

Day 10

Forecasting

Exercise #10

- Clothing Store Sales Example

- <http://www.census.gov/retail/> □ Monthly Retail Trade Report □ Time Series/Trend Charts: Create your own customizable time series □ Clothing store (4481) □ Clothing store sales from 2010 to 2018. Forecast monthly sales in 2019. 미국인구조사국에 가서 2010년부터 2018년 옷가게 세일데이터를 받은후 2019년 월별세일 예측

- Furniture Store Sales Example

- Find furniture store sales from 2010 to 2018 from the US Census and forecast monthly sales in 2019 2010년부터 2018년까지 가구점세일 데이터를 받은후 2019년 월별 가구세일 예측

Clothing Store Sales from US Census

옷가게 세일 예측

TIME SERIES / TREND CHARTS

Please follow the numbers in order.

1 Select the report/survey from which you wish to retrieve data:
Monthly Retail Trade and Food Services ▼

2 Select a date range:
Start: 2010 ▼ End: 2012 ▼

3 Select Industry or Category:
4481: Clothing Stores ▼

4 Select one Item :
Sales - Monthly ▼

5 Select Geographical Level:
U.S. Total ▼

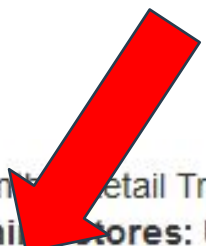
Select as available:

- ☐ Seasonally Adjusted
☒ Not Seasonally Adjusted
☐ Show Estimates of Sampling Variability

GET DATA

Monthly clothing store sales
from 2010 to 2012.

Source: Monthly Retail Trade and Food Services ([Definitions](#))
4481: Clothing Stores: U.S. Total — Not Seasonally Adjusted Sa
[TXT](#) [XLS-V](#) [XLS-H](#) [Bar Chart](#) [Line Chart](#)



Year	Jan	Feb	Mar	Apr
2010	9,931	10,605	13,174	12,951
2011	10,201	11,407	13,760	13,912
2012	10,752	12,720	15,342	14,148

Time Series Forecast Using RNN

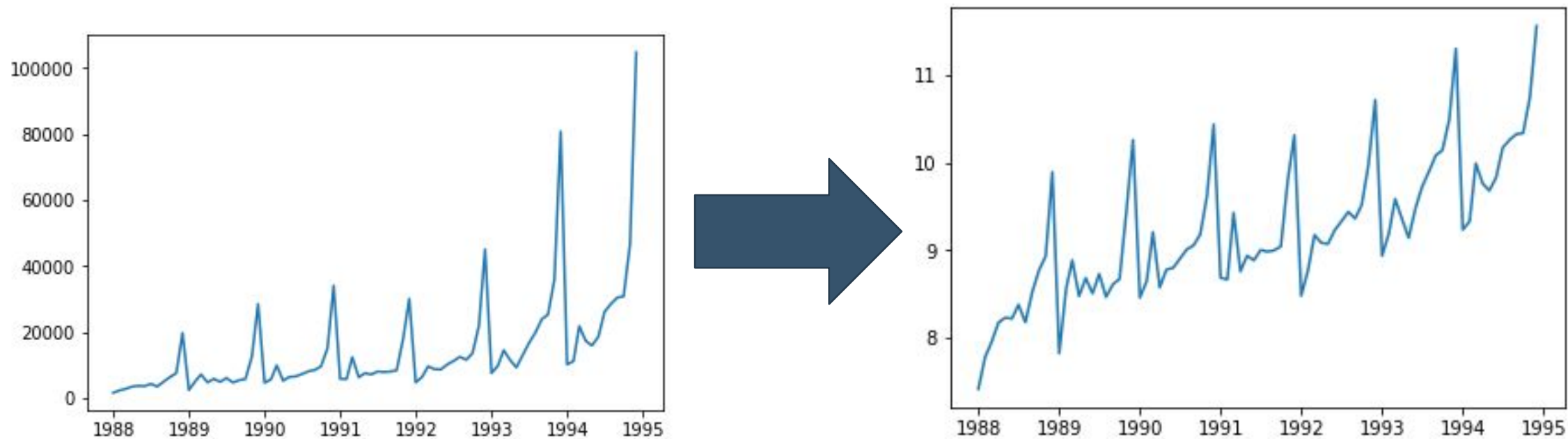
- 데이터로딩
- 시간열 제거 또는 인덱스화
- 시계열 데이터 정규화 (MinMaxScaler)
- 타임스텝에 따라 데이터셋을 만들어 x와 y로 구분
- 훈련, 테스트 데이터구분. e.g., 70%
- 샘플의수, 타임스텝의수, 변주의수로 reshape
- LSTM 모델 생성. e.g., LSTM (50 units), Dropout (.2), LSTM (100), Dropout (.2), Dense (1)
- 컴파일. e.g., loss=mse, optimizer=adam
- 학습. e.g., batch_size=512, epoch=1, validation_split= 0.05
- 예측 및 역정규화
- 평가. e.g., train and test RMSE, loss, mse chart, history table
- 예측시각화
- 테스트 데이터로 예측

Time Series Forecast With Seasonal Decomposition

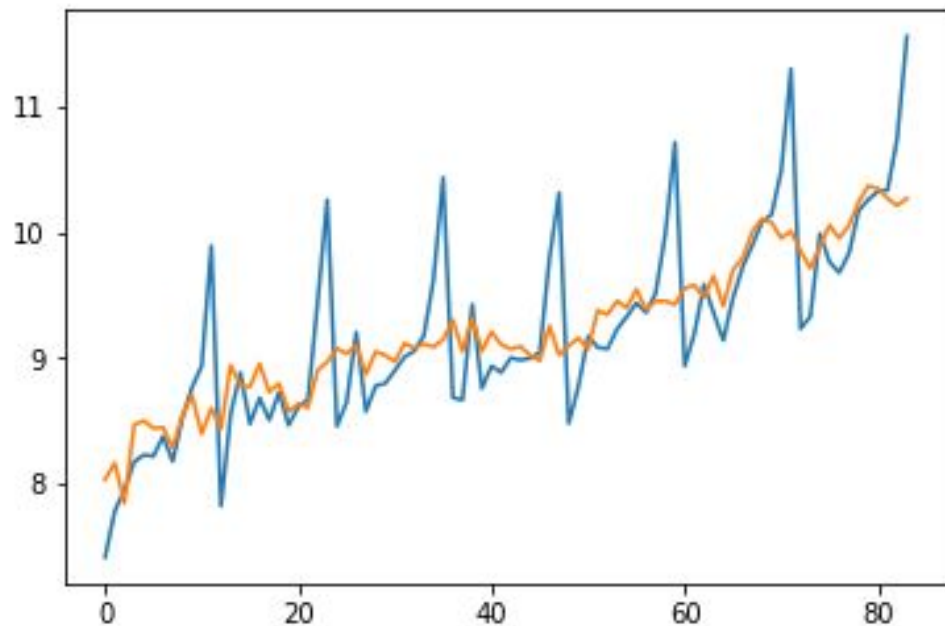
- 데이터로딩
- 데이터탐색
- 증폭되는 데이터면 로그변환
- 계절성 제거
- 회귀분석 (x 와 y 로 구분)
- 계절성 추가
- 지수변환
- 평가
- 시각화
- 테스트데이터로 예측

Log Transformation

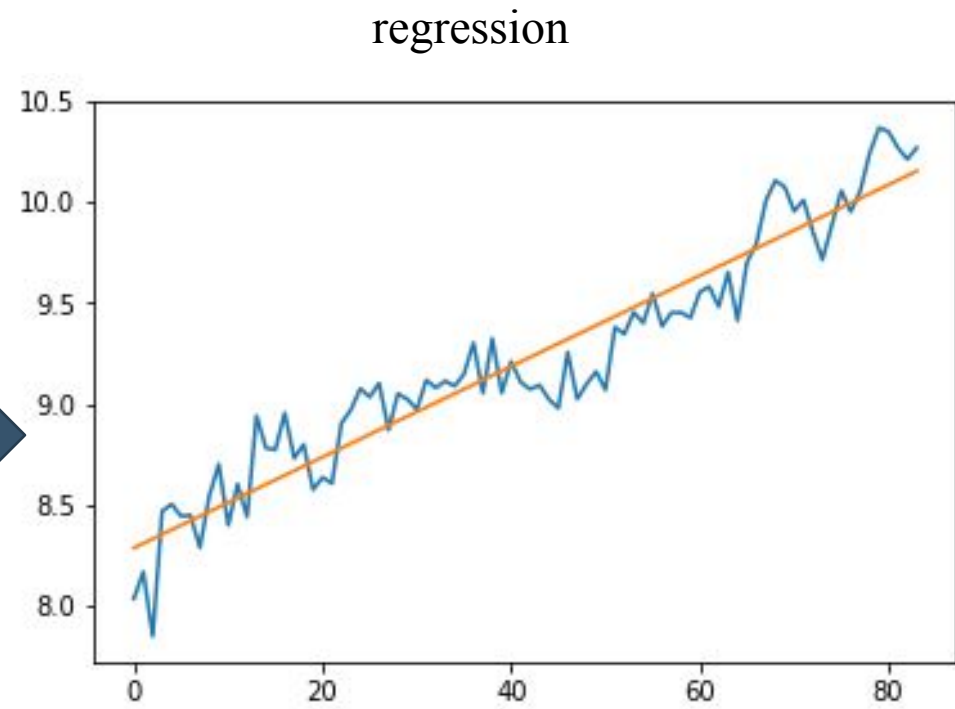
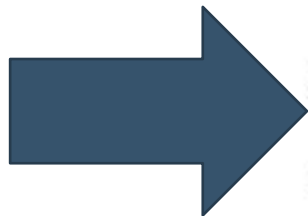
```
souvenir_log = np.log(ts)
```



Seasonal Decomposition to Trend

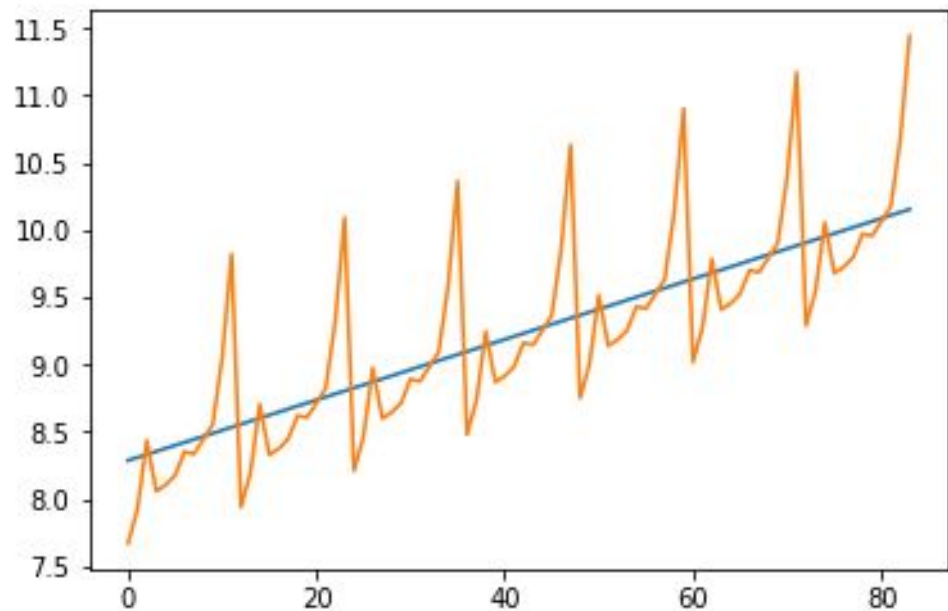


- seasonal

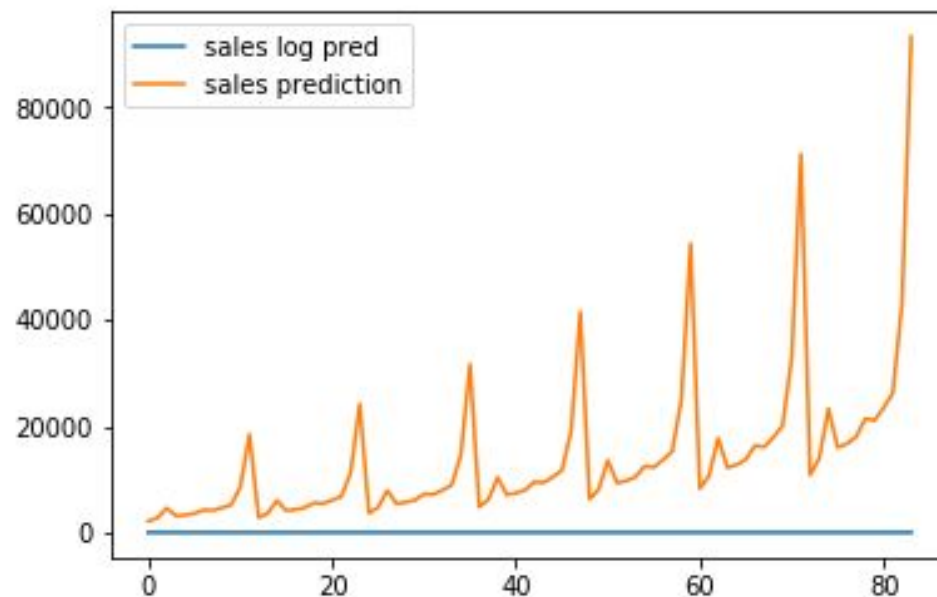
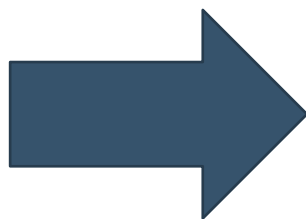


Recomposition

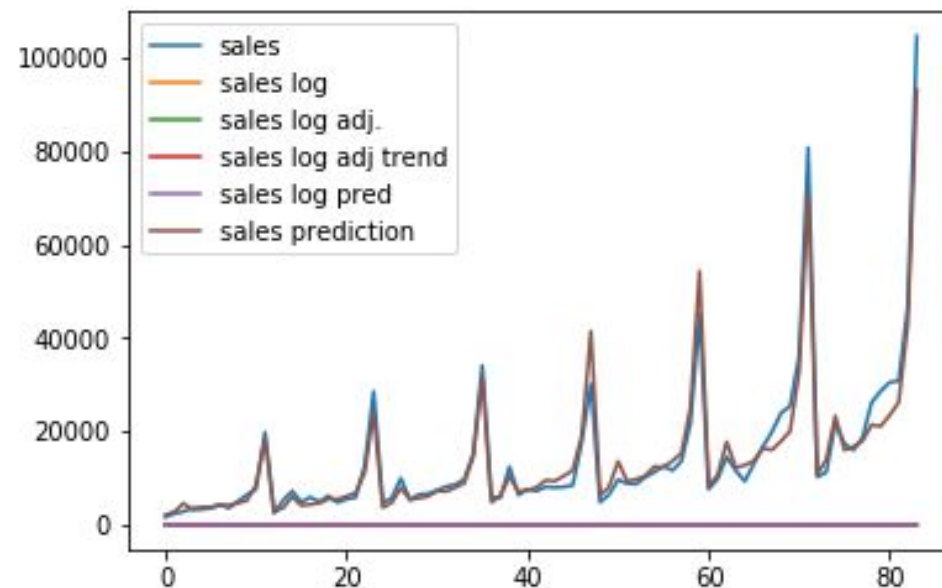
`np.exp(souvenir_log_pred)`



+ seasonal



Forecast Next Year's Sales



```
[ 14180.83493182  18278.10641411  30486.09457869  20929.09697533
 21892.05582565  23517.50491165  28040.34584245  27589.10065928
 30716.67823395  34369.422007    56326.12511216 122091.86430953]
```

Time Series Forecast Using Multiple Regression

- 데이터로딩
- 데이터 탐색 (`plt.plot`)
- 계절성으로 더미변수 생성 (`pd.get_dummies`, `OneHotEncoder`)
- 시간생성 (`np.arange`)
- 회귀분석 (`statsmodels.api`, `sklearn.linear_model`)
- 예측 (`model.predict`)
- 평가 (`me`, `mae`, `mape`, `mse`, `ts`)
- 시각화 (`plt.plot`)
- 테스트데이터로 예측 (`model.predict`)

Loading Stock Price

```
from pandas_datareader import data
start_date = '2010-01-01'
end_date = '2018-12-31 '
Stock = data.DataReader('INPX',
                        'yahoo', start_date, end_date)
```

Filling Null Values

- Forward-fill to propagate the previous value forward 앞에 있는 값으로 채움

```
data.fillna(method='ffill')
```

```
data.fillna(method='pad')
```

- Back-fill to propagate the next values backward 뒤에 있는 값으로 채움

```
data.fillna(method='bfill')
```

```
data.fillna(method='backfill')
```

Filling Null Values

- You can specify an axis along-which the fills take place
채워지는 축을 정함

```
df = pd.DataFrame([[1, np.nan, 2], [2, 3, 5], [np.nan, 4, 6]])  
df  
df.fillna(method='ffill', axis=1)
```

- You can also limit the numbers of fill 채워지는 값의 갯수를 제한

```
df.fillna(method='ffill', limit=1)
```

Filling Null Values

- Fill the missing values with many kinds of interpolations between the values 값들 사이의 값을 채우는 방법
- Example - linear, quadratic, polynomial, etc.

```
df.interpolate()
```

```
df.interpolate(method='time')
```

```
#consider time
```

Time Series Models

시계열모델의 종류

Smoothing Models 평활모델

- Simple Moving Average 단순이동평균법
- Weighted Moving Average 가중이동평균법
- Exponential Smoothing 지수평활법

Trend Models 추세모델

- Linear 일차
- Quadratic 이차
- Exponential 지수
- Auto regression 자기회귀

Gasoline Sales Example

가솔린세일 비선형추세 예측

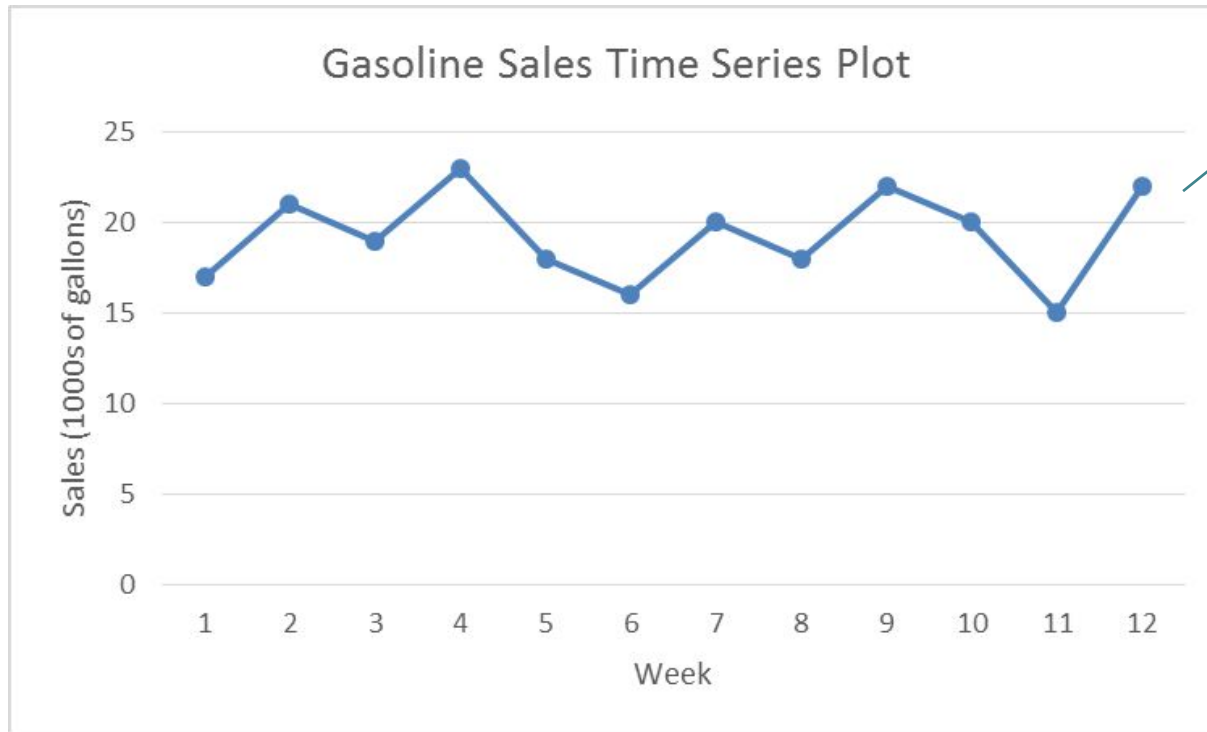
Week	Sales (1000s of gallons)
------	--------------------------

1	17
2	21
3	19
4	23
5	18
6	16
7	20
8	18
9	22
10	20
11	15
12	22

The table shows the number of gallons of gasoline sold by a gasoline distributor in Bennington, Vermont, over the past 12 weeks. Forecast the gasoline sales for week 13.

다음 테이블은 지난 12주동안 버몬트 베닝톤에 있는 가솔린 유통업자에 의해서 팔린 가솔린의 양 (갤론단위)을 보여주고 있습니다. 13주가 됐을때 가솔린 세일을 예측하십시오.

Time Series Plot 시계열차트



A horizontal pattern is present!

The data fluctuate around the sample mean of 19,250 gallons. 특정값 주위에서 변동됨

Three-Week Moving Average

3주 이동평균

Week	Sales (1000s of gallons)	3PMA
1	17	
2	21	
3	19	
4	23	19
5	18	21
6	16	20
7	20	19
8	18	18
9	22	18
10	20	20
11	15	20
12	22	19
13		19

- The average of the most recent three data values in the time series as the forecast for the next period
최근 3주를 평균냄
- Forecast for Week 13
 $= (20 + 15 + 22) / 3 = 19$

3MA Code

```
ts_pred = []
for i in range(len(ts)-2):
    ts_pred.append((ts[i+2] + ts[i+1] + ts[i])/3)
ts_pred
plt.plot(ts)
plt.plot(np.arange(3,13), ts_pred)
plt.ylim(0,25)
plt.legend(['sales', '3MA'])
```

Six-Week Moving Average

6주 이동평균

Week	Sales (1000s of gallons)	6PMA
1	17	
2	21	
3	19	
4	23	
5	18	
6	16	
7	20	19.00
8	18	19.50
9	22	19.00
10	20	19.50
11	15	19.00
12	22	18.50
13		19.50

- An equal weight is placed on each value that is being averaged. **가중치가 같음**
- Forecast for Week 13

$$= (20 + 18 + 22 + 20 + 15 + 22) / 6 = 19.5$$

Moving Average Forecast of Order k

k차의 이동평균예측

$$\begin{aligned} F_{t+1} &= \frac{\sum(\text{most recent } k \text{ data values})}{k} \\ &= \frac{Y_t + Y_{t-1} + \dots + Y_{t-k+1}}{k} \end{aligned}$$

Where

F_{t+1} = forecast of the times series for period t + 1

Y_t = actual value of the time series in period t

6MA Code

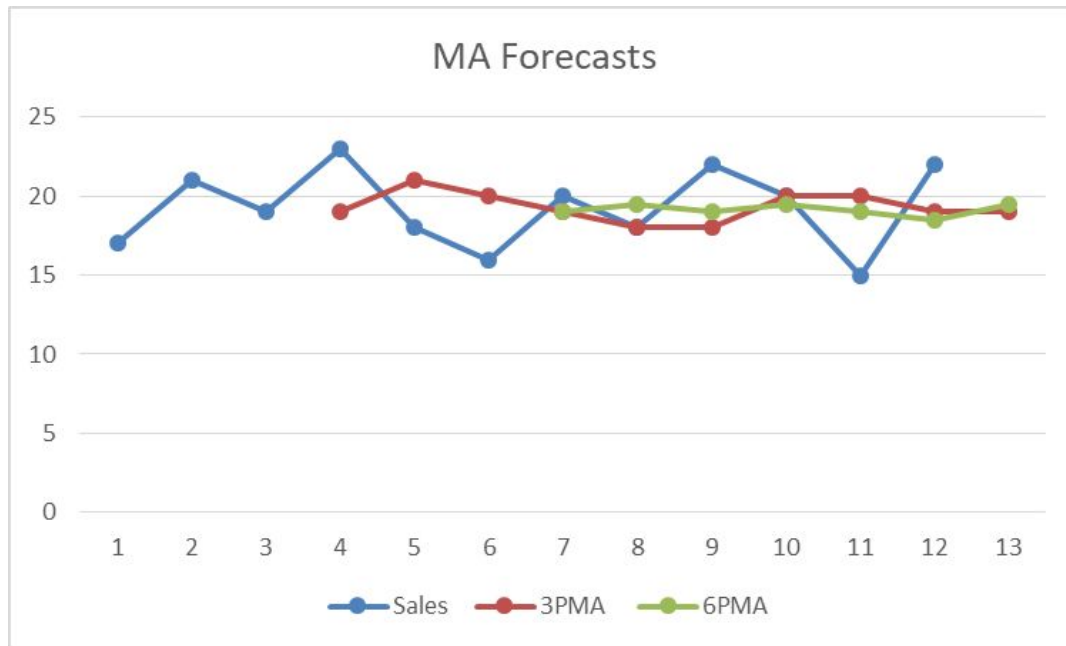
```
ts_pred = []
for i in range(len(ts)-5):

    ts_pred.append((ts[i+5]+ts[i+4]+ts[i+3]+ts[i+2]+ts[i+1]+ts[i])
/6)
ts_pred
plt.plot(ts)
plt.plot(np.arange(6,13), ts_pred)
plt.ylim(0,25)
plt.legend(['sales', '6MA'])
```

Order Selection 차수선택

- You must first select the order, or number of time series values, to be included in the moving average
이동평균을 낼때 우선 평균낼 기간을 선택
- Use trial and error to determine the value of k that minimizes MSE. 평균오차제곱값을 최소화
할수있는 기간을 여러번의 시도로 결정

Moving Average Chart 이동평균표



Longer gives more smoothing. 기간이 길수록 더 스무드한 예측

Shorter reacts quicker to trends. 시간이 짧을수록 더 트렌드를 잘 반영

The most accurate moving average forecasts of gasoline sales can be obtained using a moving average of order $k=6$ with $MSE = 6.79$
기간이 6일때 오차가 가장 작음.

rolling Function

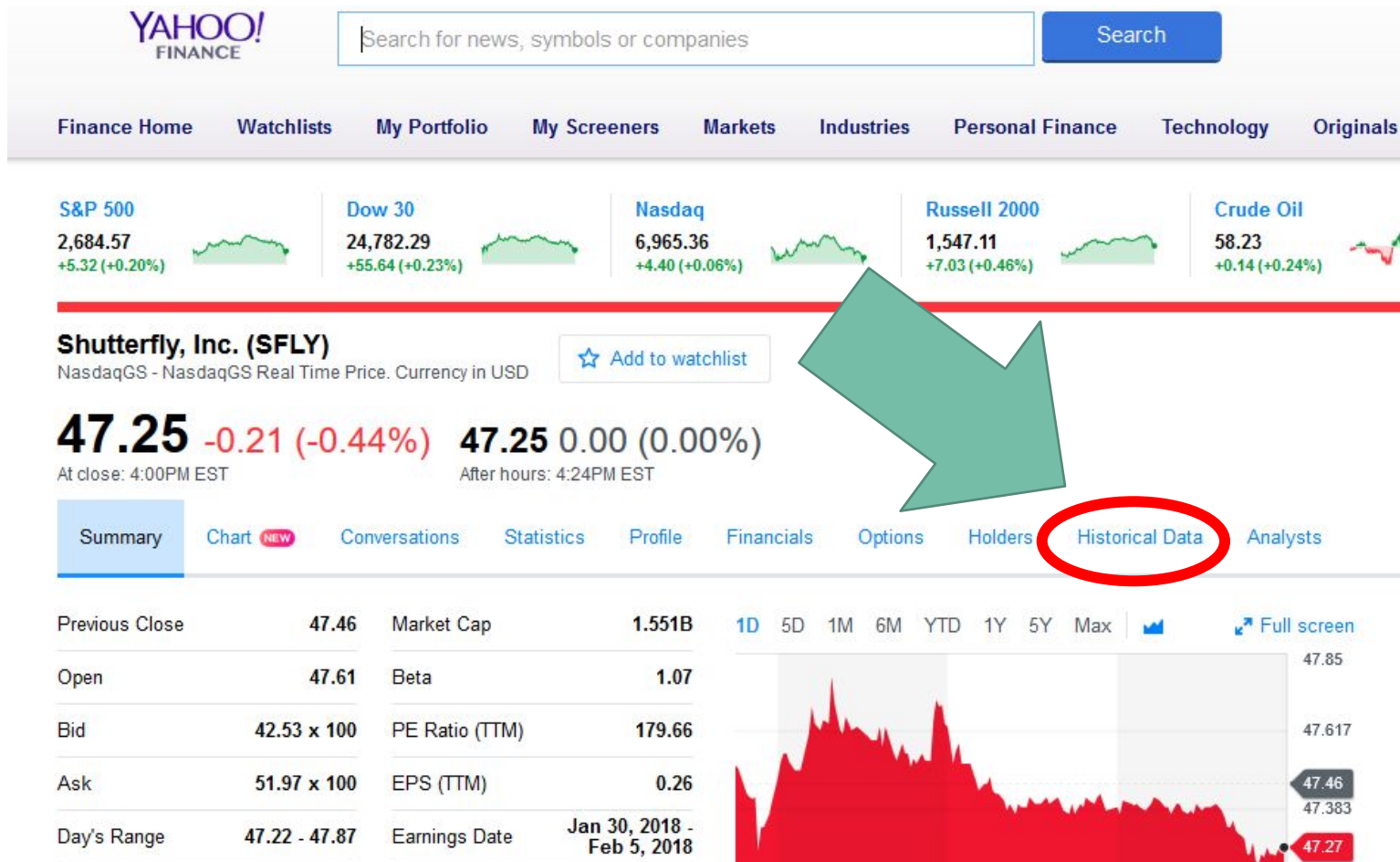
- Can be applied on a series of data. 연속된 데이터에 적용할수 있음
- Specify the window=n argument and apply the appropriate statistical function on top of it 윈도우갯수를 정해준후 적절한 함수를 적용

rolling Code

```
gas = np.array([17, 21, 19, 23, 18, 16, 20, 18,
22, 20, 15, 22])
df = pd.Series(gas)
roll3 = df.rolling(window=3).mean()
roll6 = df.rolling(window=4).mean()
plt.plot(gas, label='original')
plt.plot(roll3, label='3ma')
plt.plot(roll6, label='6ma')
plt.legend()
```

Stock Price Forecasting

주식가격 예측예제



Exercise #10

- Download the historical daily stock prices of Shutterfly from yahoo finance (approximately two years). Use the close prices for the analysis
야후 파이낸스에서 셔터플라이의 주식가격을 다운로드 받아서 종가를 이용하여 분석하시요
- Use the linear regression to forecast next 30 days' stock price
회귀분석을 이용하여 30일 예측
- Use the rolling function (20 days, 100 days) to forecast the next day's stock price
롤링함수를 이용하여 다음날 주식가격 예측
- Use the RNN to forecast next 30 days' stock price
RNN이용하여 30일 예측
- Compare three methods. 비교
- Draw a forecast chart 시각화

Three-Week Weighted Moving Average

3주 가중이동평균

Week	Sales (1000s of gallons)	Wt.	WMA
1	17	0.166667	
2	21	0.333333	
3	19	0.5	
4	23		19.33
5	18		21.33
6	16		19.83
7	20		17.83
8	18		18.33
9	22		18.33
10	20		20.33
11	15		20.33
12	22		17.83
13			19.33

- The weighted average of the most recent three values as the forecast 최근 3주동안 가중평균
- Forecast for Week 13 = $(3/6)*22 + (2/6)*15 + (1/6)*20 = 19.33$

Weighted Moving Average

가중이동평균

$$F_{t+1} = \sum w_t A_t$$

Where

F_{t+1} = forecast of the times series for period $t + 1$

A_t = actual value of the time series in period t

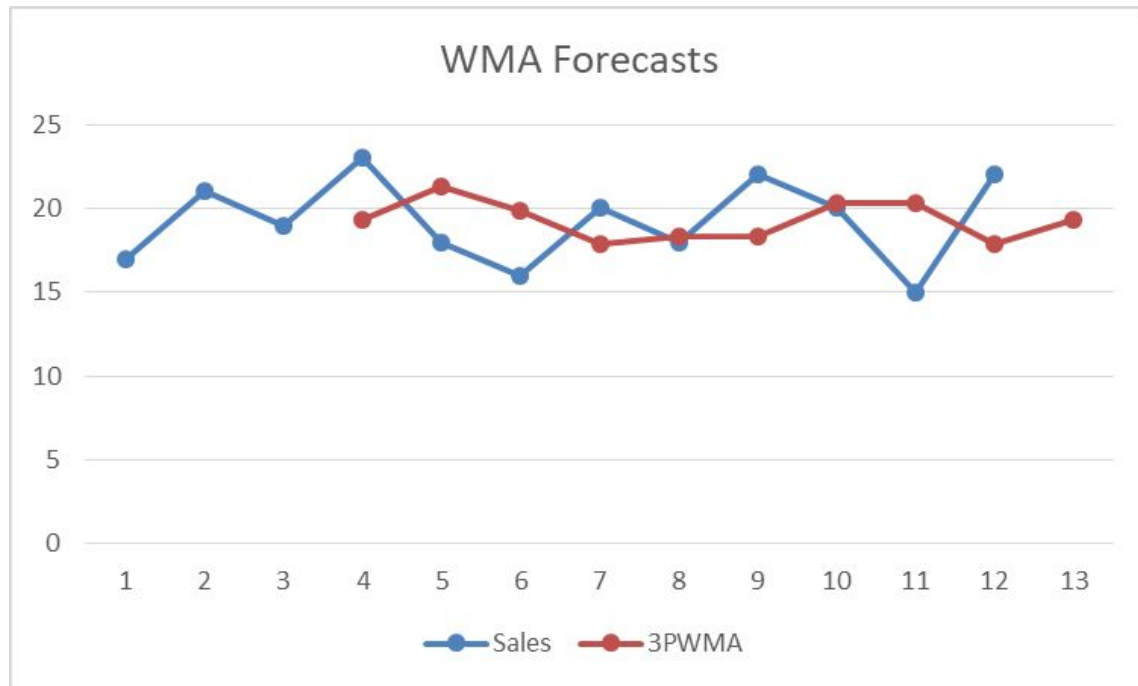
- All the weights must sum to one.
- The weighted moving average permits an unequal weighting on prior time periods.

3WMA Code

```
ts_pred = []
for i in range(len(ts)-2):
    ts_pred.append(sales[i+2]*(3/6) + sales[i+1]*(2/6) +
sales[i]*(1/6))
ts_pred
plt.plot(ts)
plt.plot(np.arange(3,13), ts_pred)
plt.ylim(0,25)
plt.legend(['sales', '3WMA'])
```

Weighted Moving Average Chart

가중이동평균차트



- More responsive to trends because of usually more weight on recent data!

일반적으로 최근 데이터에 더 가중치를 두기 때문에 더 변화를 잘 반영

Weight Selection 가중치 선택

- Use trial and error to determine the number of data values and weights. 이것도 여러번의 시도로 가중치를 결정
- If the recent past is a better predictor of the future than the distant past, larger weights should be given to the more recent observations. 가까운 년도가 더 영향력이 있는 경우가 많아서 최근것에 더 가중치를 둠
- When the time series is highly variable, selecting approximately equal weights for the data values may be best. 너무 변동이 많은 경우는 같은 가중치를 줌
- Use the combination of number of data values and weights that minimizes MSE! MSE를 최소로 하는 값을 찾음

Exponential Smoothing 지수평활

Week	Sales (1000s of gallons)	alpha	Exp
1	17	0.2	
2	21		17.00
3	19		17.80
4	23		18.04
5	18		19.03
6	16		18.83
7	20		18.26
8	18		18.61
9	22		18.49
10	20		19.19
11	15		19.35
12	22		18.48
13			19.18

- The weighted average of actual value in period 12 and the forecast for period 12. 지난기간의 실제값과 예측값의 가중평균

- Forecast for Week 13 = $.2 * 22 + (1 - .2) * 18.48$
 $= 18.48 + (22 - 18.48) * .2 = 19.18$

- The forecast for week2 equals the actual value of the time series in week1 (naïve method). 처음 예측값은 일주의 실제값으로 함

Exponential Smoothing Forecast

지수평활예측

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

Where

F_{t+1} = forecast of the time series for period $t + 1$

Y_t = actual value of the time series in period t

F_t = forecast of the time series for period t

α = smoothing constant ($0 \leq \alpha \leq 1$)

- Need just three pieces of data to start: last period's forecast, last period's actual value, smoothing coefficient, α .

Exp Code

```
ts_pred = [np.nan]*13
ts_pred[1] = ts[0]
for i in range(len(ts)-1):
    ts_pred[i+2] = ts[i+1]*(.2) + ts_pred[i+1]*(.8)
ts_pred
plt.plot(ts)
plt.plot(ts_pred)
plt.ylim(0,25)
plt.legend(['sales', 'exp. smoothing'])
```

Alpha Selection 알파의 선택

- Use trial and error to determine the value of alpha minimizes the MSE! 여러번의 시도로 MSE를 최소화하는 알파값을 구함
- Larger values of the smoothing constant allows the forecast to react more quickly to changing conditions. 숫자가 클수록 변화에 더 잘 반응하는 예측을 할수있음

α = 0.05-0.1, relatively stable

α = 0.15-0.3, rapid growth

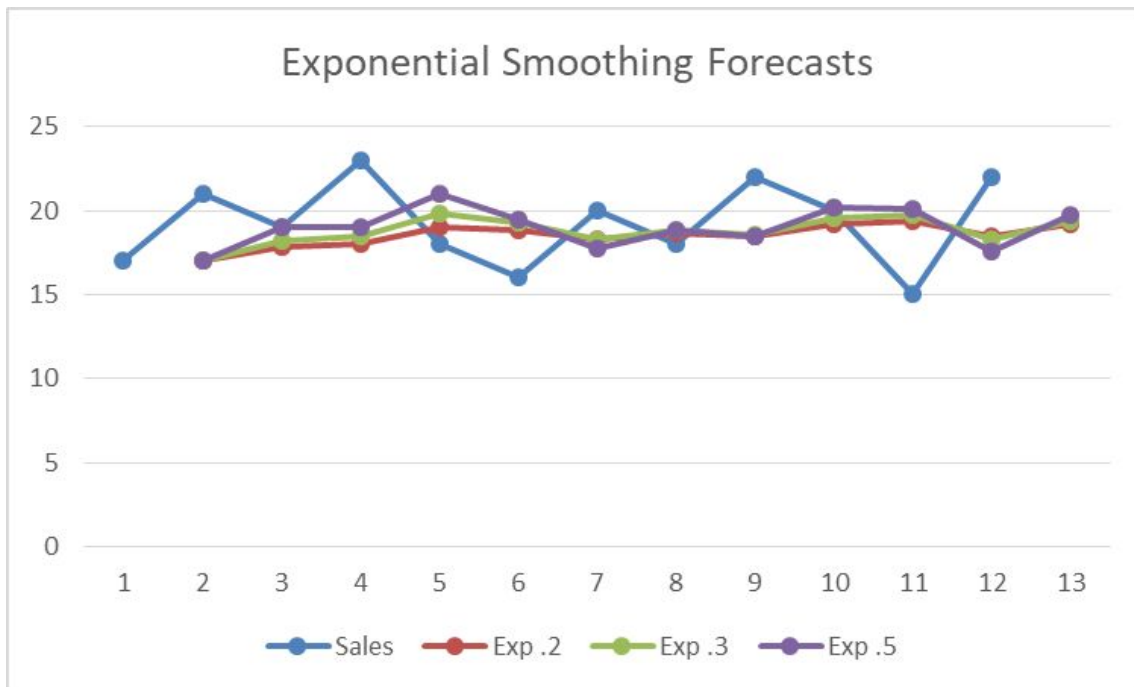
Different Alpha Values

알파값의 비교

		Forecast					
	Sales (1000s of gallons)						
Week		alpha	Exp	alpha	Exp	alpha	Exp
1	17	0.2		0.3		0.5	
2	21		17.00		17.00		17.00
3	19		17.80		18.20		19.00
4	23		18.04		18.44		19.00
5	18		19.03		19.81		21.00
6	16		18.83		19.27		19.50
7	20		18.26		18.29		17.75
8	18		18.61		18.80		18.88
9	22		18.49		18.56		18.44
10	20		19.19		19.59		20.22
11	15		19.35		19.71		20.11
12	22		18.48		18.30		17.55
13			19.18		19.41		19.78

Exponential Smoothing Chart

지수평활표



- Most frequently used method 가장 많이 사용되는 방법

.ewm Function

- Assigns the weights exponentially. 가중치를 기하급수적으로 할당
- Specify any of the com, span, halflife argument and apply the appropriate statistical function on top of it. 계수를 정해서 예측한뒤 함수를 적용

```
ts.ewm(alpha=.2).mean()
```

```
ts.ewm(alpha=.3).mean()
```

```
ts.ewm(alpha=.5).mean()
```


Exercise #9

Gasoline Sales Example

- Forecast week 13 using simple calculation, formulas, and forecast functions **단순계산, 공식, 예측함수등을 이용하여 13주를 예측**
 - Simple moving average (3 weeks, 6 weeks) **이동평균**
 - Weighted moving average (3/6, 2/6, 1/6) **가중이동평균**
 - Exponential smoothing method ($\alpha = .2, .3, .5$) **지수평활**

Exercise #9

- Use Umbrella Sales, TV Sets Sales, and Lawn-Maintenance Expense to forecast next time period using simple calculation, formulas, and forecast functions **단순계산, 공식, 예측함수를 이용하여 예측**
 - Simple moving average (3 quarters or months, 6 quarters or months) **이동평균**
 - Weighted moving average (3/6, 2/6, 1/6) **가중이동평균**
 - Exponential smoothing method ($\alpha = .2, .3, .5$) **지수평활**

Holt's Exponential Smoothing

홀트의 지수평활법

- Charles Holt developed a version of exponential smoothing that can be used to forecast a time series with level, trend, or seasonality
찰스홀트가 만든 지수평활법의 한 버전으로 레벨, 트렌드, 계절성을 예측하는 방법
- Smoothing is controlled by three parameters: alpha, beta, and gamma which estimate the level, slope and seasonal component at the current time point
평활은 세개의 변수에 의해 조정되는데 각각은 현재 시간포인트의 레벨, 기울기, 계절요소를 예측함

HoltWinters Function

`HoltWinters(myts, alpha, beta, gamma)`

- Alpha, beta and gamma all have values between 0 and 1
알파, 베타, 감마는 0과 1사이의 값이어야 함
- Initial states are selected heuristically
첫값은 휴리스틱한 방법으로 선택

Type of HoltWinters

홀트윈터스의 종류

Simple exponential

- models level

Double exponential

- models level and trend

Triple exponential

- models level, trend, and seasonal components

HoltWinters Formulas

홀트윈터스 공식들

(Level) $L_t = \alpha * (Y_t - S_{t-s}) + (1 - \alpha) * (L_{t-1} + b_{t-1})$

(Trend) $b_t = \beta * (L_t - L_{t-1}) + (1 - \beta) * b_{t-1}$

(Seasonal) $S_t = \gamma * (Y_t - L_t) + (1 - \gamma) * S_{t-s}$

(Forecast for period m) $F_{t+m} = L_t + m*b_t + S_{t+m-s}$

m —Number of periods ahead to forecast

s —Length of the seasonality

L_t —Level of the series at time t

b_t —Trend of the series at time t

S_t —Seasonal component at time t

Auger's Plumbing Service Example

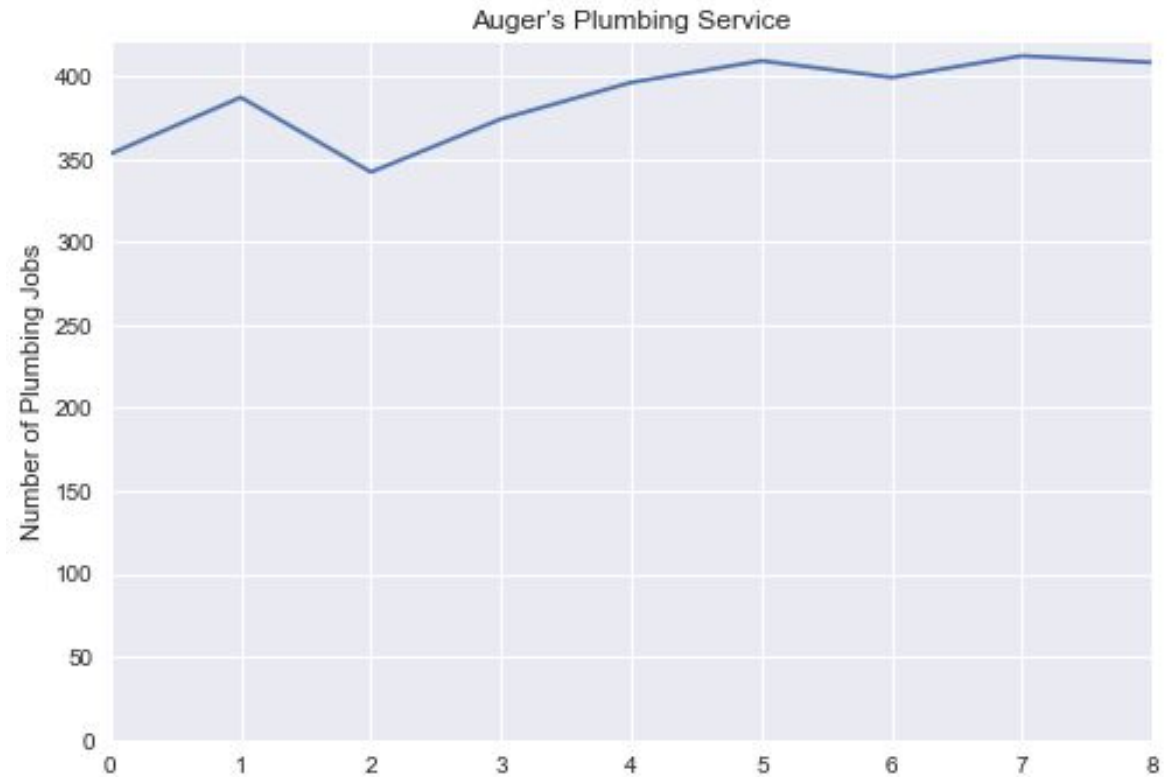
어거스 배관서비스 예제

Forecast the number of plumbing jobs. Auger's will have in months April through December using Holt's exponential smoothing method, with $\alpha = .1$ and $\beta = .2$.
테이블에 어거스 배관서비스의 4월부터 12월까지의 작업수를 보여주고 있습니다. 홀트의 지수평활법을 이용하여 (알파 = 0.1, 베타 = 0.2) 배관작업의 수를 예측해 보십시오.

Month	Jobs	Month	Jobs
April	353	September	409
May	387	October	399
June	342	November	412
July	374	December	408
August	396		

Time Series Chart

```
auger = np.array([353, 387, 342,  
374, 396, 409, 399, 412, 408])  
auger = pd.Series(auger)  
plt.style.available  
plt.style.use("seaborn")  
auger.plot()  
plt.title("Auger's Plumbing  
Service")  
plt.ylabel("Number of Plumbing  
Jobs")  
plt.ylim(0, 420)  
plt.show()
```



Simple Exponential Smoothing

단수지수평활

- Models **level** without trend and seasonal component (`beta=False`, `gamma=False`) 추세와 계절요소 없는 데이터를 예측
- Unknown parameters are determined by minimizing the squared prediction error 에러를 최소화하는 방향으로 계수를 정함

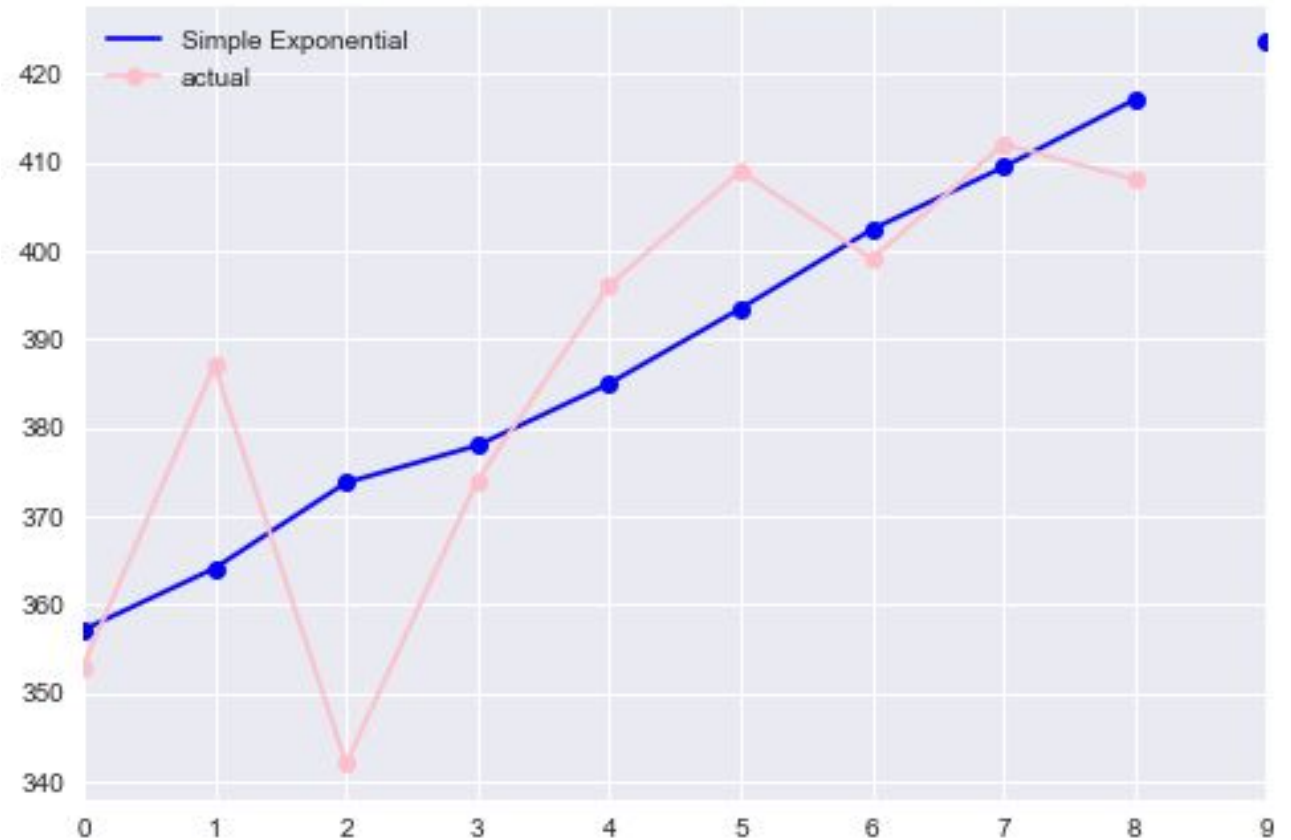
```
fit1 = Holt(ts).fit(smoothing_level=0.8)
```

```
fit1.forecast(12)
```

```
fit2 = Holt(ts).fit(smoothing_slope=False,  
optimized=True) #optimal
```

Simple Exponential Smoothing Code

```
from sklearn.metrics import  
mean_squared_error  
  
auger_sholt =  
Holt(auger).fit(smoothing_level=.1)  
  
y_pred_simple1 =  
auger_sholt.forecast(1).rename('Simple  
Exponential')  
  
auger.plot(marker='o', color='pink',  
label='actual', legend=True)  
  
auger_sholt.fittedvalues.plot(marker='o',  
color='blue')  
  
y_pred_simple1.plot(marker='o',  
color='blue', label='simple holt  
winter', legend=True)  
  
simple =  
np.sqrt(mean_squared_error(auger,  
auger_sholt.fittedvalues))
```



Comparison with Optimal Alpha

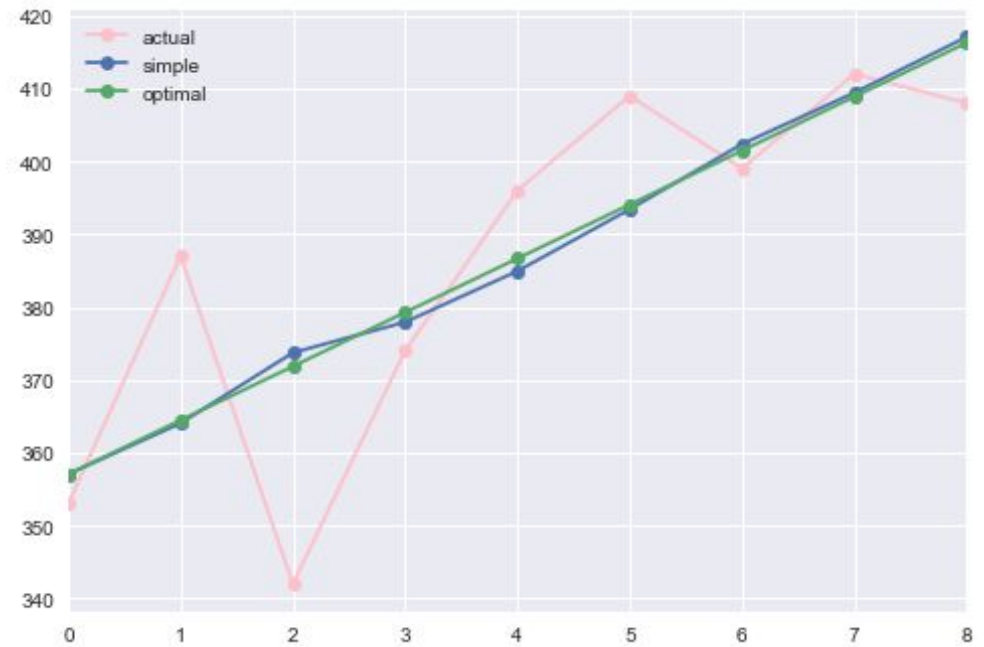
```
print(simple)
print(simple_optimal)
#better

auger.plot(marker='o',
color='pink',
label='actual')

auger.sholt.fittedvalues.p
lot(marker='o',
label='simple')

auger.sholt2.fittedvalues.
plot(marker='o',
label='optimal')

plt.legend()
```



Double Exponential Smoothing

이중지수평활

- Models level and **trend** without seasonal component (`gamma=False`) 계절성이 없고 추세만 있는 데이터를 예측

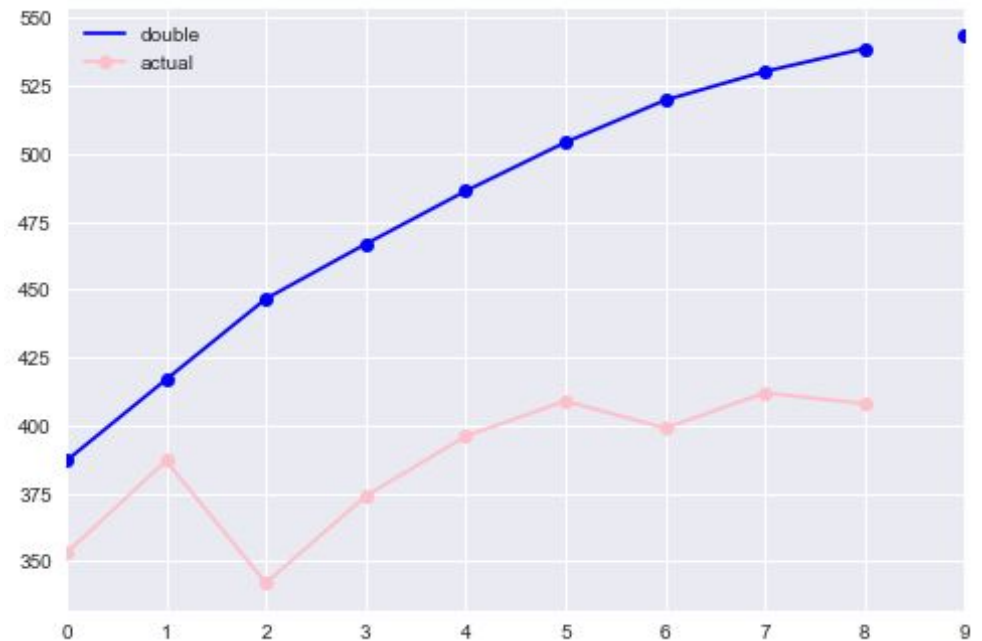
```
fit1 = Holt(ts).fit(smoothing_level=0.8,  
smoothing_slope=0.2)
```

```
fit1.forecast(12)
```

```
fit2 = Holt(auger).fit(optimized=True)
```

Double Exponential Smoothing Code

```
auger_dholt =  
Holt(auger).fit(smoothing_level=.1,  
smoothing_slope=.2)  
  
y_pred =  
auger_dholt.forecast(1).rename('Double')  
  
auger_dholt.fittedvalues.plot(marker='o', color='blue')  
  
y_pred.plot(marker='o',  
color='blue', legend=True,  
label='double')  
  
auger.plot(marker='o', color='pink',  
legend=True, label='actual')  
  
double =  
np.sqrt(mean_squared_error(auger,  
auger_dholt.fittedvalues))
```



Comparison with Optimal Beta

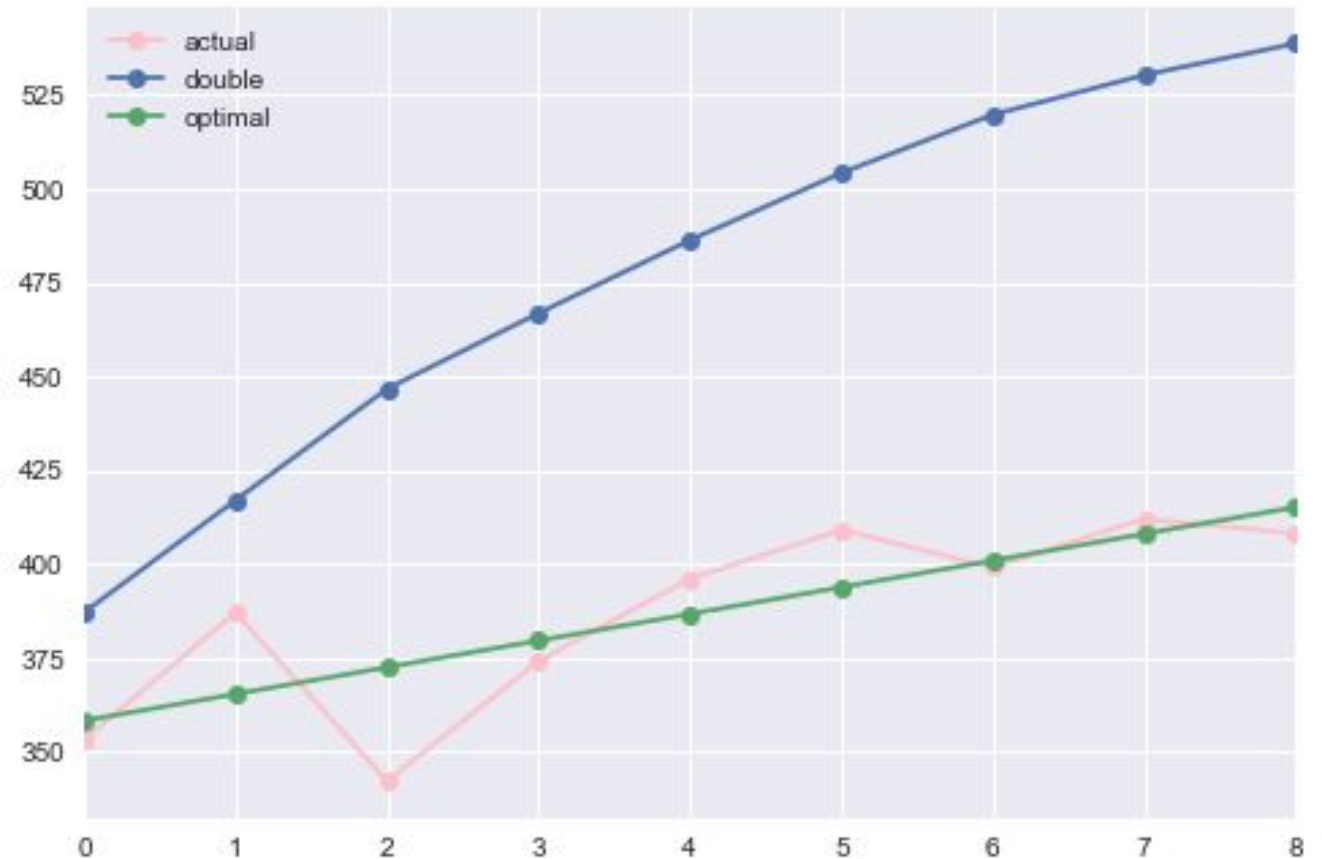
```
print(double)
print(double_optimal)
#much better

auger.plot(marker='o',
color='pink',
label='actual')

auger_dholt.fittedvalues.p
lot(marker='o',
label='double')

auger_dholt2.fittedvalues.
plot(marker='o',
label='optimal')

plt.legend()
```



Triple Exponential Smoothing

삼중지수평활

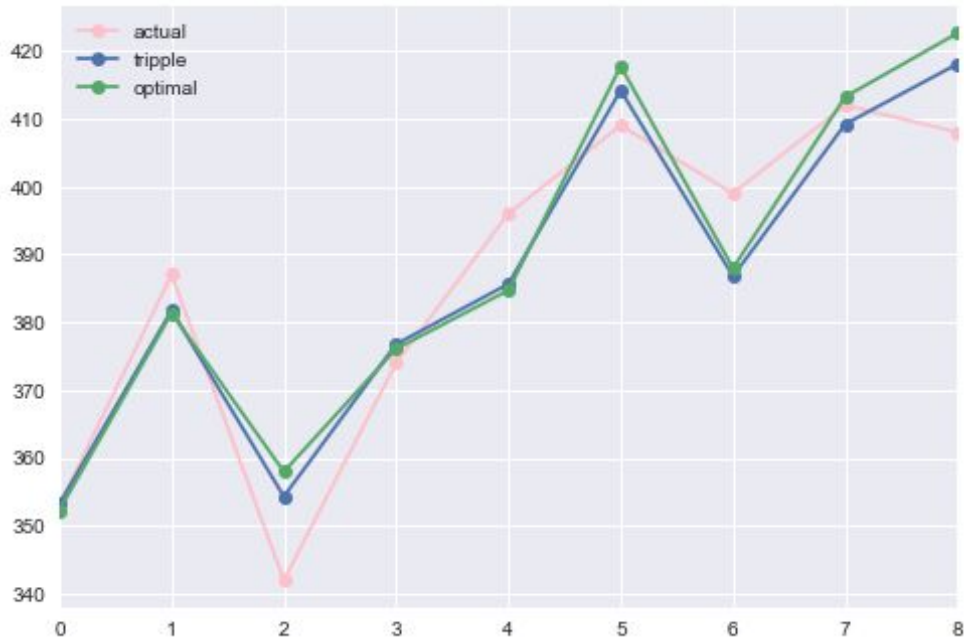
- Models level, trend, and **seasonal** component. 추세와 계절성 둘다 있는 데이터를 예측

```
fit1 = ExponentialSmoothing(salaedata,  
    seasonal_periods=4, trend='add', seasonal='add').fit()  
  
fit2 = ExponentialSmoothing(saledata, seasonal_periods=4,  
    trend='add', seasonal='mul').fit()  
  
y_pred = fit1.forecast(12)
```

Triple Exponential Smoothing Code

```
auger_tripple1 = ExponentialSmoothing(auger,  
seasonal_periods=4, trend='add', seasonal='add').fit()  
  
y_pred = auger_tripple1.forecast(1)  
  
auger_tripple1.fittedvalues.plot(marker='o',  
color='blue', legend=True)  
  
y_pred.plot(marker='o', color='blue')  
  
auger.plot(marker='o', color='pink', legend=True)  
  
tripple_add = np.sqrt(mean_squared_error(auger[0],  
auger_tripple1.fittedvalues))
```


Comparison with Optimal Gamma



```
auger.plot(marker='o',  
color='pink',  
label='actual')
```

```
auger_tripple1.fittedvalues.plot(marker='o',  
label='tripple')
```

```
auger_tripple2.fittedvalues.plot(marker='o',  
label='optimal')
```

```
plt.legend()
```

Exercise #10

Auger's Plumbing Service Example

- Simple exponential smoothing ($\alpha=.1$, $\beta=\text{FALSE}$, $\gamma=\text{FALSE}$). Simple exponential smoothing ($\beta=\text{FALSE}$, $\gamma=\text{FALSE}$). Compare the accuracy of these two methods
- Double exponential smoothing ($\alpha=.1$, $\beta=.2$, $\gamma=\text{FALSE}$). Double exponential smoothing ($\gamma=\text{FALSE}$). Compare the accuracy of these two methods
- Compare the accuracy of simple optimal and double optimal.
- Tripple exponential smoothing ($\text{seasonal_periods}=4$, $\text{seasonal}=\text{'add'}$). Tripple exponential smoothing ($\text{seasonal_periods}=4$, $\text{seasonal}=\text{'mul'}$). Compare the accuracy of these two methods

Souvenir Shop Sales Example

The file below contains monthly sales for a souvenir shop at a beach resort town in Queensland, Australia, for January 1987-December 1993. Forecast the monthly sales for 1994.

다음의 파일은 오스트렐리아 퀸즈랜드 해변가 리조트에 있는 기념품가게의 1987년 1월부터 1993년 12월까지의 월별매출입니다. 1994년 월별 매출을 예측하시요.

<http://robjhyndman.com/tsdldata/data/fancy.dat>

Exercise #9

Souvenir Shop Sales Example

- Plot the sales and forecast using seasonal decomposition
계절분해를 이용하여 예측
- Triple exponential smoothing using HoltWinters 트리플
홀트윈터스로 예측
- Compare the accuracy of the triple exp. smoothing to the
seasonal decomposition method 정확성 비교

King Example

The file below contains data on the age of death of successive kings of England, starting with William the Conqueror. Forecast the age of death of next king. 아래 파일은 영국왕들의 죽은 나이를 포함합니다. 다음 왕의 죽을 나이를 예측하시요

<http://robjhyndman.com/tsdldata/misc/kings.dat>

Birth Example

A data set of the number of births per month in New York city, from January 1946 to December 1959 is available in the file.

Forecast the number of births per month for 1960. 데이터셋은 1946년 1월부터 1959년 12월까지 뉴욕시에서 달마다 태어난 아이들의 숫자를 포함합니다. 1960년에 태어날 아이들의 숫자를 예측하시요.

<http://robjhyndman.com/tsdldata/data/nybirths.dat>

Annual Rainfall Example

The file below contains total annual rainfall in inches for London, from 1813-1912. Forecast the annual rainfall for 1913. 다음 파일은 1813년부터 1912년까지 영국의 연간강수량 (인치) 을 포함합니다. 1913년 강수량을 예측하십시오.

<http://robjhyndman.com/tsdldata/hurst/precip1.dat>

Skirt Diameter Example

The time series of the annual diameter of women's skirts at the hem, from 1866 to 1911. The data is available in the file.

Forecast the diameter of women's skirts at the hem for 1912.

다음 파일은 1866년부터 1911년까지 여성치마단의 연간직경을 포함합니다. 1912년 여성치마단의 직경을 예측하시요.

hem, <http://robjhyndman.com/tsdldata/roberts/skirts.dat>

Volcano Example

The file below contains data on the volcanic dust veil index in the northern hemisphere, from 1500-1969. This is a measure of the impact of volcanic eruptions' release of dust and aerosols into the environment.

Forecast the volcanic dust veil index for 1970. 다음 파일은 1500년에서 1969년 북반구의 화산분진베일지수를 포함합니다. 이 지수는 화산폭발로 인한 먼지와 에어로졸의 영향을 측정합니다. 1970년의 화산분진베일지수를 예측하시요.

<http://robjhyndman.com/tsdldata/annual/dvi.dat>

Exercise #10

- Select the appropriate HoltWinters exponential smoothing method and compare the accuracy
적당한 홀트윈터스
지수평활법을 선택한 후 결과를 비교
 - Kings example
 - Births example
 - Rain example
 - Skirts example
 - Volcano example

ARIMA

Autoregressive Integrated
Moving Averages
(자기회귀누적이동평균)

Autoregressive Integrated Moving Averages 자기회귀누적이동평균

- Used for modeling and forecasting stationary, stochastic time-series processes 정지돼있고 확률적인 시계열 데이터를 예측할때 사용
- Allows none-zero autocorrelations in the irregular component. 불규칙요소안에 자기상관관계가 있음

Assumptions 가정의 차이점

HoltWinters

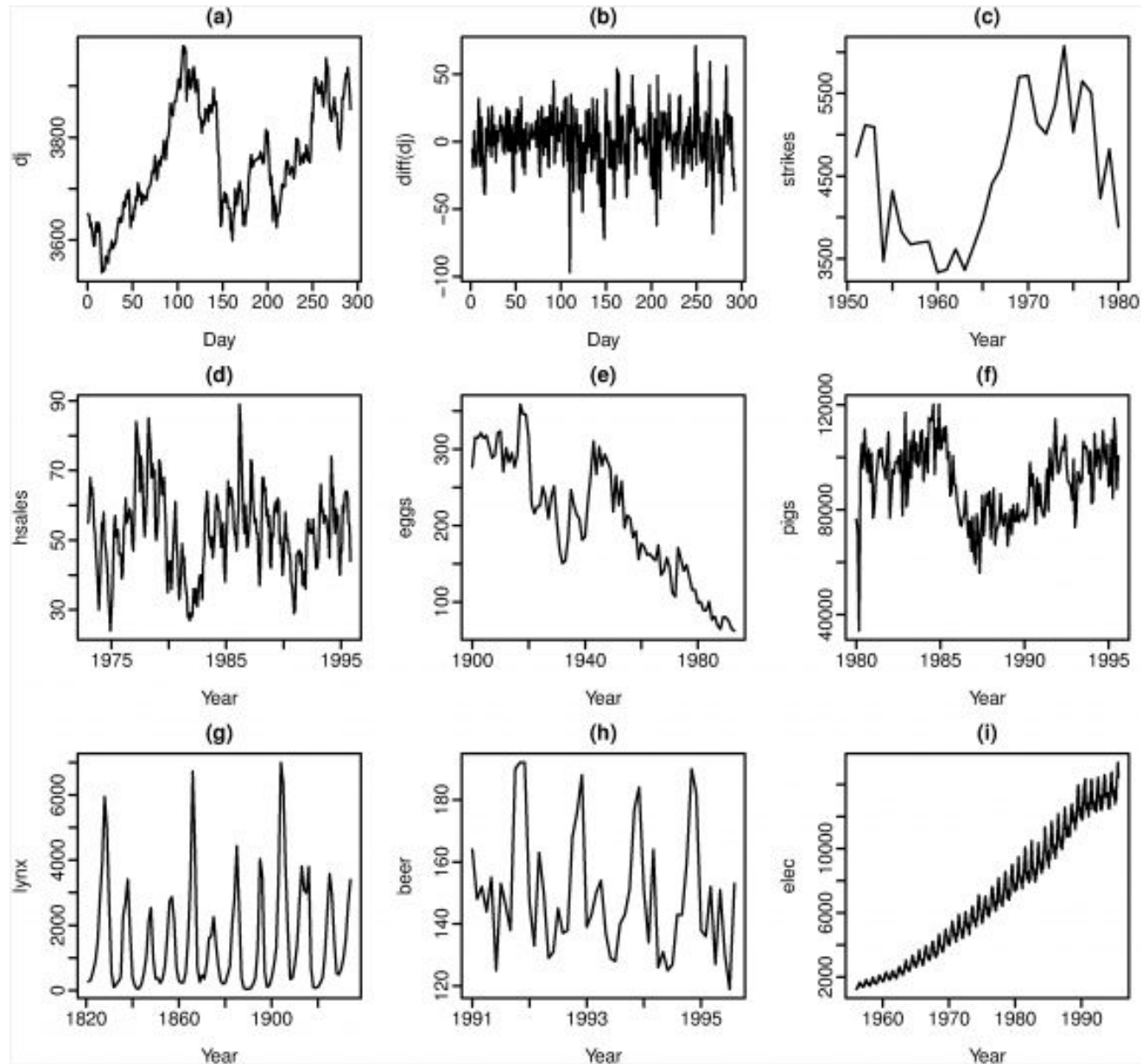
- Make no assumptions about the correlations between successive values of the time series.
데이터값사이에 상관관계에 대한 가정이 없음

ARIMA

- The forecast errors are uncorrelated and are normally distributed with mean zero and constant variance.
예측에러는 서로 관련이 없고, 평균이 0인 정규분포를 이루고 편차도 일정하다고 가정

Stationary Time Series 정상시계열

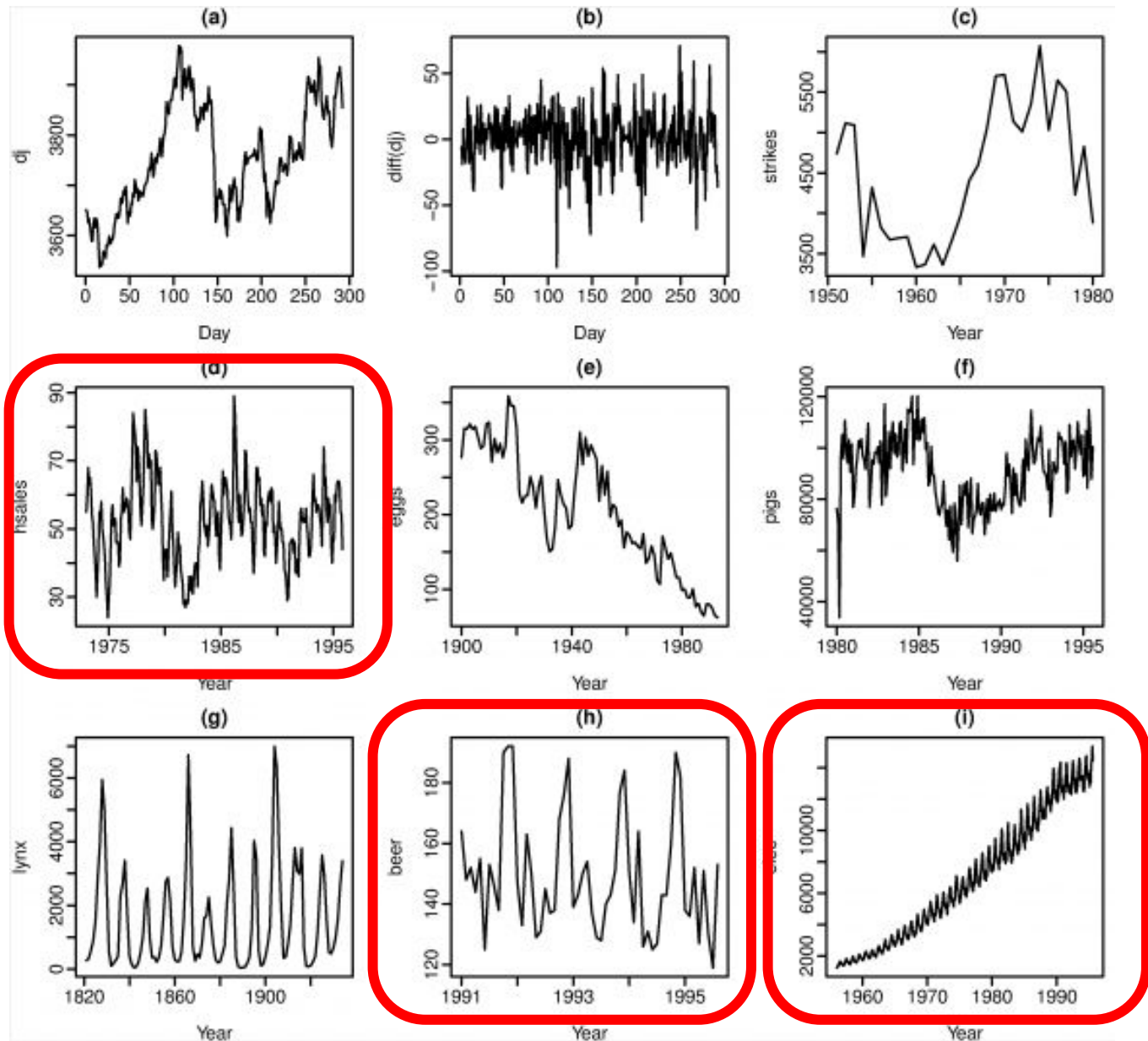
- Its properties do not depend on the time at which the series is observed.
시간이랑 상관없는 데이터
- Time series with no trends or no seasonality 트렌드도 없고, 계절성도 없는 데이터
- A white noise series is stationary 화이트노이즈가 대표적인 정상시계열



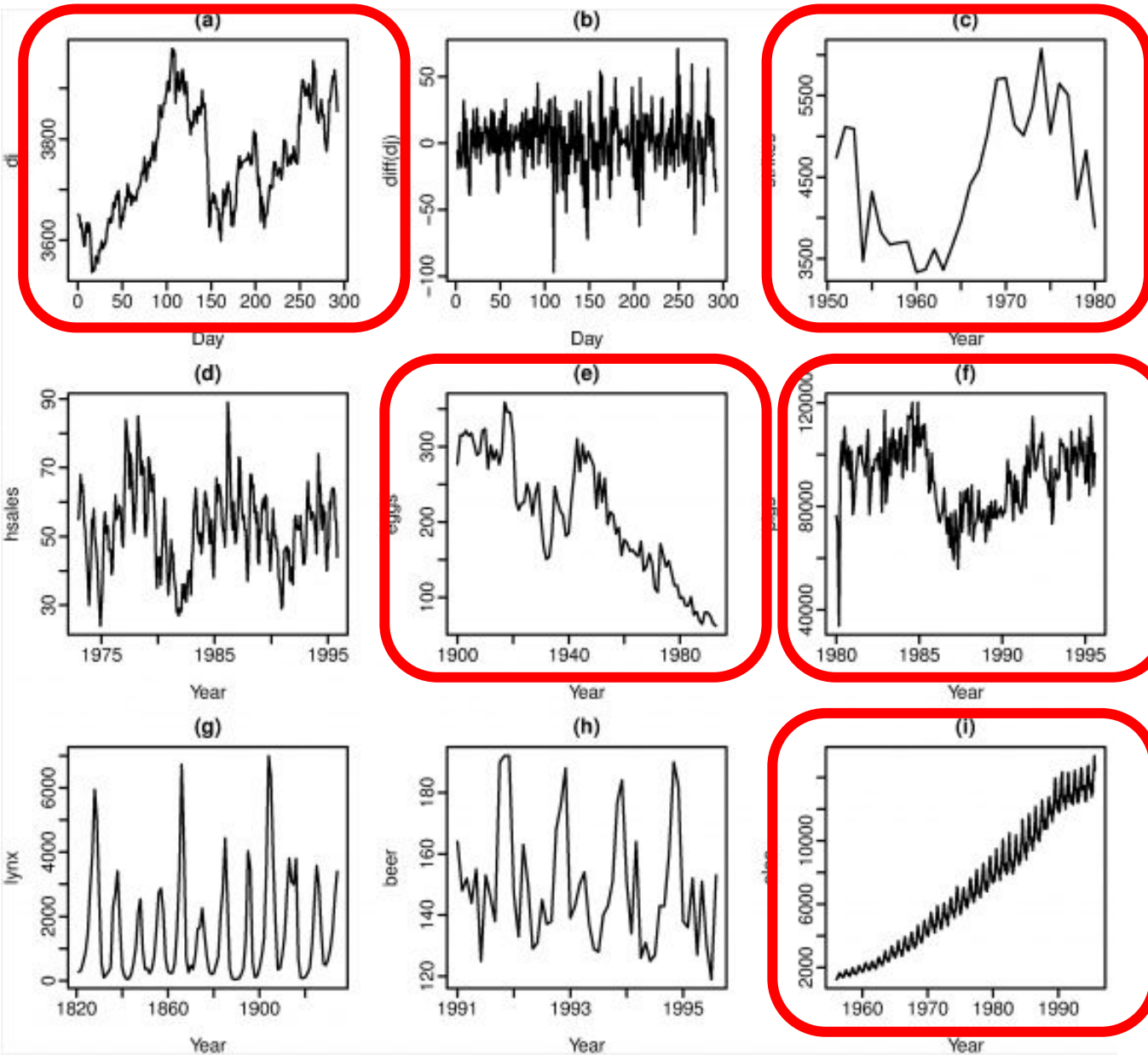
Seasonality?

Trends?

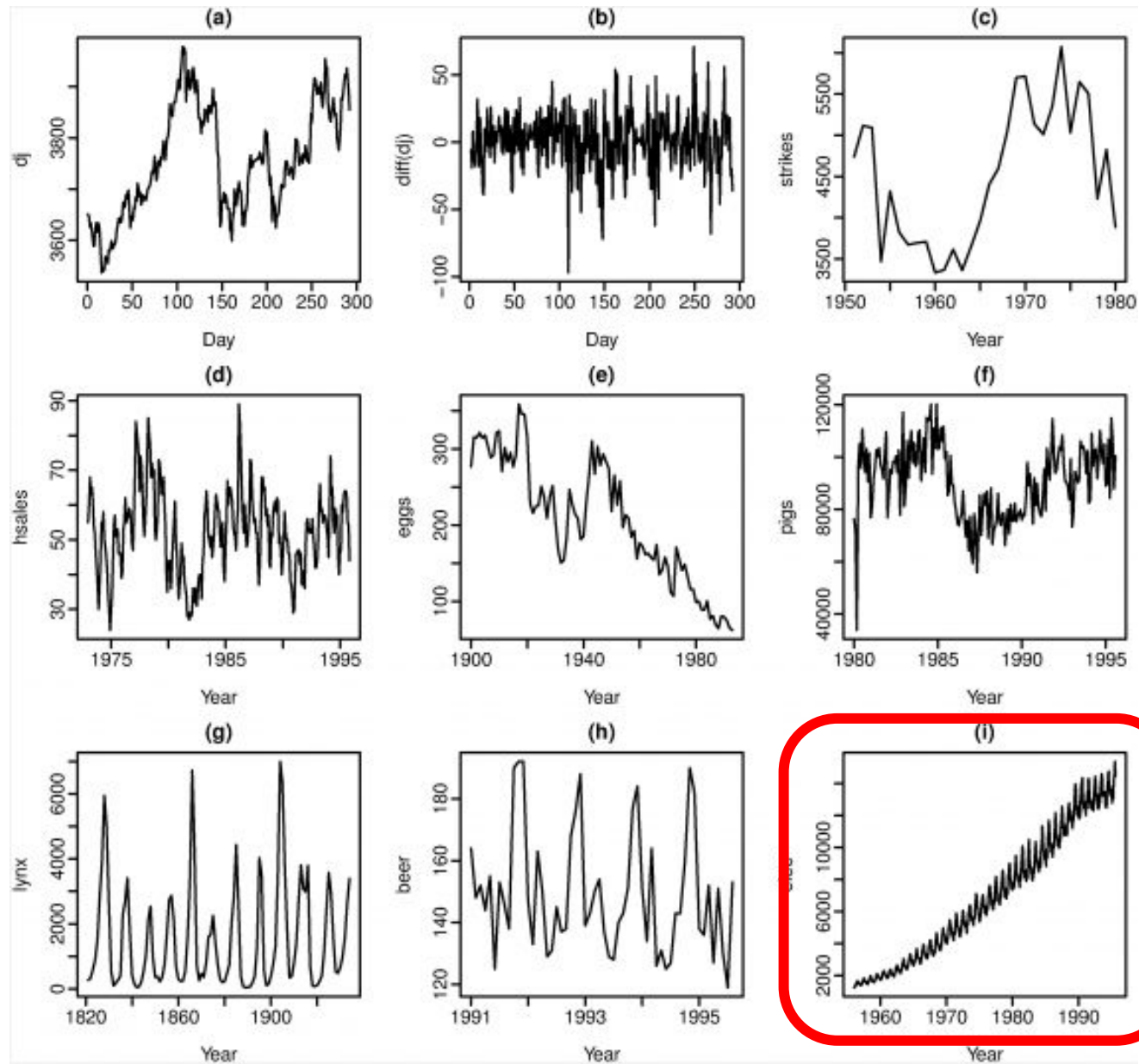
**Increasing
variance?**



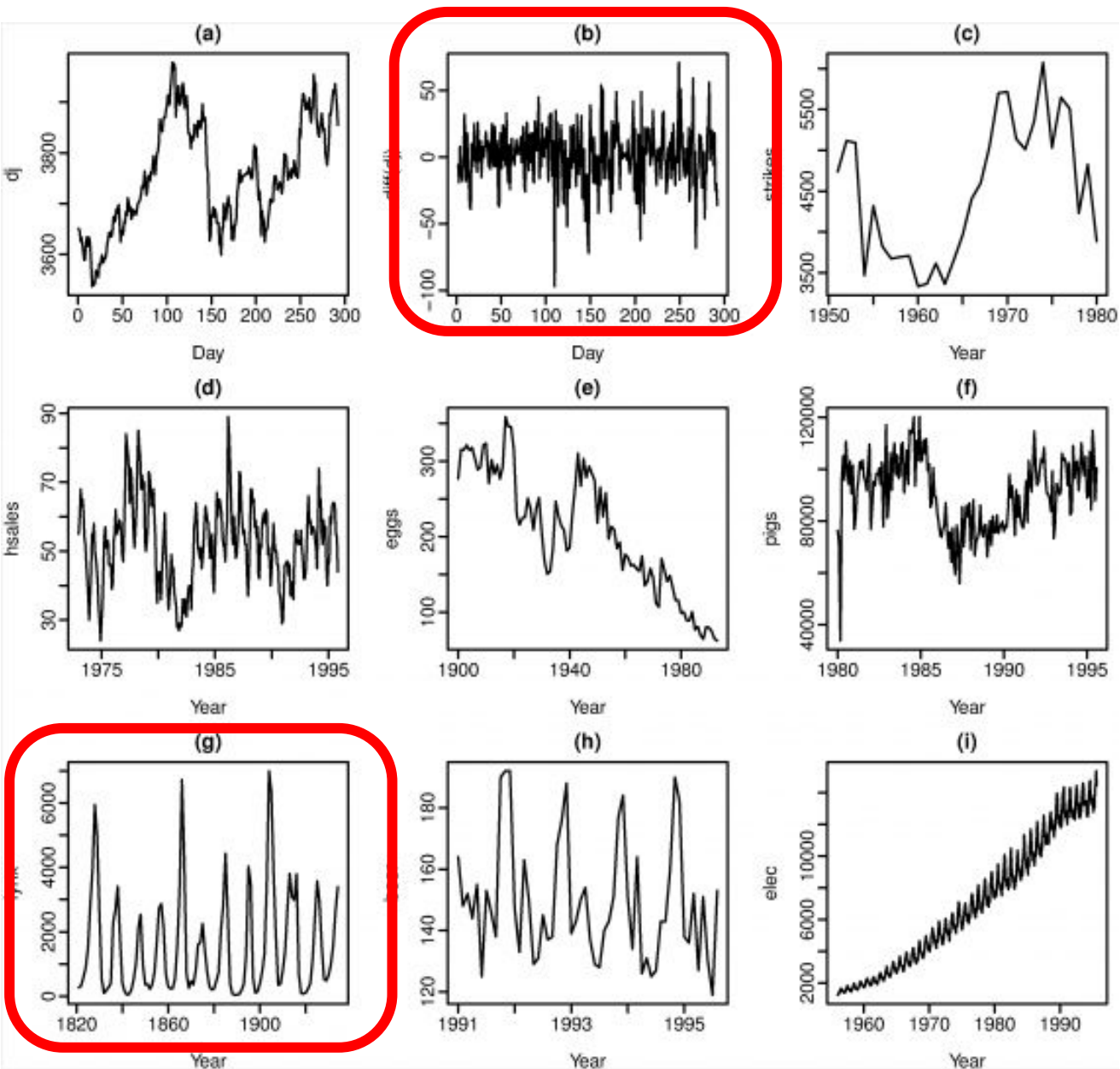
Seasonality!



Trends!



**Increasing
variance!**



(b) Daily change
in Dow Jones
index on 292
consecutive days

(g) strong cycles,
but it is
aperiodic.

ARIMA = AR + I + MA Models

- The combination of two previously developed statistical techniques, the Autoregressive (AR) and Moving Average (MA) models
자기회귀모델과 이동평균모델을 합침
- I (Integrated) corresponds to the differencing step.
차분단계가 더해짐

Autoregressive Models

자기회귀모델

- Current values are a function of prior values
현재값은 과거값의 함수값
- AR(p), p is the number of prior values used in the model. 사용한 전단계값의 갯수
 - AR(1) → $x_t = b_0 + b_1x_{t-1} + \varepsilon_t$
 - AR(2) → $x_t = b_0 + b_1x_{t-1} + b_2x_{t-2} + \varepsilon_t$

Moving Average Models

이동평균모델

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}$$

- MA(q), a linear regression of the current value of the series compared to ε_t terms in the previous period, ε_{t-1} . 현재의 에러와 전기간의 에러에 의해 현재값을 예측
 - q=1, MA(1) is explained by the current error ε_t in the same period and the past error value, ε_{t-1} . 현재의 에러와 전기간의 오차를 사용
 - q=2, MA(2) is explained by the past two error values, ε_{t-1} and ε_{t-2} . 지난 2기간의 에러를 사용

Autoregressive Moving Average Model 자기회귀이동평균모델

- ARMA(p, q) uses two polynomials, AR(p) and MA(q). p 와 q 값을 사용
- Describes a stationary stochastic process.
정상확률과정을 묘사

ARIMA Models

- The ARIMA model has three parameters, p, d, q .
세개의 계수를 가짐
- If you have to difference the time series d times to obtain a stationary series, then you have an ARIMA(p, d, q) model, where d is the order of differencing used. 차분을 하면 정상시계열로 바뀜

Differencing Step 차분단계

- If the data you are working with contains seasonal trends, you “difference” in order to make the data stationary. 계절성의 갖는 데이터를 차분하면 정상시계열이 됨
- This differencing step generalizes the ARMA model into an ARIMA model. **ARMA와 ARIMA**모델의 차이점은 차분

Differencing 차분

- Computes the differences between consecutive observations to make a time series stationary
정상시계열을 만들기 위해 연속된 관측치 사이의 차이를 계산

```
diff1 =  
skirts.diff(periods=1).iloc[1:]  
diff1.plot()
```

Transformation vs Differencing

변환과 차분의 다른점

Transformations such as logarithms can help to stabilize the variance of a time series.

변환은 분산을 안정시킴

Differencing can help stabilize the mean of a time series by removing changes in the level of a time series, and so eliminating trend and seasonality

차분은 평균을 안정시킴

Skirt Diameter Example

- The time series of the annual diameter of women's skirts at the hem, from 1866 to 1911. The data is available in the file.

Forecast the diameter of women's skirts at the hem for 1912.

다음 파일은 1866년부터 1911년까지 여성치마단의 연간직경을 포함합니다. 1912년 여성치마단의 직경을 예측하시요.

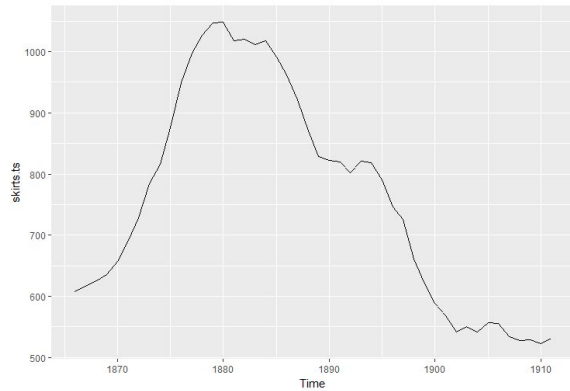
hem, <http://robjhyndman.com/tsdldata/roberts/skirts.dat>

Exercise #10

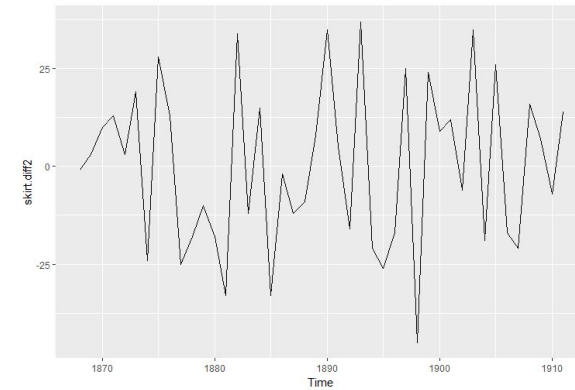
Skirts Diameter Example

- Plot the time series 시계열 차트
- Difference the time series once ($\text{differences}=1$) and plot the differenced series 차분을 한번하고 그래프
- Difference the time series twice ($\text{differences}=2$) and plot the differenced series 차분을 두번하고 그래프

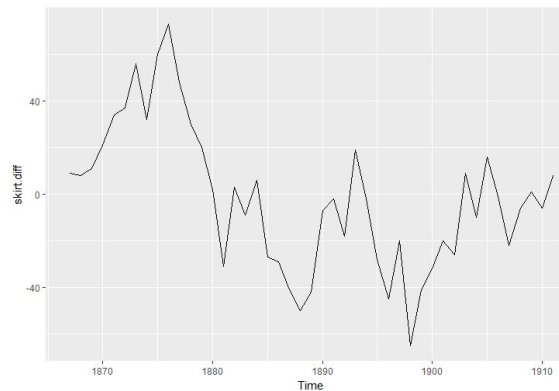
Differencing Steps



```
skirt_diff2 =  
skirts.diff(  
    periods=1)
```



```
skirt_diff1 =  
skirts.diff(  
    periods=1)
```

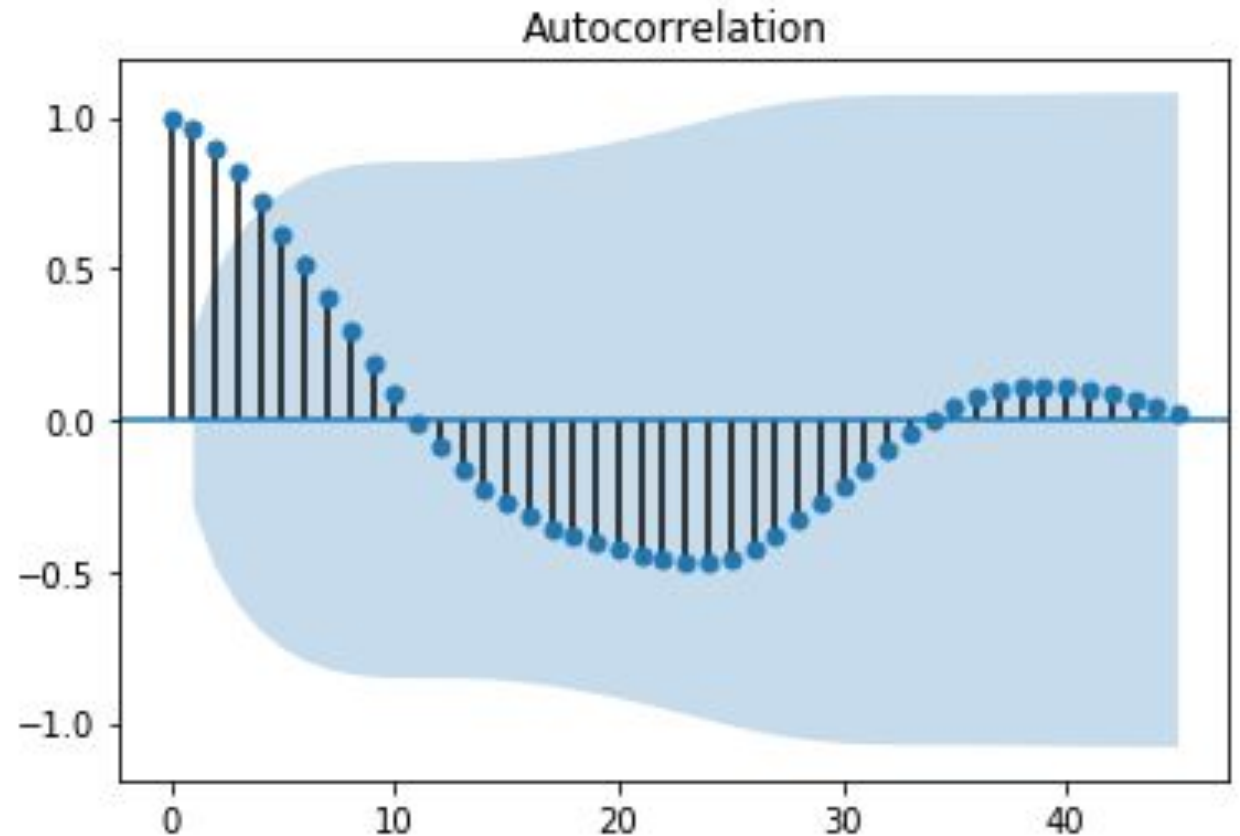


d=2

ACF (Autocorrelation) 자기상관

- Useful for identifying non-stationary time series
비정상 시계열을 식별할 때 사용됨
- Can estimate the q value q값을 측정

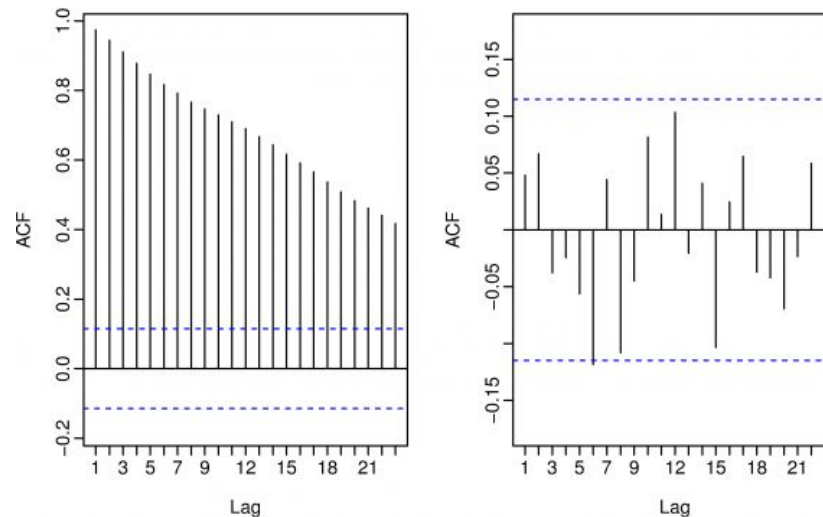
```
from  
statsmodels.graphics.t  
saplots import  
plot_acf, plot_pacf  
plot_acf(skirts)
```



ACF Plot

Only one autocorrelation lying
just outside the 95% limits
자기상관계수중 1개만 95%
한계밖에 존재

The ACF of
non-stationary data
decreases slowly.
정상시계열이 아닌
데이터는 값이
천천히 떨어짐

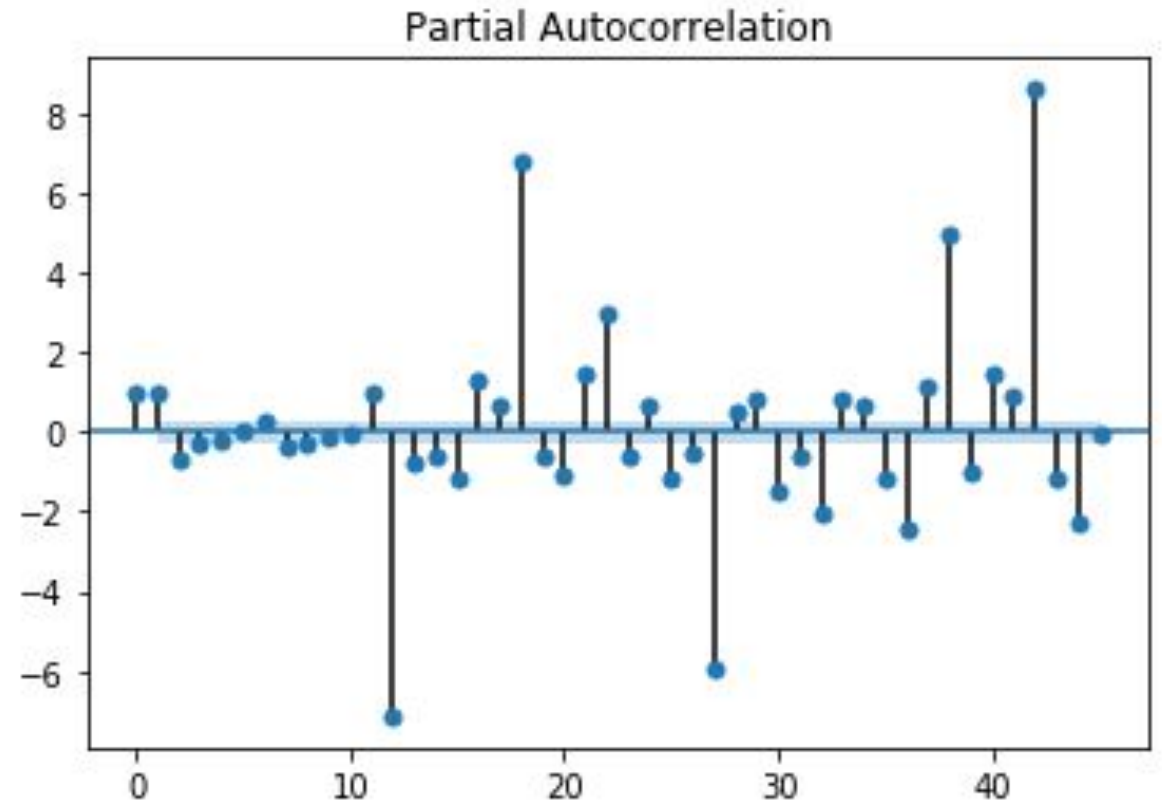


For a stationary time
series, the ACF will
drop to zero relatively
quickly
정상시계열은
상대적으로 빨리
0으로 떨어짐

PACF (Partial Autocorrelation)

- Can estimate the p value
p값을 예측

```
from  
statsmodels.graphics.tsaplots import  
plot_acf,  
plot_pacf  
plot_pacf(skirts)
```

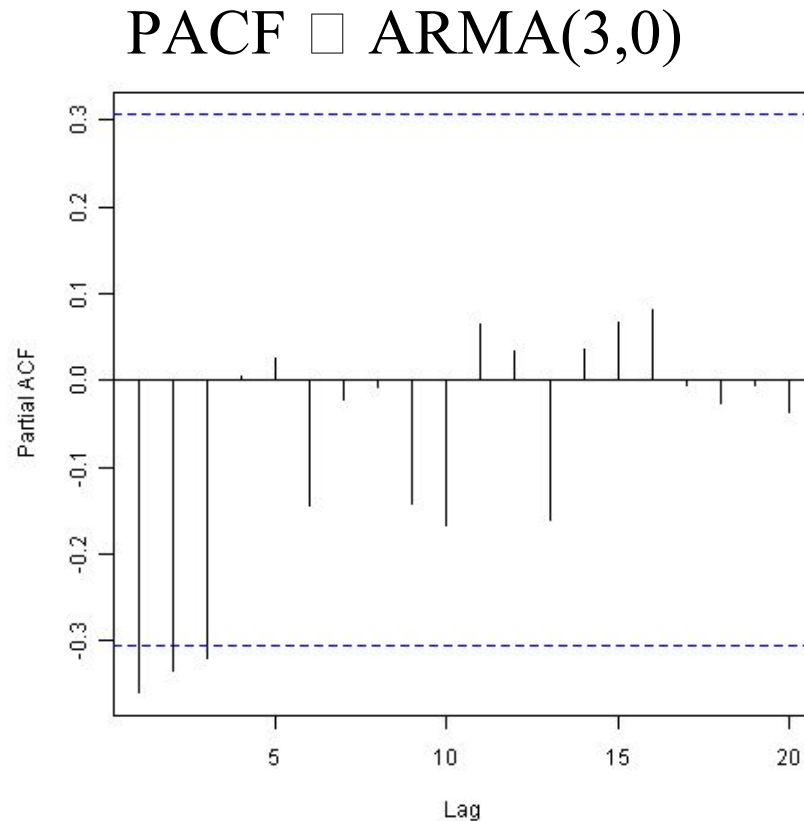
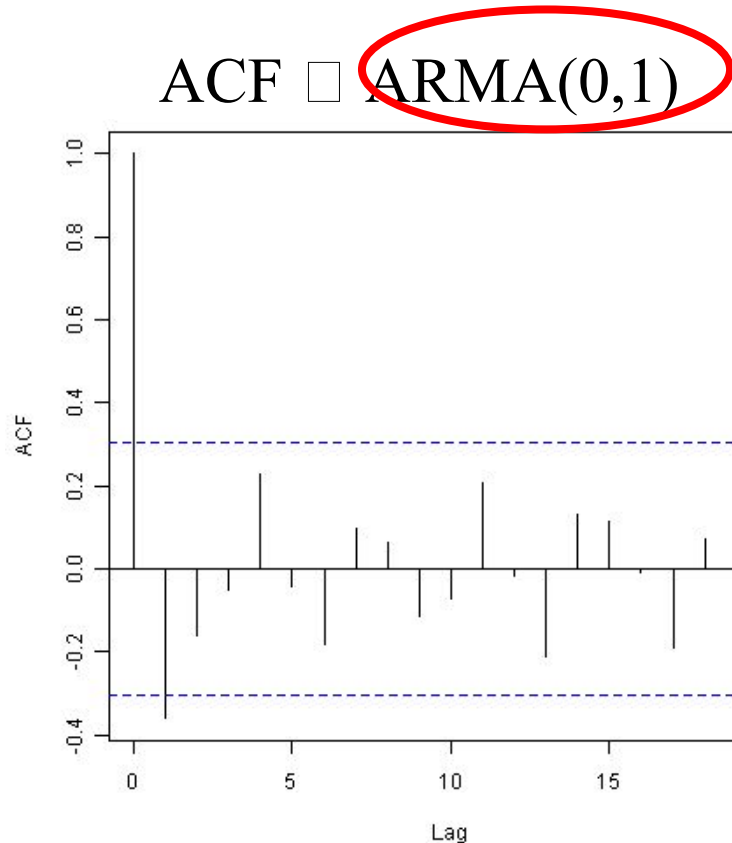


King Example

- The file below contains data on the age of death of successive kings of England, starting with William the Conqueror. Forecast the age of death of next king. 아래 파일은 영국왕들의 죽은 나이를 포함합니다. 다음 왕의 죽을 나이를 예측하시요

<http://robjhyndman.com/tsdldata/misc/kings.dat>

Estimating the Parameters 계수의 추정



The model with the fewest parameters is best!! 계수의 갯수가 적은 모델이 가장 좋음

Building ARIMA Model

```
from statsmodels.tsa.arima_model import ARIMA
model = ARIMA(skirts[0], order=(0,1,1))
model_fit = model.fit(dis=0)
print(model_fit.summary())
# Actual vs Fitted
model_fit.plot_predict(dynamic=False)
plt.show()
```

auto_arima

- Used to find the appropriate ARIMA model. e.g., ARIMA(0,1,1) 가장 적절한 모델을 찾아줌

```
from pyramid.arima import auto_arima
stepwise_model = auto_arima(data, start_p=1, start_q=1,
max_p=3, max_q=3, m=12, start_P=0, seasonal=True, d=1, D=1,
trace=True, error_action='ignore', suppress_warnings=True,
stepwise=True)
print(stepwise_model.aic())
Stepwise_mode.fit(train)
stepwise_model.predict(n_periods=37)
```

Checking Model 모델검증

- Testing to see if the model conforms to a stationary univariate time series. 정상 일변량 시계열 데이터인지를 테스트
- Confirming the residuals are independent of each other and exhibit constant mean and variance over time 잔차끼리 서로 독립적인가를 테스트
- Performing a Ljung-Box test or plotting the autocorrelation and partial autocorrelation of the residuals. 융박스 테스트를 실행하거나 잔차의 부분자기상관과 자기상관을 그림

```
residuals = pd.DataFrame(model_fit.resid)  
plot_acf(residuals)
```

Ljung-Box Statistics

```
statsmodels.stats.diagnostic.acorr_ljungbox(residuals,  
lags=20, boxpierce=False)
```

Box-Ljung test data:

```
kingstimeseriesforecasts$residuals
```

```
X-squared = 13.5844, df = 20, p-value = 0.851
```

- Very little evidence for non-zero autocorrelations in the forecast errors at lags 1-20. That is no autocorrelations!! **p값이 0.05보다 크므로 자기상관관계가 없음**

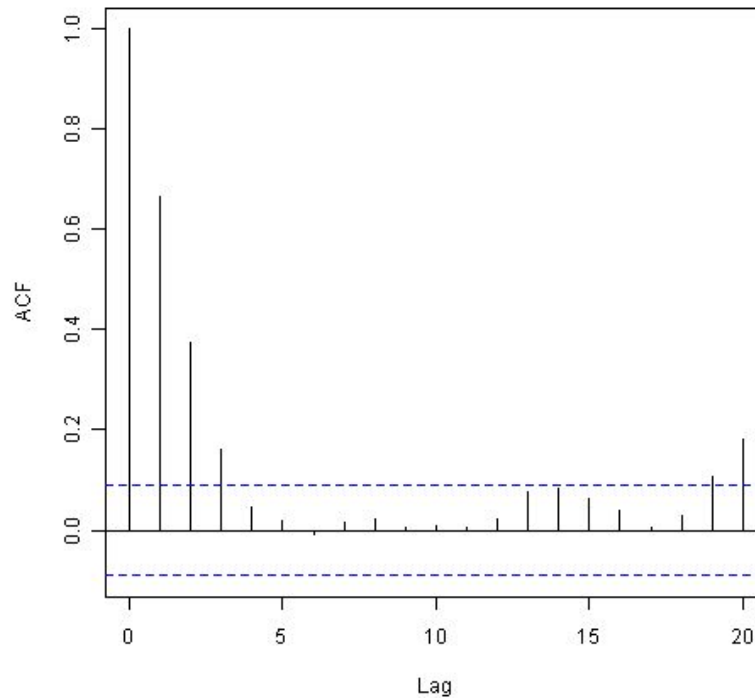
Volcano Example

- The file below contains data on the volcanic dust veil index in the northern hemisphere, from 1500-1969. This is a measure of the impact of volcanic eruptions' release of dust and aerosols into the environment. Forecast the volcanic dust veil index for 1970. 다음 파일은 1500년에서 1969년 북반구의 화산분진베일지수를 포함합니다. 이 지수는 화산폭발로 인한 먼지와 에어로졸의 영향을 측정합니다. 1970년의 화산분진베일지수를 예측하시요.

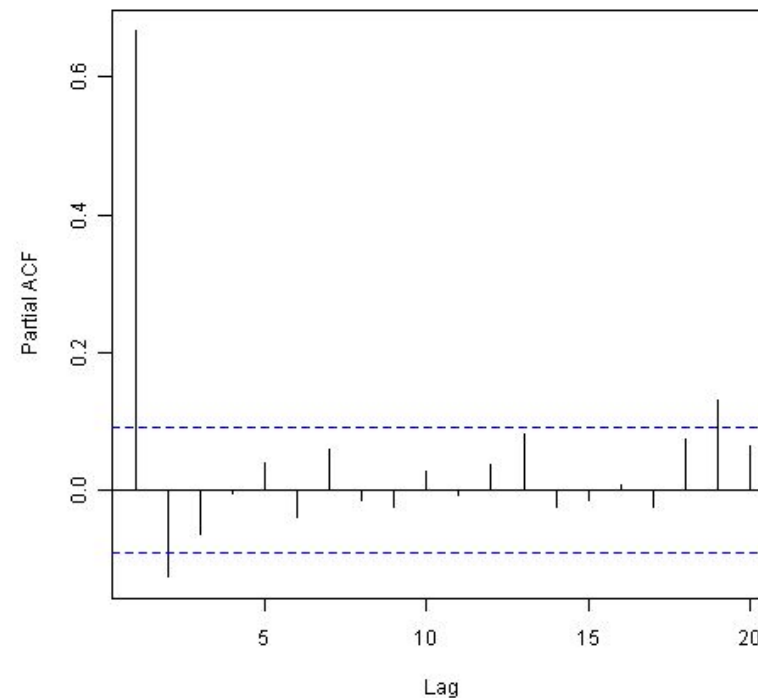
<http://robjhyndman.com/tsdldata/annual/dvi.dat>

Estimating the Parameters 계수추정

ACF □ ARMA(0,3)



PACF □ ARMA(2,0)



The model with the fewest parameters is best!!

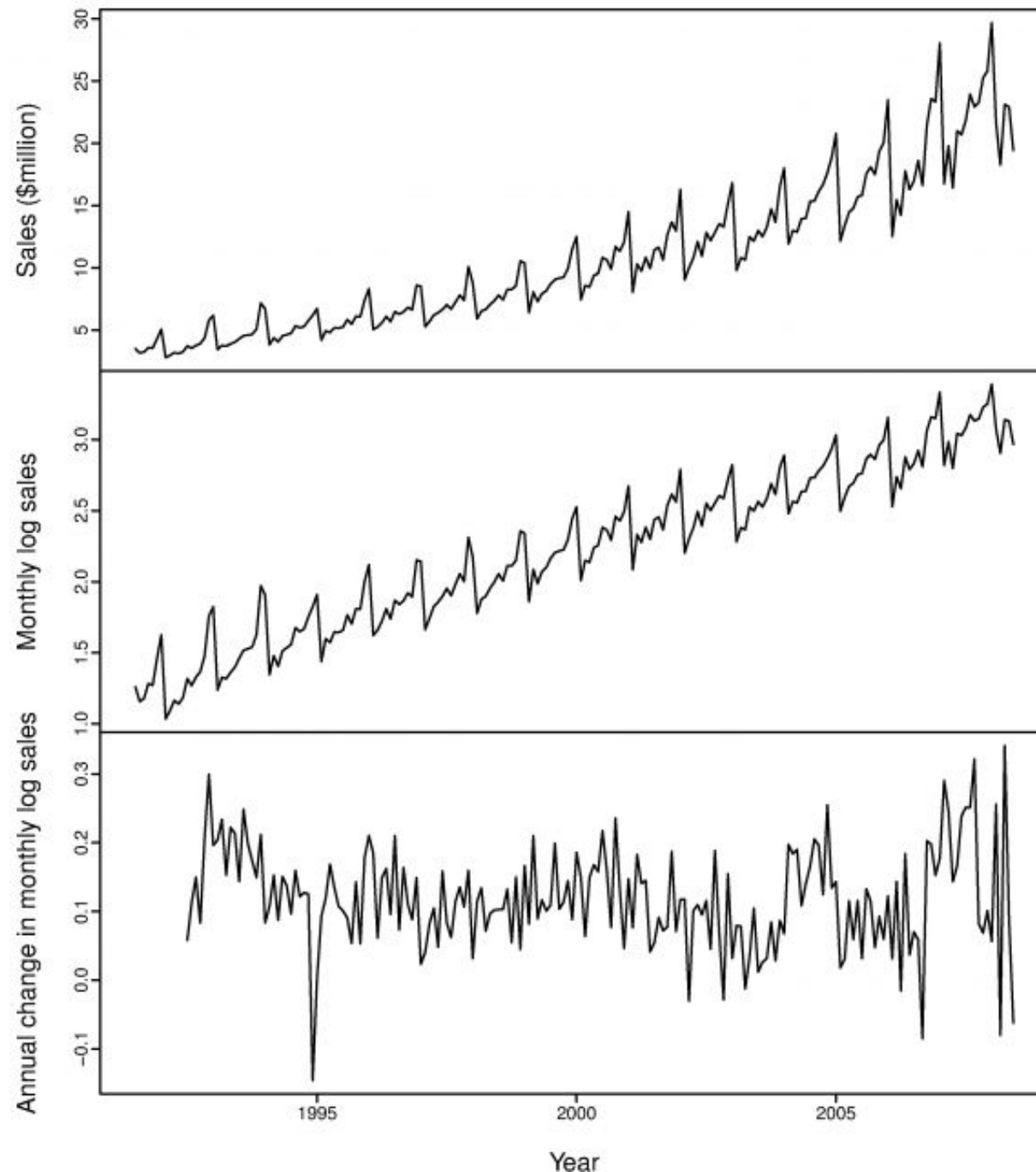
계수의 수가 적은게 가장 좋음

Tractor Sales Example

- You could find the data shared by PowerHorse's MIS team at the following link. Forecast the tractor sales for next 36 months. 다음 링크에서 파워홀스 MIS팀에서 공유한 데이터를 찾아서 다음 36개월의 트랙터세일을 예측하시요

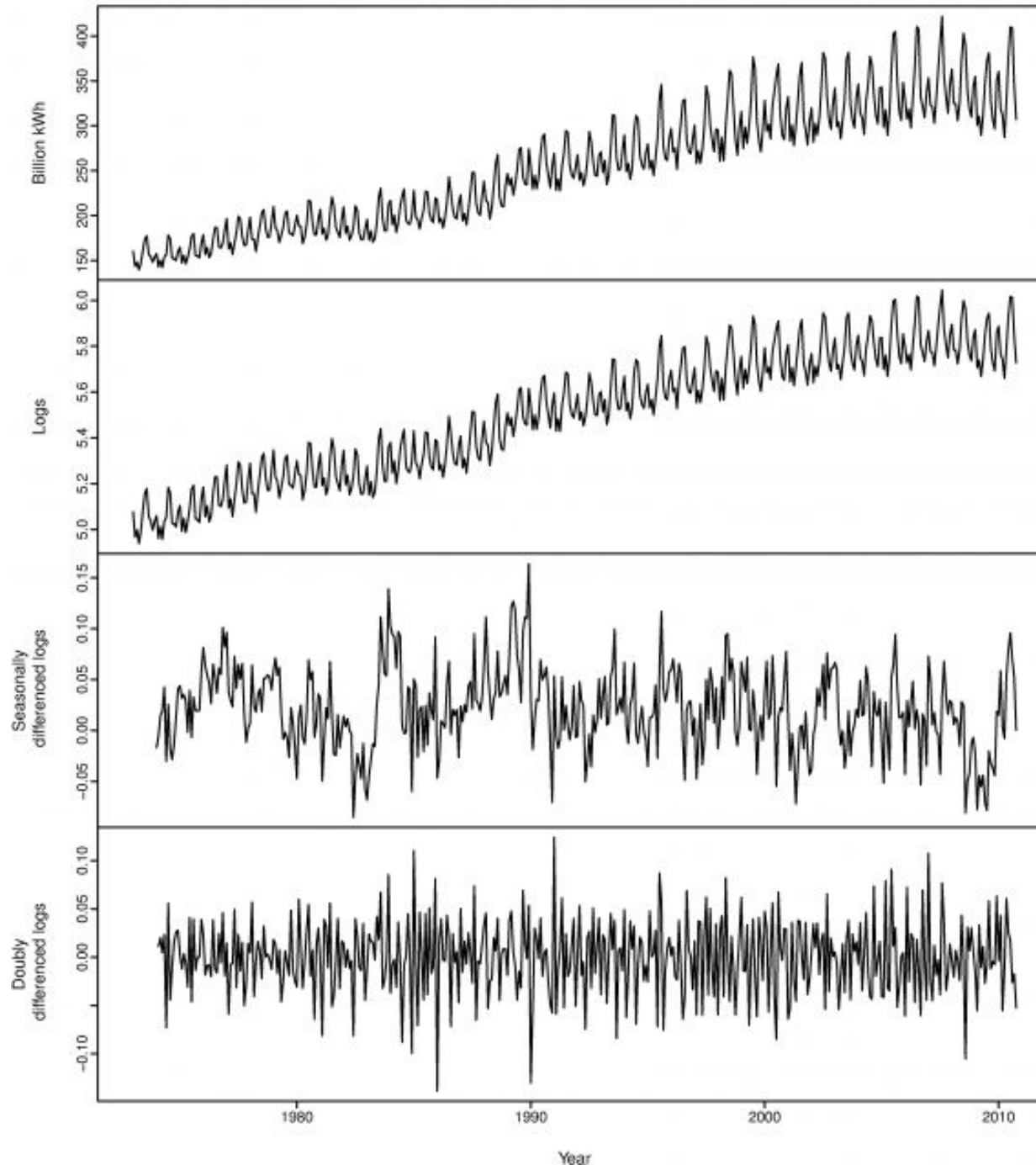
<http://ucanalytics.com/blogs/wp-content/uploads/2015/06/Tractor-Sales.csv>

Antidiabetic drug sales



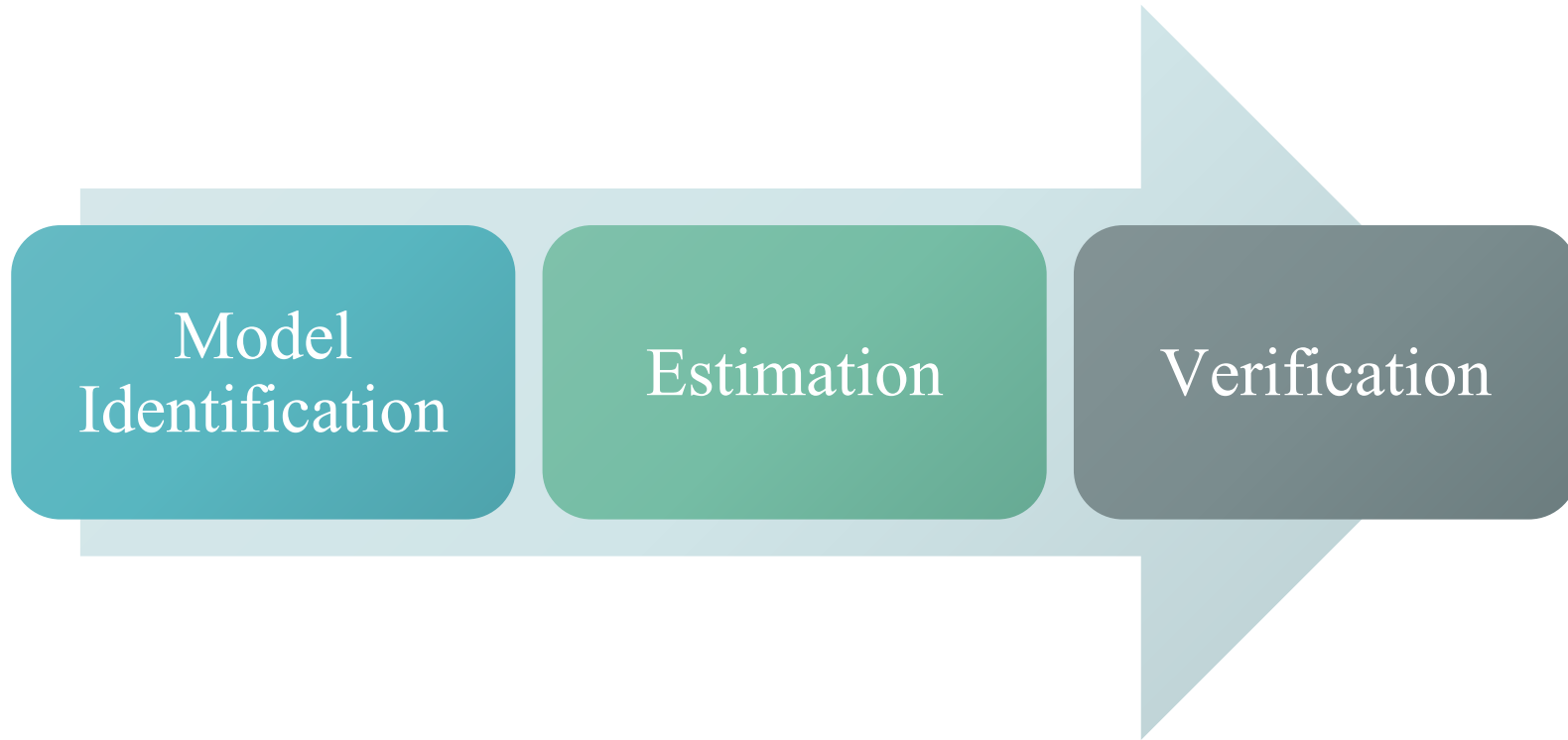
Log
transformation
then differencing

Monthly US net electricity generation



Log transformation
then differencing
twice

Box-Jenkins Method



Unit Root Tests

- Determine more objectively if differencing is required
차분이 필요한지 검증하는 테스트
- Statistical hypothesis tests of stationarity 정상성에 대한 통계가설 테스트
- Augmented Dickey-Fuller (ADF) test,
Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, etc.

ADF Test

```
from statsmodels.tsa.stattools import adfuller
result = adfuller(residuals[0])
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))
```

- The null-hypothesis for an ADF test is that the data are non-stationary. So large p-values are indicative of non-stationarity, and small p-values suggest stationarity. 귀무가설은 데이터가 비정상 데이터이다 임. p값이 작으면 비정상, p값이 크면 정상시계열
- Using the usual 5% threshold, differencing is required if the p-value is greater than 0.05. p값이 0.05 보다 크면 차분이 필요함

Summary of Functions ARIMA

명령어 정리

<code>lag(ts, k)</code>	lagged version of time series, shifted back k observations
<code>diff(ts, differences=d)</code>	difference the time series d times
<code>ndiffs(ts)</code>	Number of differences required to achieve stationarity (from the <u>forecast</u> package)
<code>acf(ts)</code>	autocorrelation function
<code>pacf(ts)</code>	partial autocorrelation function
<code>adf.test(ts)</code>	Augmented Dickey-Fuller test. Rejecting the null hypothesis suggests that a time series is stationary (from the <u>tseries</u> package)
<code>Box.test(x, type="Ljung-Bo x")</code>	Pormanteau test that observations in vector or time series x are independent

Exercise #10

Kings Death Example

- Difference the time series once (differences=1) and plot the differenced series 차분을 한번하고 그래프
- Plot the correlogram and partial correlogram of the stationary time series (lag.max=20) and show the actual values of the autocorrelations and partial autocorrelations 자기상관관계, 부분자기상관관계 그래프
- Auto.arima and forecast next time periods using arima function 오토아리마로 예측
- Correlation between successive forecast errors using acf, pacf, and box.test 예측오차의 상관관계

Exercise #10

Volcano Example

- Plot the correlogram and partial correlogram of the stationary time series (lag.max=20) and show the actual values of the autocorrelations and partial autocorrelations 자기상관관계, 부분 자기상관관계 그래프
- Auto.arima and arima to forecast the next time periods 오토아리마로 예측
- Acf and box.test to check the correlation between forecast errors 예측오차의 상관관계
- Plot residuals and draw a histogram 잔차 히스토그램

Exercise #10

Tractor Example

- Log transformation and Difference the time series once (differences=1) and plot the differenced series 로그변환, 차분 한번하고 그래프
- Plot the correlogram and partial correlogram of the stationary time series (lag.max=20) and show the actual values of the autocorrelations and partial autocorrelations 자기상관관계, 부분 자기상관관계 그래프
- Auto.arima and arima to forecast the next time periods 아리마로 예측
- Acf and box.test to check the correlation between forecast errors 오차의 상관관계
- Plot residuals and draw a histogram 잔차 히스토그램