

06-Pandas-Time-Series-Exercises-SET-ONE-Solutions

October 19, 2022

Copyright Pierian Data

For more information, visit us at www.pieriandata.com

1 Pandas Time Series Exercise Set #1 - Solution

For this set of exercises we'll use a dataset containing monthly milk production values in pounds per cow from January 1962 to December 1975.

IMPORTANT NOTE! Make sure you don't run the cells directly above the example output shown, otherwise you will end up writing over the example output!

```
[16]: # RUN THIS CELL
import pandas as pd
%matplotlib inline

df = pd.read_csv('../Data/monthly_milk_production.csv', encoding='utf8')
title = "Monthly milk production: pounds per cow. Jan '62 - Dec '75"

print(len(df))
print(df.head())
```

```
168
      Date  Production
0  1962-01         589
1  1962-02         561
2  1962-03         640
3  1962-04         656
4  1962-05         727
```

So df has 168 records and 2 columns.

1.0.1 1. What is the current data type of the Date column?

HINT: We show how to list column dtypes in the first set of DataFrame lectures.

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

```
[17]: # DON'T WRITE HERE
df.dtypes
```

```
[17]: Date          object
      Production    int64
      dtype: object
```

1.0.2 2. Change the Date column to a datetime format

```
[ ]:
```

```
[18]: # DON'T WRITE HERE
df['Date']=pd.to_datetime(df['Date'])
df.dtypes
```

```
[18]: Date          datetime64[ns]
      Production    int64
      dtype: object
```

1.0.3 3. Set the Date column to be the new index

```
[ ]:
```

```
[19]: # DON'T WRITE HERE
df.set_index('Date',inplace=True)
df.head()
```

```
[19]:
```

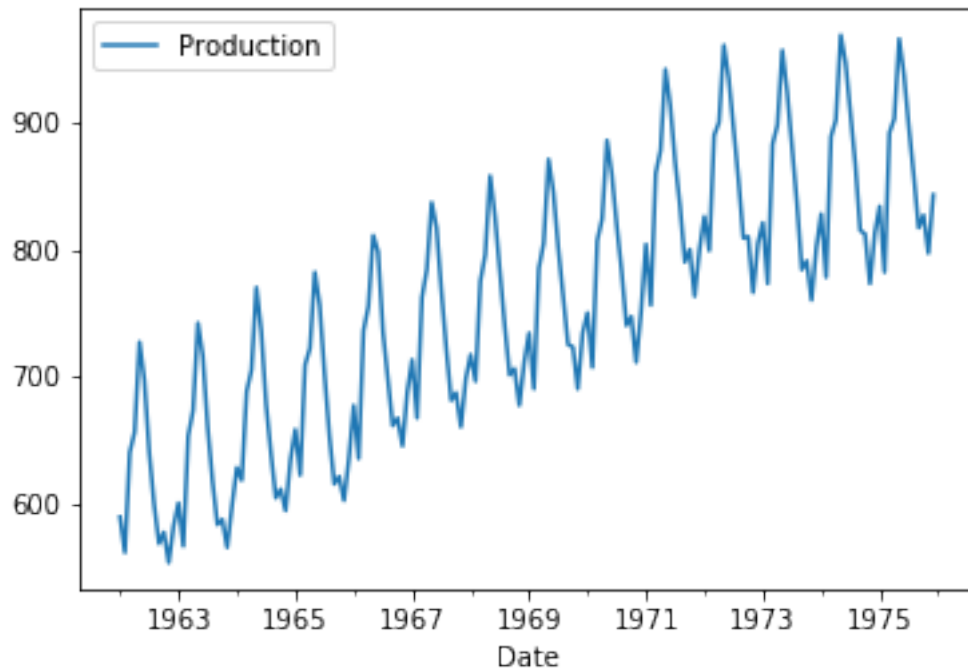
Date	Production
1962-01-01	589
1962-02-01	561
1962-03-01	640
1962-04-01	656
1962-05-01	727

1.0.4 4. Plot the DataFrame with a simple line plot. What do you notice about the plot?

```
[ ]:
```

```
[20]: # DON'T WRITE HERE
df.plot();

# THE PLOT SHOWS CONSISTENT SEASONALITY, AS WELL AS AN UPWARD TREND
```



1.0.5 5. Add a column called 'Month' that takes the month value from the index

HINT: You have to call `df.index` as `df['Date']` won't work.

BONUS: See if you can obtain the name of the month instead of a number!

```
[ ]:
```

```
[28]: # DON'T WRITE HERE
df['Month']=df.index.month
df.head()
```

```
[28]:
```

	Production	Month
Date		
1962-01-01	589	1
1962-02-01	561	2
1962-03-01	640	3
1962-04-01	656	4
1962-05-01	727	5

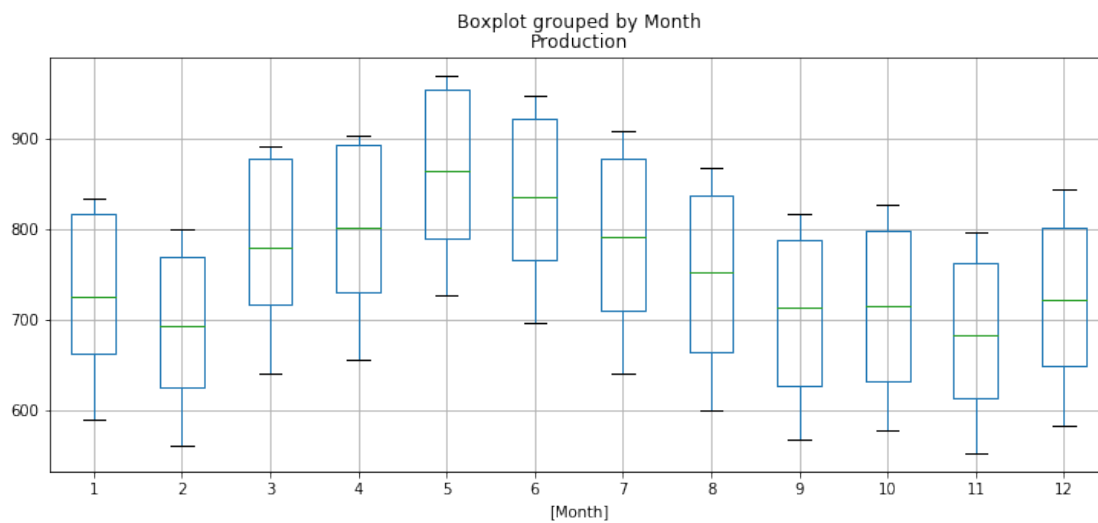
```
[22]: # BONUS SOLUTION:
df['Month']=df.index.strftime('%B')
df.head()
```

```
[22]:      Production  Month
Date
1962-01-01      589  January
1962-02-01      561  February
1962-03-01      640   March
1962-04-01      656   April
1962-05-01      727    May
```

1.0.6 6. Create a BoxPlot that groups by the Month field

```
[ ]:
```

```
[29]: # DON'T WRITE HERE
df.boxplot(by='Month',figsize=(12,5));
```



2 Great Job!