08-Time-Series-with-Pandas-Project-Exercise-SET-TWO-Solutions

October 19, 2022

---

1 Time Series with Pandas Project Exercise

For this exercise, answer the questions below given the dataset: https://fred.stlouisfed.org/series/UMTMVS

This dataset is the Value of Manufacturers' Shipments for All Manufacturing Industries.

Import any necessary libraries.

[42]:
```
# CODE HERE
```

[43]:
```
import numpy as np
import pandas as pd
%matplotlib inline
```

Read in the data UMTMVS.csv file from the Data folder

[44]:
```
# CODE HERE
```

[45]:
```
df = pd.read_csv('../Data/UMTMVS.csv')
```

Check the head of the data

[46]:
```
# CODE HERE
```

[47]:
```
df.head()
```

[47]:

	DATE	UMTMVS
0	1992-01-01	209438.0
1	1992-02-01	232679.0
2	1992-03-01	249673.0
3	1992-04-01	239666.0
4	1992-05-01	243231.0

Set the DATE column as the index.

[48]:
```
# CODE HERE
```

[49]:
```
df = df.set_index('DATE')
```

[50]:
```
df.head()
```

[50]:
```
              UMTMVS
DATE
1992-01-01  209438.0
1992-02-01  232679.0
1992-03-01  249673.0
1992-04-01  239666.0
1992-05-01  243231.0
```

Check the data type of the index.

[51]:
```
# CODE HERE
```

[52]:
```
df.index
```

[52]:
```
Index(['1992-01-01', '1992-02-01', '1992-03-01', '1992-04-01', '1992-05-01',
       '1992-06-01', '1992-07-01', '1992-08-01', '1992-09-01', '1992-10-01',
       …
       '2018-04-01', '2018-05-01', '2018-06-01', '2018-07-01', '2018-08-01',
       '2018-09-01', '2018-10-01', '2018-11-01', '2018-12-01', '2019-01-01'],
      dtype='object', name='DATE', length=325)
```

Convert the index to be a datetime index. Note, there are many, many correct ways to do this!

[53]:
```
# CODE HERE
```

[54]:
```
df.index = pd.to_datetime(df.index)
```

[55]:
```
df.index
```

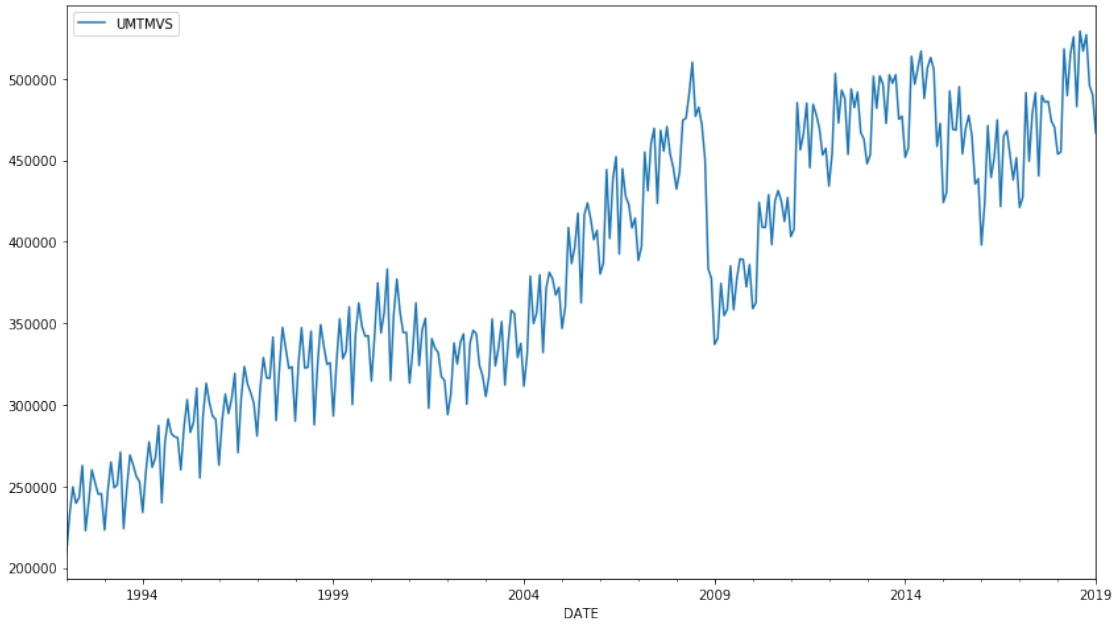
[55]:
```
DatetimeIndex(['1992-01-01', '1992-02-01', '1992-03-01', '1992-04-01',
               '1992-05-01', '1992-06-01', '1992-07-01', '1992-08-01',
               '1992-09-01', '1992-10-01',
               …
               '2018-04-01', '2018-05-01', '2018-06-01', '2018-07-01',
               '2018-08-01', '2018-09-01', '2018-10-01', '2018-11-01',
               '2018-12-01', '2019-01-01'],
              dtype='datetime64[ns]', name='DATE', length=325, freq=None)
```

Plot out the data, choose a reasonable figure size

[56]:
```
# CODE HERE
```

[69]:
```
df.plot(figsize=(14,8))
```

[69]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d10ba9bcc0>
```



What was the percent increase in value from Jan 2009 to Jan 2019?

[71]:
```
#CODE HERE
```

[76]:
```
100 * (df.loc['2019-01-01'] - df.loc['2009-01-01']) / df.loc['2009-01-01']
```

[76]:
```
UMTMVS    38.472149
dtype: float64
```

What was the percent decrease from Jan 2008 to Jan 2009?

[ ]:
```
#CODE HERE
```

[77]:
```
100 * (df.loc['2009-01-01'] - df.loc['2008-01-01']) / df.loc['2008-01-01']
```

[77]:
```
UMTMVS   -22.022775
dtype: float64
```

What is the month with the least value after 2005? HINT

[59]:
```
#CODE HERE
```

[61]:
```
df.loc['2005-01-01':].idxmin()
```

[61]:
```
UMTMVS   2009-01-01
dtype: datetime64[ns]
```

What 6 months have the highest value?

[68]:
```
# CODE HERE
```

[80]:
```
df.sort_values(by='UMTMVS',ascending=False).head(5)
```

[80]:

DATE	UMTMVS	Yearly Mean
2018-08-01	529157.0	490453.500000
2018-10-01	527031.0	496482.333333
2018-06-01	525660.0	483611.000000
2018-03-01	518285.0	474351.250000
2018-09-01	516992.0	493075.583333

How many millions of dollars in value was lost in 2008? (Another way of posing this question is what was the value difference between Jan 2008 and Jan 2009)

[17]:
```
# CODE HERE
```

[18]:
```
df.loc['2008-01-01'] - df.loc['2009-01-01']
```

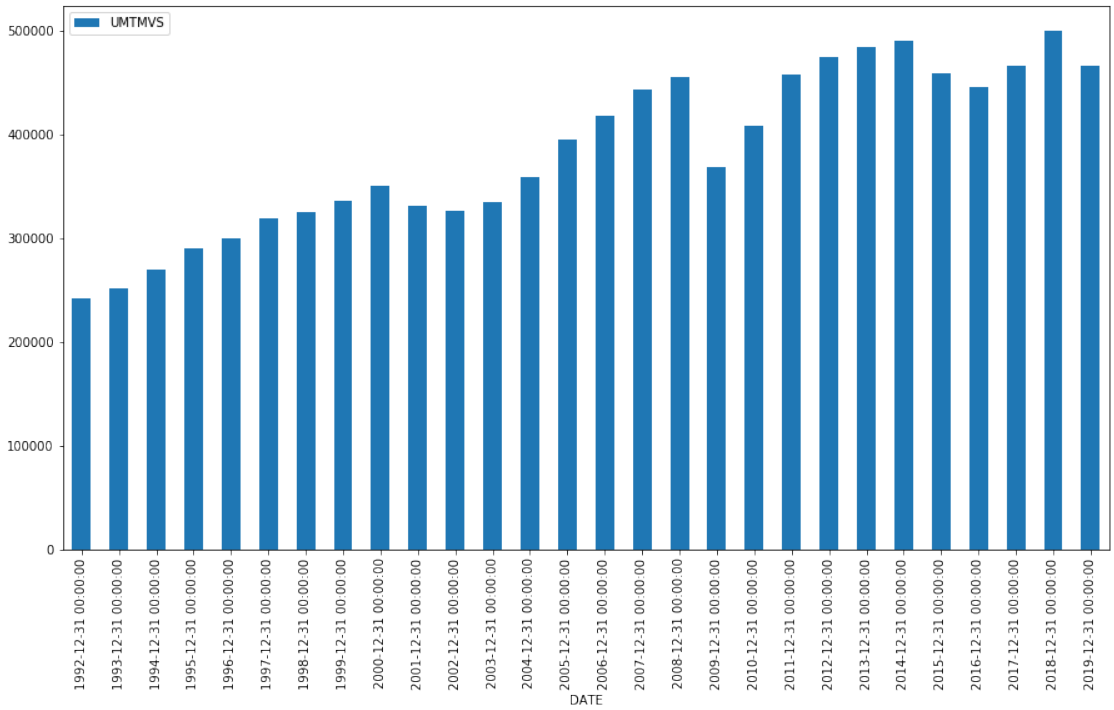
[18]:
```
UMTMVS    95206.0
dtype: float64
```

Create a bar plot showing the average value in millions of dollars per year

[19]:
```
# CODE HERE
```

[20]:
```
df.resample('Y').mean().plot.bar(figsize=(15,8))
```

[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d10a074588>
```



What year had the biggest increase in mean value from the previous year's mean value? (Lots of ways to get this answer!)

HINT for a useful method

[21]:
```
# CODE HERE
```

[22]:
```
yearly_data = df.resample('Y').mean()
yearly_data_shift = yearly_data.shift(1)
```

[23]:
```
yearly_data.head()
```

[23]:
```
                   UMTMVS
DATE
1992-12-31  242002.000000
1993-12-31  251708.083333
1994-12-31  269842.666667
1995-12-31  289973.083333
1996-12-31  299765.666667
```

[24]:
```
change = yearly_data - yearly_data_shift
```

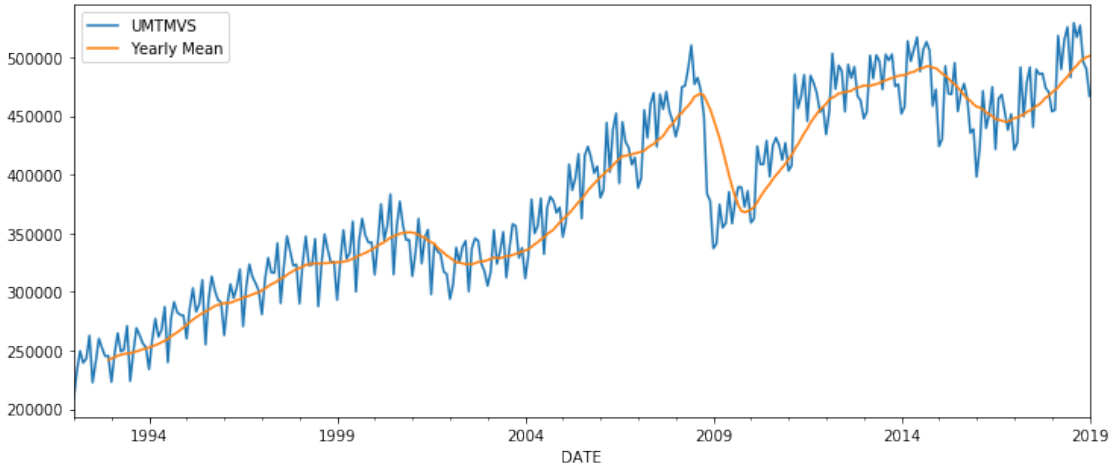
[25]:
```
change['UMTMVS'].idxmax()
```

[25]:
```
Timestamp('2011-12-31 00:00:00', freq='A-DEC')
```

Plot out the yearly rolling mean on top of the original data. Recall that this is monthly data and there are 12 months in a year!

[26]:
```
# CODE HERE
```

[78]:
```
df['Yearly Mean'] = df['UMTMVS'].rolling(window=12).mean()
df[['UMTMVS','Yearly Mean']].plot(figsize=(12,5)).
 ↪autoscale(axis='x',tight=True);
```



BONUS QUESTION (HARD).

Some month in 2008 the value peaked for that year. How many months did it take to surpass that 2008 peak? (Since it crashed immediately after this peak) There are many ways to get this answer. NOTE: I get 70 months as my answer, you may get 69 or 68, depending on whether or not you count the start and end months. Refer to the video solutions for full explanation on this.

[91]:
```
#CODE HERE
```

[97]:
```
df = pd.read_csv('../Data/UMTMVS.csv',index_col='DATE',parse_dates=True)
```

[98]:
```
df.head()
```

[98]:

DATE	UMTMVS
1992-01-01	209438.0
1992-02-01	232679.0
1992-03-01	249673.0
1992-04-01	239666.0
1992-05-01	243231.0

[99]:
```
df2008 = df.loc['2008-01-01':'2009-01-01']
```

[100]:
```
df2008.idxmax()
```

[100]:
```
UMTMVS   2008-06-01
dtype: datetime64[ns]
```

[101]:
```
df2008.max()
```

[101]:
```
UMTMVS    510081.0
dtype: float64
```

[105]:
```
df_post_peak = df.loc['2008-06-01':]
```

[106]:
```
df_post_peak[df_post_peak>=510081].dropna()
```

[106]:

DATE	UMTMVS
2008-06-01	510081.0
2014-03-01	513700.0
2014-06-01	516935.0
2014-09-01	512988.0
2018-03-01	518285.0
2018-05-01	515105.0
2018-06-01	525660.0
2018-08-01	529157.0
2018-09-01	516992.0
2018-10-01	527031.0

[108]:
```
len(df.loc['2008-06-01':'2014-03-01'])
```

[108]:
```
70
```

2 GREAT JOB!