

# Tarea1

---

ESTADÍSTICA MULTIVARIADA

Benitez Patiño Ian, Garcia Garcia Adrian, Osornio Buenrostro Marco  
Antonio

UNIVERSIDAD LA SALLE | GRUPO ACT. 700

**Ejercicio 3.** Genera una muestra aleatoria de tamaño  $n = 100$  a partir de una distribución gaussiana 3-dimensional, en donde una de las variables tenga varianza alta. Lleva a cabo un PCA usando la matriz de covarianza y la matriz de correlación. En cada caso, encuentra los eigenvalores y eigenvectores, dibuja un *scree plot* y calcula los resultados de los PC.

```
# Definimos la media y la matriz de covarianza
mean = [0, 0, 0] # media
cov = [[1, 0, 0], [0, 1, 0], [0, 0, 10]] # matriz de covarianza

# Generamos una muestra aleatoria de tamaño n=100
X = np.random.multivariate_normal(mean, cov, 100)
```



#PCA con matriz de covarianzas

```
def PCAc(X , num_componentes):

    X_media = X - np.mean(X , axis = 0)

    cov_mat = np.cov(X_media , rowvar = False)

    eigen_val , eigen_vec = np.linalg.eigh(cov_mat)

    sorted_index = np.argsort(eigen_val)[::-1]
    sorted_eigenval = eigen_val[sorted_index]
    sorted_eigenvec = eigen_vec[:,sorted_index]

    eigenvector_ = sorted_eigenvec[:,0:num_componentes]

    X_red = np.dot(eigenvector_.transpose() , X_media.transpose() ).transpose()

    return X_red

# Aplicamos PCA a los datos elegidos
mat_red = PCAc(X , 2)

# Tabla que muestra a las dos componentes principales
principal_df = pd.DataFrame(mat_red , columns = ['PC1','PC2'])

principal_df = pd.concat([principal_df ] , axis = 1)
principal_df
```

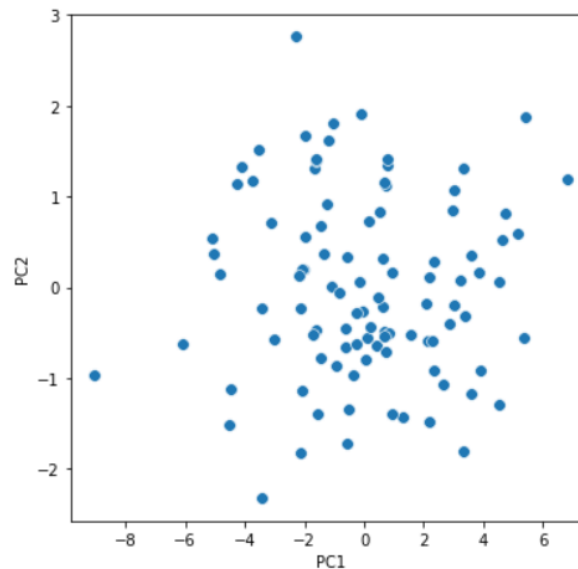
	PC1	PC2
0	4.648144	0.528609
1	3.563850	0.354466
2	-0.598335	-0.461517
3	2.139965	-0.585436
4	3.032060	-0.199938
...	...	...
95	0.681188	1.158687
96	-6.095056	-0.621084
97	2.076606	-0.176087
98	-1.351889	0.374912
99	1.544269	-0.517533

100 rows × 2 columns

```
import seaborn as sb
import matplotlib.pyplot as plt
```

```
plt.figure(figsize = (6,6))
sb.scatterplot(data = principal_df , x = 'PC1',y = 'PC2' , s = 60 , palette= 'icefire')
```

<AxesSubplot:xlabel='PC1', ylabel='PC2'>



```
#PCA con matriz de correlación
```

```
def PCA(X , num_componentes):
```

```
    X_media = X - np.mean(X , axis = 0)
```

```
    cor_mat = np.corrcoef(X_media , rowvar = False)
```

```
    eigen_val , eigen_vec = np.linalg.eigh(cor_mat)
```

```
    sorted_index = np.argsort(eigen_val)[::-1]
```

```
    sorted_eigenval = eigen_val[sorted_index]
```

```
    sorted_eigenvec = eigen_vec[:,sorted_index]
```

```
    eigenvector_ = sorted_eigenvec[:,0:num_componentes]
```

```
    X_red = np.dot(eigenvector_.transpose() , X_media.transpose() ).transpose()
```

```
    return X_red
```

```
# Aplicamos PCA a los datos elegidos
```

```
mat_red = PCA(X , 2)
```

```
# Tabla que muestra a las dos componentes principales
```

```
principal_df = pd.DataFrame(mat_red , columns = ['PC1','PC2'])
```

```
principal_df = pd.concat([principal_df ] , axis = 1)
```

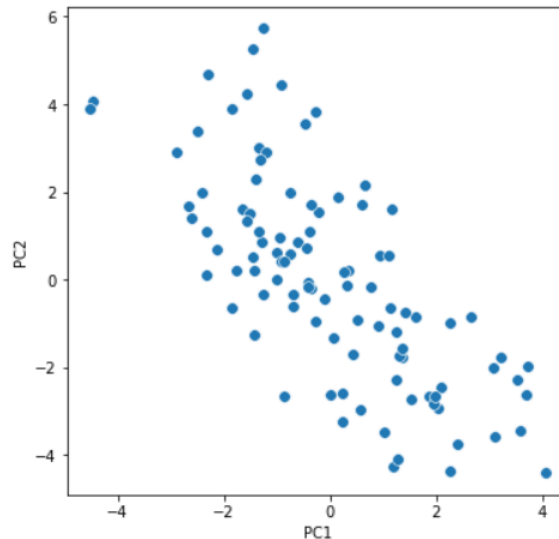
```
principal_df
```

	PC1	PC2
0	-3.607428	1.427363
1	-2.435195	0.983170
2	0.818808	0.077143
3	-0.884138	1.191351
4	-1.880775	1.301595
...	...	...
95	-1.217052	-0.652959
96	4.522421	-1.846717
97	-1.460361	1.022955
98	0.693147	-0.826772
99	-1.424793	1.366502

```
import seaborn as sb
import matplotlib.pyplot as plt

plt.figure(figsize = (6,6))
sb.scatterplot(data = principal_df , x = 'PC1',y = 'PC2' , s = 60 , palette= 'icefire')
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f2ef55da400>



**Ejercicio 4.** Considera los siguientes puntos en  $\mathbb{R}^2$ :

$$(-1, 1), (0, 0), (1, 1).$$

- (1) ¿Cuáles son las componentes principales de los datos?
- (2) Si proyectamos los puntos de datos originales en el subespacio unidimensional abarcado por el componente principal que elija, ¿cuáles son las coordenadas en este subespacio? ¿Cuál es la varianza de los datos proyectados?

```
X=np.array([[ -1,1],[0,0],[1,1]])

# Aplicamos PCA a los datos elegidos
mat_red = PCA(X , 1)

# Tabla que muestra a las dos componentes principales
principal_df = pd.DataFrame(mat_red , columns = ['PC1'])

principal_df = pd.concat([principal_df ] , axis = 1)
principal_df

print("los datos proyectados son:",principal_df)
print("la varianza de los datos proyectados es:",np.var(mat_red))
```

```
los datos proyectados son:    PC1
0 -1.0
1  0.0
2  1.0
la varianza de los datos proyectados es: 0.6666666666666666
```