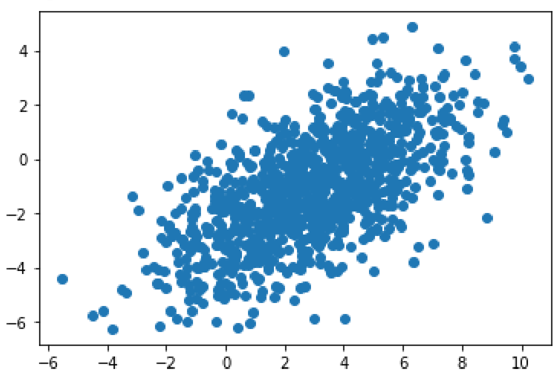
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Part a

```
data=pd.read_csv("/content/Hw9_part1_data.xls")

data=data.drop(data.columns[0], axis=1)


data_n=np.array(data)


plt.scatter(data_n[:,0],data_n[:,1])
```

```
<matplotlib.collections.PathCollection at 0x7f3a79a50be0>
```



Answer

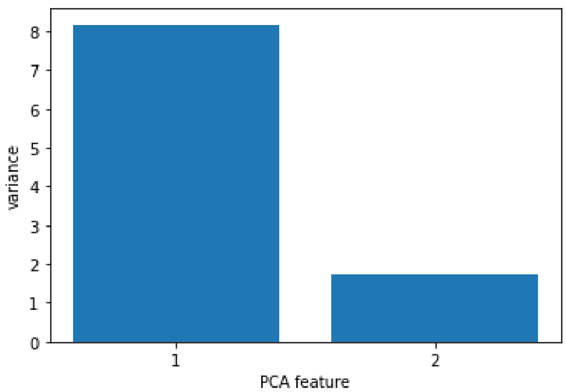
The data is bounded by an elliptical shape. There is more variation along a diagonal axis in the xy plane with positive slope.

Part b

```
from sklearn.decomposition import PCA

pca=PCA()

pca.fit(data)
features=range(1,pca.n_components_+1)

plt.bar(features,pca.explained_variance_)
plt.xticks(features)
plt.ylabel('variance')
plt.xlabel('PCA feature')

x_mean= np.mean(data_n[:,0])
y_mean=np.mean(data_n[:,1])

x_h=np.array([x_mean,y_mean])


u,s,vt=np.linalg.svd(np.cov(np.transpose  (data_n-x_h)))
```

```
print(np.round(vt,2))

print(np.round(s,2))


print("Eigenvectors", np.round(pca.components_,2))
print("Eigenvalues", np.round(pca.explained_variance_,2))
```

```
[[-0.81 -0.58]
 [-0.58  0.81]]
[8.17 1.73]
Eigenvectors [[-0.81 -0.58]
 [-0.58  0.81]]
Eigenvalues [8.17 1.73]
```



Answer

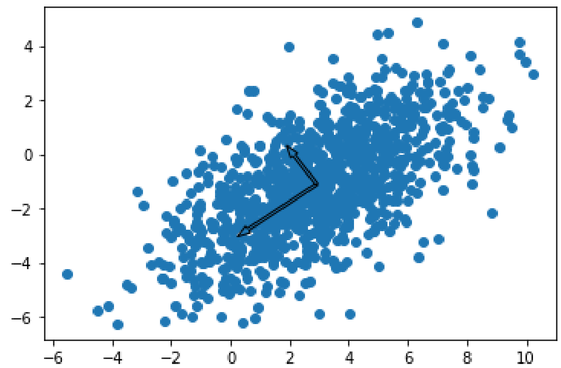
The eigenvalues are $\lambda_1 = 8.17, \lambda_2 = 1.73$ and the eigenvectors $v_1 = [-0.81, -0.58]$ and $v_2 = [-0.58, 0.81]$

Part c

```
x_mean= np.mean(data_n[:,0])
y_mean=np.mean(data_n[:,1])


slambda_1=np.sqrt(pca.explained_variance_[0])
slambda_2=np.sqrt(pca.explained_variance_[1])

dx1=slambda_1*pca.components_[0,0]
dy1=slambda_1*pca.components_[0,1]


dx2=slambda_2*pca.components_[1,0]
dy2=slambda_2*pca.components_[1,1]



plt.scatter(data_n[:,0],data_n[:,1])

print(x_mean)
```

```
print(y_mean)


plt.arrow(x_mean, y_mean, dx1,dy1, width = 0.1)

plt.arrow(x_mean, y_mean,  dx2,dy2, width = 0.1)

plt.show()
```

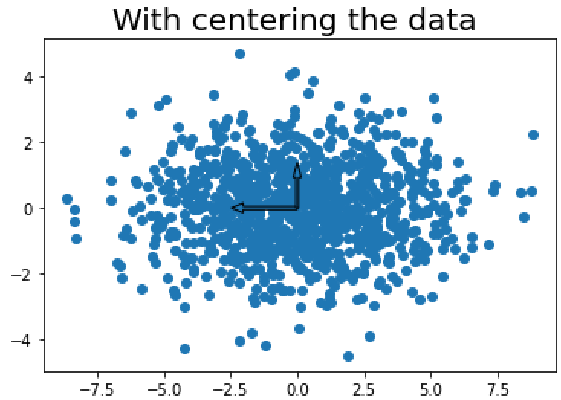
```
2.919892489952931
-1.1182927925208217
```




```
print(pca.components_)




principalComponents = pca.fit_transform(data)



transformed_pca_components=pca.fit_transform(pca.components_)


print("Transformed",transformed_pca_components1)

transformed_pca_componentsx1= transformed_pca_components1[0,0]
transformed_pca_componentsy1= transformed_pca_components1[0,1]
dx1=slambda_1*transformed_pca_componentsx1
dy1=slambda_1*transformed_pca_componentsy1



transformed_pca_componentsx2=transformed_pca_components[1,1]
transformed_pca_componentsy2= transformed_pca_components[1,0]
dx2=slambda_2*transformed_pca_componentsx2
dy2=slambda_2*transformed_pca_componentsy2



fig2 = plt.figure()

plt.scatter(principalComponents[:,0],principalComponents[:,1])

x_mean_pca=np.mean(principalComponents[:,0])
y_mean_pca=np.mean(principalComponents[:,1])

plt.arrow(x_mean_pca, y_mean_pca, dx1,dy1, width = 0.1)

plt.arrow(x_mean_pca, y_mean_pca,  dx2,dy2, width = 0.1)
plt.title(label="With centering the data ",
          fontsize=20,
```

```
            color="black")
plt.show()
```

```
[[-0.81319933 -0.58198527]
 [-0.58198527  0.81319933]]
Transformed [[-0.70710678  0.        ]
 [ 0.70710678  0.        ]]
```



Answer

1) The directions of the eigenvectors are orthogonal to each other.

2) The eigenvector with the greatest eigenvalue points towards the direction with greatest variation in the data, and the second eigenvector points towards the second direction with greatest variation.

3) The two statements mentions, imply that the eigenvectors point towards the directions where there is greatest variation along that axis.

Colab paid products - Cancel contracts here

0s completed at 2:47 PM