

detrend(1)

October 19, 2023

1 Detrending a TS

Detrend a time series using **differencing**

```
[26]: from pandas import read_csv
      from pandas import datetime
      from matplotlib import pyplot
      import numpy as np
      import os
      import pandas as pd
      from sklearn.linear_model import LinearRegression
```

```
C:\Users\Enric\AppData\Local\Temp\ipykernel_33316\3013253819.py:2:
FutureWarning: The pandas.datetime class is deprecated and will be removed from
pandas in a future version. Import from datetime module instead.
      from pandas import datetime
```

```
[27]: def parser(x):
      return datetime.strptime('190'+x, '%Y-%m')
```

```
[28]: #os.chdir('H:/Dropbox/Personale/PC/Università/Computer Science/DIdattica/
      ↪Cattolica/Materiale Didattico/Time Series Analysis and Forecasting/
      ↪Datasets-master')
      os.chdir('C:/Users/Enric/Dropbox/Personale/PC/Università/Computer Science/
      ↪DIdattica/Cattolica/Datasets-master')
      print("Local path is:",os.getcwd())
      series = read_csv('shampoo.csv', header=0, index_col=0, parse_dates=True).
      ↪squeeze()
      print(series.head(10))
```

```
Local path is: C:\Users\Enric\Dropbox\Personale\PC\Università\Computer
Science\DIdattica\Cattolica\Datasets-master
```

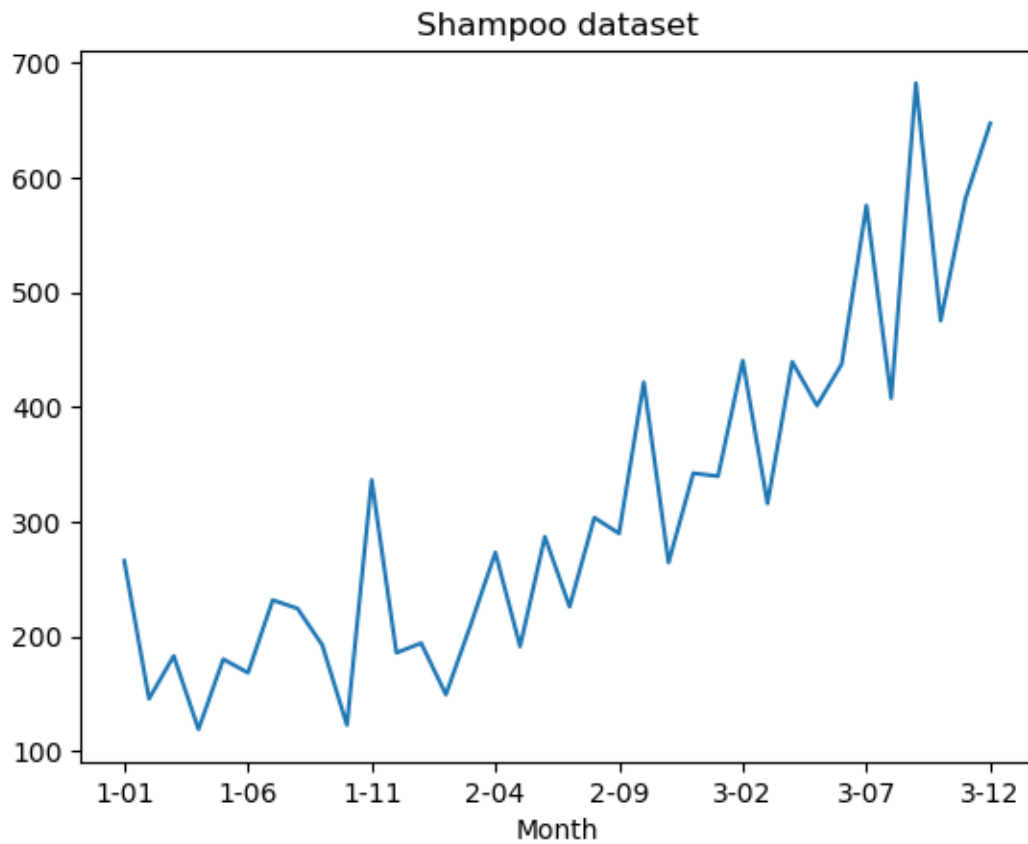
Month

1-01	266.0
1-02	145.9
1-03	183.1
1-04	119.3
1-05	180.3

```
1-06    168.5
1-07    231.8
1-08    224.5
1-09    192.8
1-10    122.9
Name: Sales, dtype: float64
```

```
[29]: series.plot(title = "Shampoo dataset")
```

```
[29]: <AxesSubplot:title={'center':'Shampoo dataset'}, xlabel='Month'>
```



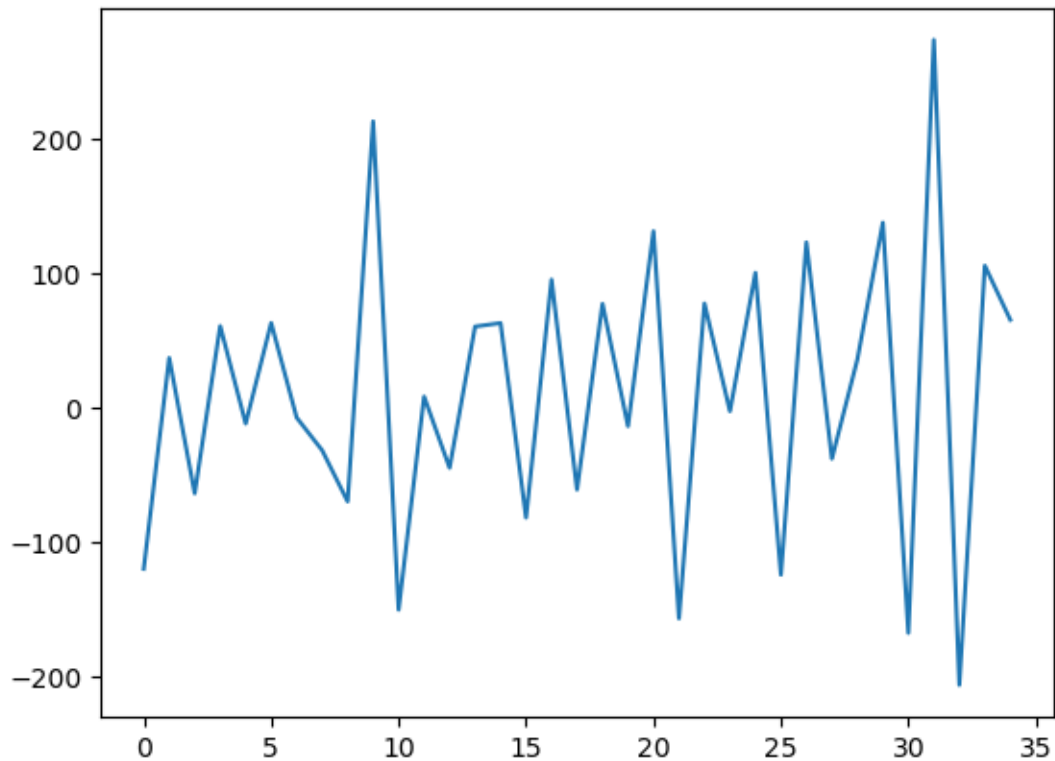
```
[30]: X = series.values

diff = list()

for i in range(1, len(X)):
    value = X[i] - X[i - 1]
    diff.append(value)

pyplot.plot(diff)
```

```
pyplot.show()
```



2 Detrend by Model Fitting

A **trend** is often easily visualized as a line through the observations.

Linear trends can be summarized by a linear model, and nonlinear trends may be best summarized using a polynomial or other curve-

fitting method.

Because of the subjective and domain-specific nature of identifying trends, this approach can help to identify whether a trend is present. Even

fitting a linear model to a trend that is clearly super-linear or exponential can be helpful.

```
[31]: # fit linear model
X = [i for i in range(0, len(series))]
```

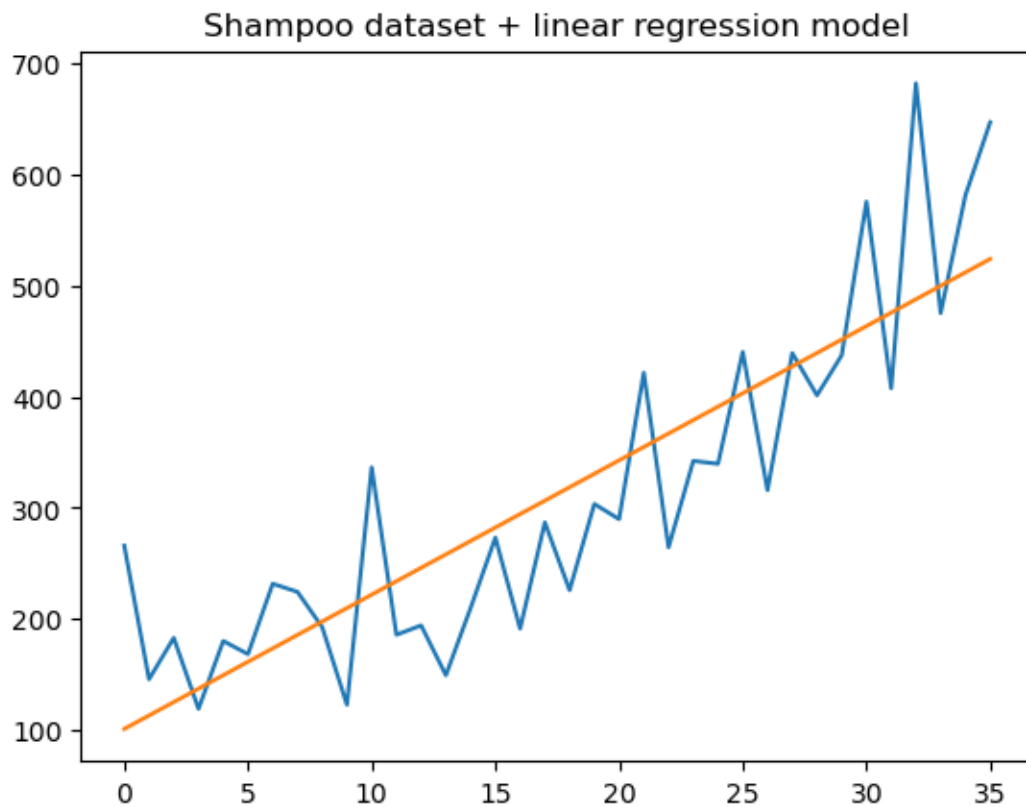
```
[32]: X = np.reshape(X, (len(X), 1))
y = series.values
```

```
[33]: model = LinearRegression()  
model.fit(X, y)
```

```
[33]: LinearRegression()
```

```
[34]: # calculate trend  
trend = model.predict(X)
```

```
[35]: # plot trend  
pyplot.plot(y)  
pyplot.plot(trend)  
pyplot.title("Shampoo dataset + linear regression model")  
pyplot.show()  
# detrend  
detrended = [y[i]-trend[i] for i in range(0, len(series))]
```



```
[ ]:
```