carga de variables
VOL=CC(:,1);HP=CC(:,2);MPG=CC(:,3);VM=CC(:,4);WT=CC(:,5);         %  3 Cambio nombres a variables
p=polyfit(WT,MPG,1)                                               %  4 Ajustamos recta MPG=a+b(WT)
N=length(VOL)                                                     %  5 Numero de registros
mWT=mean(WT),vWT=var(WT)*(N-1)/N                                  %  6 Media y varianza de WT
mMPG=mean(MPG),vMPG=var(MPG)*(N-1)/N                              %  7 Media y varianza de MPG
cov1=sum(MPG.*WT)/N-mMPG*mWT                                      %  8 Covarianza entre MPG y WT
%cov1b=cov(MPG,WT)*(N-1)/N                                        %  9 Otra forma de hallar covar.
r=cov1/sqrt(vMPG*vWT) % rb=corrcoef(WT,MPG)                       % 10 Coef. correlacion (2 formas)
Vr=(1-r^2)*vMPG                                                   % 11 Varianza residual caso recta
SSE=Vr*N                                                          % 12 Suma de errores al cuadrado
CD_lineal=r^2                                                     % 13 Coef. determinacion caso lineal
disp('Aptdo 2:')                                                  % 14
A=[N sum(VOL) sum(HP) sum(VM) sum(WT);                            % 15 Matriz para ajustar:
   sum(VOL) sum(VOL.^2) sum(HP.*VOL) sum(VM.*VOL) sum(WT.*VOL); % 16 VMG=A+B*VOL+C*HP+D*VM+E*WT
   sum(HP) sum(VOL.*HP) sum(HP.^2) sum(VM.*HP) sum(WT.*HP);   % 17 Ec1: <E,1>=0, Ec2: <E,VOL>=0,
   sum(VM) sum(VOL.*VM) sum(HP.*VM) sum(VM.^2) sum(WT.*VM);   % 18 Ec3: <E,HP>=0, Ec4: <E,VM>=0,
   sum(WT) sum(VOL.*WT) sum(HP.*WT) sum(VM.*WT) sum(WT.^2)]   % 19 Ec5: <E,WT>=0
B=[sum(MPG);sum(MPG.*VOL);sum(MPG.*HP);sum(MPG.*VM);sum(MPG.*WT)] % 20 Vector para el ajuste
sol=A\B                                                           % 21 Resolvemos el sistema lineal
MPGe=sol(1)+sol(2)*VOL+sol(3)*HP+sol(4)*VM+sol(5)*WT              % 22 MPG estimado por el ajuste
res=MPG-MPGe                                                      % 23 Vector residuo o error
Vres=var(res)*(N-1)/N                                             % 24 Varianza residual del ajuste
R2=1-Vres/vMPG                                                    % 25 Coeficiente de determinación
disp('Apartado 3:')                                               % 26
CORR=corrcoef([MPG,VOL,HP,VM,WT])                                 % 27 Matriz de coef. de corr. lineal
```

```
disp('Apartado 4:')                                                  % 28
[MAX,modmax]=max(res),modelMAX=MODEL(modmax,:)                       % 29 Modelo con maximo residuo pos.
[MIN,modmin]=min(res),modelMIN=MODEL(modmin,:)                       % 30 Modelo con máximo residuo neg.
disp('Apartado 5')                                                    % 31
ImpMedia=mMPG                                                         % 32 La media ya estaba calculada
ImpMediana=median(MPG)                                                % 33 Imputado por la mediana
ImpHiperpl=sol(1)+sol(2)*70+sol(3)*84+sol(4)*100+sol(5)*30 % 34 Imputado por el hiperplano
```

RESULTADOS:

Aptdo. 1: La recta es: $\mathbf{MPG} = -\mathbf{1.1122(WT)} + \mathbf{68.1655}$.

Hay $N = 82$ registros, la media de $mWT = 30.9146$ y la varianza es $vWT = 65.4744$. La media de MPG es $mMPG = 33.7817$ y la varianza es $vMPG = 98.8715$.

La covarianza es $cov1 = -72.8217$, elcoeficiente de correlación lineal vale $r = -0.9051$ y la varianza residual $Vr = 17.8781$.

La suma de los errores al cuadrado vale $SSE = 1466.0016$ y el coeficiente de determinación del caso lineal es $CD_l ineal = 0.8192$

Aptdo 2: Vamos a ajustar un hiperplano del tipo $MPG = A + B * VOL + C * HP + D * VM + E * WT$, es decir, hallar el vector generado por $B = \{\vec{1}, V\vec{O}L, H\vec{P}, V\vec{M}, W\vec{T}\}$ más próximo a $M\vec{P}G$. Plantando las condiciones: $< \vec{e}, \vec{1} >= 0$, $< \vec{e}, V\vec{O}L >= 0$, $< \vec{e}, H\vec{P} >= 0$, $< \vec{e}, V\vec{M} >= 0$, $< \vec{e}, W\vec{T} >= 0$, obtenemos el sistema a resolver:

$$\begin{pmatrix} N & \sum_i VOL_i & \sum_i HP_i & \sum_i VM_i & \sum_i WT_i \\ \sum_i VOL_i & \sum_i VOL_i^2 & \sum_i HP_i * VOL_i & \sum_i VM_i * VOL_i & \sum_i WT_i * VOL_i \\ \sum_i HP_i & \sum_i VOL_i * HP_i & \sum_i HP_i^2 & \sum_i VM_i * HP_i & \sum_i WT_i * HP_i \\ \sum_i VM_i & \sum_i VOL_i * VM_i & \sum_i HP_i * VM_i & \sum_i VM_i^2 & \sum_i WT_i * VM_i \\ \sum_i WT_i & \sum_i VOL_i * WT_i & \sum_i {*HP_i WT_i} & \sum_i VM_i * WT_i & \sum_i WT_i^2 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} =$$

$$= \begin{pmatrix} \sum_i MPG_i \\ \sum_i MPG_i * VOL_i \\ \sum_i MPG_i * HP_i \\ \sum_i MPG_i * VM_i \\ \sum_i MPG_i * WT_i \end{pmatrix}$$

que al resolverlo nos proporciona el modelo:

$$VMG = 192.4378 - 0.01564 * VOL + 0.3922 * HP - 1.2948 * VM - 1.8598 * WT$$

La varianza residual vale $Vres = 12.5290$ y la razón de determinación: $R2 = 0.8733$

Aptdo. 3: Nos resulta la matriz de correlaciones lineales:

```
CORR =
    1.0000   -0.3686   -0.7899   -0.6884   -0.9051
   -0.3686    1.0000    0.0765   -0.0431    0.3850
   -0.7899    0.0765    1.0000    0.9665    0.8322
   -0.6884   -0.0431    0.9665    1.0000    0.6785
   -0.9051    0.3850    0.8322    0.6785    1.0000
```

que podemos interpretar como que la variable que más afecta a la primera variable (MPG) es la que tiene el módulo mayor en la primera fila, es decir la 5ª variable (WT) y la que menos la 2ª , es decir (VOL).

Además todas tienen correlación negativa, indicando que si aumentan, disminuye MPG (millas recorridas por galón de combustible).

Es destacable que el modelo lineal con solo la variable WT explica el 81.92% de la varianza de MPG, mientras que el hiperplano (mucho más complejo) explica el 87.33% de la varianza. Evidentemente el hiperplano ajusta mejor, pero un buen modelo de ajuste debe ser además lo más simple posible, no quedando en este caso claro si compensa usar 4 variables, en lugar de 1, para explicar un 6.41% más de las variaciones observadas en MPG.

Aptdo 4: Podemos buscar que modelo se aparta más del hiperplano de regresión de forma positiva, es decir, tiene un valor MPG superior al estimado por la regresión: $MAX = 11.9854$ (lo que se aparta), $modmax = 1$ (registro que lo hace) y lo hace el modelo $modelMAX = GM/GeoMetroXF1$.

También podemos conseguir cual se aparta más del valor estimado por la regresión pero en sentido negativo: $MIN = -9.0108$, $modmin = 29$ y lo hace el $modelMIN = SubaruLoyale$.

Aptdo. 5: Los valores imputados son:

ImpMedia = 33.7817, **ImpMediana = 32.45**, **ImpHiperpl = 39.0125**.

Prácticas de Estadística

Práctica 3: Variables aleatorias. Simulación.

Introducción:

La simulación de un sistema real se usa para prever resultados, comparar estrategias,... a un costo muy inferior a hacerlo realmente. Tradicionalmente se han desarrollado prototipos a escala, ensayos en condiciones controladas y seguras, etc. Actualmente, la simulación mediante ordenador resulta generalmente la más barata y es la más extensamente aplicada. (Leer 'Simulación' en Wikipedia).

Herramientas para simulación:

MATLAB básico dispone de las funciones **rand(m,n)** y **randn(m,n)** que devuelven una matriz de m filas y n columnas de números aleatorios siguiendo una Uniforme [0,1) y una N(0,1), respectivamente. Con ellas pueden simularse otras muchas, por ejemplo:

Uniforme [a,b]: Mediante **x=a+(b-a)*rand(m,n)**

Uniforme discreta en $\{Nmin, Nmin+1, \ldots, Nmax\}$**:** Mediante **x=Nmin+floor(Nmax*rand(m,n))** (las versiones nuevas de MATLAB disponen de **randi([Nmin,Nmax],m,n)** para este fin).

Normal(a,b): Mediante **x=a+b*randn(m,n)**

Además de estas, la toolbox "stats", simula fácilmente muchas otras funciones útiles:

Distribución	Función de densidad (pdf)	Función de Distribución (cdf)	Función cuantil (inv)	Números aleatorios (rnd)
Beta	betapdf	betacdf	betainv	betarnd
Binomial	binopdf	binocdf	binoinv	binornd
Binomial negativa	nbinpdf	nbincdf	nbininv	nbinrnd
χ^2	chi2pdf	chi2cdf	chi2inv	chi2rnd
Exponencial	exppdf	expcdf	expinv	exprnd
F Fisher-Sn.	fpdf	fcdf	finv	frnd
Gamma	gampdf	gamcdf	gaminv	gamrnd
Geométrica	geopdf	geocdf	geoinv	geornd
Hipergeométrica	hygepdf	hygecdf	hygeinv	hygernd
Log-normal	lognpdf	logncdf	logninv	lognrnd
Multinomial	mnpdf	mncdf	mninv	mnrnd
Normal multivariante	mvnpdf	mvncdf	-	mnrnd
Normal	normpdf	normcdf	norminv	normrnd
Poisson	poisspdf	poisscdf	poissinv	poissrnd
Rayleigh	raylpdf	raylcdf	raylinv	raylrnd
T-Student	tpdf	tcdf	tinv	trnd
Uniforme discreta	unidpdf	unidcdf	unidinv	unidrnd
Uniforme cont.	unifpdf	unifcdf	unifinv	unifrnd
Weibull	wblpdf	wblcdf	wblinv	wblrnd

Ejercicio 5:

Hallar una matriz de tamaño 5×10, con números aleatorios siguiendo la distribución especificada:

1. a) $\xi \rightsquigarrow N(4,7)$, b) $\eta \rightsquigarrow N(-3,2)$, c) $\zeta \rightsquigarrow t_{15}$ (T-Student).
2. a) Uniforme discreta en $\{1,2,\ldots,6\}$, b) Uniforme continua en [-2,2].
3. a) Binomial(100,0.4), b) Binomial(500,0.003).
4. a) Chi-Cuadrado con 20 g.d.l. b) Poisson con $\lambda = 9$, c) Rayleigh R(4).

Prácticas de Estadística

Práctica 4: Método de Montecarlo.

Introducción:

Consiste en estimar la probabilidad de que ocurra un suceso mediante la repetición del experimento aleatorio un gran número de veces. Se basa en las **"Leyes de los grandes números"** aplicada al experimento de Bernouilli.

Leyes de los grandes números

Si realizamos N experimentos aleatorios de forma independiente de una misma distribución ξ, entonces la media de los resultados obtenidos $\bar{x} = \frac{\sum_i^N x_i}{N}$ converge a la media de ξ. Esto es: $\lim_{\mathbf{N}\to\infty} \bar{\mathbf{x}} = \mu = \mathbf{E}(\xi)$

Así, la media de las N realizaciones de un experimento de Bernouilli ($\hat{p} = \frac{N_a}{N}$, N_a=Número de veces en que ocurre A), converge a la media de la Bernouilli, es decir p. Además se sabe que para N grande ($N > 30$), resulta: $\hat{p} \rightsquigarrow N\left(p, \sqrt{\frac{pq}{N}}\right)$, lo que nos lleva a:

$$P\left(\hat{p} - z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}\hat{q}}{N}} \leq p \leq \hat{p} + z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}\hat{q}}{N}}\right) = \alpha \quad \Rightarrow \quad I_p = \left[\hat{p} - z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}\hat{q}}{N}},\ \hat{p} + z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}\hat{q}}{N}}\right]$$

Otra convergencia importante es que la media de N variables aleatorias iguales e independientes de media μ y desviación típica σ, cuando N es grande, converge a la media de la población según una normal: $\bar{x} \rightsquigarrow N\left(\mu, \frac{\sigma}{\sqrt{N}}\right)$, por tanto:

$$P\left(\bar{x} - z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{N}} \leq \mu \leq \bar{x} + z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{N}}\right) = \alpha \quad \Rightarrow \quad I_\mu = \left[\bar{x} - z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{N}},\ \bar{x} + z_{\frac{\alpha}{2}}\frac{\sigma}{\sqrt{N}}\right]$$

Ejercicio 6:

Supongamos la región rectangular R delimitada por los puntos $(0,0), (20,0), (20,10)$ y $(0,10)$, donde las unidades expresan Km. Tenemos antenas de TV situadas en los puntos $(-1,5)$, $(5,4)$, $(15,7)$, $(10,1)$, $(12,2)$, $(19,1)$ y $(15,2)$, que pueden recoger/enviar la señal a algún punto de R y que tienen un alcance de 3Km.

Si todos los puntos de R tienen la misma probabilidad de emitir una señal, hallar mediante simulación con 10.000 ejecuciones:

1. La proporción de señales emitidas desde R que serán recogidas por alguna antena y el área de la misma.
2. La proporción de señales emitidas desde R que serán recogidas por varias antenas.
3. Si queremos seleccionar el lugar idóneo para una nueva antena entre $A = (1,1)$, $B = (10,6)$ y $C = (14,4)$. En el sentido que incrementa más la región con cobertura, ¿Cuál es mejor?

Solución:

a: Podemos programarlo realizando las experiencias de forma sucesiva y empleando contadores de veces en que ocurre el experimento, o realizando las 10.000 ejecuciones a la vez (más rápido, pero ocupa más memoria).

De todas formas, vamos a hacerlo de forma simultanea:

a1) Primero debemos generar 10.000 veces una posición aleatoria en $[0,20] \times [0,10]$:

Lo hacemos con **N=10000; x=[20*rand(N,1) 10*rand(N,1)];**

Tendremos una matriz 10000×2 donde cada fila es un punto aleatorio en R.

a2) Estudiamos si el punto (fila de x) está en la cobertura de cada una de las antenas:

Para antena en [-1,5]: **xx=[x(:,1)+1 x(:,2)-5];d1=sqrt(xx(:,1).^2+xx(:,2).^2);c1=(d1<=3);**

Para antena en [5,4]: **xx=[x(:,1)-5 x(:,2)-4];d2=sqrt(xx(:,1).^2+xx(:,2).^2);c2=(d2<=3);**

...

Para antena en [15,2]: **xx=[x(:,1)-15 x(:,2)-2];d7=sqrt(xx(:,1).^2+xx(:,2).^2);c7=(d7<=3);**

a3) Calculamos si tiene alguna cobertura, lo que podemos hacer mediante $c = c1|c2|c3|c4|c5|c6|c7$ pero teniendo en cuenta el apartado b, resulta mejor:

c=c1+c2+c3+c4+c5+c6+c7; que nos dice el número de antenas en su entorno.

a4) Calculamos el número de puntos con cobertura: **Na=sum(**$c > 0$**)**

a5) Estimamos la proporción y el área con cobertura: **p=Na/N, S=p*20*10, q=1-p**

a6) Intervalo de confianza para p y S al 95% ($z_{\frac{\alpha}{2}} = 1.96$ **z=norminv(0.975,0,1)**):

Ip=[p-1.96*sqrt(p*q/N), p+1.96*sqrt(p*q/N)], Is=Ip*200

b) Ya está hecho todo el trabajo hasta calcular Na, falta:

Nb=sum($c > 1$)

b1) Estimamos la proporción y el área con cobertura: **pb=Nb/N, S=pb*20*10,qb=1-pb**

b2) Intervalo de confianza para pb y Sb al 95%:

Ipb=[pb-1.96*sqrt(pb*qb/N), pb+1.96*sqrt(pb*qb/N)], Isb=Ipb*200

c) Resulta repetitivo, pues basta con añadir una antena en el apartado a) y ver cuál da un $\hat{p}$ mayor, pero sirve para ver como la simulación permite evaluar estrategias.

Ejercicio 7:

Repetir el ejercicio anterior, pero ahora se va a tener en cuenta la existencia de un núcleo de población en el punto (12,6) y una carretera, por lo que los puntos aleatorios $(x, y) \in R$ deberán seleccionarse considerando este hecho desde una normal bivariante de media (12,6) y matriz de covarianzas $V = \begin{pmatrix} 3 & 0.5 \\ 0.5 & 2 \end{pmatrix}$. Se considera una población de 25000 personas. Hallar la proporción con cobertura y el número de las mismas.

Nota: Ahora un punto (x,y) aleatorio, podrá salirse de R y debemos filtrar los válidos.

Solución

a1) Mediante **help stats** y **help mvnrnd** vemos que podemos generar los 10000 vectores aleatorios con:

N=10000; x=mvnrnd([12 6],[3 0.5;0.5 2],N);

a1-bis) Tenemos que ver si el punto es de R:

R=(x(:,1)<=20).*(x(:,1)>=0).*(x(:,2)>=0).*(x(:,2)<=10);

Nr=sum(R>0);

los puntos a2) y a3) serían iguales, pero a4) y siguientes quedarían:

Na=sum(c.*R>0);

p=Na/Nr, q=1-p, Ip=[p-1.96*sqrt(p*q/Nr), p+1.96*sqrt(p*q/Nr)]

Pobl=25000*p, Ipobl=25000*Ip

Prácticas de Estadística

Práctica 5: Proceso de Poisson.

Ejercicio 8:

Una cola simple consiste en un punto de servicio A al que llegan demandas según una distribución de Poisson de parámetro λ y que son atendidas tras un tiempo de servicio cuya duración sigue una distribución exponencial de media μ. Cuando llega una demanda y el servidor está ocupado, la petición se encola esperando el servicio. Al inicio no existe ningún proceso en cola y el servidor está libre. Simular 100 veces la primera hora de servicio 3600 seg. y hallar para $\lambda = 0.05$ y $\mu = 18$:

1. Tiempo medio hasta que se ocupa el servidor por primera vez.
2. Tiempo medio en que está libre el servidor.
3. Porcentaje de procesos que son atendidos sin demora, calculados sobre el total de iteraciones.
4. Tamaño medio de la cola.
5. Si el tamaño máximo de la cola es de 5 procesos. Estimar la probabilidad de que se rebase en la primera hora.
6. Media de procesos atendidos en la primera hora.
7. Intervalo de confianza al 95% para la media de procesos atendidos.

Solución:
Nos basamos en las funciones MATLAB: **poissrnd(lambda,m,n)** y **exprnd(mu,m,n)**.
El programa consta de las partes:

1. **Inicio:** Parámetros de la simulación e inicio de contadores globales.
2. **Bucle para cada iteración:** Para cada iteración:
   (a) Iniciamos parámetros locales de la iteración.
   (b) Simulamos las llegadas, tiempo de servicio, comportamiento de la cola, etc.
   (c) Calculamos parámetros locales e incrementamos con ellos los globales. Guardamos los que nos interesen.
3. **Cálculo de los parámetros globales pedidos:** Se calculan las medias, procentajes, etc. sobre las 'Nit' iteraciones realizadas.

```
clear all,clc
format compact
disp('====================================')
disp('               Cola1                ')
disp('====================================')
disp('Parametros')
T=3600  %Tiempo considerado
Nit=100 %Numero Iteraciones
lambda=0.05
mu=18
disp('====================================')
% Iniciamos contadores globales
T0=0; % para el tiempo inicial
TL=0; % para el tiempo libre del servidor
NPS=0;% Contador del numero de procesos sin demora
NP=0; % Contador del numero total de procesos
NC=0; % Para el número medio de procesos en cola.
N5=0; % para contar los desbordes en cola
NM=0; % Para la media de procesos terminados
NM2=0;% Para el momento orden 2 del numero de procesos terminados

for k=1:Nit
    %Iniciamos parametros
```

```
cola=0; % Inicialmente no existe ningun proceso en cola
ocupado=0; % Flag: Inicialmente el servidor está libre
servicio=0; % Marcará el proceso en servicio/ultimo servido
proceso=[]; % Al final será una matriz con 5 columnas.
        % Cada fila un proceso que llega contiene:
        % [Num_Proceso T_llegada T_servicio T_inicio T_final]
for t=1:T
    x(t)=poissrnd(lambda); % Simulo llegada en intervalo [t-Delta(t), t]
    if x(t)>0
      d=exprnd(mu,x(t),1); % Simula tiempos de servicio para los procesos que llegan
      proceso=[proceso;[servicio(t)+cola(t)+[1:x(t)]' t*ones(x(t),1) d zeros(x(t),2)]];
      cola(t+1)=cola(t)+x(t);
    else
      cola(t+1)=cola(t);
    end

    if (ocupado==1)&(proceso(servicio(t),5)<=t)
        ocupado=0; %Ha terminado y queda libre el servidor
    end
    servicio(t+1)=servicio(t); % Sigue el proceso en curso
                               % Excepto desocupado y elementos en cola
    if (ocupado==0)&(cola(t+1)>0),
        ocupado=1; % Se ocupa el servidor
        cola(t+1)=cola(t+1)-1; % Se reduce la cola
        servicio(t+1)=servicio(t)+1; %Entra el siguiente en el servidor
        % Podemos indicar cuando entra y ¿termina? (si no rebasa T)
        proceso(servicio(t+1),[4,5])=[t,min([T,proceso(servicio(t+1),3)+t])];
    end
end
    %Incrementamos parametros globales pedidos
    [nproc,nfil]=size(proceso);

    % 1: Inicio de trabajo del servidor
    if nproc>0, T00=proceso(1,4);else T00=T;end
    T0=T0+T00; % para el tiempo inicial

    % 2: Tiempo libre del servidor
    if nproc>0, TTrabajo=sum(proceso(:,5)-proceso(:,4));else TTrabajo=0;end
    TL0=T-TTrabajo; % Tiempo libre= T_total - Tiempo trabajando
    TL=TL+TL0; % para el tiempo libre del servidor

    % 3: Porcentaje de sin demora
    if nproc>0,NPS0=sum(proceso(:,4)-proceso(:,2)==0);end
    NP=NP+nproc; % Número de procesos
    NPS=NPS+NPS0; % Numero de procesos sin demora

    % 4: tamaño medio de la cola
    NC0=mean(cola);
    NC=NC+NC0; % Para el número medio de procesos en cola.

    % 5: Desbordes en la cola si se limita a 5
    if max(cola)>5, N50=1; else N50=0;end
    N5=N5+N50; % para contar los desbordes en cola

    % 6: Media de procesos terminados.
    if nproc>0,NM0=sum(proceso(:,5)>0);else NM0=0;end
       % pero puede que el ultimo no se haya terminado
    if (NM0>0)&(proceso(NM0,4)+proceso(NM0,3)>T), NM0=NM0-1;end
    NM=NM+NM0; % Para la media de procesos terminados
    NM2=NM2+NM0^2; % Para momento orden 2
```

```
end
disp('Tiempo medio inicio')
T0=T0/Nit
disp('Tiempo libre y Porcentaje libre')
TL=TL/Nit
PL=TL/T*100
disp('Proporcion de sin demora (Total procesos sin demora/Total procesos)')
disp(' (ambos en las n iteraciones)')
P=NPS/NP
disp('Tamaño medio de cola')
NC=NC/Nit
disp(' Proporcion de desbordes (Iteraciones con desborde/Num. iteraciones)')
N5=N5/Nit
disp('Media e Intervalo de terminados')
NM=NM/Nit
NM2=NM2/Nit; V=NM2-Nm^2; % Nota: norminv(0.975)=1.96
I=[NM-norminv(0.975)*sqrt(V)/sqrt(Nit),NM+norminv(0.975)*sqrt(V)/sqrt(Nit)]
```

Prácticas de Estadística

Práctica 6: Proceso de Markov.

Ejercicio 9:

Un proceso puede estar en uno de los 5 estados:

1. Fuera del sistema
2. Punto de entrada
3. Activo
4. Espera
5. Punto de salida

Inicialmente ($t = 0$) existen 10.000 procesos estando todos fuera del sistema. La matriz de transición viene dada por:

$$T = \begin{pmatrix} 0.99 & 0.01 & 0 & 0 & 0 \\ 0.05 & 0.80 & 0.10 & 0.05 & 0 \\ 0 & 0 & 0.80 & 0.15 & 0.05 \\ 0 & 0 & 0.15 & 0.85 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Estimar:

- Probabilidad de que un proceso A se encuentre en cada una de los estados ("Fuera", "Inicio", "Activo", "Espera" y "Salida") para $t = 10$.
- Probabilidad de que un proceso A se encuentre en cada una de los estados ("Fuera", "Inicio", "Activo", "Espera" y "Salida") para $t = 100$.
- Probabilidad de que un proceso A se encuentre en cada una de los estados ("Fuera", "Inicio", "Activo", "Espera" y "Salida") para $t = 10000$.
- Estimar las probabilidades para $t = 10$ y $t = 100$ mediante simulación con los 1000 procesos.

Solución:

a: Vector inicial de estados $v_0 = (1, 0, 0, 0, 0)$. Vector estado para $t = 10$: $v_{10} = v_0 T^{10}$

b: Vector estado para $t = 100$: $v_{100} = v_0 T^{100}$

c: Vector estado para $t = 10000$: $v_1 0000 = v_0 T^{10000}$

NOTA: El resultado muestra experimentalmente que el estado 5 es absorbente. Si el proceso cae en él, ya no sale.

d: Se simula mediante:

```
T=[0.99 0.01 0 0 0;0.05 0.8 0.1 0.05 0;0 0 0.8 0.15 0.05;0 0 0.15 0.85 0;0 0 0 0 1]
v0=[1 0 0 0 0], v10=v0*T^10, v100=v0*T^100
NPROC=10000
v(1,:)=ones(1,NPROC);
for k=1:100
    for c=1:NPROC
        e=v(k,c);
        x=rand(1,1);
        switch e
            case 1
                if x<=.99, v(k+1,c)=1; else v(k+1,c)=2;end
            case 2
                if x<=.05, v(k+1,c)=1;
                elseif x<=0.85, v(k+1,c)=2;
                elseif x<=0.95, v(k+1,c)=3;
                else v(k+1,c)=4;
                end
            case 3
                if x<0.8, v(k+1,c)=3;
```

```
                elseif x<=0.95, v(k+1,c)=4;
                else v(k+1,c)=5;
                end
            case 4
                if x<=.15, v(k+1,c)=3; else v(k+1,c)=4;end
            case 5
                v(k+1,c)=5;
        end
    end
end
v10=v(11,:);
Freq10=[sum(v10==1) sum(v10==2) sum(v10==3) sum(v10==4) sum(v10==5)]/NPROC
v100=v(101,:);
Freq100=[sum(v100==1) sum(v100==2) sum(v100==3) sum(v100==4) sum(v100==5)]/NPROC
```

Rutinas Matlab relacionadas:

1: [H,P]=ztest(X,M,sigma,alfa,tipo)

Implementa el contraste paramétrico de la media de una Normal con desviación típica conocida.

Parámetros:

H: Resultado del contraste. Si H=0 se acepta la hipótesis nula y si H=1 se rechaza.

P: Significación. Es la probabilidad de equivocarnos al rechazar H_0, así si $p \geq \alpha$ aceptamos H_0.

X: Son los datos del test.

M: Es el valor medio contrastado.

sigma: Desviación típica poblacional.

alfa: Es el nivel de significación del test.

tipo: Es el tipo de contraste. Si ponemos como región crítica:

- 'both' es bilateral ($\mu \neq M$).
- 'right' es unilateral ($\mu > M$).
- 'left' es unilateral ($\mu < M$).

Ejercicio 10: Queremos contrastar si el valor 10 es la media de los datos $x = \{6.4106, 11.6808, 8.2239, 10.2002, 8.9109, 10.6070, 8.7993, 10.9799, 11.4787, 13.4238\}$, conocido que provienen de una normal de desviación típica $\sigma = 2$.

X=[6.4106 11.6808 8.2239 10.2002 8.9109 10.6070 8.7993 10.9799 11.4787 13.4238]

[H,P]=ztest(X,10,2,0.05,'both')

produce la salida H=0, p=0.91 que se interpreta como que debemos aceptar la hipótesis nula pues tenemos el 91% de probabilidad de equivocarnos al rechazarla. (NOTA: Los datos x fueron obtenidos mediante x=randn(1,10)*2+10)

2: [H,P]=ttest(X,M,alfa,tipo)

Implementa el contraste paramétrico de la media de una Normal con desviación típica desconocida.

Los parámetros tienen el mismo significado que en 'ztest'.

Ejercicio 11: Un servicio de pizzas asegura que atiende una petición en menos de 40 minutos. Se hace una prueba obteniendo los valores (en min.) {53,29,65,60,17,27,42,37}. ¿Podemos rechazar la afirmación al 5%?

X=[53,29,65,60,17,27,42,37]

[H,p]=ttest(X,40,0.05,'left')

produce la salida $H = 0$ y $P = 0.5796$, que se interpreta como que no rechazamos la hipótesis nula.

3: [H,P]=ttest2(X,Y,alfa,tipo)

Implementa el contraste paramétrico de la diferencia de medias entre dos distribuciones normales con desviaciones típicas desconocidas. Los parámetros X e Y son los datos, el resto tienen el mismo significado que en 'ztest'.

En tipo distinguimos cuál es la región crítica: 'both' contrasta que $\mu_X \neq \mu_Y$; 'right' que $\mu_X > \mu_Y$, y mediante 'left' que $\mu_X < \mu_Y$.

Ejercicio 12: Dos rutinas A y B resuelven el mismo problema, los tiempos de ejecución de la primera son: $t_A = \{5.6118, 1.7233, 4.3208, 8.7092, 3.8557, 7.9219, 6.2481, 8.8734, 2.0782, 5.6046, 3.5843, 11.8160, 7.6504, 8.7579, 3.8836, 5.0628\}$ y los de la segunda: $t_B = \{7.7275, 9.0984, 7.7221, 8.7015, 5.9482, 7.6462, 7.1764, 6.4229\}$ ¿Podemos, al 5% de significación, rechazar la afirmación de que A es más rápida que B?

X=[5.6118, 1.7233, 4.3208, 8.7092, 3.8557, 7.9219, 6.2481, 8.8734, 2.0782, 5.6046, 3.5843, 11.8160, 7.6504, 8.7579, 3.8836, 5.0628]

Y=[7.7275, 9.0984, 7.7221, 8.7015, 5.9482, 7.6462, 7.1764, 6.4229]

[H,p]=ttest2(X,Y,0.05,'left')

produce la salida $H = 0$ y $P = 0.0699$, que se interpreta como que no rechazamos la hipótesis nula pero que ha faltado muy poco para hacerlo, por ejemplo se rechazaría al 7% de significación.

4: [H,P]=vartest(X,V,alfa,tipo)

Realiza un contraste paramétrico de la varianza. El valor de la varianza a contrastar es el parámetro V. El contraste será de una o dos colas según el parámetro tipo. Si tipo es 'both' se contrastará si el verdadero valor de la varianza σ^2 es distinto V, si 'right' se contrasta si $\sigma^2 > V$ y si 'left', si $\sigma^2 < V$.

5: [H,P]=vartest2(X,Y,alfa,tipo)

Realiza un contraste paramétrico (F) de la igualdad de varianzas. Si tipo es 'both' se contrastará si $\sigma_X^2 \neq \sigma_Y^2$, si 'right' se contrasta si $\sigma_X^2 > \sigma_Y^2$ y si 'left', si $\sigma_X^2 < \sigma_Y^2$.

Ejercicio 13: Una máquina envasadora debe hacer paquetes con 1000 g. de peso. Una vez calibrada se miden los 10 primeros paquetes envasados obteniendo (en Kg.): X={1.0001, 1.0039, 1.0019, 1.0000, 0.9993, 1.0011, 1.0021, 1.0000, 1.0032, 0.9984}. Cuando lleva 10 semanas funcionando se vuelve a hacer una revisión tomando 8 medidas: Y={0.9900, 0.9963,0.9990, 0.9981, 0.9958, 1.0067, 0.9919, 0.9975}. Estudiar al 5%:

1. Si la media de los nuevos datos es 1Kg.=1000g.
2. Si las varianzas son iguales.
3. Si la máquina está sufriendo de holguras ($V_Y > V_X$)

4. Si la nueva varianza es el doble, o más del doble, que la anterior.

Solución: **1: [H,P]=ttest(Y,1,0.05,'both')**
Sale H=0, p=0.1255 y aceptamos la hipótesis nula.
2: [H,P]=vartest2(X,Y,0.05,'both')
Sale H=1, p=0.0054, indicando que son diferentes.
3: [H,P]=vartest2(X,Y,0.05,'left')
Sale H=1, p=0.0027, indicando que son diferentes.
4: [H,P]=vartest2(sqrt(2)*X,Y,0.05,'left')
Sale H=1, p=0.0270, indicando que rechazamos la nula de que $2\sigma_X^2 \geq \sigma_Y^2$.

NOTAS:

1) Nos hemos apoyado en el contraste de igualdad de varianzas para contrastar que $\sigma_Y^2 \geq 2\sigma_X^2$.

2) Podemos usar el contraste de diferencias de medias para contrastar, por ejemplo, que:

$\mu_X = \mu_Y + 6$, mediante: [H,P]=ttest2(X,Y+6,alfa,'both')

3) Podemos usar el contraste de la media "ttest" para contrastar la igualdad de medias para datos pareados, calculando previamente D=X-Y.

4) Todos los contrastes mostrados pueden devolver 4 variables, las H y P comentadas y además I y par. En I se devuelve el intervalo de confianza y en par los parámetros usados en el contraste.