finalproject_Echo

May 13, 2019

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn
import math
```

In [2]:
```
XandY = pd.read_csv('X_and_Y_7-Day Lag.csv')
```

In [3]:
```
XandY
```

Out[3]:
```
      Shanghai  Shanghai 2  Shanghai 2.1  Shanghai 2.2  Shanghai 2.3  \
0          NaN         NaN           NaN           NaN           NaN
1       104.39         NaN           NaN           NaN           NaN
2       109.13      104.39           NaN           NaN           NaN
3       114.55      109.13        104.39           NaN           NaN
4       120.25      114.55        109.13        104.39           NaN
5       125.27      120.25        114.55        109.13        104.39
6       125.28      125.27        120.25        114.55        109.13
7       126.45      125.28        125.27        120.25        114.55
8       127.61      126.45        125.28        125.27        120.25
9       128.84      127.61        126.45        125.28        125.27
10      130.14      128.84        127.61        126.45        125.28
11      131.44      130.14        128.84        127.61        126.45
12      132.06      131.44        130.14        128.84        127.61
13      132.68      132.06        131.44        130.14        128.84
14      133.34      132.68        132.06        131.44        130.14
15      133.97      133.34        132.68        132.06        131.44
16      134.60      133.97        133.34        132.68        132.06
17      134.67      134.60        133.97        133.34        132.68
18      134.74      134.67        134.60        133.97        133.34
19      134.24      134.74        134.67        134.60        133.97
20      134.25      134.24        134.74        134.67        134.60
21      134.24      134.25        134.24        134.74        134.67
22      134.24      134.24        134.25        134.24        134.74
23      133.72      134.24        134.24        134.25        134.24
24      133.17      133.72        134.24        134.24        134.25
25      132.61      133.17        133.72        134.24        134.24
26      132.05      132.61        133.17        133.72        134.24
```

27	131.46	132.05	132.61	133.17	133.72
28	130.95	131.46	132.05	132.61	133.17
29	130.44	130.95	131.46	132.05	132.61
...	...	...	...	...	...
6865	2535.77	2553.83	2535.10	2544.34	2526.46
6866	2570.34	2535.77	2553.83	2535.10	2544.34
6867	2570.42	2570.34	2535.77	2553.83	2535.10
6868	2559.64	2570.42	2570.34	2535.77	2553.83
6869	2596.01	2559.64	2570.42	2570.34	2535.77
6870	2610.51	2596.01	2559.64	2570.42	2570.34
6871	2579.70	2610.51	2596.01	2559.64	2570.42
6872	2581.00	2579.70	2610.51	2596.01	2559.64
6873	2591.69	2581.00	2579.70	2610.51	2596.01
6874	2601.72	2591.69	2581.00	2579.70	2610.51
6875	2596.98	2601.72	2591.69	2581.00	2579.70
6876	2594.25	2596.98	2601.72	2591.69	2581.00
6877	2575.58	2594.25	2596.98	2601.72	2591.69
6878	2584.57	2575.58	2594.25	2596.98	2601.72
6879	2618.23	2584.57	2575.58	2594.25	2596.98
6880	2653.90	2618.23	2584.57	2575.58	2594.25
6881	2671.89	2653.90	2618.23	2584.57	2575.58
6882	2721.07	2671.89	2653.90	2618.23	2584.57
6883	2719.70	2721.07	2671.89	2653.90	2618.23
6884	2682.39	2719.70	2721.07	2671.89	2653.90
6885	2754.36	2682.39	2719.70	2721.07	2671.89
6886	2755.65	2754.36	2682.39	2719.70	2721.07
6887	NaN	2755.65	2754.36	2682.39	2719.70
6888	NaN	NaN	2755.65	2754.36	2682.39
6889	NaN	NaN	NaN	2755.65	2754.36
6890	NaN	NaN	NaN	NaN	2755.65
6891	NaN	NaN	NaN	NaN	NaN
6892	NaN	NaN	NaN	NaN	NaN
6893	NaN	NaN	NaN	NaN	NaN
6894	NaN	NaN	NaN	NaN	NaN

	Shanghai 2.4	Shanghai 2.5	DOW	DOW.1	DOW.2 \
0	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	921.340027	NaN	NaN
2	NaN	NaN	925.500000	921.340027	NaN
3	NaN	NaN	921.729980	925.500000	921.340027
4	NaN	NaN	925.690002	921.729980	925.500000
5	NaN	NaN	920.409973	925.690002	921.729980
6	104.39	NaN	920.369995	920.409973	925.690002
7	109.13	104.39	920.609985	920.369995	920.409973
8	114.55	109.13	914.890015	920.609985	920.369995
9	120.25	114.55	907.070007	914.890015	920.609985
10	125.27	120.25	905.599976	907.070007	914.890015
11	125.28	125.27	891.429993	905.599976	907.070007

12	126.45	125.28	890.739990	891.429993	905.599976
13	127.61	126.45	880.820007	890.739990	891.429993
14	128.84	127.61	889.229980	880.820007	890.739990
15	130.14	128.84	889.099976	889.229980	880.820007
16	131.44	130.14	882.070007	889.099976	889.229980
17	132.06	131.44	883.900024	882.070007	889.099976
18	132.68	132.06	890.530029	883.900024	882.070007
19	133.34	132.68	932.859985	890.530029	883.900024
20	133.97	133.34	944.880005	932.859985	890.530029
21	134.60	133.97	939.210022	944.880005	932.859985
22	134.67	134.60	931.320007	939.210022	944.880005
23	134.74	134.67	937.140015	931.320007	939.210022
24	134.24	134.74	954.210022	937.140015	931.320007
25	134.25	134.24	957.820007	954.210022	937.140015
26	134.24	134.25	955.179993	957.820007	954.210022
27	134.24	134.24	955.659973	955.179993	957.820007
28	133.72	134.24	975.520020	955.659973	955.179993
29	133.17	133.72	981.479980	975.520020	955.659973
...	...	...	...	...	...
6865	2533.09	2514.87	8239.509766	8261.849609	8234.610352
6866	2526.46	2533.09	8039.109863	8239.509766	8261.849609
6867	2544.34	2526.46	7899.600098	8039.109863	8239.509766
6868	2535.10	2544.34	8085.709961	7899.600098	8039.109863
6869	2553.83	2535.10	7993.259766	8085.709961	7899.600098
6870	2535.77	2553.83	7991.089844	7993.259766	8085.709961
6871	2570.34	2535.77	8092.529785	7991.089844	7993.259766
6872	2570.42	2570.34	7949.359863	8092.529785	7991.089844
6873	2559.64	2570.42	8072.439941	7949.359863	8092.529785
6874	2596.01	2559.64	8118.490234	8072.439941	7949.359863
6875	2610.51	2596.01	8189.089844	8118.490234	8072.439941
6876	2579.70	2610.51	8003.240234	8189.089844	8118.490234
6877	2581.00	2579.70	8007.479980	8003.240234	8189.089844
6878	2591.69	2581.00	8106.609863	8007.479980	8003.240234
6879	2601.72	2591.69	8054.729980	8106.609863	8007.479980
6880	2596.98	2601.72	8133.740234	8054.729980	8106.609863
6881	2594.25	2596.98	8119.729980	8133.740234	8054.729980
6882	2575.58	2594.25	8229.530273	8119.729980	8133.740234
6883	2584.57	2575.58	8280.209961	8229.530273	8119.729980
6884	2618.23	2584.57	8307.450195	8280.209961	8229.530273
6885	2653.90	2618.23	8268.690430	8307.450195	8280.209961
6886	2671.89	2653.90	8199.370117	8268.690430	8307.450195
6887	2721.07	2671.89	NaN	8199.370117	8268.690430
6888	2719.70	2721.07	NaN	NaN	8199.370117
6889	2682.39	2719.70	NaN	NaN	NaN
6890	2754.36	2682.39	NaN	NaN	NaN
6891	2755.65	2754.36	NaN	NaN	NaN
6892	NaN	2755.65	NaN	NaN	NaN
6893	NaN	NaN	NaN	NaN	NaN

6894	NaN	NaN	NaN	NaN	NaN

	...	DOW.6	Y	Y.1	Y.2	Y.3	Y.4	Y.5	\
0	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	...	NaN	5.2400	NaN	NaN	NaN	NaN	NaN	
2	...	NaN	5.2400	5.2400	NaN	NaN	NaN	NaN	
3	...	NaN	5.2400	5.2400	5.2400	NaN	NaN	NaN	
4	...	NaN	5.2400	5.2400	5.2400	5.2400	NaN	NaN	
5	...	NaN	5.2400	5.2400	5.2400	5.2400	5.2400	NaN	
6	...	NaN	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
7	...	921.340027	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
8	...	925.500000	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
9	...	921.729980	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
10	...	925.690002	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
11	...	920.409973	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
12	...	920.369995	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
13	...	920.609985	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
14	...	914.890015	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
15	...	907.070007	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
16	...	905.599976	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
17	...	891.429993	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
18	...	890.739990	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
19	...	880.820007	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
20	...	889.229980	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
21	...	889.099976	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
22	...	882.070007	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
23	...	883.900024	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
24	...	890.530029	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
25	...	932.859985	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
26	...	944.880005	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
27	...	939.210022	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
28	...	931.320007	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
29	...	937.140015	5.2400	5.2400	5.2400	5.2400	5.2400	5.2400	
...	...	...	...	...	...	...	...	...	
6865	...	8320.969727	6.6395	6.6340	6.6289	6.6397	6.6631	6.6170	
6866	...	8244.429688	6.6395	6.6395	6.6340	6.6289	6.6397	6.6631	
6867	...	8269.990234	6.5758	6.6395	6.6395	6.6340	6.6289	6.6397	
6868	...	8316.879883	6.6122	6.5758	6.6395	6.6395	6.6340	6.6289	
6869	...	8234.610352	6.6295	6.6122	6.5758	6.6395	6.6395	6.6340	
6870	...	8261.849609	6.6763	6.6295	6.6122	6.5758	6.6395	6.6395	
6871	...	8239.509766	6.6629	6.6763	6.6295	6.6122	6.5758	6.6395	
6872	...	8039.109863	6.6899	6.6629	6.6763	6.6295	6.6122	6.5758	
6873	...	7899.600098	6.6899	6.6899	6.6629	6.6763	6.6295	6.6122	
6874	...	8085.709961	6.6899	6.6899	6.6899	6.6629	6.6763	6.6295	
6875	...	7993.259766	6.6865	6.6899	6.6899	6.6899	6.6629	6.6763	
6876	...	7991.089844	6.7017	6.6865	6.6899	6.6899	6.6899	6.6629	
6877	...	8092.529785	6.7164	6.7017	6.6865	6.6899	6.6899	6.6899	
6878	...	7949.359863	6.7697	6.7164	6.7017	6.6865	6.6899	6.6899	

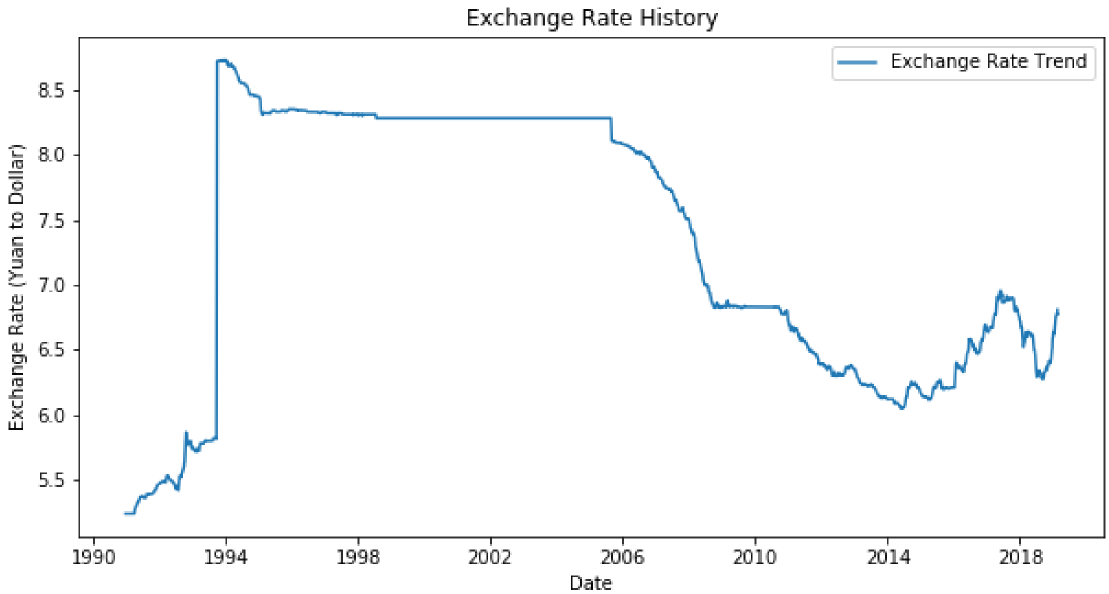
6879	...	8072.439941	6.7658	6.7697	6.7164	6.7017	6.6865	6.6899
6880	...	8118.490234	6.7658	6.7658	6.7697	6.7164	6.7017	6.6865
6881	...	8189.089844	6.7658	6.7658	6.7658	6.7697	6.7164	6.7017
6882	...	8003.240234	6.7926	6.7658	6.7658	6.7658	6.7697	6.7164
6883	...	8007.479980	6.7911	6.7926	6.7658	6.7658	6.7658	6.7697
6884	...	8106.609863	6.7657	6.7911	6.7926	6.7658	6.7658	6.7658
6885	...	8054.729980	6.7864	6.7657	6.7911	6.7926	6.7658	6.7658
6886	...	8133.740234	6.8096	6.7864	6.7657	6.7911	6.7926	6.7658
6887	...	8119.729980	NaN	6.8096	6.7864	6.7657	6.7911	6.7926
6888	...	8229.530273	NaN	NaN	6.8096	6.7864	6.7657	6.7911
6889	...	8280.209961	NaN	NaN	NaN	6.8096	6.7864	6.7657
6890	...	8307.450195	NaN	NaN	NaN	NaN	6.8096	6.7864
6891	...	8268.690430	NaN	NaN	NaN	NaN	NaN	6.8096
6892	...	8199.370117	NaN	NaN	NaN	NaN	NaN	NaN
6893	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6894	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN

	Y.6	Y.7	Date
0	NaN	5.2400	20-Dec-90
1	NaN	5.2400	21-Dec-90
2	NaN	5.2400	24-Dec-90
3	NaN	5.2400	25-Dec-90
4	NaN	5.2400	26-Dec-90
5	NaN	5.2400	27-Dec-90
6	NaN	5.2400	28-Dec-90
7	5.2400	5.2400	31-Dec-90
8	5.2400	5.2400	2-Jan-91
9	5.2400	5.2400	3-Jan-91
10	5.2400	5.2400	4-Jan-91
11	5.2400	5.2400	7-Jan-91
12	5.2400	5.2400	8-Jan-91
13	5.2400	5.2400	9-Jan-91
14	5.2400	5.2400	10-Jan-91
15	5.2400	5.2400	11-Jan-91
16	5.2400	5.2400	14-Jan-91
17	5.2400	5.2400	15-Jan-91
18	5.2400	5.2400	16-Jan-91
19	5.2400	5.2400	17-Jan-91
20	5.2400	5.2400	18-Jan-91
21	5.2400	5.2400	21-Jan-91
22	5.2400	5.2400	22-Jan-91
23	5.2400	5.2400	23-Jan-91
24	5.2400	5.2400	24-Jan-91
25	5.2400	5.2400	25-Jan-91
26	5.2400	5.2400	28-Jan-91
27	5.2400	5.2400	29-Jan-91
28	5.2400	5.2400	30-Jan-91
29	5.2400	5.2400	31-Jan-91

```
...       ...     ...        ...
6865  6.6170  6.6395  15-Jan-19
6866  6.6170  6.5758  16-Jan-19
6867  6.6631  6.6122  17-Jan-19
6868  6.6397  6.6295  18-Jan-19
6869  6.6289  6.6763  21-Jan-19
6870  6.6340  6.6629  22-Jan-19
6871  6.6395  6.6899  23-Jan-19
6872  6.6395  6.6899  24-Jan-19
6873  6.5758  6.6899  25-Jan-19
6874  6.6122  6.6865  28-Jan-19
6875  6.6295  6.7017  29-Jan-19
6876  6.6763  6.7164  30-Jan-19
6877  6.6629  6.7697  31-Jan-19
6878  6.6899  6.7658   1-Feb-19
6879  6.6899  6.7658  11-Feb-19
6880  6.6899  6.7658  12-Feb-19
6881  6.6865  6.7926  13-Feb-19
6882  6.7017  6.7911  14-Feb-19
6883  6.7164  6.7657  15-Feb-19
6884  6.7697  6.7864  18-Feb-19
6885  6.7658  6.8096  19-Feb-19
6886  6.7658     NaN        NaN
6887  6.7658     NaN        NaN
6888  6.7926     NaN        NaN
6889  6.7911     NaN        NaN
6890  6.7657     NaN        NaN
6891  6.7864     NaN        NaN
6892  6.8096     NaN        NaN
6893     NaN     NaN        NaN
6894     NaN     NaN        NaN

[6895 rows x 23 columns]
```

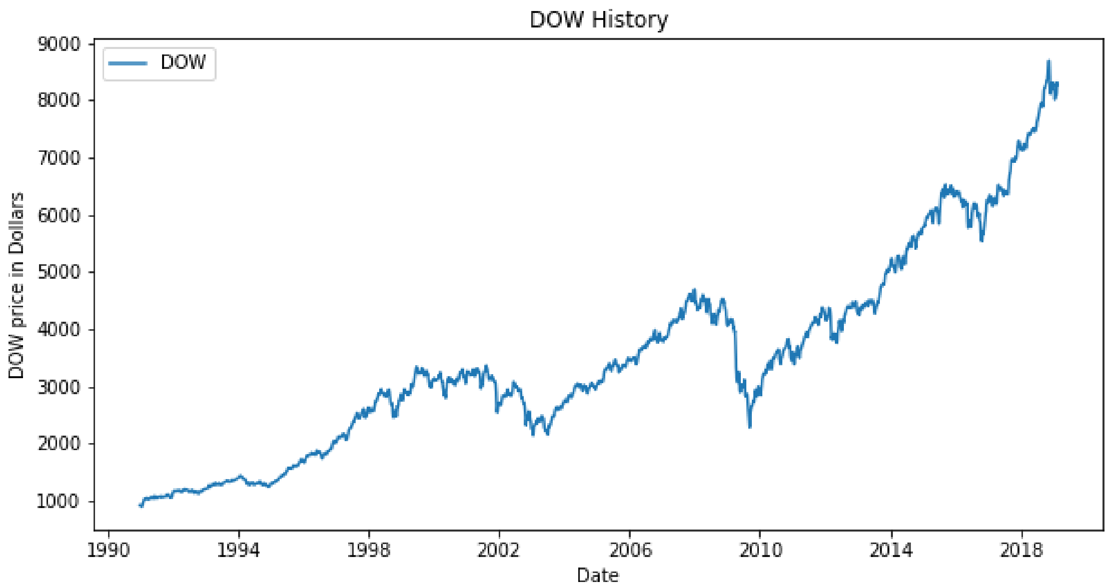
In [4]:
```python
#Moving Average
tenp = (6886-3700)/10.0
endLoc = int(6886-tenp)
def movingAverage(Y):
    ma_Y = np.copy(Y)
    for i in range(Y.size-3):
        if i > 3:
            ma_Y[i][0] = (Y[i-3][0]+Y[i-2][0]+Y[i-1][0]+Y[i][0]+Y[i+1][0]+Y[i+2][0]+Y[i
    return(ma_Y)

Y = np.array(XandY.iloc[0:6886,21])
Y = np.reshape(Y, (Y.size,1))
dow = np.array(XandY.iloc[0:6886,7])
dow = np.reshape(dow, (dow.size,1))
```

```
shangCom = np.array(XandY.iloc[0:6886,0])
shangCom = np.reshape(shangCom, (dow.size,1))


scatterRange = np.array(XandY.iloc[0:6886,22])
scatterRange = pd.to_datetime(scatterRange)

plt.figure(figsize=(10,5))
plt.plot(scatterRange,movingAverage(Y), label='Exchange Rate Trend')
plt.legend()
plt.title('Exchange Rate History')
plt.ylabel('Exchange Rate (Yuan to Dollar)')
plt.xlabel('Date')




plt.legend()
plt.savefig('exch_rate_Hist_all.png')
plt.show()
```



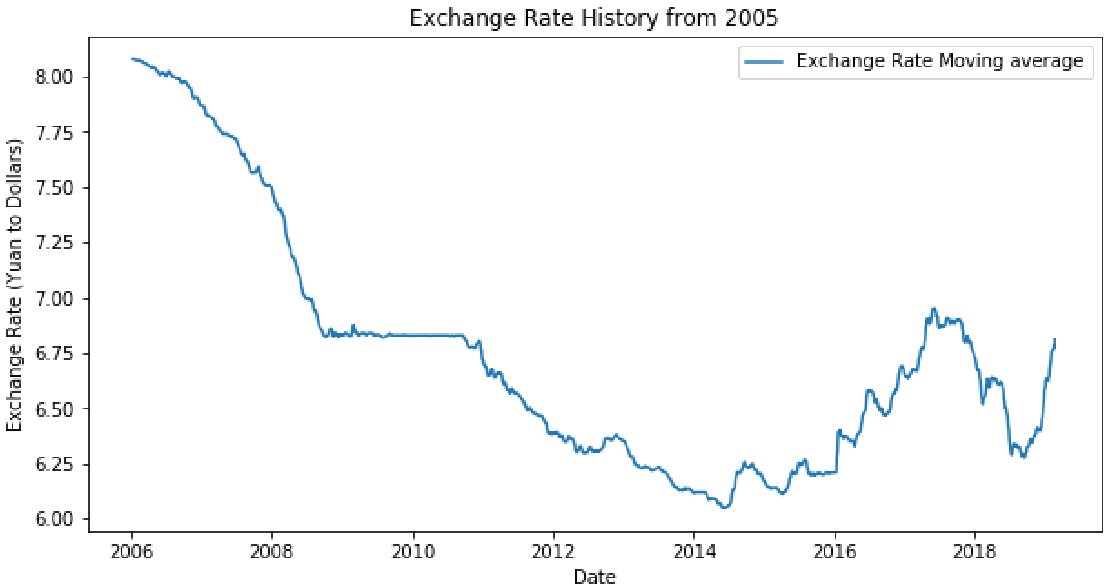

```
In [5]: plt.figure(figsize=(10,5))
        plt.plot(scatterRange,movingAverage(dow), label='DOW')
        plt.title('DOW History')
        plt.ylabel('DOW price in Dollars')
        plt.xlabel('Date')
        plt.legend()
```

```
plt.savefig('Dow_Hist_all.png')
plt.show()
```




```
In [6]: plt.figure(figsize=(10,5))
        plt.plot(scatterRange,movingAverage(shangCom), label='Shanghai Composite Index')
        plt.legend()
        plt.title('Shanghai Composite Index History')
        plt.ylabel('SSE Composite Index price in Dollars')
        plt.xlabel('Date')
        plt.savefig('Shang_Hist_all.png')
        plt.show()
```

In [7]:
```
Y = np.array(XandY.iloc[3700:(6886),21])
Y = np.reshape(Y, (Y.size,1))

scatterRange = np.array(XandY.iloc[3700:(6886),22])
scatterRange = pd.to_datetime(scatterRange)


plt.figure(figsize=(10,5))
plt.plot(scatterRange,movingAverage(Y), label='Exchange Rate Moving average')

plt.legend()
plt.title('Exchange Rate History from 2005')
plt.ylabel('Exchange Rate (Yuan to Dollars)')
plt.xlabel('Date')
plt.savefig('exch_rate_Hist_2005.png')
plt.show()
```



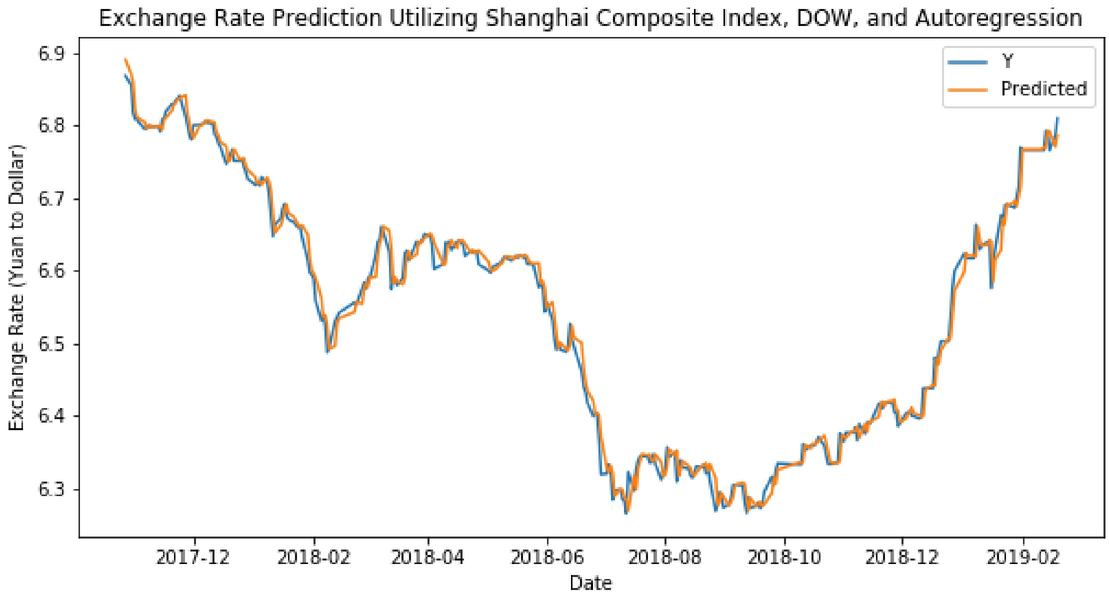
In [ ]:
```
Y = np.array(XandY.iloc[3700:(6886),21])
Y = np.reshape(Y, (Y.size,1))


scatterRange = np.array(XandY.iloc[3700:(6886),22])
scatterRange = pd.to_datetime(scatterRange)


plt.figure(figsize=(10,5))
plt.plot(scatterRange,movingAverage(Y), label='Exchange Rate Moving average')

plt.legend()
plt.title('Exchange Rate History from 2005')
plt.ylabel('Exchange Rate (Yuan to Dollars)')
plt.xlabel('Date')
plt.savefig('exch_rate_Hist_2005.png')
plt.show()
```

In [8]:
```
def linRegress(XandY):

    tenp = (6886-3700)/10.0

    endLoc = int(6886-tenp)

    X = (np.array(XandY.iloc[3700:endLoc,0:20], dtype = float))

    #oneS = np.ones(np.size(X, axis = 0))
    #oneS = np.reshape(oneS, (oneS.size,1))
```

```
        #X = np.concatenate((oneS,X), axis = 1)
        #X = np.array(X,dtype=float)
        Y = np.array(XandY.iloc[3700:endLoc,21])
        Y = np.reshape(Y, (Y.size,1))
        xPrimeX = np.matmul(X.T,X)
        xPrimeXInv = np.linalg.inv(xPrimeX)
        #print(xPrimeXInv)
        xPrimeY = np.matmul(X.T,Y)
        coef = np.matmul(xPrimeXInv,xPrimeY)
        #print(coef)

        X = (np.array(XandY.iloc[(endLoc):(6886),0:20], dtype = float))
        Y = np.array(XandY.iloc[endLoc:(6886),21])
        Y = np.reshape(Y, (Y.size,1))

        predictedY = np.matmul(X,coef)
        scatterRange = np.array(XandY.iloc[endLoc:(6886),22])
        scatterRange = pd.to_datetime(scatterRange)


        plt.figure(figsize=(10,5))
        plt.plot(scatterRange,Y, label='Y')
        plt.plot(scatterRange,predictedY,label='Predicted')
        SSE = np.sum((Y-predictedY)**2)
        MSE = SSE/Y.size
        SD = math.sqrt(MSE)
        print('SSE: ',SSE)
        print('MSE: ',MSE)
        print('SD: ',SD)
        plt.title('Exchange Rate Prediction Utilizing Shanghai Composite Index, DOW, and Au
        plt.legend()
        plt.xlabel('Date')
        plt.ylabel('Exchange Rate (Yuan to Dollar)')
        plt.show()
```

```
In [9]: linRegress(XandY)
```

```
SSE:  0.0900962530893456
MSE:  0.0002824333952644063
SD:  0.01680575482578531
```

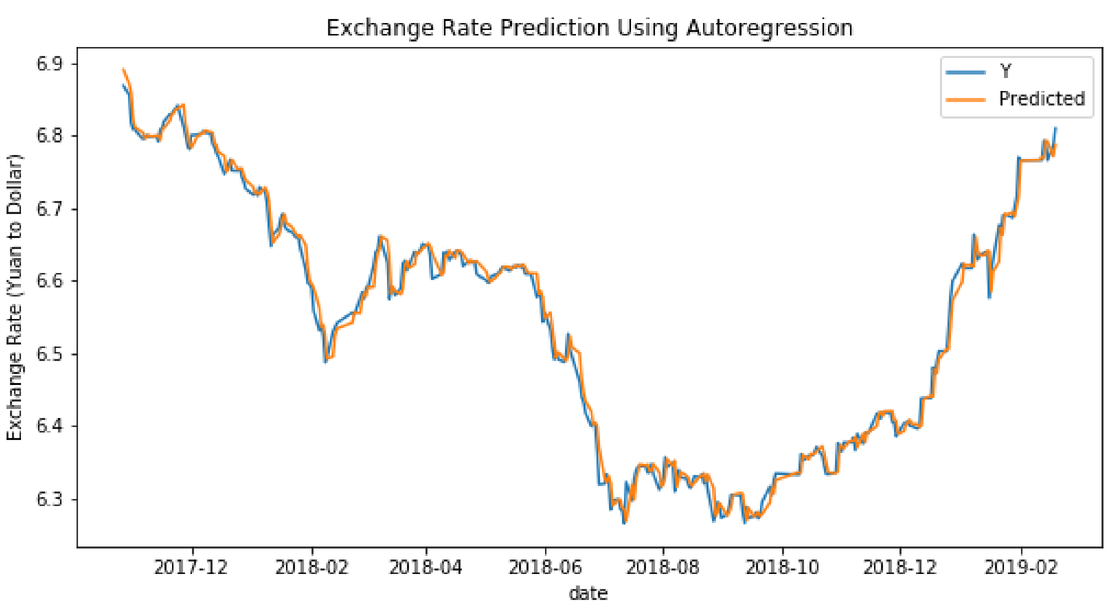


In [10]:
```
def linRegress(XandY):
    tenp = (6886-3700)/10.0
    endLoc = int(6886-tenp)

    X = (np.array(XandY.iloc[3700:endLoc,13:20], dtype = float))

    #oneS = np.ones(np.size(X, axis = 0))
    #oneS = np.reshape(oneS, (oneS.size,1))
    #X = np.concatenate((oneS,X), axis = 1)
    #X = np.array(X,dtype=float)
    Y = np.array(XandY.iloc[3700:endLoc,21])
    Y = np.reshape(Y, (Y.size,1))
    xPrimeX = np.matmul(X.T,X)
    xPrimeXInv = np.linalg.inv(xPrimeX)
    xPrimeY = np.matmul(X.T,Y)
    coef = np.matmul(xPrimeXInv,xPrimeY)
    X = (np.array(XandY.iloc[(endLoc):(6886),13:20], dtype = float))
    Y = np.array(XandY.iloc[endLoc:(6886),21])
    Y = np.reshape(Y, (Y.size,1))

    predictedY = np.matmul(X,coef)

    scatterRange = np.array(XandY.iloc[endLoc:(6886),22])
    scatterRange = pd.to_datetime(scatterRange)
    plt.figure(figsize=(10,5))
    plt.plot(scatterRange,Y, label='Y')
    plt.plot(scatterRange,predictedY,label='Predicted')
```

```
    SSE = np.sum((Y-predictedY)**2)
    MSE = SSE/Y.size
    SD = math.sqrt(MSE)
    print('SSE: ',SSE)
    print('MSE: ',MSE)
    print('SD: ',SD)
    plt.title('Exchange Rate Prediction Using Autoregression')
    plt.xlabel('date')
    plt.ylabel('Exchange Rate (Yuan to Dollar)')
    plt.legend()
    plt.show
```

```
In [11]: linRegress(XandY)
```

```
SSE:  0.09044357120366141
MSE:  0.00028352216678263764
SD:  0.016838116485600095
```

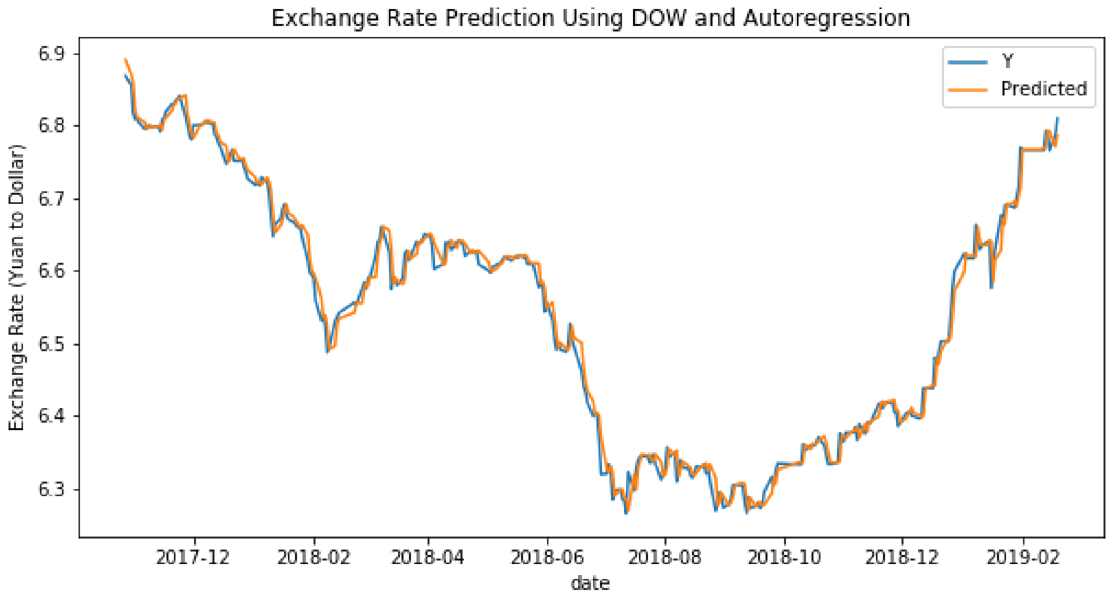



```
In [12]: def linRegress(XandY):
    tenp = (6886-3700)/10.0

    endLoc = int(6886-tenp)

    X = (np.array(XandY.iloc[3700:endLoc,6:20], dtype = float))

    #oneS = np.ones(np.size(X, axis = 0))
    #oneS = np.reshape(oneS, (oneS.size,1))
```

```
    #X = np.concatenate((oneS,X), axis = 1)
    #X = np.array(X,dtype=float)
    Y = np.array(XandY.iloc[3700:endLoc,21])
    Y = np.reshape(Y, (Y.size,1))
    xPrimeX = np.matmul(X.T,X)
    xPrimeXInv = np.linalg.inv(xPrimeX)

    xPrimeY = np.matmul(X.T,Y)
    coef = np.matmul(xPrimeXInv,xPrimeY)

    X = (np.array(XandY.iloc[(endLoc):(6886),6:20], dtype = float))
    Y = np.array(XandY.iloc[endLoc:(6886),21])
    Y = np.reshape(Y, (Y.size,1))

    predictedY = np.matmul(X,coef)

    scatterRange = np.array(XandY.iloc[endLoc:(6886),22])
    scatterRange = pd.to_datetime(scatterRange)

    plt.figure(figsize=(10,5))
    plt.plot(scatterRange,Y, label='Y')
    plt.plot(scatterRange,predictedY,label='Predicted')

    SSE = np.sum((Y-predictedY)**2)

    SSE = np.sum((Y-predictedY)**2)
    MSE = SSE/Y.size
    SD = math.sqrt(MSE)
    print('SSE: ',SSE)
    print('MSE: ',MSE)
    print('SD: ',SD)
    plt.title('Exchange Rate Prediction Using DOW and Autoregression')
    plt.xlabel('date')
    plt.ylabel('Exchange Rate (Yuan to Dollar)')
    plt.legend()
    plt.show

    plt.legend()
    plt.show
```

```
In [13]: linRegress(XandY)
```

```
SSE:  0.09017017947553893
MSE:  0.0002826651394217521
SD:  0.016812648197763257
```




```
In [14]: def linRegress(XandY):
             tenp = (6886-3700)/10.0

             endLoc = int(6886-tenp)

             X1 = (np.array(XandY.iloc[3700:endLoc,13:20], dtype = float))
             X2 = (np.array(XandY.iloc[3700:endLoc,0:6], dtype = float))
             X = np.concatenate((X1,X2),axis=1)

             #oneS = np.ones(np.size(X, axis = 0))
             #oneS = np.reshape(oneS, (oneS.size,1))
             #X = np.concatenate((oneS,X), axis = 1)
             #X = np.array(X,dtype=float)
             Y = np.array(XandY.iloc[3700:endLoc,21])
             Y = np.reshape(Y, (Y.size,1))
             xPrimeX = np.matmul(X.T,X)
             xPrimeXInv = np.linalg.inv(xPrimeX)

             xPrimeY = np.matmul(X.T,Y)
             coef = np.matmul(xPrimeXInv,xPrimeY)


             X1 = (np.array(XandY.iloc[(endLoc):(6886),13:20], dtype = float))
             X2 = (np.array(XandY.iloc[(endLoc):(6886),0:6], dtype = float))
             X = np.concatenate((X1,X2),axis=1)

             Y = np.array(XandY.iloc[endLoc:(6886),21])
```

```
    Y = np.reshape(Y, (Y.size,1))

    predictedY = np.matmul(X,coef)
    scatterRange = np.array(XandY.iloc[endLoc:(6886),22])
    scatterRange = pd.to_datetime(scatterRange)

    plt.figure(figsize=(10,5))
    plt.plot(scatterRange,Y, label='Y')
    plt.plot(scatterRange,predictedY,label='Predicted')

    SSE = np.sum((Y-predictedY)**2)
    MSE = SSE/Y.size
    SD = math.sqrt(MSE)

    plt.title("Exchange Rate Prediction Using Shanghai Composite Index and Autoregres
    plt.legend()
    plt.show
    print('SSE: ',SSE)
    print('MSE: ',MSE)
    print('SD: ',SD)
```

```
In [15]: linRegress(XandY)
```

```
SSE:  0.0903813451732948
MSE:  0.0002833271008567235
SD:  0.01683232309744331
```