Q1a)

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
```

When loading this dataset we can see that the categorical features are the alcohol, malic_acid, ash, alcalinity_of_ash, magnesium, total_phenols, flavonoids, nonflavanoid_phenols, proanthocyanins, color_intensity, hue, od280/od315_of_diluted_wines and proline
. We were able to derive this result by firstly importing the pandas library to load the dataset, we load the load_wine() method into the data variable and then set the data frame to max so that we can see every value in the dataset and then it is converted to a column row format so that we can view it easily because it is unordered and unstructured data originally.

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
```

values_counts is used to find the frequency of the values

Q1b)

UNIVARIATE

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
```

```
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['alcohol'].mean()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['alcohol'].median()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['malic_acid'].mean()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['malic_acid'].median()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['ash'].mean()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['ash'].median()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['alcalinity_of_ash'].mean()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
```

```
print(data.head())
print(data.value_counts())
data['alcalinity_of_ash'].median()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['magnesium'].mean()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['magnesium'].median()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['total_phenols'].mean()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
```

```
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['total_phenols'].median()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['flavanoids'].mean()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['flavanoids'].median()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data[' nonflavanoid_phenols'].mean()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data[' nonflavanoid_phenols'].median()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['proanthocyanins'].mean()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['proanthocyanins'].median()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
```

```
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['color_intensity'].mean()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['color_intensity'].median()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['hue'].mean()

import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['hue'].median()

import pandas as pd
from sklearn.datasets import load_wine
```

```
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data[' od280/od315_of_diluted_wines'].mean()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data[' od280/od315_of_diluted_wines’].median()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
data['proline'].mean()
```

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
```

data['proline’].median()

The mean and median give the mean and median values of the given variables.

MULTIVARIATE

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.pairplot(df, hue = 'target', vars = ['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash',
'magnesium', 'total_phenols', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins',
'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline'])
plt.show()
```

pairplot() function creates a complete grid such that each variable that is passed as an argument to the function will be shown as the x and y axis. The hue parameter allows us to plot the data of each variable onto the graph.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.distplot(df['alcohol'], bins=None, kde=False)
```

```
plt.show()

sns.distplot(df['malic_acid'], bins=None, kde=False)
plt.show()

sns.distplot(df['ash'], bins=None, kde=False)
plt.show()

sns.distplot(df['alcalinity_of_ash'], bins=None, kde=False)
plt.show()

sns.distplot(df['magnesium'], bins=None, kde=False)
plt.show()

sns.distplot(df['total_phenols'], bins=None, kde=False)
plt.show()

sns.distplot(df['flavanoids'], bins=None, kde=False)
plt.show()

sns.distplot(df['nonflavanoid_phenols'], bins=None, kde=False)
plt.show()

sns.distplot(df['proanthocyanins'], bins=None, kde=False)
plt.show()

sns.distplot(df['color_intensity'], bins=None, kde=False)
plt.show()

sns.distplot(df['hue'], bins=None, kde=False)
plt.show()

sns.distplot(df['od280/od315_of_diluted_wines'], bins=None, kde=False)
plt.show()

sns.distplot(df['proline'], bins=None, kde=False)
plt.show()
```

distplot() function allows us to create a histogram as another means to visualise the data.

1c)

```
import pandas as pd
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())
display(df.groupby('alcohol').median())
display(df.groupby('malic_acid').median())
display(df.groupby('ash').median())
display(df.groupby('alcalinity_of_ash').median())
display(df.groupby('magnesium').median())
display(df.groupby('total_phenols').median())
display(df.groupby('flavanoids').median())
display(df.groupby('nonflavanoid_phenols').median())
display(df.groupby('proanthocyanins').median())
display(df.groupby('color_intensity').median())
display(df.groupby('hue').median())
display(df.groupby('od280/od315_of_diluted_wines').median())
display(df.groupby('proline').median())
```

I used the groupby() function to categorise the data and the median() function to calculate the median.

1d)

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
```

```
print(data.head())
print(data.value_counts())

correlation = df.corr()
round(correlation,2)

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="alcohol", y="alcohol", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="malic_acid", y="malic_acid", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
```

```
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="ash", y="ash", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="alcalinity_of_ash", y="alcalinity_of_ash", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="magnesium", y="magnesium", data=df);
plt.show()
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="total_phenols", y="total_phenols", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="flavanoids", y="flavanoids", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
```

```
print(data.head())
print(data.value_counts())

sns.lmplot(x="nonflavanoid_phenols", y="nonflavanoid_phenols", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="proanthocyanins", y="proanthocyanins", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="color_intensity", y="color_intensity", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
```

```
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="hue", y="hue", data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="od280/od315_of_diluted_wines", y="od280/od315_of_diluted_wines",
data=df);
plt.show()

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())
print(data.value_counts())

sns.lmplot(x="proline", y="proline", data=df);
```

```
plt.show()
```

I used the corr() method to calculate the correlation coefficients amongst all the variables and by looking at those values I could judge which have the highest correlation, by using the lmplot() method it allowed me to
not only construct a scatter plot but also draw the line of best fit to correctly visualise the correlation.

Q2.

```
import pandas as pd

df = pd.read_csv('country-income.csv')
print(df)

mv=df['Age'].mean()
df['Age'].fillna(value=mv, inplace=True)
print(df)

mv=df['Income'].mean()
df['Income'].fillna(value=mv, inplace=True)
print(df)
```

We are first loading the CSV file into the library by reading it and then creating an ‘mv’ variable that will find the mean of the age column and then within the ‘Age’ column, by calling the fillna() method it will replace our NaN value with the mean value we calculated, the inplace is set to True so that it will replace it in this data frame.

```
import pandas as pd

df = pd.read_csv('country-income.csv')
print(df)

mv=df['Age'].mean()
df['Age'].fillna(value=mv, inplace=True)
print(df)
```

```
mv=df['Income'].mean()
df['Income'].fillna(value=mv, inplace=True)
print(df)

dfnew = df.rename({'Region': '0', 'Age': '1', 'Income': '2', 'Online Shopper': '3'}, axis=1)
print(dfnew)
```

To rename the categorical labels and replace them with numerical labels I used the rename() function which would rename the columns or rows and by following the documentation I specified what I want to replace the original name with and I set the axis to 1 as the 1 indicates the columns, I created a new copy of this data frame to do this.

Q3.

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

df = pd.read_csv('shoesize.csv')
print(df)

group1 = df.groupby(df.Gender)
dfnew1 = group1.get_group("F")
dfnew1

sns.scatterplot(x="Size", y="Height", data=dfnew1);
plt.show()

group2 = df.groupby(df.Gender)
dfnew2 = group2.get_group("M")
dfnew2

sns.scatterplot(x="Size", y="Height", data=dfnew2);
plt.show()
```

————————————————————————————————

```
import pandas as pd
```

```
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats

df = pd.read_csv('shoesize.csv')
print(df)

group1 = df.groupby(df.Gender)
dfnew1 = group1.get_group("F")
dfnew1

sns.scatterplot(x="Size", y="Height", data=dfnew1);
plt.show()

group2 = df.groupby(df.Gender)
dfnew2 = group2.get_group("M")
dfnew2

sns.scatterplot(x="Size", y="Height", data=dfnew2);
plt.show()

stats.pearsonr(dfnew1['Size'], dfnew1['Height'])
```

————————————————————————————————————

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

df = pd.read_csv('shoesize.csv')
print(df)

group1 = df.groupby(df.Gender)
dfnew1 = group1.get_group("F")
dfnew1

sns.scatterplot(x="Size", y="Height", data=dfnew1);
plt.show()
```

```
group2 = df.groupby(df.Gender)
dfnew2 = group2.get_group("M")
dfnew2

sns.scatterplot(x="Size", y="Height", data=dfnew2);
plt.show()

stats.pearsonr(dfnew2['Size'], dfnew2['Height'])
```

I first used the groupby() function to segregate the 2 categories for gender and created 2 new separate data frames with the separated values and then used seaboard to create the scatterplots. I imported the scipy library to use the pearsonr() function to derive the coefficients. By analysing the scatterplots and the coefficients we can infer that both sets of data have high correlations since both values are an approximate 0.7 and the scatterplot is going up visually.

Q4.

```
import pandas as pd
from sklearn.datasets import load_wine
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import seaborn as sns
import matplotlib.pyplot as plt

data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)
pd.set_option("display.max_rows", None, "display.max_columns", None)
data = pd.DataFrame(data=data['data'],columns=data['feature_names'])
print(data.head())

categories = ['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',
'total_phenols', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins',
'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
x = df.loc[:, categories].values
y = df.loc[:,['target']].values
x = StandardScaler().fit_transform(x)
print(x)
```

```
pca1 = PCA(n_components=2)
newpca = pca1.fit_transform(x)
pdf = pd.DataFrame(data = newpca, columns = ['component 1', 'component 2'])
print(pdf)

final = pd.concat([pdf, df[['target']]], axis = 1)
print(final)

sns.pairplot(final, hue='target', diag_kind='hist')
plt.show()
```

To perform the first step of PCA I first standardised the data in order to scale the data since that is the whole purpose of performing PCA by making the mean = 0 and variance = 1 so that they are all the same size. I used the StandardScaler package to do this. I used df.loc() to gain access to the 2 components including the new values and the target. In order to see our output visually I reduced the dimensions to 2 columns and then combining that with the target column. I then used the pairplot() function to plot the final scatterplot. The first principal component represents a mildly high correlation whereas the second one indicates a mixed one. The classes do seem quite distinctive from each other and it is easy to distinguish the classes.

5)

$x_1 = (1, 2)$ $y_1 = 1$

$x_2 = (-1, 0)$ $y_2 = 0$

$x_3 = (3, 2)$ 2

$x_4 = (0, 1)$ 1.41

~~$\sqrt{(3-(-1))^2 + (2-0)^2}$~~

~~$= 16 + 4 = 20$~~

~~$\sqrt{20}$~~

$\sqrt{((-1)-(1))^2 + (0-2)^2}$

$4 + 4 = 8 = \sqrt{8}$ $x_1 \rightarrow x_2$

$\| 2.83$

$\sqrt{(3-1)^2 + (2-2)^2}$

$2^2 + 0^2 = \sqrt{4}$ $\sqrt{4}$ $x_1 \rightarrow x_3$

$\| 2$

$\sqrt{(0-1)^2 + (1-2)^2}$

$1 + 1 = \sqrt{2}$ $x_1 \rightarrow x_4$

$\sqrt{(-1-1)^2 + (0-2)^2}$ $\| 1.41$

$4 + 4 = 8$

x_3's computed distance is 2 which is closer to the y_1 class therefore it would have the same observation as x_1 and x_4 could belong to the ~~y_2~~ y_1 class as 1.41 is closer to $y_1 = 1$ as well