First thing first , we import our libraries and dataset and then we see the head of the data to know how the data looks like and use describe function to see the percentile's and other key statistics.

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import mpl_toolkits
        %matplotlib inline
```

```
In [2]: data = pd.read_csv("kc_house_data.csv")
```

```
In [3]: data.head()
```

Out[3]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_built
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	...	7	1180	0	1955
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	...	7	2170	400	1951
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	...	6	770	0	1933
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	...	7	1050	910	1965
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	...	8	1680	0	1987

5 rows × 21 columns

```
In [4]: data.describe()
```

Out[4]:

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000	21613.0
mean	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.494309	0.007542	0.234303	3.409430	7.6
std	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	0.539989	0.086517	0.766318	0.650743	1.1
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000	0.000000	0.000000	1.000000	1.0
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000	0.000000	0.000000	3.000000	7.0
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000	0.000000	0.000000	3.000000	7.0
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000	0.000000	0.000000	4.000000	8.0
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000	1.000000	4.000000	5.000000	13.0

Look at the bedroom columns , the dataset has a house where the house has 33 bedrooms , seems to be a massive house and would be interesting to know more about it as we progress.

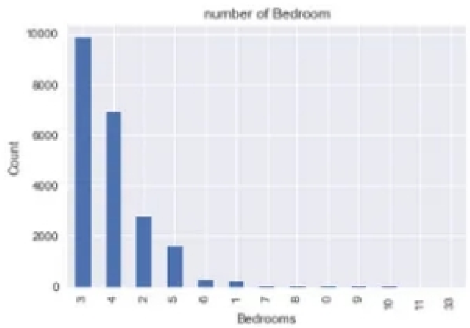
1.Maximum square feet is 13,450 where as the minimum is 290. we can see that the data is distributed.

2.Now , we are going to see some visualization and also going to see how and what can we infer from visualization.

3.Let's see which is most common bedroom number. You may wonder why is it important ? Let's look at this problem from a builder's perspective, sometimes it's important for a builder to see which is the highest selling house type which enables the builder to make house based on that. Here in India , for a good locality a builder opts to make houses which are more than 3 bedrooms which attracts the higher middle class and upper class section of the society.

In [5]:
```
data['bedrooms'].value_counts().plot(kind='bar')
plt.title('number of Bedroom')
plt.xlabel('Bedrooms')
plt.ylabel('Count')
sns.despine
```

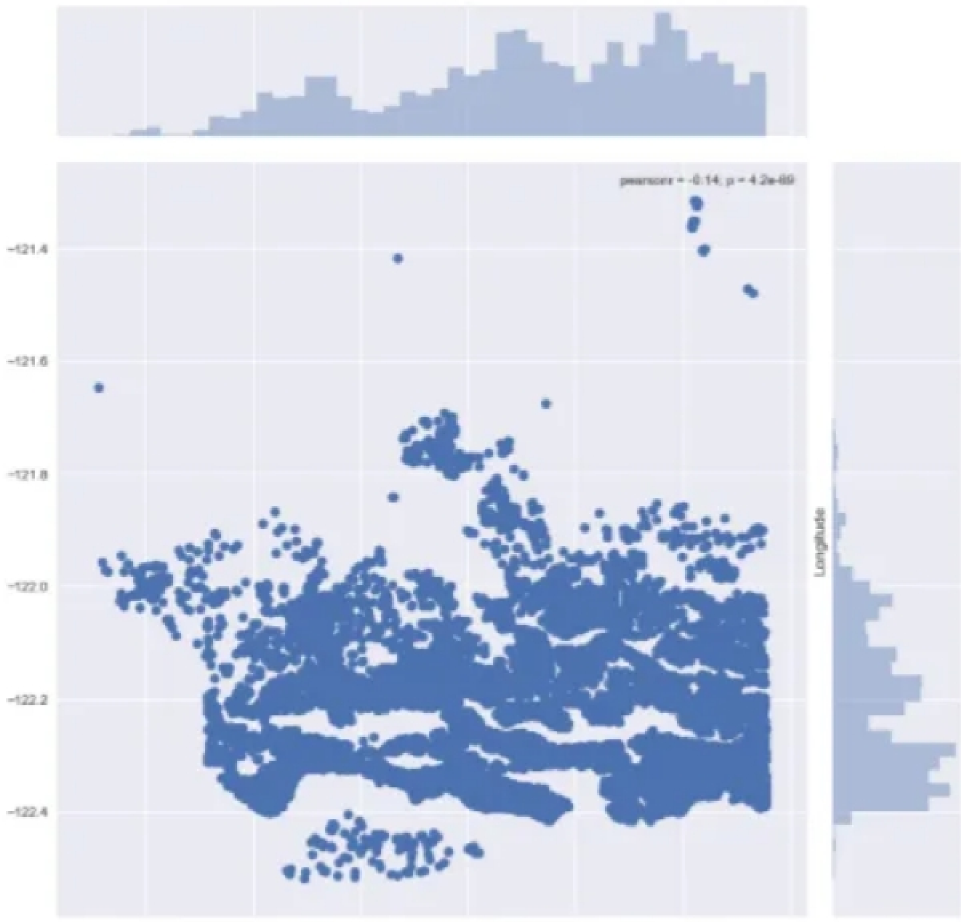
Out[5]: <function seaborn.utils.despine>



So according to the dataset , we have latitude and longitude on the dataset for each house. We are going to see the common location and how the houses are placed.

```
plt.figure(figsize=(10,10))
sns.jointplot(x=data.lat.values, y=data.long.values, size=10)
plt.ylabel('Longitude', fontsize=12)
plt.xlabel('Latitude', fontsize=12)
plt.show()
sns.despine
```

```
<matplotlib.figure.Figure at 0x1c1eba30400>
```



We saw the common locations and now we're going to see few common factors affecting the prices of the house and if so ? then by how much ?

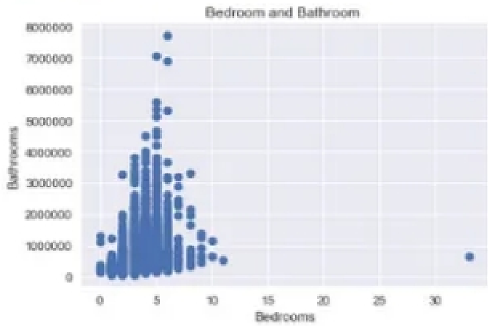
```
In [81]: plt.scatter(data.price,data.sqft_living)
         plt.title("Price vs Square Feet")

Out[81]: <matplotlib.text.Text at 0x1c1ee8d4e48>
```


```
In [8]: plt.scatter(data.price,data.long)
        plt.title("Price vs Location of the area")

Out[8]: <matplotlib.text.Text at 0x1c1ebf410f0>
```

The plot that we used above is called scatter plot , scatter plot helps us to see how our data points are scattered and are usually used for two variables. From the first figure we can see that more the living area , more the price though data is concentrated towards a particular price zone , but from the figure we can see that the data points seem to be in linear direction. Thanks to scatter plot we can also see some irregularities that the house with the highest square feet was sold for very less , maybe there is another factor or probably the data must be wrong. The second figure tells us about the location of the houses in terms of longitude and it gives us quite an interesting observation that -122.2 to -122.4 sells houses at much higher amount.

```
Out[82]: <matplotlib.text.Text at 0x1c1ee873d30>
```


```
In [83]: plt.scatter(data.bedrooms,data.price)
         plt.title("Bedroom and Price ")
         plt.xlabel("Bedrooms")
         plt.ylabel("Price")
         plt.show()
         sns.despine
```



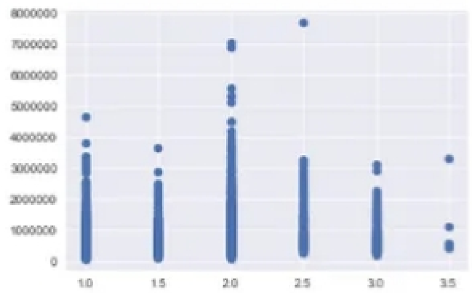

```
Out[83]: <function seaborn.utils.despine>
```

We can see more factors affecting the price

```
In [18]: plt.scatter(data.zipcode,data.price)
         plt.title("Which is the pricey location by zipcode?")
Out[18]: <matplotlib.text.Text at 0x1c1ec26a898>
```

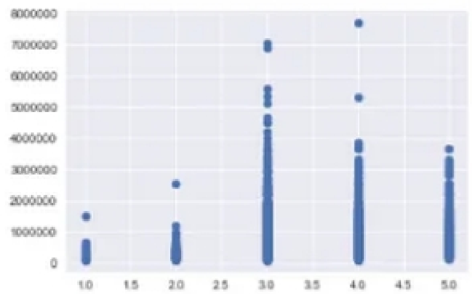
```
In [16]: plt.scatter(data.floors,data.price)
```

Out[16]: <matplotlib.collections.PathCollection at 0x1c1ec1a61d0>



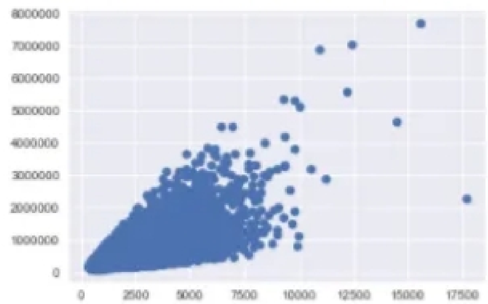
```
In [17]: plt.scatter(data.condition,data.price)
```

Out[17]: <matplotlib.collections.PathCollection at 0x1c1ec214630>



```
In [84]: plt.scatter((data['sqft_living']+data['sqft_basement']),data['price'])
```

Out[84]: <matplotlib.collections.PathCollection at 0x1c1edfea160>



```
In [12]: plt.scatter(data.waterfront,data.price)
         plt.title("Waterfront vs Price ( 0= no waterfront)")
```

Out[12]: <matplotlib.text.Text at 0x1c1eb8a87f0>

ex.no-4
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("House_Rent_Dataset.csv")
print(data.head())

```
    Posted On  BHK   Rent  Size            Floor    Area Type  \
0  2022-05-18    2  10000  1100  Ground out of 2   Super Area
1  2022-05-13    2  20000   800       1 out of 3   Super Area
2  2022-05-16    2  17000  1000       1 out of 3   Super Area
3  2022-07-04    2  10000   800       1 out of 2   Super Area
4  2022-05-09    2   7500   850       1 out of 2  Carpet Area

             Area Locality     City Furnishing Status  Tenant Preferred  \
0                   Bandel  Kolkata       Unfurnished  Bachelors/Family
1  Phool Bagan, Kankurgachi  Kolkata    Semi-Furnished  Bachelors/Family
2   Salt Lake City Sector 2  Kolkata    Semi-Furnished  Bachelors/Family
3              Dumdum Park  Kolkata       Unfurnished  Bachelors/Family
4            South Dum Dum  Kolkata       Unfurnished         Bachelors

   Bathroom Point of Contact
0         2    Contact Owner
1         1    Contact Owner
2         1    Contact Owner
3         1    Contact Owner
4         1    Contact Owner
```

print(data.isnull().sum())

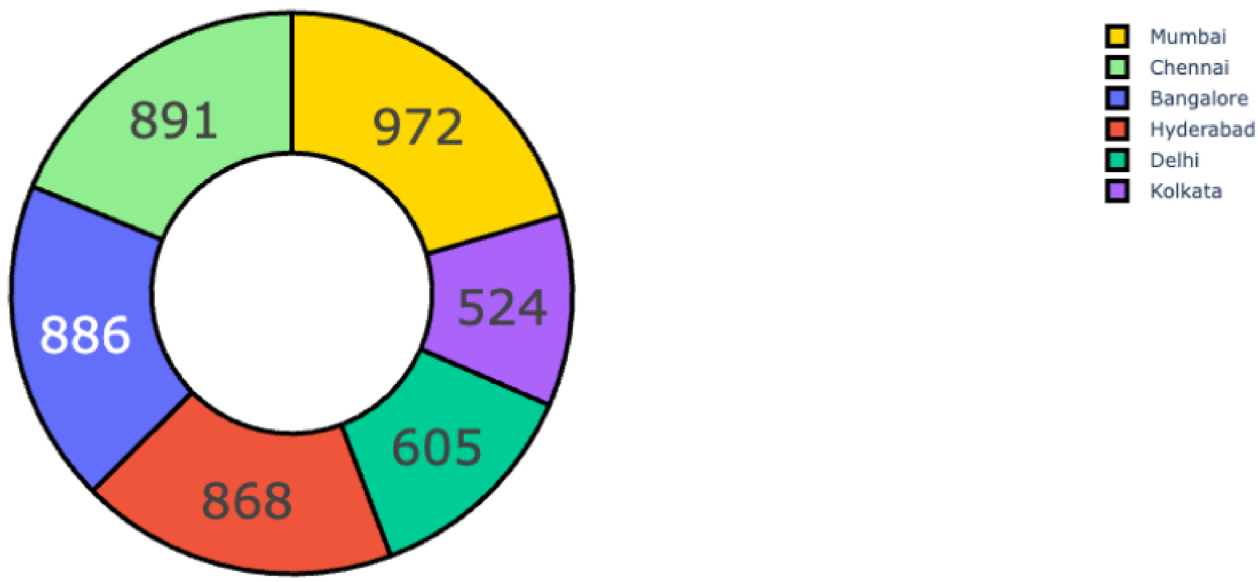
```
Posted On            0
BHK                  0
Rent                 0
Size                 0
Floor                0
Area Type            0
Area Locality        0
City                 0
Furnishing Status    0
Tenant Preferred     0
Bathroom             0
Point of Contact     0
dtype: int64
```

print(data.describe())

```
               BHK          Rent         Size     Bathroom
count  4746.000000  4.746000e+03  4746.000000  4746.000000
mean      2.083860  3.499345e+04   967.490729     1.965866
std       0.832256  7.810641e+04   634.202328     0.884532
min       1.000000  1.200000e+03    10.000000     1.000000
25%       2.000000  1.000000e+04   550.000000     1.000000
50%       2.000000  1.600000e+04   850.000000     2.000000
75%       3.000000  3.300000e+04  1200.000000     2.000000
max       6.000000  3.500000e+06  8000.000000    10.000000
```

```
figure = px.bar(data, x=data["City"],
               y = data["Rent"],
               color = data["BHK"],
           title="Rent in Different Cities According to BHK")
figure.show()
```


```
cities = data["City"].value_counts()
label = cities.index
counts = cities.values
colors = ['gold','lightgreen']

fig = go.Figure(data=[go.Pie(labels=label, values=counts, hole=0.5)])
fig.update_layout(title_text='Number of Houses Available for Rent')
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=30,
                  marker=dict(colors=colors, line=dict(color='black', width=3)))
fig.show()
```

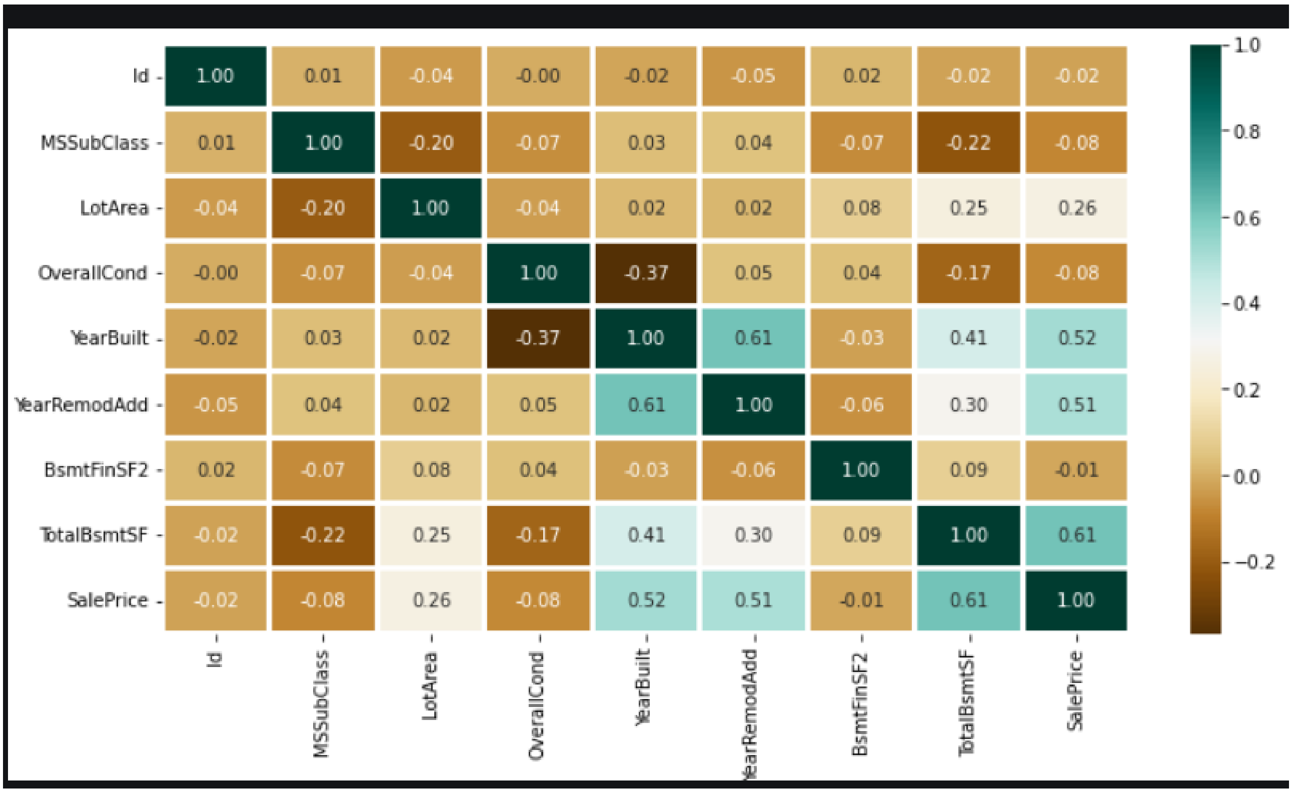


ex-5

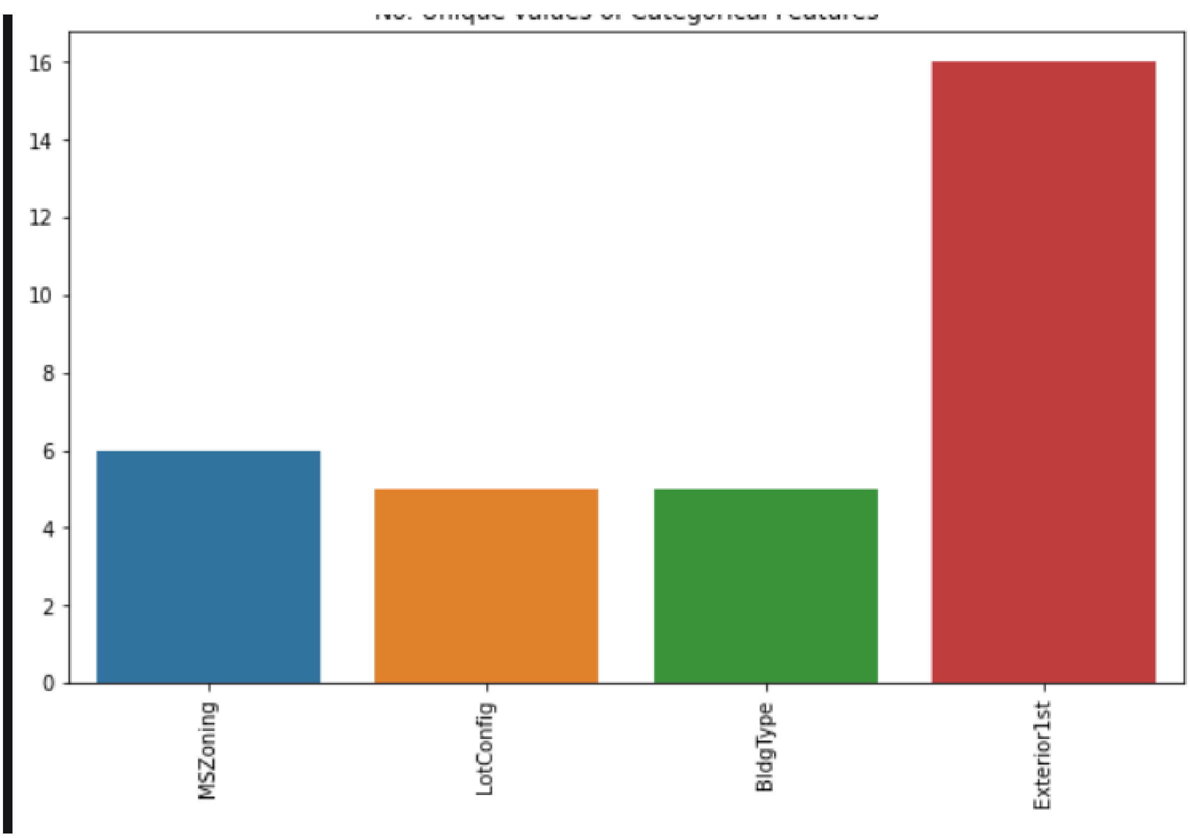
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
dataset = pd.read_excel("HousePricePrediction.csv")
print(dataset.head(5))
```

```
   MSSubClass MSZoning  LotArea LotConfig BldgType  OverallCond  YearBuilt
0          60       RL     8450    Inside     1Fam            5       2003
1          20       RL     9600       FR2     1Fam            8       1976
2          60       RL    11250    Inside     1Fam            5       2001
3          70       RL     9550    Corner     1Fam            5       1915
4          60       RL    14260       FR2     1Fam            5       2000

   YearRemodAdd Exterior1st  BsmtFinSF2  TotalBsmtSF  SalePrice
0          2003     VinylSd         0.0        856.0   208500.0
1          1976     MetalSd         0.0       1262.0   181500.0
2          2002     VinylSd         0.0        920.0   223500.0
3          1970     Wd Sdng         0.0        756.0   140000.0
4          2000     VinylSd         0.0       1145.0   250000.0
```

```
plt.figure(figsize=(12, 6))
sns.heatmap(dataset.corr(),
 cmap = 'BrBG',
fmt = '.2f',
linewidths = 2,
annot = True)
```




```
unique_values = []
for col in object_cols:
  unique_values.append(dataset[col].unique().size)
plt.figure(figsize=(10,6))
plt.title('No. Unique values of Categorical Features')
plt.xticks(rotation=90)
sns.barplot(x=object_cols,y=unique_values)
```

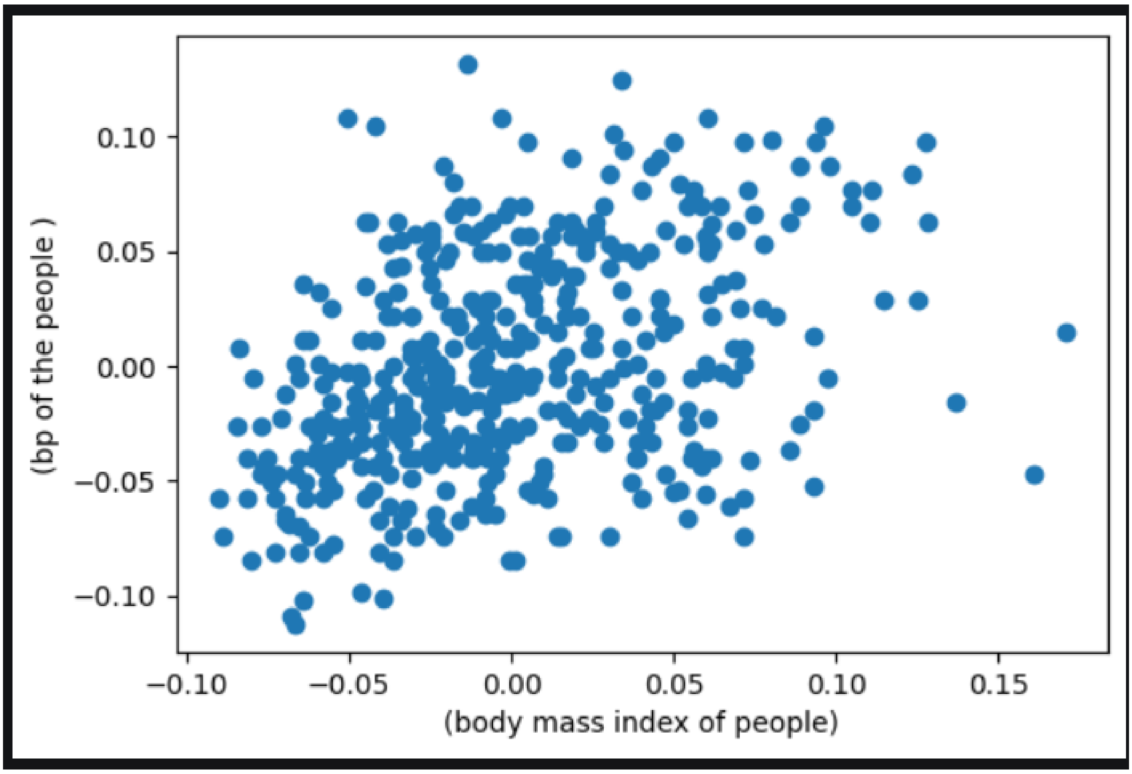

16
14
12
10
8
6
4
2
0
MSZoning
LotConfig
BldgType
Exterior1st

Ex.6:

```
import sklearn
from sklearn.datasets import load_diabetes
import pandas as pd
import matplotlib.pyplot as plt

diabetics = load_diabetes()
column_name = diabetics.feature_names
df_diabetics = pd.DataFrame(diabetics.data)
df_diabetics.columns = column_name
df_diabetics.head()
```

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641

```
fig, ax = plt.subplots(figsize = (6,4))
ax.scatter(df_diabetics['bmi'],df_diabetics['bp'])

# x-axis label
ax.set_xlabel('(body mass index of people)')

# y-axis label
ax.set_ylabel('(bp of the people )')
plt.show()
```


0.10
0.05
0.00
−0.05
−0.10
(bp of the people)
−0.10
−0.05
0.00
0.05
0.10
0.15
(body mass index of people)