Homework 2

STA 307

Malcolm, Vejay, Rohan, Tyler

February 25, 2024

```
# Imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from ISLP import load_data
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
AUTO = load_data("Auto")
```

Question 1

```
# Question1
selected_coloumns = ['mpg','cylinders', 'displacement',  'weight', 'acceleration','horsepower' ]
X = AUTO[selected_coloumns]
y = AUTO['origin'] # Origin of car (1. American, 2. European, 3. Japanese)
print(X.head)
print("American cars: {0}".format((y == 1).sum())) # American
print("European cars: {0}".format((y == 2).sum())) # European
print("Japanese cars: {0}".format((y == 3).sum())) # Japanese
```

```
<bound method NDFrame.head of      mpg  cylinders  displacement  weight  acceleration  horsepow
0    18.0          8         307.0    3504          12.0         130
1    15.0          8         350.0    3693          11.5         165
2    18.0          8         318.0    3436          11.0         150
3    16.0          8         304.0    3433          12.0         150
4    17.0          8         302.0    3449          10.5         140
..    ...        ...           ...     ...           ...         ...
387  27.0          4         140.0    2790          15.6          86
388  44.0          4          97.0    2130          24.6          52
389  32.0          4         135.0    2295          11.6          84
390  28.0          4         120.0    2625          18.6          79
391  31.0          4         119.0    2720          19.4          82

[392 rows x 6 columns]>
American cars: 245
European cars: 68
Japanese cars: 79
```

Question 2

```
# Question2
print("Standard deviation")
print(X.std())
print("\n")
print("Mean")
print(X.mean())
```

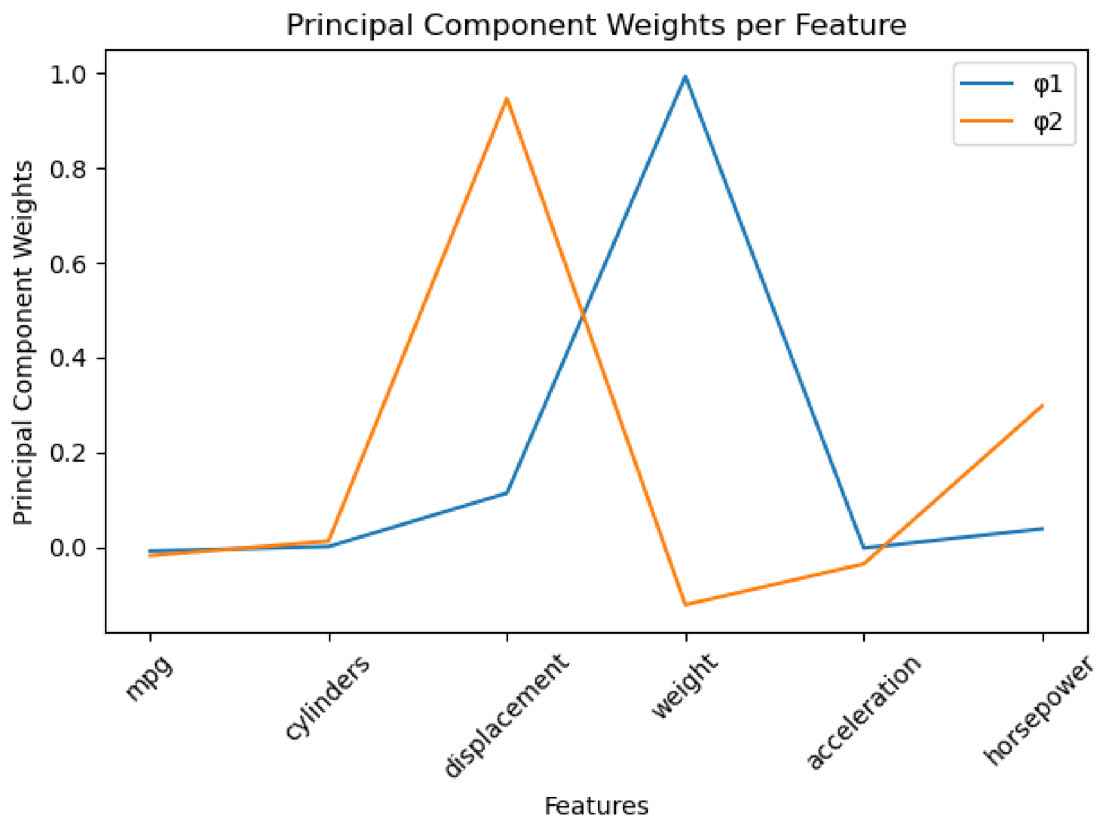
```
Standard deviation
mpg               7.805007
cylinders         1.705783
displacement    104.644004
weight          849.402560
acceleration      2.758864
horsepower       38.491160
dtype: float64


Mean
mpg               23.445918
cylinders          5.471939
displacement     194.411990
weight          2977.584184
acceleration      15.541327
horsepower       104.469388
dtype: float64
```

Qustion 3

a

```
# Question3_parta
pca = PCA(n_components=2)
X_reduced = pca.fit_transform(X)
components = pca.components_
x = np.arange(components.shape[1]) # 6
plt.plot(x, components[0], label='1')
plt.plot(x, components[1], label='2')
plt.xticks(ticks=x, labels=X.columns, rotation=45)
plt.xlabel('Features')
plt.ylabel('Principal Component Weights')
plt.title('Principal Component Weights per Feature')
plt.legend()
plt.tight_layout()
plt.savefig("plot.png", bbox_inches='tight')
plt.close()
```



b

It seems pca is caputring the data with larger scales this can lead to misleading conclusions about the importance of features

Question 4

```
# Question4
means = np.mean(X, axis=0)
stds = np.std(X, axis=0)
Z = (X - means) / stds
means_Z = np.mean(Z, axis=0)
stds_Z = np.std(Z, axis=0)
print(means_Z, stds_Z)
```

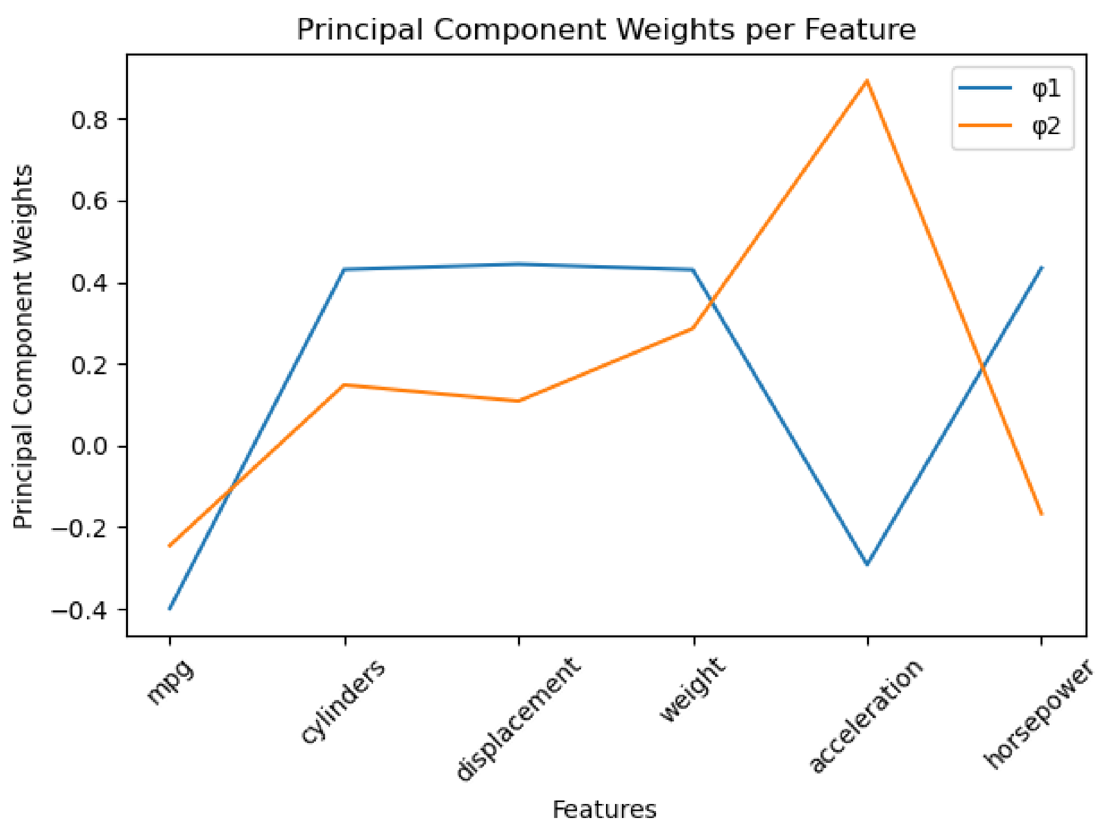
```
mpg              1.450087e-16
cylinders       -1.087565e-16
displacement    -7.250436e-17
weight          -1.812609e-17
acceleration     4.350262e-16
horsepower      -1.812609e-16
dtype: float64 mpg              1.0
cylinders        1.0
displacement     1.0
```

```
weight          1.0
acceleration    1.0
horsepower      1.0
dtype: float64
```

Question 5

a

```
# Question5_parta
pca = PCA(n_components=2)
Z_reduced = pca.fit_transform(Z)
print("Reduced data size:", Z_reduced.shape)
components = pca.components_
plt.plot(x, components[0], label='1')
plt.plot(x, components[1], label='2')
plt.xticks(ticks=x, labels=Z.columns, rotation=45)
plt.xlabel('Features')
plt.ylabel('Principal Component Weights')
plt.title('Principal Component Weights per Feature')
plt.legend()
plt.tight_layout()
plt.savefig("plot-standard.png", bbox_inches='tight')
plt.close()
# plt.show()
```



b

It seems that the three most important features are 'cylinders', 'displacement', and 'weight', as the first principal component ($\phi 1$) weighs these more heavily, with 'displacement' appearing to have the highest contribution to variance by a small margin.

c

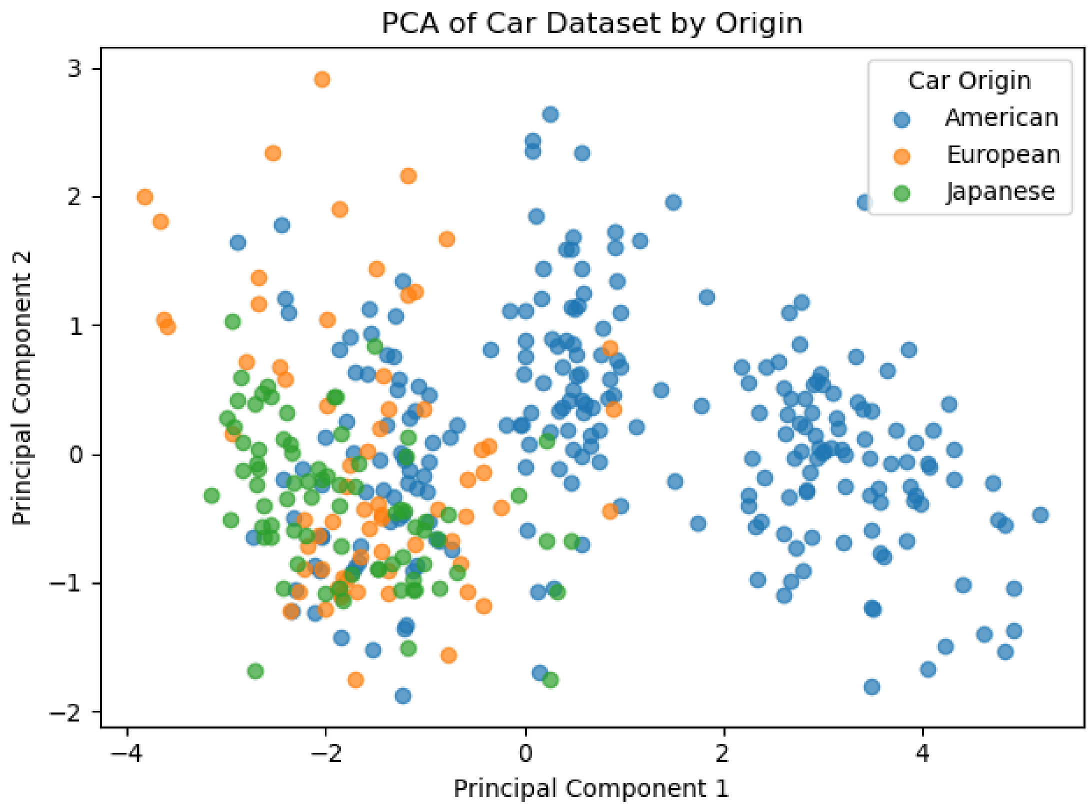
```python
# Question5_partc
dot_product = np.dot(components[0], components[1])
# Calculate the magnitude (norm) of each principal component to check if it's equal to one
magnitude_phi1 = np.linalg.norm(components[0])
magnitude_phi2 = np.linalg.norm(components[1])
print("Dot product", dot_product)
print("Magnitude phi 1",magnitude_phi1)
print("Magnitude phi 2", magnitude_phi2)
```

```
Dot product 0.0
Magnitude phi 1 1.0000000000000002
Magnitude phi 2 1.0
```

Question 6

a

```python
# Question6_parta
origins = ["American", "European", "Japanese"]
# plt.figure(figsize=(8, 6))
for origin in [1, 2, 3]:
    subset = Z_reduced[origin == y]
    plt.scatter(subset[:, 0], subset[:, 1], label=origins[origin - 1], alpha=0.7)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Car Dataset by Origin')
plt.legend(title='Car Origin')
plt.tight_layout()
plt.savefig("plot-scatter.png")
plt.close()
```



b

```python
# Assuming AUTO is a pandas DataFrame containing the dataset

# Calculate mean value of the feature (e.g., 'mpg') for each class
mean_mpg_american = AUTO[AUTO['origin'] == 1]['displacement'].mean()
mean_mpg_european = AUTO[AUTO['origin'] == 2]['displacement'].mean()
mean_mpg_japanese = AUTO[AUTO['origin'] == 3]['displacement'].mean()

# Display the mean values
print("Mean displacement for American cars:", mean_mpg_american)
print("Mean displacement for European cars:", mean_mpg_european)
print("Mean displacement for Japanese cars:", mean_mpg_japanese)

# Determine which class has the largest mean value for the feature
if mean_mpg_american > mean_mpg_european and mean_mpg_american > mean_mpg_japanese:
    print("American cars have the largest mean displacement.")
elif mean_mpg_european > mean_mpg_american and mean_mpg_european > mean_mpg_japanese:
    print("European cars have the largest mean displacement.")
else:
    print("Japanese cars have the largest mean displacement.")
```

```
Mean displacement for American cars: 247.5122448979592
Mean displacement for European cars: 109.63235294117646
Mean displacement for Japanese cars: 102.70886075949367
American cars have the largest mean displacement.
```

c

```
mean_acceleration_american = AUTO[AUTO['origin'] == 1]['acceleration'].mean()
mean_acceleration_european = AUTO[AUTO['origin'] == 2]['acceleration'].mean()
mean_acceleration_japanese = AUTO[AUTO['origin'] == 3]['acceleration'].mean()

print("Mean acceleration for American cars:", mean_acceleration_american)
print("Mean acceleration for European cars:", mean_acceleration_european)
print("Mean acceleration for Japanese cars:", mean_acceleration_japanese)
print("European cars have the largest mean acceleration:", mean_acceleration_european)
```

```
Mean acceleration for American cars: 14.990204081632651
Mean acceleration for European cars: 16.79411764705882
Mean acceleration for Japanese cars: 16.17215189873418
European cars have the largest mean acceleration: 16.79411764705882
```

Question 7

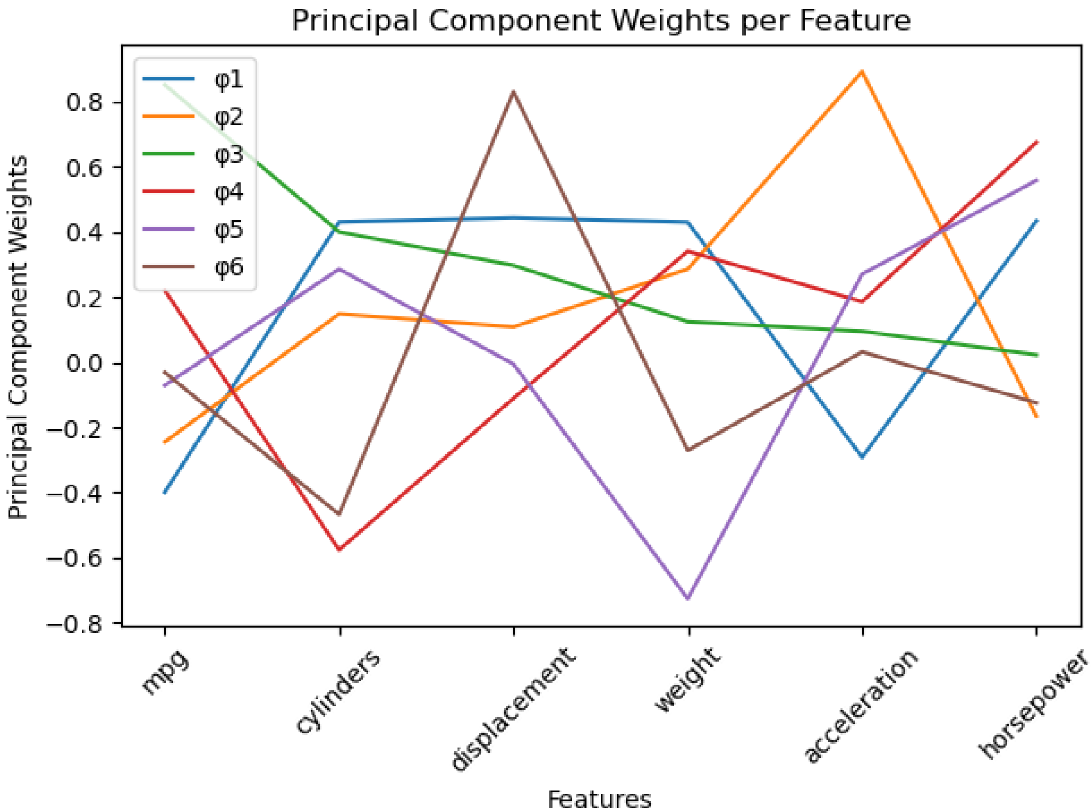
a

plot-pca-6

```

pca = PCA(n_components=6)
Z_reduced = pca.fit_transform(Z)
print("here")
components = pca.components_
x = np.arange(components.shape[1]) # 6
print("here")

plt.plot(x, components[0], label='1')
plt.plot(x, components[1], label='2')
plt.plot(x, components[2], label='3')
plt.plot(x, components[3], label='4')
plt.plot(x, components[4], label='5')
plt.plot(x, components[5], label='6')
plt.xticks(ticks=x, labels=Z.columns, rotation=45)
plt.xlabel('Features')
plt.ylabel('Principal Component Weights')
plt.title('Principal Component Weights per Feature')
plt.legend()
plt.tight_layout()
plt.savefig("plot-pca-6.png", bbox_inches='tight')
plt.close()
```

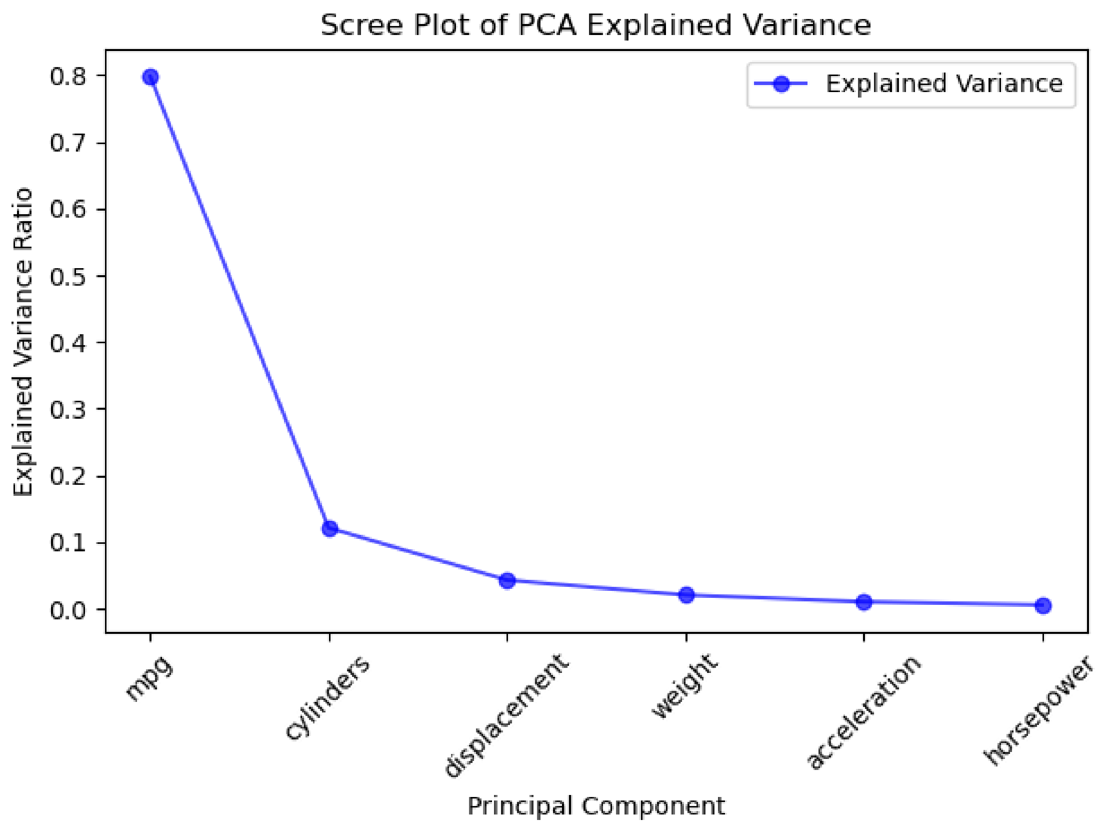


b

```
# Question 7
Z = Z.copy()
pca = PCA(n_components=6)
pca.fit(Z)

# Extract explained variance ratio
explained_variance_ratio = pca.explained_variance_ratio_

# sets up and outputs the plot.
plt.plot(range(6), explained_variance_ratio, alpha=0.7, color='blue', label='Explained
↪  Variance',marker='o')
plt.xlabel('Principal Component')
plt.ylabel('Explained Variance Ratio')
plt.title('Scree Plot of PCA Explained Variance')
plt.xticks(ticks=x, labels=Z.columns, rotation=45)
plt.legend()
plt.tight_layout()
plt.savefig("plot-scree.png", bbox_inches='tight')
plt.close()
```


Scree Plot of PCA Explained Variance
Explained Variance
Explained Variance Ratio
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
mpg
cylinders
displacement
weight
acceleration
horsepower
Principal Component