약국 일별 판매량 예측

Drugstore Sales Prediction

버닝썬조 : 최인서 최재영 박준영 이태욱 주여진

목 차

 데이터 소개

 데이터 전처리

 모델 선정

 다음주 예고

💊 데이터 소개

1 데이터 테이블

Data File	No. Variables	No. Observations	Description
Store	10	1,115	Details on 1,115 stores
Train	9	949,194	881 days (13.01.01~15.05.31)
Test	8	46,830	6 weeks (15.06.20~15.07.31)

1 데이터 테이블

Data File	No. Variables	No. Observations	Description
Store	10	1,115	Details on 1,115 stores
Train	9	949,194	881 days (13.01.01~15.05.31) *881 * 1,115 = 982,315*
Test	8	46,830	6 weeks (15.06.20~15.07.31)

1 데이터 테이블

Store.csv

Variables	Values	Type
StoreType	a, b, c, d	Nominal
Assortment	a: Basic b: Extra c: Extended	Nominal
CompetitionDistance	20 ~ 75,860	Ratio
CompetitionOpenSince Month/Year	1(Jan) to 12(Dec) / 1900-2015	Interval
Promo2	0 or 1	Nominal
Promo2SinceWeek/Year	1 ~ 50 / 2009 ~ 2015	Interval
PromoInterval	(jan, apr, jul, oct) (fab, may, aug, nov) (mar, jun, sept, dec)	Ordinal

1 데이터 테이블

Train.csv

Variables	Values	Type
Store	1 ~ 1,115	Nominal
DayOfWeek	1 ,2 ,3 ,4 ,5 ,6 ,7	Nominal
Date	01/01/2013 ~ 05/31/2015	Interval
Open/Promo	0 0r 1	Nominal
StateHoliday	a: Public Holiday b: Easter Holiday c: Christmas Holiday 0: None	Nominal
SchoolHoliday	0 or 1	Nominal
Customers	0 ~ 7,338	Ratio
Sales	0 ~ 41,551	Ratio

1 데이터 테이블

Train.csv

Variables	Values	Type
Store	1 ~ 1,115	Nominal
DayOfWeek	1 ,2 ,3 ,4 ,5 ,6 ,7	Nominal
Date	01/01/2013 ~ 05/31/2015	Interval
Open/Promo	0 0r 1	Nominal
StateHoliday	a: Public Holiday b: Easter Holiday c: Christmas Holiday 0: None	Nominal
SchoolHoliday	0 or 1	Nominal
Customers	0 ~ 7,338	Ratio
Sales	0 ~ 41,551	Ratio

1 평가지표

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2}$$

y_i	1	10		
$\hat{y}_i$	10	1		
RMSE	81	81		
RMSPE	81	0.81		

1 평가지표

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2}$$

y_i	1	10	91	100
$\hat{y}_i$	10	1	100	91
RMSE	81	81	81	81
RMSPE	81	0.81	0.0098	0.0081

1 평가지표

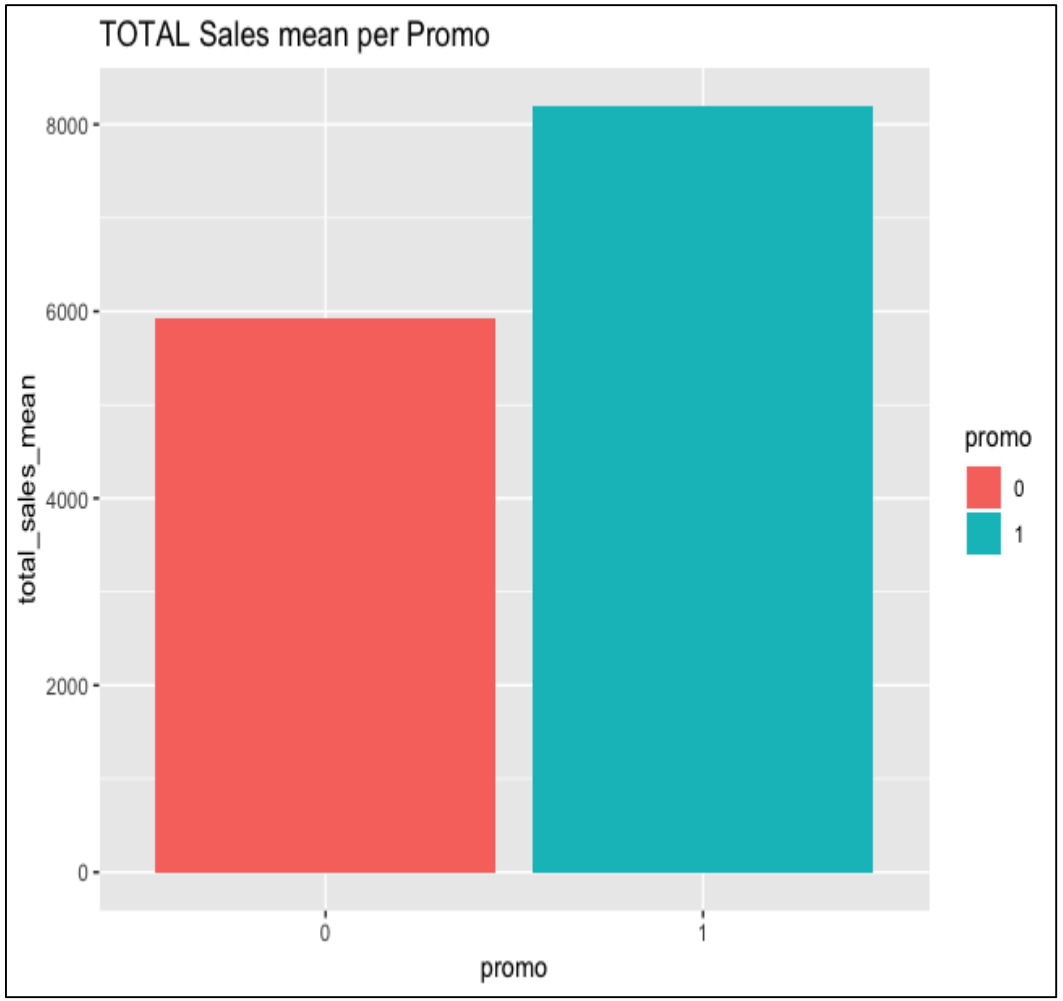
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{y_i})^2}$$

y_i	1	10	91	100
$\hat{y}_i$	10	1	100	91
RMSE	81	81	81	81
RMSPE	81	0.81	0.0098	0.0081

2 변수 분포 알아보기

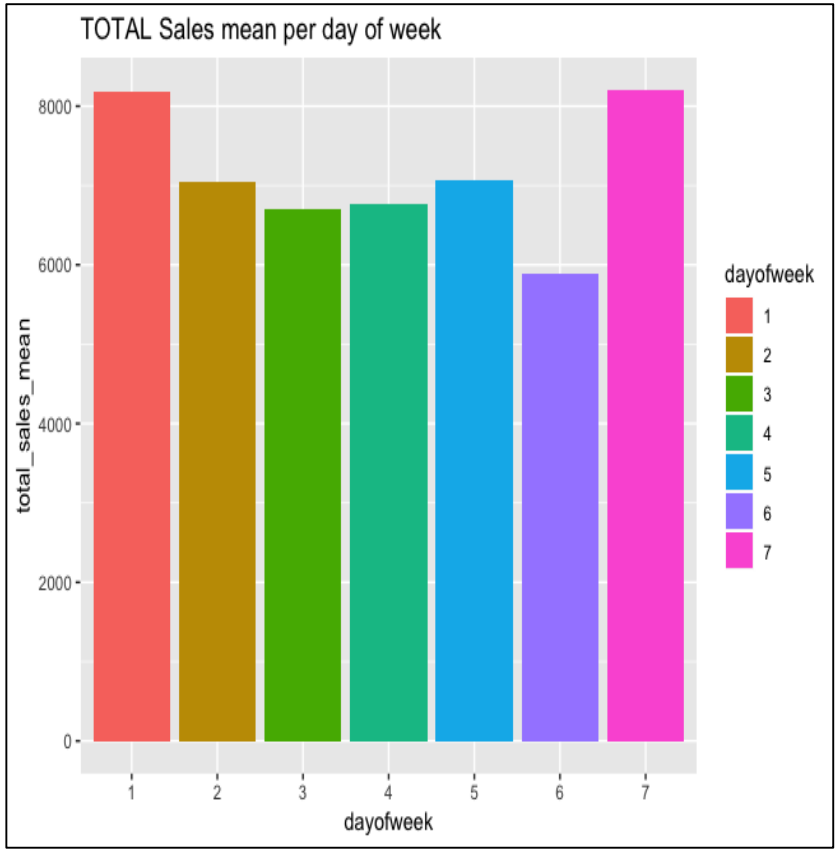
▶Promotion 진행 여부



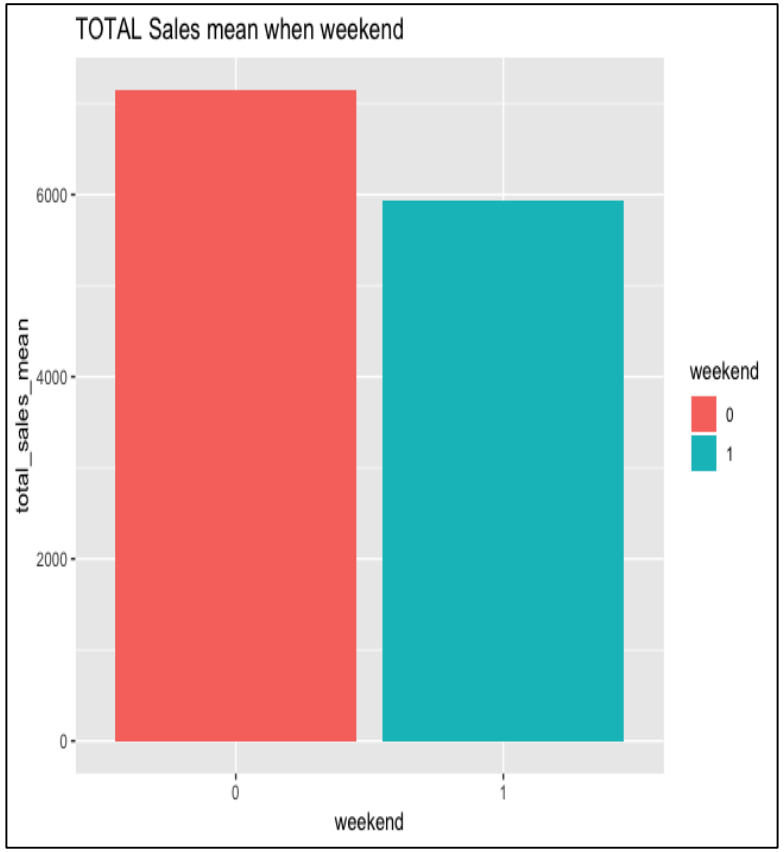
▶State Holiday 종류

2 변수 분포 알아보기

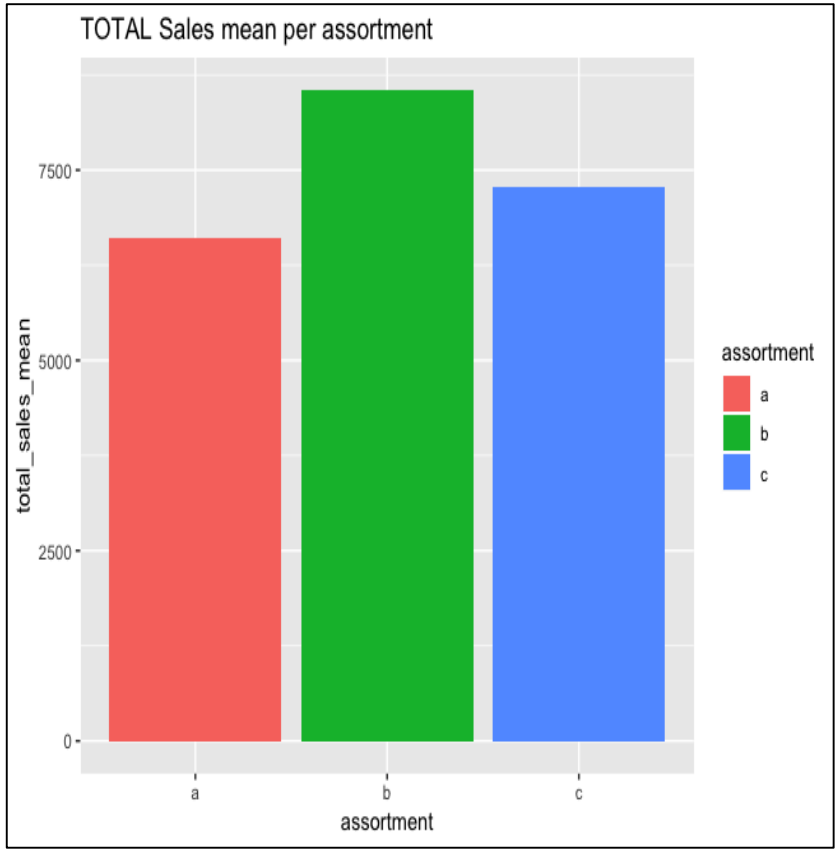
▶day of week 기준



▶weekend 여부

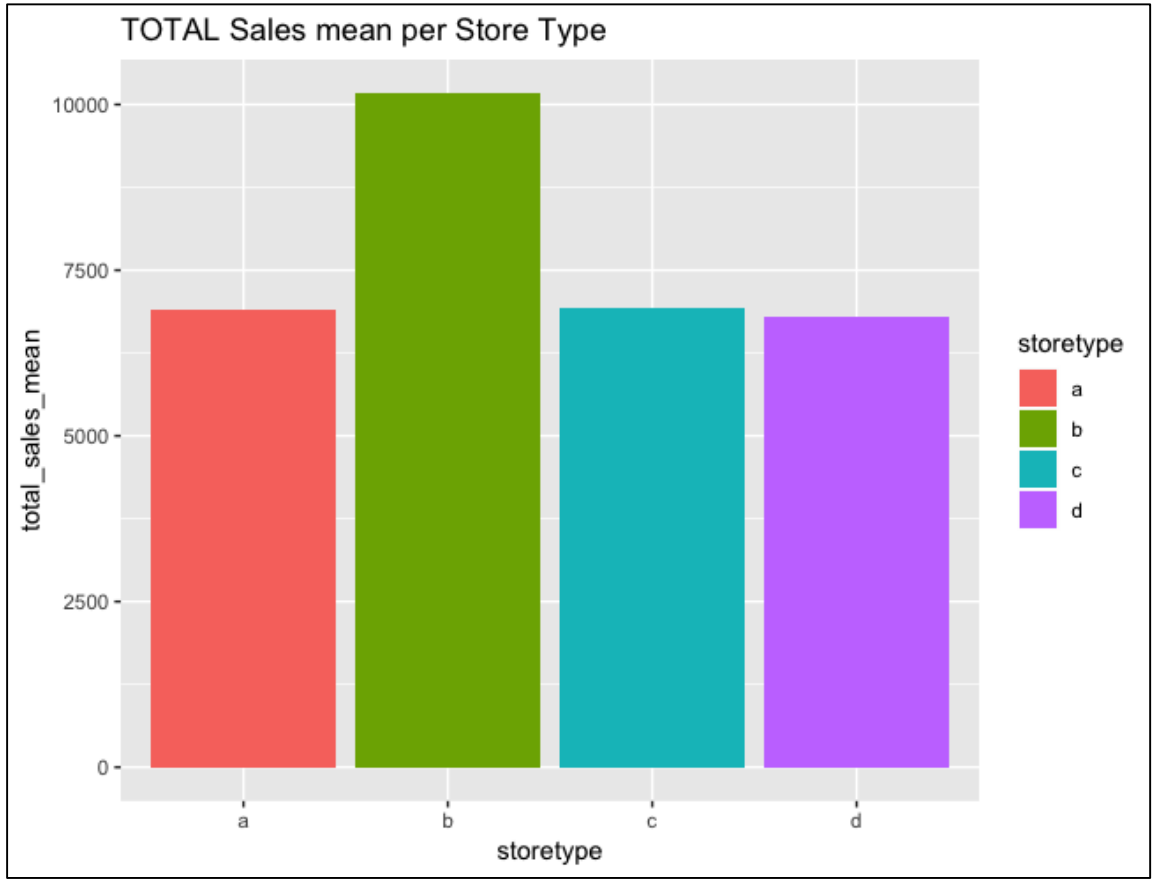


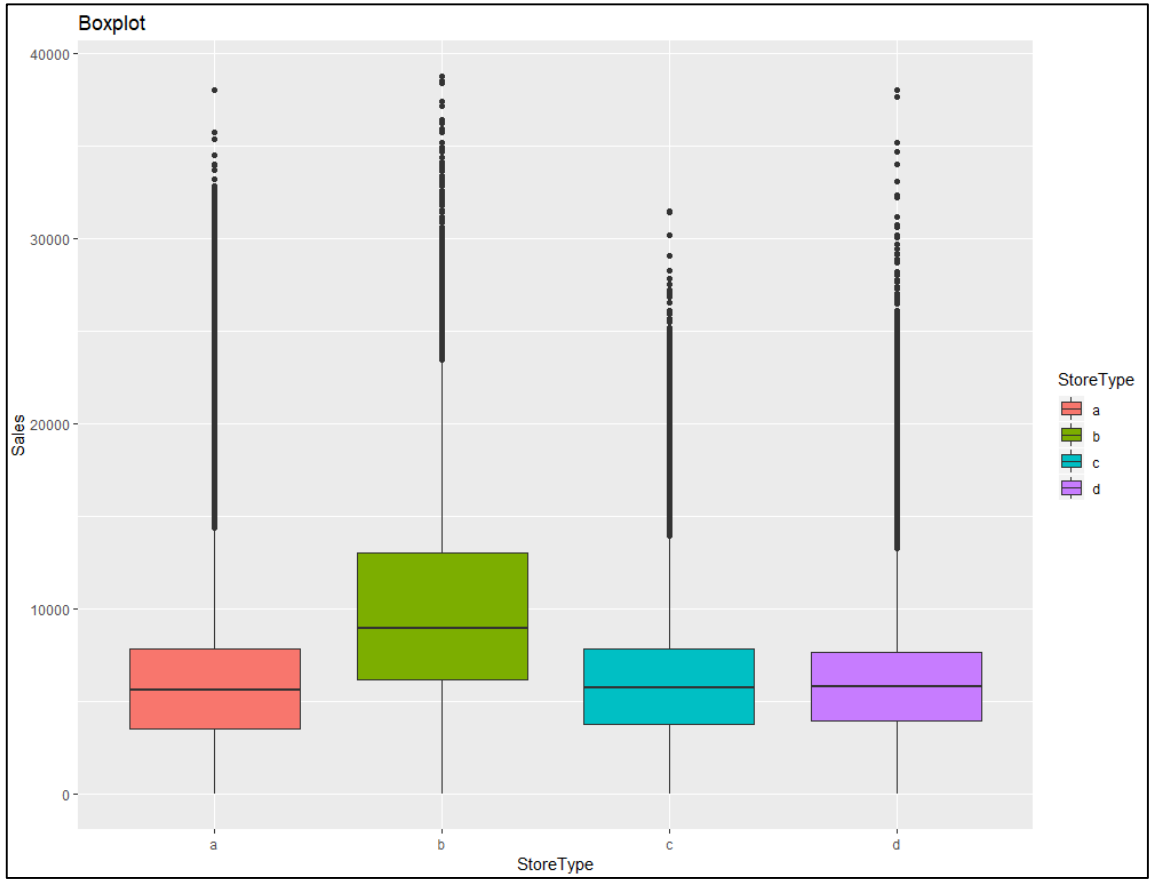
▶assortment 종류



2 변수 분포 알아보기

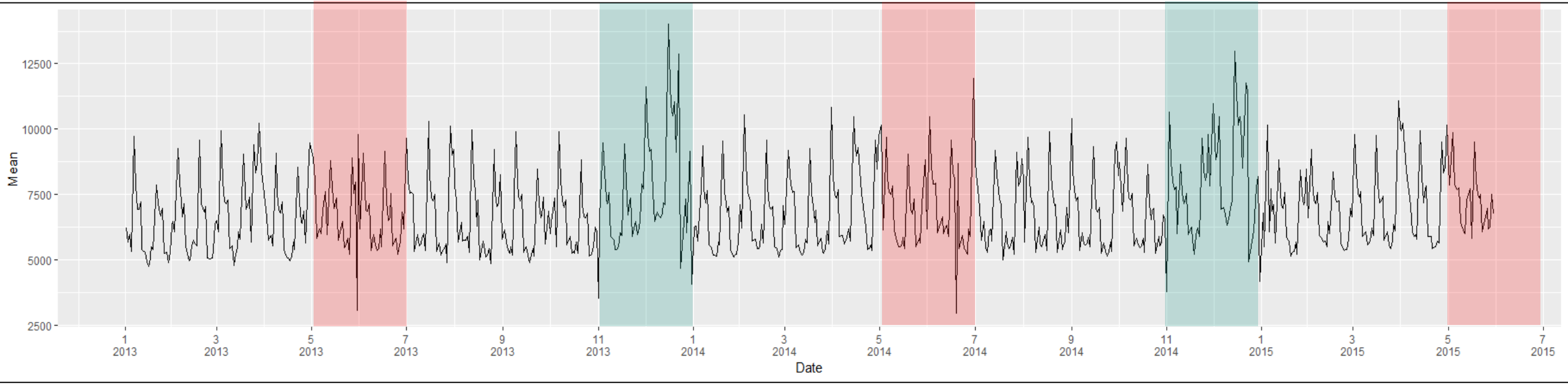
▶Store type 종류





3 추세 살펴보기

▶1,115개 Store 의 Sales 평균 [train set]



Test set 기간(6월-7월)

연말(11월-12월)에 매출 증가

💊 데이터 전처리

1 Raw Data

Train.csv

```
> str(raw.train)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':       949194 obs. of  9 variables:
 $ Store         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ DayOfWeek     : int  2 2 2 2 2 2 2 2 2 2 ...
 $ Date          : Date, format: "2013-01-01" "2013-01-01" "2013-01-01" ...
 $ Sales         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Customers     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Open          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Promo         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ StateHoliday : Factor w/ 4 levels "0","a","b","c": 2 2 2 2 2 2 2 2 2 2 ...
 $ SchoolHoliday: int  1 1 1 1 1 1 1 1 1 1 ...
```

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday
1	1	2	2013-01-01	0	0	0	0	a	1
2	2	2	2013-01-01	0	0	0	0	a	1
3	3	2	2013-01-01	0	0	0	0	a	1
4	4	2	2013-01-01	0	0	0	0	a	1
5	5	2	2013-01-01	0	0	0	0	a	1
6	6	2	2013-01-01	0	0	0	0	a	1
7	7	2	2013-01-01	0	0	0	0	a	1
8	8	2	2013-01-01	0	0	0	0	a	1
9	9	2	2013-01-01	0	0	0	0	a	1
10	10	2	2013-01-01	0	0	0	0	a	1
11	11	2	2013-01-01	0	0	0	0	a	1
12	12	2	2013-01-01	0	0	0	0	a	1
13	13	2	2013-01-01	0	0	0	0	a	1
14	14	2	2013-01-01	0	0	0	0	a	1
15	15	2	2013-01-01	0	0	0	0	a	1
16	16	2	2013-01-01	0	0	0	0	a	1
17	17	2	2013-01-01	0	0	0	0	a	1
18	18	2	2013-01-01	0	0	0	0	a	1
19	19	2	2013-01-01	0	0	0	0	a	1
20	20	2	2013-01-01	0	0	0	0	a	1

Showing 1 to 20 of 949,194 entries

1 Raw Data

Test.csv

	ID	Store	DayOfWeek	Date	Open	Promo	StateHoliday	SchoolHoliday
1	1	1	6	2015-06-20	1	0	0	0
2	2	2	6	2015-06-20	1	0	0	0
3	3	3	6	2015-06-20	1	0	0	0
4	4	4	6	2015-06-20	1	0	0	0
5	5	5	6	2015-06-20	1	0	0	0
6	6	6	6	2015-06-20	1	0	0	0
7	7	7	6	2015-06-20	1	0	0	0
8	8	8	6	2015-06-20	1	0	0	0
9	9	9	6	2015-06-20	1	0	0	0
10	10	10	6	2015-06-20	1	0	0	0
11	11	11	6	2015-06-20	1	0	0	0
12	12	12	6	2015-06-20	1	0	0	0
13	13	13	6	2015-06-20	1	0	0	0
14	14	14	6	2015-06-20	1	0	0	0
15	15	15	6	2015-06-20	1	0	0	0
		16	6	2015-06-20	1	0	0	0
		17	6	2015-06-20	1	0	0	0
		18	6	2015-06-20	1	0	0	0
		19	6	2015-06-20	1	0	0	0
		20	6	2015-06-20	1	0	0	0

46,830 entries

```
> str(raw.test)
Classes 'tbl_df', 'tbl' and 'data.frame':       46830 obs. of  8 variables:
 $ ID           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Store        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ DayOfWeek    : int  6 6 6 6 6 6 6 6 6 6 ...
 $ Date         : Date, format: "2015-06-20" "2015-06-20" "2015-06-20" ...
 $ Open         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Promo        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ StateHoliday : int  0 0 0 0 0 0 0 0 0 0 ...
 $ SchoolHoliday: int  0 0 0 0 0 0 0 0 0 0 ...
```

Raw Data

Store.csv

	Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear	PromoInterval
1	1	c	a	1270	9	2008	0	NA	NA	
2	2	a	a	570	11	2007	1	13	2010	Jan,Apr,Jul,Oct
3	3	a	a	14130	12	2006	1	14	2011	Jan,Apr,Jul,Oct
4	4	c	c	620	9	2009	0	NA	NA	
5	5	a	a	29910	4	2015	0	NA	NA	
6	6	a	a	310	12	2013	0	NA	NA	
7	7	a	c	24000	4	2013	0	NA	NA	
8	8	a	a	7520	10	2014	0	NA	NA	
9	9	a	c	2030	8	2000	0	NA	NA	
10	10	a	a	3160	9	2009	0	NA	NA	
11	11	a	c	960	11	2011	1	1	2012	Jan,Apr,Jul,Oct
12	12	a	c	1070	NA	NA	1	13	2010	Jan,Apr,Jul,Oct
13	13	d	a	310	NA	NA	1	45	2009	Feb,May,Aug,Nov
14	14	a	a	1300	3	2014	1	40	2011	Jan,Apr,Jul,Oct
15	15	d	c	4110	3	2010	1	14	2011	Jan,Apr,Jul,Oct
16	16	a	c	3270	NA	NA	0	NA	NA	
17	17	a	a	50	12	2005	1	26	2010	Jan,Apr,Jul,Oct
18	18	d	c	13840	6	2010	1	14	2012	Jan,Apr,Jul,Oct
19	19	a	c	3240	NA	NA	1	22	2011	Mar,Jun,Sept,Dec
20	20	d	a	2340	5	2009	1	40	2014	Jan,Apr,Jul,Oct

Showing 1 to 20 of 1,115 entries

Data Merge

Store | Train

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	SchoolHoliday	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	Promo2SinceWeek	Promo2SinceYear	PromoInterval
1	1	2	2013-01-01	0	0	0	0	a	1	c	a	1270	9	2008	0	NA	NA	
2	2	2	2013-01-01	0	0	0	0	a	1	a	a	570	11	2007	1	13	2010	Jan,Apr,Jul,Oct
3	3	2	2013-01-01	0	0	0	0	a	1	a	a	14130	12	2006	1	14	2011	Jan,Apr,Jul,Oct
4	4	2	2013-01-01	0	0	0	0	a	1	c	c	620	9	2009	0	NA	NA	
5	5	2	2013-01-01	0	0	0	0	a	1	a	a	29910	4	2015	0	NA	NA	
6	6	2	2013-01-01	0	0	0	0	a	1	a	a	310	12	2013	0	NA	NA	
7	7	2	2013-01-01	0	0	0	0	a	1	a	c	24000	4	2013	0	NA	NA	
8	8	2	2013-01-01	0	0	0	0	a	1	a	a	7520	10	2014	0	NA	NA	
9	9	2	2013-01-01	0	0	0	0	a	1	a	c	2030	8	2000	0	NA	NA	
10	10	2	2013-01-01	0	0	0	0	a	1	a	a	3160	9	2009	0	NA	NA	
11	11	2	2013-01-01	0	0	0	0	a	1	a	c	960	11	2011	1	1	2012	Jan,Apr,Jul,Oct
12	12	2	2013-01-01	0	0	0	0	a	1	a	c	1070	NA	NA	1	13	2010	Jan,Apr,Jul,Oct
13	13	2	2013-01-01	0	0	0	0	a	1	d	a	310	NA	NA	1	45	2009	Feb,May,Aug,Nov
14	14	2	2013-01-01	0	0	0	0	a	1	a	a	1300	3	2014	1	40	2011	Jan,Apr,Jul,Oct
15	15	2	2013-01-01	0	0	0	0	a	1	d	c	4110	3	2010	1	14	2011	Jan,Apr,Jul,Oct
16	16	2	2013-01-01	0	0	0	0	a	1	a	c	3270	NA	NA	0	NA	NA	
17	17	2	2013-01-01	0	0	0	0	a	1	a	a	50	12	2005	1	26	2010	Jan,Apr,Jul,Oct
18	18	2	2013-01-01	0	0	0	0	a	1	d	c	13840	6	2010	1	14	2012	Jan,Apr,Jul,Oct
19	19	2	2013-01-01	0	0	0	0	a	1	a	c	3240	NA	NA	1	22	2011	Mar,Jun,Sept,Dec
20	20	2	2013-01-01	0	0	0	0	a	1	d	a	2340	5	2009	1	40	2014	Jan,Apr,Jul,Oct

Showing 1 to 20 of 949,194 entries

Store.csv와 train.csv, test.csv를 merge

Key 는 'Store'

1 TRAIN SET 결측값 살펴보기

Train Set 결측값

1) 데이터가 없는 경우

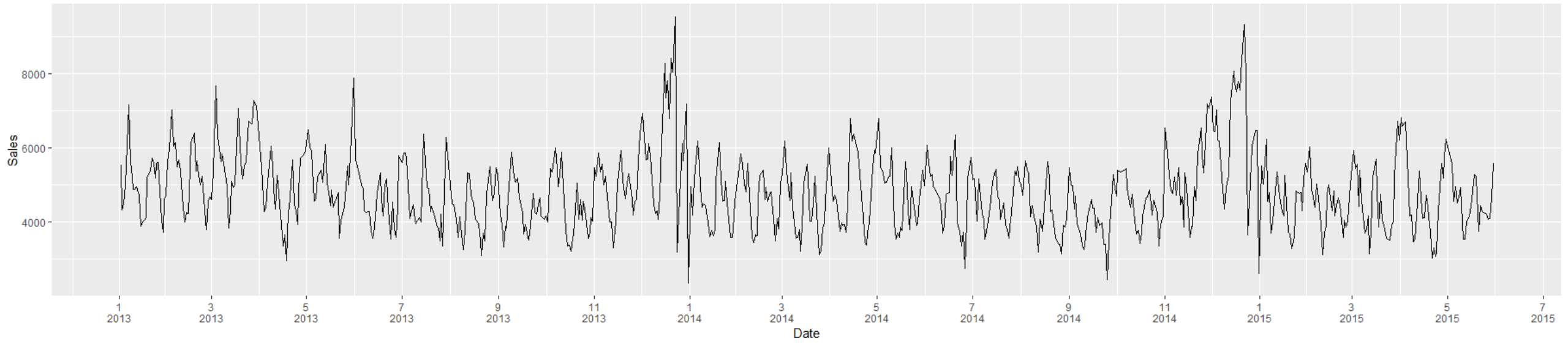
2) 데이터가 있는 경우

(공휴일도 일요일도 아니지만 sales가 없는 경우)

1 TRAIN SET 결측값 살펴보기

(1) 데이터가 없는 경우

Plot(Store 1)



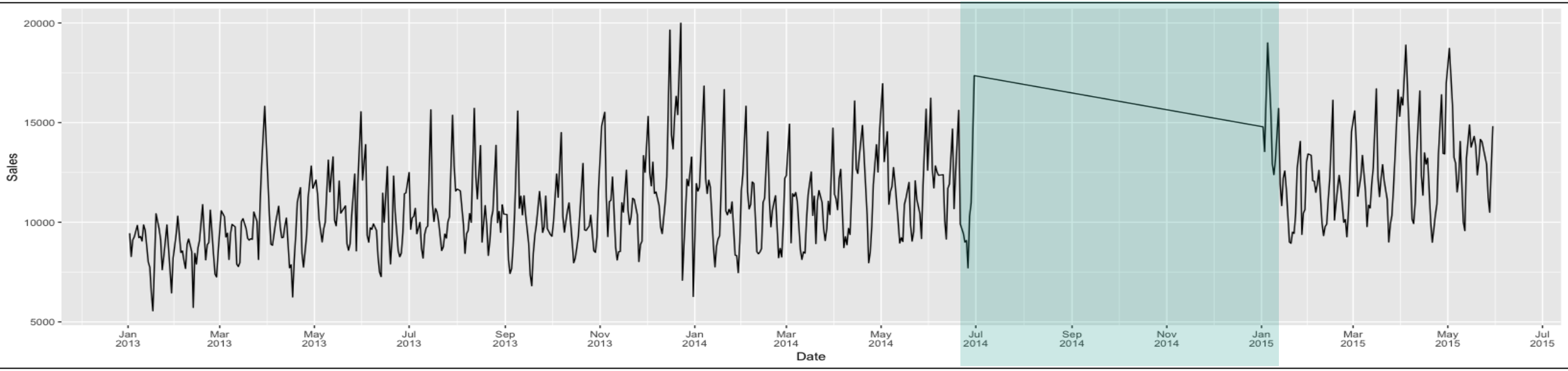
(패턴의 가시성을 헤치는 Sales=0인 데이터 제외)

일정한 평균/분산으로 보이지 않으나
연속적인 그래프를 보임

1 TRAIN SET 결측값 살펴보기

(1) 데이터가 없는 경우

Plot(Store 108)



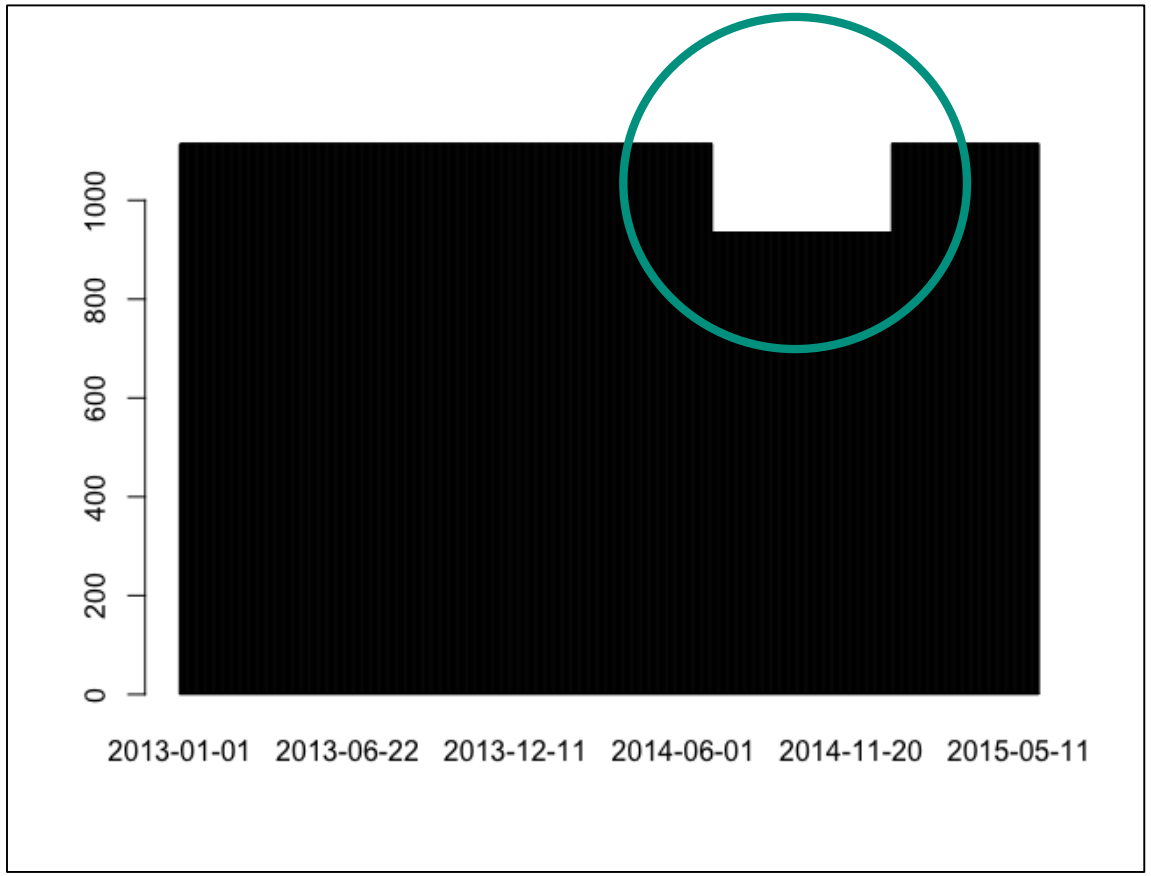
반면 Store 108의 경우 특정 행 부분이 빠져 있음.

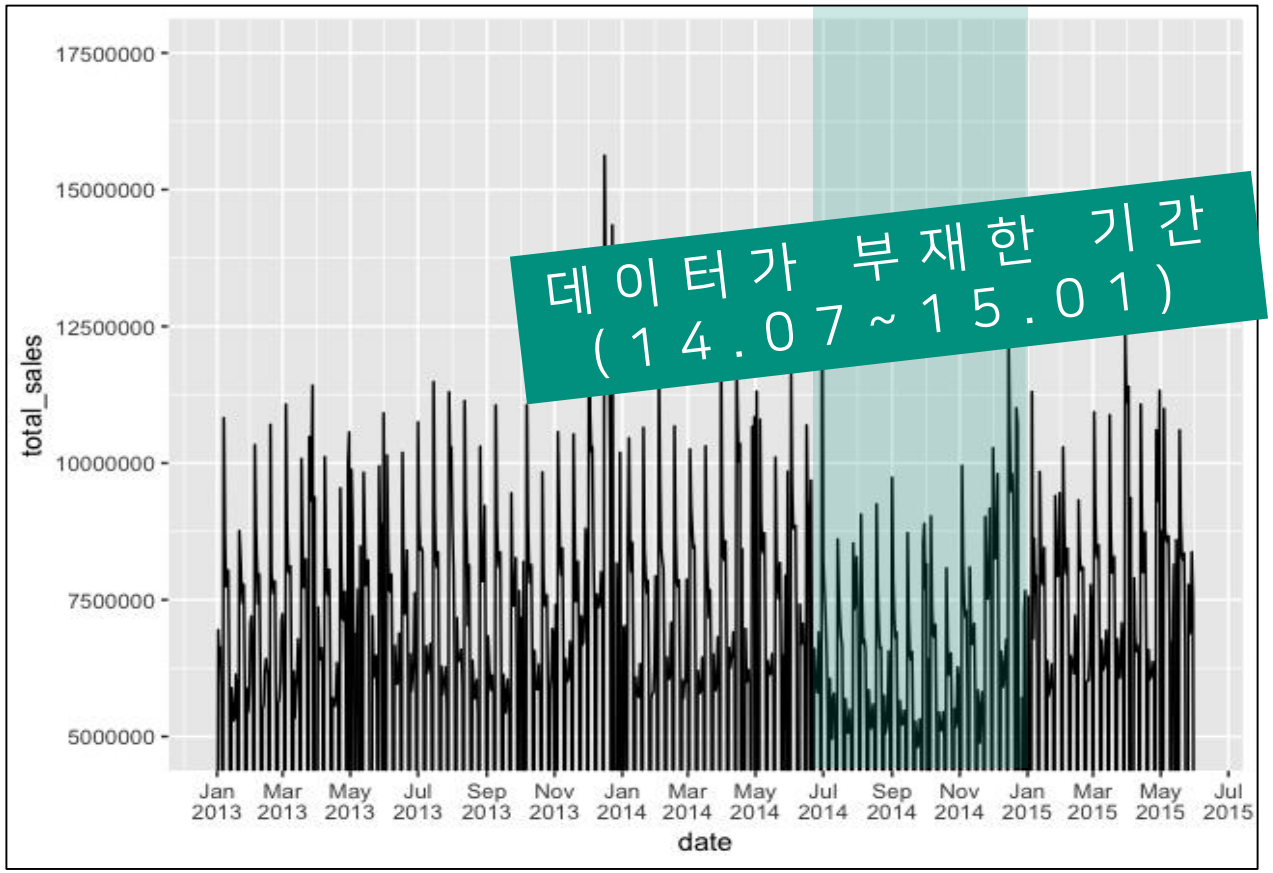
=> 결측치 처리 필요

1 TRAIN SET 결측값 살펴보기

(1) 데이터가 없는 경우

▶1,115개 Store 의 기간별 분포[train set]





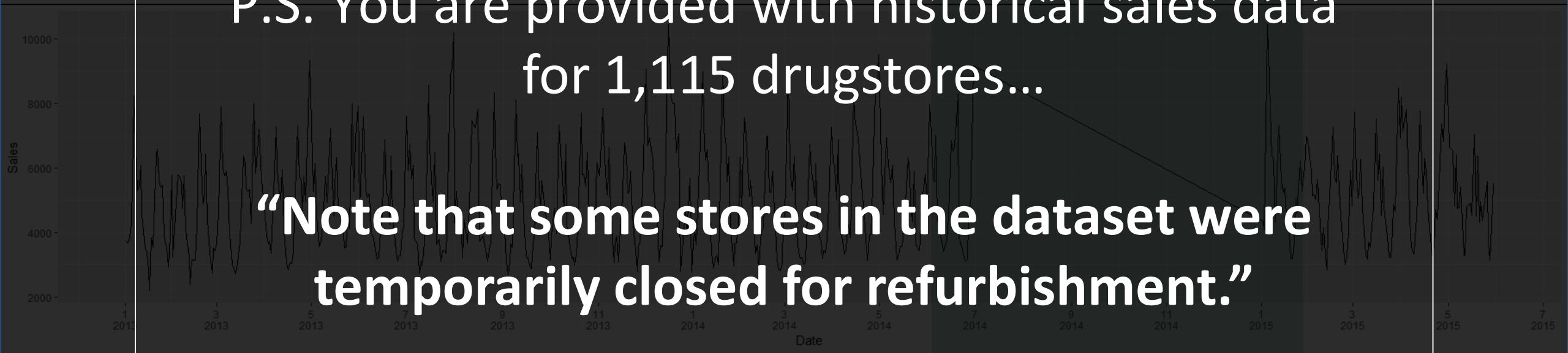
WHY?

columns_description.csv

Store 13 (881일 중 697일치 데이터만 존재)

P.S. You are provided with historical sales data for 1,115 drugstores...

“Note that some stores in the dataset were temporarily closed for refurbishment.”

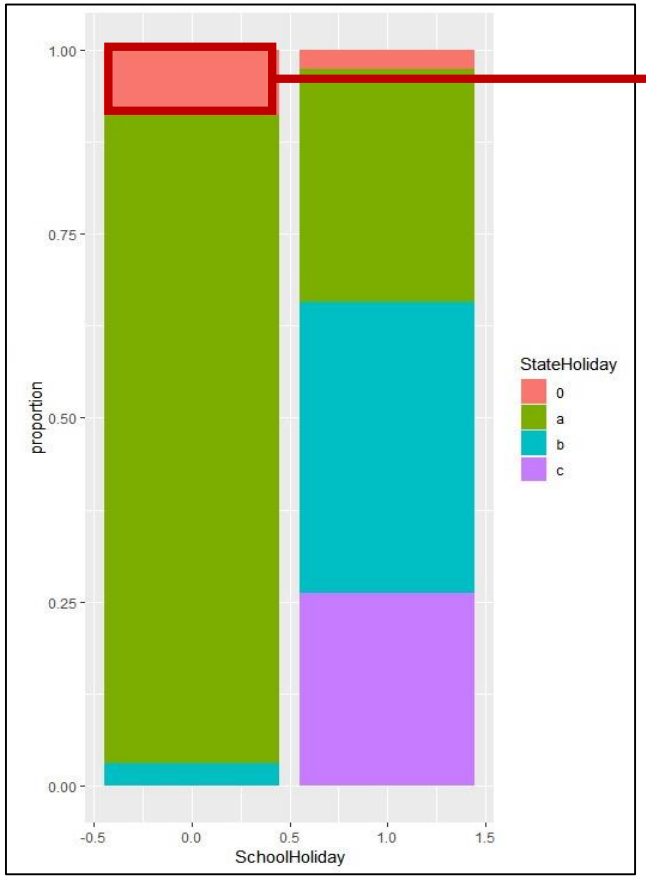


하지만 모든 store이 보수공사를 진행한 것이 아니다.

1115개의 store중 181개의 store이 부재!

1 TRAIN SET 결측값 살펴보기

(2) 데이터가 있는 경우 * 일요일도 공휴일도 아니지만 문을 닫은 경우



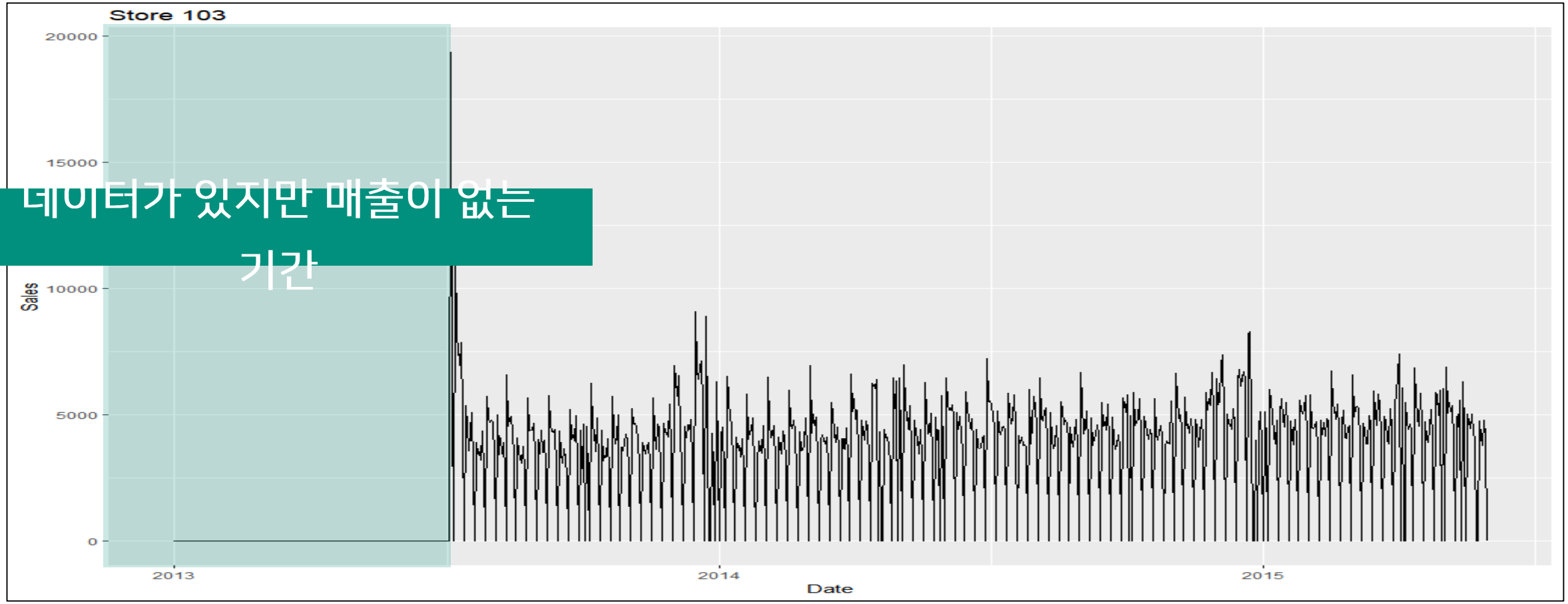
<School holiday와 state holiday 의 결합분포>

storename	frequency
5	1
20	3
22	3
25	27
28	2
30	2
40	3
57	10
74	3
86	1
90	1
100	16
102	1
103	123
105	19
106	2
118	1
123	6

<School holiday, state holiday 모두 0인 경우의 빈도수>

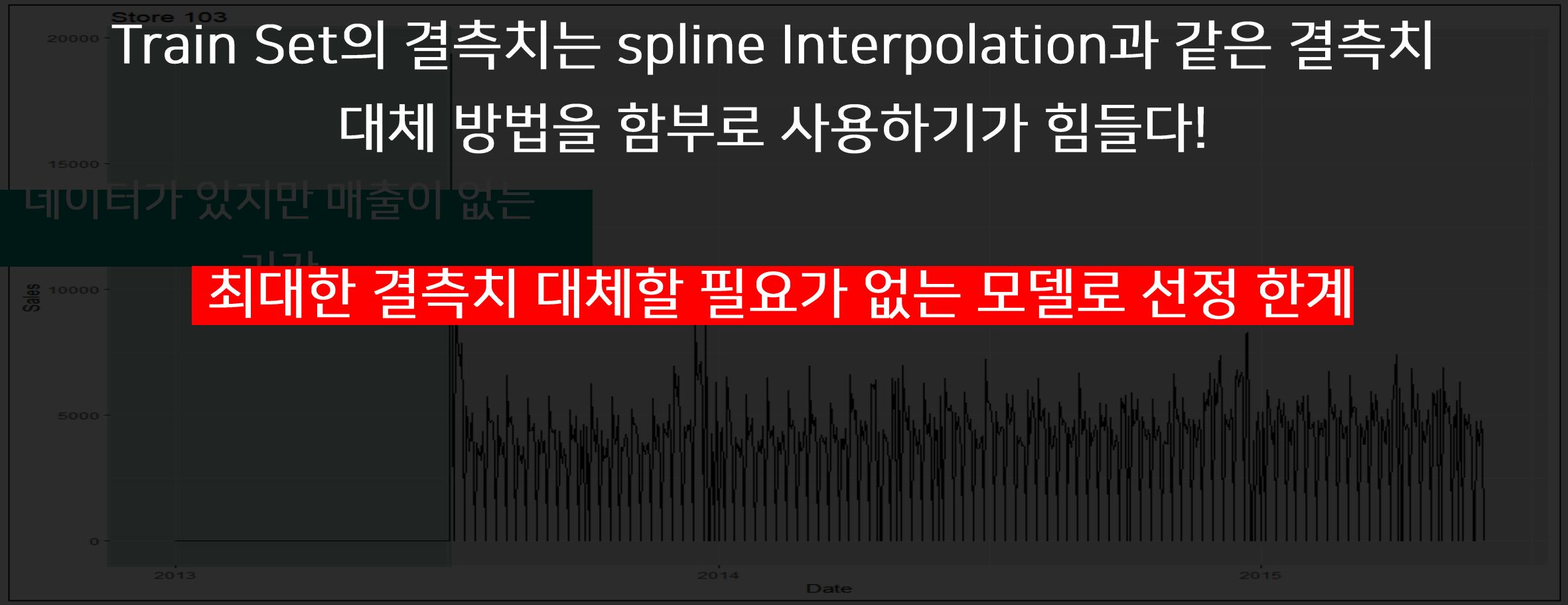
1 TRAIN SET 결측치 살펴보기

(2) 데이터가 있는 경우 * Store 103를 살펴보면,



1 TRAIN SET 결측치 살펴보기

(2) 데이터가 있는 경우 * Store 103를 살펴보면,



Train Set의 결측치는 spline Interpolation과 같은 결측치 대체 방법을 함부로 사용하기가 힘들다!

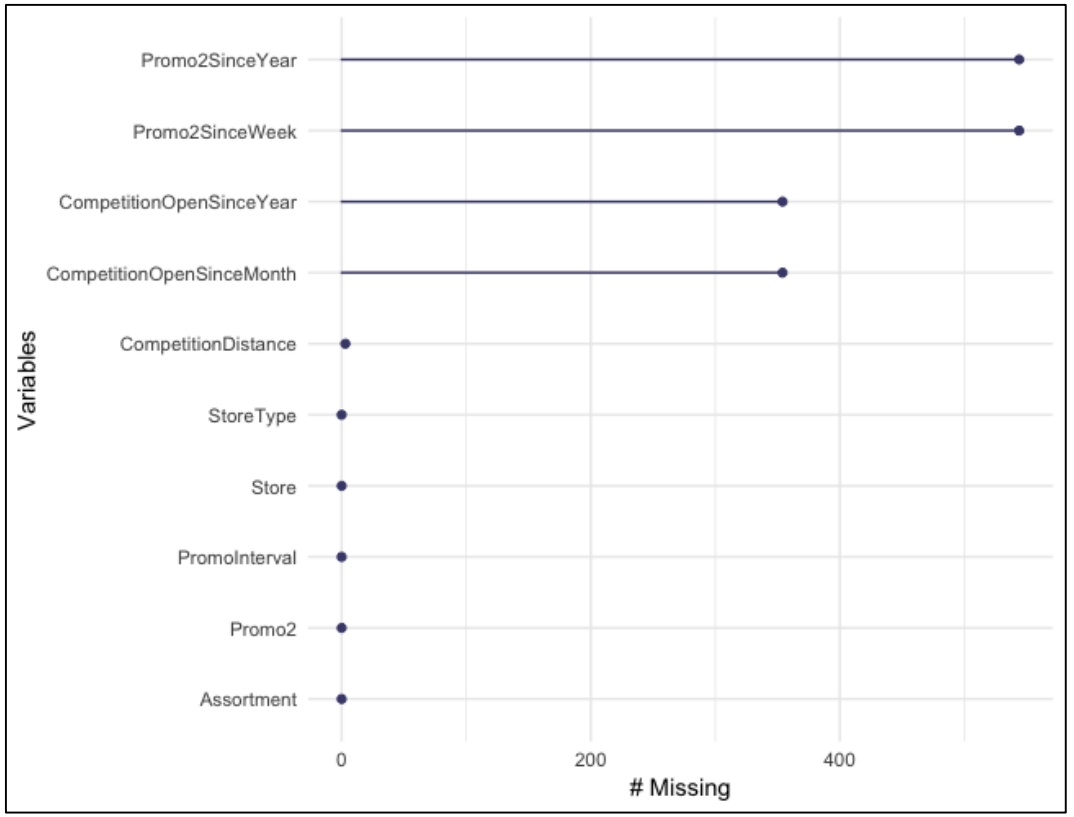
데이터가 있지만 매출이 없는 경우

최대한 결측치 대체할 필요가 없는 모델로 선정 한계

2 STORE SET 결측치 살펴보기

▶Store set: Competition, Promo2 관련 결측치 존재

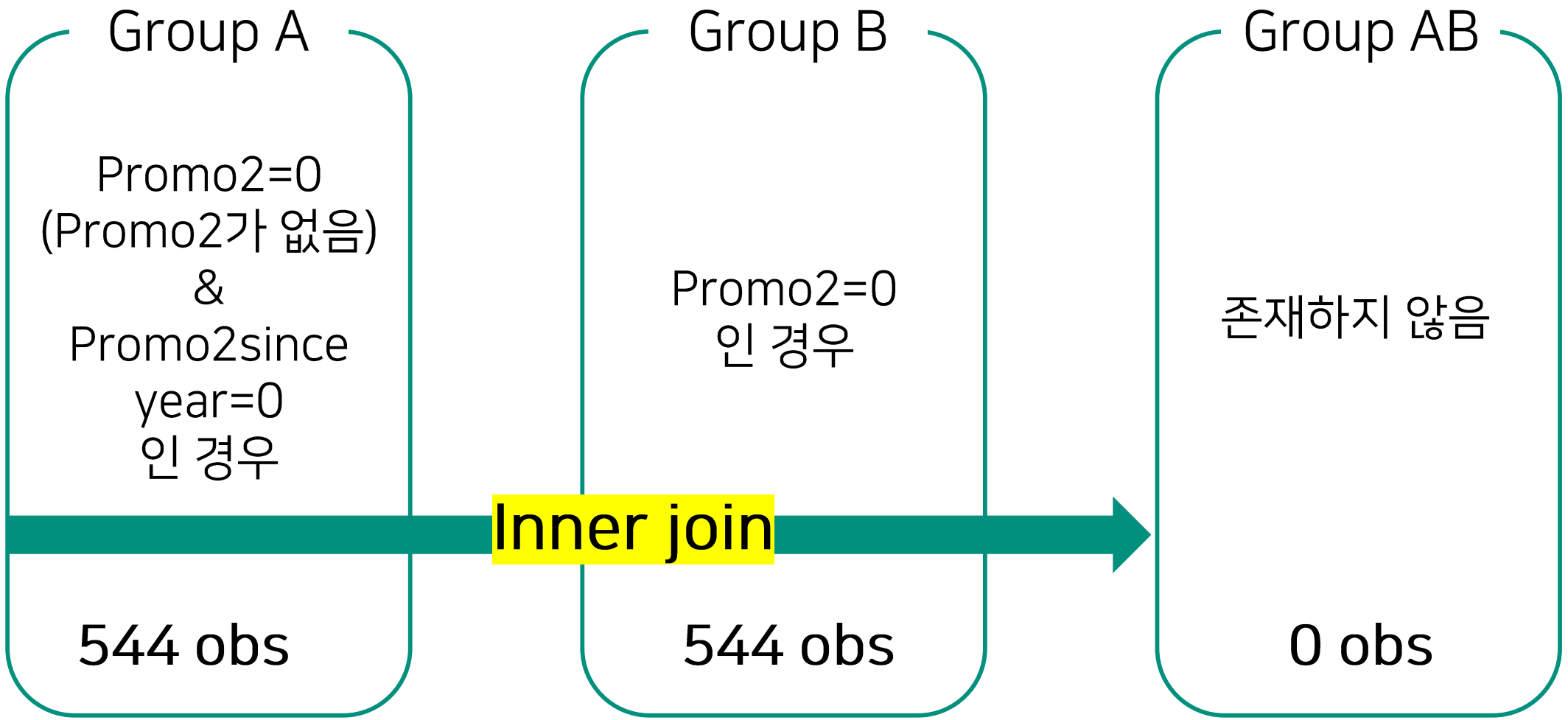
```
> colSums(is.na(store))

                Store             StoreType            Assortment
                    0                     0                     0
CompetitionDistance CompetitionOpenSinceMonth  CompetitionOpenSinceYear
                    3                   354                   354
               Promo2          Promo2SinceWeek          Promo2SinceYear
                    0                   544                   544
        PromoInterval
                    0
```



Competition 관련 결측치의 경우, 정보가 부재하여 삭제

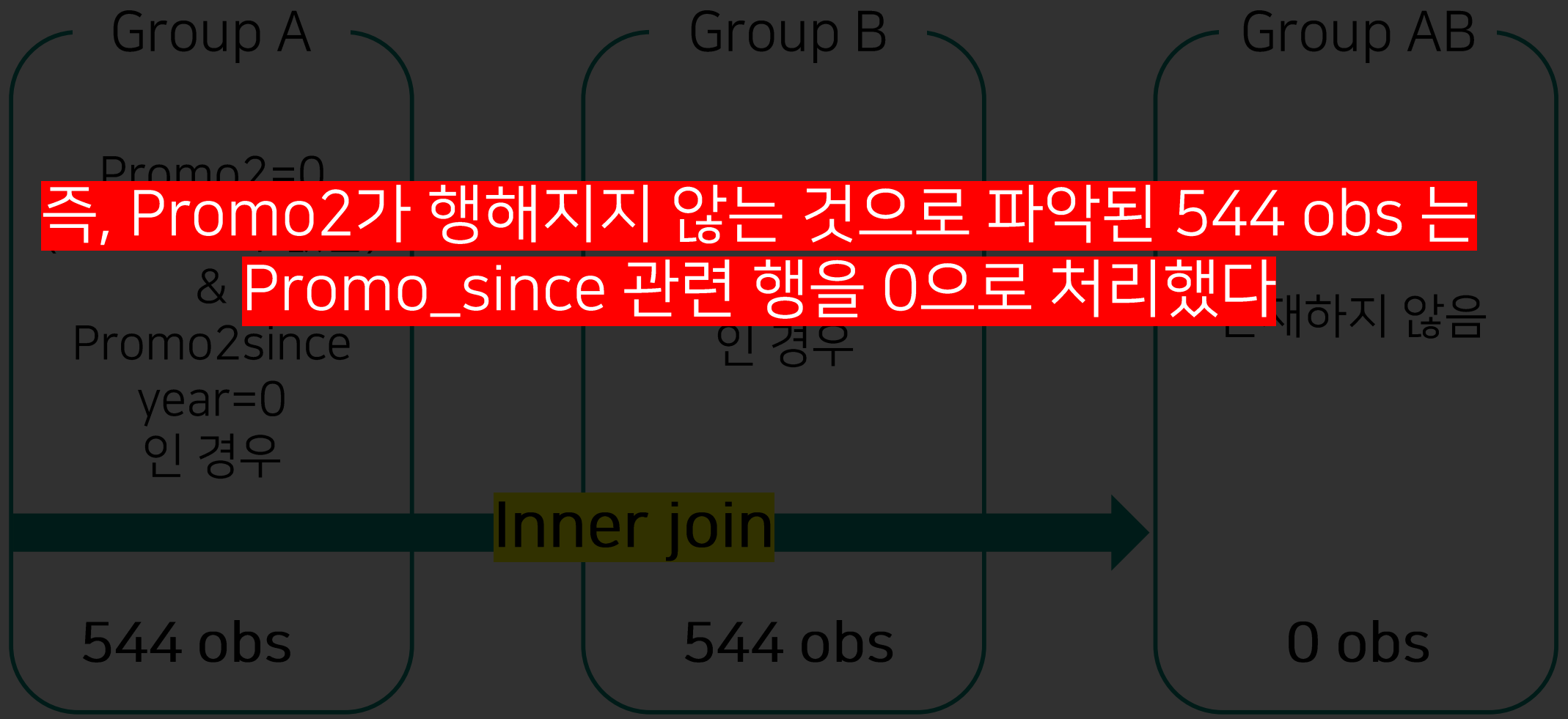
3 STORE SET 결측치 살펴보기

▶Promo2의 결측값: 0으로 처리



3 STORE SET 결측치 살펴보기

▶Promo2의 결측값: 0으로 처리



모델 선정

모델 선정과정

데이터셋을 본 후 고려한 모델은 크게 2가지로 나뉨

시계열	머신러닝
ARIMA SARIMA ARIMAX	Random Forest XGBoost AdaBoost

모델 선정과정

● 모델 고려 기준

= 데이터의 결측값과 변수들을 고려할 수 있는 모델 고려

시계열	머신러닝
ARIMA SARIMA ARIMAX	Random Forest XGBoost AdaBoost

모델 선정과정

1. 시계열

	결측치 처리	계절성	단일변수 / 다중변수
ARIMA	필요	고려할 순 있으나 복잡	단변량
SARIMA	필요	고려됨	단변량
ARIMAX	필요	고려할 순 있으나 복잡	다변량

ARIMA와 SARIMA의 경우 단일 변수만 고려할 수 있는 모델.

따라서 시계열 변수들을 추가로 고려할 수 있는 ARIMAX 모델을 고려해보았음.

모델 선정과정

1. 시계열 – ARIMA

Adf.test 귀무가설 기각 = kpss.test 귀무가설 유지
=> time-series data is stationary

store 1050의 경우 adf.test와 kpss.test 모두 stationary하다고 나왔음.

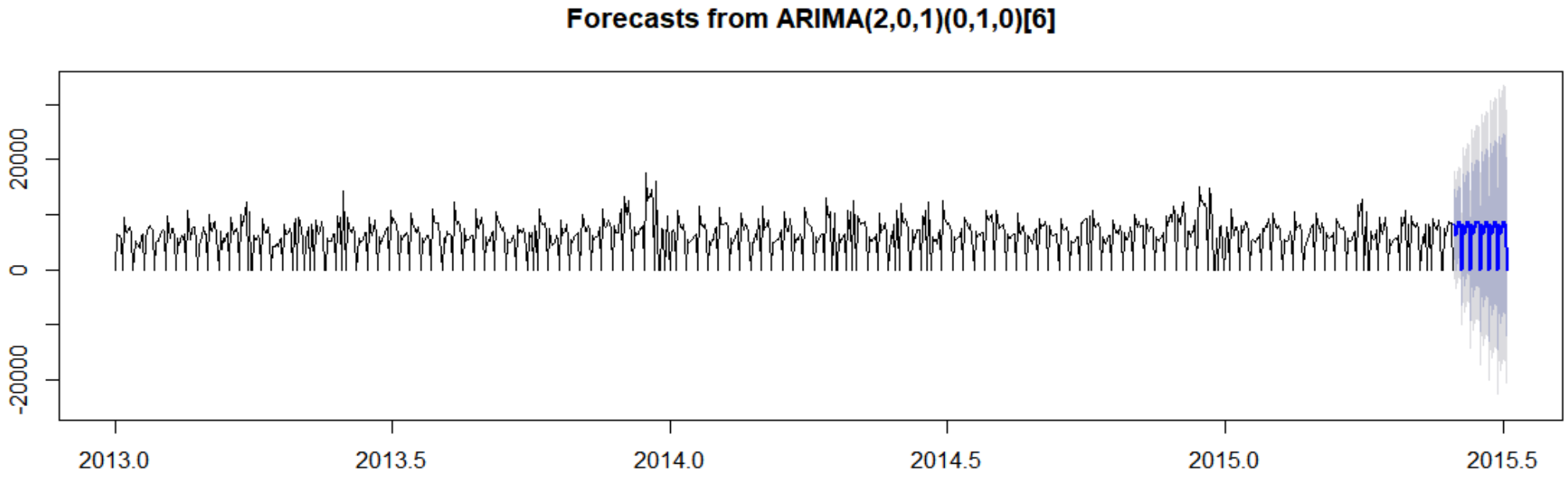
```
> adf.test(sales1050.ts)

        Augmented Dickey-Fuller Test

data:  sales1050.ts
Dickey-Fuller = -14.379, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(sales1050.ts) : p-value smaller than printed p-value
> kpss.test(sales1050.ts)

        KPSS Test for Level Stationarity

data:  sales1050.ts
KPSS Level = 0.39001, Truncation lag parameter = 6, p-value = 0.08146
```

모델 선정과정

1. 시계열 – ARIMA

무작위로 고른 다른 가게들도 결과가 비슷해서 ARIMA 모델에 계절성으로 고려해본 결과

예측 기간이 길어질 수록 예측의 정확성이 떨어지는 문제가 발생!



모델 선정과정

1. 시계열 – ARIMA

Adf.test 귀무가설 기각 = kpss.test 귀무가설 유지
=> time-series data is stationary

추가적으로, 모든 가게에 대해서 분석을 해야 하는데
개별 가게에 대해 예측을 해야 한다면
시간이 오래 걸리게 됨.

store 1050의 경우 acf.test, pkpss.test 모두 stationary 하다고 나왔음

```
> adf.test(sales1050.ts)

        Augmented Dickey-Fuller Test

alternative hypothesis: stationary

Warning message:
In adf.test(sales1050.ts) : p-value smaller than printed p-value
> kpss.test(sales1050.ts)

        KPSS Test for Level Stationarity

data:  sales1050.ts
KPSS Level = 0.39001, Truncation lag parameter = 6, p-value = 0.08146
```

⇒ **시계열 모델 전체 기각 (ARIMAX도 사용 X)**

모델 선정과정

2. 머신러닝

	결측치 처리	변수고려
Random Forest	필요	다변량
Xgboost	불필요	다변량
Adaboost	불필요	다변량

Xgboost와 Adaboost의 장점은 결측치를 처리할 필요가 없다는 점.

Random Forest는 Stacking을 위해 추가로 채택하였음.

모델 선정과정

2. 머신러닝

Random Forest는 결측치 처리가 필요한 모델
비록 모델에 내장된 imputation 기법이 존재하지만 앞서 말한대로
결측치를 채움으로 인해서 오히려 예측력이 떨어지는 결과가 발생할 수 있음.

따라서 오차가 비슷할 경우 Random Forest + Xgboost 모델 보다는
Xgboost 단일 모델을 선택할 예정.

다음주 예고

다음주 예고

1. 결측치 처리

a) Random Forest 모델

- 내장 Imputation 기법을 이용해 결측치 처리 후 예측

b) Xgboost 모델, Adaboost 모델

- 따로 결측치를 처리할 필요가 없음.

다음주 예고

2. Stacking + RMSPE 산출

- RandomForest, Xgboost 등의 RMSPE 산출
- RandomForest + Xgboost (Stacking) RMSPE 산출
- 모델 선정 후 결과 발표

다음주 예고

Train set 의 Frequency

			Assortment			
			a	b	c	
StoreType		a	323148	0	191757	514905
		b	6167	7745	881	14793
		c	66181	0	61631	127812
		d	105776	0	185908	291684
			501272	7745	440177	

Q & A