Appendix A

June 20, 2018

1 Appendix A

This jupyter notebook provides an example of how two lists of values that have very low correlation scores (and have no reason to be correlated), can have seemingly high measures of correlation if there is an arbitary trend occuring over the list.

Any list of values, that are measured at discrete intervals in time can be reffered to as a timeseries. We will use the term timeseries in this appendix.

Import useful libraries

```
In [1]: import numpy as np
        import pandas as pd
        from numpy.random import random_sample

        import matplotlib.pyplot as plt
        import seaborn as sns

        %matplotlib inline
        sns.set()
```

2 Correlations between two lists of measurements

creating two lists of random values: a & b. a & b can be thought of as individual timeseries.

```
In [2]: i = 300 # number of points to create
        np.random.seed(7) # set seed for random generation: so produces the same random sample f
        rand = pd.DataFrame({'a': np.random.randn(i),'b':np.random.randn(i)})
```

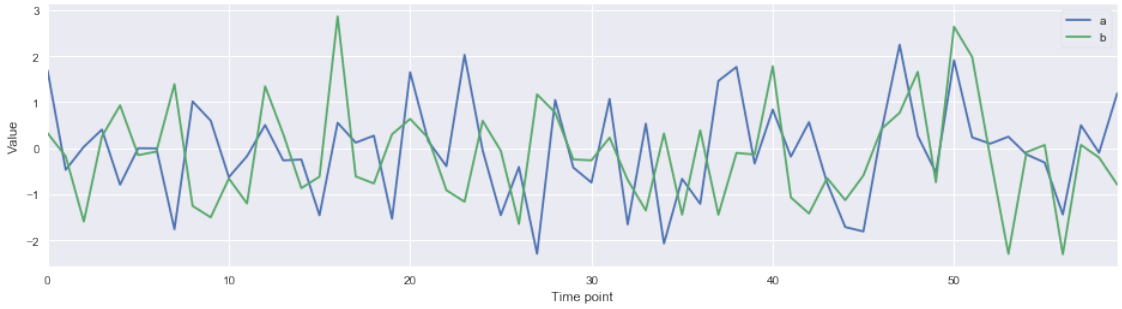
Showing the first 5 points in each timeseries:

```
In [3]: rand.head()
```

```
Out[3]:           a         b
        0  1.690526  0.329750
        1 -0.465937 -0.173824
        2  0.032820 -1.588248
        3  0.407516  0.257973
        4 -0.788923  0.932750
```

Plot first points in each timeseries

```
In [5]: fig, ax = plt.subplots();
        rand[0:60].plot(figsize=(16,4),ax=ax);
        ax.legend(frameon=True);
        ax.set_ylabel('Value');
        ax.set_xlabel('Time point');
```



Calculating the pearson correlation between the values in each timeseries: we find there is low correlation (0.08) between a & b.

```
In [9]: # calc correlation matrix
        rand.corr().round(2)
```

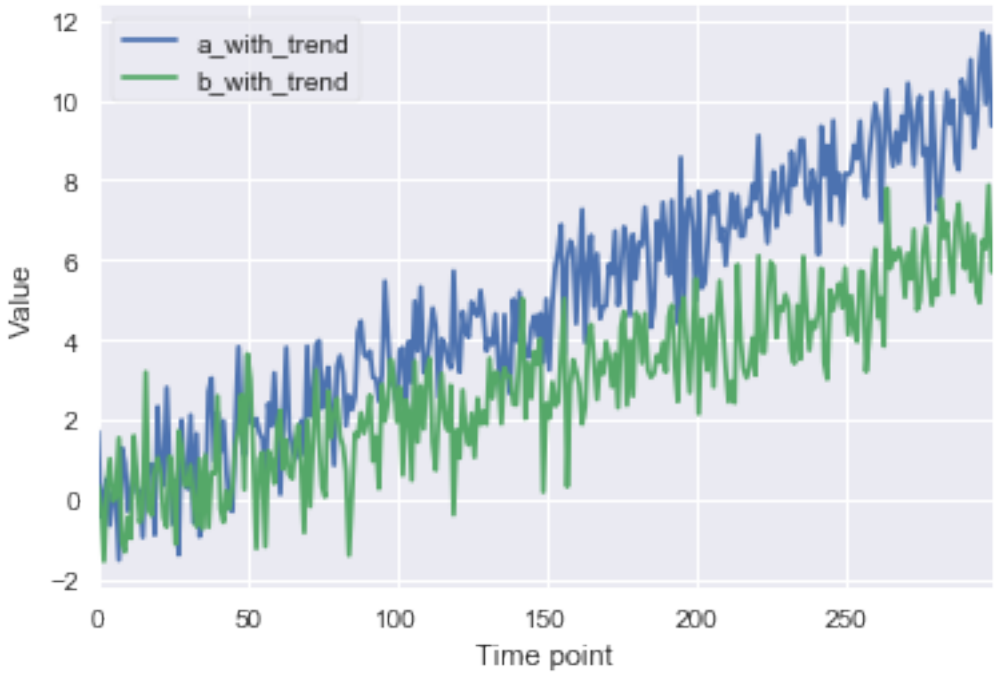
```
Out[9]:       a     b
        a  1.00  0.08
        b  0.08  1.00
```

3 Adding an arbitrary trend to each time series...

```
In [10]: rand['a_with_trend'] = rand.a + rand.index/30
```

```
In [11]: rand['b_with_trend'] = rand.b + rand.index/50
```

```
In [12]: fig, ax = plt.subplots();
         rand[['a_with_trend','b_with_trend']].plot(ax=ax);
         ax.legend(frameon=True);
         ax.set_ylabel('Value');
         ax.set_xlabel('Time point');
```




```
In [13]: # calc correlation matrix
         rand.corr().round(2)
```

Out[13]:

	a	b	a_with_trend	b_with_trend
a	1.00	0.08	0.33	0.05
b	0.08	1.00	0.11	0.56
a_with_trend	0.33	0.11	1.00	0.85
b_with_trend	0.05	0.56	0.85	1.00

We observe the pearson correlation coefficient between each list with added trend (a_with_trend & b_with_trend) is very high (0.85).

4 Removing the trend in the time series

Differencing each timeseries (subtracting each point from the previous one in the same timeseries) is one way of removing the trend (and hence dependence between points over time).

```
In [14]: rand['a_trend_diff'] = rand.a_with_trend.diff()
         rand['b_trend_diff'] = rand.b_with_trend.diff()
```

The pearson correlation scores between the differenced timeseries (a_trend_diff & b_trend_diff) are very small. THe value is closer to the original correlations between lists (a & b); 0.6 compared with 0.8 in original timeseries.

```
In [15]: rand.corr().round(2)
```

Out[15]:

	a	b	a_with_trend	b_with_trend	a_trend_diff	\
a	1.00	0.08	0.33	0.05	0.72	
b	0.08	1.00	0.11	0.56	0.03	
a_with_trend	0.33	0.11	1.00	0.85	0.23	
b_with_trend	0.05	0.56	0.85	1.00	0.02	
a_trend_diff	0.72	0.03	0.23	0.02	1.00	
b_trend_diff	0.06	0.70	0.02	0.34	0.06	

	b_trend_diff
a	0.06
b	0.70
a_with_trend	0.02
b_with_trend	0.34
a_trend_diff	0.06
b_trend_diff	1.00