Tarea 4 de Métodos Estadísticos Multivariados

Instrucciones: Contesta cada uno de los ejercicios en un archivo en Word. Puedes usar R o Excel, favor de anexar el código de R o el archivo de Excel. Al finalizar sube tus evidencias en el lugar correspondiente en Teams en formato PDF.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats

xls = pd.ExcelFile('/content/datos_tarea 4.xlsx')
data1 = pd.read_excel(xls, 'prob1' ).iloc[1:,1:8].astype('float') # ejercicio 1
data2 = pd.read_excel(xls, 'prob2').iloc[1:,1:7].astype('float') # ejercicio 2
data3 = pd.read_excel(xls, 'PROB 3').iloc[1:, 1:7].astype('float') # ejercicio 3
data4 = pd.read_excel(xls, 'PROB 4').iloc[1:,1:4].astype('float') # ejercicio 4
```

Ejercicio 1

1.- Se colectaron datos sobre la contaminación del aire en cierta ciudad. **Suponer normalidad en los datos.**

A continuación se muestran los datos con los que se trabajo.

```
data1.head()
```

	X1	X2	X3	X4	X5	X6	X7
1	8.0	98.0	7.0	2.0	12.0	8.0	2.0
2	7.0	107.0	4.0	3.0	9.0	5.0	3.0
3	7.0	103.0	4.0	3.0	5.0	6.0	3.0
4	10.0	88.0	5.0	2.0	8.0	15.0	4.0
5	6.0	91.0	4.0	2.0	8.0	10.0	3.0

Ejercicio 1 A)

Determinar si el vector de media poblacional es $\mu_0' = (8, 74, 5, 2, 10, 9, 3)$ con un nivel de significancia del 5% incluir el p-valor.

Procedimiento 1 A)

Se plantea la prueba de hipótesis:

- $H_0 : \mu = \mu_0$
- $H_1 : \mu \neq \mu_0$
- Con el estadístico de prueba: $T^2 = n(\bar{X} - \mu_0)' S^{-1} (\bar{X} - \mu_0)$
- Regla de decisión: Se rechaza H_0 con nivel de significancia:
  - α si $T^2 > \frac{(n-1)p}{n-p} F_{\alpha,p,n-p}$
  - $p_valor = P[F > (n-p)T^2/(n-1)p]$

```
mean_vector = data1.mean()
cov_mat = data1.cov()
mean_0 = np.matrix([8,74,5,2,10,9,3])
alfa = 0.05
p = data1.shape[1]
n = data1.shape[0]
print('Nivel de significancia, número de columnas y número de muestras es: ' , alfa,',', p,',',n, ' respectivamente.\n')
print('Vector de medias de la muestra de datos: \n', round(mean_vector,2 ), '\n')
print('Matriz de covarianza de la muestra de datos: \n',  cov_mat, '\n')
print('Vector de medias a comparar: \n', pd.DataFrame(mean_0.T, index = mean_vector.index))
```

```
Nivel de significancia, número de columnas y número de muestras es:  0.05 , 7 , 42  respectivamente.
```

```
Vector de medias de la muestra de datos:
 X1     7.50
X2    73.86
X3     4.55
X4     2.19
X5    10.05
X6     9.40
X7     3.10
dtype: float64

Matriz de covarianza de la muestra de datos:
           X1          X2        X3        X4         X5         X6        X7
X1  2.500000   -2.780488 -0.378049 -0.463415  -0.585366  -2.231707  0.170732
X2 -2.780488  300.515679  3.909408 -1.386760   6.763066  30.790941  0.623693
X3 -0.378049    3.909408  1.522067  0.673635   2.314750   2.821719  0.141696
X4 -0.463415   -1.386760  0.673635  1.182346   1.088269  -0.810685  0.176539
X5 -0.585366    6.763066  2.314750  1.088269  11.363531   3.126597  1.044135
X6 -2.231707   30.790941  2.821719 -0.810685   3.126597  30.978513  0.594657
X7  0.170732    0.623693  0.141696  0.176539   1.044135   0.594657  0.478513

Vector de medias a comparar:
      0
X1   8
X2  74
X3   5
X4   2
X5  10
X6   9
X7   3
```

```
T2 = n * (mean_vector.values - mean_0) * np.linalg.inv(cov_mat.values) * (mean_vector.values - mean_0).T
T2
```

```
matrix([[27.06840031]])
```

```
F_005_7_35 =  2.285
valor_critico = ((n-1)*p) / (n-p) * F_005_7_35
valor_critico_pvalor = (n-p) * T2 / (n-1)*p
print('valor crítico: ', valor_critico)
```

```
valor crítico:  18.737
```

Solución 1 A)

- Como $T^2 = 27.0684$ y el valor crítico , $F* = 18.37$, entonces $F* < T^2$, se rechaza H_0 lo cual quiere decir que existe evidencia suficiente que el vector de medias μ no toma esos valores y esta en la región de rechazo

Ejercicio 1 B)

Obtener IC simultáneos para las medias de cada componente, con un nivel de confianza global del 95%.

Para los IC se utiliza:

$$\bar{x}_i - \sqrt{\frac{p(n-1)}{n(n-p)} F_{\alpha,p,n-p} s_{ii}} < \mu_i < \bar{x}_i + \sqrt{\frac{p(n-1)}{n(n-p)} F_{\alpha,p,n-p} s_{ii}}$$

Procedimiento 1 B)

```
def IC_simultaneos(media_x_i, s_ii, n_rows, p_cols, F_valor):
  factor = np.sqrt( ( (p_cols*(n_rows-1)) / (n_rows*(n_rows-p_cols) )) * F_valor *  s_ii )
  low_lim = media_x_i - factor
  upp_lim = media_x_i + factor
  return low_lim, upp_lim


fvalor = 2.285#a = 0.05, p = 7, n-p = 35

# i = 0 # i va desde 0 a 6
for i in range(0, data1.shape[1]):
  low_interval, upper_interval=  IC_simultaneos( mean_vector[i] ,cov_mat.iloc[i][i] , data1.shape[0], data1.shape[1],fvalor)
  print('Para X_' , str(i+1), 'el intervalo de confianza para su media va de ',round(low_interval,4 ), ' a ', round(upper_interval,4),
```

```
Para X_ 1 el intervalo de confianza para su media va de  6.4439  a  8.5561 y el valor de mu_0 de X_ 1  es  8
Para X_ 2 el intervalo de confianza para su media va de  62.2785  a  85.4358 y el valor de mu_0 de X_ 2  es  74
Para X_ 3 el intervalo de confianza para su media va de  3.7236  a  5.3716 y el valor de mu_0 de X_ 3  es  5
Para X_ 4 el intervalo de confianza para su media va de  1.4642  a  2.9167 y el valor de mu_0 de X_ 4  es  2
Para X_ 5 el intervalo de confianza para su media va de  7.7961  a  12.2992 y el valor de mu_0 de X_ 5  es  10
Para X_ 6 el intervalo de confianza para su media va de  5.6872  a  13.1223 y el valor de mu_0 de X_ 6  es  9
Para X_ 7 el intervalo de confianza para su media va de  2.6332  a  3.5573 y el valor de mu_0 de X_ 7  es  3
```

Solución 1 B)

- Para X_1 el intervalo de confianza para su media va de 6.4439 a 8.5561 y el valor de μ_0 de X_1 es 8
- Para X_2 el intervalo de confianza para su media va de 62.2785 a 85.4358 y el valor de μ_0 de X_2 es 74
- Para X_3 el intervalo de confianza para su media va de 3.7236 a 5.3716 y el valor de μ_0 de X_3 es 5
- Para X_4 el intervalo de confianza para su media va de 1.4642 a 2.9167 y el valor de μ_0 de X_4 es 2
- Para X_5 el intervalo de confianza para su media va de 7.7961 a 12.2992 y el valor de μ_0 de X_5 es 10
- Para X_6 el intervalo de confianza para su media va de 5.6872 a 13.1223 y el valor de μ_0 de X_6 es 9
- Para X_7 el intervalo de confianza para su media va de 2.6332 a 3.5573 y el valor de μ_0 de X_7 es 3

Ejercicio 1 C)

Obtener IC simultáneos para las diferencias de medias poblacionales (ignorando unidades) con un nivel de confianza global del 95%.

Se conoce que la fórmula para los intervalos de confianza de la diferencia de medias es:

$$(x_i - x_j) + / - \sqrt{\frac{p(n-1)}{n(n-p} F_{\alpha,p,n-p}(s_{ii} - 2s_{i,j} + s_{jj})}$$

Procedimiento 1 C)

```

def intervalos_confianza_diferencia_medias(media_i, media_j, p,n, f_valor, cov_ij, var_i, var_j):
  dif_medias = (media_i - media_j)
  factor = np.sqrt( ( (p * (n-1) ) / ( n * (n-p) ) ) * f_valor * (var_i - (2 *cov_ij) + var_j) )
  lim_low = dif_medias - factor
  lim_upp = dif_medias + factor
  return lim_low, lim_upp

F_valor =2.285 #F a,p,n-p = alfa = 0.05, p = 7, 42-7

```

```
for i in range(0,6):

  for j in range(1,7):
    if (i != j)  :
      if i >j:
        continue
      mean_i = mean_vector[i]
      mean_j = mean_vector[j] #comparacion 1 y 2
      covariance_ij = cov_mat.values[i][j]
      variance_i = cov_mat.values[i][i]
      variance_j = cov_mat.values[j][j]
      inter_low, inter_upp = intervalos_confianza_diferencia_medias(mean_i, mean_j, p,n, F_valor, covariance_ij, variance_i, variance_
      print('Para X_i= ', str(i+1), 'y X_j = ', str(j+1), 'los intervalos de confianza son: ', str( round(inter_low, 4)), ' y ', str(
      j = j +1
  i = i + 1

```

```
Para X_i=  1 y X_j =  2 los intervalos de confianza son:  -78.0901  y  -54.6242
Para X_i=  1 y X_j =  3 los intervalos de confianza son:  1.4924  y  4.4124
Para X_i=  1 y X_j =  4 los intervalos de confianza son:  3.8756  y  6.7435
Para X_i=  1 y X_j =  5 los intervalos de confianza son:  -5.1374  y  0.0422
Para X_i=  1 y X_j =  6 los intervalos de confianza son:  -6.019  y  2.2094
Para X_i=  1 y X_j =  7 los intervalos de confianza son:  3.3201  y  5.4894
Para X_i=  2 y X_j =  3 los intervalos de confianza son:  57.8528  y  80.7663
Para X_i=  2 y X_j =  4 los intervalos de confianza son:  60.012  y  83.3213
Para X_i=  2 y X_j =  5 los intervalos de confianza son:  52.2726  y  75.3465
Para X_i=  2 y X_j =  6 los intervalos de confianza son:  53.4791  y  75.4257
Para X_i=  2 y X_j =  7 los intervalos de confianza son:  59.198  y  82.3258
Para X_i=  3 y X_j =  4 los intervalos de confianza son:  1.579  y  3.1352
```

```
Para X_i=  3 y X_j =  5 los intervalos de confianza son:  -7.4192  y  -3.5808
Para X_i=  3 y X_j =  6 los intervalos de confianza son:  -8.3186  y  -1.3957
Para X_i=  3 y X_j =  7 los intervalos de confianza son:  0.5771  y  2.3276
Para X_i=  4 y X_j =  5 los intervalos de confianza son:  -10.0079  y  -5.7063
Para X_i=  4 y X_j =  6 los intervalos de confianza son:  -11.0964  y  -3.3322
Para X_i=  4 y X_j =  7 los intervalos de confianza son:  -1.6686  y  -0.1409
Para X_i=  5 y X_j =  6 los intervalos de confianza son:  -3.3696  y  4.6553
Para X_i=  5 y X_j =  7 los intervalos de confianza son:  4.8664  y  9.0384
Para X_i=  6 y X_j =  7 los intervalos de confianza son:  2.6349  y  9.9842
```

Solución 1 C)

- Para $X_i = 1$ y $X_j = 2$ los intervalos de confianza son: -78.0901 y -54.6242
- Para $X_i = 1$ y $X_j = 3$ los intervalos de confianza son: 1.4924 y 4.4124
- Para $X_i = 1$ y $X_j = 4$ los intervalos de confianza son: 3.8756 y 6.7435
- Para $X_i = 1$ y $X_j = 5$ los intervalos de confianza son: -5.1374 y 0.0422
- Para $X_i = 1$ y $X_j = 6$ los intervalos de confianza son: -6.019 y 2.2094
- Para $X_i = 1$ y $X_j = 7$ los intervalos de confianza son: 3.3201 y 5.4894
- Para $X_i = 2$ y $X_j = 3$ los intervalos de confianza son: 57.8528 y 80.7663
- Para $X_i = 2$ y $X_j = 4$ los intervalos de confianza son: 60.012 y 83.3213
- Para $X_i = 2$ y $X_j = 5$ los intervalos de confianza son: 52.2726 y 75.3465
- Para $X_i = 2$ y $X_j = 6$ los intervalos de confianza son: 53.4791 y 75.4257
- Para $X_i = 2$ y $X_j = 7$ los intervalos de confianza son: 59.198 y 82.3258
- Para $X_i = 3$ y $X_j = 4$ los intervalos de confianza son: 1.579 y 3.1352
- Para $X_i = 3$ y $X_j = 5$ los intervalos de confianza son: -7.4192 y -3.5808
- Para $X_i = 3$ y $X_j = 6$ los intervalos de confianza son: -8.3186 y -1.3957
- Para $X_i = 3$ y $X_j = 7$ los intervalos de confianza son: 0.5771 y 2.3276
- Para $X_i = 4$ y $X_j = 5$ los intervalos de confianza son: -10.0079 y -5.7063
- Para $X_i = 4$ y $X_j = 6$ los intervalos de confianza son: -11.0964 y -3.3322
- Para $X_i = 4$ y $X_j = 7$ los intervalos de confianza son: -1.6686 y -0.1409
- Para $X_i = 5$ y $X_j = 6$ los intervalos de confianza son: -3.3696 y 4.6553
- Para $X_i = 5$ y $X_j = 7$ los intervalos de confianza son: 4.8664 y 9.0384
- Para $X_i = 6$ y $X_j = 7$ los intervalos de confianza son: 2.6349 y 9.9842

Ejercicio 1 D)

Obtener IC simultáneos para la media de cada componente, con un nivel de confianza global del 95% usando el método de Bonferroni.

```
data2.head()
```

	X1	X2	X3	X4	X5	X6
1	0.889	1.389	1.555	2.222	1.945	1.0
2	2.813	1.437	0.999	2.312	2.312	2.0
3	1.454	1.091	2.364	2.455	2.909	3.0
4	0.294	0.941	1.059	2.000	1.000	1.0
5	2.727	2.545	2.819	2.727	4.091	0.0

El método de Bonferroni plante las siguientes hipótesis:

- $H_0 : \mu_1 = \mu_0$
- $H_1 : \mu_1 \neq \mu_0$

La fórmula es:

$$\bar{x}_i - t_{\alpha/2p,n-1}\sqrt{s_{ii}/n} < \mu_i < \bar{x}_i + t_{\alpha/2p,n-1}\sqrt{s_{ii}/n}$$

Procedimiento 1 D)

```
def IC_simultaneos_Bonferroni(media_i, t_valor, var_i, n_rows):
  factor = t_valor * np.sqrt(var_i /n_rows)
  low_lim = media_i - factor
  upp_lim = media_i + factor
```

```
  return low_lim, upp_lim

tvalor = 2.8319 # 0.05/2 * 7 ,  41 = 0.17500, 41
for i in range(0, data1.shape[1]):
  media = mean_vector[i]
  varianza = cov_mat.values[i,i]
  lim_inf , lim_sup = IC_simultaneos_Bonferroni(media, tvalor, varianza, data1.shape[0])
  print('El intervalo de confianza para x_', str(i), 'es de :', str(round(lim_inf,4)), ' a ', str(round(lim_sup,4)) )
```

```
El intervalo de confianza para x_ 0 es de : 6.8091  a  8.1909
El intervalo de confianza para x_ 1 es de : 66.2821  a  81.4322
El intervalo de confianza para x_ 2 es de : 4.0085  a  5.0867
El intervalo de confianza para x_ 3 es de : 1.7153  a  2.6656
El intervalo de confianza para x_ 4 es de : 8.5746  a  11.5206
El intervalo de confianza para x_ 5 es de : 6.9726  a  11.8369
El intervalo de confianza para x_ 6 es de : 2.793  a  3.3975
```

Solución 1 D)

- El intervalo de confianza para x_0 es de : 6.8091 a 8.1909
- El intervalo de confianza para x_1 es de : 66.2821 a 81.4322
- El intervalo de confianza para x_2 es de : 4.0085 a 5.0867
- El intervalo de confianza para x_3 es de : 1.7153 a 2.6656
- El intervalo de confianza para x_4 es de : 8.5746 a 11.5206
- El intervalo de confianza para x_5 es de : 6.9726 a 11.8369
- El intervalo de confianza para x_6 es de : 2.793 a 3.3975

Ejercicio 2

Se colectaron datos de tratamiento de radioterapia en muchos pacientes. El investigador considera que la **muestra es muy grande, usar la teoría de muestras muy grandes.**

Ejercicio 2 A)

Determinar cuáles vectores μ están en la región de confianza del 95%.

μ_1	μ_2	$\mu 3$
3.60	3.60	3.60
2.00	1.90	2.00
2.10	2.10	2.10
2.15	2.15	2.15
2.60	2.60	3.00
1.30	1.30	1.30

Inferencias para μ , considerando muestra grande:

- $H_0 : \mu = \mu_0$
- $H_1 : \mu \neq \mu_0$

Con estadístico de prueba $T^2 = n(\bar{x} - \mu)' S^{-1} (\bar{x} - \mu)$

Con regla de decisión: Se rechaza H_0 si $T^2 > X^2_{\alpha,p}$

Procedimiento 2 A)

```
mu_2_1 = np.matrix([3.60,2.00,2.10,2.15,2.60,1.30])
mu_2_2 = np.matrix([3.60,1.90,2.10,2.15,2.60,1.30])
mu_2_3 = np.matrix([3.60,2.00,2.10,2.15,3.00,1.30])
medias2 = data2.mean().values.T
covarianza2 = data2.cov().values
X_a_p = 12.5915 # a=0.05 , p = 6
```

```
def region_confianza_para_medias_muestras_grandes(nrows, media_datos, cov, media_dada_i):
  dif_media_datos_dada = media_datos - media_dada_i
  T2_estadistico = nrows * dif_media_datos_dada * np.linalg.inv(cov) * dif_media_datos_dada.T
  return T2_estadistico

```

```
for mu_2_N in [mu_2_1,mu_2_2,mu_2_3]:
  T2_calculado = region_confianza_para_medias_muestras_grandes(data2.shape[0], medias2, covarianza2, mu_2_N)
  print('Para: ', str(mu_2_N), ' el T Cuadrado es: ', str(T2_calculado))
```

```
Para:  [[3.6  2.   2.1  2.15 2.6  1.3 ]]  el T Cuadrado es:  [[18.9681475]]
Para:  [[3.6  1.9  2.1  2.15 2.6  1.3 ]]  el T Cuadrado es:  [[11.10784112]]
Para:  [[3.6  2.   2.1  2.15 3.   1.3 ]]  el T Cuadrado es:  [[89.89236002]]
```

Solución 2 A)

- Para $\mu_1 = [3.62.2.12.152.61.3]$ el $T^2 = 18.9681$ Si $\mathrm{X}^2 = 12.5918$, entonces $T^2 > \mathrm{X}^2$ por tanto esta media no esta en la región de confianza.
- Para: $\mu_2 = [3.61.92.12.152.61.3]$ el $T^2 = 11.1078$ Si $\mathrm{X}^2 = 12.5918$, entonces $T^2 < \mathrm{X}^2$ por tanto esta media si esta en la región de confianza.
- Para: $\mu_2 = [3.62.2.12.153.1.3]$ el $T^2 = 89.8923$ Si $\mathrm{X}^2 = 12.5918$, entonces $T^2 > \mathrm{X}^2$ por tanto esta media no esta en la región de confianza.

Ejercicio 2 B)

Con la teoría de muestras muy grandes obtener IC simultáneos con un nivel de confianza global del 95% para la media de cada variable.

Procedimiento 2 B)

```
def IC_muestras_muyGrandes(media_x, chi_cuadrada,var_xi, nrows):
  factor = np.sqrt( chi_cuadrada * (var_xi/nrows )  )
  lim_inferior = media_x - factor
  lim_superior = media_x + factor
  return lim_inferior, lim_superior

for i in range(0,data2.shape[1]):
  lim_inferior_2, lim_superior_2 = IC_muestras_muyGrandes(medias2[i], X_a_p,covarianza2[i,i], data2.shape[0])
  print('Para X_',str(i+1), 'el intervalo de confianza es: ', str(round(lim_inferior_2,4)), ' , ', str(round(lim_superior_2,4)))
```

```
Para X_ 1 el intervalo de confianza es:  2.769  ,  4.3157
Para X_ 2 el intervalo de confianza es:  1.5288  ,  2.09
Para X_ 3 el intervalo de confianza es:  1.8666  ,  2.4086
Para X_ 4 el intervalo de confianza es:  2.0899  ,  2.3281
Para X_ 5 el intervalo de confianza es:  2.242  ,  2.9077
Para X_ 6 el intervalo de confianza es:  0.9428  ,  1.6082
```

Solución 2 B)

- Para X_1 el intervalo de confianza es de: 2.769 a 4.3157
- Para X_2 el intervalo de confianza es de : 1.5288 a 2.09
- Para X_3 el intervalo de confianza es de : 1.8666 a 2.4086
- Para X_4 el intervalo de confianza es de: 2.0899 a 2.3281
- Para X_5 el intervalo de confianza es de: 2.242 a 2.9077
- Para X_6 el intervalo de confianza es de: 0.9428 a 1.6082

Ejercicio 2 C)

Resolver B) con el método de Bonferroni.

Procedimiento 2 C)

```
def IC_simultaneos_Bonferroni(media_i, t_valor1, var_i, n_rows):
  factor = t_valor1 * np.sqrt(var_i /n_rows)
  low_lim = media_i - factor
  upp_lim = media_i + factor
  return low_lim, upp_lim
```

```
tvalor =2.6940 # t de 0.05/2*p , n-1 = 0.05/2*6 , 97
for i in range(0 , data2.shape[1]):
```

```
lim_inf, lim_sup = IC_simultaneos_Bonferroni(medias2[i], tvalor, covarianza2[i,i], data2.shape[0])
print( 'Los intervalos de confianza para X_', str(i+1), 'por el método de Bonferroni es de: ', str(round(lim_inf,4)),
      'a ', str(round(lim_sup,4)))
```

```
Los intervalos de confianza para X_ 1 por el método de Bonferroni es de:  2.9552 a  4.1295
Los intervalos de confianza para X_ 2 por el método de Bonferroni es de:  1.5963 a  2.0224
Los intervalos de confianza para X_ 3 por el método de Bonferroni es de:  1.9319 a  2.3433
Los intervalos de confianza para X_ 4 por el método de Bonferroni es de:  2.1186 a  2.2994
Los intervalos de confianza para X_ 5 por el método de Bonferroni es de:  2.3221 a  2.8275
Los intervalos de confianza para X_ 6 por el método de Bonferroni es de:  1.0229 a  1.5281
```

Solución 2 C)

- Los intervalos de confianza para X_1 por el método de Bonferroni es de: 2.9552 a 4.1295
- Los intervalos de confianza para X_2 por el método de Bonferroni es de: 1.5963 a 2.0224
- Los intervalos de confianza para X_3 por el método de Bonferroni es de: 1.9319 a 2.3433
- Los intervalos de confianza para X_4 por el método de Bonferroni es de: 2.1186 a 2.2994
- Los intervalos de confianza para X_5 por el método de Bonferroni es de: 2.3221 a 2.8275
- Los intervalos de confianza para X_6 por el método de Bonferroni es de: 1.0229 a 1.5281

Ejercicio 3

Se tiene una muestra aleatoria de datos sobre minerales contenidos en los huesos

Se plantea la prueba de hipótesis:

- $H_0 : \mu = \mu_0$
- $H_1 : \mu \neq \mu_0$
- Con el estadístico de prueba: $T^2 = n(\bar{X} - \mu_0)' S^{-1} (\bar{X} - \mu_0)$
- Regla de decisión: Se rechaza H_0 con nivel de significancia:
  - α si $T^2 > \frac{(n-1)p}{n-p} F_{\alpha,p,n-p}$
  - $p_v alor = P[F > (n-p)T^2/(n-1)p]$

Ejercicio 3 A)

Determinar si $\mu_0 = (0.85, 0.79, 1.80, 1.70, 0.70, 0.70)$ con un nivel de significancia del 5%.

Procedimiento 3 A)

```
data3.head()
```

	X1	X2	X3	X4	X5	X6
1	1.103	1.052	2.139	2.238	0.873	0.872
2	0.842	0.859	1.873	1.741	0.590	0.744
3	0.925	0.873	1.887	1.809	0.767	0.713
4	0.857	0.744	1.739	1.547	0.706	0.674
5	0.795	0.809	1.734	1.715	0.549	0.654

```
medias_3 = data3.mean().values
media_3_0 = np.matrix([0.85, 0.79, 1.80, 1.70, 0.70, 0.70 ])
covarianza_3 = data3.cov().values

T2_ejercicio3 = data3.shape[0] * (medias_3 - media_3_0) * np.linalg.inv(covarianza_3) * (medias_3 - media_3_0).T
T2_ejercicio3
```

```
matrix([[15.4904249]])
```

```
Fvalor = 2.6283
coef = ((data3.shape[0]-1)* data3.shape[1]) / (data3.shape[0] - data3.shape[1])
valor_critico_data3 = coef * Fvalor
valor_critico_data3
```

```
19.91974736842105
```

Solución 3 A)

Si el estadístico de prueba $T^2 = 15.4909$ y el valor crítico es 19.9197, se concluye que $T^2 <$ valor crítico, por tant no se rechaza H_0, las medias no son diferentes

Ejercicio 3 B)

Obtener IC simultáneos para las diferencias de medias con un nivel de confianza global del 95%.

Procedimiento 3 B)

```
data3.shape
```

```
(25, 6)
```

```
F_valor =2.6283

for i in range(0,data3.shape[1]-1):

  for j in range(1,data3.shape[1]):
    if (i != j)  :
      if i >j:
        continue
      mean3_i = medias_3[i]
      mean3_j = medias_3[j] #comparacion 1 y 2
      covariance3_ij = covarianza_3[i][j]
      variance3_i = covarianza_3[i][i]
      variance3_j = covarianza_3[j][j]
      inter_low3, inter_upp3 = intervalos_confianza_diferencia_medias(mean3_i, mean3_j, data3.shape[1],data3.shape[0], F_valor,
                                                                      covariance3_ij, variance3_i, variance3_j)
      print('Para X_i= ', str(i+1), 'y X_j = ', str(j+1), 'los intervalos de confianza son: ', str( round(inter_low3, 4)), ' y ', str(
      j = j +1
  i = i + 1
```

```
Para X_i=  1 y X_j =  2 los intervalos de confianza son:  -0.0285  y  0.0795
Para X_i=  1 y X_j =  3 los intervalos de confianza son:  -1.1458  y  -0.752
Para X_i=  1 y X_j =  4 los intervalos de confianza son:  -1.0747  y  -0.7074
Para X_i=  1 y X_j =  5 los intervalos de confianza son:  0.0684  y  0.2104
Para X_i=  1 y X_j =  6 los intervalos de confianza son:  0.0717  y  0.2282
Para X_i=  2 y X_j =  3 los intervalos de confianza son:  -1.1831  y  -0.7656
Para X_i=  2 y X_j =  4 los intervalos de confianza son:  -1.0921  y  -0.7409
Para X_i=  2 y X_j =  5 los intervalos de confianza son:  0.0452  y  0.1826
Para X_i=  2 y X_j =  6 los intervalos de confianza son:  0.0666  y  0.1823
Para X_i=  3 y X_j =  4 los intervalos de confianza son:  -0.0561  y  0.1718
Para X_i=  3 y X_j =  5 los intervalos de confianza son:  0.8728  y  1.3037
Para X_i=  3 y X_j =  6 los intervalos de confianza son:  0.8708  y  1.3269
Para X_i=  4 y X_j =  5 los intervalos de confianza son:  0.8399  y  1.221
Para X_i=  4 y X_j =  6 los intervalos de confianza son:  0.8485  y  1.2335
Para X_i=  5 y X_j =  6 los intervalos de confianza son:  -0.0587  y  0.0798
```

Solución 3 B)

- Para $X_i = 1$ y $X_j = 2$ los intervalos de confianza son de : -0.0285 a 0.0795
- Para $X_i = 1$ y $X_j = 3$ los intervalos de confianza son de : -1.1458 a -0.752
- Para $X_i = 1$ y $X_j = 4$ los intervalos de confianza son de: -1.0747 a -0.7074
- Para $X_i = 1$ y $X_j = 5$ los intervalos de confianza son de: 0.0684 a 0.2104
- Para $X_i = 1$ y $X_j = 6$ los intervalos de confianza son de: 0.0717 a 0.2282
- Para $X_i = 2$ y $X_j = 3$ los intervalos de confianza son de: -1.1831 a -0.7656
- Para $X_i = 2$ y $X_j = 4$ los intervalos de confianza son de: -1.0921 a -0.7409
- Para $X_i = 2$ y $X_j = 5$ los intervalos de confianza son de: 0.0452 a 0.1826
- Para $X_i = 2$ y $X_j = 6$ los intervalos de confianza son de: 0.0666 a 0.1823
- Para $X_i = 3$ y $X_j = 4$ los intervalos de confianza son de: -0.0561 a 0.1718
- Para $X_i = 3$ y $X_j = 5$ los intervalos de confianza son de: 0.8728 a 1.3037
- Para $X_i = 3$ y $X_j = 6$los intervalos de confianza son de: 0.8708 a 1.3269
- Para $X_i = 4$ y $X_j = 5$ los intervalos de confianza son de: 0.8399 a 1.221

- Para $X_i = 4$ y $X_j = 6$los intervalos de confianza son de: 0.8485 a 1.2335
- Para $X_i = 5$ y $X_j = 6$ los intervalos de confianza son de: -0.0587 a 0.0798

Ejercicio 3 C)

Validar los supuestos para que los resultados anteriores sean válidos.

Procedimiento 3 C)

Solución 3 C)

Se anexa el procedimiento al final del documento con el titulo "Prueba de normalidad para datos del ejercicio 3"

Ejercicio 4

Se colectaron datos sobre el puntaje de exámenes en estudiantes universitarios.

Ejercicio 4 A)

Determinar que vectores de medias están en la región de confianza del 95%

μ_1	μ_2	$\mu 3$
527	523	525
53	55	53
25	26	26

Procedimiento 4 A)

```
mean_data4 = data4.mean().values
cov_data4 = data4.cov().values
mean_4_1 = np.matrix([527,53,25])
mean_4_2 = np.matrix([523,55,26])
mean_4_3 = np.matrix([525,53,26])

```

```
for mu_2_N in [mean_4_1,mean_4_2,mean_4_3]:
  T2_calculado = region_confianza_para_medias_muestras_grandes(data4.shape[0], mean_data4, cov_data4, mu_2_N)
  print('Para: ', str(mu_2_N), ' el T Cuadrado es: ', str(T2_calculado))
```

```
Para:  [[527  53  25]]  el T Cuadrado es:  [[4.14625914]]
Para:  [[523  55  26]]  el T Cuadrado es:  [[6.82611991]]
Para:  [[525  53  26]]  el T Cuadrado es:  [[8.51455066]]
```

Solución 4 A)

- Para: μ_1 = [527, 53 , 25] el $T^2 = 4.1462$
- Para: μ_2 =[523, 55 , 26] el $T^2 = 6.8261$
- Para: μ_3 =[525 , 53 , 26] el $T^2 = 8.5145$

El $X_{\alpha=0.05,p=3} = 7.4182$

Se rechaza H_0 si $T^2 > X^2_{\alpha,p}$ por tanto:

- Los vector μ_1 y μ_2 estan dentro de la región de confianza, sin embargo μ_3 esta en la región de rechazo.

Ejercicio 4 B)

Obtener IC simultáneos para diferencias de medias con un nivel de confianza global del 95%

Procedimiento 4 B)

```
```

```
X_a_p_ejercicio4 = 7.8147

for i in range(0,data4.shape[1]):
  lim_inferior_4, lim_superior_4 = IC_muestras_muyGrandes(mean_data4[i], X_a_p_ejercicio4 ,cov_data4[i,i], data4.shape[0])
  print('Para X_',str(i+1), 'el intervalo de confianza es de : ', str(round(lim_inferior_4,4)), ' a ', str(round(lim_superior_4,4)))
```

```
Para X_ 1 el intervalo de confianza es de :  503.7454  a  549.427
Para X_ 2 el intervalo de confianza es de :  51.3247  a  58.0546
Para X_ 3 el intervalo de confianza es de :  23.6856  a  26.5673
```

Solución 4 B)

- Para X_1 el intervalo de confianza es de 503.7454 a 549.427
- Para X_2 el intervalo de confianza es de 51.3247 a 58.0546
- Para X_3 el intervalo de confianza es de 23.6856 a 26.5673

Ejercicio 4 C)

Mencionar los supuestos para que los resultados anteriores sean válidos.

Procedimiento 4 C)

```
data4.shape
```

```
(87, 3)
```

Solución 4 C)

Se observa que el número de filas, $n = 87$ y el número de columnas $p = 3$, el supuesto de muestra grande se cumple si:

- n>>p, sustituyendo 87 >> 3 .
- n-p > 50 , lo cual al sustituir con los los datos 4, 87-3= 85 > 50

Se concluye que los datos cumplen con los requisitos para muestra grande.