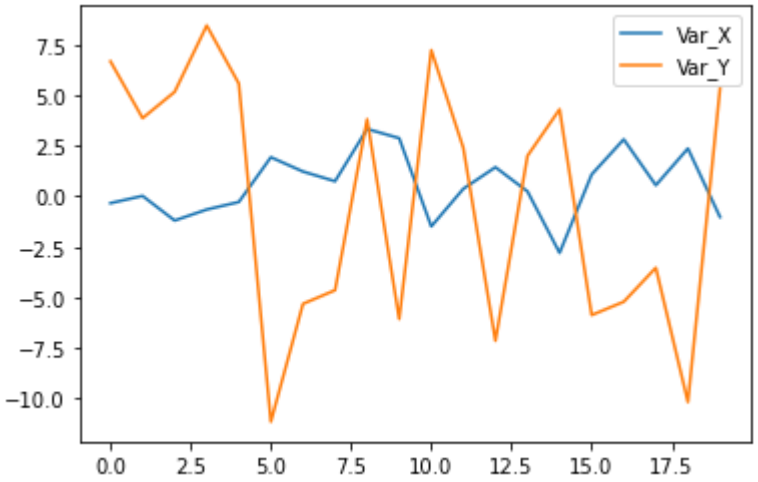
In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
```

In [ ]:
```
# loading the data
data = pd.read_csv('data.csv')
display(data)
```

	Var_X	Var_Y
0	-0.33532	6.66854
1	0.02160	3.86398
2	-1.19438	5.16161
3	-0.65046	8.43823
4	-0.28001	5.57201
5	1.93258	-11.13270
6	1.22620	-5.31226
7	0.74727	-4.63725
8	3.32853	3.80650
9	2.87457	-6.06084
10	-1.48662	7.22328
11	0.37629	2.38887
12	1.43918	-7.13415
13	0.24183	2.00412
14	-2.79140	4.29794
15	1.08176	-5.86553
16	2.81555	-5.20711
17	0.54924	-3.52863
18	2.36449	-10.16202
19	-1.01925	5.31123

In [ ]:
```
# plting the data
data.plot();
```



In [ ]:
```
# Spliting the data
X = data['Var_X'].values.reshape(-1, 1)
y = data['Var_Y'].values
display(X, y)
```

```
array([[-0.33532],
       [ 0.0216 ],
       [-1.19438],
       [-0.65046],
       [-0.28001],
       [ 1.93258],
       [ 1.2262 ],
       [ 0.74727],
       [ 3.32853],
       [ 2.87457],
       [-1.48662],
       [ 0.37629],
       [ 1.43918],
       [ 0.24183],
       [-2.7914 ],
       [ 1.08176],
       [ 2.81555],
       [ 0.54924],
       [ 2.36449],
       [-1.01925]])
array([  6.66854,   3.86398,   5.16161,   8.43823,   5.57201, -11.1327 ,
        -5.31226,  -4.63725,   3.8065 ,  -6.06084,   7.22328,   2.38887,
        -7.13415,   2.00412,   4.29794,  -5.86553,  -5.20711,  -3.52863,
       -10.16202,   5.31123])
```

In [ ]:
```
# Create polynomial features
# default degree is 2
poly_features_D2 = PolynomialFeatures()
poly_features_D3 = PolynomialFeatures(3)
poly_features_D4 = PolynomialFeatures(4)
poly_features_D5 = PolynomialFeatures(5)



x2 = poly_features_D2.fit_transform(X)
x3 = poly_features_D3.fit_transform(X)
x4 = poly_features_D4.fit_transform(X)
x5 = poly_features_D5.fit_transform(X)
```

```
# displaying x2 , x3 and x4 just for example
display(pd.DataFrame(x2), pd.DataFrame(x3), pd.DataFrame(x4) )
```

	0	1	2
0	1.0	-0.33532	0.112440
1	1.0	0.02160	0.000467
2	1.0	-1.19438	1.426544
3	1.0	-0.65046	0.423098
4	1.0	-0.28001	0.078406
5	1.0	1.93258	3.734865
6	1.0	1.22620	1.503566
7	1.0	0.74727	0.558412
8	1.0	3.32853	11.079112
9	1.0	2.87457	8.263153
10	1.0	-1.48662	2.210039
11	1.0	0.37629	0.141594
12	1.0	1.43918	2.071239
13	1.0	0.24183	0.058482
14	1.0	-2.79140	7.791914
15	1.0	1.08176	1.170205
16	1.0	2.81555	7.927322
17	1.0	0.54924	0.301665
18	1.0	2.36449	5.590813
19	1.0	-1.01925	1.038871

	0	1	2	3
0	1.0	-0.33532	0.112440	-0.037703
1	1.0	0.02160	0.000467	0.000010
2	1.0	-1.19438	1.426544	-1.703835
3	1.0	-0.65046	0.423098	-0.275208
4	1.0	-0.28001	0.078406	-0.021954
5	1.0	1.93258	3.734865	7.217926
6	1.0	1.22620	1.503566	1.843673
7	1.0	0.74727	0.558412	0.417285
8	1.0	3.32853	11.079112	36.877157
9	1.0	2.87457	8.263153	23.753011
10	1.0	-1.48662	2.210039	-3.285488
11	1.0	0.37629	0.141594	0.053280
12	1.0	1.43918	2.071239	2.980886
13	1.0	0.24183	0.058482	0.014143
14	1.0	-2.79140	7.791914	-21.750349
15	1.0	1.08176	1.170205	1.265881
16	1.0	2.81555	7.927322	22.319771
17	1.0	0.54924	0.301665	0.165686
18	1.0	2.36449	5.590813	13.219421
19	1.0	-1.01925	1.038871	-1.058869

	0	1	2	3	4
0	1.0	-0.33532	0.112440	-0.037703	1.264264e-02
1	1.0	0.02160	0.000467	0.000010	2.176782e-07
2	1.0	-1.19438	1.426544	-1.703835	2.035027e+00
3	1.0	-0.65046	0.423098	-0.275208	1.790121e-01
4	1.0	-0.28001	0.078406	-0.021954	6.147438e-03
5	1.0	1.93258	3.734865	7.217926	1.394922e+01
6	1.0	1.22620	1.503566	1.843673	2.260712e+00
7	1.0	0.74727	0.558412	0.417285	3.118245e-01
8	1.0	3.32853	11.079112	36.877157	1.227467e+02
9	1.0	2.87457	8.263153	23.753011	6.827969e+01
10	1.0	-1.48662	2.210039	-3.285488	4.884272e+00
11	1.0	0.37629	0.141594	0.053280	2.004891e-02
12	1.0	1.43918	2.071239	2.980886	4.290031e+00
13	1.0	0.24183	0.058482	0.014143	3.420115e-03
14	1.0	-2.79140	7.791914	-21.750349	6.071392e+01
15	1.0	1.08176	1.170205	1.265881	1.369379e+00
16	1.0	2.81555	7.927322	22.319771	6.284243e+01
17	1.0	0.54924	0.301665	0.165686	9.100152e-02
18	1.0	2.36449	5.590813	13.219421	3.125719e+01
19	1.0	-1.01925	1.038871	-1.058869	1.079252e+00

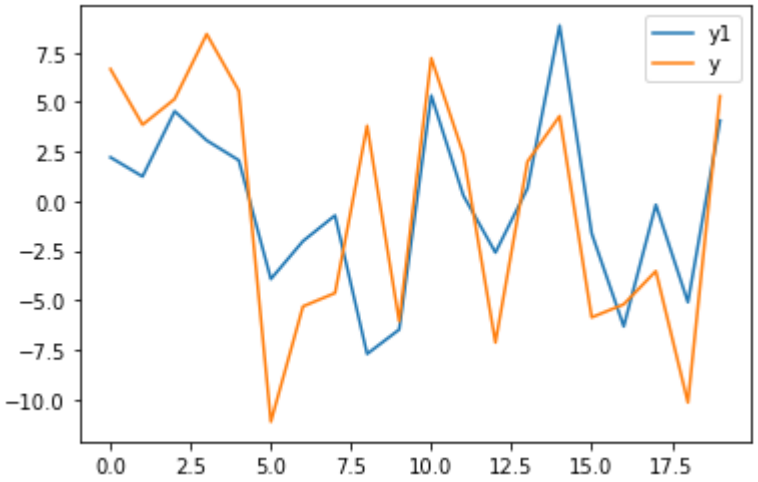
In [ ]:
```
# showing the different degrees
# 1. the Linear
Linear = LinearRegression()
Linear.fit(X, y)
```

Out[ ]:
```
LinearRegression()
```

In [ ]:
```
# I know I didn't split the data into trainning and testing
# as this is just a test
y1 = Linear.predict(X)
y1
```

Out[ ]:
```
array([ 2.21611784,  1.24911677,  4.54356358,  3.06992462,  2.06626681,
       -3.92828908, -2.01449828, -0.71693631, -7.71032746, -6.48041655,
        5.33532754,  0.28815743, -2.59152363,  0.65244904,  8.87035962,
       -1.62316792, -6.32051404, -0.1804149 , -5.09846008,  4.069085  ])
```

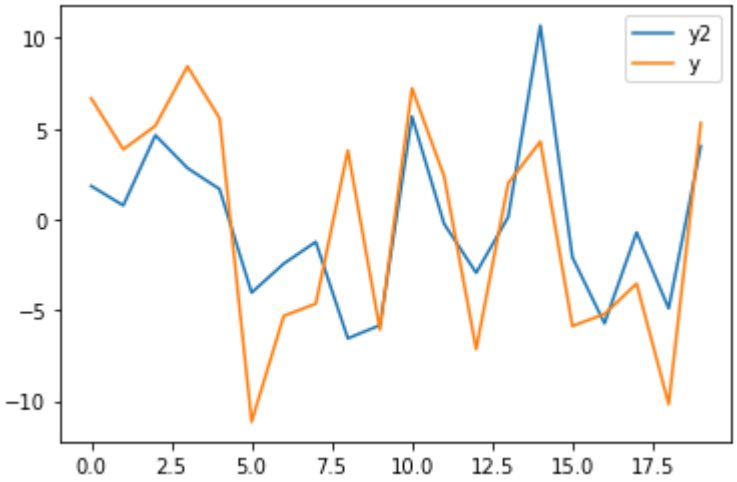
In [ ]:
```
pd.DataFrame({'y1': y1, 'y': y}).plot();
```



In [ ]:
```
Linear.fit(x2, y)
y2 =Linear.predict(x2)
y2
```

Out[ ]:
```
array([ 1.84578332,  0.77610159,  4.64343625,  2.83547842,  1.67645806,
       -4.0255836 , -2.43238113, -1.23095514, -6.54737621, -5.81861156,
        5.66700621, -0.23299811, -2.93518632,  0.14321961, 10.6819701 ,
       -2.08036143, -5.71739853, -0.70555706, -4.89474699,  4.04752251])
```

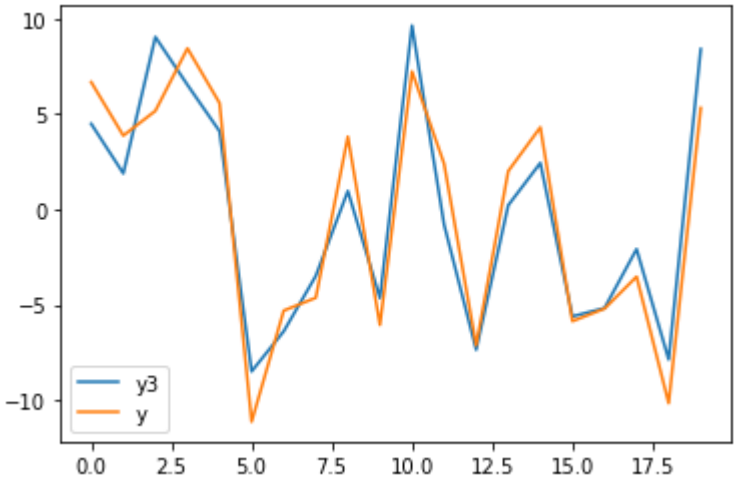
In [ ]:
```
pd.DataFrame({'y2': y2, 'y': y}).plot();
```



In [ ]:
```
Linear.fit(x3, y)
y3 =Linear.predict(x3)
y3
```

Out[ ]:
```
array([ 4.48222683,  1.88230375,  9.03265795,  6.52198786,  4.09463527,
       -8.49556163, -6.39672831, -3.49169832,  0.95776844, -4.65092976,
        9.63289367, -0.80305802, -7.36089053,  0.21346967,  2.43039622,
       -5.6145034 , -5.17972144, -2.08593807, -7.8719917 ,  8.3985015 ])
```

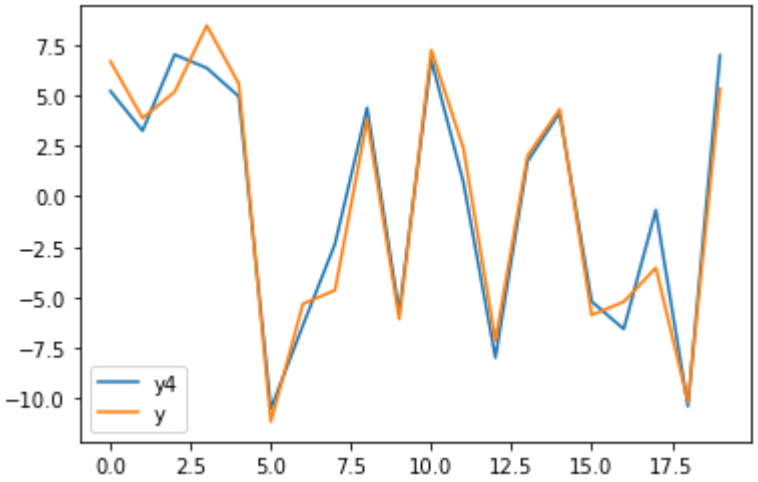
In [ ]:
```
pd.DataFrame({'y3': y3, 'y': y}).plot();
```



In [ ]:
```
Linear.fit(x4, y)
y4 =Linear.predict(x4)
y4
```

Out[ ]:
```
array([  5.19641403,   3.23885556,   7.00399695,   6.34330534,
         4.93719303, -10.48940158,  -6.37026049,  -2.34493858,
         4.37232952,  -5.7001477 ,   6.7478636 ,   0.7092552 ,
        -7.96880783,   1.72612006,   4.12622364,  -5.19188792,
        -6.55748759,  -0.67958875, -10.36912438,   6.96590788])
```

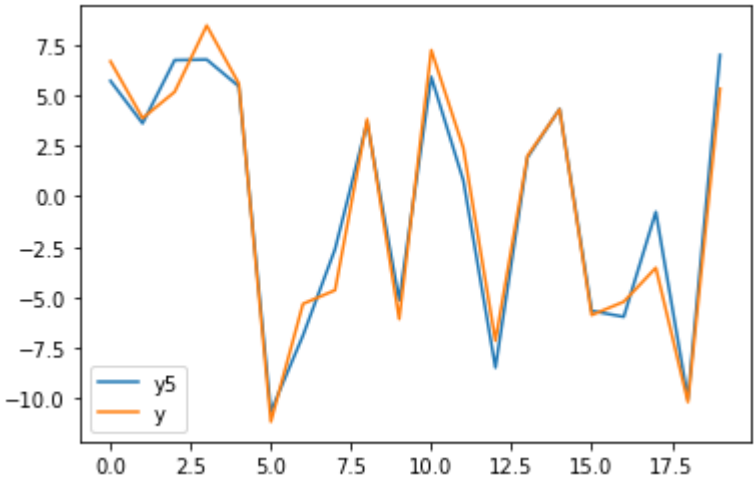
In [ ]:
```
pd.DataFrame({'y4': y4, 'y': y}).plot();
```



In [ ]:
```
Linear.fit(x5, y)
y5 =Linear.predict(x5)
y5
```

Out[ ]:
```
array([  5.70182022,   3.60634262,   6.73006389,   6.75796177,
         5.4361751 , -10.63037595,  -6.87602358,  -2.60656549,
         3.64317897,  -5.12968991,   5.90070852,   0.78268161,
        -8.46091817,   1.92155724,   4.31346811,  -5.65979089,
        -5.95116748,  -0.76792072,  -9.99727319,   6.98158735])
```

In [ ]:
```
pd.DataFrame({'y5': y5, 'y': y}).plot();
# welcome to the overfitting area 👋
```


7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
y5
y
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5