INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Autocorrelation Function

Rob Reider
Adjunct Professor, NYU-Courant
Consultant, Quantopian

Autocorrelation Function

sample autocorrelation function (or ACF) shows not only the lag-one, but the entire autorrelation function for different lags.

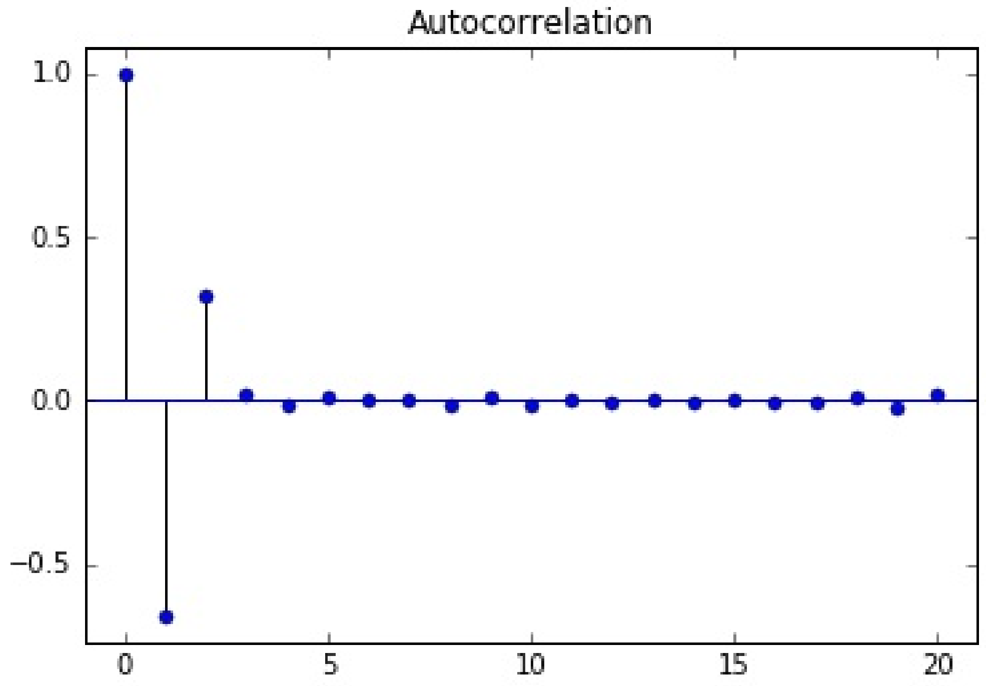
- Autocorrelation Function (ACF): The autocorrelation as a function of the lag

any significant non-zero autocorrelations implies that the series can be forecast from the past.

- Equals one at lag-zero
- Interesting information beyond lag-one

ACF Example 1: Simple Autocorrelation Function

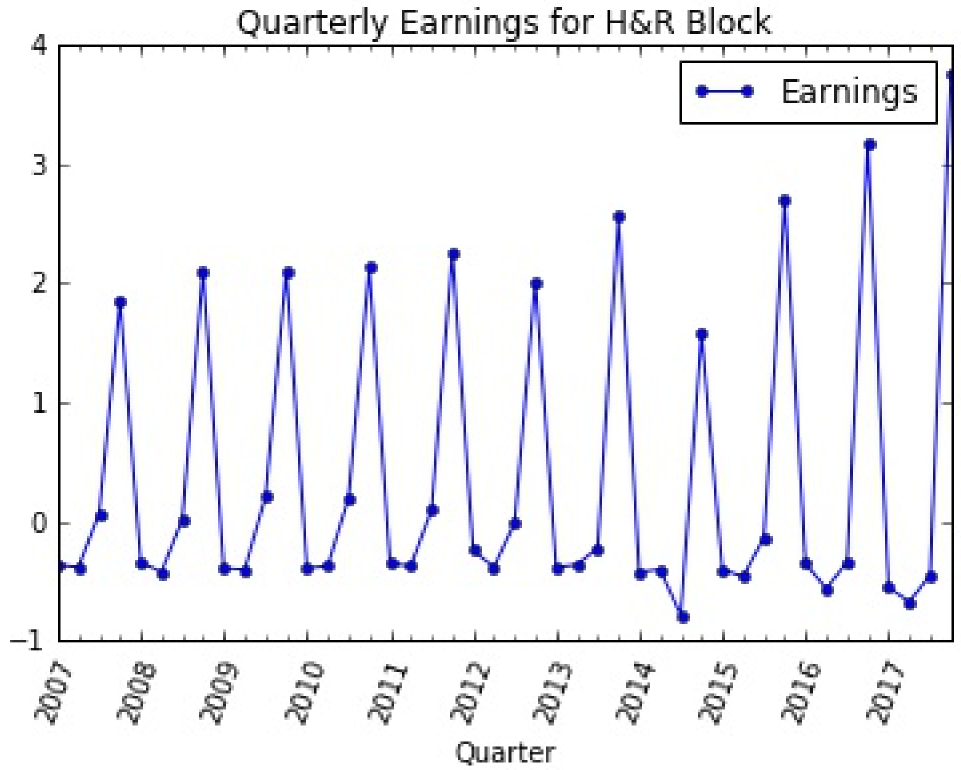
- Can use last two values in series for forecasting



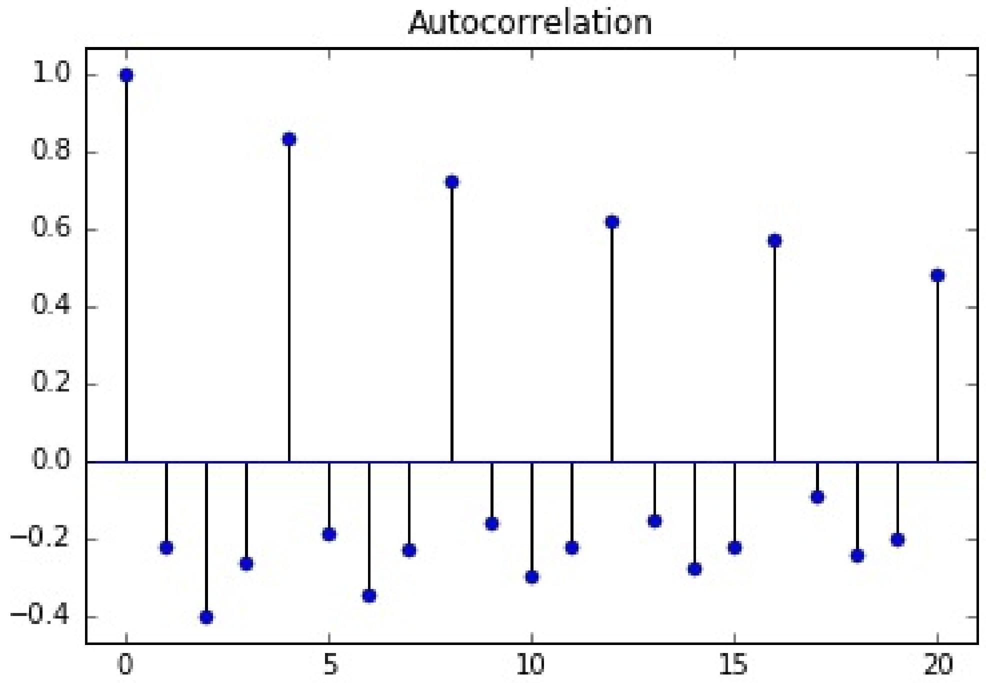
this autocorrelation function implies that u can forecast the next value of the series from the last 2 values, since lag-one and lag-two autocorrelations differ from zero.

ACF Example 2: Seasonal Earnings

- Earnings for H&R Block



- ACF for H&R Block

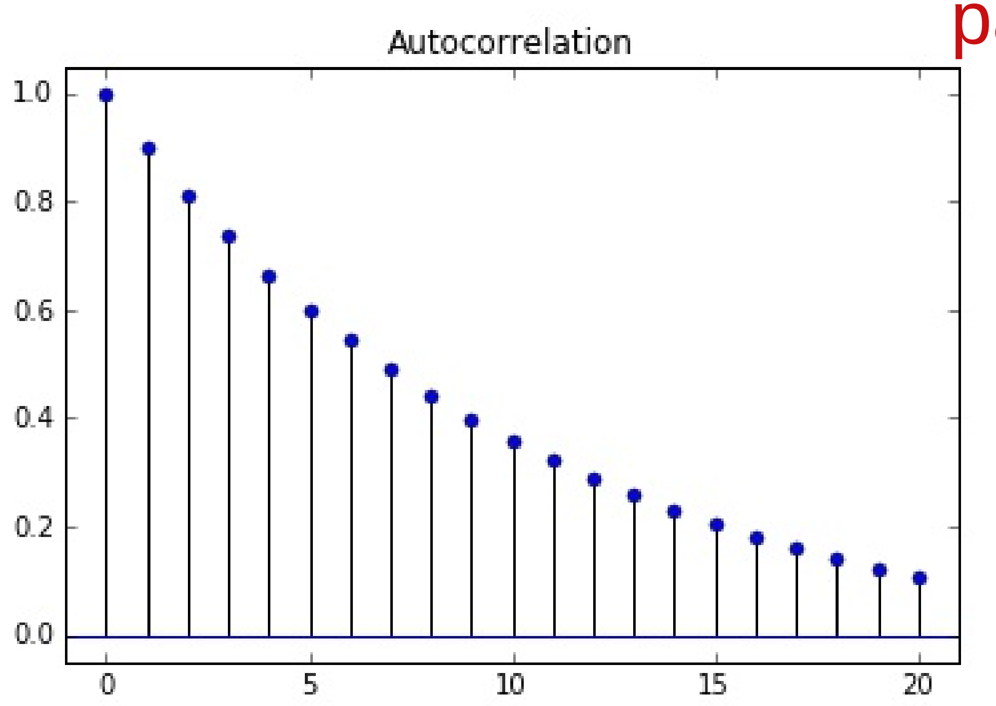


autocorrelation function on the right shows strong autocorrelation at lags 4, 8, 12, 16 and 20

ACF Example 3: Useful for Model Selection

- Model selection

ACF can also be useful for selecting a parsiminious model for fitting data.



Plot ACF in Python

- Import module:

```
from statsmodels.graphics.tsaplots import plot_acf
```

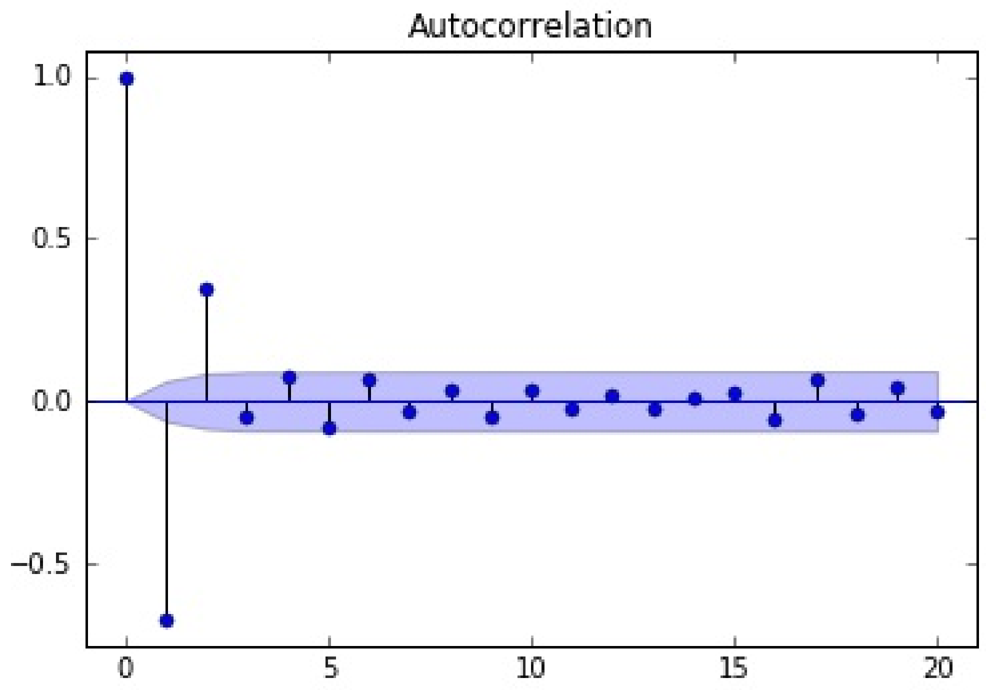
- Plot the ACF:

```
plot_acf(x, lags= 20, alpha=0.05)
```

how many lags of the acf will be plotted

alpha: set the width of confidence interval

Confidence Interval of ACF



ACF plot that contains confidence interval for each lag, which is the blue region

Confidence Interval of ACF

- Argument `alpha` sets the width of confidence interval
- Example: `alpha=0.05`
  - 5% chance that if true autocorrelation is zero, it will fall outside blue band
- Confidence bands are wider if:
  - Alpha lower
  - Fewer observations

approximation the width of 95% confidence interval

- Under some simplifying assumptions, 95% confidence bands are $\pm 2/\sqrt{N}$
- If you want no bands on plot, set `alpha=1`

no confidence interval

ACF Values Instead of Plot

extract ACF numerical values

```
from statsmodels.tsa.stattools import acf
print(acf(x))

[ 1.         -0.6765505   0.34989905 -0.01629415 -0.02507013  0.01930354
 -0.03186545  0.01399904 -0.03518128  0.02063168 -0.02620646 -0.00509828
...
  0.07191516 -0.12211912  0.14514481 -0.09644228  0.05215882]
```

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Let's practice!

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

White Noise

general definition, white noise is a series with mean=const, variance =const, zero autocorrelation at all lags.

Rob Reider
Adjunct Professor, NYU-Courant
Consultant, Quantopian

What is White Noise?

- White Noise is a series with:
  - Constant mean
  - Constant variance
  - Zero autocorrelations at all lags
- Special Case: if data has normal distribution, then *Gaussian White Noise*

Simulating White Noise

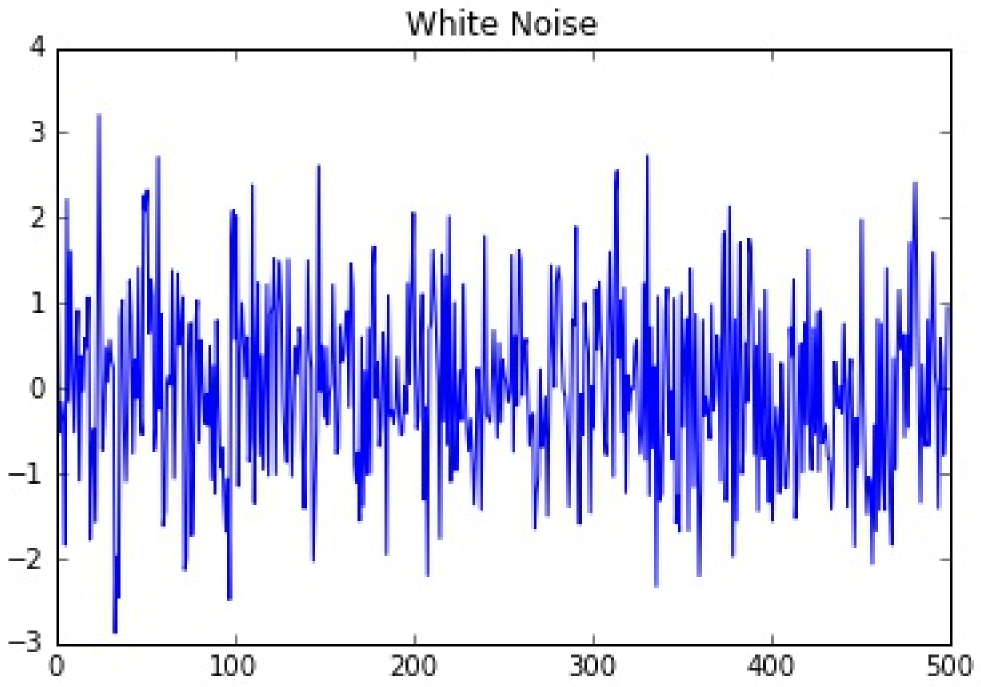
- It's very easy to generate white noise

```
import numpy as np
noise = np.random.normal(loc=0, scale=1, size=500)
```

loc is mean
scale is std

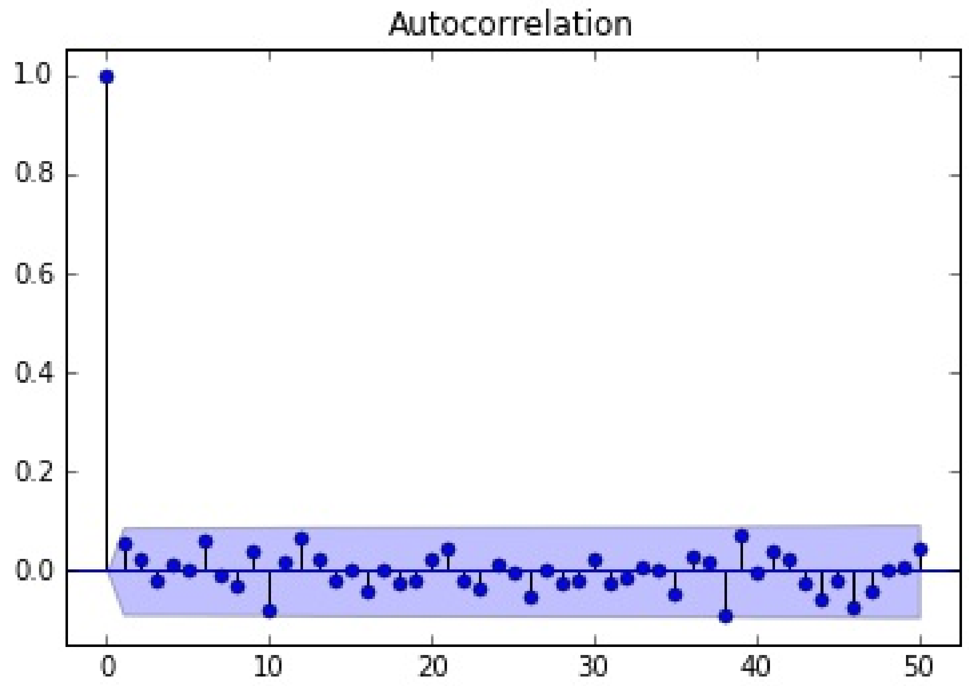
What Does White Noise Look Like?

```
plt.plot(noise)
```



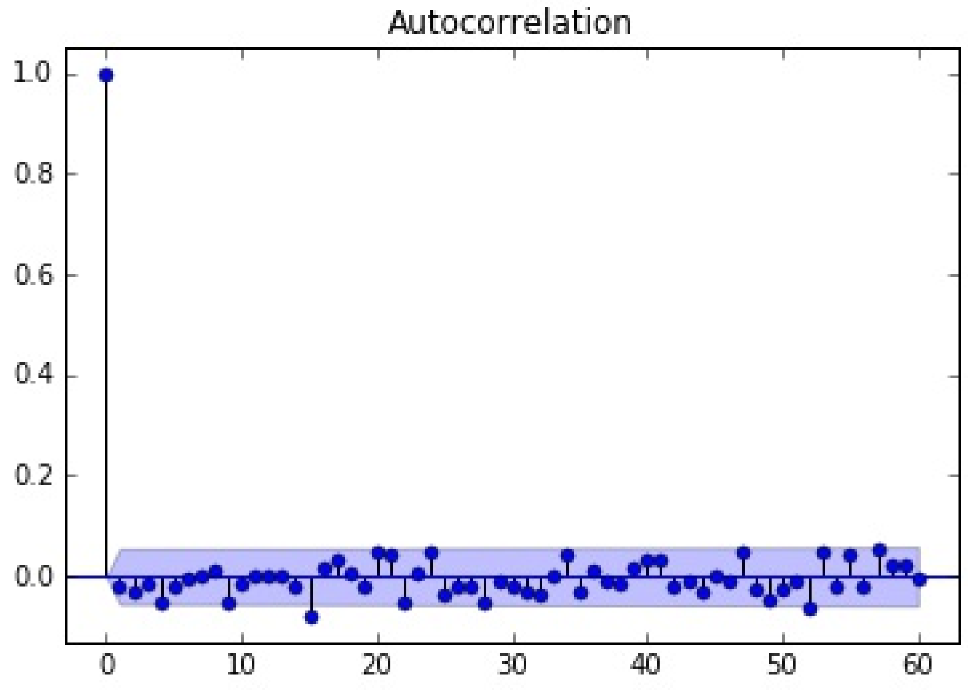
Autocorrelation of White Noise

```
plt_acf(noise, lags=50)
```



Stock Market Returns: Close to White Noise

- Autocorrelation Function for the S&P500



notice that there are pretty much no lags where autocorrelation is significantly different from zero.

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Let's practice!

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Random Walk

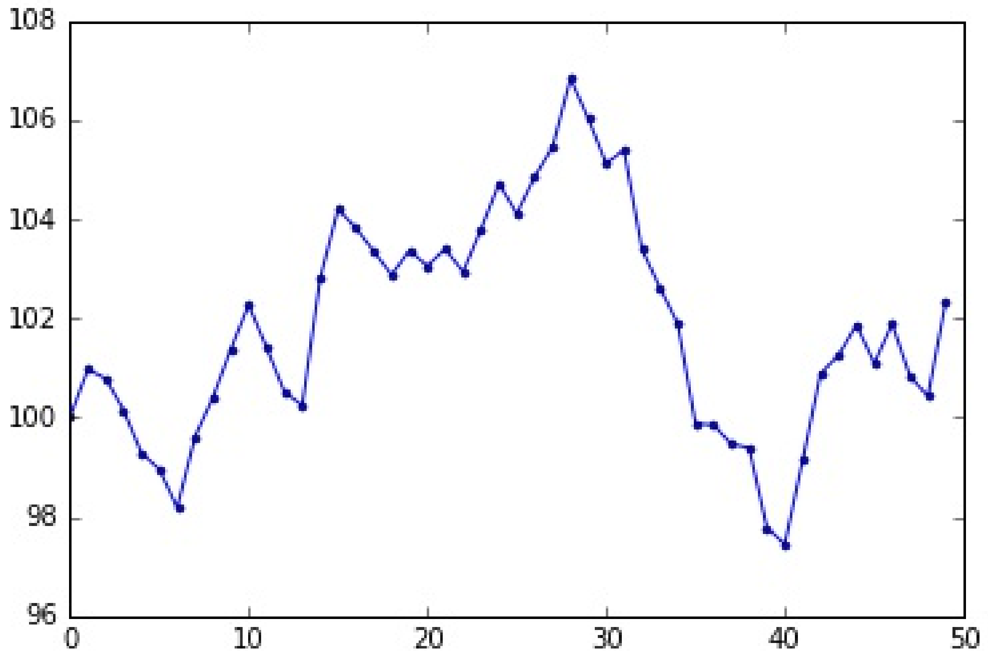
Rob Reider
Adjunct Professor, NYU-Courant
Consultant, Quantopian

What is a Random Walk?

- Today's Price = Yesterday's Price + Noise

$$P_t \quad = \quad P_{t-1} \quad + \quad \epsilon_t$$

- Plot of simulated data



What is a Random Walk?

- Today's Price = Yesterday's Price + Noise

$$P_t = P_{t-1} + \epsilon_t$$

- Change in price is white noise

$$P_t - P_{t-1} = \epsilon_t$$

- Can't forecast a random walk
- Best forecast for tomorrow's price is today's price

incidentally, if prices are in logs, then the difference in log prices is one way to measure returns.

bottom line is that if stock prices follow a random walk then stock returns are white noise.

What is a Random Walk?

- Today's Price = Yesterday's Price + Noise

$$P_t = P_{t-1} + \epsilon_t$$

- Random walk with drift: prices on average drift by mu every period

$$P_t = \mu + P_{t-1} + \epsilon_t$$

- Change in price is white noise with non-zero mean:

$$P_t - P_{t-1} = \mu + \epsilon_t$$

returns are still white noise, but with an average return of mu instead of zero

Statistical Test for Random Walk

to test whether a series follows a random walk, u can regress current prices on lagged prices.

- Random walk with drift

$$P_t = \mu + P_{t-1} + \epsilon_t$$

- Regression test for random walk

Ho: series is a random walk

$$P_t = \alpha + \beta\, P_{t-1} + \epsilon_t$$

if slope (beta) coefficient is not significant different from one. Can not reject Ho.

- Test:

$H_0 : \beta = 1$ (random walk)

$H_1 : \beta < 1$ (not random walk)

if slope coeff is significantly less than one, reject Ho

Statistical Test for Random Walk

identical way to do that test is to regress the difference in prices on the lagged price.

- Regression test for random walk

$$P_t = \alpha + \beta P_{t-1} + \epsilon_t$$

test slope coeff whether it is zero.

- Equivalent to

$$P_t - P_{t-1} = \alpha + \beta P_{t-1} + \epsilon_t$$

- Test:

$H_0 : \beta = 0$ (random walk)

$H_1 : \beta < 0$ (not random walk)

Statistical Test for Random Walk

- Regression test for random walk

$$P_t - P_{t-1} \quad = \alpha \; + \; \beta \, P_{t-1} \; + \; \epsilon_t$$

- Test:

  $H_0 : \beta = 0$ (random walk)

  $H_1 : \beta < 0$ (not random walk)

- This test is called the **Dickey-Fuller** test
- If you add more lagged changes on the right hand side, it's the **Augmented Dickey-Fuller** test

ADF Test in Python

- Import module from statsmodels

```
from statsmodels.tsa.stattools import adfuller
```

- Run Augmented Dickey-Test

```
adfuller(x)
```

Example: Is the S&P500 a Random Walk?

- Run Augmented Dickey-Fuller Test on SPX data

```
results = adfuller(df['SPX'])
```

- Print p-value

```
print(results[1])
0.782253808587
```

= p-value --> can not reject Ho

main output is p-value of the test.
p-value < 5% --> reject Ho (95% confidence)

- Print full results

```
print(results)
(-0.91720490331127869,
0.78225380858668414,
0,
1257,
{'1%': -3.4355629707955395,
'10%': -2.567995644141416,
'5%': -2.8638420633876671},
10161.888789598503)
```

test statistic

no of observations

critical values of test statistic

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Let's practice!

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Stationarity

in its strictest sense, it means that the joint distribution of the observations do not depend on time.

Rob Reider
Adjunct Professor, NYU-Courant
Consultant, Quantopian

What is Stationarity?

- **Strong stationarity**: entire distribution of data is time-invariant
- **Weak stationarity**: mean, variance and autocorrelation are time-invariant (i.e., for autocorrelation, corr$(X_t, X_{t-\tau})$ is only a function of τ)

less restrictive version of stationary (easier to test) is weak stationary.

corr(Xt, X(t-tau)) is only a function of lag tau, not a function of time.

Why Do We Care?

a process is not stationary, it becomes difficult to model.

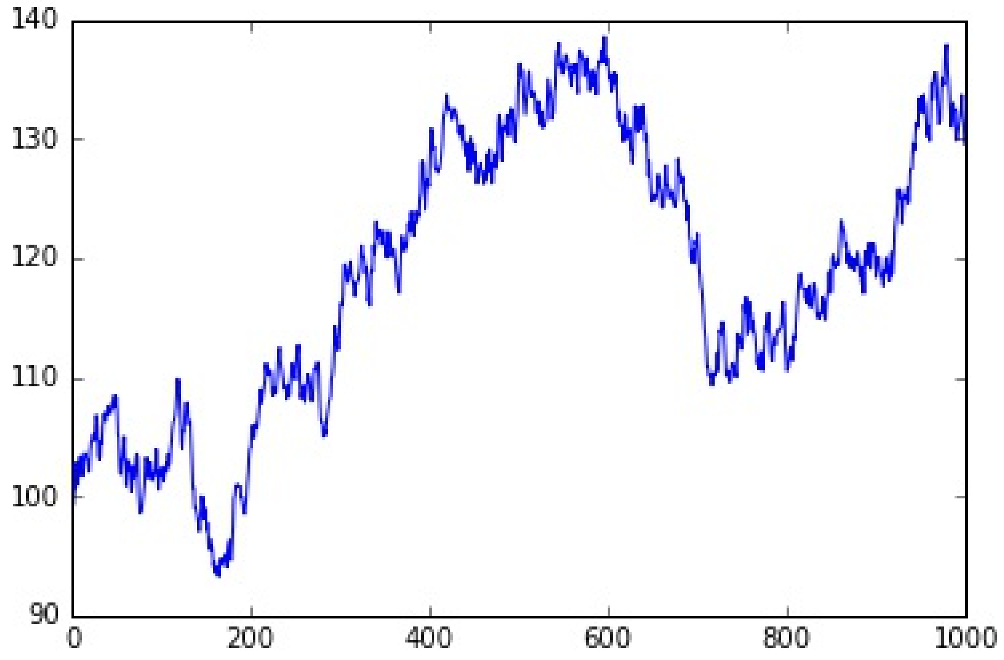
- If parameters vary with time, too many parameters to estimate
- Can only estimate a parsimonious model with a few parameters

modeling involves estimating a set of parameters, if a process is not stationary, the parameters are different at each point in time, there are too many parameters to estimate.
End up having more parameters than actual data

stationary is neccesary for a parsimonious model, one with a smaller set of parameter to estimate

Examples of Nonstationary Series

- Random Walk

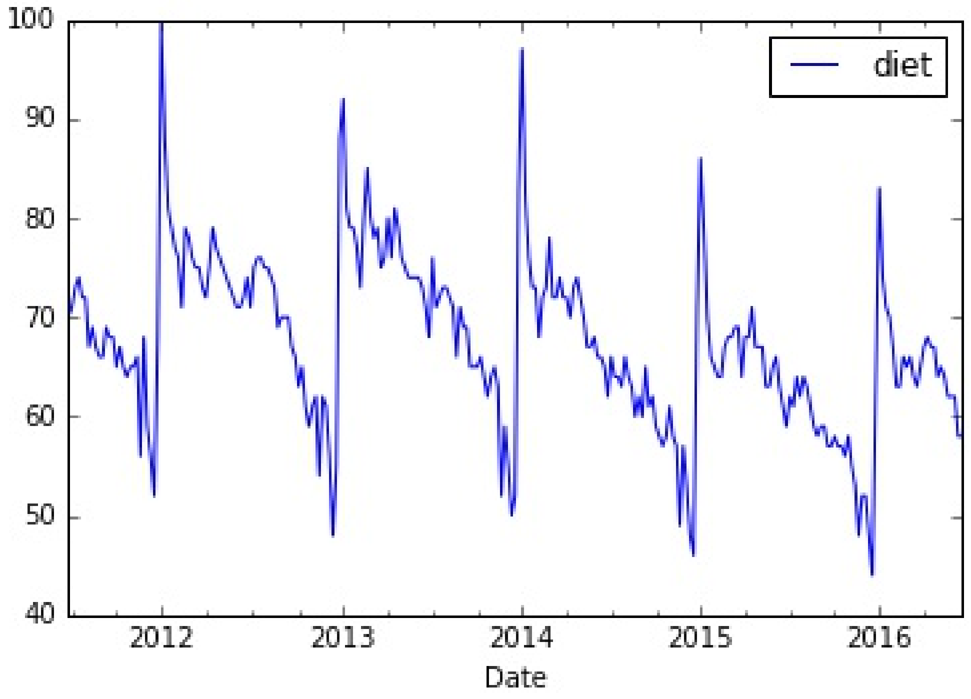


variance grows with time.
EX: stock prices are a random walk, uncertainty about prices tomorrow is much less than the uncertainty 10 years from now.

Examples of Nonstationary Series

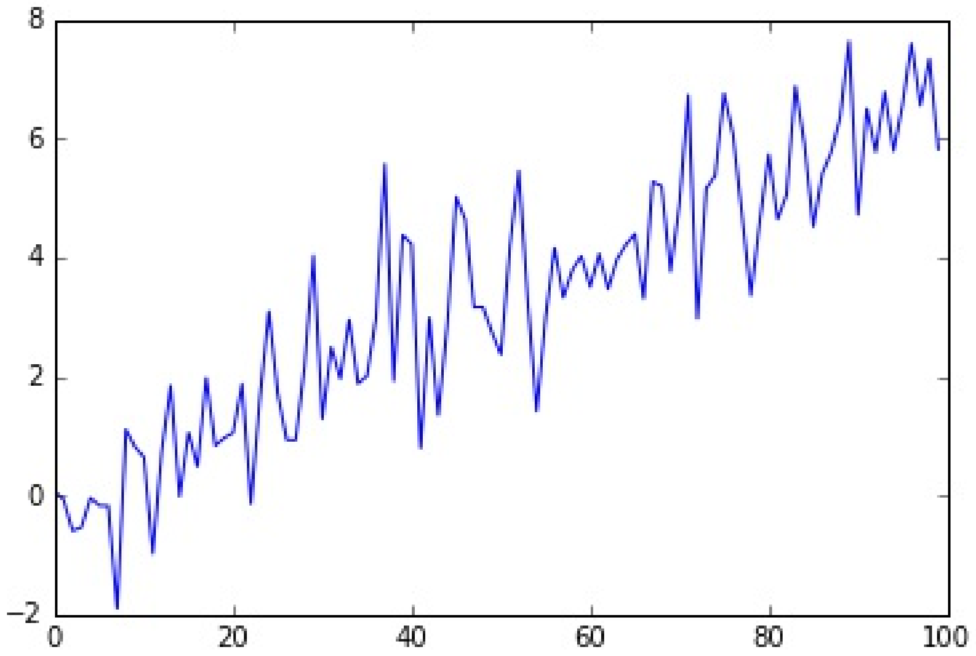
- Seasonality in series

mean varies with time of the year.



Examples of Nonstationary Series

- Change in Mean or Standard Deviation over time



here is white noise, which would ordinarily be a stationary process,

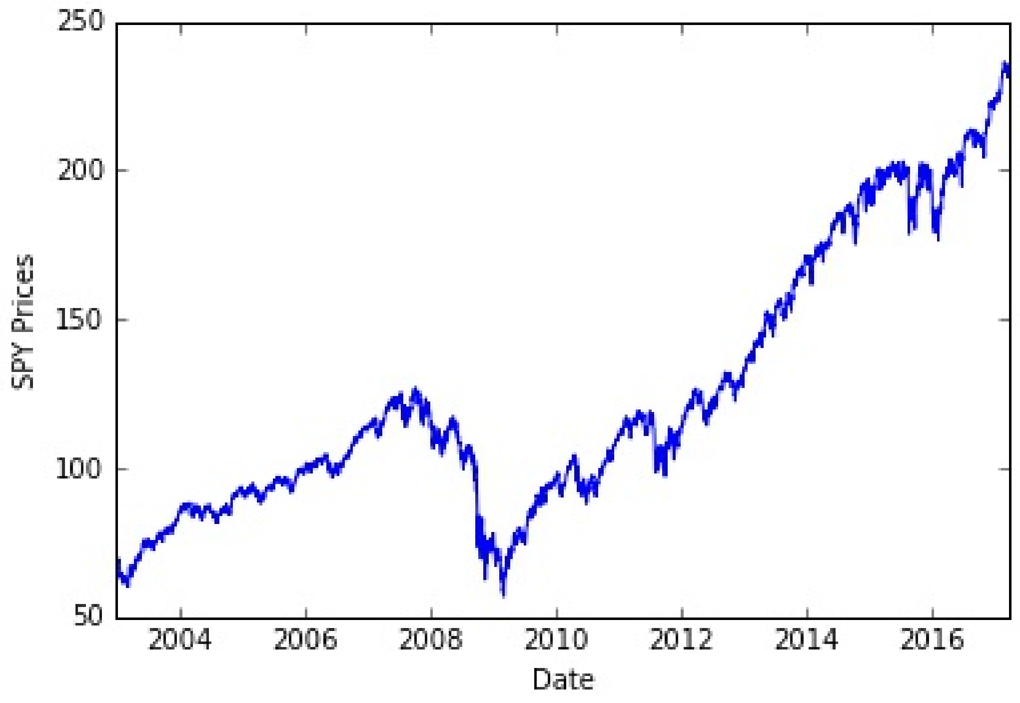
but mean increases over time, make it non-stationary.

Transforming Nonstationary Series Into Stationary Series

non-stationary

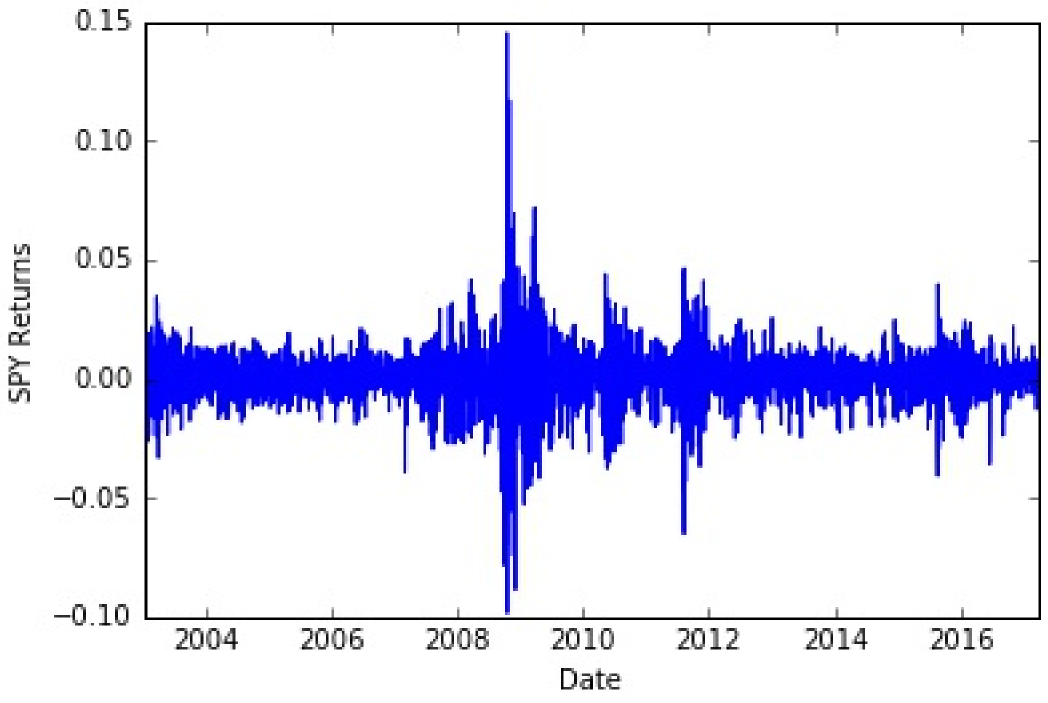
- Random Walk

```
plot.plot(SPY)
```



- First difference --> new series is white noise which is stationary

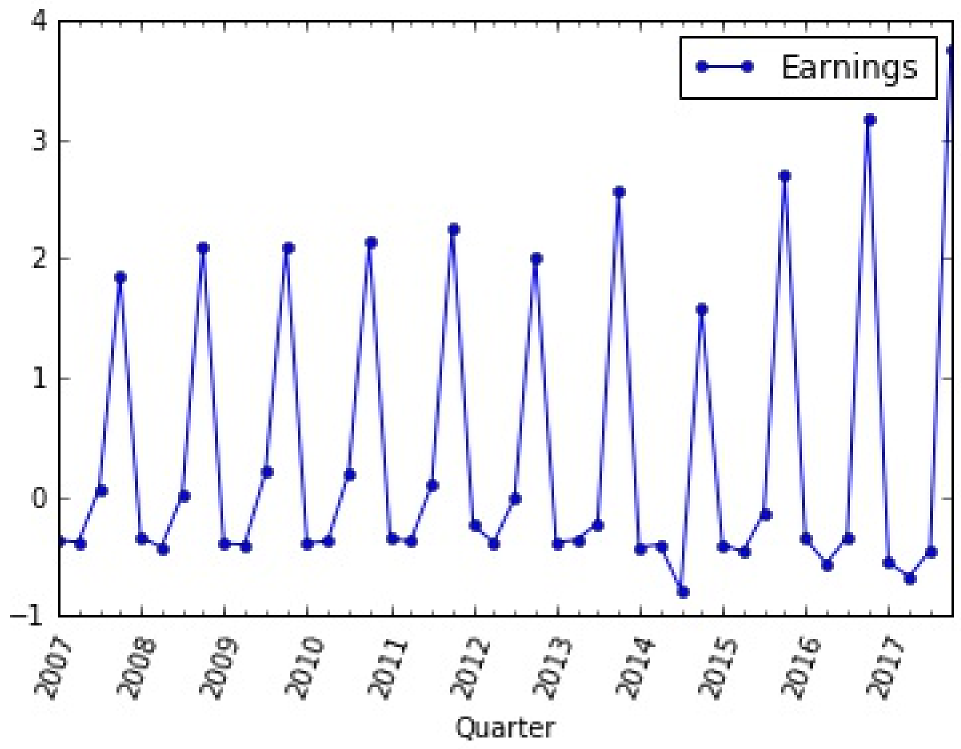
```
plot.plot(SPY.diff())
```



Transforming Nonstationary Series Into Stationary Series

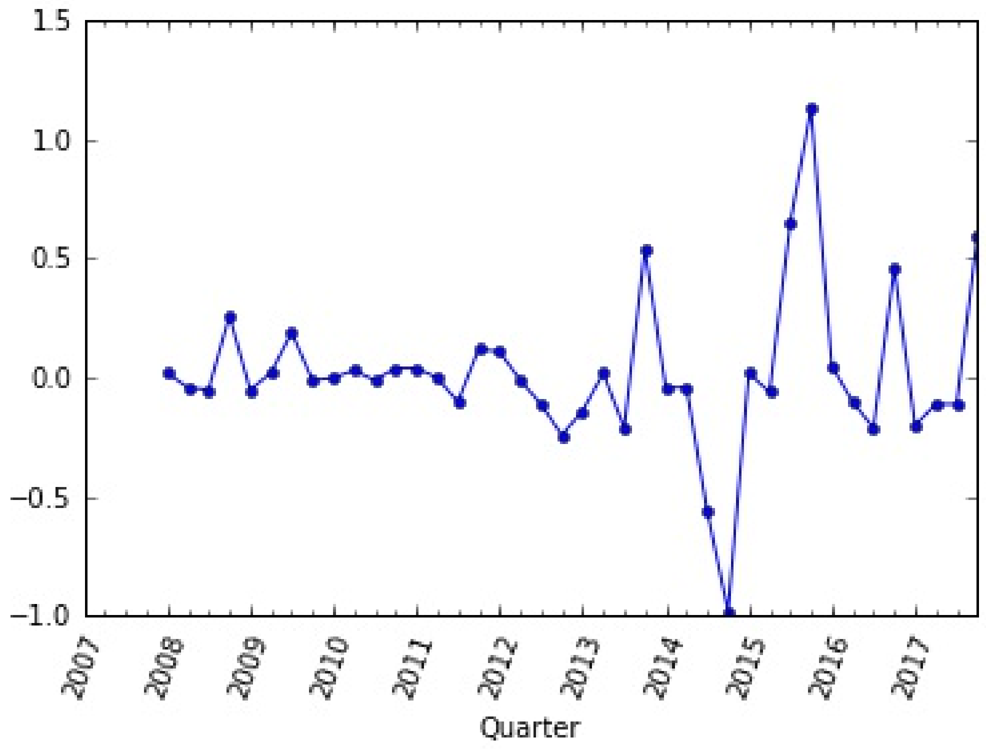
- Seasonality

```
plot.plot(HRB)
```



- Seasonal difference (take the difference with lag of 4)

```
plot.plot(HRB.diff(4))
```



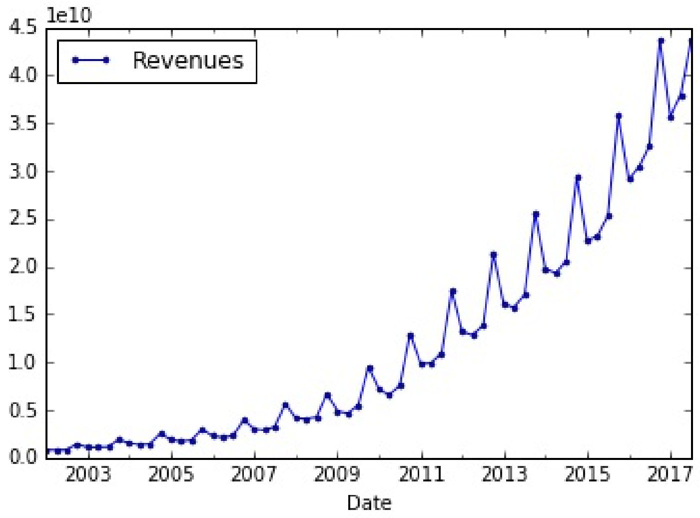
series looks stationary

Transforming Nonstationary Series Into Stationary Series

need to make two transformations.

- AMZN Quarterly Revenues

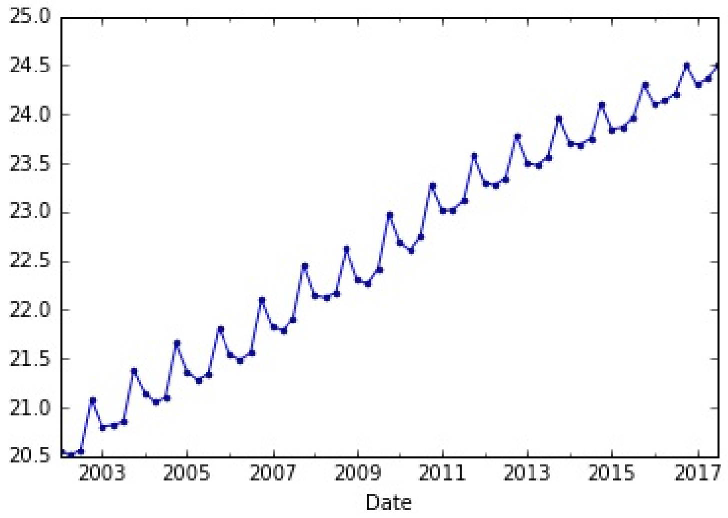
```
plt.plot(AMZN)
```



grow exponentially and strong seasonal pattern

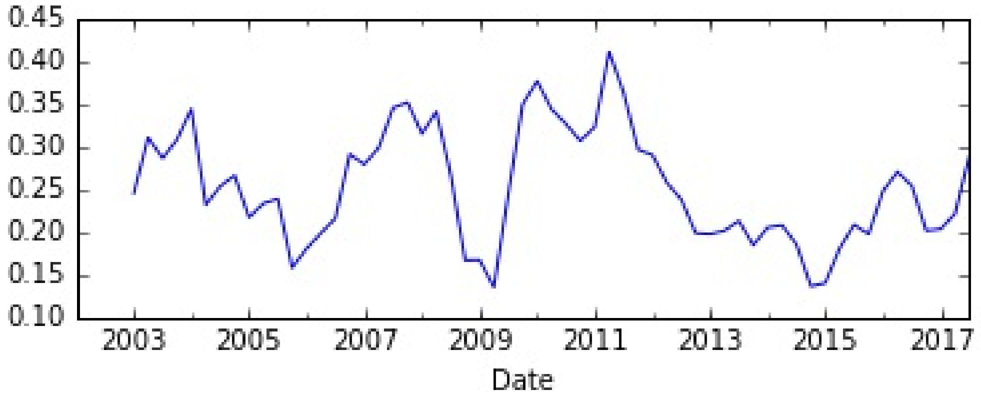
- Log of AMZN Revenues (eliminate the exponential growth)

```
plt.plot(np.log(AMZN))
```



- Log, then seasonal difference

```
plt.plot(np.log(AMZN).diff(4))
```



looks stationary

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Let's practice!