시계열 데이터 분석

with Pandas

이홍주 (Software Engineer)
lee.hongjoo@yandex.com

시계열 데이터 분석 with Pandas

- **Time Series with Pandas**
- 시계열 데이터 특성
- ETS 모델
- SMA, WMA, SES 모델
- ARIMA 모델

Time Series with Pandas

- 시계열 데이터는 시간 데이터를 인덱스로 하는 연속된 데이터 입니다.
- 시간 데이터를 Python 의 DateTime 타입으로 바꾸어 Pandas 가 제공하는 시계열 데이터를 처리하는 아래 기능들을 학습합니다.
  - DateTime 인덱스
  - Time Resampling
  - Time Shifts
  - Rolling and Expanding

Time Series with Pandas

- DateTime Index
  - time 또는 date 정보는 별개의 칼럼이기보다 인덱스인 경우가 많습니다.
  - Pandas 에 내장된 기능들로 DateTime 인덱스를 생성하고 활용하는 방법을 다룹니다.

Time Series with Pandas

- Time Resampling
  - 시계열 데이터의 인덱스는 시(hours), 분(minutes) 등 작은 단위 DateTime 인덱스로 이뤄진 경우도 많습니다.
  - 더 넓은 주기로 데이터를 집계(aggregate) 해야 하는 경우 Time Resampling 이 필요합니다.
  - groupby 를 사용함으로써 Time Resampling 을 수행할 수는 있지만, 비지니스 도메인에서 분기나 회계년도를 편리하게 처리할 방법은 못됩니다.
  - Pandas 는 이런 경우에 활용할 수 있는 frequency sampling 도구를 지원합니다.

Time Series with Pandas

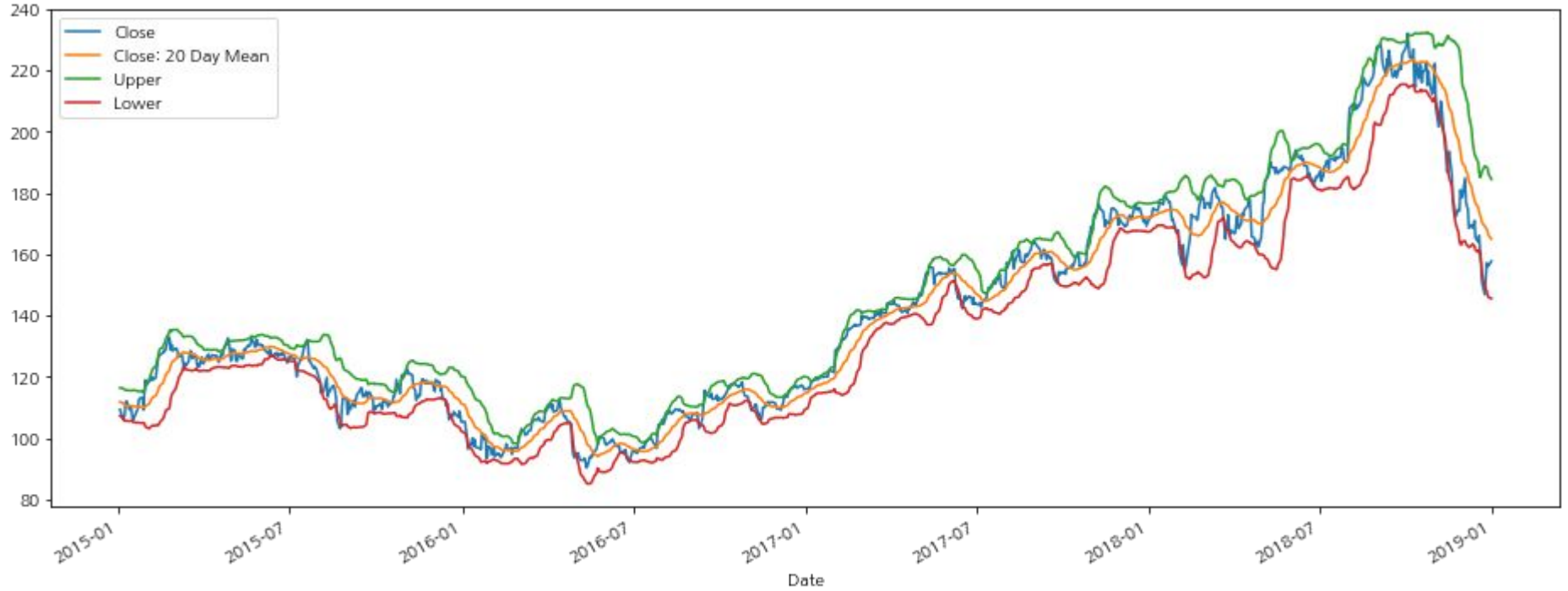
- Time Shifting
  - 시계열 분석 알고리즘을 사용하기 위해 데이터를 임의 시간만큼 앞 또는 뒤로 이동시켜야 할 때가 있습니다.
  - Pandas 는 이런 경우에도 매우 쉬운 방법을 제공합니다.

Time Series with Pandas

- Rolling and Expanding
  - 매일 수집한 데이터들에는 노이즈가 포함되기도 합니다.
  - 이럴 경우 데이터의 일반적인 트렌드를 구하기 위해 rolling mean (또는 moving average) 을 사용하기도 합니다.
  - Pandas 에 내장된 rolling 함수를 이용하면 주어진 시한 내 평균 (rolling mean) 등을 구할 수 있습니다.
  - 임의 시간 간격의 window 를 만들고 그 안에서 mean 같은 통계적 aggregation 을 실행하면 됩니다.

Time Series with Pandas

- 볼린저 밴드 (Bollinger Band)



시계열 데이터 분석

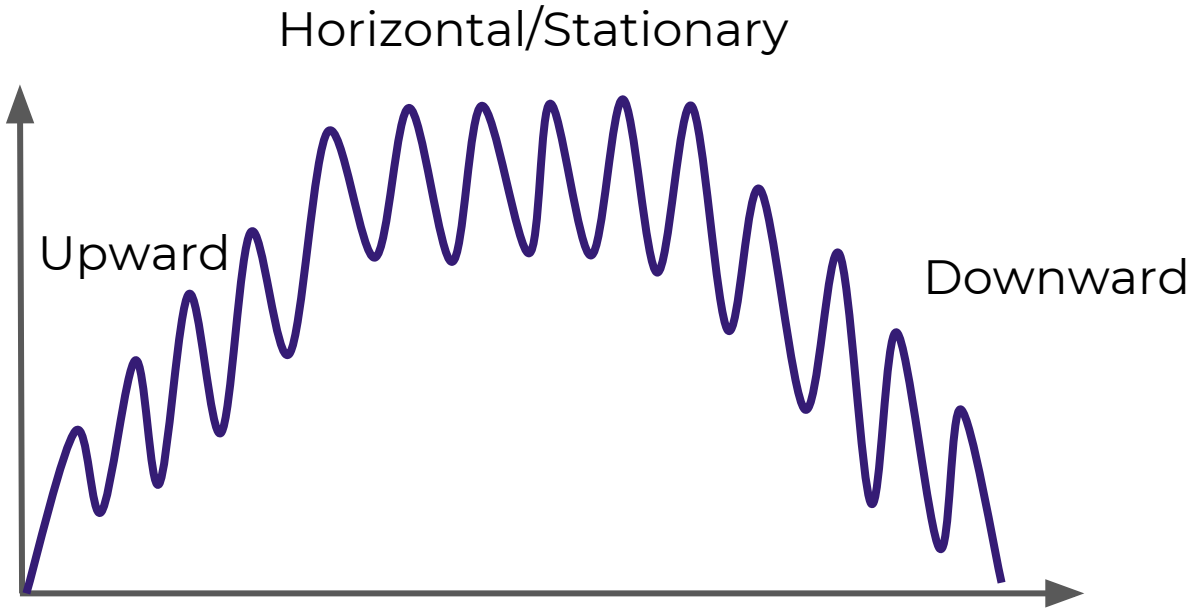
- Time Series with Pandas
- 시계열 데이터 특성
- ETS 모델
- SMA, WMA, SES 모델
- ARIMA 모델

시계열 데이터 특성

- 시계열 데이터는 몇가지 속성들을 가지고 있는데 그림을 통해 그것들을 알아보고 중요 용어를 정리합니다.
  - Trends
  - Seasonality
  - Cyclical

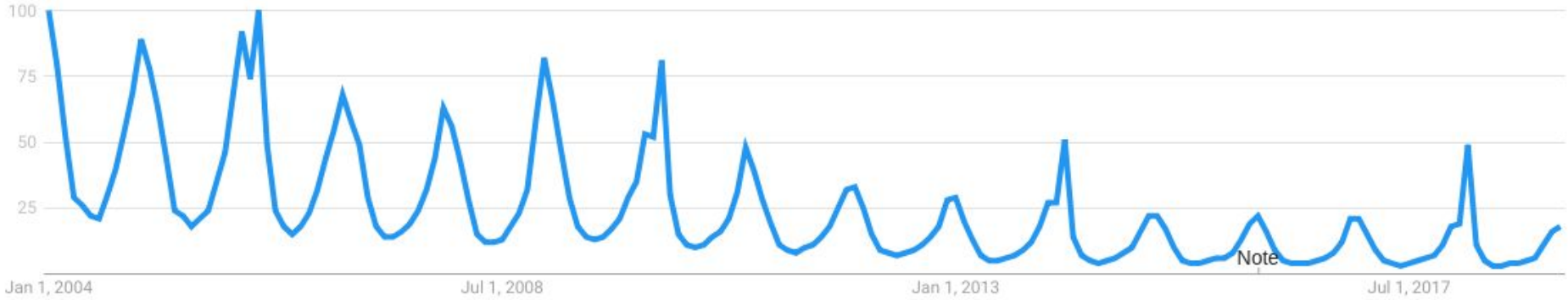
시계열 데이터 특성

- Trends



시계열 데이터 특성

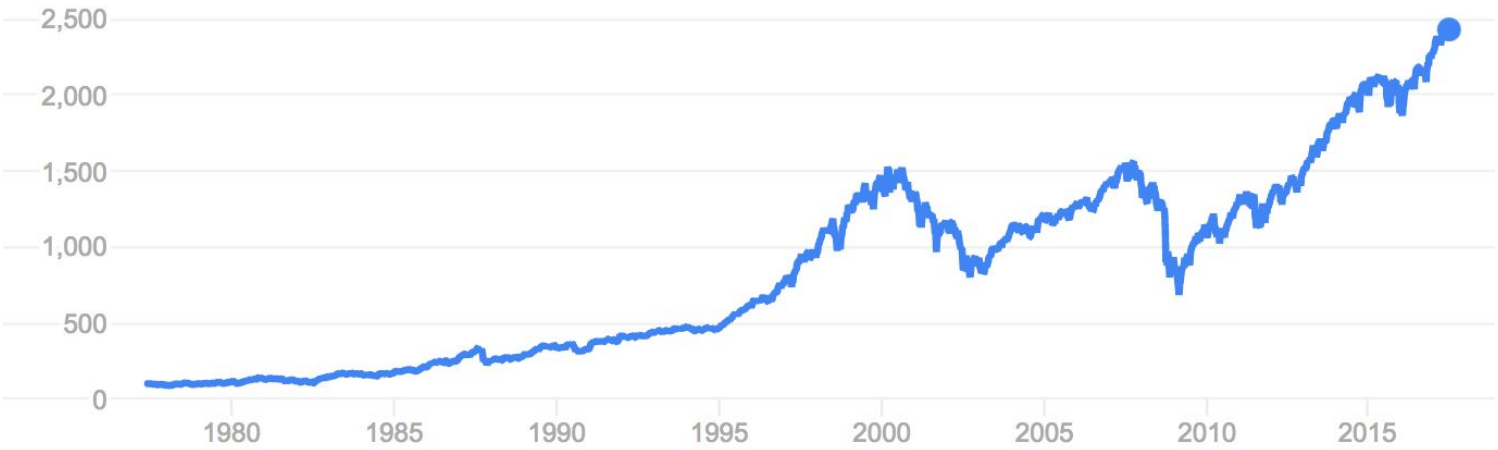
- Seasonality - 반복되는 트렌드



Google Trends : “Snowboarding”

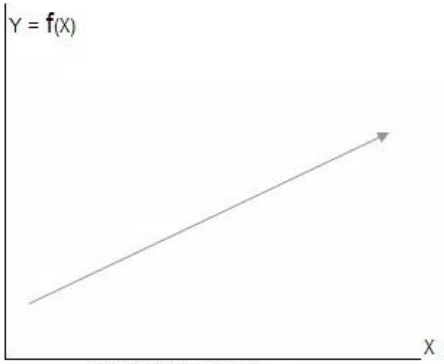
시계열 데이터 특성

- Cyclical - 일정하지 않은 기간의 트렌드

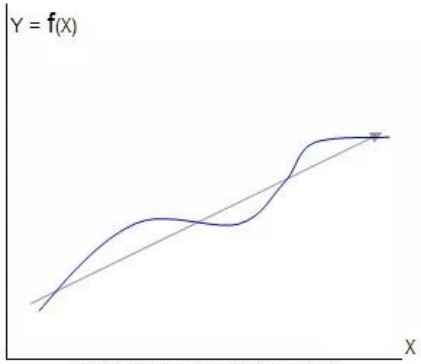


S&P500

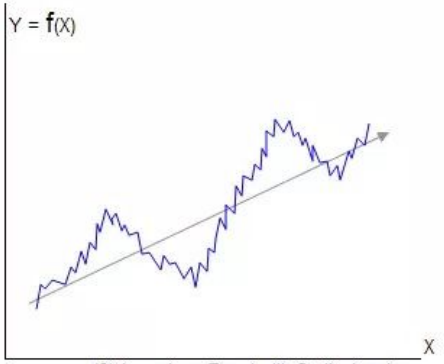
시계열 데이터 특성



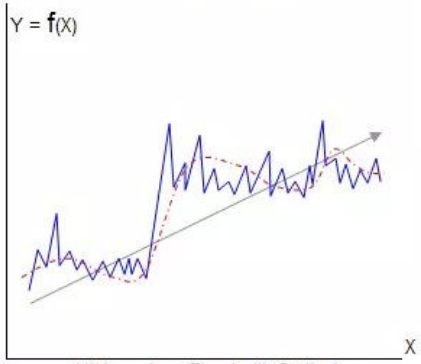
(1) Long-term Trend



(2) Long-term Trend with Cyclical Variations/Movements

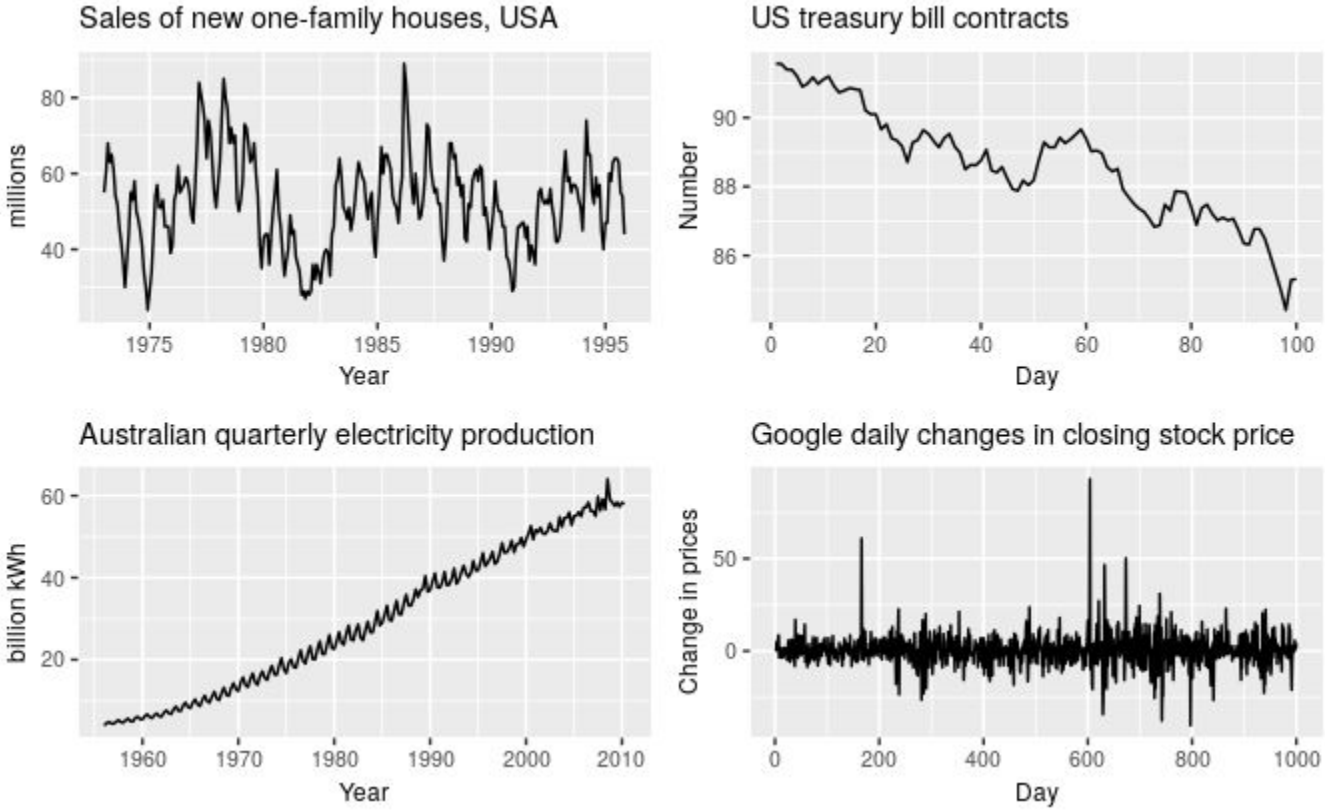


(3) Long-term Trend with Cyclical and Seasonal Variations/Movements



(4) Long-term Trend with Cyclical, Seasonal, and Random Variations/Movements

시계열 데이터 특성


Sales of new one-family houses, USA
millions
80
60
40
1975
1980
1985
1990
1995
Year
US treasury bill contracts
Number
90
88
86
0
20
40
60
80
100
Day
Australian quarterly electricity production
billion kWh
60
40
20
1960
1970
1980
1990
2000
2010
Year
Google daily changes in closing stock price
Change in prices
50
0
0
200
400
600
800
1000
Day

시계열 데이터 분석 with Pandas

- Time Series with Pandas
- 시계열 데이터 특성
- **ETS 모델**
- SMA, WMA, SES 모델
- ARIMA 모델

ETS 모델

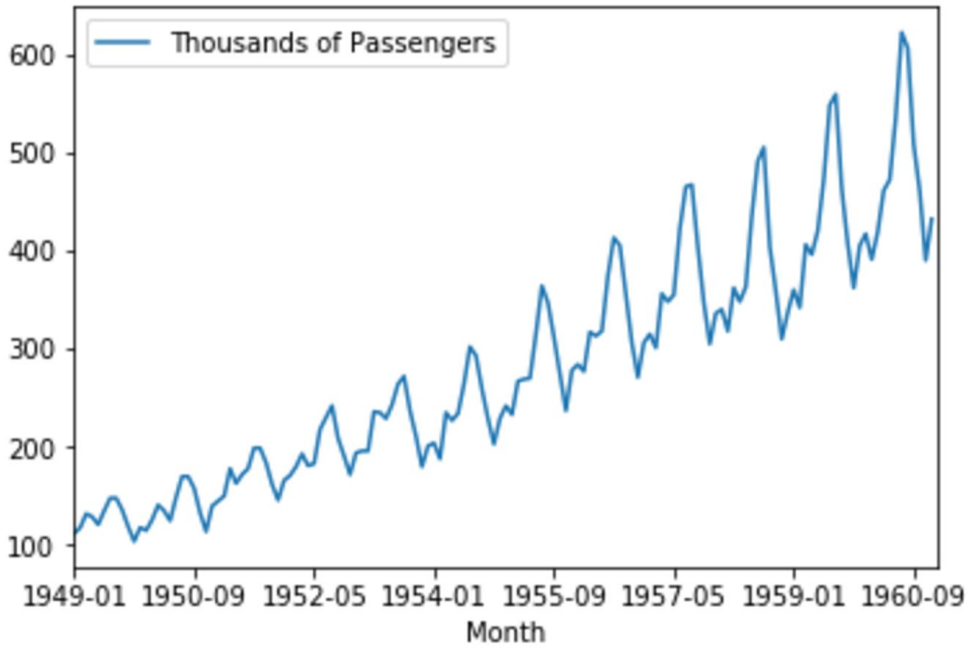
- 데이터의 패턴을 더 잘 파악하기 위해서 또는 예측을 수행하기 위해 Smoothing 을 합니다.
- Smoothing 위해서 Error , Trend , Seasonality 요소들을 활용하는데, 각각을 더하거나 곱하여 Smoothing 을 합니다.
- 또한 이것들을 가지고 시계열 데이터를 모델링 할 수 있습니다.

ETS 모델

- ETS Decomposition
  - ETS 컴포넌트들을 시각화 하는 것은 데이터의 흐름을 이해하는 데 큰 도움이 됩니다.

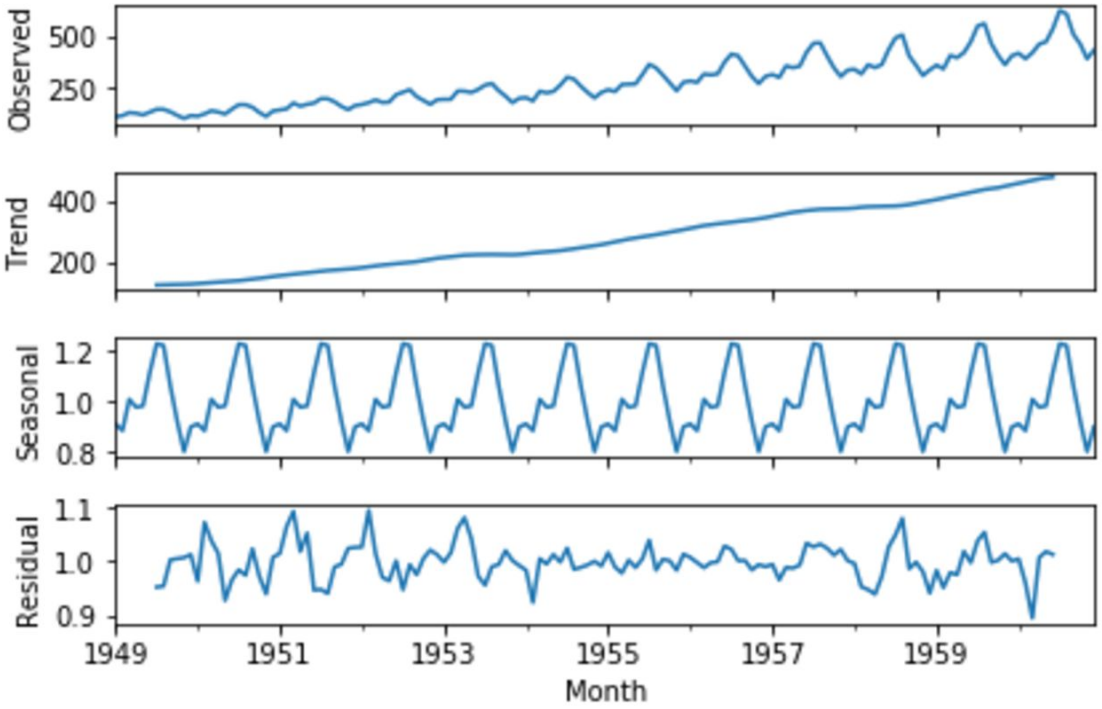
ETS 모델

- ETS Decomposition - Airline Passengers



ETS 모델

- ETS Decomposition - Airline Passengers



시계열 데이터 분석 with Pandas

- Time Series with Pandas
- 시계열 데이터 특성
- ETS 모델
- **SMA, WMA, SES 모델**
- ARIMA 모델

Simple Moving Average

week	sales	3MA
1	39	
2	44	
3	40	
4	45	
5	38	
6	43	
7	39	
8		

Simple Moving Average

week	sales	3MA
1	39	
2	44	
3	40	
4	45	41
5	38	
6	43	
7	39	
8		

$F_4 = (40 + 44 + 39) / 3$

Simple Moving Average

week	sales	3MA
1	39	
2	44	
3	40	
4	45	41
5	38	43
6	43	41
7	39	42
8		40

$F_4 = (40 + 44 + 39) / 3$

$F_5 = (45 + 40 + 34) / 3$

$F_6 = (38 + 45 + 40) / 3$

$F_7 = (43 + 38 + 45) / 3$

$F_8 = (38 + 45 + 40) / 3$

Weighted Moving Average

week	sales	4WMA
1	39	
2	44	
3	40	
4	45	
5	38	
6	43	
7	39	
8		

Weights : 0.4, 0.3, 0.2, 0.1

Weighted Moving Average

week	sales	4WMA
1	0.1x 39	
2	0.2x 44	
3	0.3x 40	
4	0.4x 45	
5	38	
6	43	
7	39	
8		

Weights : 0.4, 0.3, 0.2, 0.1

$F_5 = 0.4(45) + 0.3(40) + 0.2(44) + 0.1(39)$

Weighted Moving Average

week	sales	4WMA
1	39	
2	0.1x 44	
3	0.2x 40	
4	0.3x 45	
5	0.4x 38	42.7
6	43	41.1
7	39	
8		

Weights : 0.4, 0.3, 0.2, 0.1

$F_5 = 0.4(45) + 0.3(40) + 0.2(44) + 0.1(39)$

$F_6 = 0.4(38) + 0.3(45) + 0.2(40) + 0.1(44)$

Weighted Moving Average

week	sales	4WMA
1	39	
2	44	
3	0.1x 40	
4	0.2x 45	
5	0.3x 38	42.7
6	0.4x 43	41.1
7	39	41.6
8		

Weights : 0.4, 0.3, 0.2, 0.1

$F_5 = 0.4(45) + 0.3(40) + 0.2(44) + 0.1(39)$

$F_6 = 0.4(38) + 0.3(45) + 0.2(40) + 0.1(44)$

$F_7 = 0.4(43) + 0.3(38) + 0.2(45) + 0.1(40)$

Weighted Moving Average

week	sales	4WMA
1	39	
2	44	
3	40	
4	0.1x 45	
5	0.2x 38	42.7
6	0.3x 43	41.1
7	0.4x 39	41.6
8		40.6

Weights : 0.4, 0.3, 0.2, 0.1

$F_5 = 0.4(45) + 0.3(40) + 0.2(44) + 0.1(39)$

$F_6 = 0.4(38) + 0.3(45) + 0.2(40) + 0.1(44)$

$F_7 = 0.4(43) + 0.3(38) + 0.2(45) + 0.1(40)$

$F_8 = 0.4(39) + 0.3(43) + 0.2(38) + 0.1(45)$

Simple Exponential Smoothing

	A_t	F_t
week	sales	forecast
1	39	
2	44	
3	40	
4	45	
5	38	
6	43	
7	39	

$F_{t+1} = F_t + \alpha(A_t - F_t)$

smoothing constant : $0 \leq \alpha \leq 1$

$F_{t+1} = \alpha A_t + (1 - \alpha)F_t$

Simple Exponential Smoothing

	A_t	F_t
week	sales	forecast
1	39	
2	44	39.00
3	40	40.00
4	45	40.00
5	38	41.00
6	43	40.40
7	39	40.92

Let $\alpha = 0.2$,

$F_{t+1} = 0.2\ A_t + 0.8\ F_t$

$F_2 = A_1$

$F_3 = 0.2(44) + 0.8(39.00)$

$F_4 = 0.2(40) + 0.8(40.00)$

$F_5 = 0.2(45) + 0.8(40.00)$

$F_6 = 0.2(38) + 0.8(41.00)$

$F_7 = 0.2(43) + 0.8(40.30)$

Simple Exponential Smoothing

	A_t	F_t
week	sales	forecast
1	39	
2	44	39.00
3	40	40.00
4	45	40.00
5	38	41.00
6	43	40.40
7	39	40.92
8		40.54

Let $\alpha = 0.2$,

$F_{t+1} = 0.2\ A_t + 0.8\ F_t$

$F_2 = A_1$

$F_3 = 0.2(44) + 0.8(39.00)$

$F_4 = 0.2(40) + 0.8(40.00)$

$F_5 = 0.2(45) + 0.8(40.00)$

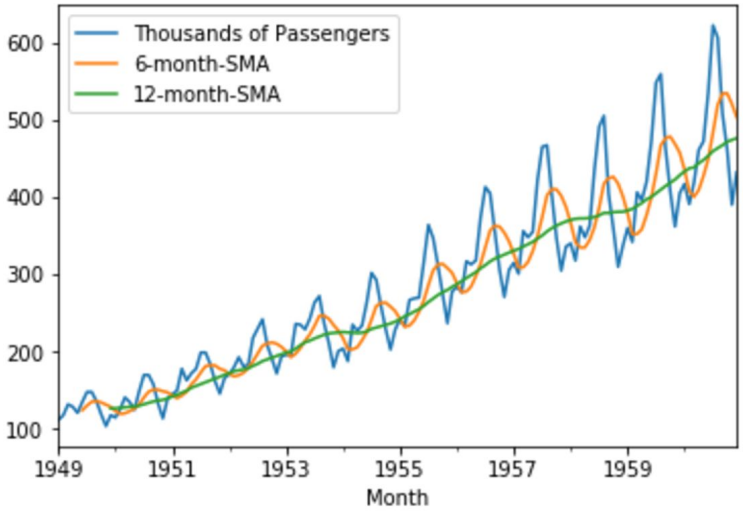
$F_6 = 0.2(38) + 0.8(41.00)$

$F_7 = 0.2(43) + 0.8(40.30)$

$F_8 = 0.2(39) + 0.8(40.92)$

EWMA 모델

- SMA (Simple Moving Averages) 를 통해서 간단한 트렌드 모델을 만들 수 있습니다.



EWMA 모델

- SMA 의 취약점
  - 윈도우 크기가 작을 수록 노이즈가 생기기 십상이다
  - 윈도우 크기만큼 lag 이 발생한다
  - 평균치이기 때문에 데이터에 상하 정점에 도달할 수가 없다
  - 데이터의 트렌드를 반영할 뿐 미래의 대한 예측 자료로서 근거가 약하다
  - 과거의 극단적으로 높거나 낮은 값들이 SMA 를 왜곡시킬 수 있다

시계열 데이터 분석

- SMA 의 문제점을 보완하기 위해 EWMA 를 (Exponentially Weighted Moving Average) 사용합니다.
- EWMA 는 SMA 의 lag 에 의한 영향을 경감시키고 최근의 데이터에 더 많은 가중치(weight)를 부여합니다.

시계열 데이터 분석 with Pandas

- Time Series with Pandas
- 시계열 데이터 특성
- ETS 모델
- SMA, WMA, SES 모델
- **ARIMA 모델**

ARIMA 모델

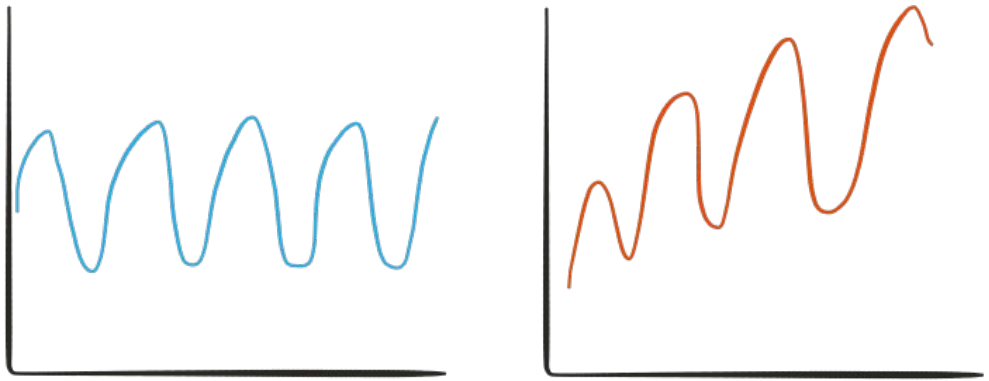
- Autoregressive Integrated Moving Average 는 개발된지 오래된 방법으로 시계열 데이터 분석을 위해 이해해야 하는 중요한 모델링 또는 예측 기법입니다.
- Stationary vs Non-stationary time series
- Seasonal vs Non-seasonal ARIMA
  - Non-seasonal ARIMA : ARIMA(p, d, q)
  - Seasonal ARIMA : ARIMA(p, d, q)(P, D, Q)m
- ARIMA
  - Autoregressive - AR(p)
  - Integrated - I(d)
  - Moving Average - MA(q)

Stationary vs Non-Stationary

- Stationary 데이터 특성
  - 연속되는 숫자들의 평균(mean)이 time invariant
  - 연속되는 숫자들의 분산(variance)이 time invariant
  - 연속되는 숫자들의 공분산(covariance)이 time invariant
- Stationary Test
  - ARIMA 모델은 시계열 데이터가 stationary 특성을 보일 때 효과적이므로 데이터가 stationary 특성을 보이는지 확인할 수 있어야 합니다.
- Differencing
  - 시계열 데이터가 Non-Stationary 하다면 초기 **differencing** 작업을 (“Integrated”) 한 번 이상 적용해서 데이터를 **stationary** 하게 만드는 단계가 필요합니다.
  -

Stationary 데이터 특성

- Stationary 데이터는 **평균(mean)**, 분산(variance), 공분산(covariance) 이 시점에 따라 달라지지 않습니다.

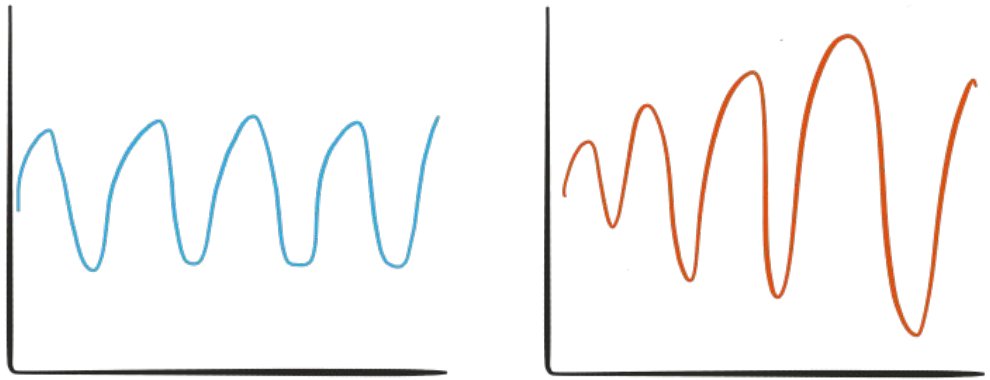


stationary

non-stationary

Stationary 데이터 특성

- Stationary 데이터는 평균(mean), **분산(variance)**, 공분산(covariance) 이 시점에 따라 달라지지 않습니다.

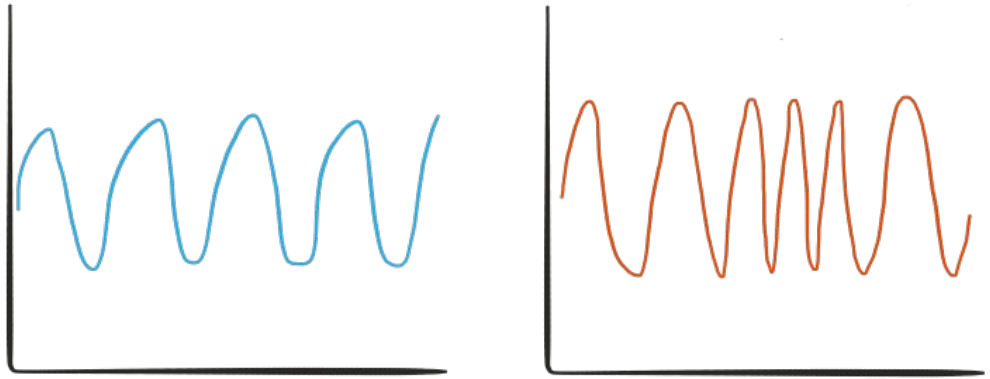


stationary

non-stationary

Stationary 데이터 특성

- Stationary 데이터는 평균(mean), 분산(variance), **공분산(covariance)** 이 시점에 따라 달라지지 않습니다.



stationary

non-stationary

Stationarity Test

- Augmented Dickey-Fuller test
  - 통계적 시험을 통해 시계열 데이터가 stationary 특성을 보이는지 확인할 수 있습니다.

In [181]:
```
from statsmodels.tsa.stattools import adfuller

dftest = adfuller(weekly_dn_ts, autolag='AIC')

print('''Test Statistic : {:.4f}'
Critical Value (1%) : {:.4f}
Critical Value (5%) : {:.4f}
Critical Value (10%) : {:.4f}'''.format(dftest[0],
                                        dftest[4]['1%'], dftest[4]['5%'], dftest[4]['10%']))
```

```
Test Statistic : -4.0462'
Critical Value (1%) : -3.4716
Critical Value (5%) : -2.8797
Critical Value (10%) : -2.5764
```

Differencing

- Non-stationary 데이터는 differencing 을 통해 stationary 하게 변환해줍니다.
- differencing 한 데이터에 대해 stationary 할 때까지 differencing 을 반복합니다.
- seasonal 데이터일 경우 season 을 기준으로 differencing 합니다.
  - 예를 들어 1년 주기의 seasonality 를 갖은 월간 데이터에 대해 differencing 할 때, differencing 의 시간 단위는 1 이 아니라 12 로 하게 됩니다.
  - seasonal ARIMA 모델의 경우 1차 differencing 후 seasonal differencing 하는 것도 흔히 사용하는 방법입니다.

Differencing

- Integrated - I(d)
  - $I(d) = Y_t - Y_{t-d}$

Original Data

Time1	10
Time2	12
Time3	8
Time4	14
Time5	7

First Difference

Time1	NA
Time2	2
Time3	-4
Time4	6
Time5	-7

Second Difference

Time1	NA
Time2	NA
Time3	-6
Time4	10
Time5	-13

Seasonal vs Non-seasonal

- ARIMA (p, d, q)
- ARIMA (p, d, q) (P, D, Q) m

Autoregression - AR(p)

- t 시점의 데이터와 이전 시점 (t-p; lagged p) 의 데이터 사이의 관계에 대한 회귀 모델 (regression model)
- $Y_t = \square_0 + \square_1 Y_{t-1} + \cdots\cdots + \square_p Y_{t-p} + e_t$

Moving Average - MA(q)

- t 시점의 데이터 이전 시점의 (t-q) moving average 의 residual 에 대한 회귀 모델
- $\varepsilon_t = \theta_0 + \theta_1\varepsilon_{t-1} + \cdots\cdots + \theta_p\varepsilon_{t-q} + e_t$

ARIMA(p, d, q)

- Autoregressive Integrated Moving Average
  - AR : A model that uses dependent relationship between an observation and some number of lagged observations.
  - I : The use of differencing of raw observations in order to make the time series stationary.
  - MA : A model that uses the dependency between an observation and a residual error from a MA model.
- parameters of ARIMA model
  - p : The number of lag observations included in the model
  - d : the degree of differencing, the number of times that raw observations are differenced
  - q : The size of moving average window.

Identification of ARIMA

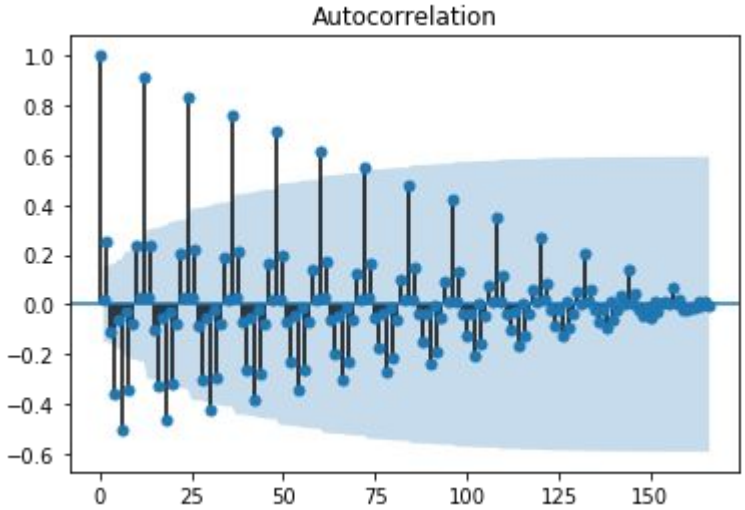
- Autocorrelation function(ACF) : measured by a simple correlation between current observation Y_t and the observation p lags from the current one Y_{t-p}.
- Partial Autocorrelation Function (PACF) : measured by the degree of association between Y_t and Y_{t-p} when the effects at other intermediate time lags between Y_t and Y_{t-p} are removed.
- Inference from ACF and PACF : theoretical ACFs and PACFs are available for various values of the lags of AR and MA components. Therefore, plotting ACFs and PACFs versus lags and comparing leads to the selection of the appropriate parameter p and q for ARIMA model

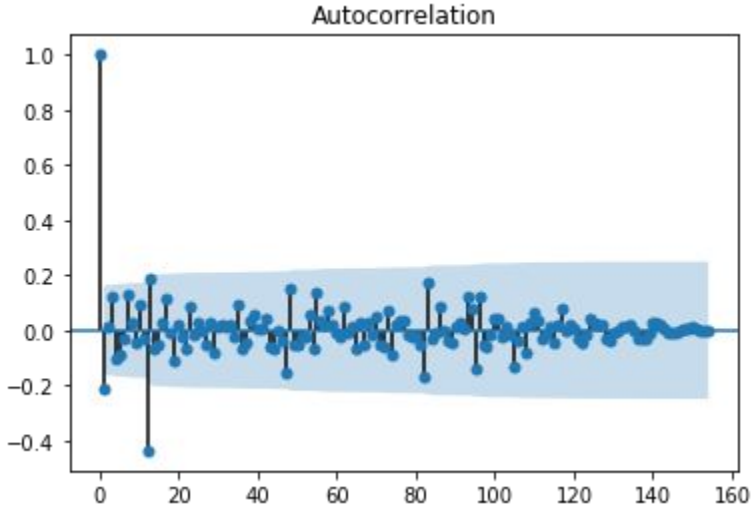
Identification of ARIMA

- I(d) : stationary 로 변환한 order d
- AR(p), MA(q):
  - AutoCorrelation 플롯과 Partial AutoCorrelation Plot 을 참고해서 p 와 q 를 결정합니다.

Identification of ARIMA

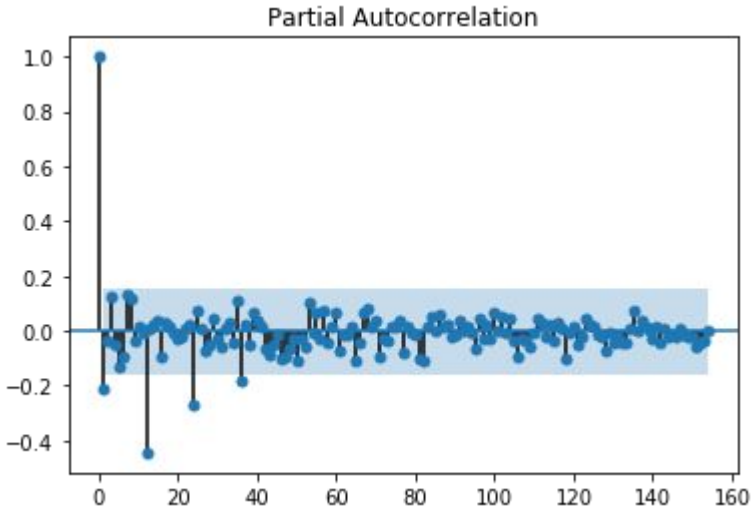
- AutoCorrelation Plot (a.k.a Correlogram)
  - 특정 시간만큼 지연된(lag) 시점의 데이터와의 연관성을 보여줍니다.





Identification of ARIMA

- Partial AutoCorrelation Plot
  - t 시점과 특정 시간만큼 지연된(lag) 시점 t-p 의 연관성을 그 사이 (t..t-p) 데이터의 영향을 배제하고 보여줍니다.



Identification of ARIMA

- General characteristics of theoretical ACFs and PACFs

model	ACF	PACF
AR(p)	Tail off; Spikes decay towards zero	Spikes cutoff to zero after lag p
MA(q)	Spikes cutoff to zero after lag q	Tails off; Spikes decay towards zero
ARMA(p,q)	Tails off; Spikes decay towards zero	Tails off; Spikes decay towards zero

- Reference :
  - http://people.duke.edu/~rnau/411arim3.htm
  - Prof. Robert Nau

Identification of ARIMA

```
In [25]: from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

fig = plt.figure(figsize=(15,6))
ax1 = fig.add_subplot(211)
plot_acf(series, ax=ax1)
ax2 = fig.add_subplot(212)
plot_pacf(series, ax=ax2)
plt.show()
```