```
import seaborn as sns

print(sns.get_dataset_names())
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes',
'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri',
'geyser', 'glue', 'healthexp', 'iris', 'mpg', 'penguins', 'planets',
'seaice', 'taxis', 'tips', 'titanic']
```

```
df=sns.load_dataset('car_crashes')

df
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium
0	18.8	7.332	5.640	18.048	15.040	784.55
1	18.1	7.421	4.525	16.290	17.014	1053.48
2	18.6	6.510	5.208	15.624	17.856	899.47
3	22.4	4.032	5.824	21.056	21.280	827.34
4	12.0	4.200	3.360	10.920	10.680	878.41
5	13.6	5.032	3.808	10.744	12.920	835.50
6	10.8	4.968	3.888	9.396	8.856	1068.73
7	16.2	6.156	4.860	14.094	16.038	1137.87
8	5.9	2.006	1.593	5.900	5.900	1273.89
9	17.9	3.759	5.191	16.468	16.826	1160.13
10	15.6	2.964	3.900	14.820	14.508	913.15
11	17.5	9.450	7.175	14.350	15.225	861.18
12	15.3	5.508	4.437	13.005	14.994	641.96
13	12.8	4.608	4.352	12.032	12.288	803.11
14	14.5	3.625	4.205	13.775	13.775	710.46
15	15.7	2.669	3.925	15.229	13.659	649.06
16	17.8	4.806	4.272	13.706	15.130	780.45
17	21.4	4.066	4.922	16.692	16.264	872.51
18	20.5	7.175	6.765	14.965	20.090	1281.55

19	15.1	5.738	4.530	13.137	12.684	661.88
20	12.5	4.250	4.000	8.875	12.375	1048.78
21	8.2	1.886	2.870	7.134	6.560	1011.14
22	14.1	3.384	3.948	13.395	10.857	1110.61
23	9.6	2.208	2.784	8.448	8.448	777.18
24	17.6	2.640	5.456	1.760	17.600	896.07
25	16.1	6.923	5.474	14.812	13.524	790.32
26	21.4	8.346	9.416	17.976	18.190	816.21
27	14.9	1.937	5.215	13.857	13.410	732.28
28	14.7	5.439	4.704	13.965	14.553	1029.87
29	11.6	4.060	3.480	10.092	9.628	746.54
30	11.2	1.792	3.136	9.632	8.736	1301.52
31	18.4	3.496	4.968	12.328	18.032	869.85
32	12.3	3.936	3.567	10.824	9.840	1234.31
33	16.8	6.552	5.208	15.792	13.608	708.24
34	23.9	5.497	10.038	23.661	20.554	688.75
35	14.1	3.948	4.794	13.959	11.562	697.73
36	19.9	6.368	5.771	18.308	18.706	881.51
37	12.8	4.224	3.328	8.576	11.520	804.71
38	18.2	9.100	5.642	17.472	16.016	905.99
39	11.1	3.774	4.218	10.212	8.769	1148.99
40	23.9	9.082	9.799	22.944	19.359	858.97
41	19.4	6.014	6.402	19.012	16.684	669.31
42	19.5	4.095	5.655	15.990	15.795	767.91
43	19.4	7.760	7.372	17.654	16.878	1004.75
44	11.3	4.859	1.808	9.944	10.848	809.38

```
45   13.6     4.080    4.080          13.056        12.920        716.20

46   12.7     2.413    3.429          11.049        11.176        768.95

47   10.6     4.452    3.498           8.692         9.116        890.03

48   23.8     8.092    6.664          23.086        20.706        992.61

49   13.8     4.968    4.554           5.382        11.592        670.31

50   17.4     7.308    5.568          14.094        15.660        791.14
```

```
    ins_losses abbrev
0       145.08     AL
1       133.93     AK
2       110.35     AZ
3       142.39     AR
4       165.63     CA
5       139.91     CO
6       167.02     CT
7       151.48     DE
8       136.05     DC
9       144.18     FL
10      142.80     GA
11      120.92     HI
12       82.75     ID
13      139.15     IL
14      108.92     IN
15      114.47     IA
16      133.80     KS
17      137.13     KY
18      194.78     LA
19       96.57     ME
20      192.70     MD
21      135.63     MA
22      152.26     MI
23      133.35     MN
24      155.77     MS
25      144.45     MO
26       85.15     MT
27      114.82     NE
28      138.71     NV
29      120.21     NH
30      159.85     NJ
31      120.75     NM
32      150.01     NY
33      127.82     NC
34      109.72     ND
```

```
35      133.52     OH
36      178.86     OK
37      104.61     OR
38      153.86     PA
39      148.58     RI
40      116.29     SC
41       96.87     SD
42      155.57     TN
43      156.83     TX
44      109.48     UT
45      109.61     VT
46      153.72     VA
47      111.62     WA
48      152.56     WV
49      106.62     WI
50      122.04     WY
```

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

```
df
```

```
    total  speeding  alcohol  not_distracted  no_previous  ins_premium
\
0    18.8     7.332    5.640          18.048       15.040       784.55

1    18.1     7.421    4.525          16.290       17.014      1053.48

2    18.6     6.510    5.208          15.624       17.856       899.47

3    22.4     4.032    5.824          21.056       21.280       827.34

4    12.0     4.200    3.360          10.920       10.680       878.41

5    13.6     5.032    3.808          10.744       12.920       835.50

6    10.8     4.968    3.888           9.396        8.856      1068.73

7    16.2     6.156    4.860          14.094       16.038      1137.87

8     5.9     2.006    1.593           5.900        5.900      1273.89

9    17.9     3.759    5.191          16.468       16.826      1160.13

10   15.6     2.964    3.900          14.820       14.508       913.15

11   17.5     9.450    7.175          14.350       15.225       861.18

12   15.3     5.508    4.437          13.005       14.994       641.96
```

13	12.8	4.608	4.352	12.032	12.288	803.11
14	14.5	3.625	4.205	13.775	13.775	710.46
15	15.7	2.669	3.925	15.229	13.659	649.06
16	17.8	4.806	4.272	13.706	15.130	780.45
17	21.4	4.066	4.922	16.692	16.264	872.51
18	20.5	7.175	6.765	14.965	20.090	1281.55
19	15.1	5.738	4.530	13.137	12.684	661.88
20	12.5	4.250	4.000	8.875	12.375	1048.78
21	8.2	1.886	2.870	7.134	6.560	1011.14
22	14.1	3.384	3.948	13.395	10.857	1110.61
23	9.6	2.208	2.784	8.448	8.448	777.18
24	17.6	2.640	5.456	1.760	17.600	896.07
25	16.1	6.923	5.474	14.812	13.524	790.32
26	21.4	8.346	9.416	17.976	18.190	816.21
27	14.9	1.937	5.215	13.857	13.410	732.28
28	14.7	5.439	4.704	13.965	14.553	1029.87
29	11.6	4.060	3.480	10.092	9.628	746.54
30	11.2	1.792	3.136	9.632	8.736	1301.52
31	18.4	3.496	4.968	12.328	18.032	869.85
32	12.3	3.936	3.567	10.824	9.840	1234.31
33	16.8	6.552	5.208	15.792	13.608	708.24
34	23.9	5.497	10.038	23.661	20.554	688.75
35	14.1	3.948	4.794	13.959	11.562	697.73
36	19.9	6.368	5.771	18.308	18.706	881.51
37	12.8	4.224	3.328	8.576	11.520	804.71
38	18.2	9.100	5.642	17.472	16.016	905.99

39	11.1	3.774	4.218	10.212	8.769	1148.99
40	23.9	9.082	9.799	22.944	19.359	858.97
41	19.4	6.014	6.402	19.012	16.684	669.31
42	19.5	4.095	5.655	15.990	15.795	767.91
43	19.4	7.760	7.372	17.654	16.878	1004.75
44	11.3	4.859	1.808	9.944	10.848	809.38
45	13.6	4.080	4.080	13.056	12.920	716.20
46	12.7	2.413	3.429	11.049	11.176	768.95
47	10.6	4.452	3.498	8.692	9.116	890.03
48	23.8	8.092	6.664	23.086	20.706	992.61
49	13.8	4.968	4.554	5.382	11.592	670.31
50	17.4	7.308	5.568	14.094	15.660	791.14

	ins_losses	abbrev
0	145.08	AL
1	133.93	AK
2	110.35	AZ
3	142.39	AR
4	165.63	CA
5	139.91	CO
6	167.02	CT
7	151.48	DE
8	136.05	DC
9	144.18	FL
10	142.80	GA
11	120.92	HI
12	82.75	ID
13	139.15	IL
14	108.92	IN
15	114.47	IA
16	133.80	KS
17	137.13	KY
18	194.78	LA
19	96.57	ME
20	192.70	MD
21	135.63	MA
22	152.26	MI
23	133.35	MN

```
24      155.77     MS
25      144.45     MO
26       85.15     MT
27      114.82     NE
28      138.71     NV
29      120.21     NH
30      159.85     NJ
31      120.75     NM
32      150.01     NY
33      127.82     NC
34      109.72     ND
35      133.52     OH
36      178.86     OK
37      104.61     OR
38      153.86     PA
39      148.58     RI
40      116.29     SC
41       96.87     SD
42      155.57     TN
43      156.83     TX
44      109.48     UT
45      109.61     VT
46      153.72     VA
47      111.62     WA
48      152.56     WV
49      106.62     WI
50      122.04     WY
```

```
df.head()
```

```
   total  speeding  alcohol  not_distracted  no_previous  ins_premium
\
0   18.8     7.332    5.640           18.048       15.040       784.55

1   18.1     7.421    4.525           16.290       17.014      1053.48

2   18.6     6.510    5.208           15.624       17.856       899.47

3   22.4     4.032    5.824           21.056       21.280       827.34

4   12.0     4.200    3.360           10.920       10.680       878.41


   ins_losses abbrev
0      145.08     AL
1      133.93     AK
2      110.35     AZ
3      142.39     AR
4      165.63     CA
```

```
df.describe()
```

```
           total   speeding    alcohol  not_distracted  no_previous  \
count  51.000000  51.000000  51.000000       51.000000    51.000000
mean   15.790196   4.998196   4.886784       13.573176    14.004882
std     4.122002   2.017747   1.729133        4.508977     3.764672
min     5.900000   1.792000   1.593000        1.760000     5.900000
25%    12.750000   3.766500   3.894000       10.478000    11.348000
50%    15.600000   4.608000   4.554000       13.857000    13.775000
75%    18.500000   6.439000   5.604000       16.140000    16.755000
max    23.900000   9.450000  10.038000       23.661000    21.280000

       ins_premium  ins_losses
count    51.000000   51.000000
mean    886.957647  134.493137
std     178.296285   24.835922
min     641.960000   82.750000
25%     768.430000  114.645000
50%     858.970000  136.050000
75%    1007.945000  151.870000
max    1301.520000  194.780000
```

```
df.isnull().any()
```

```
total             False
speeding          False
alcohol           False
not_distracted    False
no_previous       False
ins_premium       False
ins_losses        False
abbrev            False
dtype: bool
```

```
df.isnull().sum()
```

```
total             0
speeding          0
alcohol           0
not_distracted    0
no_previous       0
ins_premium       0
ins_losses        0
abbrev            0
dtype: int64
```

```
df.head()
```

```
   total  speeding  alcohol  not_distracted  no_previous  ins_premium
\
0   18.8     7.332    5.640          18.048       15.040       784.55

1   18.1     7.421    4.525          16.290       17.014      1053.48
```

```
2   18.6     6.510    5.208          15.624       17.856       899.47

3   22.4     4.032    5.824          21.056       21.280       827.34

4   12.0     4.200    3.360          10.920       10.680       878.41
```

```
   ins_losses abbrev
0      145.08     AL
1      133.93     AK
2      110.35     AZ
3      142.39     AR
4      165.63     CA
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x78c4ab817880>
```

```
import numpy as np
from google.colab import autoviz

def value_plot(df, y, figscale=1):
  from matplotlib import pyplot as plt
  df[y].plot(kind='line', figsize=(8 * figscale, 4 * figscale),
title=y)
  plt.gca().spines[['top', 'right']].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = value_plot(_df_0, *['total'], **{})
chart
```

```
import numpy as np
from google.colab import autoviz

def value_plot(df, y, figscale=1):
  from matplotlib import pyplot as plt
  df[y].plot(kind='line', figsize=(8 * figscale, 4 * figscale),
title=y)
  plt.gca().spines[['top', 'right']].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = value_plot(_df_1, *['speeding'], **{})
chart
```

```
import numpy as np
from google.colab import autoviz

def value_plot(df, y, figscale=1):
  from matplotlib import pyplot as plt
  df[y].plot(kind='line', figsize=(8 * figscale, 4 * figscale),
```

```
title=y)
  plt.gca().spines[['top', 'right']].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = value_plot(_df_2, *['alcohol'], **{})
chart

import numpy as np
from google.colab import autoviz

def value_plot(df, y, figscale=1):
  from matplotlib import pyplot as plt
  df[y].plot(kind='line', figsize=(8 * figscale, 4 * figscale),
title=y)
  plt.gca().spines[['top', 'right']].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = value_plot(_df_3, *['not_distracted'], **{})
chart

<google.colab._quickchart_helpers.SectionTitle at 0x78c4a86f20e0>

import numpy as np
from google.colab import autoviz

def histogram(df, colname, num_bins=20, figscale=1):
  from matplotlib import pyplot as plt
  df[colname].plot(kind='hist', bins=num_bins, title=colname,
figsize=(8*figscale, 4*figscale))
  plt.gca().spines[['top', 'right',]].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = histogram(_df_4, *['total'], **{})
chart

import numpy as np
from google.colab import autoviz

def histogram(df, colname, num_bins=20, figscale=1):
  from matplotlib import pyplot as plt
  df[colname].plot(kind='hist', bins=num_bins, title=colname,
figsize=(8*figscale, 4*figscale))
  plt.gca().spines[['top', 'right',]].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = histogram(_df_5, *['speeding'], **{})
chart
```

```
import numpy as np
from google.colab import autoviz

def histogram(df, colname, num_bins=20, figscale=1):
  from matplotlib import pyplot as plt
  df[colname].plot(kind='hist', bins=num_bins, title=colname,
figsize=(8*figscale, 4*figscale))
  plt.gca().spines[['top', 'right',]].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = histogram(_df_6, *['alcohol'], **{})
chart
```

```
import numpy as np
from google.colab import autoviz

def histogram(df, colname, num_bins=20, figscale=1):
  from matplotlib import pyplot as plt
  df[colname].plot(kind='hist', bins=num_bins, title=colname,
figsize=(8*figscale, 4*figscale))
  plt.gca().spines[['top', 'right',]].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = histogram(_df_7, *['not_distracted'], **{})
chart
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x78c4a8377dc0>
```

```
import numpy as np
from google.colab import autoviz

def categorical_histogram(df, colname, figscale=1,
mpl_palette_name='Dark2'):
  from matplotlib import pyplot as plt
  import seaborn as sns
  df.groupby(colname).size().plot(kind='barh',
color=sns.palettes.mpl_palette(mpl_palette_name), figsize=(8*figscale,
4.8*figscale))
  plt.gca().spines[['top', 'right',]].set_visible(False)
  return autoviz.MplChart.from_current_mpl_state()

chart = categorical_histogram(_df_8, *['abbrev'], **{})
chart
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x78c4a83fa200>
```

```
import numpy as np
from google.colab import autoviz
```

```
def scatter_plots(df, colname_pairs, figscale=1, alpha=.8):
  from matplotlib import pyplot as plt
  plt.figure(figsize=(len(colname_pairs) * 6 * figscale, 6 *
figscale))
  for plot_i, (x_colname, y_colname) in enumerate(colname_pairs,
start=1):
    ax = plt.subplot(1, len(colname_pairs), plot_i)
    df.plot(kind='scatter', x=x_colname, y=y_colname, s=(32 *
figscale), alpha=alpha, ax=ax)
    ax.spines[['top', 'right',]].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = scatter_plots(_df_9, *[[['total', 'speeding'], ['speeding',
'alcohol'], ['alcohol', 'not_distracted'], ['not_distracted',
'no_previous']]], **{})
chart
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x78c4a8234cd0>
```

```
import numpy as np
from google.colab import autoviz

def violin_plot(df, value_colname, facet_colname, figscale=1,
mpl_palette_name='Dark2', **kwargs):
  from matplotlib import pyplot as plt
  import seaborn as sns
  figsize = (12 * figscale, 1.2 * figscale *
len(df[facet_colname].unique()))
  plt.figure(figsize=figsize)
  sns.violinplot(df, x=value_colname, y=facet_colname,
palette=mpl_palette_name, **kwargs)
  sns.despine(top=True, right=True, bottom=True, left=True)
  return autoviz.MplChart.from_current_mpl_state()

chart = violin_plot(_df_10, *['total', 'abbrev'], **{'inner':
'stick'})
chart
```

```
import numpy as np
from google.colab import autoviz

def violin_plot(df, value_colname, facet_colname, figscale=1,
mpl_palette_name='Dark2', **kwargs):
  from matplotlib import pyplot as plt
  import seaborn as sns
  figsize = (12 * figscale, 1.2 * figscale *
len(df[facet_colname].unique()))
  plt.figure(figsize=figsize)
  sns.violinplot(df, x=value_colname, y=facet_colname,
```

```
palette=mpl_palette_name, **kwargs)
  sns.despine(top=True, right=True, bottom=True, left=True)
  return autoviz.MplChart.from_current_mpl_state()

chart = violin_plot(_df_11, *['speeding', 'abbrev'], **{'inner':
'stick'})
chart
```

```
import numpy as np
from google.colab import autoviz

def violin_plot(df, value_colname, facet_colname, figscale=1,
mpl_palette_name='Dark2', **kwargs):
  from matplotlib import pyplot as plt
  import seaborn as sns
  figsize = (12 * figscale, 1.2 * figscale *
len(df[facet_colname].unique()))
  plt.figure(figsize=figsize)
  sns.violinplot(df, x=value_colname, y=facet_colname,
palette=mpl_palette_name, **kwargs)
  sns.despine(top=True, right=True, bottom=True, left=True)
  return autoviz.MplChart.from_current_mpl_state()

chart = violin_plot(_df_12, *['alcohol', 'abbrev'], **{'inner':
'stick'})
chart
```

```
import numpy as np
from google.colab import autoviz

def violin_plot(df, value_colname, facet_colname, figscale=1,
mpl_palette_name='Dark2', **kwargs):
  from matplotlib import pyplot as plt
  import seaborn as sns
  figsize = (12 * figscale, 1.2 * figscale *
len(df[facet_colname].unique()))
  plt.figure(figsize=figsize)
  sns.violinplot(df, x=value_colname, y=facet_colname,
palette=mpl_palette_name, **kwargs)
  sns.despine(top=True, right=True, bottom=True, left=True)
  return autoviz.MplChart.from_current_mpl_state()

chart = violin_plot(_df_13, *['not_distracted', 'abbrev'], **{'inner':
'stick'})
chart
```

```
X=df.iloc[:,:1]
```

```
X.head()
```

```
   total
0   18.8
1   18.1
2   18.6
3   22.4
4   12.0
```

```
y=df["speeding"]
```

```
type(y)
```

```
pandas.core.series.Series
```

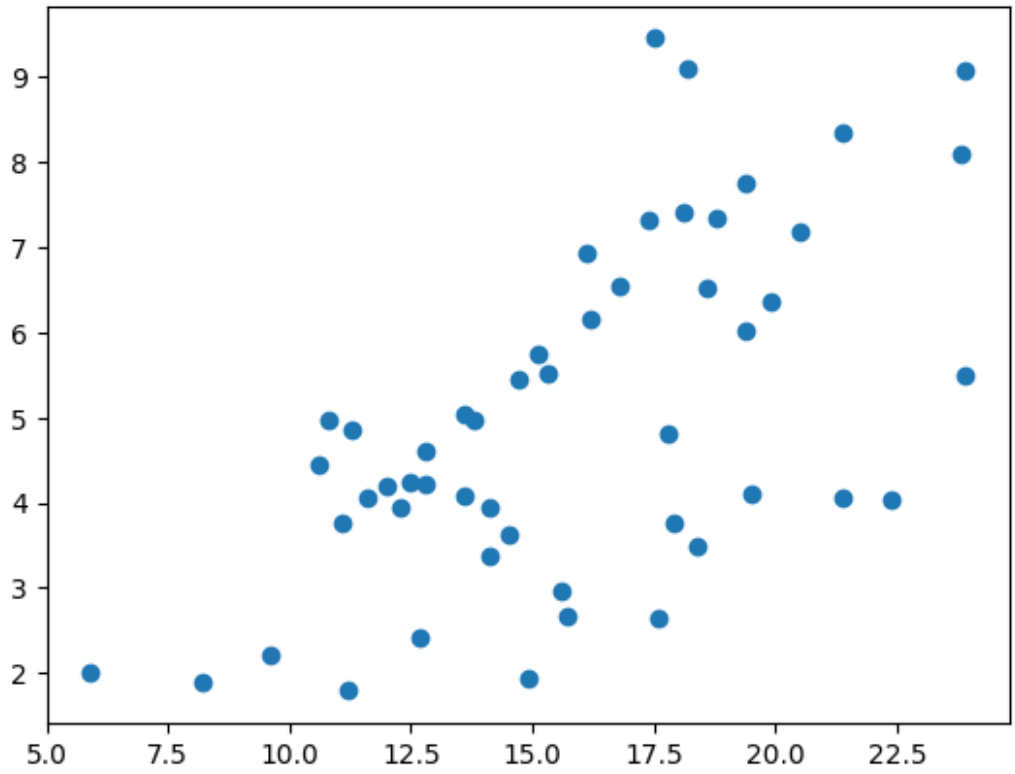
```
y.head()
```

```
0    7.332
1    7.421
2    6.510
3    4.032
4    4.200
Name: speeding, dtype: float64
```

```
plt.scatter(X,y)
plt.X_label("X")
plt.y_label("y")
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-23-a34546f1c46a> in <cell line: 2>()
      1 plt.scatter(X,y)
----> 2 plt.X_label("X")
      3 plt.y_label("y")

AttributeError: module 'matplotlib.pyplot' has no attribute 'X_label'
```




```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.2,rando
m_state=0)

x_train
```

	total
41	19.4
27	14.9
35	14.1
34	23.9
7	16.2
14	14.5
46	12.7
18	20.5
48	23.8
42	19.5
15	15.7
31	18.4
30	11.2
16	17.8
43	19.4
20	12.5
50	17.4
8	5.9

```
13   12.8
25   16.1
5    13.6
17   21.4
37   12.8
40   23.9
1    18.1
12   15.3
38   18.2
24   17.6
6    10.8
23    9.6
36   19.9
21    8.2
19   15.1
9    17.9
39   11.1
49   13.8
3    22.4
0    18.8
47   10.6
44   11.3
```

```
from sklearn.linear_model import LinearRegression
```

```
lr=LinearRegression()
```

```
lr.fit(x_train,y_train)
```

```
LinearRegression()
```

```
y=mx+c
```

```
lr.coef_
```

```
array([0.27553412])
```

```
lr.intercept_
```

```
0.5111052807658751
```

```
y_pred=lr.predict(x_test)
```

```
y_pred
```

```
array([3.70730109, 5.33295241, 4.80943758, 4.39613639, 5.63603994,
       4.56145687, 4.25836933, 3.90017497, 6.40753548, 3.81751474,
       5.14007852])
```

```
y_test
```

```
29    4.060
11    9.450
10    2.964
22    3.384
2     6.510
28    5.439
45    4.080
32    3.936
26    8.346
4     4.200
33    6.552
Name: speeding, dtype: float64
```

```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

```
0.35251712491217213
```

```
plt.scatter(x_train, y_train)
plt.plot(x_train, lr.predict(x_train),)
```

```
[<matplotlib.lines.Line2D at 0x78c4a86d5990>]
```