Homework 1

2.4.1

(a): The sample size n is extremely large, and the number of predictors p is small.

Answer: it reduce overfitting and large n give a lot of information which if flexable methode

(b): The number of predictors p is extremely large, and the number of observations n is small.

Answer: it has big chance to have a overfitting and have ti use inflexible methode

(c): The relationship between the predictors and response is highly non-linear.

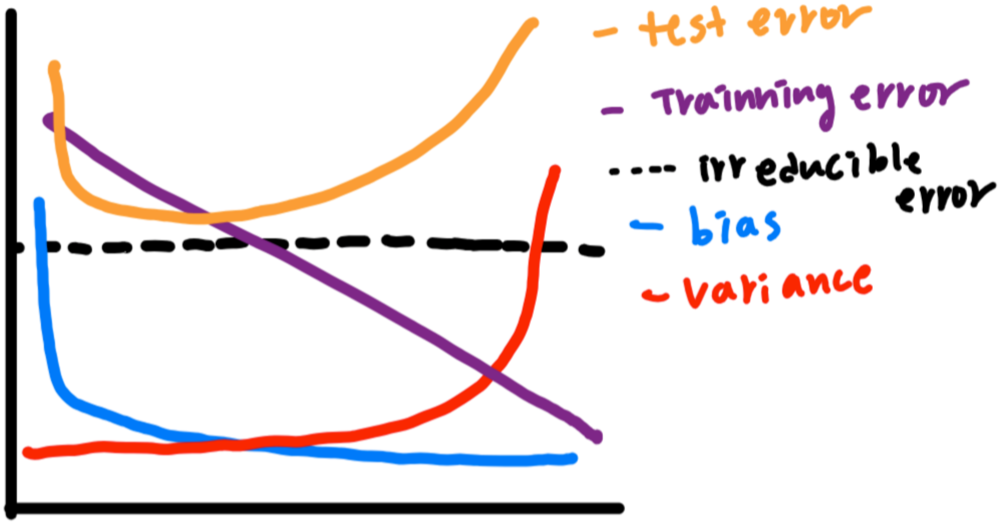
Answer: it is hard for non linear relationship and its better for flexible methode.

(d): The variance of the error terms, i.e. σ2 = Var(e), is extremely high.

Answer: a lot of noise and error which means inflexible method will overfit better

2.4.3

(a):



(b)

Bias: error from simplifing the data

variance: prediction with differnt dat and it increases more with flexiable data

trainign error: models fit to trainging data

test error: u shape curve due to bias var trade off

Irreducible Error: stable with the model

2.4.7 (a),(c)

In [1]:

```
import pandas as pd
import numpy as np

data = {'observation': [1, 2, 3, 4, 5, 6],'X1': [0, 2, 0, 0, 1, 1],'X2': [3, 0, 1, 1

euclidean_distance = [np.sqrt(3**2),np.sqrt(2**2),np.sqrt(1**2 + 3**2),np.sqrt(1**

data['Euclidean_distance'] = euclidean_distance

pd.DataFrame(data)
```

Out[1]:

	observation	X1	X2	X3	Y	Euclidean_distance
0	1	0	3	0	Red	3.000000
1	2	2	0	0	Red	2.000000
2	3	0	1	3	Red	3.162278
3	4	0	1	2	Green	2.236068
4	5	1	0	1	Green	1.414214
5	6	1	1	1	Red	1.732051

(c) : red because modt points includes red

In [10]:

```
# 8 A
df = '/Users/siony/OneDrive/바탕 화면/MSU_SS_24/CMSE 381/CMSE381SS24/DataSets/Colleg
college = pd.read_csv(df)
college
```

Out[10]:

	Unnamed: 0	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergra
0	Abilene Christian University	Yes	1660	1232	721	23	52	2885	53
1	Adelphi University	Yes	2186	1924	512	16	29	2683	122
2	Adrian College	Yes	1428	1097	336	22	50	1036	9
3	Agnes Scott College	Yes	417	349	137	60	89	510	6
4	Alaska Pacific University	Yes	193	146	55	16	44	249	86
...	...	...	...	...	...	...	...	...	.
772	Worcester State College	No	2197	1515	543	4	26	3089	202
773	Xavier University	Yes	1959	1805	695	24	47	2849	110
774	Xavier University of Louisiana	Yes	2097	1915	695	34	61	2793	16
775	Yale University	Yes	10705	2453	1317	95	99	5217	8
776	York College of Pennsylvania	Yes	2989	1855	691	28	63	2988	172

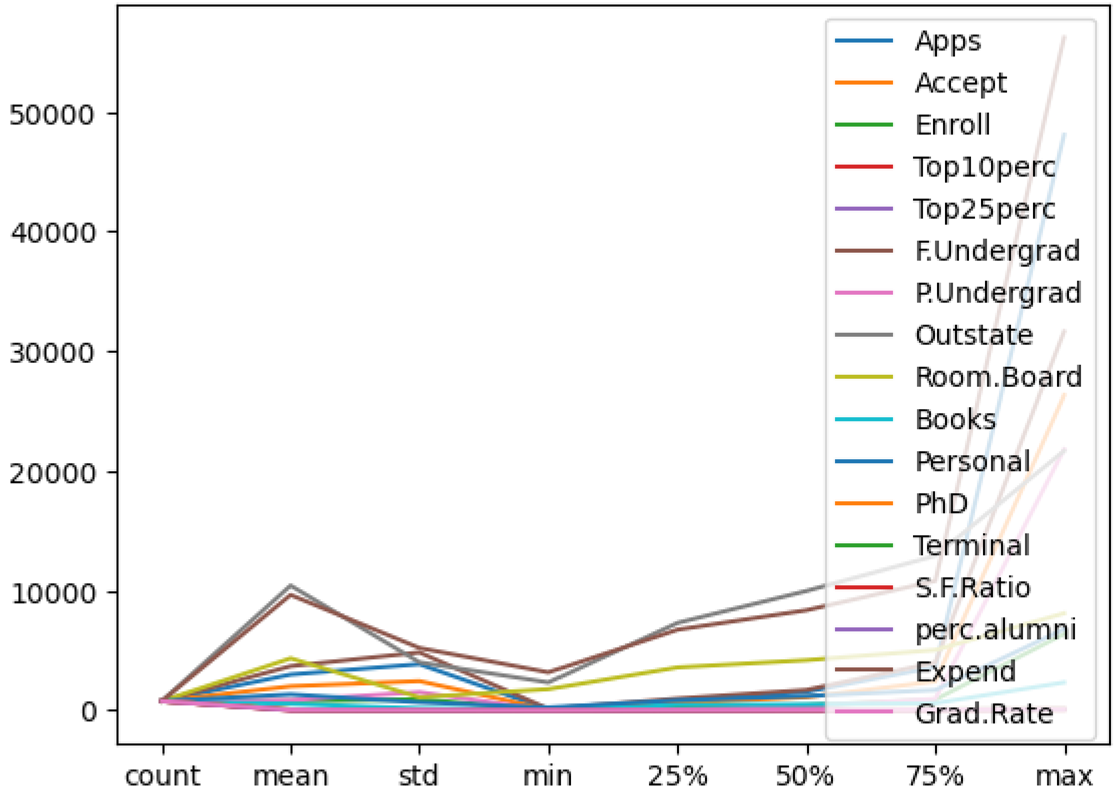
777 rows × 19 columns

In [12]:

```
college2 = pd.read_csv(df, index_col=0)
college3 = college.rename({'Unnamed: 0': 'College'},axis=1)
college3 = college3.set_index('College')
college = college3
```

In [19]:

```
import matplotlib.pyplot as plt
summary = college.describe()
summary.plot()
plt.show()
```



In [21]:

```
numerical_summary = college.describe()
numerical_summary
```

Out[21]:

	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergra
count	777.000000	777.000000	777.000000	777.000000	777.000000	777.000000	777.00000
mean	3001.638353	2018.804376	779.972973	27.558559	55.796654	3699.907336	855.29858
std	3870.201484	2451.113971	929.176190	17.640364	19.804778	4850.420531	1522.43188
min	81.000000	72.000000	35.000000	1.000000	9.000000	139.000000	1.00000
25%	776.000000	604.000000	242.000000	15.000000	41.000000	992.000000	95.00000
50%	1558.000000	1110.000000	434.000000	23.000000	54.000000	1707.000000	353.00000
75%	3624.000000	2424.000000	902.000000	35.000000	69.000000	4005.000000	967.00000
max	48094.000000	26330.000000	6392.000000	96.000000	100.000000	31643.000000	21836.00000

In [22]:

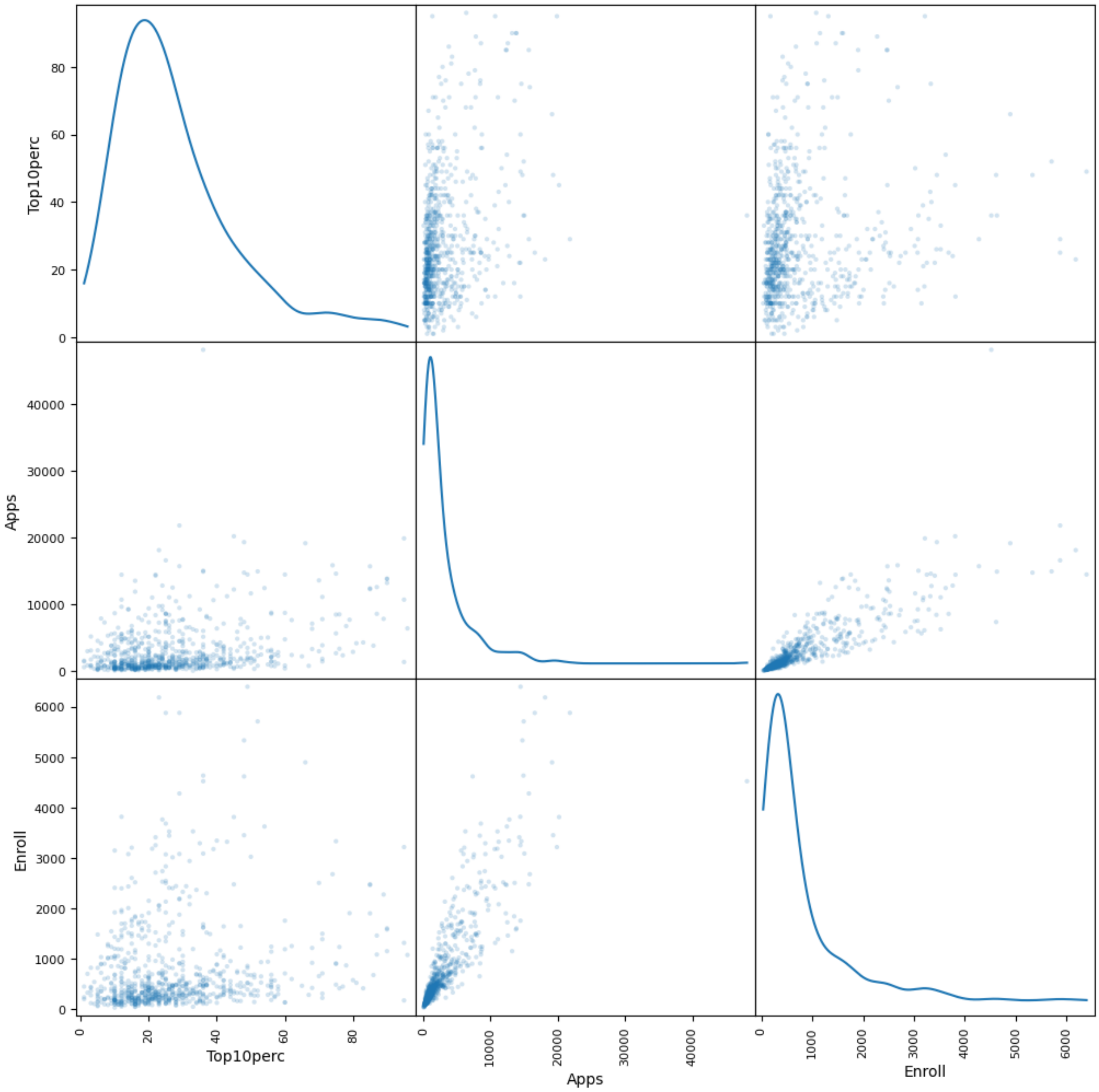
```
columns_for_scatter = ['Top10perc', 'Apps', 'Enroll']
scatter_data = college[columns_for_scatter]
```

In [23]:

```
pd.plotting.scatter_matrix(scatter_data, alpha=0.2, figsize=(12, 12), diagonal='kde'
```

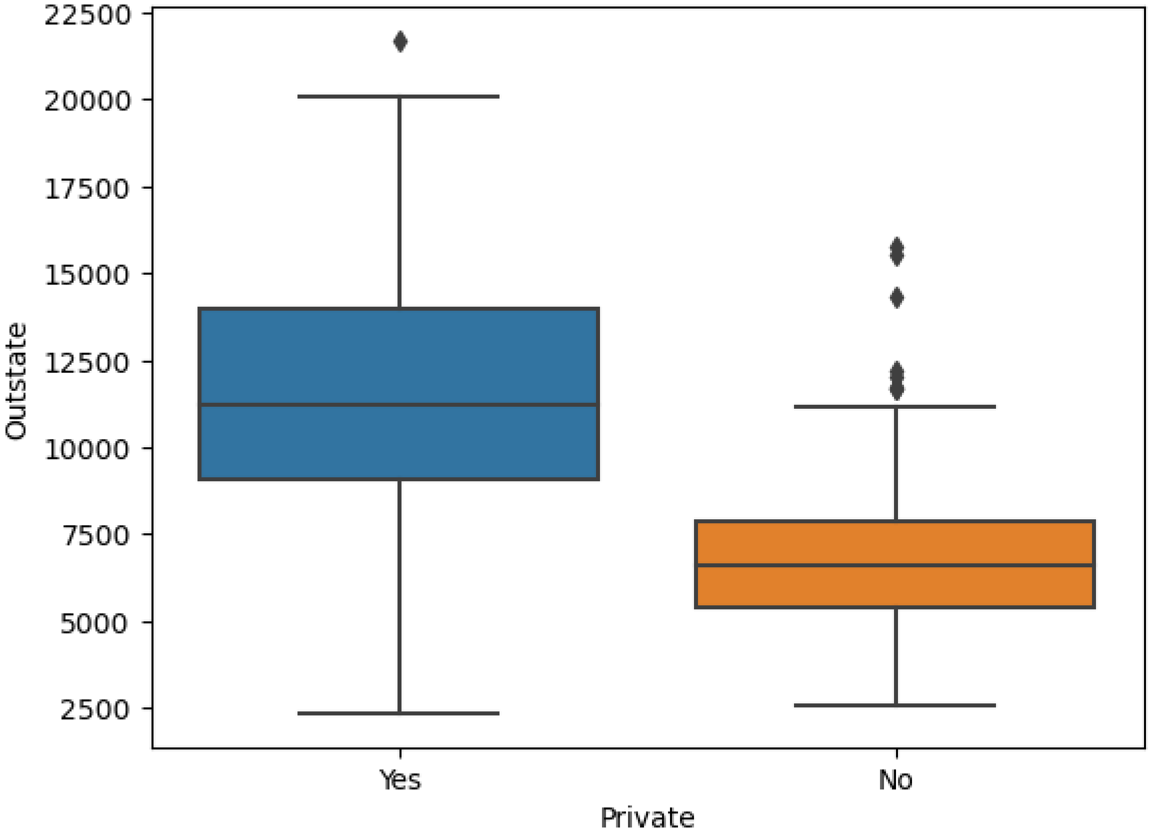
Out[23]:

```
array([[<Axes: xlabel='Top10perc', ylabel='Top10perc'>,
        <Axes: xlabel='Apps', ylabel='Top10perc'>,
        <Axes: xlabel='Enroll', ylabel='Top10perc'>],
       [<Axes: xlabel='Top10perc', ylabel='Apps'>,
        <Axes: xlabel='Apps', ylabel='Apps'>,
        <Axes: xlabel='Enroll', ylabel='Apps'>],
       [<Axes: xlabel='Top10perc', ylabel='Enroll'>,
        <Axes: xlabel='Apps', ylabel='Enroll'>,
        <Axes: xlabel='Enroll', ylabel='Enroll'>]], dtype=object)
```



```
In [25]: import seaborn as sns
         sns.boxplot(x='Private', y='Outstate', data=college)
```

```
Out[25]: <Axes: xlabel='Private', ylabel='Outstate'>
```

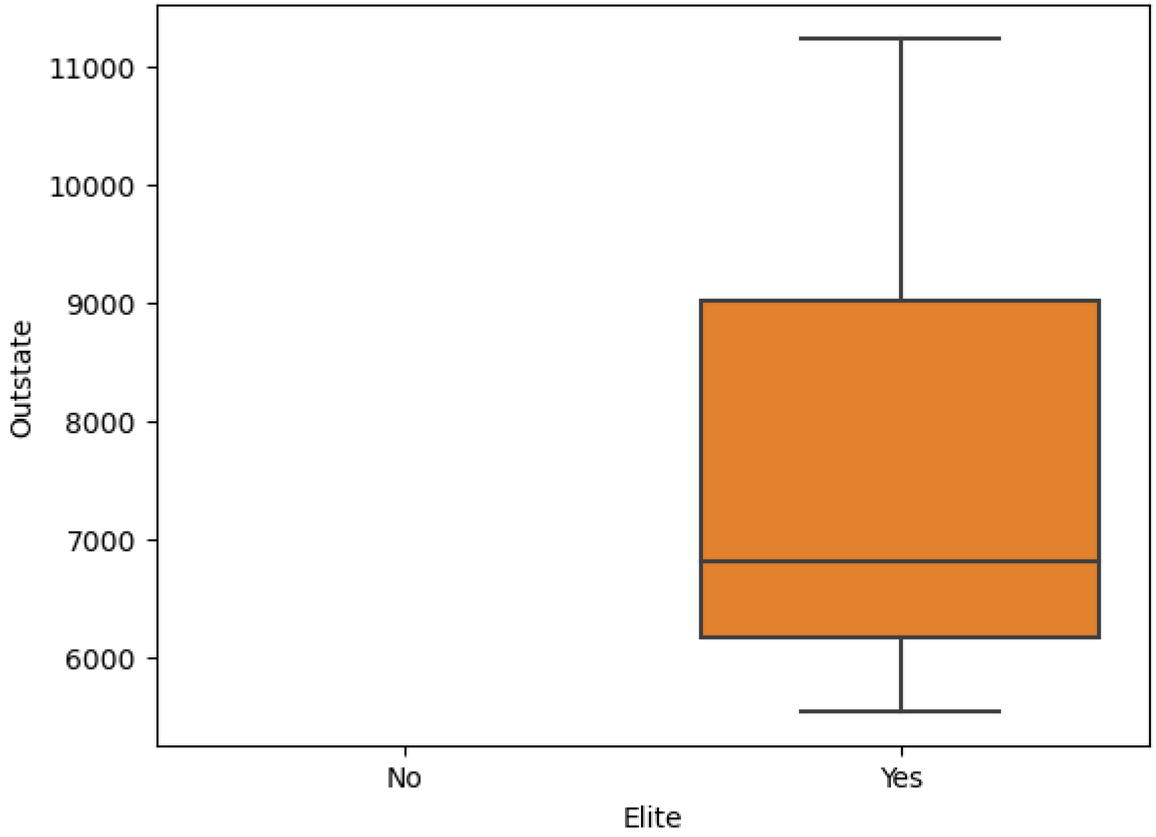


In [29]:
```
college['Elite'] = pd.cut(college['Top10perc'],
[0,0.5,1],
labels=['No', 'Yes'])

college['Elite']
```

Out[29]:
```
College
Abilene Christian University      NaN
Adelphi University                NaN
Adrian College                    NaN
Agnes Scott College               NaN
Alaska Pacific University         NaN
                                 ...
Worcester State College           NaN
Xavier University                 NaN
Xavier University of Louisiana    NaN
Yale University                   NaN
York College of Pennsylvania      NaN
Name: Elite, Length: 777, dtype: category
Categories (2, object): ['No' < 'Yes']
```

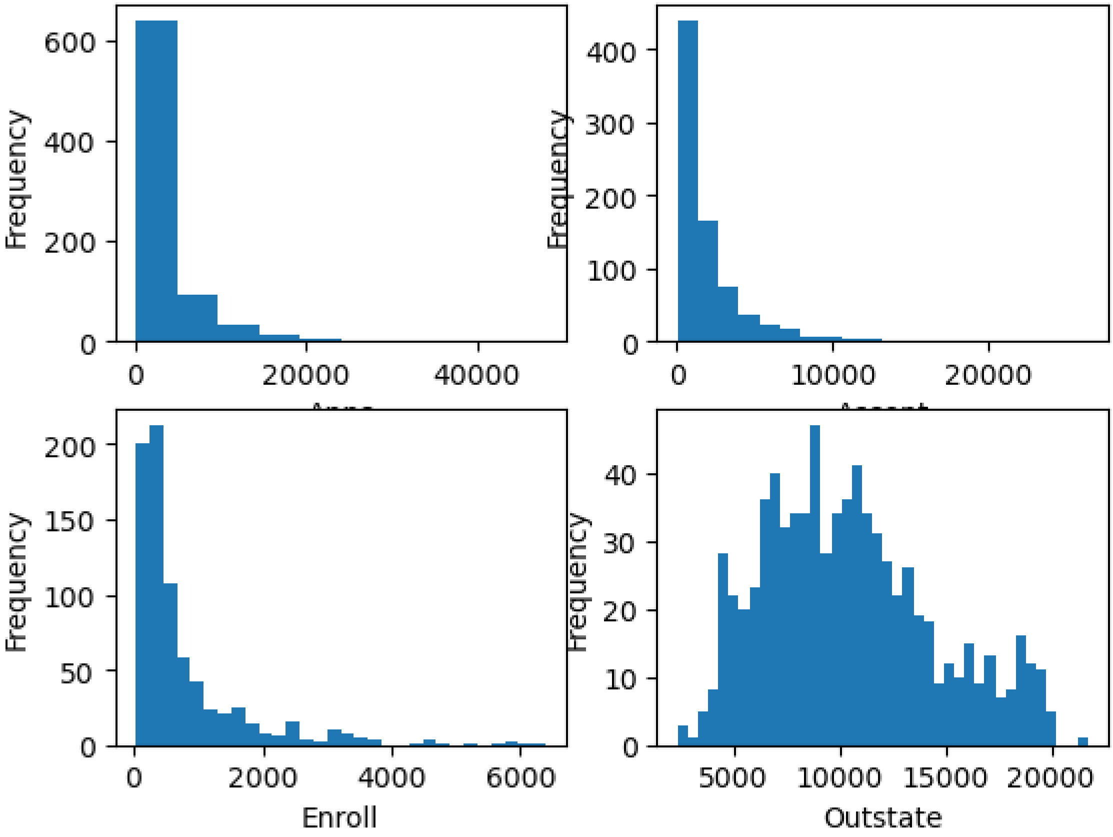
In [28]:
```
sns.boxplot(x='Elite', y='Outstate', data=college)
```

Out[28]:
```
<Axes: xlabel='Elite', ylabel='Outstate'>
```



In [38]:

```
fig, axes = plt.subplots(2, 2)

college['Apps'].plot.hist(ax=axes[0, 0], bins=10)
axes[0, 0].set_xlabel('Apps')
axes[0, 0].set_ylabel('Frequency')


college['Accept'].plot.hist(ax=axes[0, 1], bins=20)
axes[0, 1].set_xlabel('Accept')
axes[0, 1].set_ylabel('Frequency')

college['Enroll'].plot.hist(ax=axes[1, 0], bins=30)
axes[1, 0].set_xlabel('Enroll')
axes[1, 0].set_ylabel('Frequency')


college['Outstate'].plot.hist(ax=axes[1, 1], bins=40)
axes[1, 1].set_xlabel('Outstate')
axes[1, 1].set_ylabel('Frequency')

plt.show()
```



In [41]:

```
df = '/Users/siony/OneDrive/바탕 화면/MSU_SS_24/CMSE 381/CMSE381SS24/DataSets/Auto.c
auto = pd.read_csv(df)
auto.head()
```

Out[41]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	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

In [48]:

```
auto.replace('?', pd.NA, inplace=True)
auto.dropna(inplace=True)

auto['horsepower'] = pd.to_numeric(auto['horsepower'])

auto.head()
```

Out[48]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	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

In [49]:

```
auto.dtypes
```

Out[49]:

```
mpg             float64
cylinders         int64
displacement    float64
horsepower        int64
weight            int64
acceleration    float64
year              int64
origin            int64
name             object
dtype: object
```

In [51]:

```
q_d = ['mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', '
ranges = {column: (np.min(auto[column]), np.max(auto[column])) for column in q_d}
ranges
```

Out[51]:

```
{'mpg': (9.0, 46.6),
 'cylinders': (3, 8),
 'displacement': (68.0, 455.0),
 'horsepower': (46, 230),
 'weight': (1613, 5140),
 'acceleration': (8.0, 24.8),
 'year': (70, 82)}
```

In [52]:

```
meandata = {column: (np.mean(auto[column]), np.std(auto[column])) for column in q_d
meandata
```

Out[52]:

```
{'mpg': (23.445918367346938, 7.795045762682584),
 'cylinders': (5.471938775510204, 1.703606114150195),
 'displacement': (194.41198979591837, 104.51044418133284),
 'horsepower': (104.46938775510205, 38.442032714425984),
 'weight': (2977.5841836734694, 848.3184465698364),
 'acceleration': (15.541326530612244, 2.7553429127509963),
 'year': (75.9795918367347, 3.679034899615175)}
```

In [ ]:

```

```