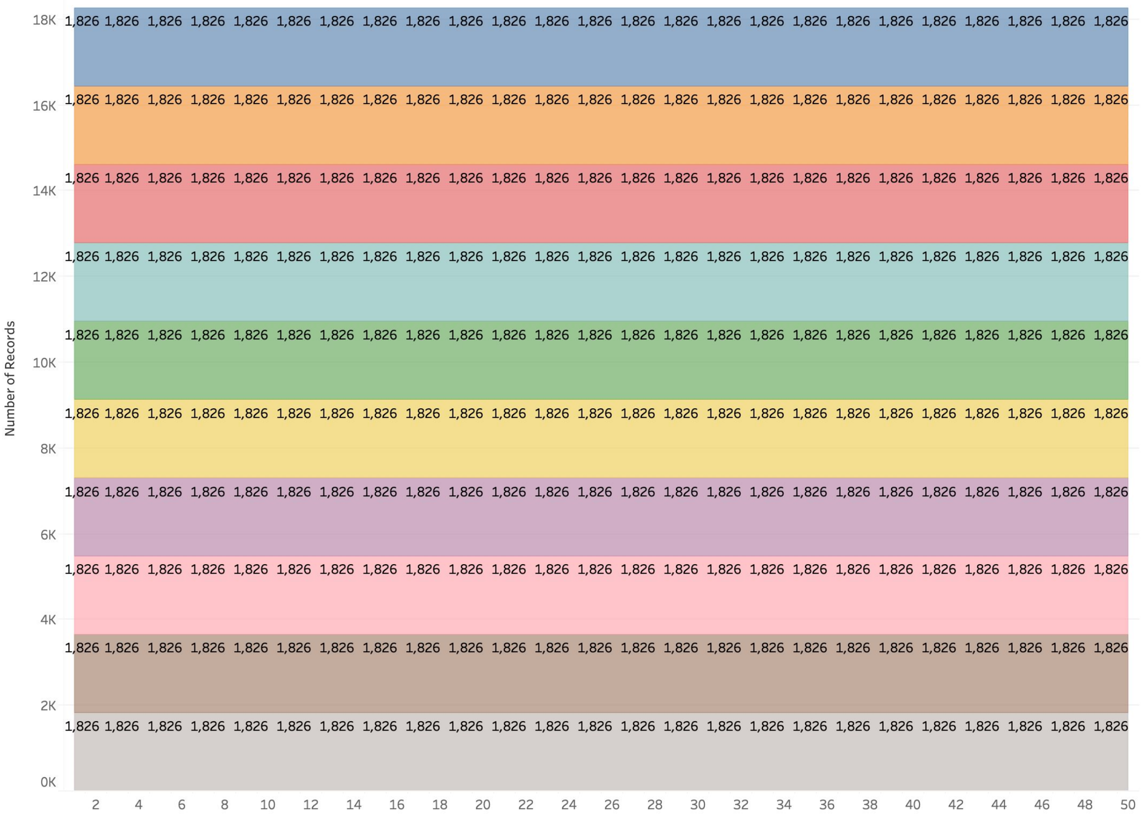
Time Series Analysis

with an application in Store Items Sales Forecasting

Kristen Li

Background

- Time range: 2013-01-01 -- 2017-12-31
- 10 stores, 50 items
- 1826 records per store per item
- 913,000 records in total.

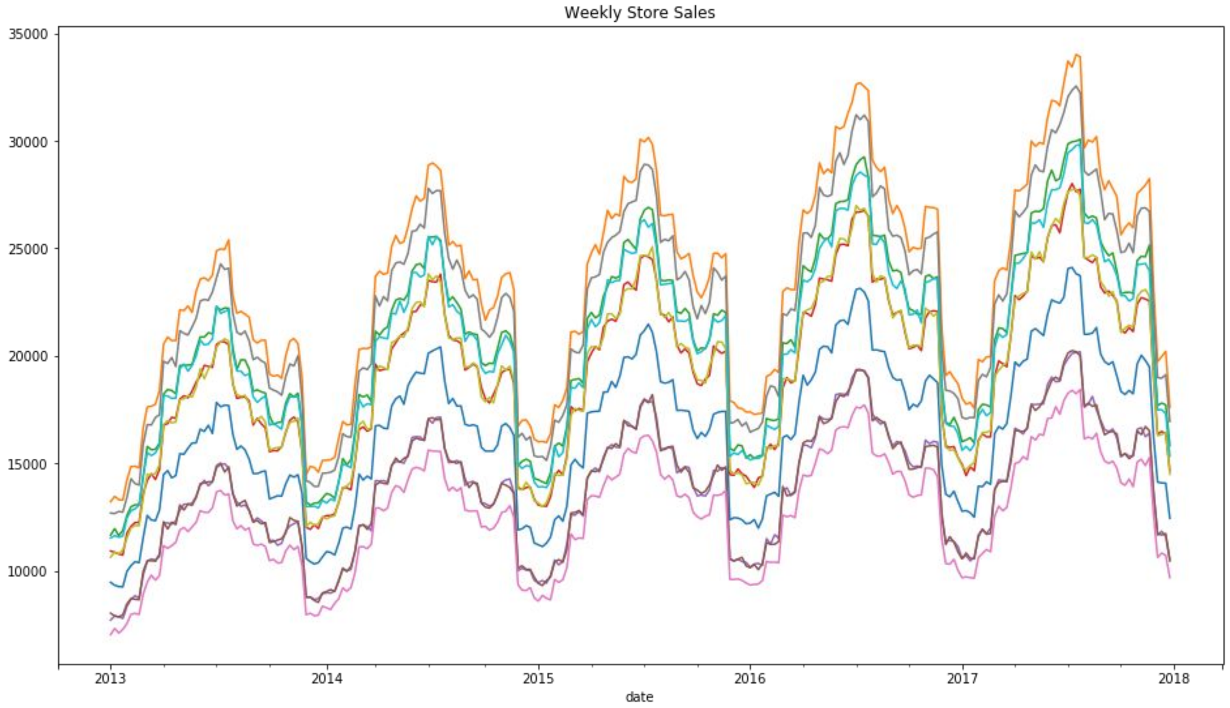


Research Question

What’s the time series pattern for each store and each item?

Descriptive Analysis

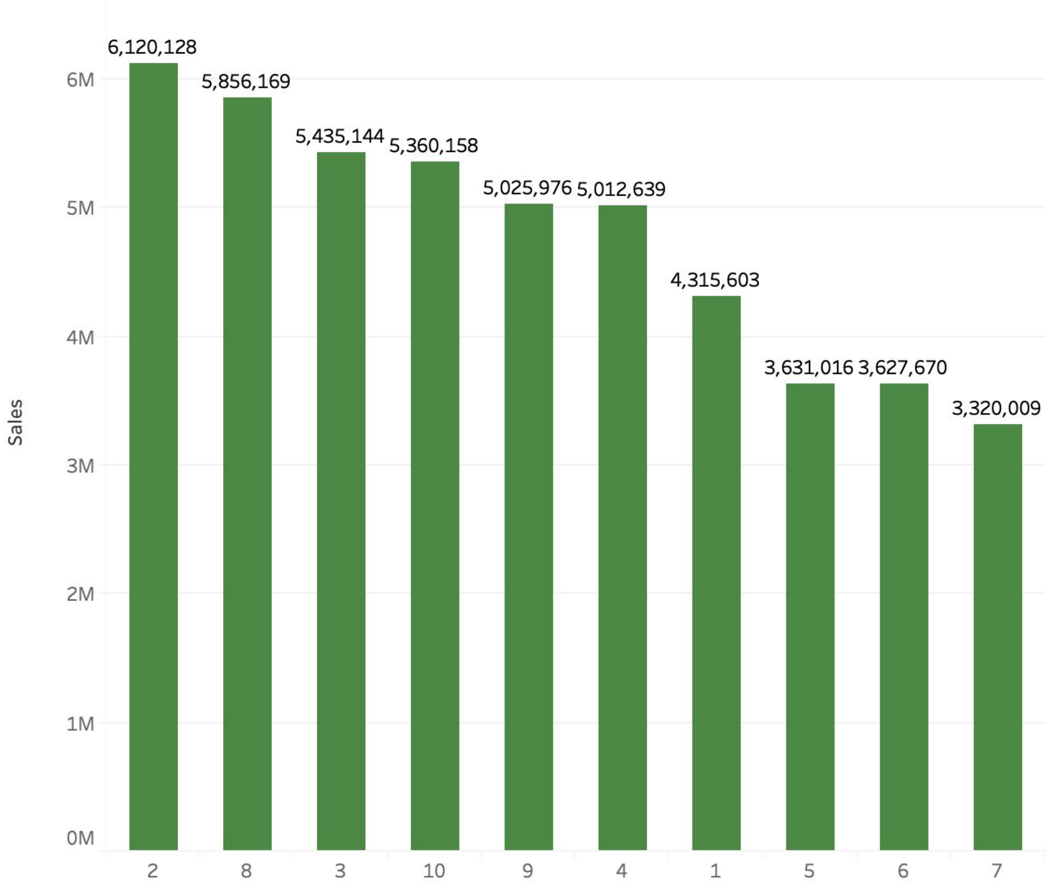
Time Series of each Store



Scale of sales of different stores are different

All 10 stores follow the very similar pattern of time series that they all indicate a **growing trend and seasonality**.

Sales by Store

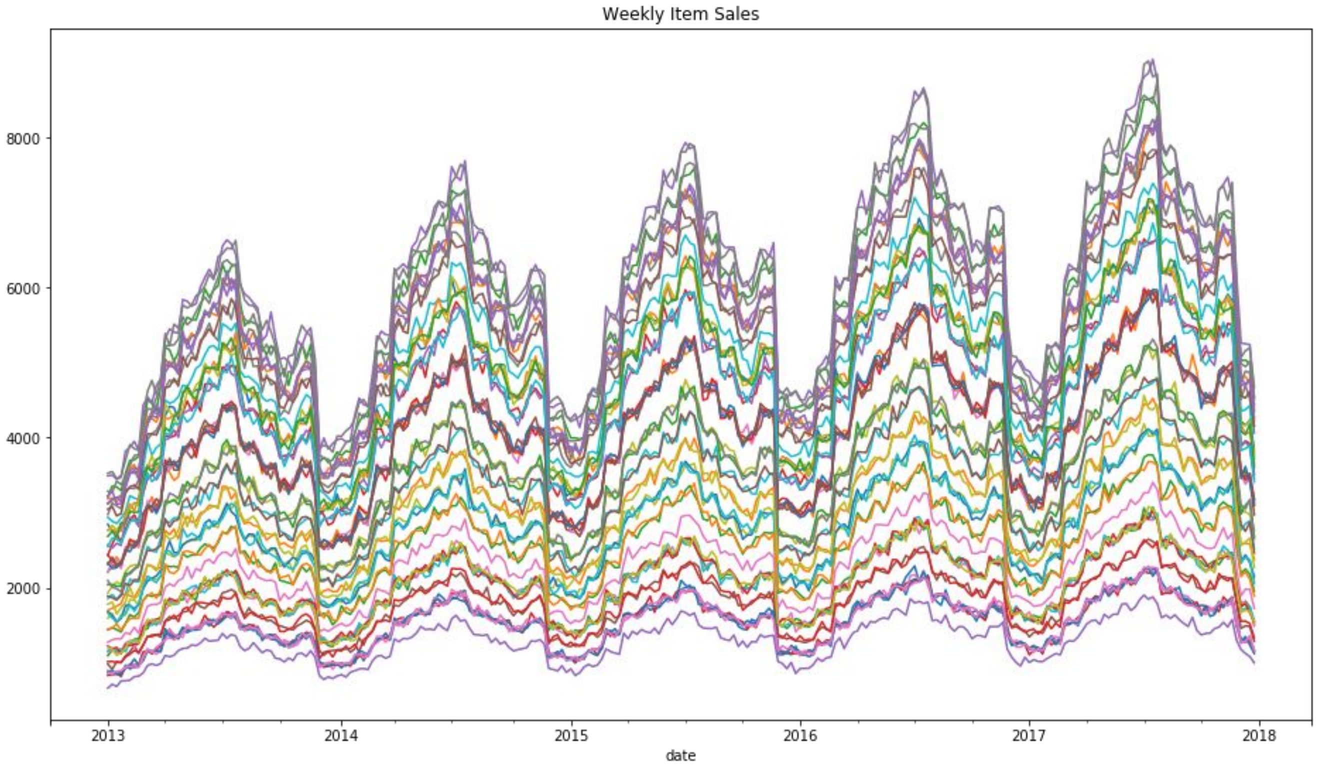


Discovery:

Store 2 has the highest sum of sales ($6,120,128) in 5 years;

Store 7 has the least highest sum of sales ($3,320,009).

Time Series of each item

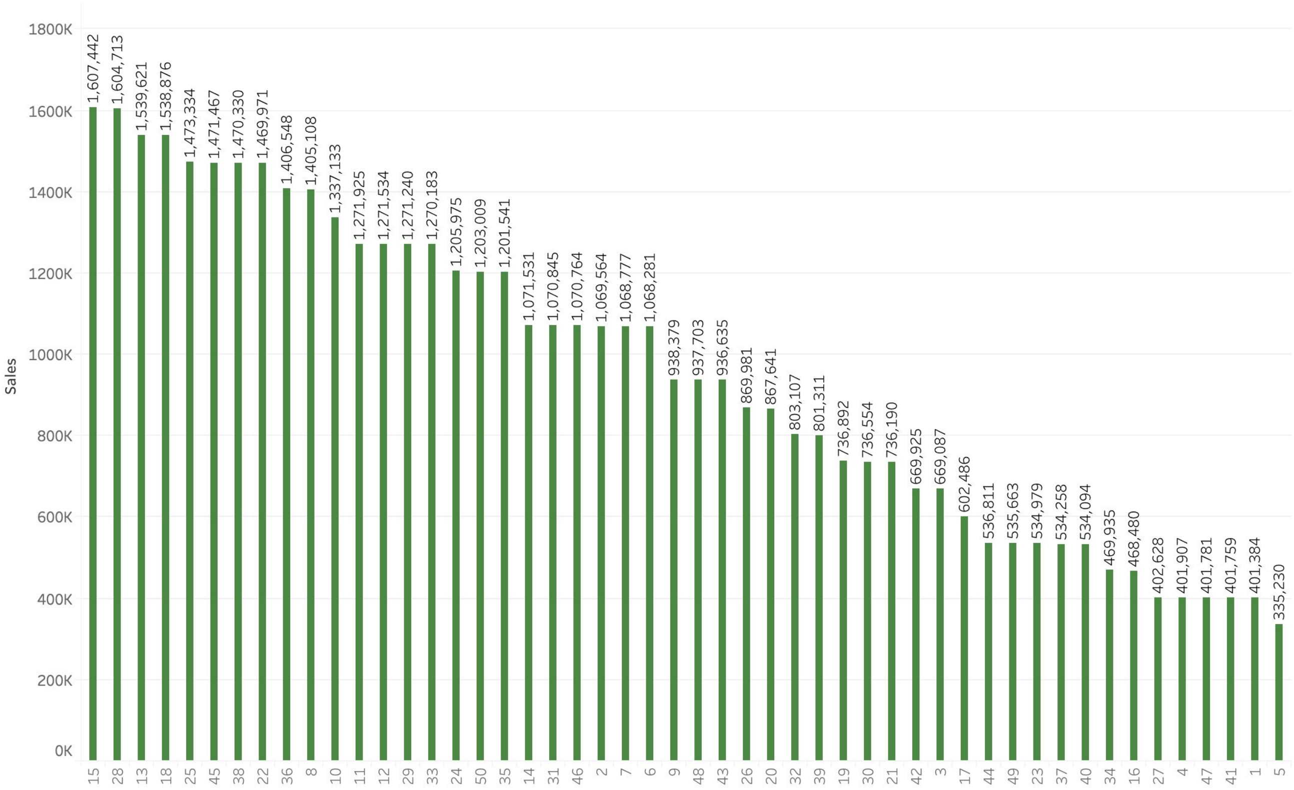


Scale of sales of different items are different

Not all 50 stores follow the same pattern of time series, but most of them indicate a **growing trend and seasonality**.

Some shows seasonality but no much trend.

Sales by Item



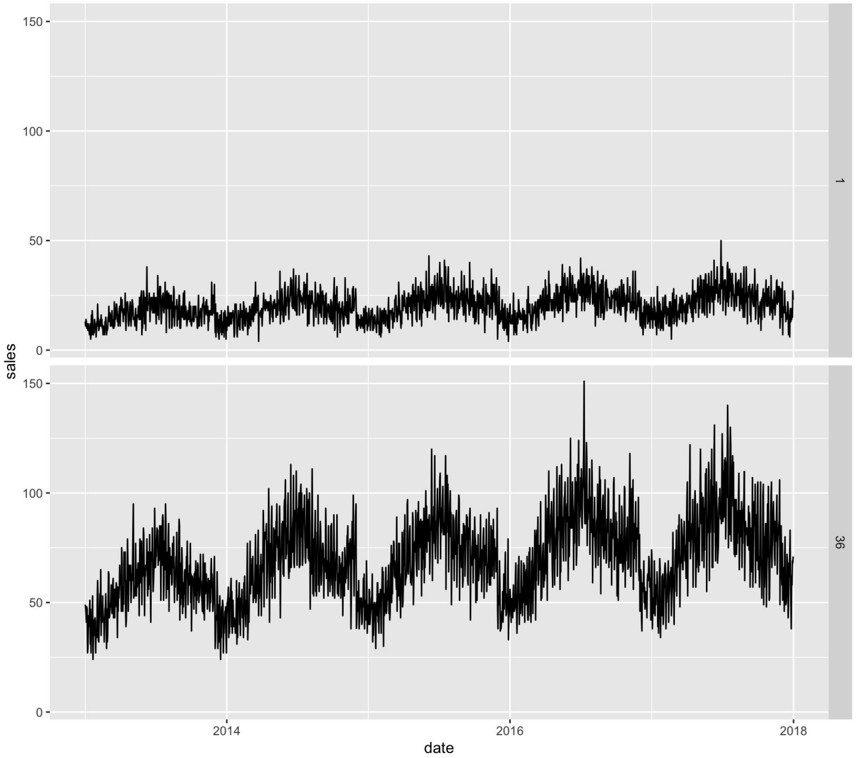
Discoveries:

Item 15 has the highest sum of sales ($1,607,442)

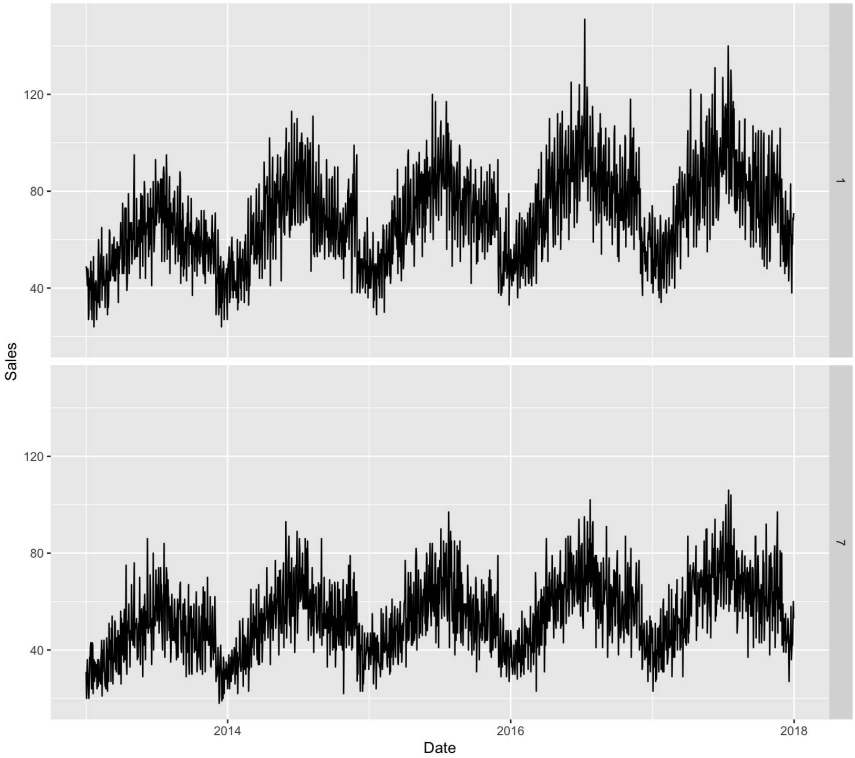
Item 5 has the lowest sum of sales ($335,230)

Different stores and items have completely different time series

Store 1: item 1, item 36



Item 36: store 1, store 7



Due to many reasons:

Store Type

Location

Customer Demographic

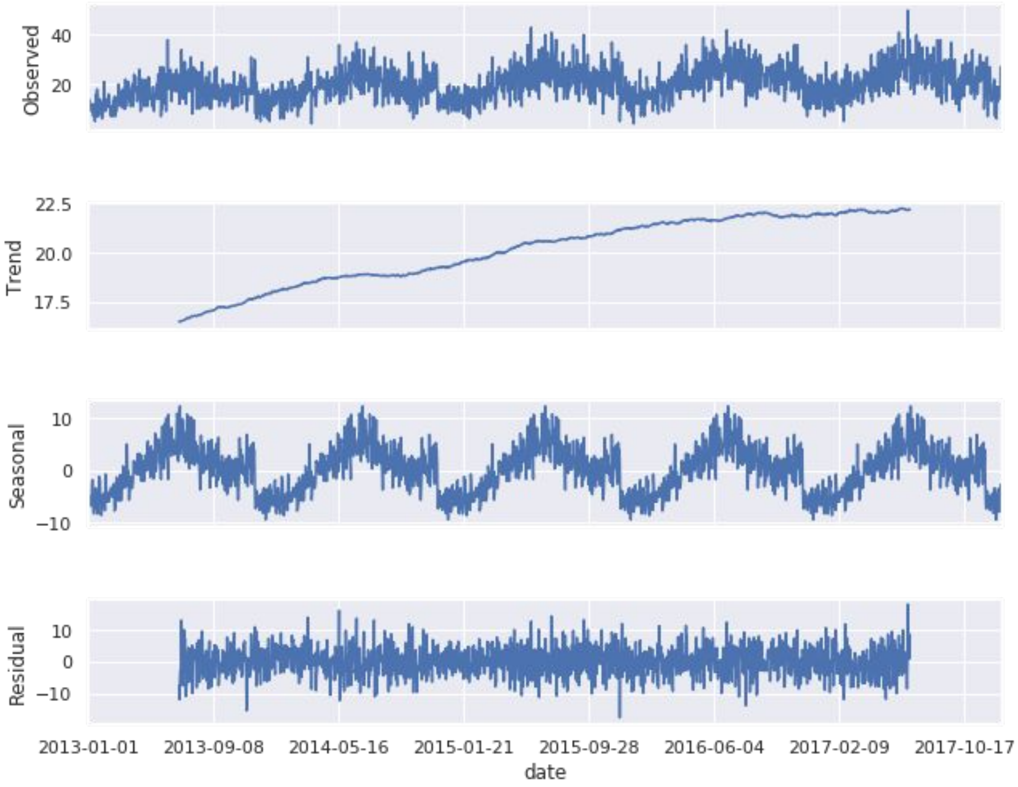
Economics

We can’t use a single model to generalize all stores’ and all items’ demands. Thus, **ideally, 50*10 = 500 models** at item in each store level need to be made for the most accurate prediction purpose.

Let's analyze Store 1, Item 1 as an example

ARIMA & SARIMA

Seasonal Decompose: upward trend and seasonal

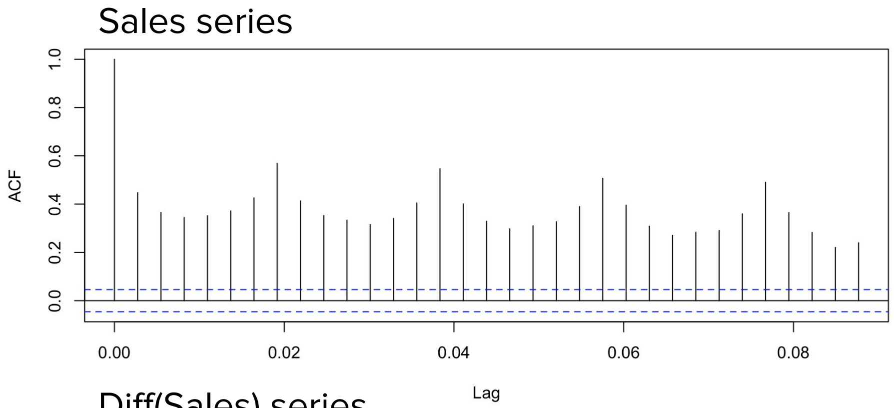


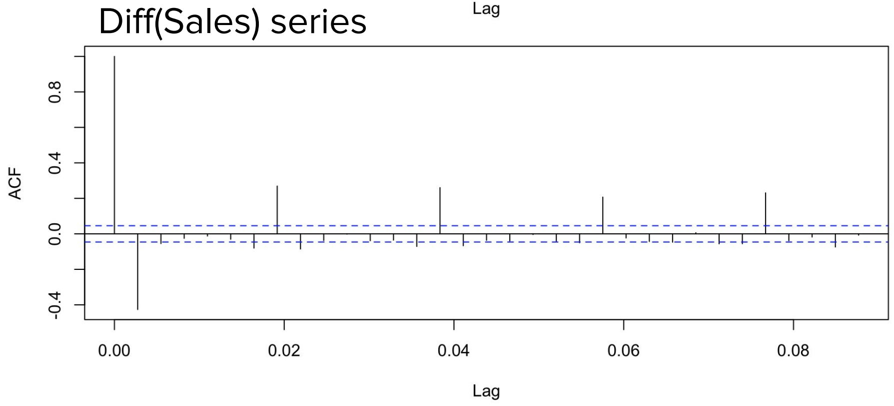
What can we notice from the charts?

- Non-stationarity (upward trend)
- Seasonality

Technically speaking, SARIMA model should be the best model for prediction.

Check Stationarity





ADF-test (Original-time-series)

P-value: 0.076 > 0.01

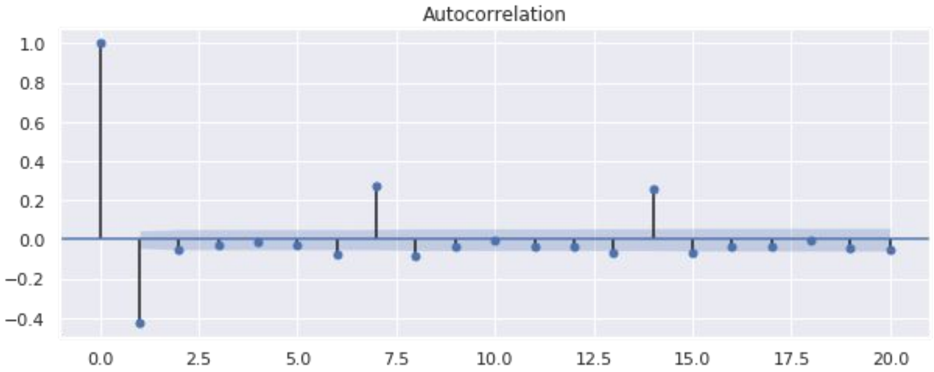
ADF-test (Differenced-time-series)

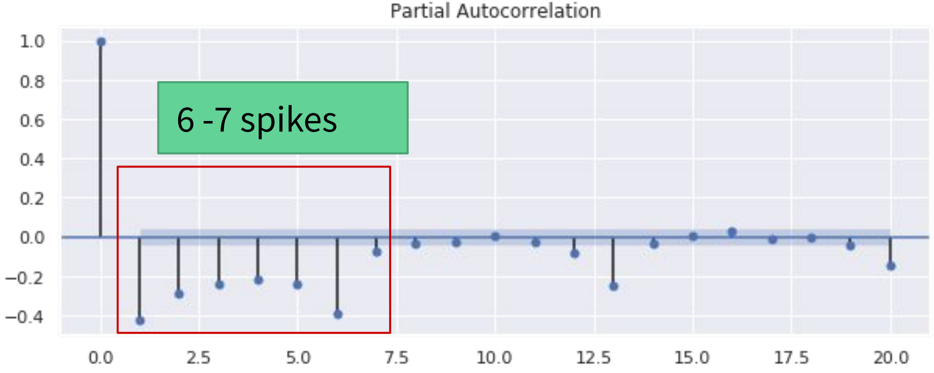
P-value: 1.211e-23 < 0.01

The ADF-test shows that the **original data is not stationary** because p-value is greater than 0.01, but **differenced data is stationary** because p-value is smaller than 0.01.

The result is consistent with the ACF, because for a stationary time series, the ACF will drop to zero relatively quickly (differenced), while the ACF of non-stationary data decreases slowly (sales).

ACF and PACF of the first differenced data & picking parameters





We use **differenced data,** because this time series is unit root process. Autocorrelogram & Partial Autocorrelogram is useful that to estimate each model's parameters.

Here we can see the acf and pacf both has a recurring pattern every 7 periods and are both exponentially decaying or sinusoidal.

From results,looks like ARIMA (p=6-7, d=1, q=?) model.

p=6-7: In the PACF, there are 6 significant spikes, and then no significant spikes thereafter.

d=1: The first order differencing make the ts stationary.

q=?: To avoid the potential for incorrectly specifying the MA order, I tried 0-7 all (see the table in the next slide).

Choosing parameters for ARIMA (p=7, d=1, q=7)

p=6

q	0	1	2	5
AIC	11209.36	11200.24	11199.85	10973.21
SSE	49256.02	48955.73	48891.21	42787.86

p=7

q	0	1	2	3	6	7
AIC	11202.3	11163.98	11165.98	11167.01	10935.74	10847.98
SSE	49011.13	47932.76	47932.76	47907	41844.4	39683.21

We got parameters (7,1,7).

ARIMA(7, 1, 7) Model Estimation

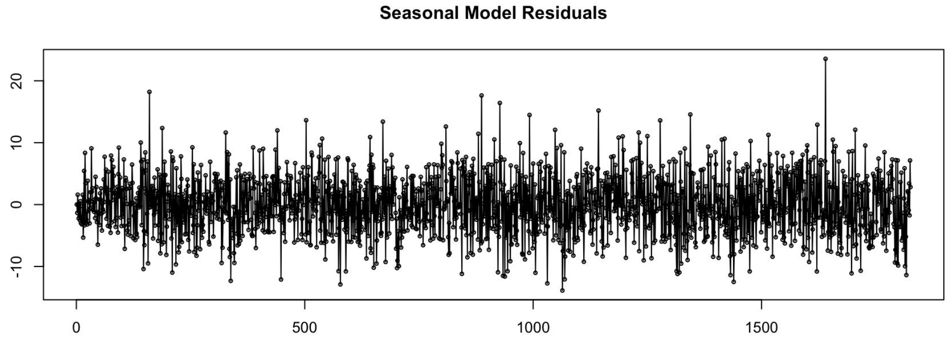
```
Call:
arima(x = store1_item1[, c("sales")], order = c(7, 1, 7))

Coefficients:
          ar1      ar2      ar3      ar4      ar5      ar6     ar7     ma1     ma2     ma3
ma4     ma5     ma6      ma7
      -0.9806  -0.9808  -0.9812  -0.9805  -0.9813  -0.9805  0.0185  0.0937  0.1096  0.1045
0.1035  0.1057  0.1036  -0.8843
s.e.   0.0266   0.0266   0.0265   0.0266   0.0265   0.0266  0.0265  0.0127  0.0123  0.0111
0.0125  0.0116  0.0115   0.0117

sigma^2 estimated as 21.74:  log likelihood = -5408.99,  aic = 10847.98
```

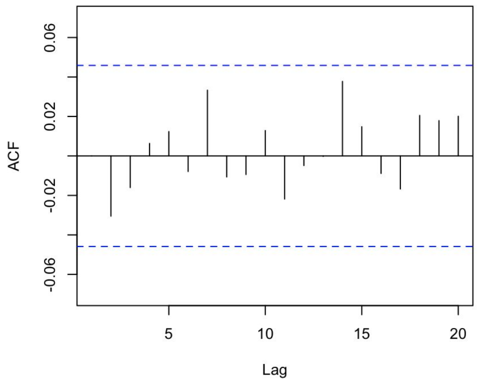
Equation:

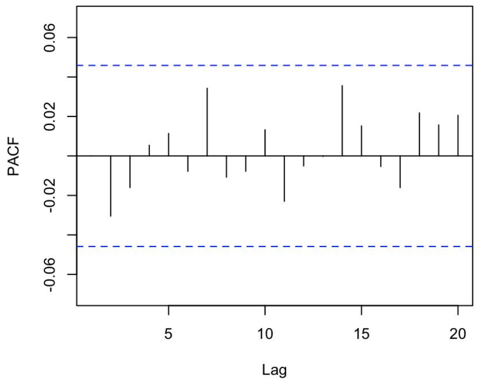
$$(1 + 0.9806B + 0.9808B^2 + 0.9812B^3 + 0.9805B^4 + 0.9805B^5 + 0.9813B^6 -0.0185B^7)\ (1-B)\ X_t =$$

$$(0.0937B + 0.1096B^2 + 0.1045B^3 + 0.1035\ B^4 + 0.1057B^5 + 0.1036B^6 - 0.8843B^7)\ Z_t$$

ARIMA(7, 1, 7) Model Estimation





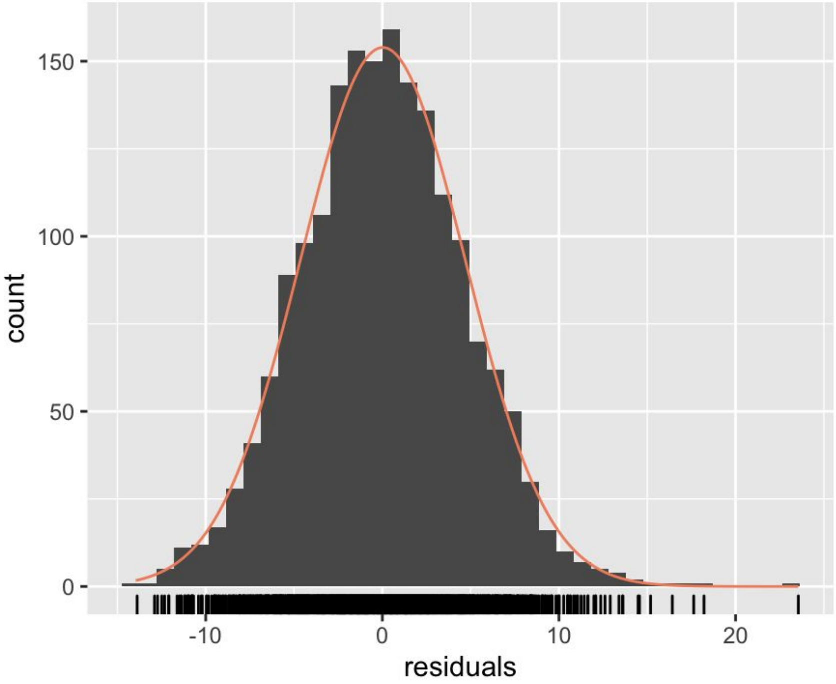


Model Equation:

$Y_t = 0.9806Y(t-1) - 0.9808Y(t-2) - 0.9812Y(t-3) - 0.9805Y(t-4) - 0.9805Y(t-5) - 0.9813Y(t-6) + 0.0185Y(t-7) + 0.0937\varepsilon(t-1) + 0.1096\varepsilon(t-2) + 0.1045\varepsilon(t-3) + 0.1035\,\varepsilon(t-4) + 0.1057\varepsilon(t-5) + 0.1036\varepsilon(t-6) - 0.8843\varepsilon(t-7) + \varepsilon_t$

Where **mean=0**, and **ε**t is white noise with a standard deviation of sqrt(21.74) = 4.66

Continuing: ARIMA(7, 1, 7) Model



Box-Pierce test

data: fit3$residuals

X-squared = 4.6593, df = 7.5099, p-value = 0.7511

Ljung-Box test

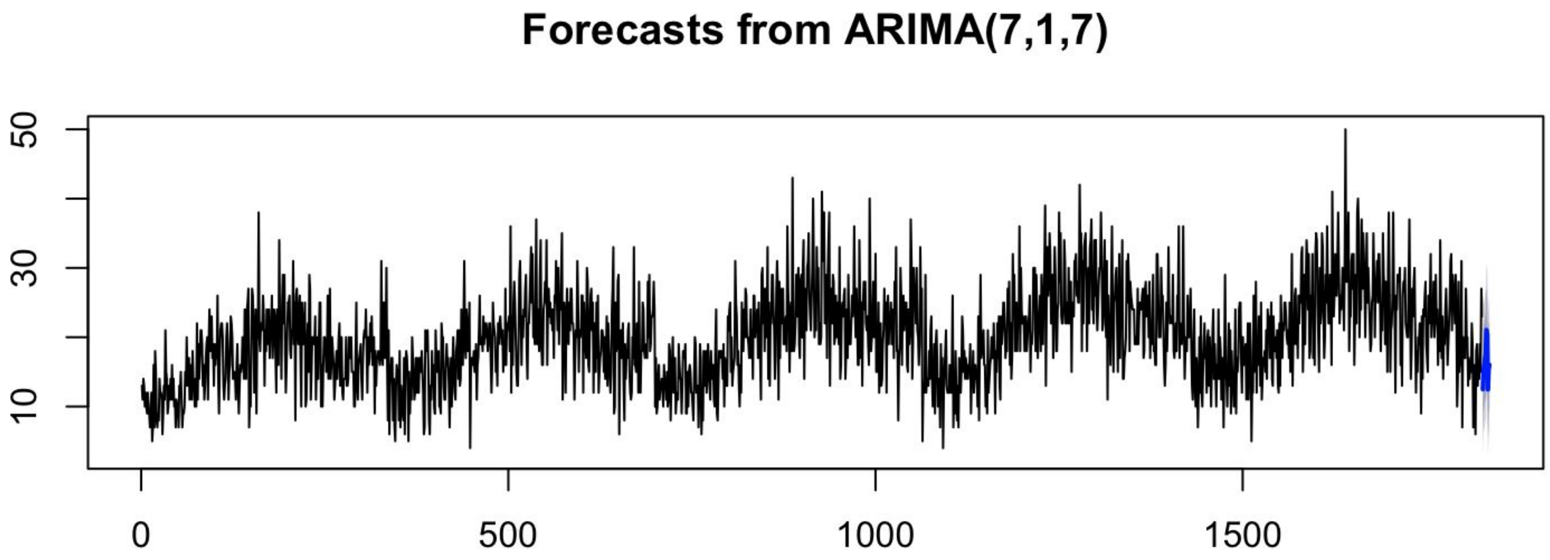
data: Residuals from ARIMA(7,1,7)

Q* = 9.9645, df = 3, p-value = 0.02

Model df: 14. Total lags used: 17

Although the graph looks very like a normal distribution, We see **a recurring correlation exists in both ACF and PACF**. So we need to deal with seasonality.

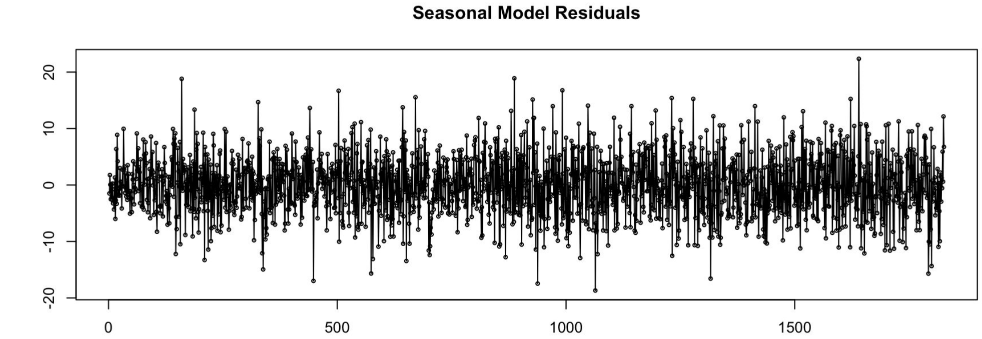
Forecasting from ARIMA (7, 1, 7)

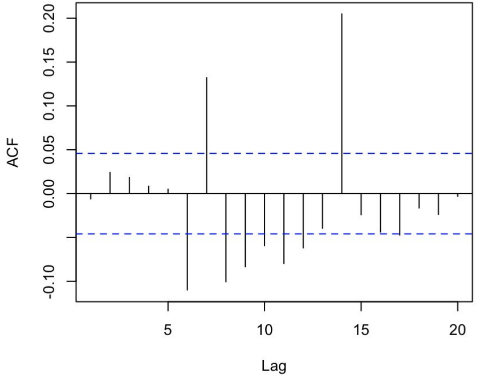


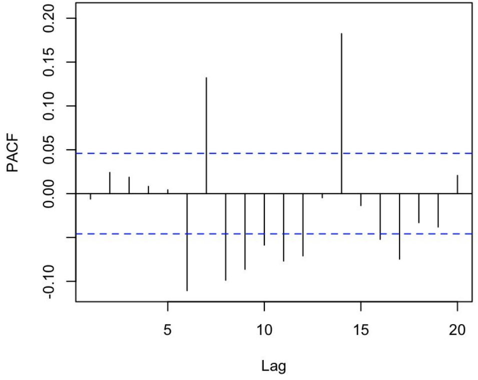
Forecasting from ARIMA (7, 1, 7)

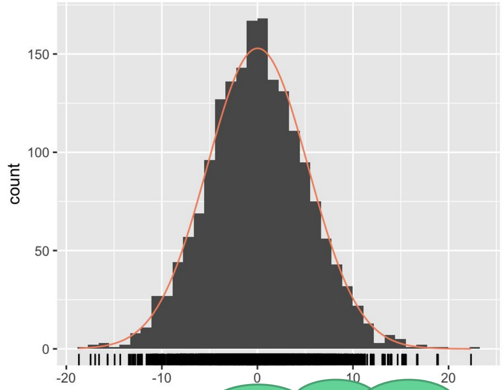
		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2018-01-01	1827	12.49981	6.499300	18.50031	3.322823	21.67679
2018-01-02	1828	15.37758	9.338677	21.41649	6.141874	24.61329
2018-01-03	1829	16.01177	9.935959	22.08758	6.719617	25.30393
2018-01-04	1830	16.88941	10.780465	22.99835	7.546587	26.23222
2018-01-05	1831	17.54160	11.399744	23.68346	8.148442	26.93476
2018-01-06	1832	21.03752	14.862346	27.21270	11.593405	30.48164
2018-01-07	1833	20.38737	14.179150	26.59558	10.892719	29.88202
2018-01-08	1834	12.45707	6.209114	18.70504	2.901644	22.01251
2018-01-09	1835	15.37206	9.097068	21.64705	5.775288	24.96883
2018-01-10	1836	16.01300	9.702686	22.32332	6.362206	25.66380

Auto algorithm result: ARIMA(5, 1, 2)









Shows autocorrelation

Box-Pierce test

data: fit$residuals

X-squared = 55.909, df = 7.5099, p-value = 1.736e-09

Ljung-Box test

data: Residuals from ARIMA(5,0,2) with zero mean

Q* = 93.97, df = 3, p-value < 2.2e-16

Model df: 7.   Total lags used: 10

Auto algorithm result: ARIMA(5, 1, 2)

```
Series: store1_item1[, c("sales")]
ARIMA(5,1,2)

Coefficients:
         ar1      ar2      ar3      ar4      ar5      ma1     ma2
      0.0549  -0.2068  -0.1894  -0.1661  -0.1541  -0.9263  0.1455
s.e.  0.0750   0.0407   0.0346   0.0321   0.0352   0.0728  0.0730

sigma^2 estimated as 27.98:  log likelihood=-5627.1
AIC=11270.2   AICc=11270.28   BIC=11314.28

Training set error measures:
                     ME     RMSE     MAE       MPE     MAPE      MASE         ACF1
Training set 0.02297043 5.278288 4.15019 -8.051611 24.49185 0.7527427 -0.006155832
```

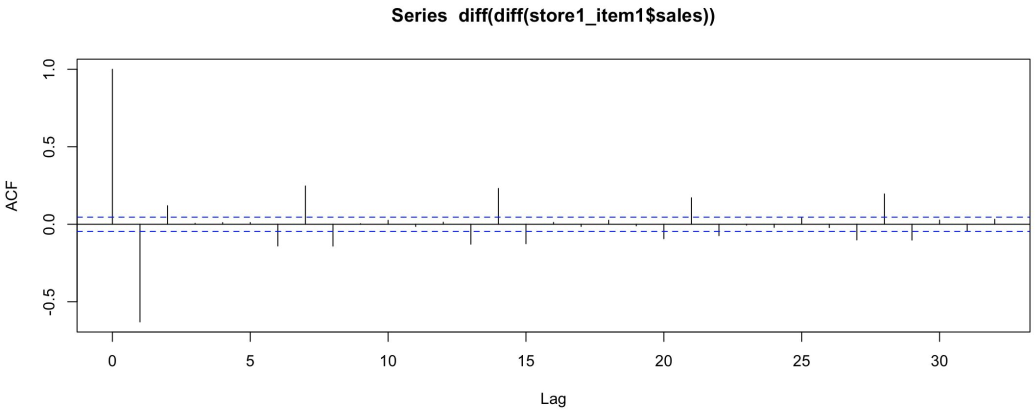
Equation:

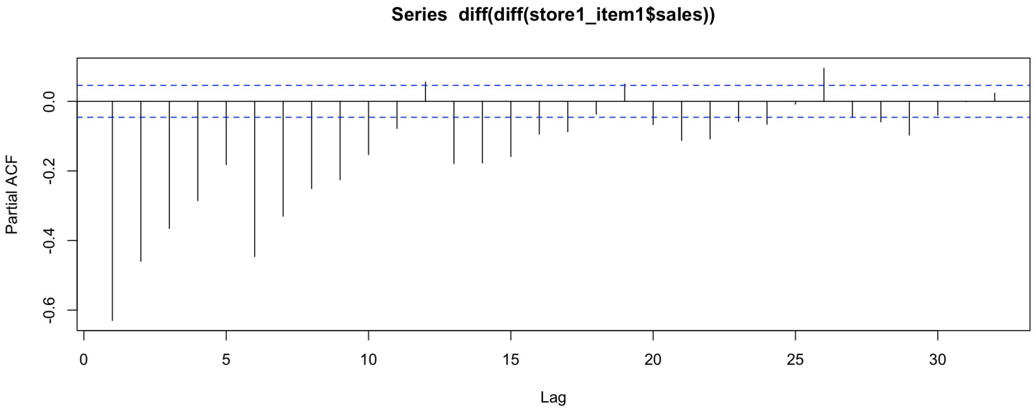
$$(1 - 0.0549B + 0.2068B^2 + 0.1894B^3 + 0.1661B^4 + 1541B^5)\,(1\text{-}B)\,X_t = (-0.9263B + 0.1455B^2)\,Z_t$$

ARIMA (7,1,7) is slightly better than auto arima ARIMA (5,1,2), however, ACF and PACF show recurring correlation existing in both models.

So we need to deal with seasonality.

SARIMA(p, d, q) (P, D, Q)m





Choosing parameters:

p=?: 6-7 non-seasonal spikes detected from the PACF.

q=1: one non-seasonal spike detected from the ACF

d=1, D=1

m=7: Here we can see the acf and pacf both has a recurring pattern every 7 periods.

Q=1: one seasonal spike at lag 7 in the ACF but no other significant spikes

P=0/1: There are several spikes, but I only use maximum one regular significant spike at lag 7 in the PACF after lag 7 -- to make the process reasonable.

Model Selection: SARIMA(p, d, q) (P, D, Q)m

```
0 1 1 0 1 1 7 AIC= 10798.11  SSE= 39581.15  p-VALUE= 0.7883058
0 1 1 0 1 2 7 AIC= 10797.93  SSE= 39513.69  p-VALUE= 0.9523192
0 1 1 1 1 0 7 AIC= 11356.21  SSE= 54463.26  p-VALUE= 1.161551e-09
0 1 1 1 1 1 7 AIC= 10797.74  SSE= 39506.51  p-VALUE= 0.9509269
0 1 1 1 1 2 7 AIC= 10801.94  SSE= 39579.37  p-VALUE= 0.7073274
1 1 1 0 1 1 7 AIC= 10800  SSE= 39579.47  p-VALUE= 0.8054869
1 1 1 0 1 2 7 AIC= 10799.65  SSE= 39508.84  p-VALUE= 0.9725113
1 1 1 1 1 0 7 AIC= 11348.23  SSE= 54169.08  p-VALUE= 2.669266e-07
1 1 1 1 1 1 7 AIC= 10799.45  SSE= 39499.22  p-VALUE= 0.9720391
2 1 1 0 1 1 7 AIC= 10800.56  SSE= 39547.78  p-VALUE= 0.925624
2 1 1 1 1 0 7 AIC= 11348.72  SSE= 54125.39  p-VALUE= 3.812788e-07
...
6 1 0 0 1 1 7 AIC= 10882.69  SSE= 41093.79  p-VALUE= 2.229754e-08
6 1 0 1 1 1 7 AIC= 10819.9  SSE= 39645.75  p-VALUE= 0.5128233
6 1 1 0 1 1 7 AIC= 10807.11  SSE= 39511.33  p-VALUE= 0.9961842
6 1 1 1 1 1 7 AIC= 10807.36  SSE= 39455.17  p-VALUE= 1
7 1 0 0 1 1 7 AIC= 10872.18  SSE= 40848.09  p-VALUE= 6.84914e-05
7 1 0 1 1 1 7 AIC= 10821.17  SSE= 39627.18  p-VALUE= 0.6559436
7 1 1 0 1 1 7 AIC= 10807.4  SSE= 39455.94  p-VALUE= 1
7 1 1 1 1 1 7 AIC= 10807  SSE= 39367.59  p-VALUE= 0.9999993
```

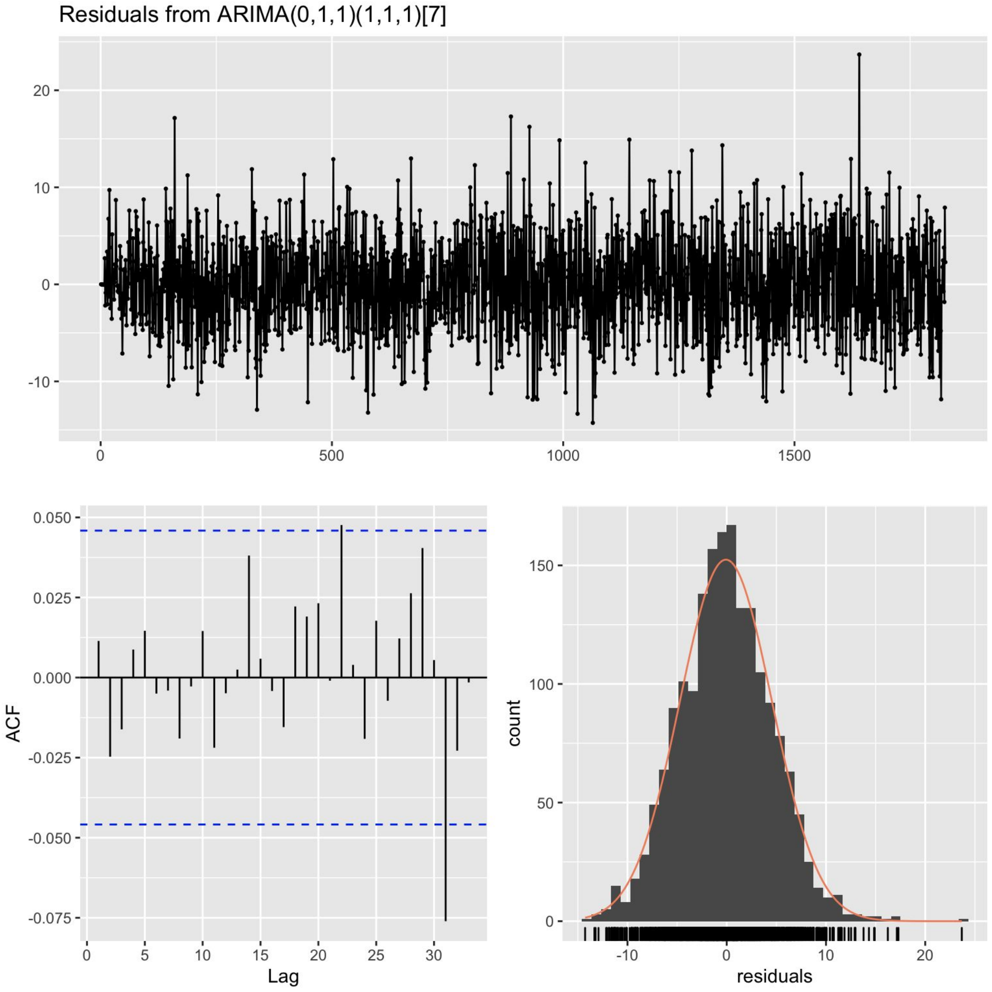
Model: SARIMA(0, 1, 1) (1, 1, 1)7

SARIMA(0, 1, 1) (1, 1, 1)7

```
Call:
arima(x = store1_item1[, c("sales")], order = c(0, 1, 1), seasonal = list(order = c(1, 
    1, 1), period = 7))

Coefficients:
          ma1    sar1     sma1
      -0.8952  0.0375  -0.9944
s.e.   0.0102  0.0244   0.0080

sigma^2 estimated as 21.73:  log likelihood = -5394.87,  aic = 10797.74

Training set error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set -0.08264037 4.651403 3.664003 -6.908788 21.47542 0.6645603 0.0114268
```

Equation:

$$(1-B)\,(1-0.0375B^7)\,(1-B^7)\,X_t = (1-0.8952B)\,(1+0.9944B^7)\,Z_t$$



SARIMA(0, 1, 1) (1, 1, 1)7

Box-Pierce test

data: sarima$residuals

X-squared = 2.4357, df = 7.5099, p-value = 0.9509

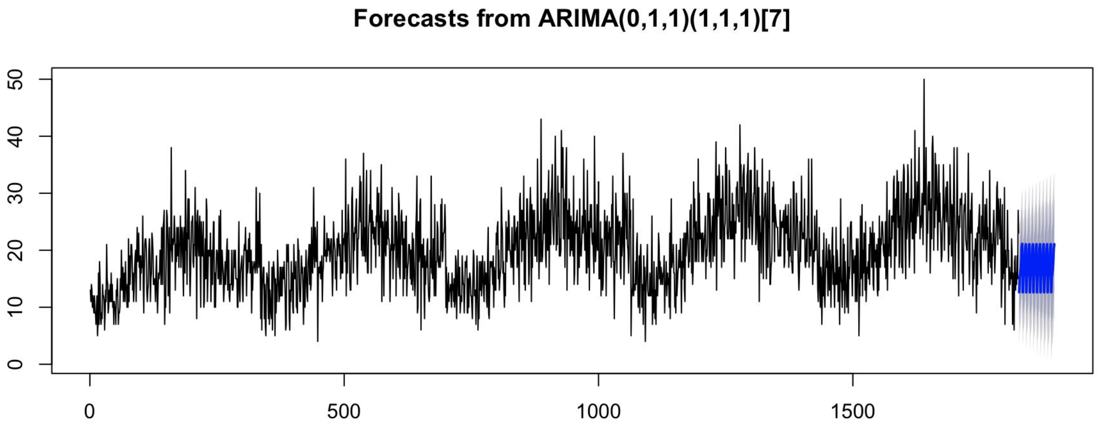
Ljung-Box test

data: Residuals from ARIMA(0,1,1)(1,1,1)[7]

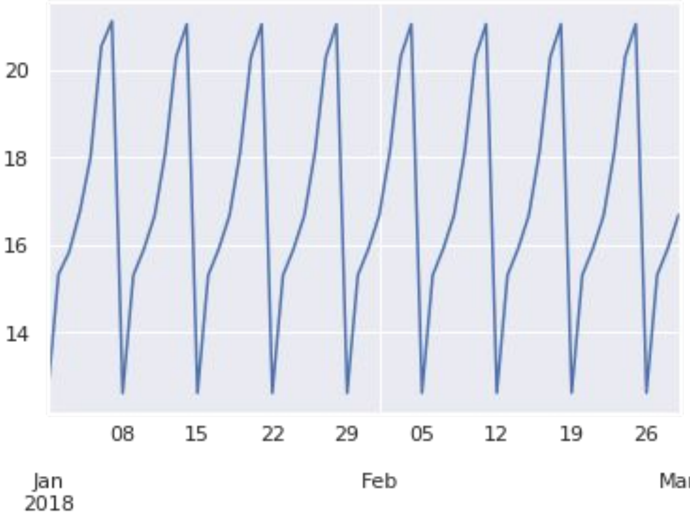
Q* = 3.5087, df = 7, p-value = 0.8343

Model df: 3.   Total lags used: 10

Forecasting from SARIMA(0, 1, 1) (1, 1, 1)7



2-month prediction



Forecasting from SARIMA(0, 1, 1) (1, 1, 1)7

		Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
2018-01-01	1827	12.63281	6.656801	18.60882	3.493292	21.77233
2018-01-02	1828	15.33444	9.325720	21.34317	6.144894	24.52399
2018-01-03	1829	15.84750	9.806230	21.88877	6.608175	25.08682
2018-01-04	1830	16.76571	10.692067	22.83934	7.476876	26.05454
2018-01-05	1831	18.01635	11.910514	24.12219	8.678278	27.35443
2018-01-06	1832	20.54045	14.402582	26.67832	11.153391	29.92751
2018-01-07	1833	21.11681	14.947075	27.28654	11.681016	30.55260
2018-01-08	1834	12.62033	6.387080	18.85357	3.087398	22.15326
2018-01-09	1835	15.31077	9.043357	21.57819	5.725587	24.89596
2018-01-10	1836	15.91804	9.616636	22.21945	6.280873	25.55521
2018-01-11	1837	16.68322	10.348011	23.01844	6.994351	26.37210
2018-01-12	1838	18.13072	11.761876	24.49956	8.390415	27.87102
2018-01-13	1839	20.29957	13.897273	26.70186	10.508104	30.09103
2018-01-14	1840	21.04749	14.611920	27.48306	11.205135	30.88984

Model Selection

Model	ARIMA (7, 1, 7)	ARIMA (5, 1, 2)	SARIMA(0, 1, 1) (1, 1, 1)7
AIC	10847.98	11270.2	10797.74
SSE	39683.21	50872.95	39506.51
Box-Pierce test X-Squared	4.6593	4.6593	2.4357
P-value	0.7511	1.736e-09	0.9509269
Ljung-Box Q-statistics	9.9645	93.97	3.5087
Ljung-Box Q-statistics p-value	0.02	2.2e-16	0.8343

```
library(dplyr)
library(data.table)
library(ggplot2)
library(tseries)
library(forecast)

# This is 5 years of store-item sales data for 50 different
items at 10 different stores.
store <- fread("~/Downloads/store_demand.csv")
head(store)
store$date <- as.Date(store$date, "%m/%d/%Y")
range(store$date)
rownames(store) <- store$Date

ggplot(store, aes(date, sales)) + geom_line() +
  scale_x_date('time')+ ylab("Daily Sales") + xlab("") +
  facet_grid(store$store)

store1_item36 <- store %>% filter(store==1&item==36)
store1_2items <- rbind(store1_item1,store1_item36)

range(store1_item36$date)
ggplot(store1_2items, aes(date, sales)) + geom_line() +
  facet_grid(store1_2items$item) +
  ylab("Daily Sales") + xlab("")

store7_item36 <- store %>% filter(store==7&item==36)
item36_2stores <- rbind(store1_item36,store7_item36)

ggplot(item36_2stores, aes(date, sales)) + geom_line() +
  facet_grid(item36_2stores$store) +
  ylab("Sales") + xlab("Date")

##### model for store 1 item 1 #####
store1_item1 <- store %>% filter(store==1&item==1)
rownames(store1_item1) <- store1_item1[,"date"]
store1_item1 <- ts(store1_item1)

cbind("Sales" = store1_item1[, "sales"],
      "Monthly log sales" = log(store1_item1[, "sales"]),
      "Annual change in log sales" = diff(store1_item1[,
"sales"],12)) %>%
  autoplot(facets=TRUE) +
  xlab("Year") + ylab("") +
  ggtitle("store 1 item 1 sales")
```

```
plot(diff(store1_item1), main='Differenced Log-transorm of
sales', ylab='', col='brown', lwd=3)

ggplot(store1_item1[,c("date", "sales")], aes(date, sales)) +
geom_line() + ylab("Sales") + xlab("Day")

# ACF plots
acf(store1_item1$sales)
pacf(store1_item1$sales)

acf(diff(diff(store1_item1$sales)))
pacf(diff(diff(store1_item1$sales)))

# ARIMA
fit <- auto.arima(store1_item1[,"sales"], seasonal=FALSE)
summary(fit)
fit2 <- auto.arima(store1_item1[,c("sales")],
seasonal=FALSE,stepwise=FALSE, approximation=FALSE)
summary(fit2)
fit3 <- Arima(store1_item1[,c("sales")], order=c(7,1,7))
summary(fit3)

AIC( arima( store1_item1[,c("sales")], order=c(6,1,0) ) ) #AIC =
[1] 11209.36
AIC( arima( store1_item1[,c("sales")], order=c(6,1,1) ) ) #AIC =
[1] 11200.24
AIC( arima( store1_item1[,c("sales")], order=c(6,1,2) ) ) #AIC =
[1] 11199.85
AIC( arima( store1_item1[,c("sales")], order=c(6,1,5) ) ) #AIC =
[1] 10973.21
AIC( arima( store1_item1[,c("sales")], order=c(7,1,0) ) ) #AIC =
[1] 11202.3
AIC( arima( store1_item1[,c("sales")], order=c(7,1,1) ) ) #AIC =
[1] 11163.98
AIC( arima( store1_item1[,c("sales")], order=c(7,1,2) ) ) #AIC =
[1] 11165.98
AIC( arima( store1_item1[,c("sales")], order=c(7,1,3) ) ) #AIC =
[1] 11167.01
AIC( arima( store1_item1[,c("sales")], order=c(7,1,6) ) ) #AIC =
[1] 10935.74
AIC( arima( store1_item1[,c("sales")], order=c(7,1,7) ) ) #AIC =
[1] 10847.98

sum(arima( store1_item1[,c("sales")],
order=c(6,1,5) )$residuals^2)
sum(fit$residuals^2)
Box.test(fit$residuals, lag=log(length(fit$residuals)))
```

```
Box.test(fit3$residuals, lag=log(length(fit3$residuals)))

tsdisplay(residuals(fit), lag.max=20, main='Seasonal Model
Residuals')
checkresiduals(fit)
tsdisplay(residuals(fit3), lag.max=20, main='Seasonal Model
Residuals')
checkresiduals(fit3)

pred = forecast(fit3)
plot(forecast(fit3))

# SARIMA Model
d=1
DD=1
per=7

for(p in 1:3){
  for(q in 1:3){
    for(i in 1:3){
      for(j in 1:3){
        if(p+d+q+i+DD+j<=10){
          model<-arima(x=store1_item1[,c("sales")], order =
c((p-1),d,(q-1)), seasonal = list(order=c((i-1),DD,(j-1)),
period=per))
          pval<-Box.test(model$residuals,
lag=log(length(model$residuals)))
          sse<-sum(model$residuals^2)
          cat(p-1,d,q-1,i-1,DD,j-1,per, 'AIC=', model$aic, '
SSE=',sse,' p-VALUE=', pval$p.value,'\n')
        }
      }
    }
  }
}

## Final model
sarima <-arima(store1_item1[,c("sales")], order = c(0,1,1),
seasonal = list(order = c(1,1,1), period = 7))
summary(sarima)
z.test(sarima)

checkresiduals(sarima)
Box.test(sarima$residuals, lag=log(length(sarima2$residuals)))
pred = forecast(sarima)
plot(forecast(sarima, h=14))
```

```
sarima2 <-arima(store1_item1[,c("sales")], order = c(6,1,1),
seasonal = list(order = c(1,1,1), period = 7))
summary(sarima2)
tsdisplay(residuals(sarima2), lag.max=20, main='Seasonal Model
Residuals')
checkresiduals(sarima2)
Box.test(sarima2$residuals, lag=log(length(sarima2$residuals)))

pred = forecast(sarima2)
plot(forecast(sarima2, h=70))
```

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(font='IPAGothic')
import numpy as np
import statsmodels.api as sm
```

In [2]:
```
df = pd.read_csv("~/Downloads/store.csv")
```

In [3]:
```
df = df.drop(columns=['Unnamed: 0'])
```

In [4]:
```
df = df.set_index('date')
```

In [5]:
```
df.head()
```

Out[5]:

	store	item	sales
date			
2013-01-01	1	1	13
2013-01-02	1	1	11
2013-01-03	1	1	14
2013-01-04	1	1	13
2013-01-05	1	1	10

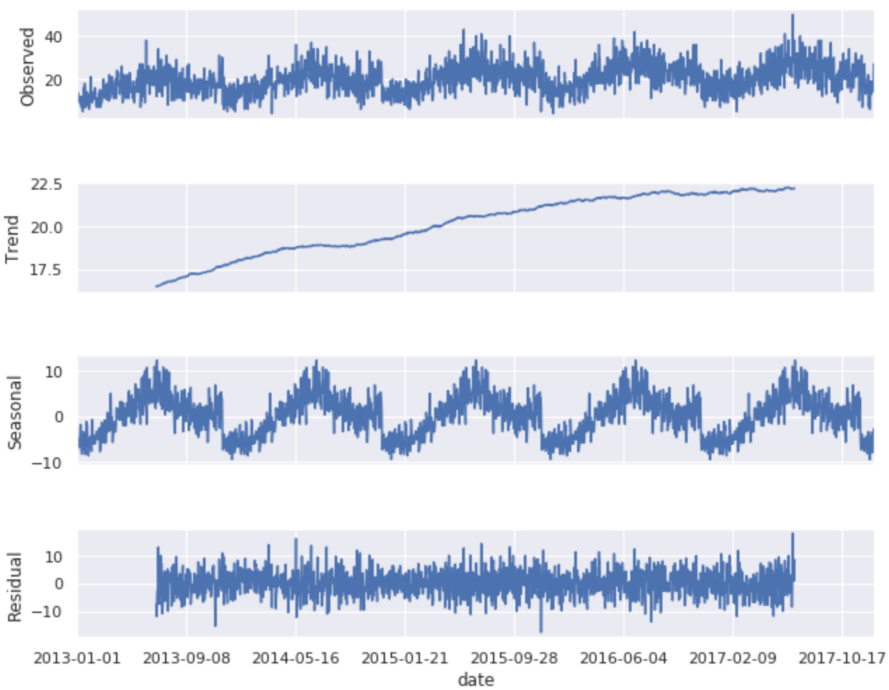
In [6]:
```
buf = df[(df.store==1)&(df.item==1)].copy()
buf.head()
```

Out[6]:

	store	item	sales
date			
2013-01-01	1	1	13
2013-01-02	1	1	11
2013-01-03	1	1	14
2013-01-04	1	1	13
2013-01-05	1	1	10

In [8]:
```
res = sm.tsa.seasonal_decompose(buf.sales.dropna(), freq=365)
fig = res.plot()
fig.set_figheight(8)
fig.set_figwidth(10)
plt.show()
```

```
/anaconda3/lib/python3.7/site-packages/matplotlib/font_manager.py:1241: UserWarning: findfont: Font family
['IPAGothic'] not found. Falling back to DejaVu Sans.
  (prop.get_family(), self.defaultFamily[fontext]))
/anaconda3/lib/python3.7/site-packages/matplotlib/font_manager.py:1241: UserWarning: findfont: Font family
['IPAGothic'] not found. Falling back to DejaVu Sans.
  (prop.get_family(), self.defaultFamily[fontext]))
```



In [9]:
```
#ADF-test(Original-time-series)
res = sm.tsa.adfuller(buf['sales'].dropna(),regression='ct')
print('p-value:{}'.format(res[1]))
```

```
p-value:0.07610688992415375
```

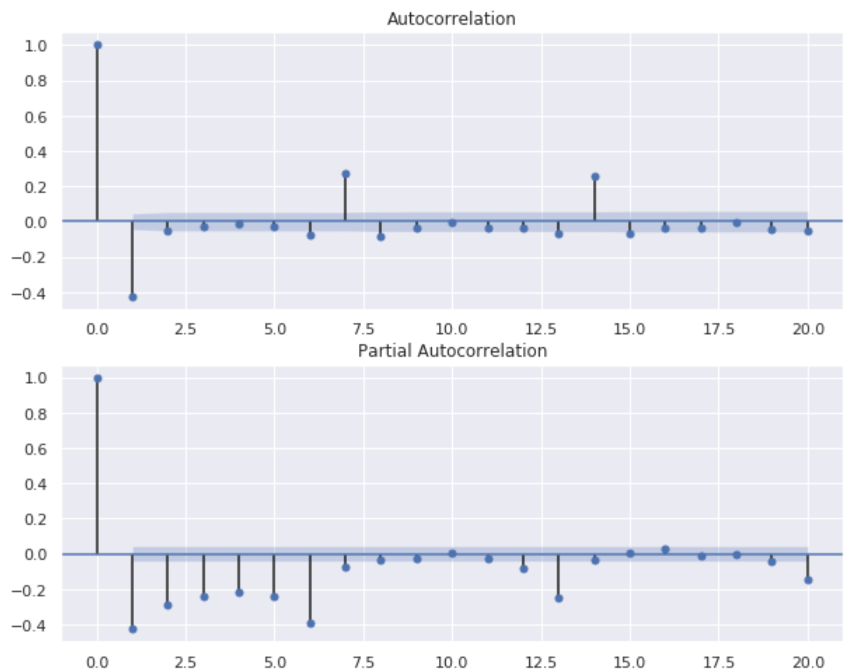
In [10]:
```
#ADF-test(differenced-time-series)
res = sm.tsa.adfuller(buf['sales'].diff().dropna(),regression='c')
print('p-value:{}'.format(res[1]))
```

```
p-value:1.2109276320428997e-23
```

In [7]:
```
tra = buf['sales'].dropna()
tra_log = np.log(buf['sales'])
```

In [8]:
```
#we use tra.diff()(differenced data), because this time series is unit root process.
fig,ax = plt.subplots(2,1,figsize=(10,8))

fig = sm.graphics.tsa.plot_acf(tra.diff().dropna(), lags=20, ax=ax[0])
fig = sm.graphics.tsa.plot_pacf(tra.diff().dropna(), lags=20, ax=ax[1])

plt.show()
```

```
/anaconda3/lib/python3.7/site-packages/matplotlib/font_manager.py:1241: UserWarning: findfont: Font family
['IPAGothic'] not found. Falling back to DejaVu Sans.
  (prop.get_family(), self.defaultFamily[fontext]))
/anaconda3/lib/python3.7/site-packages/matplotlib/font_manager.py:1241: UserWarning: findfont: Font family
['IPAGothic'] not found. Falling back to DejaVu Sans.
  (prop.get_family(), self.defaultFamily[fontext]))
```



In [13]:
```
resDiff = sm.tsa.arma_order_select_ic(tra, max_ar=7, max_ma=7, ic='aic', trend='c')
print('ARMA(p,q) =',resDiff['aic_min_order'],'is the best.')
```

```
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
```

```
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:508: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:508: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
```

```
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:508: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:508: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:508: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:508: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
/anaconda3/lib/python3.7/site-packages/statsmodels/tsa/base/tsa_model.py:171: ValueWarning: No frequency info
rmation was provided, so inferred frequency D will be used.
  % freq, ValueWarning)
```

```
ARMA(p,q) = (7, 7) is the best.

/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:488: HessianInversionWarning: Inverting hess
ian failed, no bse or cov_params available
  'available', HessianInversionWarning)
```

In [14]:
```
arima = sm.tsa.statespace.SARIMAX(tra,order=(7,1,7),freq='D',seasonal_order=(0,0,0,0),
                                  enforce_stationarity=False, enforce_invertibility=False,).fit()
arima.summary()
```

```
/anaconda3/lib/python3.7/site-packages/statsmodels/base/model.py:508: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
```

Out[14]:

Statespace Model Results

Dep. Variable:	sales	**No. Observations:**	1826
Model:	SARIMAX(7, 1, 7)	**Log Likelihood**	-5396.199
Date:	Sat, 07 Dec 2019	**AIC**	10822.397
Time:	16:14:51	**BIC**	10904.972
Sample:	01-01-2013	**HQIC**	10852.864
	- 12-31-2017		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.9106	0.030	-30.117	0.000	-0.970	-0.851
ar.L2	-0.9126	0.030	-30.314	0.000	-0.972	-0.854
ar.L3	-0.9118	0.030	-30.257	0.000	-0.971	-0.853
ar.L4	-0.9120	0.030	-30.179	0.000	-0.971	-0.853
ar.L5	-0.9129	0.030	-30.451	0.000	-0.972	-0.854
ar.L6	-0.9113	0.030	-30.049	0.000	-0.971	-0.852
ar.L7	0.0874	0.030	2.916	0.004	0.029	0.146
ma.L1	0.0615	0.018	3.501	0.000	0.027	0.096
ma.L2	0.1130	0.015	7.452	0.000	0.083	0.143
ma.L3	0.0824	0.019	4.261	0.000	0.044	0.120
ma.L4	0.0873	0.020	4.367	0.000	0.048	0.126
ma.L5	0.0970	0.016	6.187	0.000	0.066	0.128
ma.L6	0.0795	0.017	4.816	0.000	0.047	0.112
ma.L7	-0.8909	0.018	-49.083	0.000	-0.926	-0.855
sigma2	23.8817	0.914	26.129	0.000	22.090	25.673

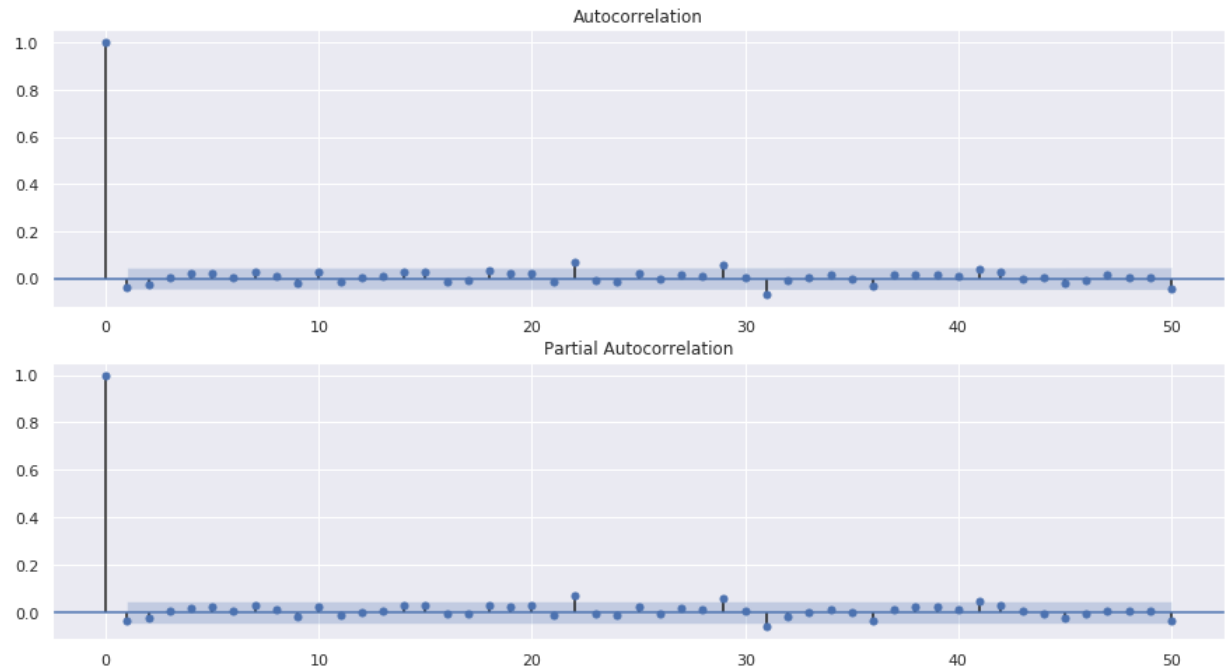
Ljung-Box (Q):	49.87	**Jarque-Bera (JB):**	13.58
Prob(Q):	0.14	**Prob(JB):**	0.00
Heteroskedasticity (H):	1.32	**Skew:**	0.15
Prob(H) (two-sided):	0.00	**Kurtosis:**	3.30

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [10]:
```
SSE = np.sum(arima.resid**2)
SSE
```

Out[10]: 40346.63812836669

In [20]:
```
res = arima.resid
fig,ax = plt.subplots(2,1,figsize=(15,8))
fig = sm.graphics.tsa.plot_acf(res, lags=50, ax=ax[0])
fig = sm.graphics.tsa.plot_pacf(res, lags=50, ax=ax[1])
plt.show()
```

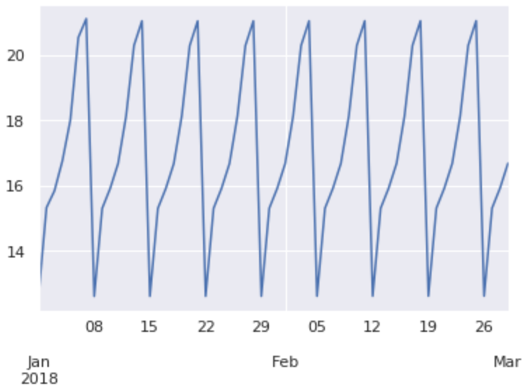


In [15]:
```
sarima = sm.tsa.statespace.SARIMAX(tra, freq='D', enforce_invertibility=False,
                                   order=(0, 1, 1), seasonal_order=(1, 1, 1, 7))
results = sarima.fit()
print(results.summary())
```

```
                                 Statespace Model Results
==========================================================================================
Dep. Variable:                              sales   No. Observations:                 1826
Model:             SARIMAX(0, 1, 1)x(1, 1, 1, 7)   Log Likelihood               -5394.870
Date:                            Sat, 07 Dec 2019   AIC                          10797.740
Time:                                    16:15:00   BIC                          10819.762
Sample:                                01-01-2013   HQIC                         10805.865
                                     - 12-31-2017
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.8952      0.011    -80.737      0.000      -0.917      -0.873
ar.S.L7        0.0375      0.023      1.596      0.110      -0.009       0.084
ma.S.L7       -1.0057      0.008   -122.255      0.000      -1.022      -0.990
sigma2        21.4865      0.707     30.403      0.000      20.101      22.872
===================================================================================
Ljung-Box (Q):                       39.47   Jarque-Bera (JB):                22.76
Prob(Q):                              0.49   Prob(JB):                         0.00
Heteroskedasticity (H):               1.33   Skew:                             0.16
Prob(H) (two-sided):                  0.00   Kurtosis:                         3.44
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

In [16]:
```
tra['fcst'] = results.predict(start='2018-01-01', end='2018-3-01', dynamic=True)
tra['fcst'].loc['2018-01-01':].plot()
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1c25118c50>



In [16]:
```
tra['fcst']
```

Out[16]:
```
2018-01-01    12.436938
2018-01-02    15.124402
2018-01-03    15.749181
2018-01-04    16.828241
2018-01-05    17.950061
2018-01-06    20.464772
2018-01-07    21.070541
2018-01-08    12.574100
2018-01-09    15.263641
2018-01-10    15.872363
2018-01-11    16.645373
2018-01-12    18.088051
2018-01-13    20.255617
2018-01-14    21.004956
2018-01-15    12.577463
2018-01-16    15.267603
2018-01-17    15.879069
2018-01-18    16.639945
2018-01-19    18.093772
2018-01-20    20.249109
2018-01-21    21.003633
2018-01-22    12.578614
2018-01-23    15.268730
2018-01-24    15.880300
2018-01-25    16.640749
2018-01-26    18.094972
2018-01-27    20.249875
2018-01-28    21.004582
2018-01-29    12.579652
2018-01-30    15.269766
2018-01-31    15.881340
2018-02-01    16.641773
2018-02-02    18.096011
2018-02-03    20.250898
2018-02-04    21.005612
2018-02-05    12.580684
2018-02-06    15.270799
2018-02-07    15.882373
2018-02-08    16.642806
2018-02-09    18.097044
2018-02-10    20.251930
2018-02-11    21.006644
2018-02-12    12.581717
2018-02-13    15.271832
2018-02-14    15.883405
2018-02-15    16.643839
2018-02-16    18.098077
2018-02-17    20.252963
2018-02-18    21.007677
2018-02-19    12.582750
2018-02-20    15.272865
2018-02-21    15.884438
2018-02-22    16.644872
2018-02-23    18.099109
2018-02-24    20.253996
2018-02-25    21.008710
2018-02-26    12.583783
2018-02-27    15.273898
2018-02-28    15.885471
2018-03-01    16.645904
Freq: D, dtype: float64
```

In [ ]: