Prueba con python y latex

Análisis de comopnenetes

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

```
iris = sns.load_dataset("iris")
iris.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
cop=iris
iris = iris.drop('species', axis=1)
iris.corr()
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

Estandarización

Necesitamos que las variables numéricas estén estandarizadas porque se observa una **diferencia de magnitud** entre ellas. Podemos ver que *petal_width* y *sepal_length* están en magnitudes totalmente distintas, con lo cual a la hora de calcular los componentes principales *sepal_length* va a dominar a *petal_width* por la mayor escala de magnitud y, por tanto, mayor rango de varianza.

```
X = iris.loc[:, ["sepal_length", "sepal_width", "petal_length", "petal_width"]]
Y = cop.loc[:, ["species"]]

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

X = scaler.fit_transform(X)
```

PCA

Una vez que tenemos nuestras variables numéricas estandarizadas, es hora de proceder con el cálculo de los componentes principales. En este caso, vamos a elegir dos componentes principales.

```
from sklearn.decomposition import PCA
PCA = PCA(n_components=2)
components = PCA.fit_transform(X)
PCA.components_
```

```
array([[ 0.52106591, -0.26934744,  0.5804131 ,  0.56485654],
       [ 0.37741762,  0.92329566,  0.02449161,  0.06694199]])
```

Se observa que el primer componente principal da casi el mismo peso a *sepal_length, petal_length* y *petal.width*, mientras que el segundo componente principal da peso primordialmente a *sepal_width*.

Además, podemos ver con los valores propios la varianza explicada por los dos componentes principales y la varianza explicada acumulada

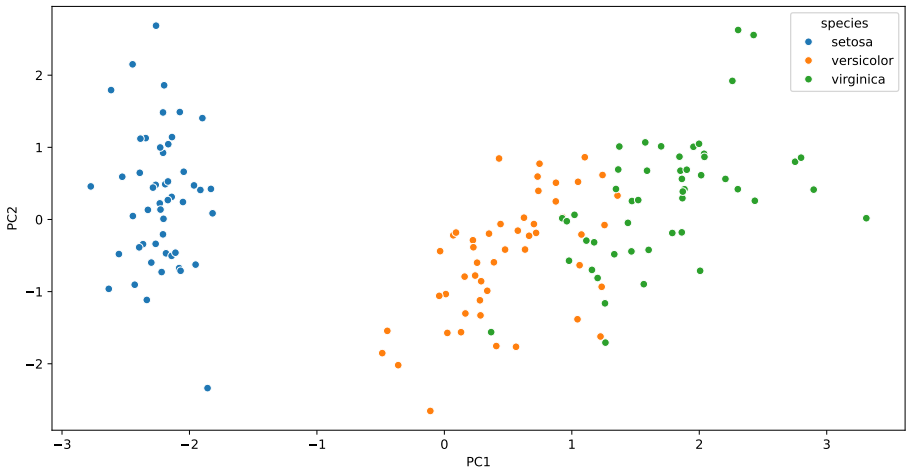
```
cumVar = pd.DataFrame(np.cumsum(PCA.explained_variance_ratio_)*100,
                      columns=["cumVarPerc"])
expVar = pd.DataFrame(PCA.explained_variance_ratio_*100, columns=["VarPerc"])
pd.concat([expVar, cumVar], axis=1)\
    .rename(index={0: "PC1", 1: "PC2"})
```

	VarPerc	cumVarPerc
PC1	72.962445	72.962445
PC2	22.850762	95.813207

El primer componente principal explica un 72.96% de la variación total de los datos originales, mientras que el segundo explica un 22.85%. Conjuntamente, los dos componentes principales explican alrededor del 95.81% de la variación total, un porcentaje bastante elevado.

```
componentsDf = pd.DataFrame(data = components, columns = ['PC1', 'PC2'])
pcaDf = pd.concat([componentsDf, Y], axis=1)

plt.figure(figsize=(12, 6))
sns.scatterplot(data=pcaDf, x="PC1", y="PC2", hue="species")
plt.show()
```



Vemos que existe una clara separación entre los diferentes tipos de especies que hay. Cada clase de flor representa un clúster específico y en el gráfico se puede ver esa distribución.

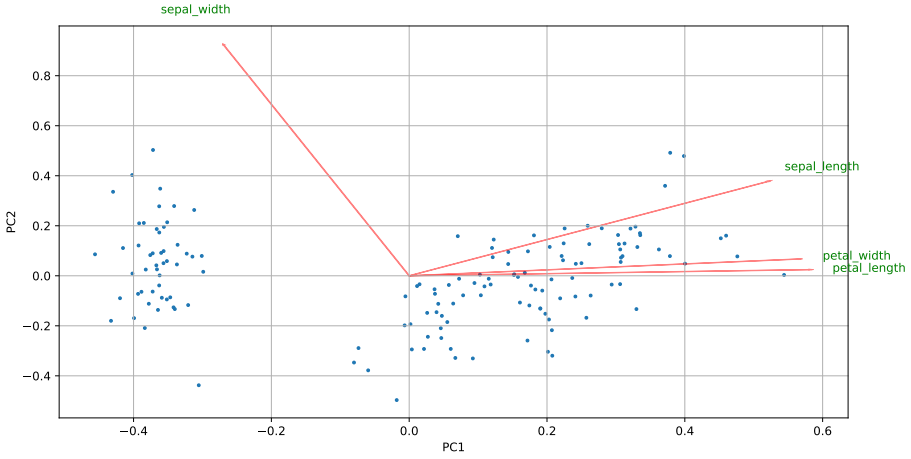
```
def biplot(score,coeff,labels=None):
    xs = score[:,0]
    ys = score[:,1]
    n = coeff.shape[0]
    scalex = 1.0/(xs.max() - xs.min())
    scaley = 1.0/(ys.max() - ys.min())
    plt.scatter(xs * scalex,ys * scaley,s=5)
    for i in range(n):
```

```
        plt.arrow(0, 0, coeff[i,0], coeff[i,1],color = 'r',alpha = 0.5)
        if labels is None:
            plt.text(coeff[i,0]* 1.15, coeff[i,1] * 1.15, "Var"+str(i+1), color = 'green', h
        else:
            plt.text(coeff[i,0]* 1.15, coeff[i,1] * 1.15, labels[i], color = 'g', ha = 'cent

    plt.xlabel("PC{}".format(1))
    plt.ylabel("PC{}".format(2))
    plt.grid()



plt.figure(figsize=(12, 6))
biplot(components, np.transpose(PCA.components_), list(iris.columns))
plt.show()
```



Vemos en el biplot el tamaño de las flechas, es un indicativo de que todas las variables originales tienen cierto peso en los componentes principales.

Además, vemos que hay una correlación positiva entre *sepal_length, petal_width* y *petal_length*, mientras que *sepal_length* y *sepal_width* tienen una correlación pequeña ya que el ángulo que forman las dos flechas es casi recto.

Entre *petal_length* y *sepal_width* hay una correlación negativa, de la misma forma que *petal_width* y *sepal_width*. El tamaño y la dirección en la que apuntan las flechas nos indican el peso (positivo o negativo) y la influencia que tiene cada variable en los dos componentes principales