EXPLORING TIME SERIES MODELING

How To Analyse Multiple Time Series Variables

Time Series Modeling With Python Code

Jiahui Wang Apr 6 · 4 min read ★

Photo by tangi bertin on Unsplash

Welcome back! This is the 3rd post in the post series to explore analysing and modeling time series data with Python code. In the 1st post, we have discussed fundamental statistics: ***Time Series Modeling With Python Code: Fundamental Statistics***. The 2nd post has covered the analysis of a single time series variable: ***Time Series Modeling With Python Code: How To Analyse A Single Time Series Variable***.

In this post, we will continue to explore how to analyse multiple time series variables.

1. Rolling Covariance

Covariance is a measure of the joint linear variability of two random variables. Covariance itself is hard to interpret, as it depends on the variable magnitude. To normalize covariance, correlation coefficient is often used. Two commonly used correlation coefficient are: Pearson Correlation Coefficient and Spearman's Ranking Correlation Coefficient.

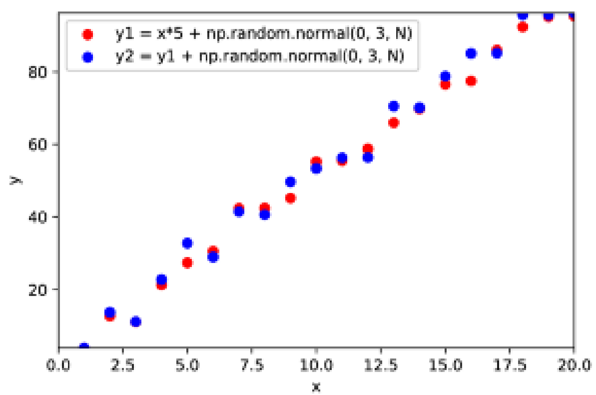
Pearson Correlation Coefficient value lies in between -1 to 1, with -1 implying a strong negative linear relationship, 0 implying no linear relationship, and 1 implying a strong positive linear relationship.

Spearman's Ranking Correlation Coefficient value also lies in between -1 and 1. However, Spearman's Ranking Correlation Coefficient measures the monotonicity between the two variables, with -1 implying a strong negative monotonic relationship, 0 implying no monotonic relationship, and 1 implying a strong positive monotonic relationship. To understand the monotonicity between the two variables, we can think of how (y_1^1-y_1^2) changes with (y_2^1-y_2^2): if they change in the same direction, then the two variables are positively monotonic; otherwise, they are negatively monotonic.

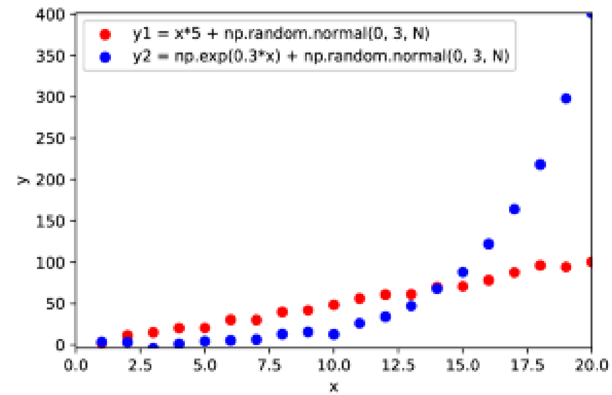
The following simulation shows the difference between Pearson Correlation Coefficient and Spearman's Ranking Correlation Coefficient. When y1 and y2 shows a linear relationship, Pearson Correlation Coefficient and Spearman's Ranking Correlation Coefficient are both close to 1. However, when y1 and y2 shows a monotonic relationship, Pearson Correlation Coefficient becomes smaller than Spearman's Ranking Correlation Coefficient.

```
import numpy as np
from scipy.stats import spearmanr
from scipy.stats import pearsonr
import matplotlib.pyplot as plt


N=20
x = np.linspace(1,N,num=N)
y1 = x*5 + np.random.normal(0, 3, N)
y2 = np.exp(0.3*x) + np.random.normal(0, 3, N)
#y2 = y1 + np.random.normal(0, 3, N)


plt.scatter(x,y1,color='r',label='y1 = x*5 + np.random.normal(0, 3,
N)')
plt.scatter(x,y2,color='b',label='y2 = y1 + np.random.normal(0, 3,
N)')
plt.legend(loc='upper left')
plt.xlabel('x')
plt.ylabel('y')
```

```
plt.xlim(0,N)
plt.ylim(min(min(y1),min(y2)),max(max(y1),max(y2)))
```



Pearson coefficient = 0.9959777037710135
Spearman coefficient = 0.993984962406015



Pearson coefficient = 0.8280842556818702
Spearman coefficient = 0.9789473684210527

Pearson coefficient measures the linearity of two variables, while Spearman coefficient measures the monotonicity of two variables

2. Spurious Relationship and Rolling Cointegration

To model time series data y using time series data x, we usually require both the x and y to be stationary I(0) process. If you are not familiar with stationarity test of a single time series variable, please refer to my previous post: ***Time Series Modeling With Python Code: How To Analyse A Single Time Series Variable***.

When both time series variables are non-stationary, they may show strong correlation even through the underlying generation processes have no casual relationships. This strong correlation may be purely caused by the fact that the two time series variables have non-constant mean. This phenomenon is called spurious relationship.

However, under a special circumstance, we can model time series data y using time series data x, when x and y are both I(1) process and cointegrated. Basically, cointegration means there is an amplifying effect in between x and y. At any time point, we can always multiply x with the same parameter to get a value which is close to y. In this way, the residual is I(0) process.

$$y_i = \alpha + \beta x_i + \mu_i$$
$$y_i \sim I(1)$$
$$x_i \sim I(1)$$
$$\mu_i \sim I(0)$$

Cointegration between x and y

Cointegration can be easily tested using statsmodels library.

```
import numpy as np
from statsmodels.tsa.stattools import coint

N=20
x = np.linspace(1,N,num=N)
y1 = x*5 + np.random.normal(0, 3, N)
y2 = np.exp(0.3*x) + np.random.normal(0, 3, N)
print(f'P value is {coint(y1,y2)[1]}')
```

Output:

P value is 0.9859002580259643

Since P value is larger than 0.05 significance level, we cannot reject the Null hypothesis that there is no cointegration.

Summary

In this post, we covered how to analyse covariance and cointegration of multiple time series variables. In the next post, we will move one step further, to explore how to use linear regression to model time series data. Please stay tuned!

Time Series Model Python Statistics Data Science Programming