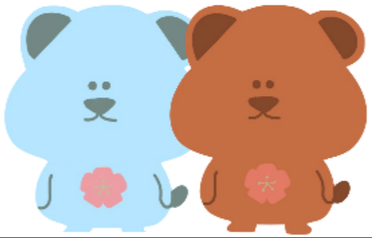
비트 트레이드 경진대회



EWHA EURON

#1-1. 대회 & 데이터 소개

✔대회 Task : 하루동안 분 단위 가격 정보를 보고 암호화폐 가격을 예측 → 암호화폐 트레이딩을 위한 수익율이 높은 모델 만들기

✔카테고리 : 시계열 데이터 분석

✔test data는 train data와 동일한 구성을 갖는 sample

✔매도량(=buy_quantity) & 매도 시각 (sell_time) 예측을 해야함

train_x에는 입력 23시간 동안의 분 단위 데이터가 (23 x 60 = 1380개)

train_y에는 출력 2시간 동안의 분 단위 데이터가 (2 x 60 = 120개)

	sample_id	buy_quantity	sell_time
0	7661	0	47
1	7662	1	62
2	7663	0	11
3	7664	0	48
4	7665	1	46
5	7666	1	70

Train : 총 7929개
Test : 총 760개

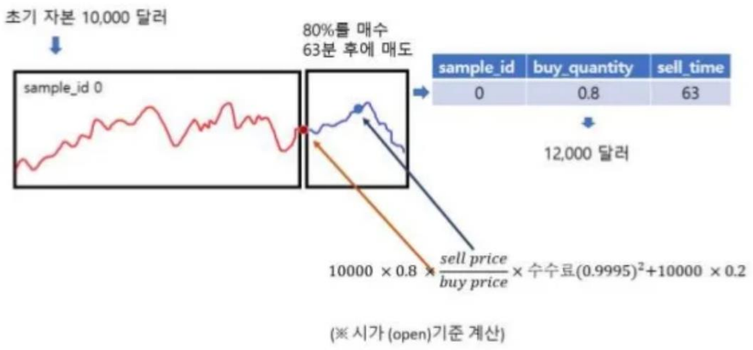
1380개의
간격으로
시간을 쪼갬

임의의 시점 부터 2021년 2월 28일까지 10가지
종류의 암호화폐의 분단위 가격정보를 가공한 데이터

	sample_id	time	coin_index	open	high	low	close	volume	quote_av	trades	tb_base_av	tb_quote_av
0	0	0	0	0.993147	0.993546	0.992857	0.992966	1379.478027	3778.584961	11.240029	329.655548	903.091614
1	0	1	0	0.993256	0.993546	0.992712	0.992712	3438.807373	9419.426758	11.602611	1363.999268	3737.512695
2	0	2	0	0.992748	0.994815	0.992458	0.994815	3714.949463	10173.972656	19.579407	1222.802856	3350.688721

EWHA EURON

#1-1. 대회 & 데이터 소개



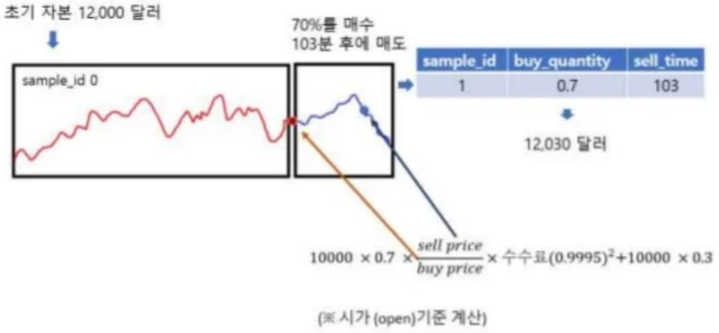
[Step1]

초기 자본으로 10000달러를 갖고 있고

63분에(sell_time) 80% (buy_quantity)

매도를 하기로 결정한 상황

▯10000달러 -> 12000달러



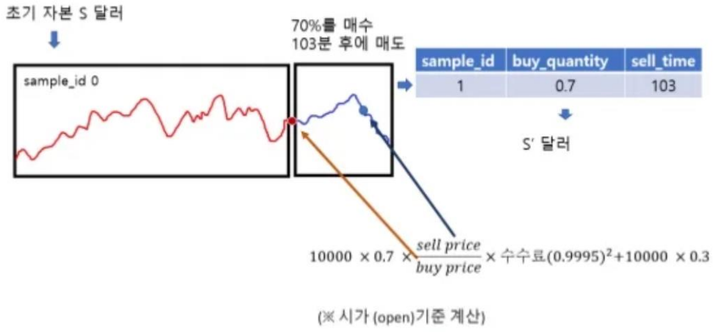
[Step2]

앞선 거래 이후 초기 자본

12000달러에서 시작

70%를 매수 하고 103분 후에

매도하는 상황



[Step N]

이와 같은 과정을 계속 반복하여

Public과 private leaderboard의

점수를 지속적으로 갱신

EWHA EURON

#1-2. 보조 지표

$$VWAP = \frac{\Sigma (Volume * Price)}{\Sigma Volume}$$

대표 가격 = (고가 + 저가 + 종가) / 3

$$RSI = \frac{AU}{(AU + AD)}$$

AU - 전일 대비 상승분의 평균
AD - 전일 대비 하락분의 평균

✳거래량에 가중치가 부여된 특정 기간 동안의 자산의 평균 가격

❗하루 동안의 일일 지표로 유리하게 사용

✳상승세와 하락세를 판단하는 지표

✳open > vwap : 상승세

✳open < vwap : 하락세

✳가격의 상승압력과 하락 압력 간의 상대적인 강도

✳RSI < 30 : 초과 매도

✳RSI > 70 : 초과 매수

EWHA EURON

#1-3. 데이터 전처리

1. train_x, train_y를 sample_id를 기준으로 합쳐서 연속되는 시계열 변수를 분석

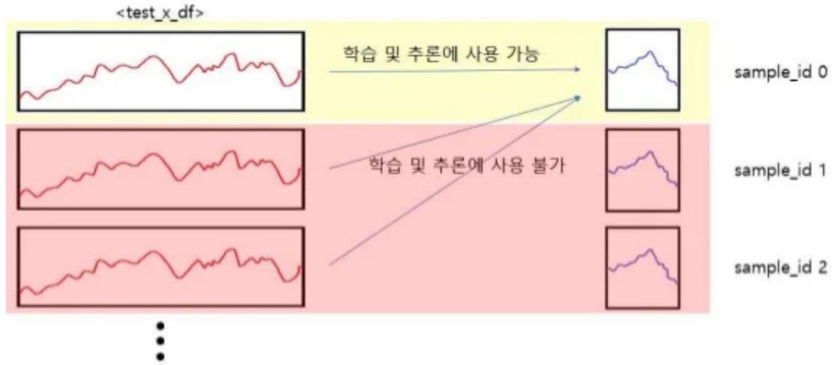
	sample_id	time	coin_index	open	high	low	close	volume	quote_av	trades	tb_base_av	tb_quote_av	is_x
0	0	0	0	0.993147	0.993546	0.992857	0.992966	1379.478027	3778.584961	11.240029	329.655548	903.091614	1
1	0	1	0	0.993256	0.993546	0.992712	0.992712	3438.807373	9419.426758	11.602611	1363.999268	3737.512695	1

2. **diff = open - vwap**를 학습에 사용

- 매도 수익이 open(=시가)를 통해 이뤄지기 때문에 open – vwap를 사용
- 안전한 지점을 찾을 수 있음
- 2개의 변수를 한꺼번에 사용할 수 있음

3. 안정적인 투자를 위한 **RSI의 응용**

- RSI > 65 : 초과 매수 국면
- 위의 초과 매수 국면 시점 이후 50분 동안은 투자를 하지 않음

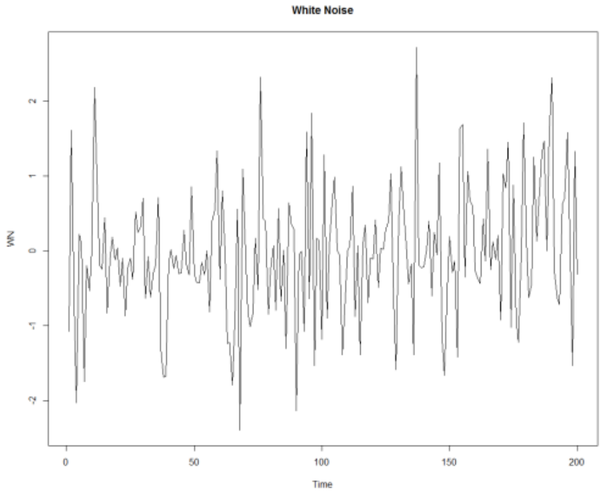


Test_y_df를 추론하는 과정에서 동일한 sample_id의 test_x는 학습과 추론에 사용이 가능하지만, 서로 다른 sample_id는 절대 사용해서는 안된다.

#2-1. 시계열 데이터 분석의 기본 모델

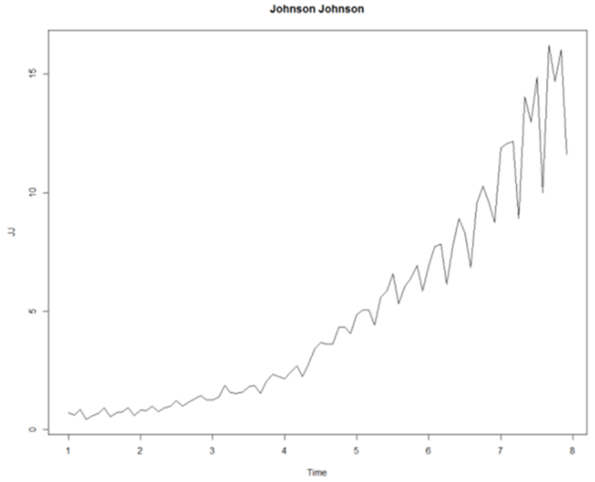
Stationary Process (정상 프로세스)

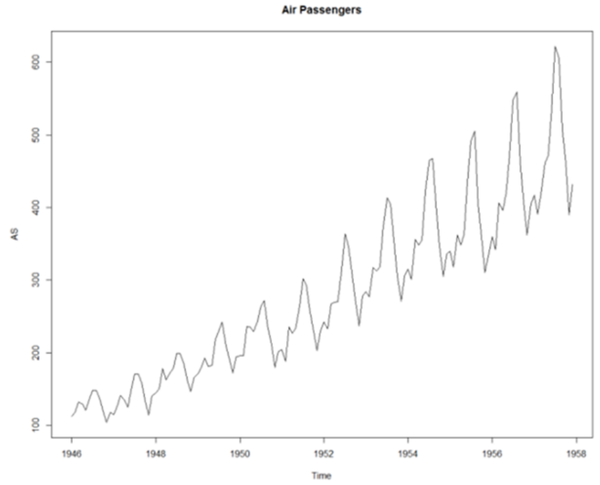
사간에 관계 없이 평균과 분산이 일정한 시계열 데이터



Non-Stationary Process (비정상 프로세스)

사간에 관계 없이 평균과 분산이 일정하지 않은 시계열 데이터





가로 축을 현재 데이터와의 시점의 차이로, y축을 ACF(AutoCorrelation Function)으로 시각화 할 때 특정 패턴이 없으면 정상 프로세스

❓ AC (Autocorrelation)

❗ 원래 correlation은 두 변수간의 관계를 -1 ~ 1 사이의 값으로 나타낸 것

❗ Auto + Correlation = time shifted된 자기 자신의 데이터와의 관계를 의미

#2-2. AR 모형 vs MA 모형 vs ARMA 모형

[Autoregressive Models]

$$AR(1) : X_t = \phi X_{t-1} + \epsilon_t$$

지기 회귀

✔이전 관측값의 오차항이 이후 관측값에 영향을 주는 모형

✔p를 hyperparameter로 가지며, 이는 위의 식에서는 1이다. (Time Lag를 의미)

[Moving Average Models]

$$MA(1) : X_t = \epsilon_t - \beta_1 \epsilon_{t-1}$$

이동 평균

✔관측값이 이전의 연속적인 오차항의 영향을 받는 모형

✔epsilon은 t시점에서의 오차항

✔ beta는 이동 평균 계수

[Autoregressive & Moving Average Models]

✔자기 자신을 종속 변수로 설정

✔이전시점의 시계열을 독립 변수로 설정

✔AR + MA 모델

EWHA EURON

#2-3. ARIMA 모형

👍Pros

✔다른 설명 변수의 도입 없이 변수의 과거치와 교란항만을 사용해서 시계열에 적합한 모형 설정 가능

✔시계열의 특성을 가장 적은 모수의 수(=3)로 표현이 가능

AR **I** **MA**

AR : 자기 회귀

I : Integrated(=누적)

MA : 이동 평균

✔기존 AR, MA, ARMA 모델은 데이터가 항상 정상이어야 했음

✔따라서 이를 정상으로 바꾸어 주기 위해 **차분(differencing)**을 사용

✔ARMA 모델에 differencing을 d번 수행 → 변수 d가 추가됨

$$AR(p) = ARIMA((p,0,0)$$

$$MA(q) = ARIMA(0,0,q)$$

$$ARMA(p,q) = ARIMA(p,0,q)$$

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \beta_1 \epsilon_{t-1} - \ldots - \beta_q \epsilon_{t-q}$$

EWHA EURON

#2-3. ARIMA의 differenciating

X
2
7
10
5
8

X	Y
2	5
7	3
10	-5
5	3
8	-

1차 차분: $Y_t = X_t - X_{t-1} = \nabla X_t$

X
2
7
10
5
8

X	Y
2	8
7	-2
10	-2
5	-
8	-

2차 차분: $Y_t^{(2)} = X_t - X_{t-2} = \nabla^{(2)} X_t$

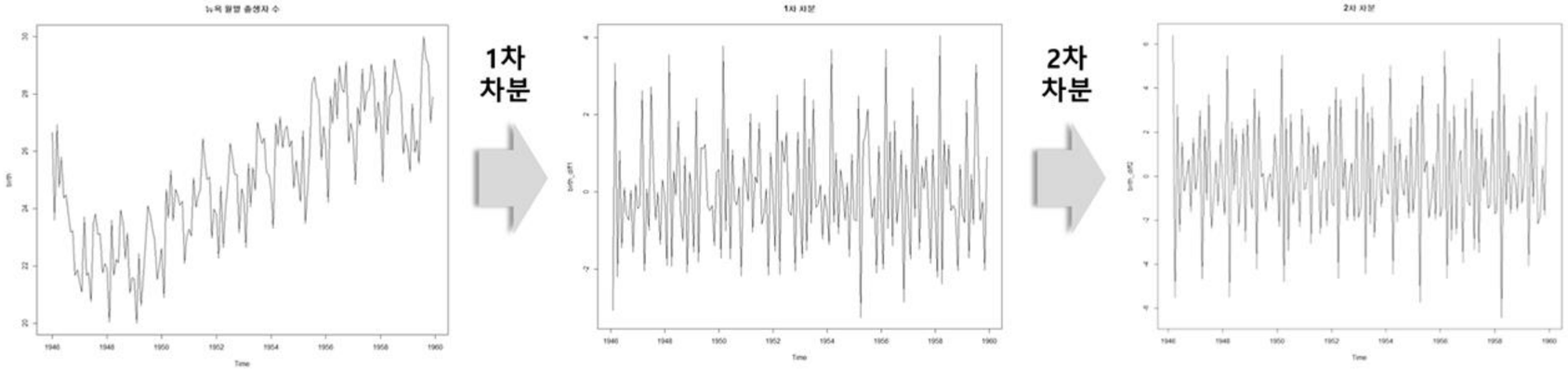
차분

:현 시점 데이터에서 d(ARIMA의 차분 횟수 변수) 시점 이전의 데이터를 뺀 것

✔시계열 곡선이 특정 trend를 갖는다면 1차 차분

✔시간에 따라 변동이 있는 trend를 갖는다면 2차 차분

ARIMA 모델에서 자체적으로 비정상 시계열을 정상 시계열로 변형하기 위해 ARMA에 d회의 차분 수행



#2-4. 평가 지표 : ARIMA의 모수 설정을 위해

[ACF 자기 상관 함수]

✳k 시간 단위로 구분된 시계열의 관측치 사이의 상관계수 함수

▯k가 커질 수록 ACF는 0에 가까워짐

✳아래 식은 y_t와 y_t-k 사이의 자기 상관을 구하는 식이다.

$$ACF(k) = \frac{\sum_{t=1}^{N-k}(y_t - y)(y_{t+k} - y)}{\sum_{t=1}^{N}(y_t - y)^2}$$

[PACF Partial ACF]

✳Partial Correlation

: 두 확률 변수 X와 Y에 의해 다른 모든 변수들에 나타난 상관 관계를 설명하고 난 이후에도 남아있는 상관 관계

✳PACF

: ▯시계열 관측치 간의 상관 관계 함수

: ▯즉, 두 시점 사이의 다른 관측치의 영향력은 완전히 배재한 오로지 두 시점 사이의 순수한 상관 관계

$$PACF(k) = Corr(e_t, e_{t-k})$$

ARIMA(p, d, q)

✳ p+q < 2 이고 p * q = 0인 값을 주로 설정

p : AR 모형의 Lag

d : Difference의 횟수

q : MA 모형의 Lag

#3. 우승 포인트 및 코드 분석

ARIMA 모델 정의 및 적용

1) 수렴하지 않는 경우 q, d, q = 1, 1, 0, 즉 AR모형만 사용

2) ARIMA 시계열 예측 FORECAST 생성

↓

매도 시점 (sell_time) 찾기

1) 앞서 예측한 FORECAST의 최댓값의 index를 매도 시점으로 지정

2) FORECAST의 최댓값 저장

3) VWAP의 마지막 값을 저장

↓

투자 전략 (buy_quantity) 찾기

1) 최댓값이 0보다 크면 가격 > vwap이므로 전체 투자

2) vwap 마자막 값이 1보다 크면 하향세 -> 투자 X

3) rsi > 65 -> 초과 매수 -> 투자 X

#04 Prophet

Abstract

- 시계열 예측은 많은 곳에서 필요로 하지만, 깊게 분석할 수 있는 전문가는 많지 않다.
- 완전히 자동화된 예측 기법들은 예측에 도움이 될 다양한 가정과 경험을 모형 안에 포함시키기 어렵다.

Prophet

- 페이스북에서 시계열 예측 모형 Prophet 제안.
- 시계열 분석/ 예측에 대한 전문적인 지식을 가지고 있지 않더라도
  몇 가지 직관적인 파라미터를 가지고 조작할 수 있도록 한다.

#04 Prophet – Model Components

Model Components

3개의 메인 컴포넌트로 구성된 Decomposable Time Series Model

Growth

Seasonality

Holidays

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

#04 Prophet – Model Components

Model Components

- ARIMA 등 시계열 모형과는 달리 시간에 종속적인 구조를 가지지 않는다.
- 대신 Curve Fitting으로 문제를 해결

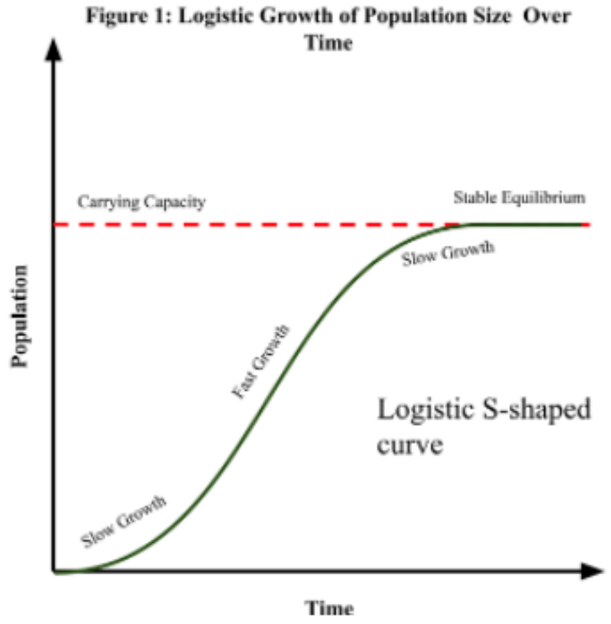
Benefits?

- 유연성이 높아진다.
- ARIMA 모델과 달리 시계열 자료 측정 주기가 일정한 간격일 필요가 없다.
- 빈 값을 interpolate 할 필요가 없다.
- Fitting이 매우 빠르다.
- 직관적으로 이해할 수 있는 파라미터를 통해 모형을 조정할 수 있다.

#04 Prophet - Growth

Model Components : Growth

- Piecewise logistic growth model
  - 데이터의 추세가 비선형인 경우
  - 자연적인 상한선이 존재하는 경우



$$g(t) = \frac{C}{1 + \exp(-k(t-m))}$$

C : carrying capacity, k :growth rate, m: offset parameter.

1. Carrying capacity is not constant!
2. Growth rate is not constant!

→ Change Point

$$\mathrm{k} + \sum adjustments$$

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathbf{a}(t)^\intercal \boldsymbol{\delta})(t - (m + \mathbf{a}(t)^\intercal \boldsymbol{\gamma})))}$$

#04 Prophet - Growth

Model Components : Growth

- Piecewise constant rate of growth model
  - 데이터의 추세가 선형인 경우
  - Change point는 분석자에 의해 명시될 수도, 자동으로 선택될 수도 있음.

$$g(t) = (k + \mathbf{a}(t)^\intercal \boldsymbol{\delta})t + (m + \mathbf{a}(t)^\intercal \boldsymbol{\gamma})$$

#04 Prophet – Seasonality

Model Components : Seasonality

- 사용자들의 행동 양식으로 인해 주기적으로 나타나는 패턴
  - Ex) 매주 5일 근무, 매년 일정 기간의 방학 등

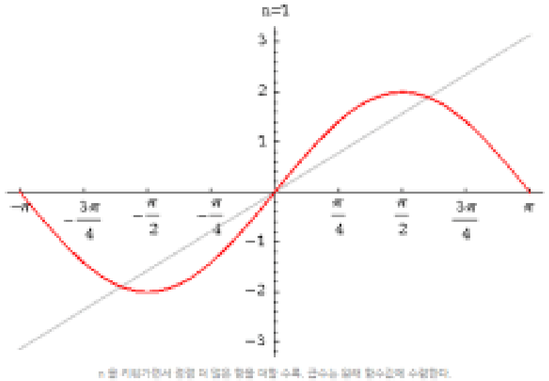
- Prophet은 Fourier Series를 이용해서 패턴의 근사치를 찾는다.
  - P는 시계열의 주기 (연도별 P =365.25, 주별 P = 7)

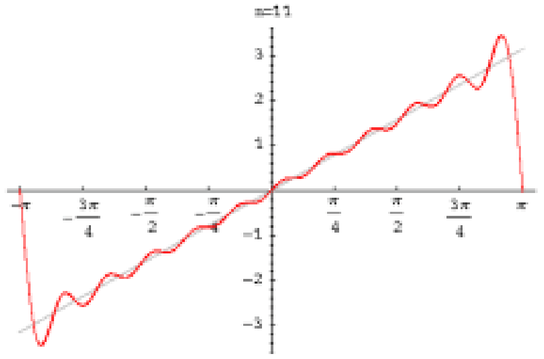
$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$$

$$\boldsymbol{\beta} = [a_1, b_1, \cdots, a_N, b_N]^T$$

$$X(t) = [cos(\frac{2\pi(1)t}{365.25}), sin(\frac{2\pi(1)t}{365.25}), \cdots, cos(\frac{2\pi(10)t}{365.25}), sin(\frac{2\pi(10)t}{365.25})]$$

$$s(t) = X(t)\boldsymbol{\beta}$$





#04 Prophet – Holidays

Model Components : Holidays

- 주기성을 가지지는 않지만 전체 추이에 큰 영향을 주는 이벤트
  → 효과를 모형화하기 어려움

- 분석자가 직접 custom 하여 list 제공
  - 이벤트의 효과는 독립적이라고 가정

$$Z(t) = [\mathbf{1}(t \in D_1), \ldots, \mathbf{1}(t \in D_L)]$$

$$h(t) = Z(t)\boldsymbol{\kappa}.$$

- 이벤트 앞뒤로 window 범위를 지정해서
  해당 이벤트가 미치는 영향의 범위를 설정할 수 있다.

Holiday	Country	Year	Date
Thanksgiving	US	2015	26 Nov 2015
Thanksgiving	US	2016	24 Nov 2016
Thanksgiving	US	2017	23 Nov 2017
Thanksgiving	US	2018	22 Nov 2018
Christmas	*	2015	25 Dec 2015
Christmas	*	2016	25 Dec 2016
Christmas	*	2017	25 Dec 2017
Christmas	*	2018	25 Dec 2018

#04 Prophet – Model Fitting

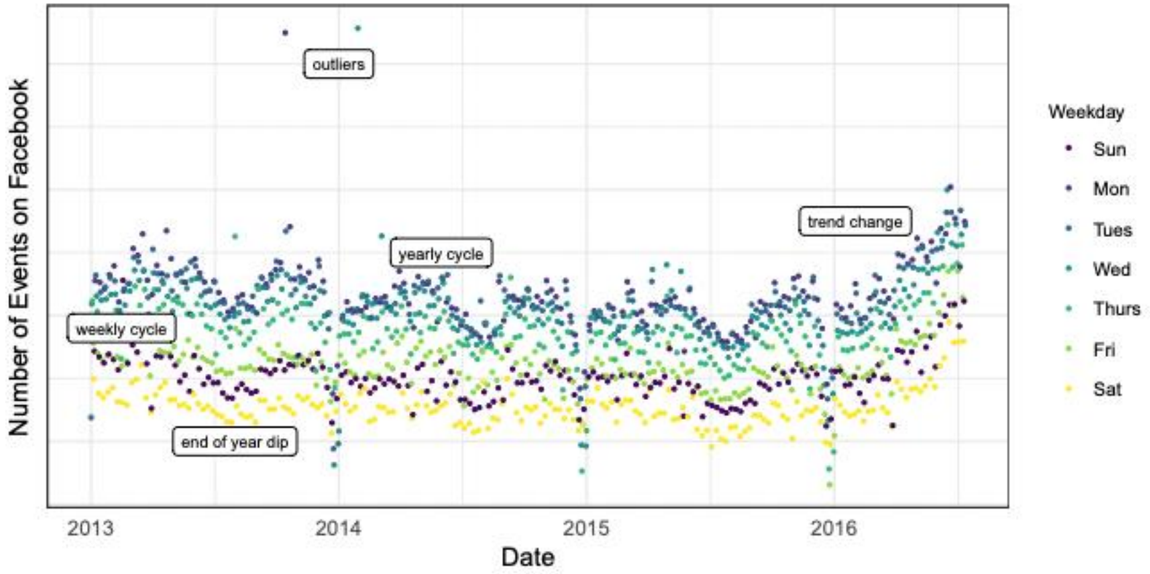
Model Fitting

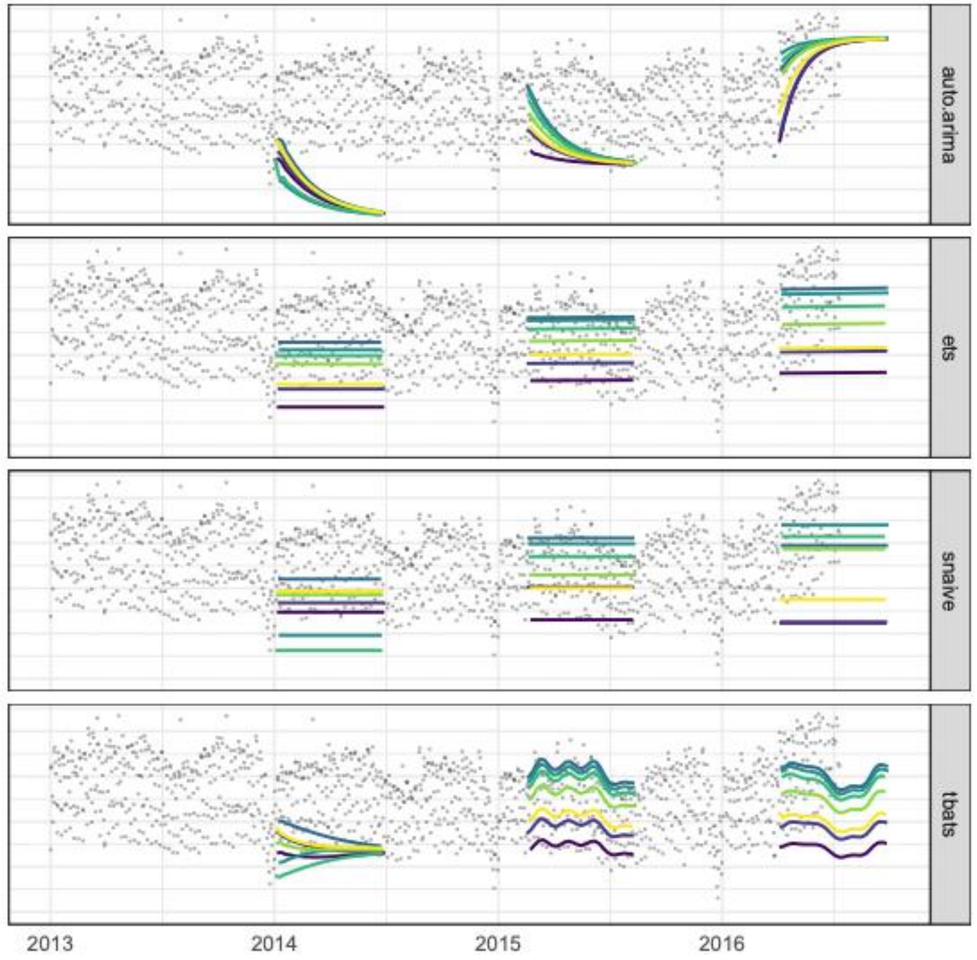
Listing 1: Example Stan code for our complete model.

```
model {
  // Priors
  k ~ normal(0, 5);
  m ~ normal(0, 5);
  epsilon ~ normal(0, 0.5);
  delta ~ double_exponential(0, tau);
  beta ~ normal(0, sigma);

  // Logistic likelihood
  y ~ normal(C ./ (1 + exp(-(k + A * delta) .* (t - (m + A * gamma)))) +
             X * beta, epsilon);
  // Linear likelihood
  y ~ normal((k + A * delta) .* t + (m + A * gamma) + X * beta, sigma);
}
```

#04 Prophet – Model Fitting

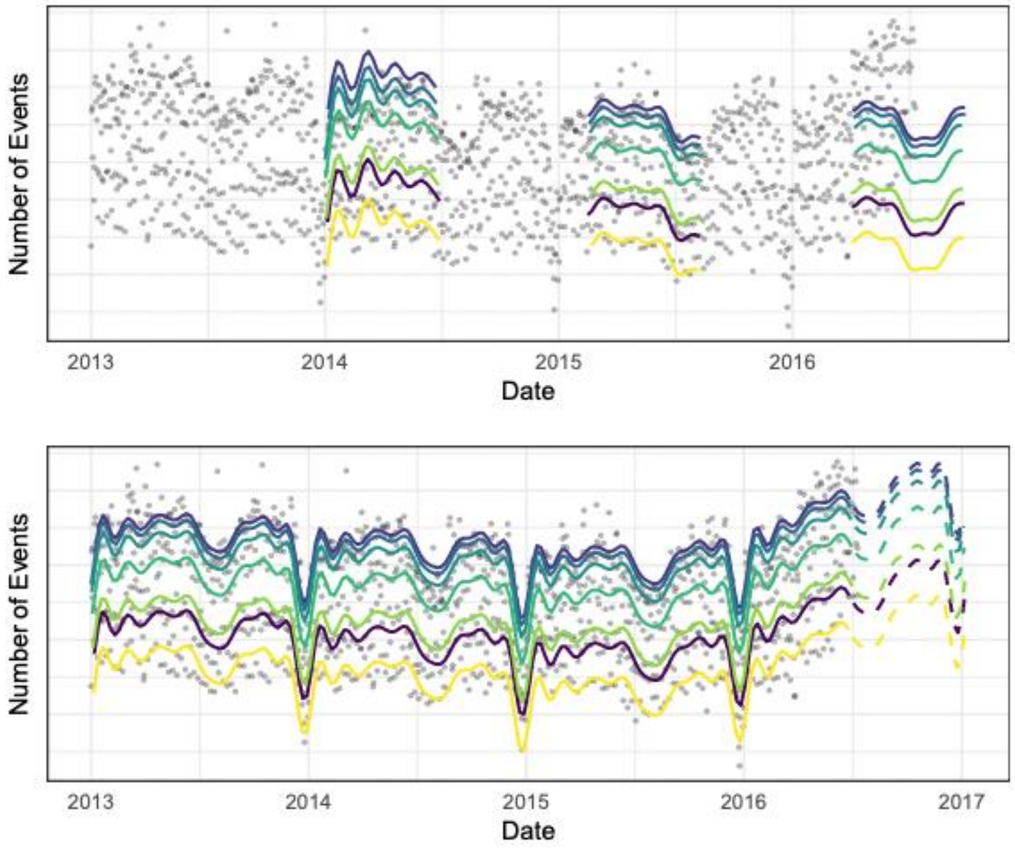
Model Fitting

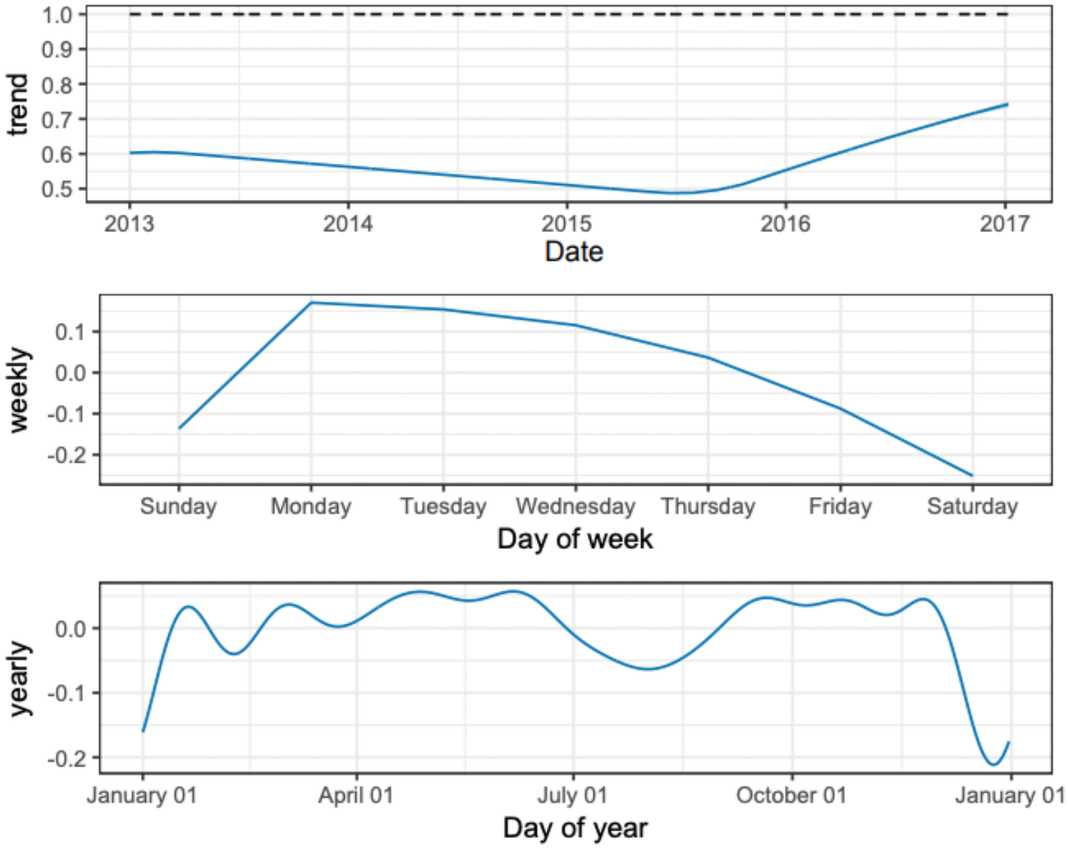




#04 Prophet – Model Fitting

Model Fitting





#04 Prophet

Analyst-in-the-Loop Modeling

- 모형을 구성하는데 필요한 통계적 지식이 없더라도
  직관적인 파라미터를 통해 모형을 조정할 수 있는 업무 경험/ 도메인 지식이 풍부한 사람.
  - Capacities (시계열 데이터 전체의 최대값)
  - Change Points (추세가 변화하는 시점)
  - Holidays & Seasonality (추세에 영향을 미치는 시기적 요인들)
  - Smoothing (각각의 요소들이 전체 추이에 미치는 영향의 정도)

  → 직관적인 파라미터들을 쉽게 조정하여 Prophet을 반복적으로 수행

- 분석가들은 필요한 모델링과 그 결과를 살펴보고, 나머지 피곤한 작업들은 도구가 알아서 해주도록 하는 것이 목적

#04 Prophet

Where Prophet Shines?

- 많은 시계열 데이터 (연 단위 이상)

- 불규칙적으로 일어나지만 사전에 시점을 알고 있는 이벤트가 있는 경우
  - Ex) 설날, 블랙프라이데이

- 결측치가 어느 정도 존재하거나, 이상치가 많은 경우

- 특정한 이벤트로 인해 장기적인 추이가 변할 수 있는 경우
  - Ex) 신제품 출시, 디자인 변경

- 지표가 선형으로 증가하지 않을 경우
  - 증가할 수 있는 지표의 최대치가 존재하고 이를 알고 있는 경우

#04 Model Evaluation

Model Evaluation

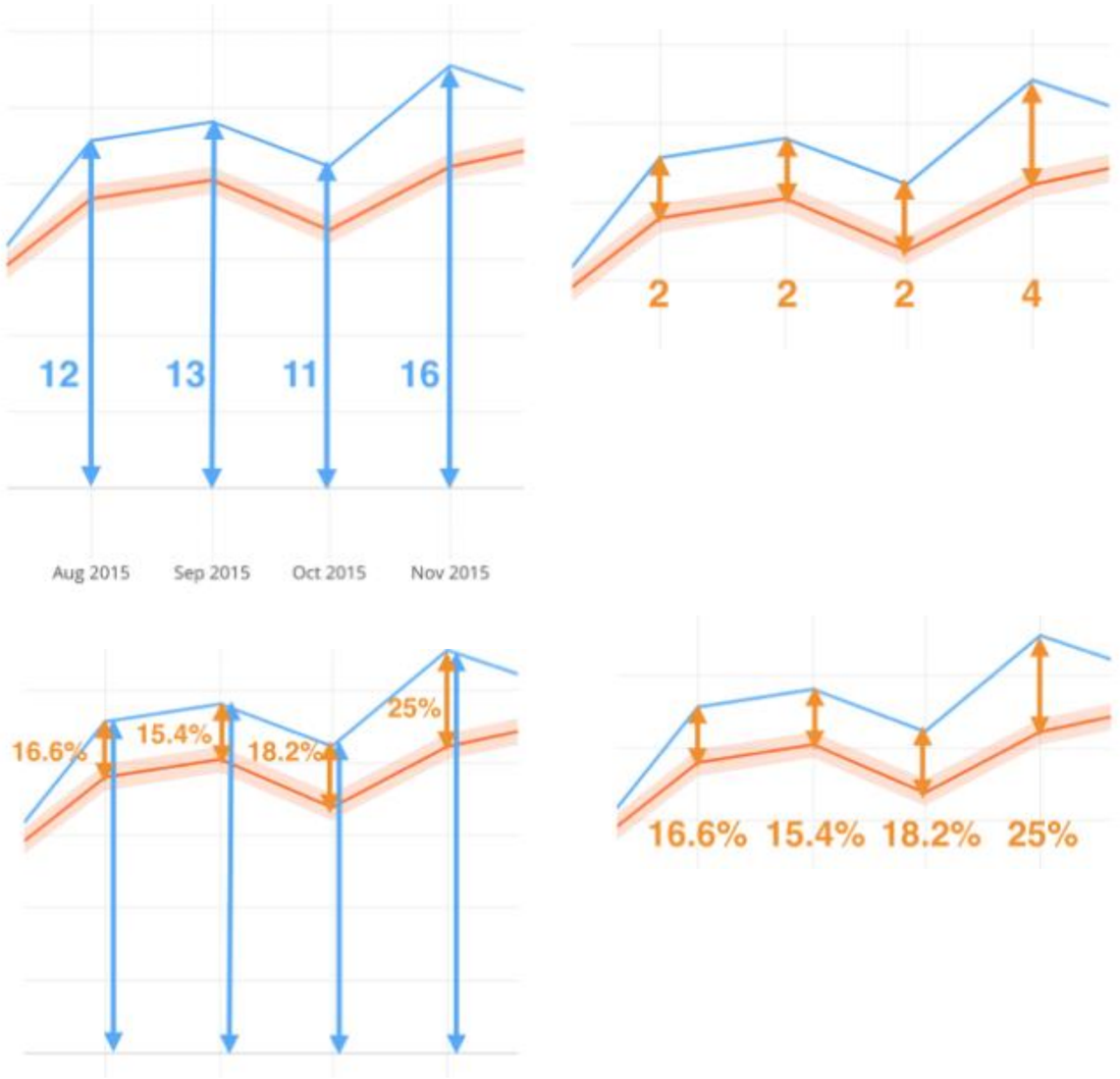
$$\phi(T,h) = d(\hat{y}(T+h|T), y(T+h))$$

- T 주차의 값과
  T 이후 h 주차간 예측 사이의 거리

MAPE - Mean Absolute Percentage Error

$$\mathrm{M} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

- MAE를 비율로 표현한 지표.
- 비율 변수로, 직관적이고 비교에 용이함.
- 실제 값이 0이 존재한다면? 정의 불가.
  실제 값이 1에 가깝다면? 무한대에 가까워짐.
  비율로 해석이 불가능하다면? Ex 기온 예측



#04 Prophet

Parameter Optimization

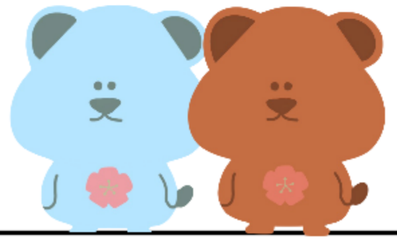
```
params_grid = {'changepoint_prior_scale':[0.1,0.01,0.001],
               'n_changepoints' : [50,100,150],
               'fourier_order' : [5,10],
               'period':[0.1,0.3,0.5]
               }
grid = ParameterGrid(params_grid)
cnt = 0
for p in grid:
    cnt = cnt+1

print('Total Possible Models',cnt)
```

```
Total Possible Models 54
```

```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100
```

```
prophet = Prophet(seasonality_mode='multiplicative',
        yearly_seasonality=False,
        weekly_seasonality=False,
        daily_seasonality=False,
        changepoint_prior_scale=0.1,
        n_changepoints=50
        )
prophet.add_seasonality(name='seasonality_1',period=0.1,fourier_order = 5)
prophet.add_seasonality(name='seasonality_2', period=0.3,fourier_order = 5)
prophet.add_seasonality(name='seasonality_3', period=0.01, fourier_order = 5)
prophet.fit(x_df)
```

THANK YOU



EWHA EURON