DATA HANDLING IN PYTHON

CODE

```
import pandas as pd
```

```
data=pd.read_csv("auto-mpg.csv")
```

```
type(data)
```

OUTPUT

```
pandas.core.frame.DataFrame
def __init__(data=None, index: Axes | None=None, columns: Axes | None=None,
dtype: Dtype | None=None, copy: bool | None=None) -> None

Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Data structure also contains labeled axes (rows and columns).
Arithmetic operations align on both row and column labels. Can be
thought of as a dict-like container for Series objects. The primary
```

```
data.shape
```

```
(398, 9)
```

```
nrow_count=data.shape[0]
print(nrow_count)
```

```
398
```

```
ncol_count=data.shape[1]
print(ncol_count)
```

```
9
```

```
data.columns
```

```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model year', 'origin', 'car name'],
      dtype='object')
```

```
data.columns=['miles_per_gallon', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model year', 'origin', 'car name']
```

```
data.columns
```

```
Index(['miles_per_gallon', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model year', 'origin', 'car name'],
      dtype='object')
```

CODE

```
data.rename(columns={'displacement':'disp'},inplace=True)
```

```
data.head()
```

OUTPUT

	miles_per_gallon	cylinders	disp	horsepower	weight	acceleration	model year	origin
0	18.0	8	307.0	130	3504	12.0	70	1
1	15.0	8	350.0	165	3693	11.5	70	1

```
data.head(3)
```

	miles_per_gallon	cylinders	disp	horsepower	weight	acceleration	model year	origin
0	18.0	8	307.0	130	3504	12.0	70	1

```
data.tail()
```

	miles_per_gallon	cylinders	disp	horsepower	weight	acceleration	model year	orig
393	27.0	4	140.0	86	2790	15.6	82	
394	44.0	4	97.0	52	2130	24.6	82	

```
data.tail(3)
```

	miles_per_gallon	cylinders	disp	horsepower	weight	acceleration	model year	orig
395	32.0	4	135.0	84	2295	11.6	82	

CODE

```
data.at[200,'cylinders']
```

```
6
```

```
#data.get_value(200,'cylinders')
```

```
data_cyl=data.loc[:,'car name']
```

```
data_cyl.head()
```

OUTPUT

```
0    chevrolet chevelle malibu
1            buick skylark 320
2           plymouth satellite
3                amc rebel sst
4                  ford torino
Name: car name, dtype: object
```

CODE

```
import numpy as np
```

```
var1=[np.nan,np.nan,np.nan,10.1,12,123.14,0.121]
var2=[40.2,11.78,7801,0.25,34.2,np.nan,np.nan]
var3=[1234,np.nan,34.5,np.nan,78.25,14.5,np.nan]
df=pd.DataFrame({'Attr_1':var1,'Attr_2':var2,'Attr_3':var3})
print(df)
```

OUTPUT

```
    Attr_1   Attr_2   Attr_3
0      NaN    40.20  1234.00
1      NaN    11.78      NaN
2      NaN  7801.00    34.50
3   10.100     0.25      NaN
4   12.000    34.20    78.25
5  123.140      NaN    14.50
6    0.121      NaN      NaN
```

CODE

```
miss_val=df[df['Attr_1'].isnull()]
print(miss_val)
```

OUTPUT

```
   Attr_1   Attr_2  Attr_3
0     NaN    40.20  1234.0
1     NaN    11.78     NaN
2     NaN  7801.00    34.5
```

```
np.mean(data[['miles_per_gallon']])
```

```
23.514572864321607
```

```
np.median(data[['miles_per_gallon']])
```

```
23.0
```

```
np.var(data[['miles_per_gallon']])
```

```
miles_per_gallon    60.936119
dtype: float64
```

```
np.std(data[['miles_per_gallon']])
```

```
miles_per_gallon    7.806159
dtype: float64
```

IRIS DATASET

CODE

```
from sklearn import datasets
```

```
iris=datasets.load_iris()
```

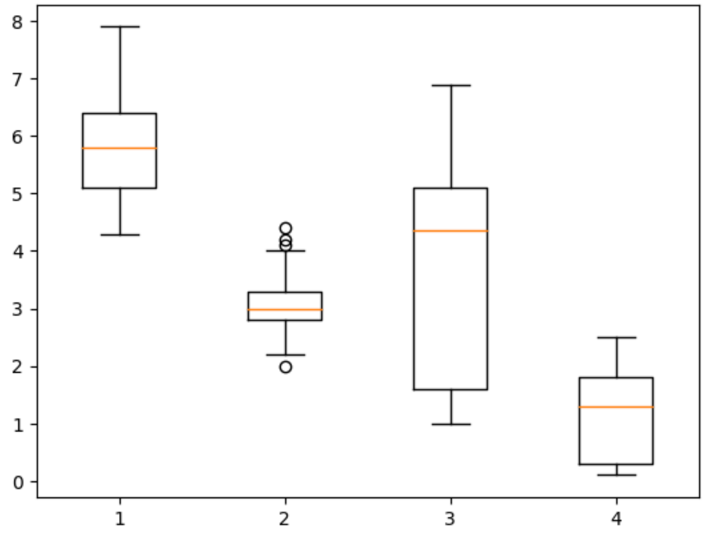
```
import matplotlib.pyplot as plt
```

```
X=iris.data[:,:4]
```

```
plt.boxplot(X)
```

OUTPUT

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f16ee3a3910>,
  <matplotlib.lines.Line2D at 0x7f16ee3a1c90>,
  <matplotlib.lines.Line2D at 0x7f16ee3a2da0>,
  <matplotlib.lines.Line2D at 0x7f16ee3a2f50>,
  <matplotlib.lines.Line2D at 0x7f16ee7a49a0>,
  <matplotlib.lines.Line2D at 0x7f16edf91600>,
  <matplotlib.lines.Line2D at 0x7f16edf91450>,
  <matplotlib.lines.Line2D at 0x7f16edf92a70>],
 'caps': [<matplotlib.lines.Line2D at 0x7f16ee3a3730>,
  <matplotlib.lines.Line2D at 0x7f16ee3a26e0>,
  <matplotlib.lines.Line2D at 0x7f16ee3a3130>,
  <matplotlib.lines.Line2D at 0x7f16ee3a3d00>,
  <matplotlib.lines.Line2D at 0x7f16edf902e0>,
  <matplotlib.lines.Line2D at 0x7f16edf92b00>,
  <matplotlib.lines.Line2D at 0x7f16edf91060>,
  <matplotlib.lines.Line2D at 0x7f16edf929b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f16ee3a3190>,
  <matplotlib.lines.Line2D at 0x7f16ee3a3b20>,
  <matplotlib.lines.Line2D at 0x7f16ee7a47f0>,
  <matplotlib.lines.Line2D at 0x7f16edf92110>],
 'medians': [<matplotlib.lines.Line2D at 0x7f16ee3a12a0>,
  <matplotlib.lines.Line2D at 0x7f16ee3a23b0>,
  <matplotlib.lines.Line2D at 0x7f16edf923e0>,
  <matplotlib.lines.Line2D at 0x7f16edf93880>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f16ee3a3340>,
  <matplotlib.lines.Line2D at 0x7f16ee7a4dc0>,
  <matplotlib.lines.Line2D at 0x7f16edf900d0>,
  <matplotlib.lines.Line2D at 0x7f16edf93220>],
 'means': []}
```

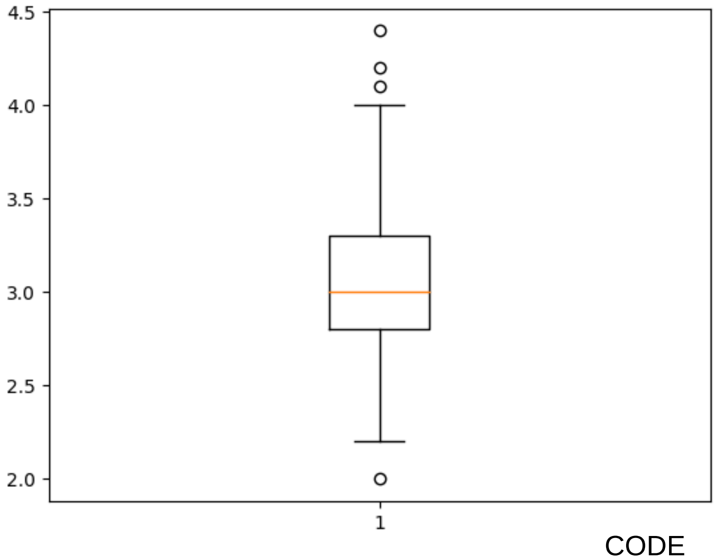


CODE

```
plt.show()
```

```
plt.boxplot(X[:,1])
plt.show()
```

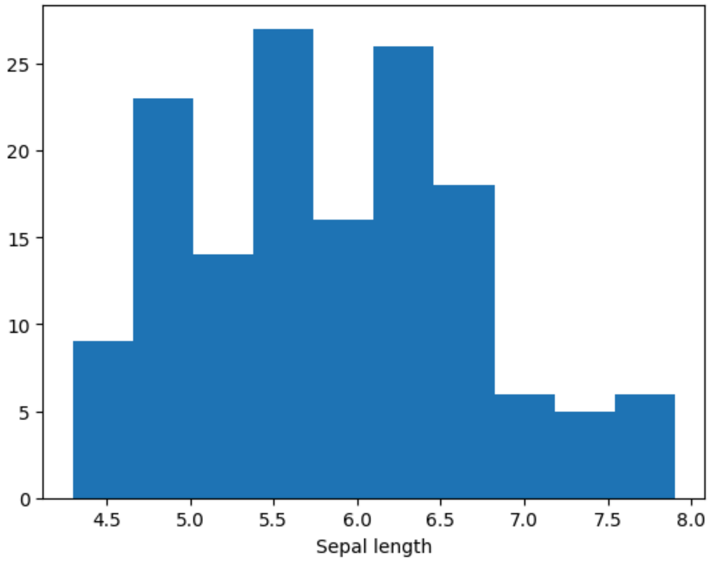
OUTPUT



CODE

```
X=iris.data[:,:1]
plt.hist(X)
plt.xlabel('Sepal length')
plt.show()
```

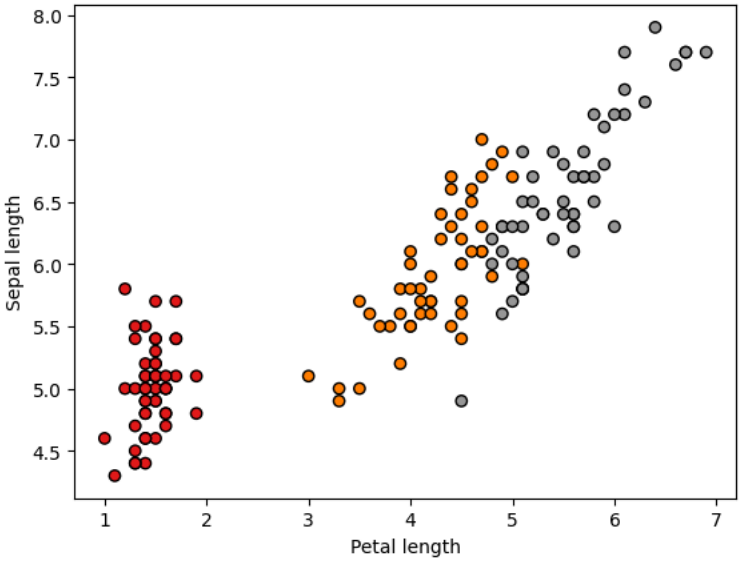
OUTPUT



CODE

```
X=iris.data[:,:4]
y=iris.target
plt.scatter(X[:,2],X[:,0],c=y,cmap=plt.cm.Set1,edgecolor='k')
plt.xlabel('Petal length')
plt.ylabel('Sepal length')
plt.show()
```

OUTPUT



DATA PRE-PROCESSING

CODE

```
df=pd.read_csv('auto-mpg.csv')
```

```
miss_val=df[df['horsepower'].isnull()]
print(miss_val)
```

OUTPUT

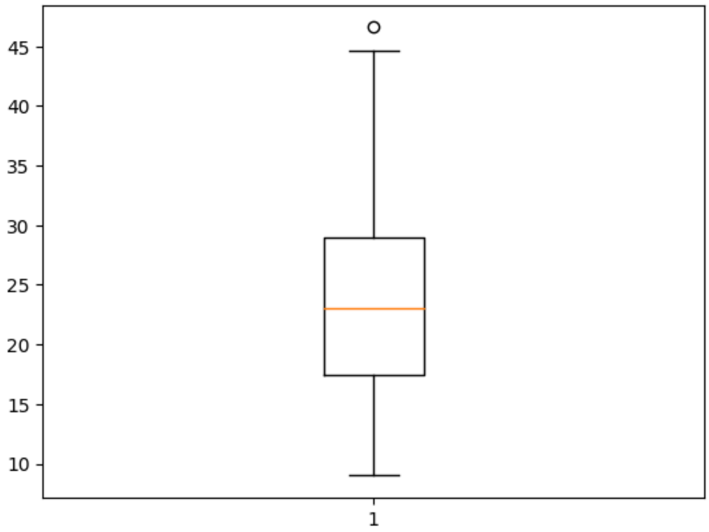
```
Empty DataFrame
Columns: [mpg, cylinders, displacement, horsepower, weight, acceleration, model year, origin, car name]
Index: []
```

Finding Outliers(Option 1):

CODE

```
X=data['miles_per_gallon']
plt.boxplot(X)
plt.show()
```

OUTPUT

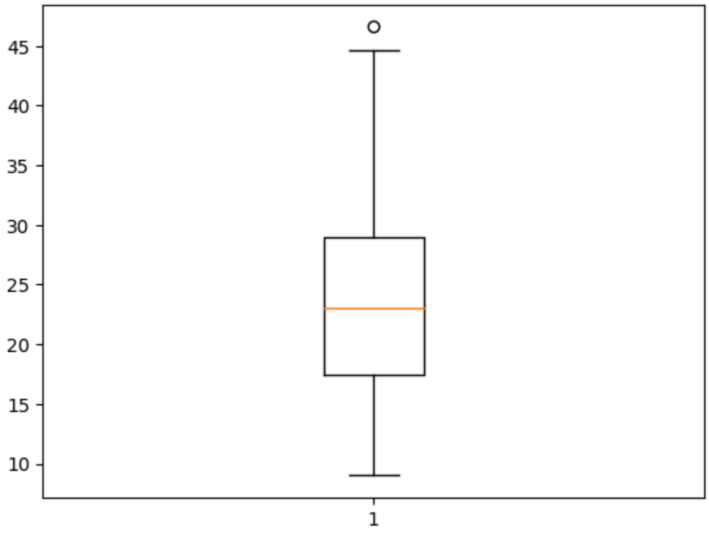


CODE

```
outliers=plt.boxplot(X[:,])['fliers'][0].get_data([1])
outliers
```

OUTPUT

(array([1.]), array([46.6]))



Finding Outliers(Option 2):

CODE

```
def find_outlier(ds, col):
  quart1 = ds[col].quantile(0.25)
  quart3 = ds[col].quantile(0.75)
  IQR = quart3 - quart1 #Inter-quartile range
  low_val = quart1 - 1.5*IQR
  high_val = quart3 + 1.5*IQR
  ds = ds.loc[(ds[col] < low_val) | (ds[col] > high_val)]
  return ds
```

```
outliers=find_outlier(data,'miles_per_gallon')
```

```
outliers
```

OUTPUT

miles_per_gallon	cylinders	disp	horsepower	weight	acceleration	model year	orig

Removing records with missing values / outliers:

CODE

```
data.dropna(axis=0, how='any')
```

OUTPUT

	miles_per_gallon	cylinders	disp	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86	2790	15.6	82	1	ford mustang gl
394	44.0	4	97.0	52	2130	24.6	82	2	vw pickup
395	32.0	4	135.0	84	2295	11.6	82	1	dodge rampage
396	28.0	4	120.0	79	2625	18.6	82	1	ford ranger
397	31.0	4	119.0	82	2720	19.4	82	1	chevy s-10

398 rows × 9 columns

CODE

```
def remove_outlier(ds, col):
  quart1 = ds[col].quantile(0.25)
  quart3 = ds[col].quantile(0.75)
  IQR = quart3 - quart1 #Interquartile range
  low_val = quart1 - 1.5*IQR
  high_val = quart3 + 1.5*IQR
  df_out = ds.loc[(ds[col] > low_val) & (ds[col] < high_val)]
```

```
  return df_out
```

```
data=remove_outlier(data,'miles_per_gallon')
```

Inputing Standard Values

CODE

```
hp_mean = np.mean(data['horsepower'])
inputedrows = data[data['horsepower'].isnull()]
inputedrows = inputedrows.replace(np.nan, hp_mean)
missval_removed_rows = data.dropna(subset=['horsepower'])
data_mod = pd.concat([missval_removed_rows,inputedrows],ignore_index=True)
data_mod
```

OUTPUT

	miles_per_gallon	cylinders	disp	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130.000000	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.000000	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.000000	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.000000	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.000000	3449	10.5	70	1	ford torino
...	...	...	...	...	...	...	...	...	...
392	21.0	6	200.0	104.570332	2875	17.0	74	1	ford maverick
393	40.9	4	85.0	104.570332	1835	17.3	80	2	renault lecar deluxe
394	23.6	4	140.0	104.570332	2905	14.3	80	1	ford mustang cobra
395	34.5	4	100.0	104.570332	2320	15.8	81	2	renault 18i
396	23.0	4	151.0	104.570332	3035	20.5	82	1	amc concord dl

397 rows × 9 columns