Project Report

Jinhang Yang, Chen Yao, Yingfan Duan, Ziqian Yang, Ximing Zhang

22 November 2019

Contents

Our group chooses the sales dataset to build models and forecast the volume of sales for the next ten days. We establish three models: linear regression model (*Model I*), differencing model (*Model II*) and parametric model (*Model III*). Among them, the *Model III* has best MSE in cross validation and its residual fits $ARIMA(5,0,0)\times(0,0,1)_7$. The prediction of next ten observations show that there exists significant values on Fourth of July and weekends.

1 Exploratory Data Analysis

First, we draw the image of raw data which is displayed as Figure 1. In the image, we can see an obvious seasonality with period of one year, although annul sales increase year by year. In each year, the volume of sales increase and then decrease, which has the similar peak seasons and slack seasons. Furthermore, we find high values in weekends and Fourth of July which is a holiday sale known from the background of the data set.

We can easily see from the data that the variation of the data seems to change from time to time, which is called heteroscedasticity. So at first we use Box-Cox transformation to eliminate the heteroscedasticity (in *Model I* & *Model II*).

Meanwhile, we have another explanation for the changing variance that it grows with time! We believe that the developing internet makes it easier for consumers to buy or return air conditions. If true, the Box-Cox transformation will decrease the variation in peak seasons too much and increase that in slack seasons. Thus we divided sales by time powers some parameter 0.33 to make variation constant (in *Model III*).

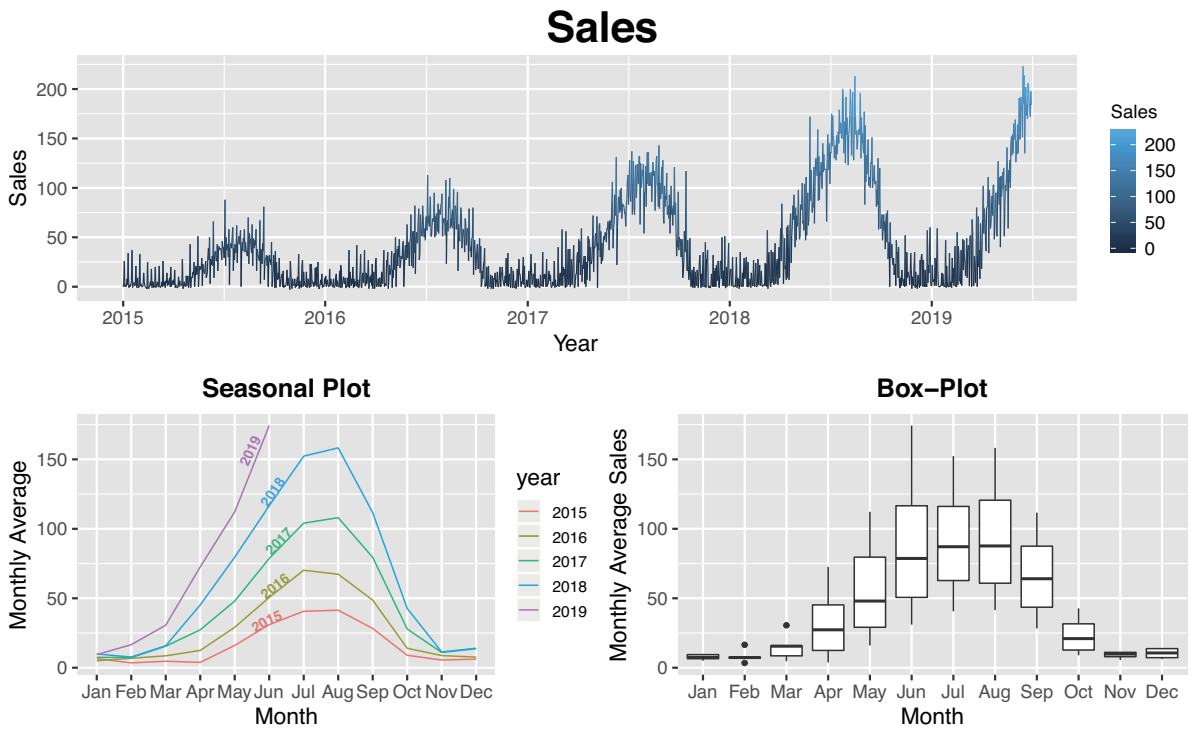

Figure 1: The plot of sales dataset

2 Model a Deterministic Function of Time

2.1 Model I: Linear Regression Model

First, in order to eliminate the influence of leap year, we choose to drop the data before March, 1st, 2016 and call the new data set A1.

Then, we use Box-Cox transformation to eliminate the heteroscedasticity and acquire the new set A2 which has a more stable variance.

Next, Our linear regression model is as the following form (using a R code style):

$$model1 = lm(sales(B2) \sim I(July4th) + I(PeakSeason) + weekdays \\ + time + time * I(PeakSeason))$$

where, $I(July4th)$ is the dummy variable of Fourth of July, $I(PeakSeason)$ is the dummy variable of peak season which includes May to September, weekdays is a vector whose elements are 1 to 7 corresponding Monday to Sunday.

2.2 Model II: Differencing Model

As we did in the linear regression model, we first drop the data before March, 1st, 2016 and conduct Box-Cox transformation. Then we use differencing method to the data, which can shows as following formula:

$$Y_t = \bigtriangledown_{365} \bigtriangledown_7^2 X_t \tag{1}$$

where, X_t is the transformed sales data.

2.3 Model III: Parametric Model

First, we conduct a linear regression of raw sales data on a dummy variable of Fourth of July and a variable whose elements are 1 to 7 corresponding Monday to Sunday. From this linear regression, we can get rid of the influence of special days (Fourth of July) and the differences between weekdays and weekends, making the curves more stable. We call the new data we get B1.

Next, we observe from the residual in the first step (and also of the original data) that the data is slowly changing linearly. But the peak will intervene with linear regression. So we manually take out all the slack seasons (which are linear) to fit the trend, and subtract the fitted value from B1 to get B2.

Then, we want to use a parametric model to fit the the data we get from last step. We can see many peaks from the figure, which is changing smoothly and periodically. They remind us of the positive part of $sin(x)$, because the sales couldn't go below 0 too much. But the peaks grow with time increasing, so we want to try $xsin(x)$ (with only positive part). Later we find the length of slack season, which is constant under $xsin(x)$. So we give the $xsin(x)$ some stretch and rotation (before taking the positive part).

The parametric model that we use is:

$$model3 = max\big(0, (at^2 + b)sin(\frac{2\pi t}{365} - \phi) + ct + d\big) \tag{2}$$

where, a, b, c, d, ϕ are all real constant and t is time.

Last, we run the nonlinear regression of B2 on our parametric model (3). The result shows in the Figure 2 and its residual shows in the Figure 3.

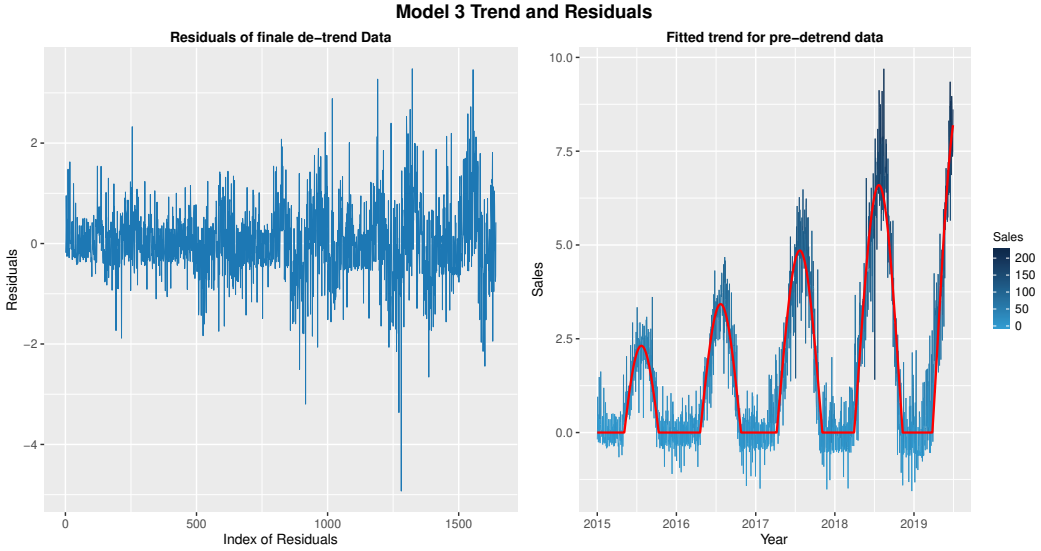

Figure 2: fitted value of parametric model and residual of parametric model

3 ARIMA Model Selection

3.1 Fit ARIMA Model

3.1.1 Model1: linear regression model fits ARIMA

PACF plot suggests all spikes are in the blue dash band after lag 7. The ACF plot is a very typical image of AR model. Hence, we choose AR(7) to fit the residuals. The figure4 is the analysis result from fitting by AR(7).

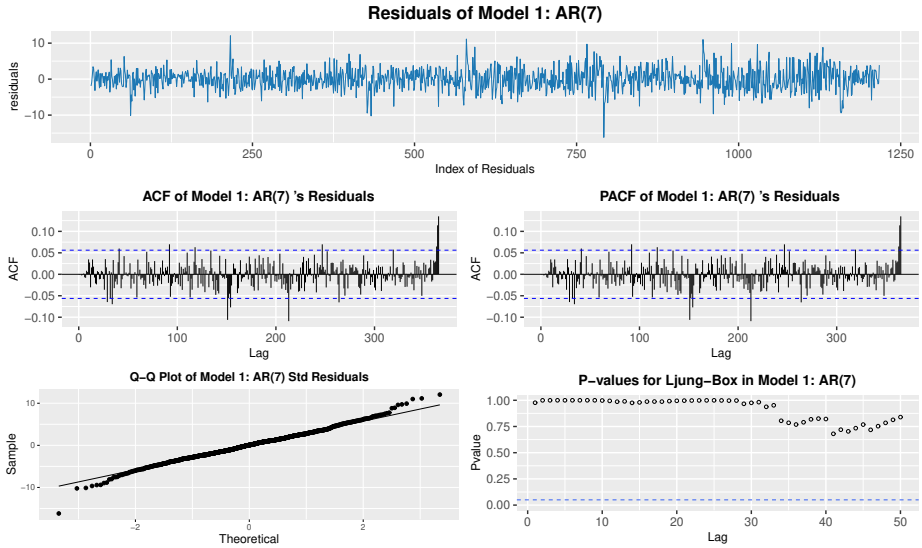

Figure 3: Analysis of SARIMA model1 fit

In the figure, the residual is stable and the spikes in the ACF plot are all in blue dash band. Q-Q plot fits fairly well and the Ljung-Box statistics are completely insignificant, which means this model is very appropriate.

3.1.2 Model2: differencing model fits ARIMA

Model2 seems to have stationary residuals. The ACF and PACF plots suggest that residuals are kinds of seasonal MA(1) with S=7. Q-Q plot fits fairly well, and the Ljung-Box sstatistics are mostly insignificant.

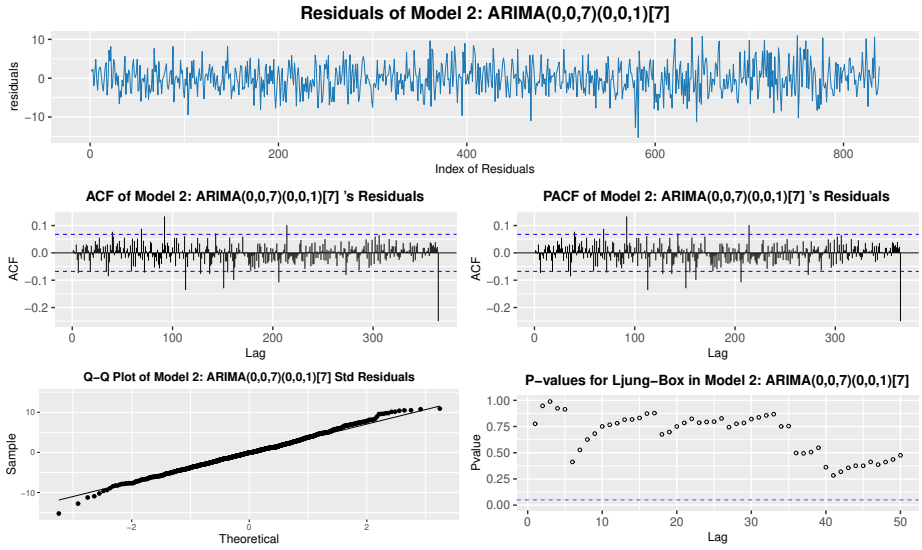

Figure 4: Analysis of SARIMA model2 fit

3.1.3 Model3: parametric model fits ARIMA

The residuals of model3 look fairly stable. From the ACF and PACF plots, we can see that most of spikes are in blue dash bands, which means that the residuals is quite stable as we see from the residuals figure. Most points are on the line of Q-Q plot, which says that residuals seeminly suit normal distribution. Ljung-Box statistics is insignificant before lag 7, however, significant after lag 7. Although Ljung-Box statistics are not so ideal, we believe it is good enough.

3.2 Cross Validation Model Comparison

We conduct cross validation to our three models and get their MSEs, which show in the following table:

Method	MSE
Model I and $AR(7)$	3977.678
Model II and $ARIMA(0,0,7) \times (0,0,1)_7$	2163.689
Model II and $ARIMA(5,0,0) \times (0,0,1)_7$	497.8664

Table 1: MSE of cross validation of three models

Comparing the MSEs of these three models, we can see that *Model III* and $ARIMA(5,0,0) \times (0,0,1)_7$ has the smallest value. Therefore, we select the *Model III* as our final model.

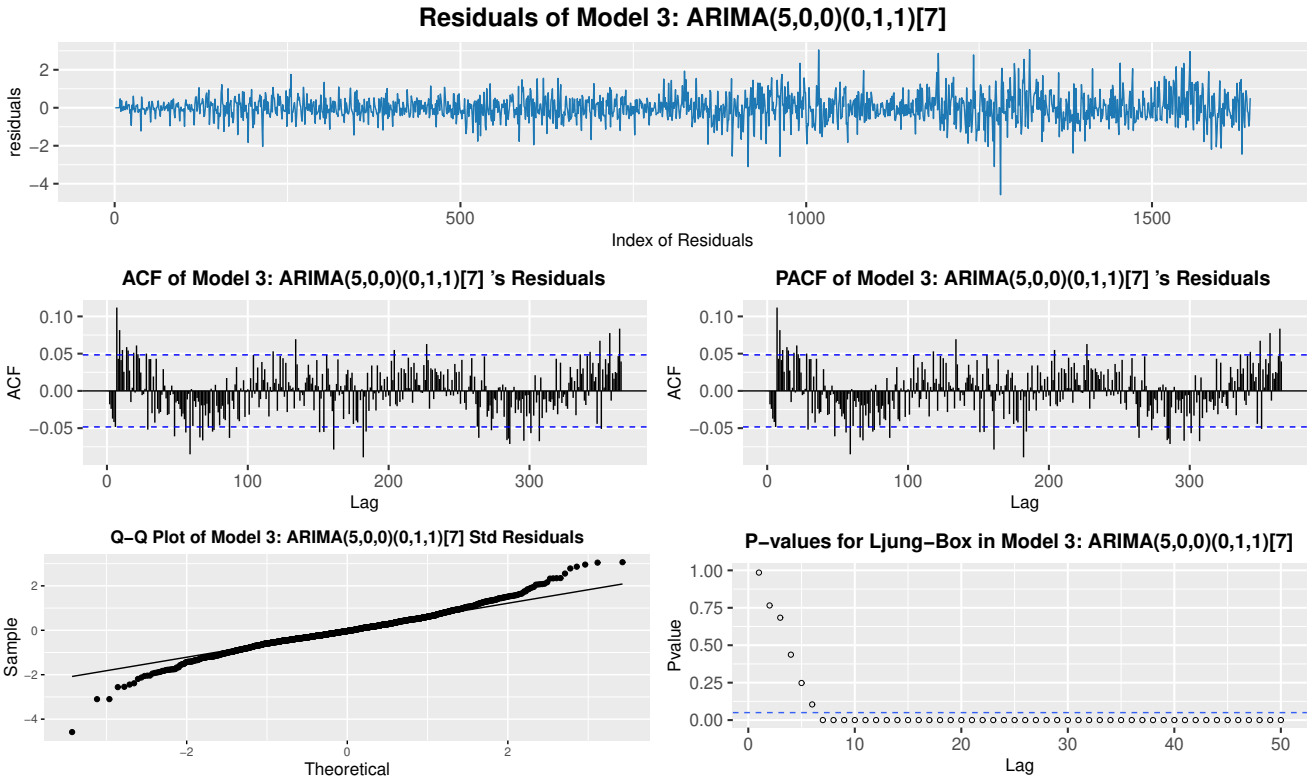

Figure 5: Analysis of SARIMA model3 fit

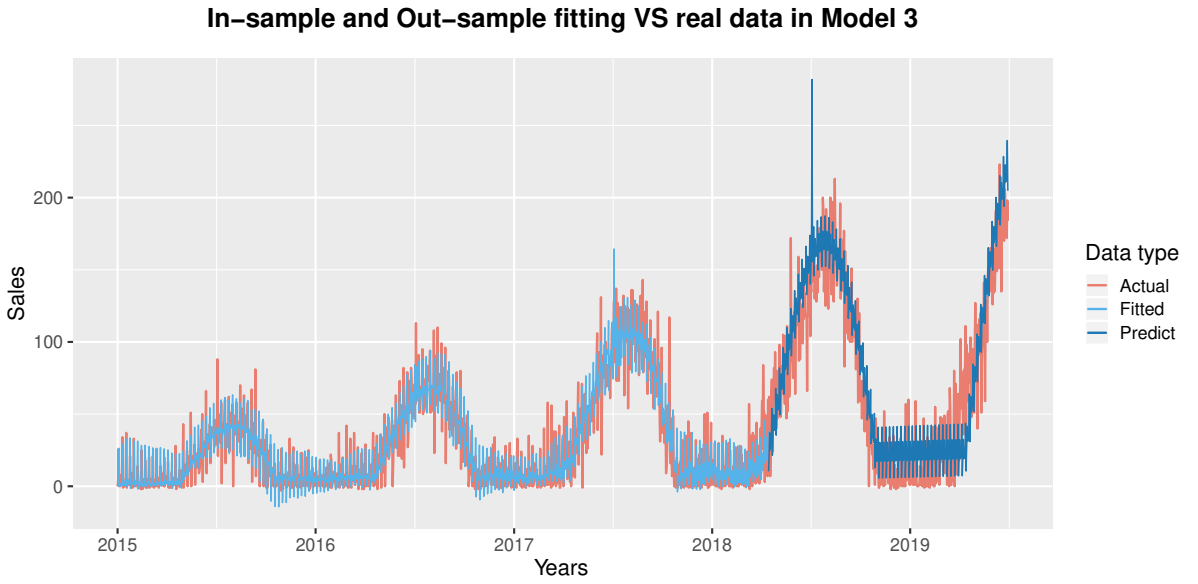

Figure 6: Cross Validation Analysis

4 Result

Through the cross validation, we select parametric model and $ARIMA(5,0,0)\times(0,0,1)_7$ as our final model. The form of parametric model is:

$$\begin{aligned} model =& max\left(0, (1.192\times 10^{-6}t^2 + 2.863)sin(\frac{2\pi t}{365} - 1.897) + 2.121\times 10^{-3}t - 1.030\right) \\ &+ 0.006973 * I(July4th) + weekdays - 0.00166 * time - 0.4403 \end{aligned} \tag{3}$$

Here $weekdays = \begin{pmatrix} 0 & 0.675801 & 0.695858 & 0.718575 & 0.624050 & 0.701860 & 1.759743 \end{pmatrix}$ for Sun. to Sat.

The concrete $ARIMA(5,0,0)\times(0,0,1)_7$ is:

$$(1-\phi_1 B-\phi_2 B^2-\phi_3 B^3-\phi_4 B^4-\phi_5 B^5)X_t = (1+\Theta B^7)Z_t \tag{4}$$

4.1 Estimation of model parameters

Table 2 is the estimated values and standard errors of the coefficients of the ARIMA model:

parameter	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	Θ
estimate	0.1157	0.0874	0.1038	0.0932	0.1134	-0.9873
s.e	0.0246	0.0246	0.0246	0.0246	0.0246	0.0246

Table 2: the estimated values and standard errors of the coefficients of the ARIMA model

4.2 Prediction

As the task requires, we predict the volumes of air condition sales for the next ten days(from July, 1st to July, 10th in 2019) using our parametric model, which are:

date	1	2	3	4	5
value	190.0042	190.5398	191.7920	324.0721	195.6331
date	6	7	8	9	10
value	215.6120	179.9023	197.2183	198.0257	199.0163

Table 3: the predicted volumes of sales for the first ten days in July, 2019

We can see from the table and the trend that the predicted value for July 4th is too high. That's because the first linear model which has dummy variable about July 4th has in fact taken average on all July 4th then minus the mean of all data, but didn't count the fact that the July 4th are in the peak season. But if the effect of July 4 isn't cleared in the beginning, it could harm the latter modeling.

Since our method pays a lot attention to model trend, it may not predict the near data very good. But in the long term, it will perform very well, as is shown in Figure 6.

Appendix: R code

```
# setwd("~/Rs/TS_Project")
setwd("/Users/yaochen/Desktop/UCB_Study/Time_Series/Project")
rm(list=ls())
dev.off()

################### load package and data ###################
library(lubridate)
library(astsa)
library(forecast)
library(dplyr)
library(lubridate)
library(forecast)
library(zoo)
library(xts)
library(ggplot2)
library(ggthemes)
library(ggthemr)
library(ggThemeAssist)
library(ggfortify)
library(hydroTSM)
library(ggpubr)
library(ggDiagnose)
sales_raw <- read.csv("sales.csv")
sales_value <- sales_raw[,3]
start_date=as.Date("2015-01-01")

############### Exploratory Analysis ####################
## Data Transform
time = (mdy(sales_raw[,2]))
sales_value = sales_raw[,3]
sales = zooreg(sales_value, order.by=time, start = time[1])
sales_ts = ts(sales, frequency = 7)
sales_frame = data.frame(time = time, sales = sales)

# 1.1 Original plot
ggthemr_reset()
theme_set(theme_gray(base_size = 16))

plot_original =
  ggplot() +
  aes(x = time, y = sales_value, colour = sales_value)+
  geom_line(size = 0.4) +
  labs(colour = "Sales")+
  ggtitle("Original Sales") + xlab("Year")+ylab("Sales")+
  theme(plot.title = element_text(
    colour = "black",size = 22,
    face = "bold", hjust = 0.5,vjust = -0.5,
    margin=margin(0,0,15,0)),
    legend.title = element_text(size = 13))
scale_color_gradient(low = "#319cd0", high = "#0E294B")

## 1.2 Seasonal Plot
plot_season = ggseasonplot(ts(daily2monthly(sales,FUN=mean,na.rm=TRUE),
                              frequency = 12,start=c(2015,1)))+
  ylab("Monthly Average")+
  ggtitle("Seasonal Plot")+theme(plot.title = element_text(
```

```
    colour = "black",size = 18,
    face = "bold", hjust = 0.5))

## 1.3 Box-plot
sales_box <- daily2monthly(sales,FUN=mean) %>% ts(frequency = 12)
Month <- factor(x= as.vector(cycle(sales_box)), levels = 1:12,
                labels = c("Jan", "Feb", "Mar", "Apr","May", "Jun", "Jul",
                           "Aug","Sep", "Oct", "Nov", "Dec"))
plot_box = ggplot(data.frame(Sales = as.vector(sales_box), Month))+
  geom_boxplot(aes(x = Month, y = Sales, group = Month))+
  ylab("Monthly Average Sales")+
  ggtitle("Box-Plot")+
  theme(plot.title = element_text(size = 18, face="bold", hjust = 0.5))

## Summary
ggarrange(plot_original,ggarrange(plot_season,plot_box,ncol=2,nrow=1),
          ncol=1,nrow=2)


################ Model1 Linear ######################
# Fit the deterministic part
## create the data frame
sales0 <- sales_value
date <- start_date+1:length(sales0)-1
time <- 1:length(sales0)
salesDf <- data.frame(time, date, sales0)

## delete all the data before 2016-03-01
salesDfNoLeap <- salesDf[-(1:425),]

## extract the time series sales
sales_ts <- ts(salesDfNoLeap[,"sales0"])

## variance stablization transformation
lambda <- BoxCox.lambda(sales_ts+10)
trans_sales <- BoxCox(sales_ts+10, lambda = lambda)

## create indicator variables for 4th July, high seasons
month <- salesDfNoLeap[,"date"] %>% month()
weekday <- salesDfNoLeap[,"date"] %>% wday() %>% as.factor()
day <- salesDfNoLeap[,"date"] %>% day()
Indicator4th <- day == 4 & month == 7
IndicatorMonth <- month %in% c(5,6,7,8,9)

## form the new data frame
salesDfNoLeap <- cbind(salesDfNoLeap,trans_sales = as.numeric(trans_sales),
                       weekday, month, Indicator4th, IndicatorMonth)
salesDfNoLeap$time <- 1:nrow(salesDfNoLeap)

## fit the linear model
model1 <- lm(trans_sales~Indicator4th + IndicatorMonth + weekday +
               time + time*IndicatorMonth,
             data = salesDfNoLeap)

# Fit the residuals
s1 <- sarima(model1$residuals, 7,0,0,0,0,0)

ggsarima(s1$fit$residuals,"Model 1: AR(7)")
```

```
################ Model2 Difference ####################
# Difference
diffTsNoleap <- diff(diff(trans_sales, lag = 7, differences = 2),
                lag = 365)

# Check the data after differencing
plot.ts(diffTsNoleap)
acf2(diffTsNoleap)

# Fit the residuals
s2 = sarima(diffTsNoleap, 0,0,7,0,0,1,7)
ggsarima(s2$fit$residuals,"Model 2: ARIMA(0,0,7)(0,0,1)[7]")


#################### Model3 Parametric ####################
# Fit the deterministic part

ND=length(sales_value)
t_in  = 1:ND
start_date=as.Date("2015-01-01")
library(lubridate)
library(astsa)
library(forecast)

sdate=start_date+t_in-1
if(ND<365) stop("Less than 1 year,periodical trend cannot be fitted")

sales=sales_value/((t_in^0.33) + 10)

July4=(day(sdate)==4&month(sdate)==7)
weekd=as.factor(wday(sdate))
model0 = lm( sales~ weekd+July4:t_in)

plain=c(1:110,290:460,660:800,1030:1180,1400:1530)[1:ND]
ptrend=lm(model0$residuals[plain]~t_in[plain])

deTreDum=model0$residuals-ptrend$coefficients[1]-ptrend$coefficients[2]*t_in

s=pmax(0,sin(t_in/365*2*pi-pi*7/12))
trend1=lm(deTreDum~s:t_in+s)
trend5=nls(deTreDum~pmax(0,(a*t_in^2+b)*sin(t_in/365*2*pi-phi)+c*t_in+d),
           start = list(a=trend1$coefficients["s:t_in"],
                        b=trend1$coefficients["s"],
                        c=0,d=0,phi=pi*7/12),
           control=nls.control(warnOnly =T))

YDa=deTreDum-predict(trend5)

# plot trend and residuals for model3
plot_trend = ggplot() +
  aes(x = date, y = deTreDum, colour = sales_value)+
  geom_line(size = 0.4) +
  geom_line(aes(x=date,y=predict(trend5)),color="red",size=1.15)+
  labs(colour = "Sales")+
  ggtitle("Fitted trend for pre-detrend data") + xlab("Year")+ylab("Sales")+
  theme(plot.title = element_text(
```

```
    colour = "black",size = 15,
    face = "bold", hjust = 0.5,vjust = -0.5),
    axis.title=element_text(size=15),
    legend.title = element_text(size = 13))+
    scale_color_gradient(low = "#319cd0", high = "#0E294B")

plot_residuals_trend = ggplot()+
  aes(1:length(YDa), YDa) +
  geom_line(color="#1f78b4")+ xlab("Index of Residuals")+
        ylab("Residuals")+
  labs(title = "Residuals of finale de-trend Data" )+
  theme(plot.title = element_text(size = 15, hjust = 0.5,vjust = -0.5 ,
                                  face="bold"),
        axis.title=element_text(size=15))
ggarrange(plot_residuals_trend,plot_trend,nrow=1,ncol=2)+
  ggtitle("Model 3 Trend and Residuals")+
  theme(plot.title = element_text(size = 20, hjust = 0.5, face="bold"))

# Fit the residuals
s3 = sarima(YDa,5,0,0,0,1,1,7)

# ggsarima for residuals
ggsarima = function(residuals,name){
  residuals = as.vector(residuals)
  plot_residuals = ggplot()+
    aes(1:length(residuals), residuals) +
    geom_line(color="#1f78b4")+ xlab("Index of Residuals")+
    labs(title = paste("Residuals of" , name))+
    theme(plot.title = element_text(size = 19, hjust = 0.5, face="bold"),
          axis.title=element_text(size=13))


  plot_ACF = ggAcf(residuals,lag.max = 365)+
    ggtitle(paste("ACF of" ,name,"'s Residuals"))+
    theme(plot.title = element_text(face="bold", hjust = 0.5,size = 15),
          axis.title=element_text(size=13))

  plot_PACF = ggAcf(residuals,lag.max = 365)+
    ggtitle(paste("PACF of" ,name,"'s Residuals"))+
    theme(plot.title = element_text(face="bold", hjust = 0.5,size = 15),
          axis.title=element_text(size=13))

  plot_gg = ggqqplot(as.vector(residuals),ggtheme = theme_gray())+
    ggtitle(paste("Q-Q Plot of" ,name,"Std Residuals"))+
    theme(plot.title = element_text(face="bold", hjust = 0.5),
          axis.title=element_text(size=13))

  ljungbox <- function(i, resi=residuals) {
    Box.test(resi, i, type = "Ljung-Box")$p.value
  }
  lag = 1:50
  Pvalue <- sapply(lag,ljungbox)
  plot_Ljungbox =
    ggplot() +
    geom_point(aes(x=lag, y=Pvalue),color="black",pch=1)+xlab("Lag")+
    geom_hline(yintercept = 0.05, colour = "#3962f3",linetype=2)
        +ylim(0,1)+
    theme(plot.title = element_text(size = 15, hjust = 0.5, face="bold"),
```

```
          axis.title=element_text(size=13))+
    labs(title = paste("P-values for Ljung-Box in", name))

  ggarrange(plot_residuals,
            ggarrange(plot_ACF,plot_PACF,ncol=2,nrow=1),
            ggarrange(plot_gg,plot_Ljungbox,ncol=2,nrow=1),
            ncol=1,nrow=3)
}

residuals = s3$fit$residuals
ggsarima(residuals,"Model 3: ARIMA(5,0,0)(0,1,1)[7]")

################ Cross Validation ######################
CrossVali <- function(Tdata,Nfun){
  Start_t=365+366
  Pre=1
  MSE=foreach(i=Start_t:length(Tdata),.combine='c') %dopar% {
    mean((Tdata[i-1+1:Pre]-tail(Nfun(sales0=Tdata[1:(i-1)],
                                     pred_num=Pre),N=Pre))^2)
  }
  return(sum(MSE))
}

# model1 forecast function(sales is a vector of raw sales data)
model1For <- function(sales0=sales_value, pred_num = 1){
  start_date=as.Date("2015-01-01")

  # create the data frame
  date <- start_date+1:length(sales0)-1
  time <- 1:length(sales0)
  salesDf <- data.frame(time, date, sales0)

  # delete all the data before 2016-03-01
  salesDfNoLeap <- salesDf[-(1:425),]

  # extract the time series sales
  sales_ts <- ts(salesDfNoLeap[,"sales0"])

  # variance stablization transformation
  lambda <- BoxCox.lambda(sales_ts+10)
  trans_sales <- BoxCox(sales_ts+10, lambda = lambda)

  # create indicator variables for 4th July, high seasons
  month <- salesDfNoLeap[,"date"] %>% month()
  weekday <- salesDfNoLeap[,"date"] %>% wday() %>% as.factor()
  day <- salesDfNoLeap[,"date"] %>% day()
  Indicator4th <- day == 4 & month == 7
  IndicatorMonth <- month %in% c(5,6,7,8,9)

  # form the new data frame
  salesDfNoLeap <- cbind(salesDfNoLeap,trans_sales =
                           as.numeric(trans_sales),
                         weekday, month, Indicator4th, IndicatorMonth)
  salesDfNoLeap$time <- 1:nrow(salesDfNoLeap)

  # fit the linear model
  model1 <- lm(trans_sales~Indicator4th + IndicatorMonth + weekday +
                 time + time*IndicatorMonth,
```

```
                data = salesDfNoLeap)

  # fit the residuals
  resModel1 <- sarima(model1$residuals, 7,0,0,0,0,0)

  # forecast the residuals
  resFor <- sarima.for(model1$residuals,
                       n.ahead = pred_num,
                       7,0,0,0,0,0)$pred

  # forecast the linear model
  preDate <- tail(salesDfNoLeap$date,1)+1:pred_num
  newDate <- c(salesDfNoLeap$date, preDate)
  newMonth <- month(newDate)
  newWeek <- wday(newDate) %>% as.factor()
  newDay <- day(newDate)
  newIndicator4th <- newDay == 4 & newMonth == 7
  newIndicatorMonth <- newMonth %in% c(5,6,7,8,9)
  newTime <- 1:length(newDate)
  predictDF <- data.frame(Indicator4th = newIndicatorMonth,
                          IndicatorMonth = newIndicatorMonth,
                          weekday = newWeek,
                          time = newTime)
  predTran <- predict.lm(model1, newdata = predictDF) +
    c(model1$residuals, resFor)

  # transform the predict value into raw data
  predRaw <- (predTran*lambda +1)^(1/lambda)
  return(predRaw)
}

# cross validation for model2
forecast_residual <- c()
forecast_value <- c()
train_set <- c()
error <- 0
for (i in 381:1217) {
  train_set <- diffTsNoleap[1:(i-379)]
  forecast_residual[i] <- sarima.for(train_set,n.ahead =
                        1,0,0,7,0,0,1,7)$pred
  forecast_value[i] <- forecast_residual[i] +
  2*trans_sales[i-7] - trans_sales[i-14]
  + trans_sales[i-365] - 2*trans_sales[i-372] + trans_sales[i-379]
  error <- error + ((forecast_value[i] - sales_ts[i])^2)[1]
}
error

# model3 forecast function(sales is a vector of raw sales data)
XSIN_Trend<-function(sales0=sales_value,pred_num=1,Peak=2,
                     tin_times=0.33,tin_int=10,WarnOnly=T)
{
  ND=length(sales0)
  t_in  = 1:ND
  start_date=as.Date("2015-01-01")
  library(lubridate)
  library(astsa)
  library(forecast)
```

```
  sdate=start_date+t_in-1
  if(ND<365) stop("Less than 1 year, periodical trend cannot be fitted")

  sales=sales0/((t_in^tin_times) + tin_int) ## 1.

  July4=(day(sdate)==4&month(sdate)==7)
  weekd=as.factor(wday(sdate))
  model0 = lm( sales~ weekd+July4:t_in) ## 2.

  plain=c(1:110,290:460,660:800,1030:1180,1400:1530)[1:ND]
  ptrend=lm(model0$residuals[plain]~t_in[plain]) ## 3.
  deTreDum=model0$residuals-ptrend$coefficients[1]
           -ptrend$coefficients[2]*t_in

  s=pmax(0,sin(t_in/365*2*pi-pi*7/12))
  trend1=lm(deTreDum~s:t_in+s)
  trend5=nls(deTreDum~pmax(0,(a*t_in^Peak+b)*sin(t_in/365*2*pi-phi)
                    +c*t_in+d),
             start = list(a=trend1$coefficients["s:t_in"],
                          b=trend1$coefficients["s"],
                          c=0,d=0,phi=pi*7/12),
             control=nls.control(warnOnly =WarnOnly)) ## 4.
  YDa=deTreDum-predict(trend5)

  # s1 = sarima(Yd_,0,0,7,0,0,1,7)
  # acf2(s1$fit$residuals,max.lag = 800)
  #
  # s1 = sarima(Yd_,0,0,0,0,0,1,7)
  # acf2(s1$fit$residuals,max.lag = 800)
  #
  # s1 = sarima(Yd_,5,0,0,0,0,1,7)
  # acf2(s1$fit$residuals,max.lag = 800)
  s1 = sarima(YDa,5,0,0,0,1,1,7) ## 5.


  if(pred_num<0) stop("pred_num>=0!!!")
  t_in=1:(ND+pred_num)
  sdate=start_date+t_in-1
  July4=(day(sdate)==4&month(sdate)==7)
  weekd=as.factor(wday(sdate))

  if(pred_num!=0)  Fitted=c(YDa-s1$fit$residuals,
                      as.numeric(sarima.for(YDa,pred_num,5,0,0,0,1,1,7)$pred))
  else Fitted=YDa-s1$fit$residuals
  Fitted=Fitted+predict(trend5,newdata = list(t_in=1:(ND+pred_num)))
  Fitted=Fitted+ptrend$coefficients[1]+ptrend$coefficients[2]*t_in
  Fitted=Fitted+predict(model0,newdata = list(weekd=weekd,July4=July4,t_in=t_in))
  Fitted[July4]=Fitted[July4]-mean(Fitted) # Adjusted lm
  Fitted=Fitted*((t_in^tin_times) + tin_int)

  # dev.off()
  return(Fitted)
}

# cross validation for model 1 and 3
a=tail(XSIN_Trend(pred_num=10,WarnOnly = T),10)
plot(1543:1642,tail(c(sales_value),100),type="l",xlim = c(1543,1652),ylim = c(0,260))
lines(1643:1652,a,col="red")
```

```
a=tail(model1For(pred_num=10),10)
plot(c(sales_value),type="l",xlim = c(0,length(sales_value+10)),ylim=c(0,300))
lines(1643:1652,a,col="red")

jpeg("TRASH.jpg")
library(doParallel)
cl <- makeCluster(4)
registerDoParallel(cl)
TIME<-system.time({
  SSE=CrossVali(sales_value,XSIN_Trend)
})
stopCluster(cl)
dev.off()

jpeg("TRASH.jpg")
library(doParallel)
cl <- makeCluster(4)
registerDoParallel(cl)
TIME<-system.time({
  SSE=CrossVali(sales_value,model1For)
})
stopCluster(cl)
dev.off()


# Plots of Cross Validation(red line is in-sample prediction, blue line is out-sample)
a=XSIN_Trend(pred_num=10,WarnOnly = T)
plot(c(sales_value),type="l",xlim = c(0,length(sales_value+10)))
lines(a,col="red")


## CV of Models
model_summary$date

### Function of plotting
plot_predict = function(model_summary, name){
  theme_set(theme_gray(base_size = 16))
  ggplot(model_summary)+
    geom_line(aes(x=date,y=Actual, colour = "Actual"), size=0.8,na.rm=TRUE) +
    geom_line(aes(x=date,y=Fitted, colour = "Fitted"), size=0.5,na.rm=TRUE)+
    geom_line(aes(x=date,y=Predict, colour = "Predict"), size=0.5,na.rm=TRUE)+
    scale_color_manual(values=c("#e87d71", "#56B4E9", "#1f78b4"))+
    ggtitle(paste("In-sample and Out-sample fitting VS real data in" ,name))+
    labs(colour = "Data type")+ylab("Sales")+xlab("Years")+
    theme(plot.title = element_text(face="bold", hjust = 0.5,vjust=-1.9,
                                    size = 19,margin=margin(0,0,30,0)))
}

### CV of Model 3

N=1200
a=XSIN_Trend(sales_value[1:N],pred_num=length(sales_value)-N,WarnOnly = T)

model3_summary = data.frame(date=date,
                            Actual = sales_value,
                            Fitted = c(a[1:N],rep(NA,length(sales_value)-N)),
                            Predict = c(rep(NA, N),a[(N+1):length(sales_value)]))
```

```
### CV summary
plot_model3_summary = plot_predict(model3_summary,"Model 3")
```