STAT 443
University of Waterloo
Faculty of Mathematics
December 5, 2016

US CANADA BORDER CROSSINGS

Predicting future traffic

Group J:
Hwang, Jihye: HoltWinters
Lin, Boxi: SARIMA
Liu, Xuan Ting: Introduction, Data, Conclusion
Luo, Huiyin: Linear Model and ARMA residuals
Truong Nhat, Minh: Linear Models

US/Canada Border Crossing

Jihye Hwang
Boxi Lin
Xuan Ting Liu
Truong Nhat Minh
Huiyin Luo

Contents

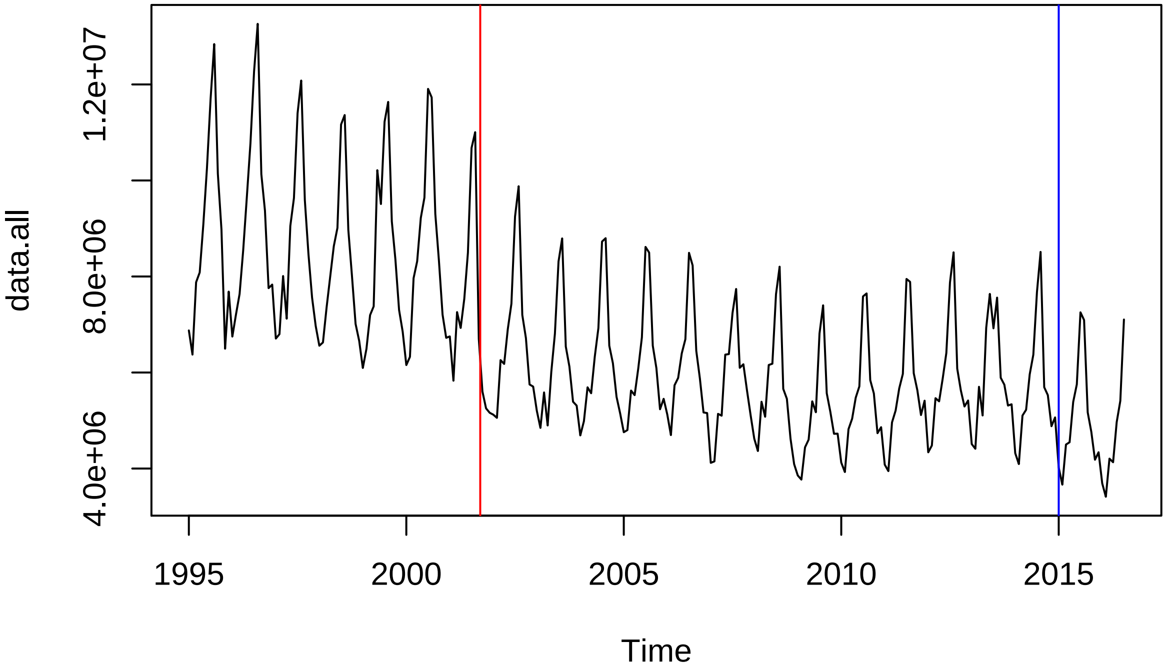
Motivation and Introduction

Cars are ubiquitous in society today and with the everyday use of vehicles, patterns depend. The following report depicts the predictions of the future volumes of traffic at the US Canada border across all 85 border ports. The data taken from the US Bureau of Transportation describes all monthly visitor crossings where visitors are defined as bus, train, truck and personal vehicle passengers. As a group, we believe this data and the predictions are valuable since traffic and congestion control is vital to the travels of the people living on both sides of the border. Knowing when to increase the amount of lanes available to cross the border with not only encourage more movement but also decrease the amount of potential chaos.

On the hand, US/Canada border crossing volume is usually closely related with currency exchange rate or sudden events. In this project, we would like to specifically investigate how border crossing volume would be affected by these factors.

Data

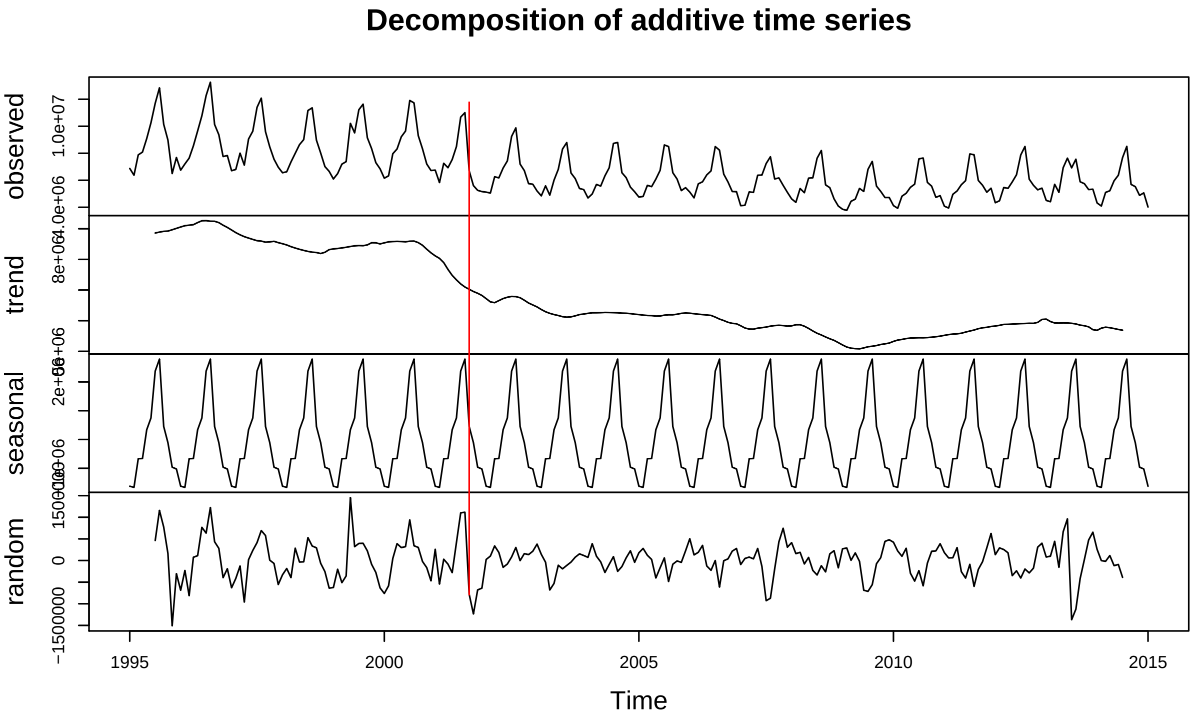
The data is monthly from January 1995 to July 2016. Prior to fitting a model, the data is split into a training (Jan 1995 - Dec 2014) set and a testing (Jan 2015 - July 2016) set, displayed below where the blue line indicates the split of the sets.



The training data is then decomposed to further analyze the features of the time series.

There appears to be a change point around September of 2011. The volume of cars crossing the border every year seems to decease significantly after. From the source, it is noted: “The close relationship between the exchange rate and cross-border shopping from 1986 to 2001 weakened substantially after the September 11, 2001 terrorist attacks as border security tightened and has not been re-established since” (US Bureau of

Transportation). A noticable fact is that, the data set before and after September 2001 looks totally different. We need further effort to see how this event affect our data.

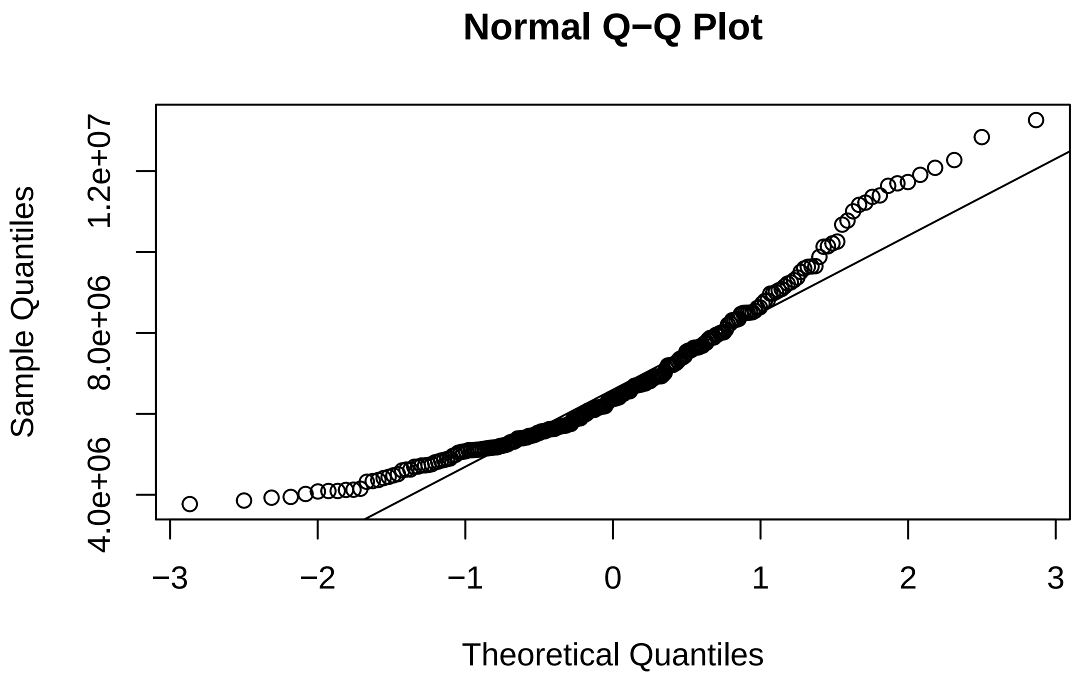


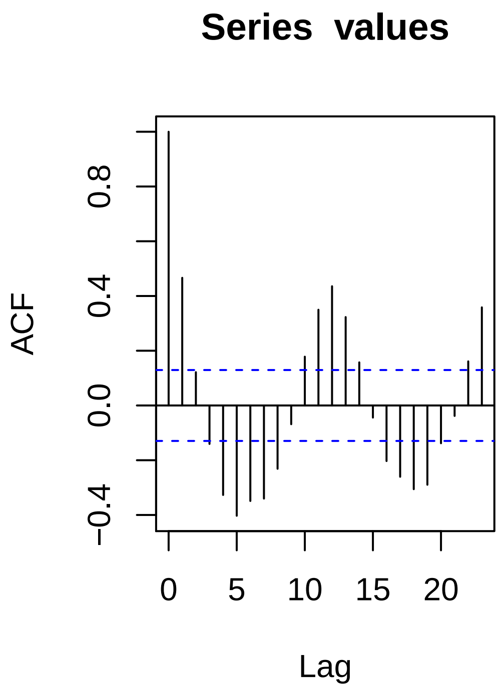
From the turning point test and the Sharpio-Wilk normality test, which gave a p-value of 2.571e-09, both tests rejects normality. ACF and PACF of the residual suggests that there is more seasonality that can be captured by removing the seasonal component from the decompse function. Furthermore, from the normal q-q plot, it is evident that the data violates the normality assumption especially at the ends of the data and appears to be right skewed.

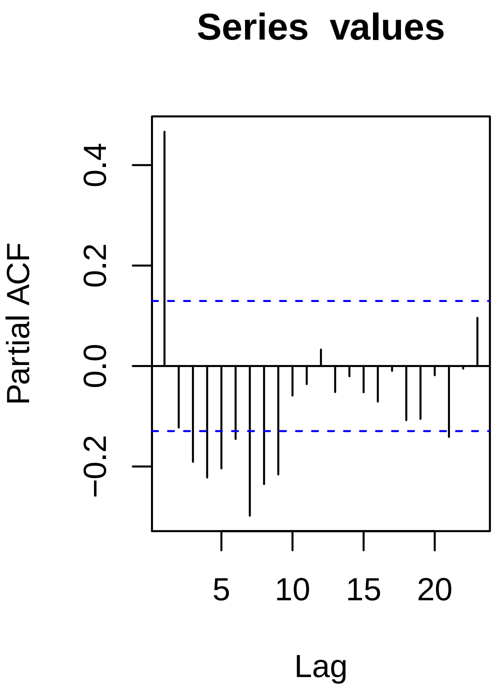
```
##
##  Shapiro-Wilk normality test
##
## data:  data.training
## W = 0.92953, p-value = 2.571e-09
```

```
## $test.sum
## [1] 88
##
## $mu
## [1] 159.3333
##
## $var
## [1] 42.52222
##
## $test.stat
## [1] 10.93917
##
## $p.value
## [1] 0
##
```

```
## $reject.null
## [1] TRUE
```

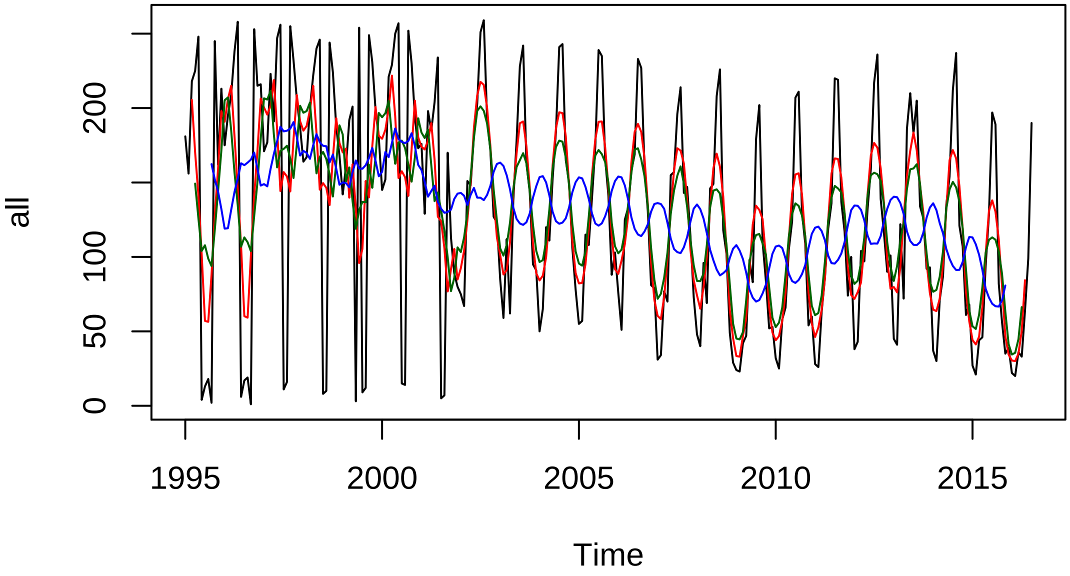






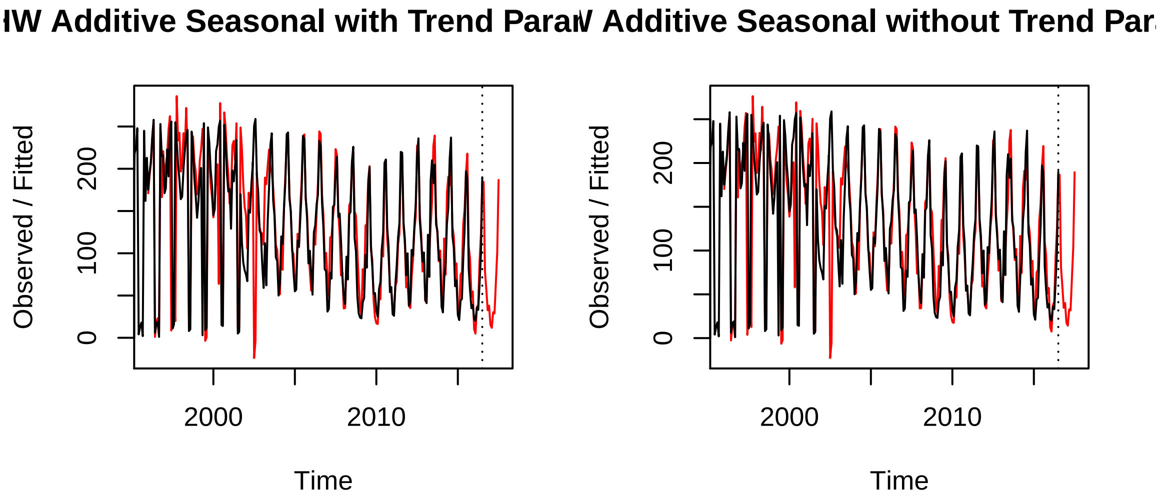
Smoothing Methods

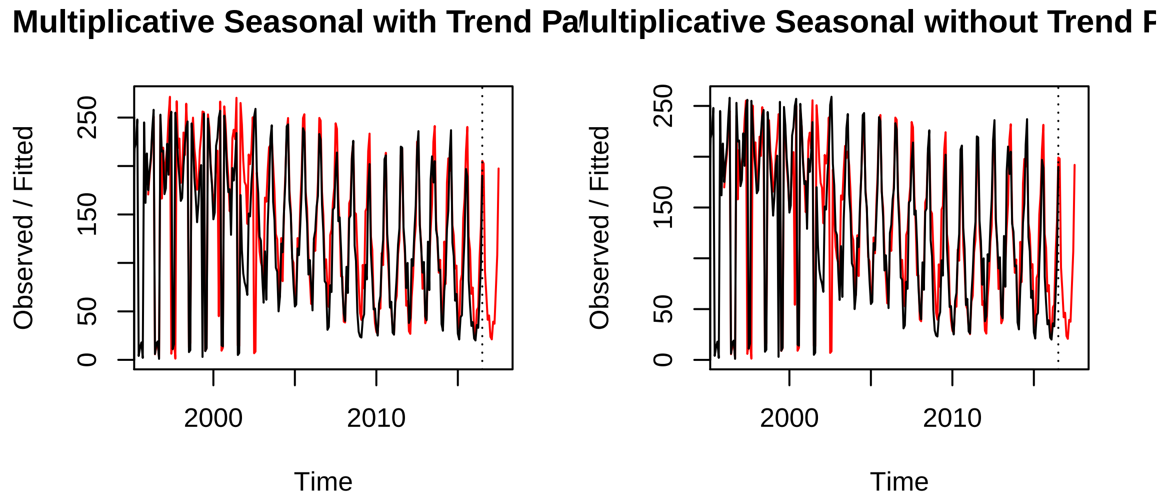
Moving Average Filter



MA(2) (red) is smoother than the plot, MA(3) (dark green) is even smoother, and MA(8) is even smoother loosing some of the real features of the data. From the Moving Average Filter, it has a trend of going up and down and this may be seasonality or due to other factors.

Holt-Winters





```
model1$SSE
```

```
## [1] 533709.3
```

```
model2$SSE
```

```
## [1] 589445.4
```

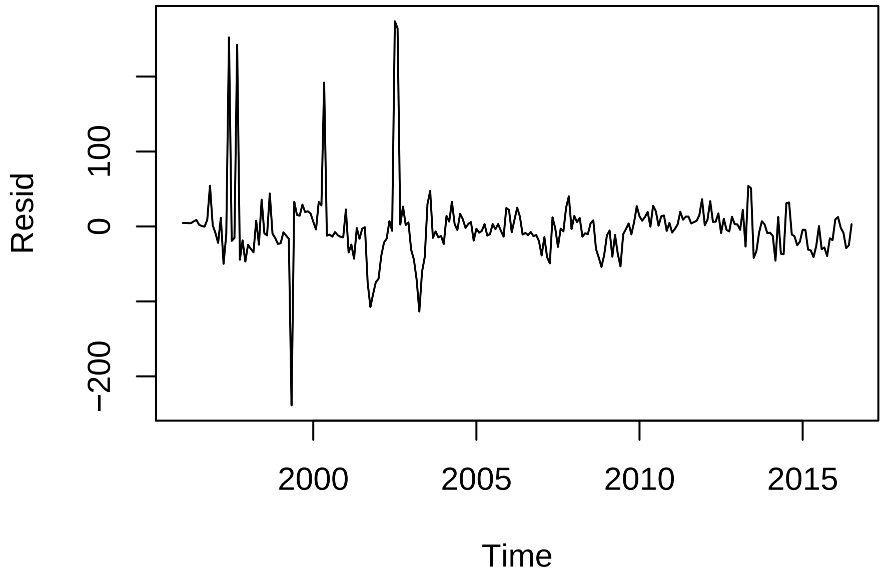
```
model3$SSE
```

```
## [1] 526358.3
```

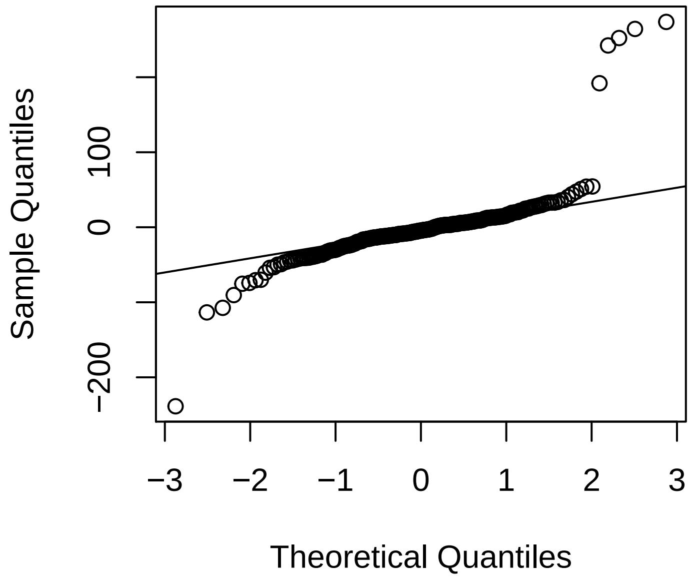
```
model4$SSE
```

```
## [1] 537103.6
```

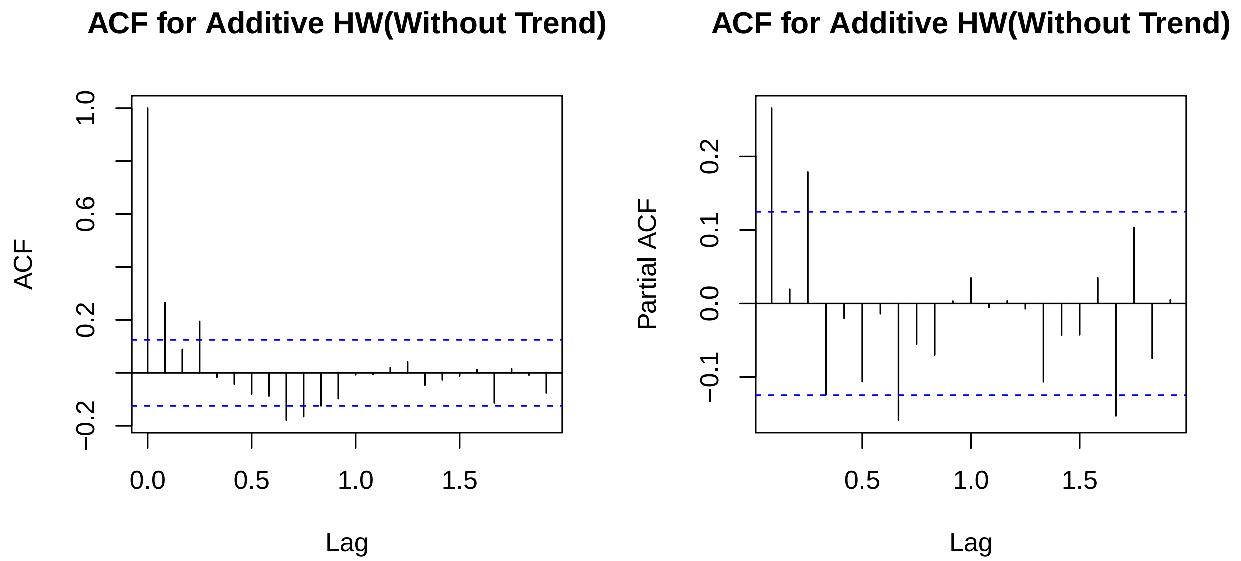
Thus based on all figures, we decide to look deeply into the model 3 which has the lowest SSE out of all 6 Holt-Winters models. We will run full residuals diagnostics on the model.



Residuals for Additive HW(Without Trend)



QQ plot for Additive HW(Without Trend)



```
## $n.runs
## [1] 81
##
## $n.pos
## [1] 107
##
## $n.neg
## [1] 140
##
## $mu
## [1] 122.2955
##
## $var
## [1] 59.31428
##
## $test.stat
## [1] 5.36196
##
## $p.value
## [1] 8.232378e-08
##
## $reject.null
## [1] TRUE
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res3
## W = 0.6705, p-value < 2.2e-16
```

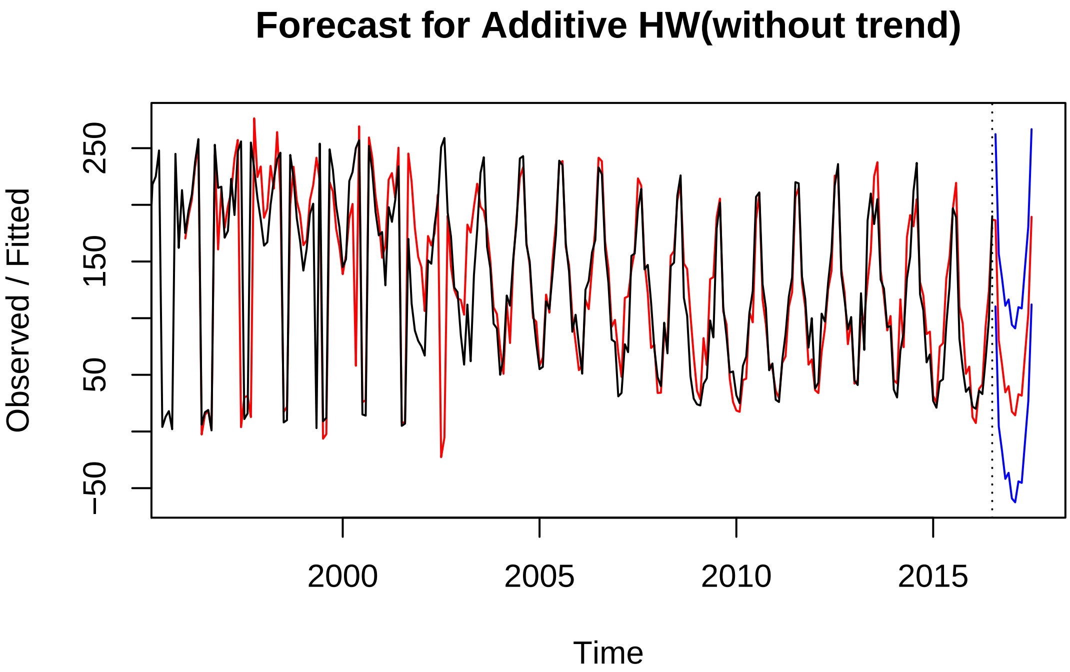
```
## $test.sum
## [1] 121
##
## $mu
## [1] 123
```

```
##
## $var
## [1] 20.66667
##
## $test.stat
## [1] 0.4399413
##
## $p.value
## [1] 0.6599796
##
## $reject.null
## [1] FALSE

## $test.sum
## [1] 153
##
## $mu
## [1] 163.3333
##
## $var
## [1] 43.58889
##
## $test.stat
## [1] 1.565138
##
## $p.value
## [1] 0.1175506
##
## $reject.null
## [1] FALSE
```

From the hypothesis test, we get the p values from shapiro test: 1.197652e-21, runs test: 8.232378e-08, difference sign test: 0.6599796 and turning point test: 0.1175506.

The first thing to notice is there is an obvious pattern in the residuals. We see that it has a constant mean around zero althought there are some spikes in 1999 and 2003, so it may indicate that something is missing from the model. Normality looks fine, both from the QQ plot and the S-W test. The acf and pacf plots look good since the lines are all inside the error bar.

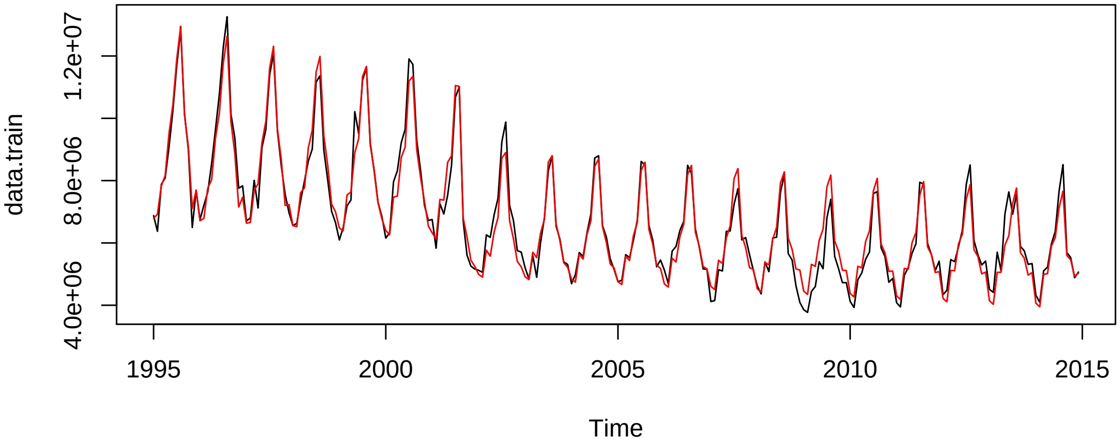

Forecast for Additive HW(without trend)
Observed / Fitted
250
150
50
−50
2000
2005
2010
2015
Time

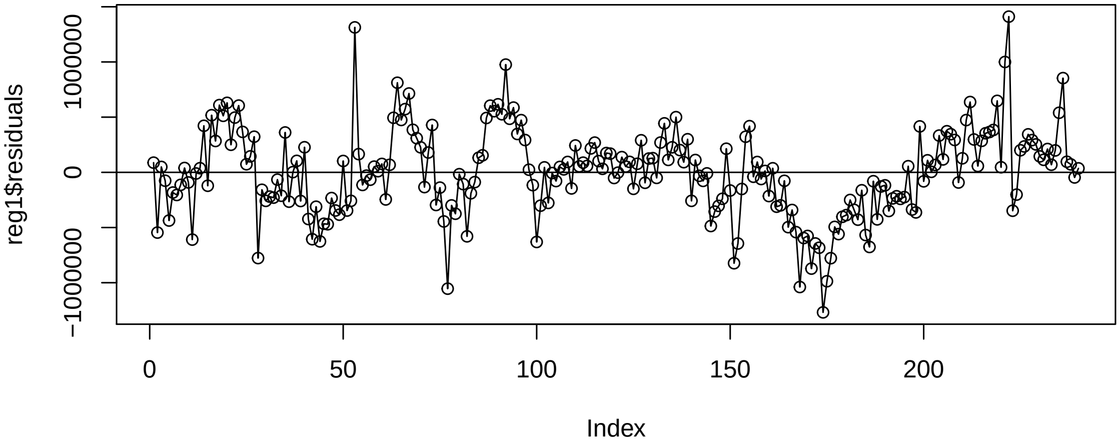
Regression

Linear Regresion Model

- The linear regression model of the number of vehicle passing between US and Canada is modeled based on the interaction between time, month and change point (which is 0 before September 2001 and 1 afterwards).

$$\text{Model 1}: reg1 \leftarrow lm(data.train \sim times \times month \times chg)$$





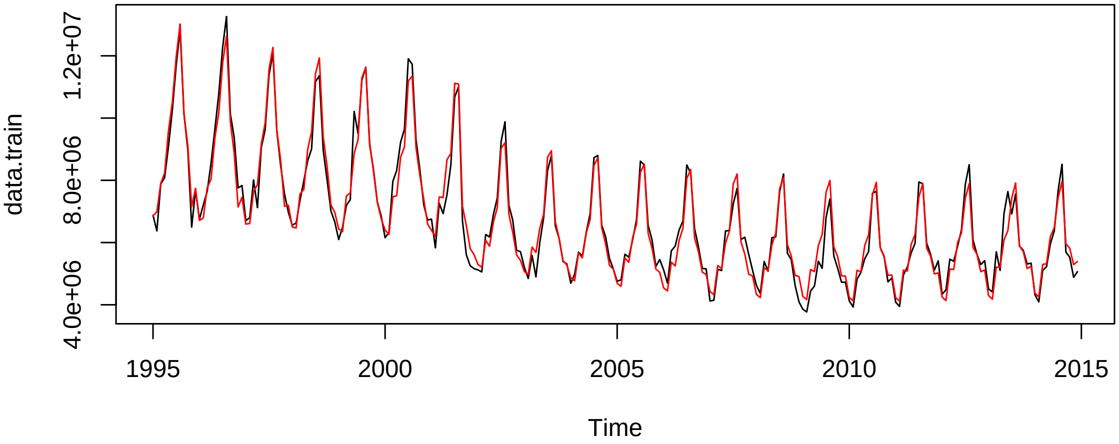
- As we can see, the fitted line of this linear regression model not very well. In addition, we can observe a trend in the residual plot. This may due to the lack of certain predictors.

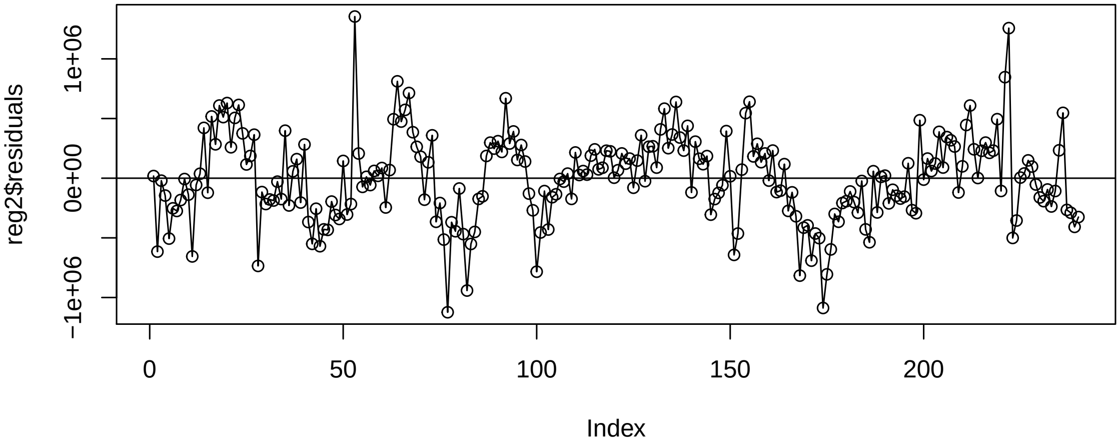
Quadratic Regression Model

- As mentioned above, the residual plot of the linear regression model suggests that there is a lack of certain predictors. Hence, we suspect that the data may follow quadratic regression model. As a result, we add in predictor time2 which is time squared.

- In order to make sure that our suspection is true, we did an anova test between the linear regression model and the quadratic regression model. We obtain p-value = $4.297e^{-}8$. Hence, we decides that the predictor time2 is significant.

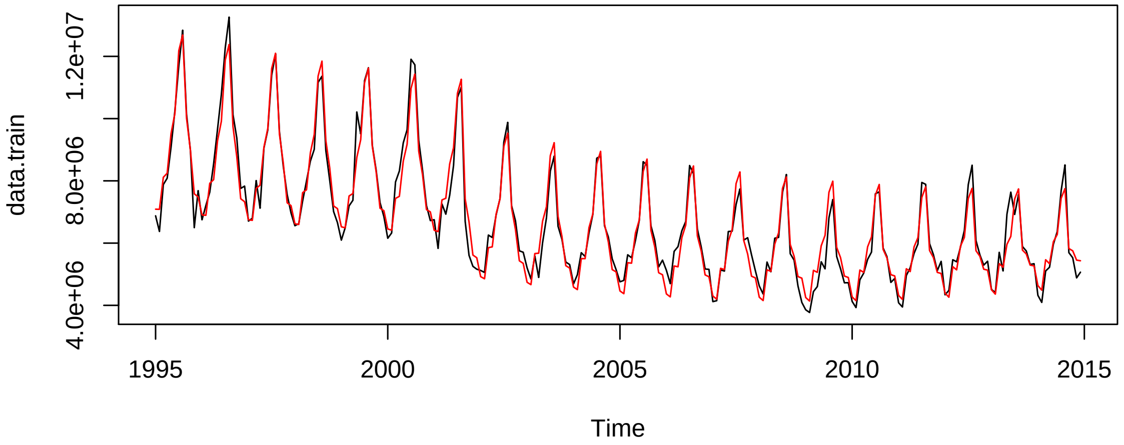
$$\text{Model 2}: reg2 \leftarrow lm(data.train \sim time \times month \times chg + time2)$$



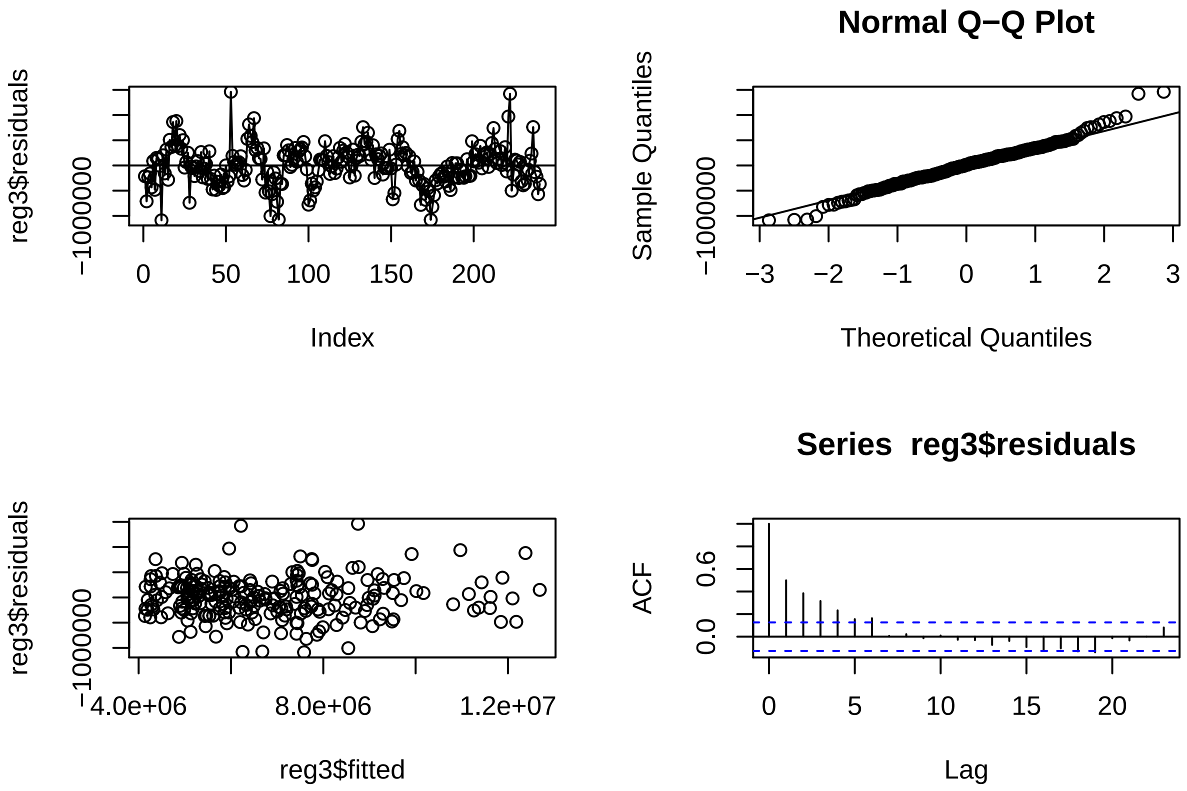


- As we can see, even though the residual plot of this model still shows a trend, its fitted line is much better than that of the linear model.
- In order to obtain the best quadratic regression model, we perform stepwise AIC from both side. As a result, we obtain the new model

Model 3 : $reg3 \leftarrow lm(data.train \sim time \times month + time \times chg + time2)$



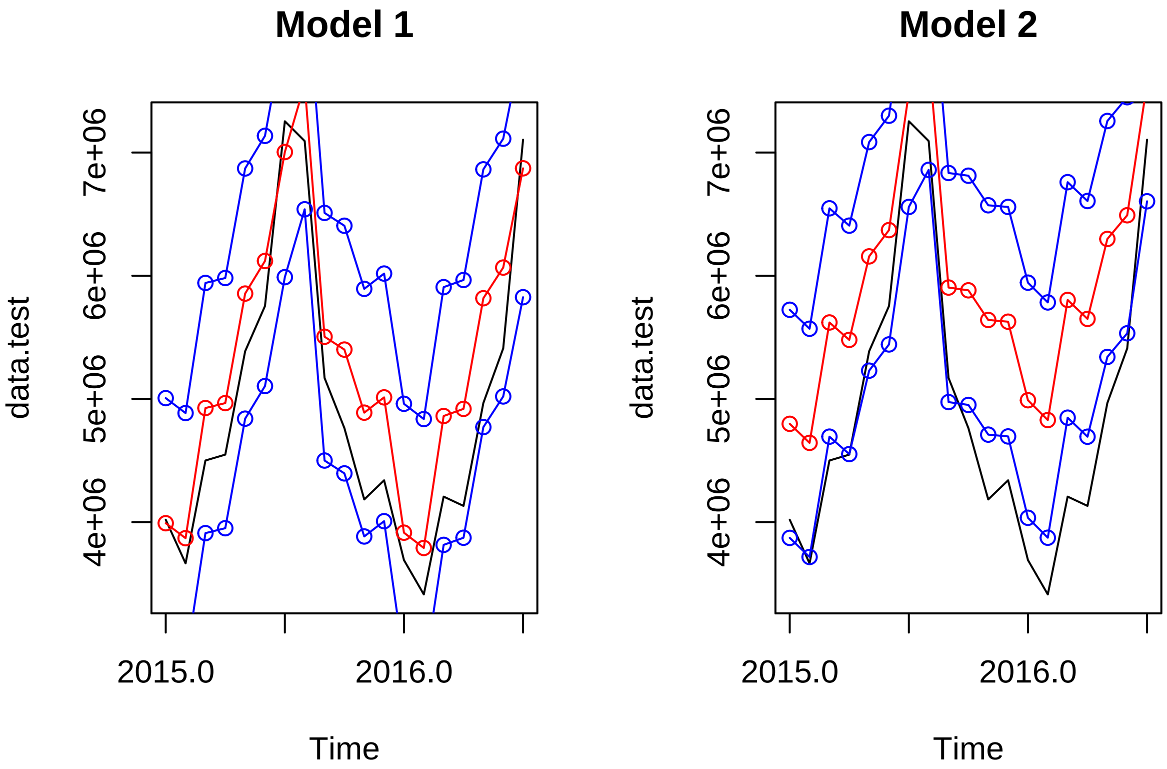
- As we can see, the AIC of this new model decreases slightly from 6248.1 to 6237.57. The fitted line Model 3 is slightly better than that of Model 2.


Normal Q−Q Plot

Series reg3$residuals

- However, for model 3, the residual plot still shows a trend. The QQ plot shows that the residuals are not normally distributed, and from the ACF plot, we can see that the residuals are correlated.

Prediction



Model 1

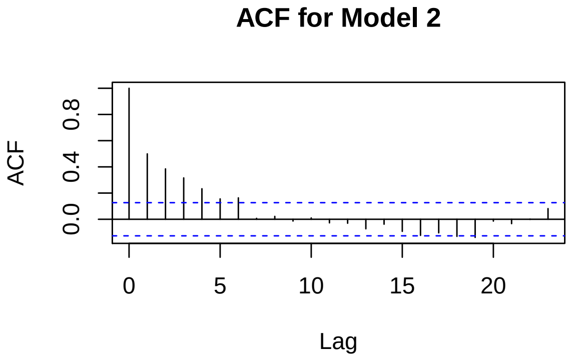
Model 2

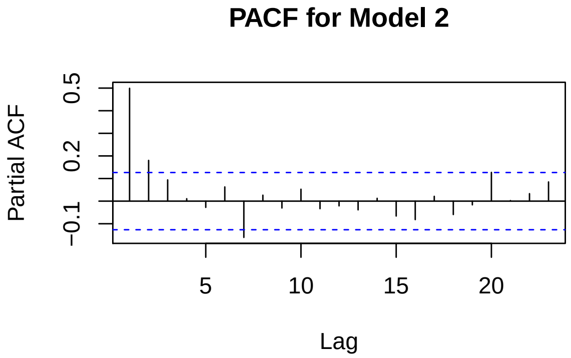
- From the prediction plot, even Model 1 has higher AIC, we can see that its prediction better than that of Model 3. However, this is understandable since more variance will naturally decrease the predictive ability of data. So we need to make a trade-off between fitting and prediction. In addition, generally, the prediction of both looks not very good, this is due to overfitting.

Linear Regression + ARMA Residual

Fitting Model 2 +ARMA

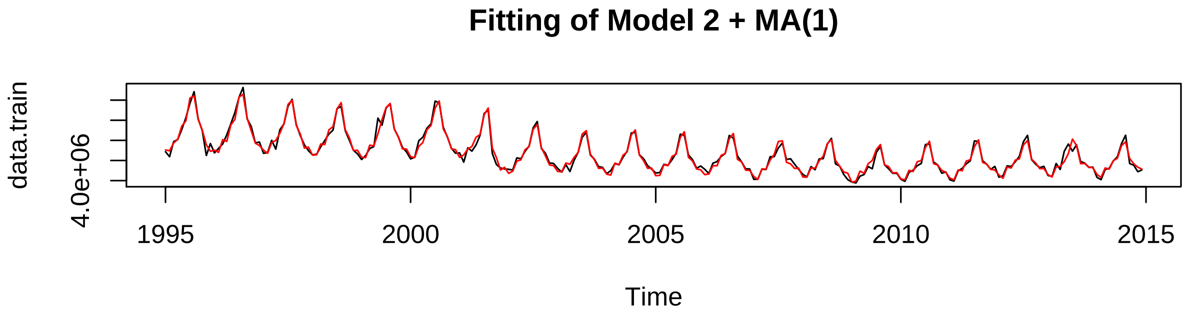
- Although the data fits with Linear Model 2 pretty well, a clear correlation can be observed from the ACF and PACF of the Residual:

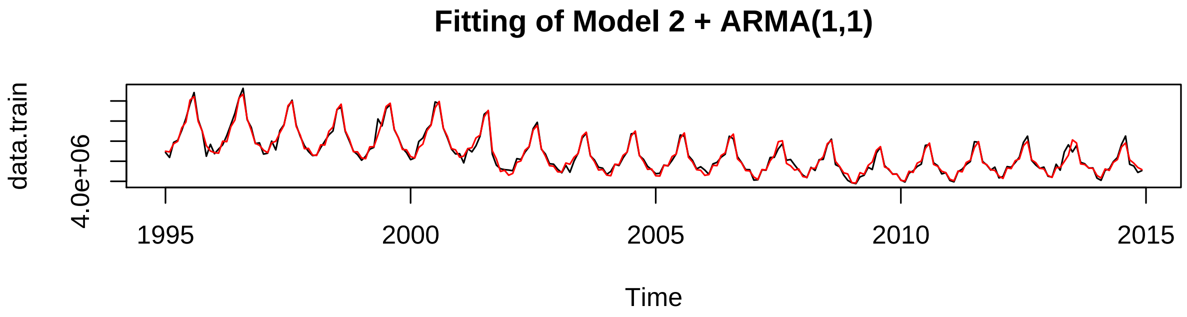




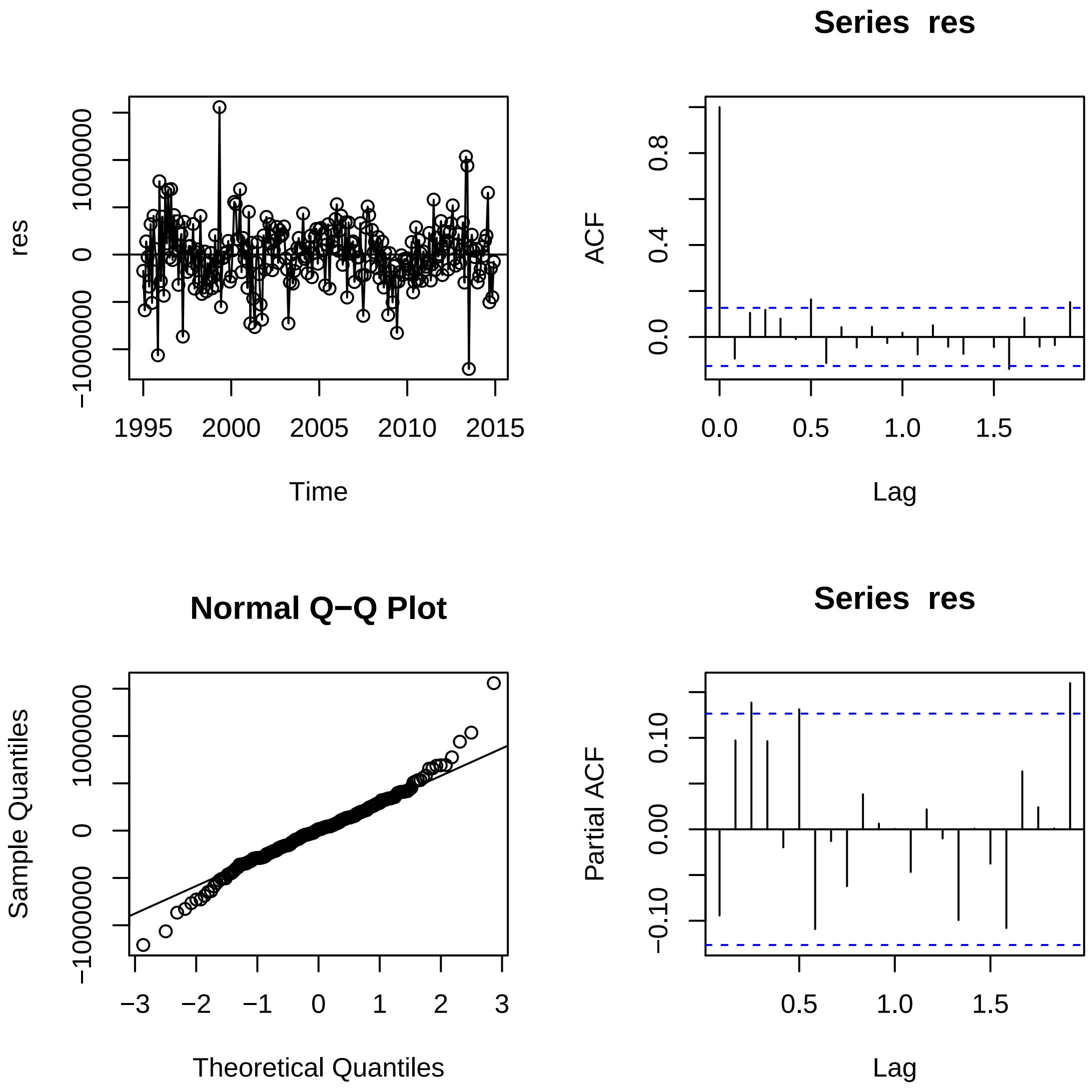
- Based on the shape spikes of ACF/PACF, MA(1) and ARMA(1,1) are two plausible candidate residual models. Testing both of them:

Residual MA(1) VS ARMA(1,1)





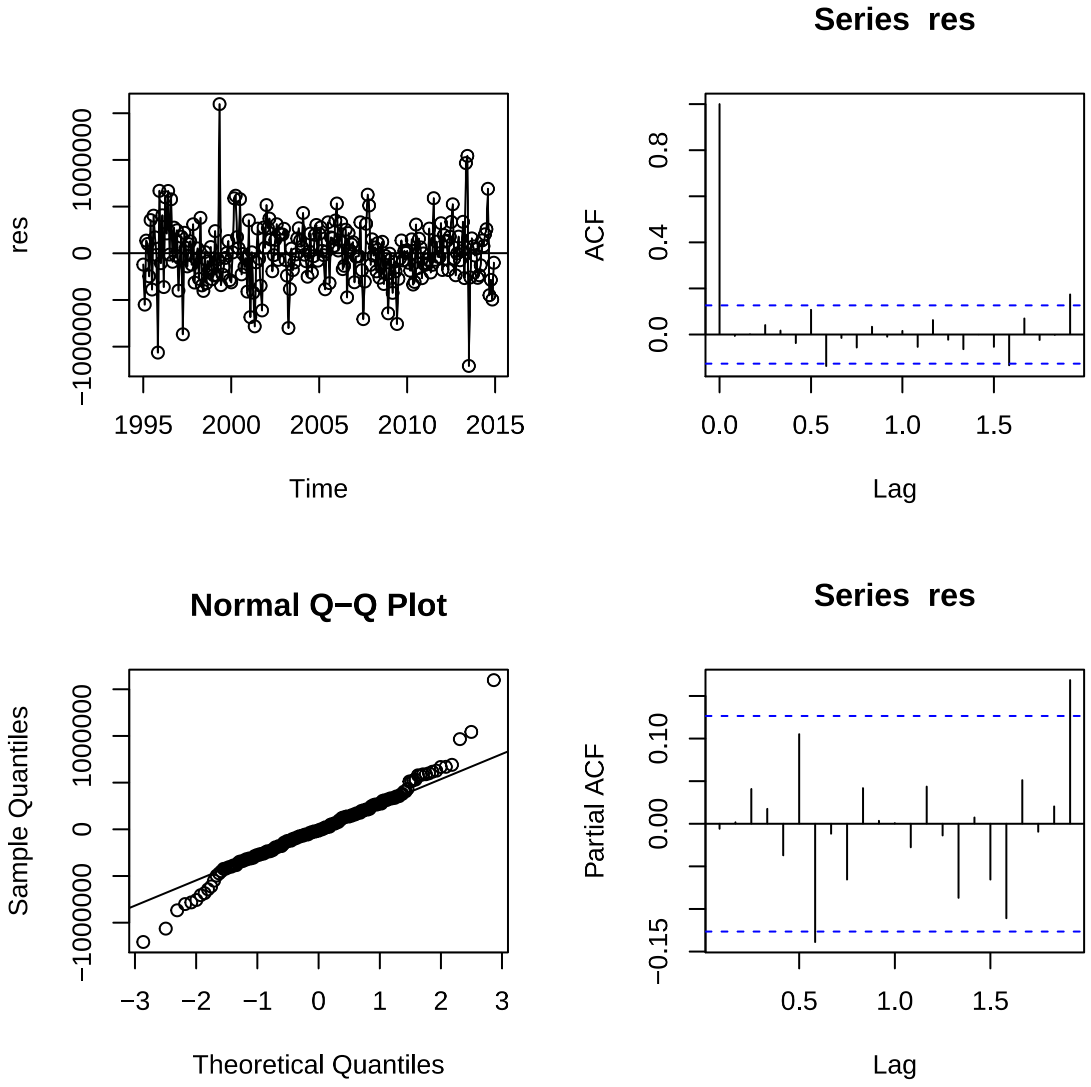
Residual diagnostics - MA(1)



AIC: 6853.409

σ^2 = 1.340919e+22

Residual diagnostics - ARMA(1,1)



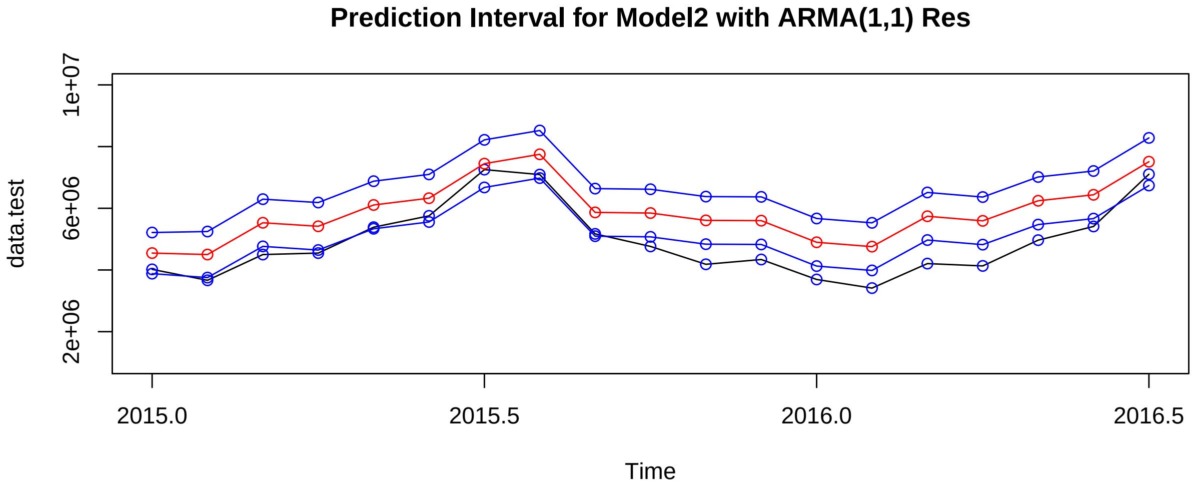
AIC: 6844.024

σ^2 = 1.218349e+22

- We can see that by both models, correlation in residual are cleaned. But since ARMA(1,1) has a lower AIC and σ^2, we would like to select it as our residual model.

Prediction

- However, the prediction ability of this model is really weak:



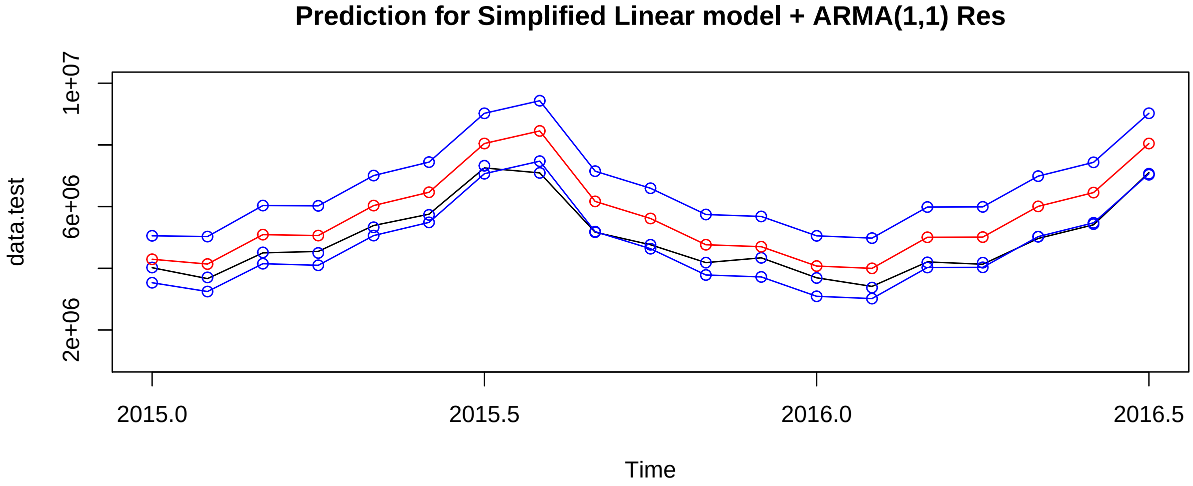
PRESS: 1.556803e+13

- According to the prediction plot, most testing set points are far away from prediction and about half of them fall out of 95% confident interval due to overfitting.
- Therefore, we would like to try to apply a different strategy: reducing the variates in linear model.

Fitting Simplified Model + ARMA

- Our simplified model is $data \sim time + time^2 + month + chg + chg : time$. We removed all interact variates of $time : month$ since most of their parameters have no significant level to be rejected to be 0. And by the same process as above, we found that ARMA(1,1) is the best ARMA model for residual.

Prediction



- The new model looks a little better in terms of prediction, at least all testing set points fall in 95% condidence interval with a lower PRESEE 1.145973e+13. However, the overfitting issue is still not solved. Most testing data are underestimated.

- So can we further simplify the linear model?

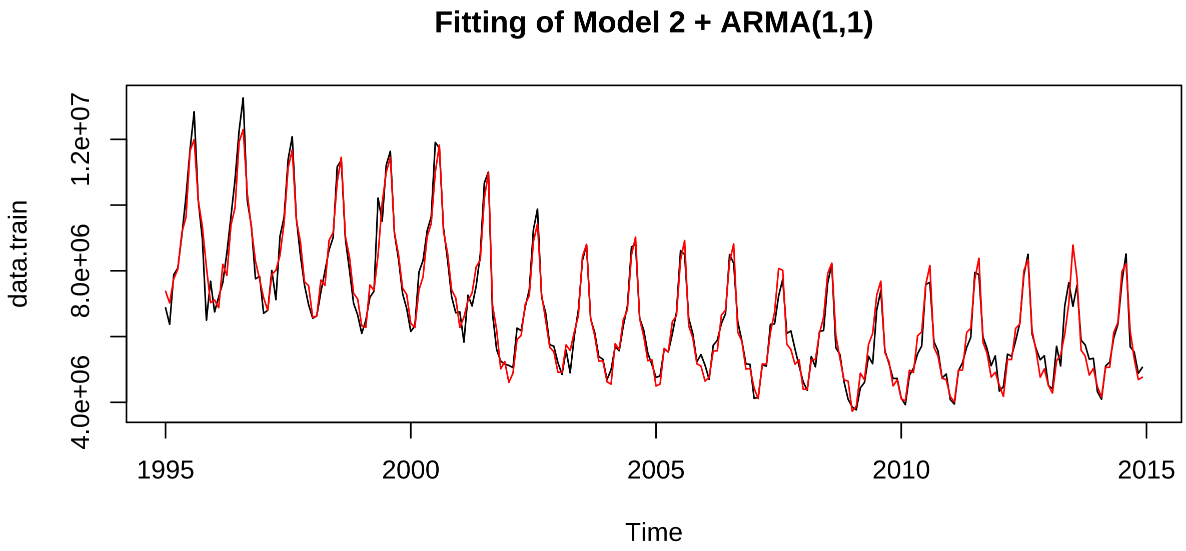
The answer is no. If we take a look at the linear model's parameters:

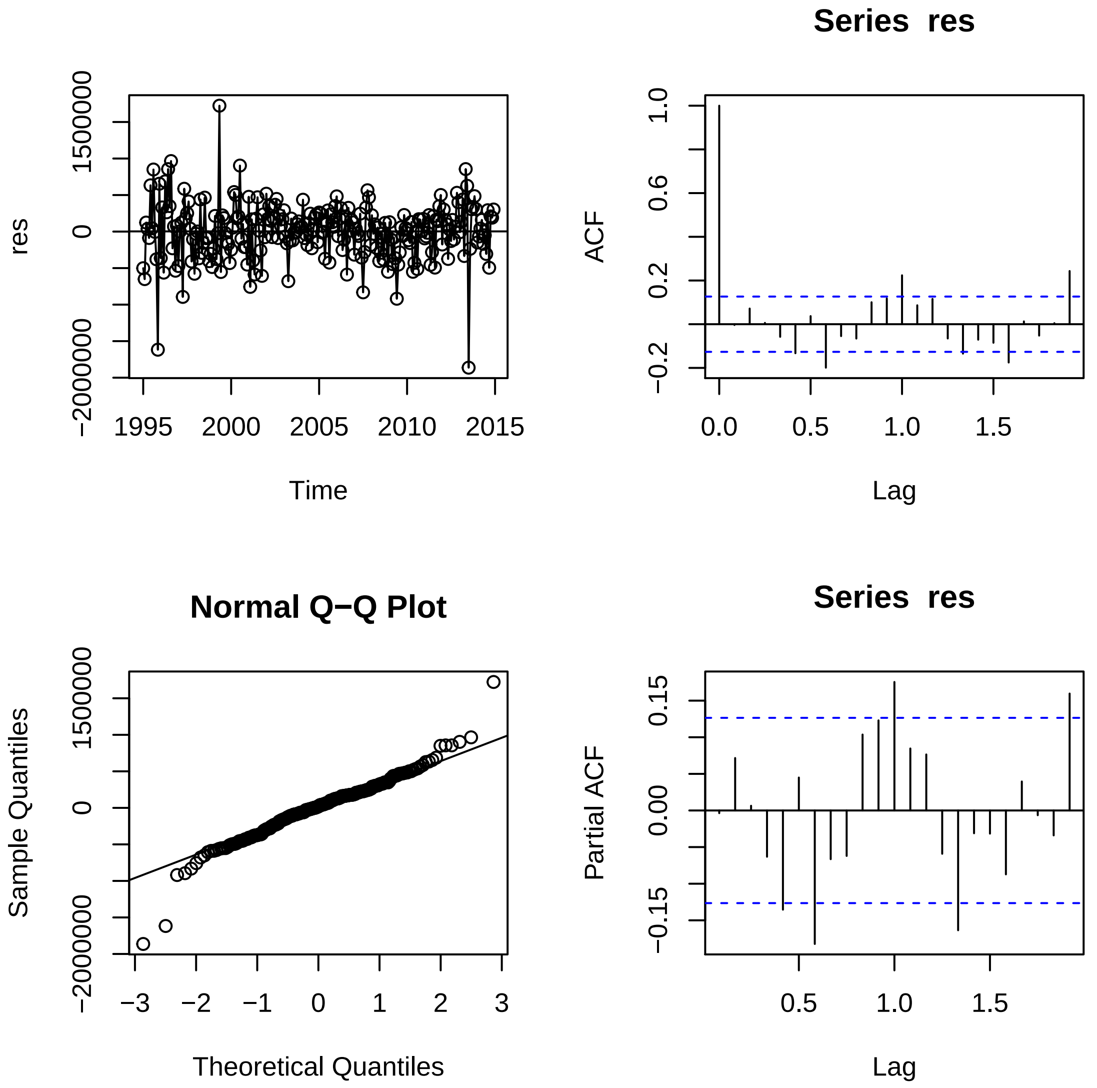
```
##                 Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept) 30883268.11 49894767.58  0.6189681 5.630454e-01
## t.test      -2299425.21  4812173.03 -0.4778351 6.529266e-01
## t2.test        47825.57   115952.24  0.4124592 6.970919e-01
## mth.test2    -289038.39    72492.51 -3.9871482 1.045609e-02
## mth.test3     552868.48    72880.54  7.5859542 6.318480e-04
## mth.test4     565465.60    73310.76  7.7132691 5.846777e-04
## mth.test5    1428647.98    73729.67 19.3768402 6.754219e-06
## mth.test6    1859366.11    74189.92 25.0622471 1.887227e-06
## mth.test7    3480710.50    74846.85 46.5044374 8.683258e-08
## mth.test8    3277265.00    94568.39 34.6549722 3.763651e-07
## mth.test9    1381578.37    95601.47 14.4514339 2.862351e-05
## mth.test10   1000889.49    96172.86 10.4071929 1.411257e-04
## mth.test11    447368.37    96250.36  4.6479657 5.592368e-03
## mth.test12    628549.00    95875.76  6.5558700 1.237529e-03
```

- We can see that all parameters of interact terms have high significant levels, there is strong evidence to reject that they are 0.

And also for actual data fitting and residual diagnostics:

```
par(mfrow=c(1,1))
plot(data.train,main="Fitting of Model 2 + ARMA(1,1)")
points(t+1995, data.regarma.fit, type="l", col="red")
```





- And the ACF/PACF of residual indicate that part of seasonality of data has been not able to be captured by data.
- Therefore, we can conclude that Linear Model + ARMA model could no work perfectly in both model fitting and prediction simultaneously. When the model fits data well, the overfitting happens; when reducing variates to get a better prediction, model's ability to explain the seasonality decreases.
- Therefore, we would turn to fit a SARIMA model.

SARIMA Model

Differencing

Figure 1 Seasonal D = 1

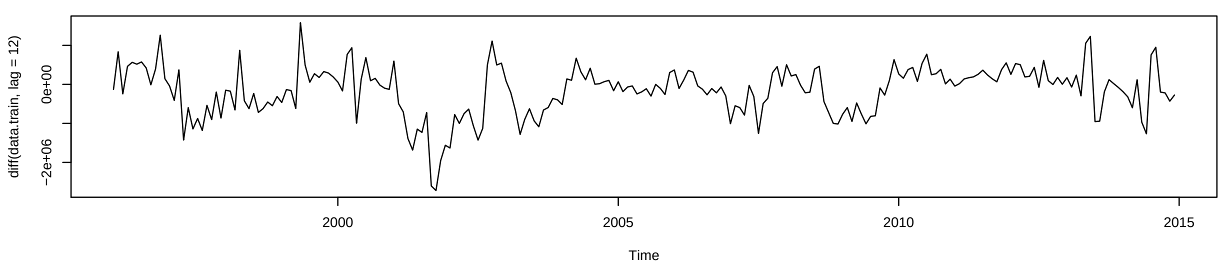


Figure 2 Seasonal D = 1, ordinary d = 1

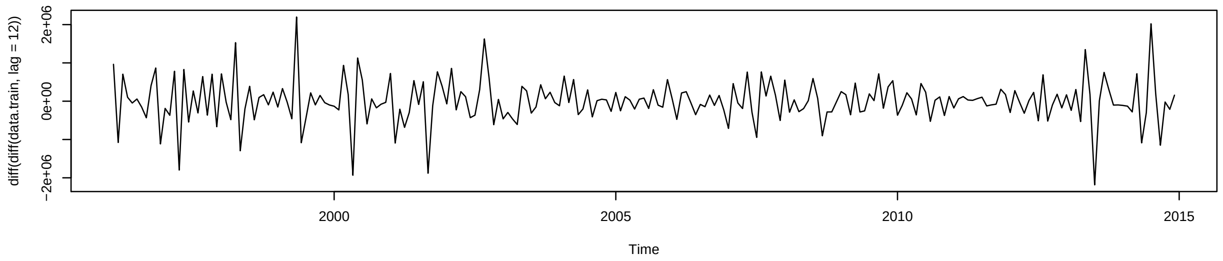
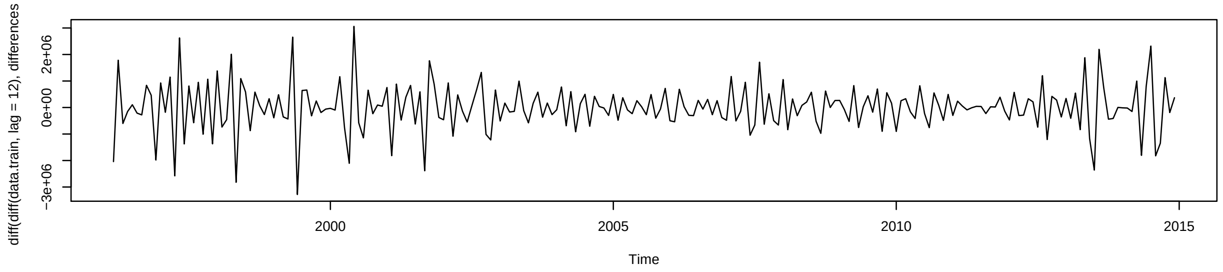


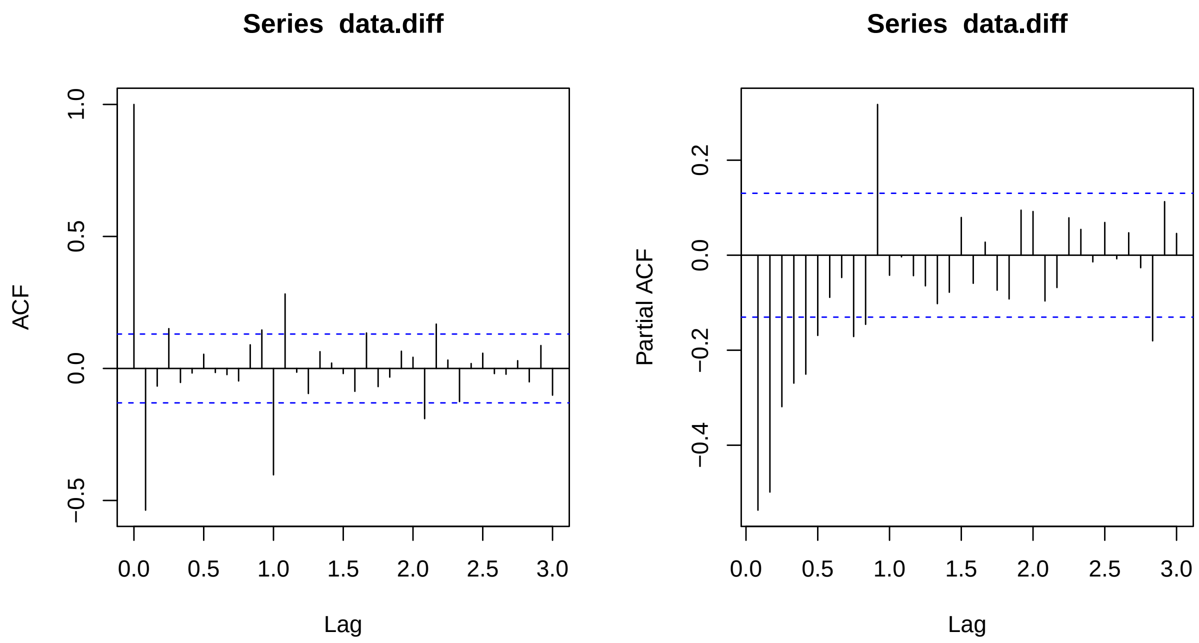
Figure 3 Seasonal D = 1, ordinary d = 2



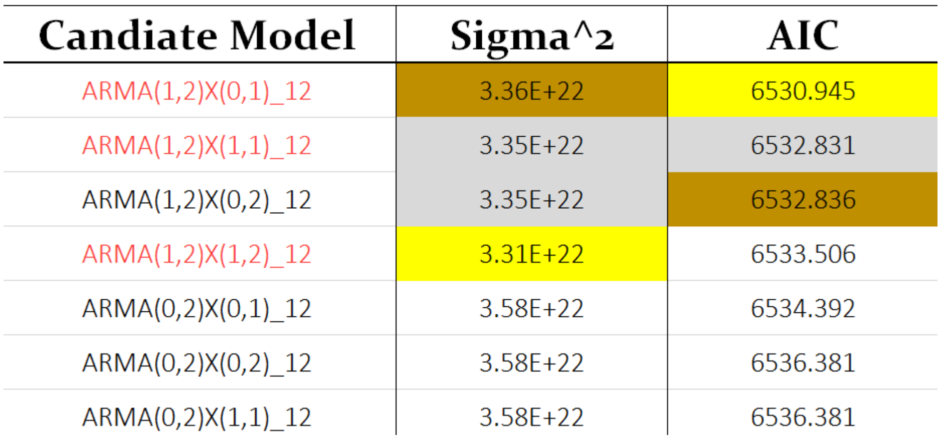
- It can be easily observed, from either the original data plot or the ACF/PACF of linear residual, that the data has a period of 12 months. Therefore, we start a seasonal Differencing by $D = 12$;
- After doing a seasonal difference in Figure 1, we can see that seasonality is well removed, but the trend is definitely not constant. So we apply an ordinary differencing, see Figure 2; The trend getting flat; However, the mean around 2000 to 2003 seems not to be constant. So we apply another ordinary differencing. Now the mean looks constant in Ficure c. However, a risk is that variance is not constant. It flutuates a lot at 1998 to 2003. This is due to the original non-constant variance caused by change point.
- We would like to fit data with a $SARIMA(p, 2, q) \times (P, 1, Q)_{12}$

Model Selection

- Check the ACF an PACF after differencing:

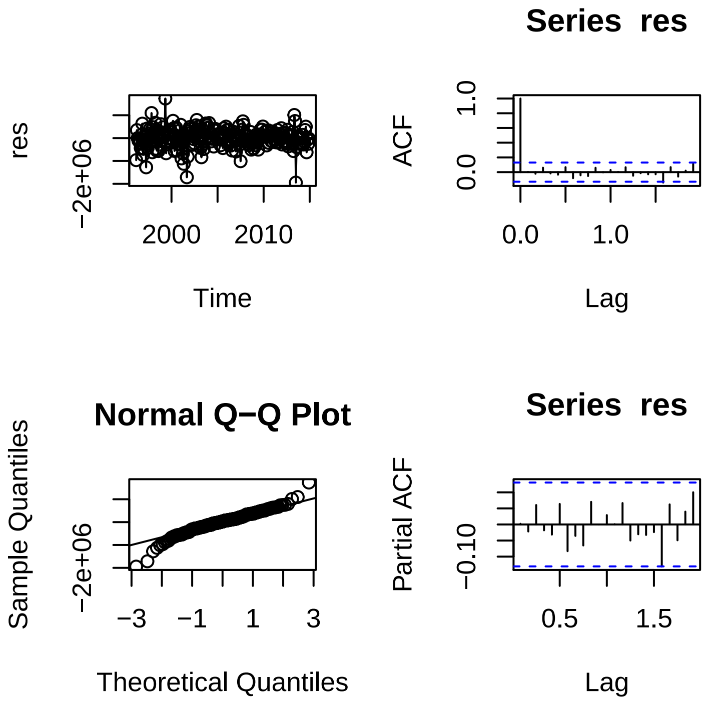


- Based on the shape and intial spikes, a series of candidate models are tested based on Sigmaˆ2 and AIC:

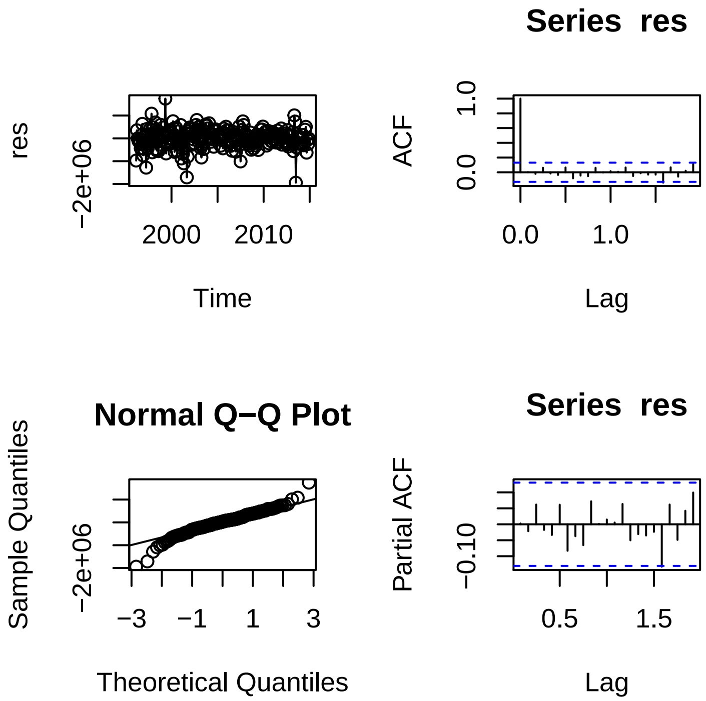


Candiate Model	Sigma^2	AIC
ARMA(1,2)X(0,1)_12	3.36E+22	6530.945
ARMA(1,2)X(1,1)_12	3.35E+22	6532.831
ARMA(1,2)X(0,2)_12	3.35E+22	6532.836
ARMA(1,2)X(1,2)_12	3.31E+22	6533.506
ARMA(0,2)X(0,1)_12	3.58E+22	6534.392
ARMA(0,2)X(0,2)_12	3.58E+22	6536.381
ARMA(0,2)X(1,1)_12	3.58E+22	6536.381

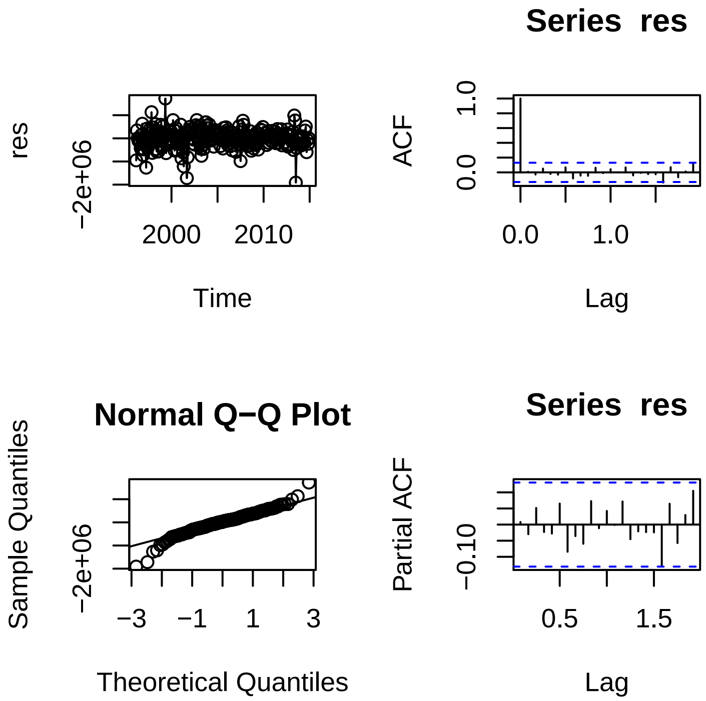
- And then, we would select $ARMA(1,2) \times (0,1)_{12}$, $ARMA(1,2) \times (1,1)_{12}$, $ARMA(1,2) \times (1,2)_{12}$ to carry out residual diagnostics, because they are the most possible candidate for best model.
- $Test\ 1 : \mathbf{ARMA(1,2)} \times \mathbf{(0,1)_{12}}$



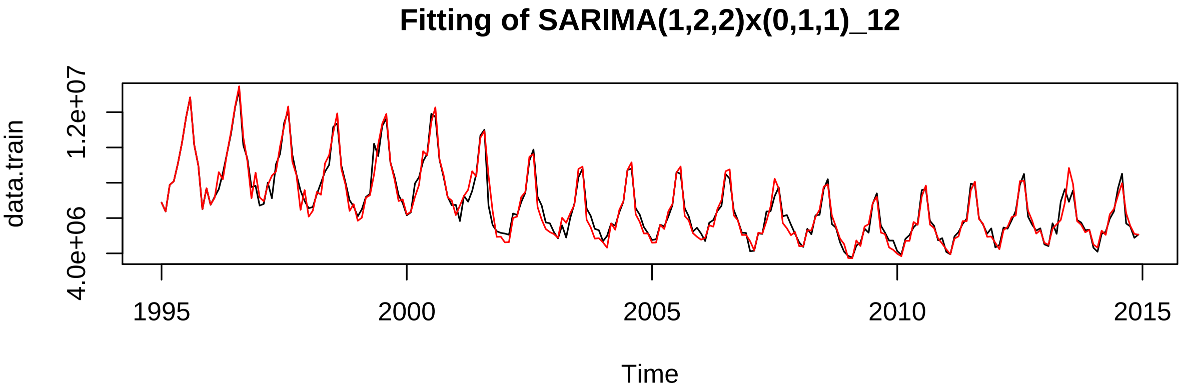
- $Test\ 2 : \mathbf{ARMA(1,2)} \times \mathbf{(1,1)_{12}}$



- $Test3 : \mathbf{ARMA(1,2)} \times \mathbf{(1,2)_{12}}$



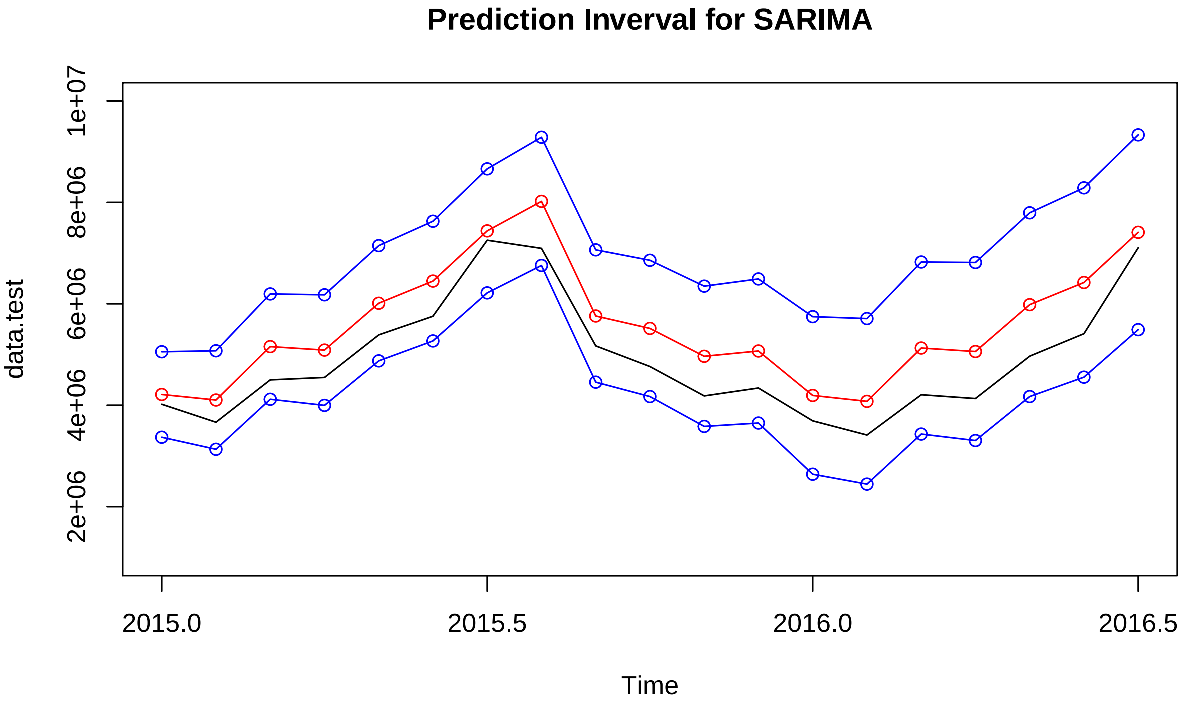
- We can see that three models perform almost the same well in terms of residual diagnostics. We decide to choose $ARMA(1,2) \times (0,1)_{12}$ as our final model. This model performs really well in terms of removing correlation of residual, AIC and lower residual errors. However, it is still not perfect: The qqplot indicates the the residual is a little light tailed.



- We can see that the data is fitted very well in generall, with 2 significant outliers at the end of 2002, and the summer of 2013.

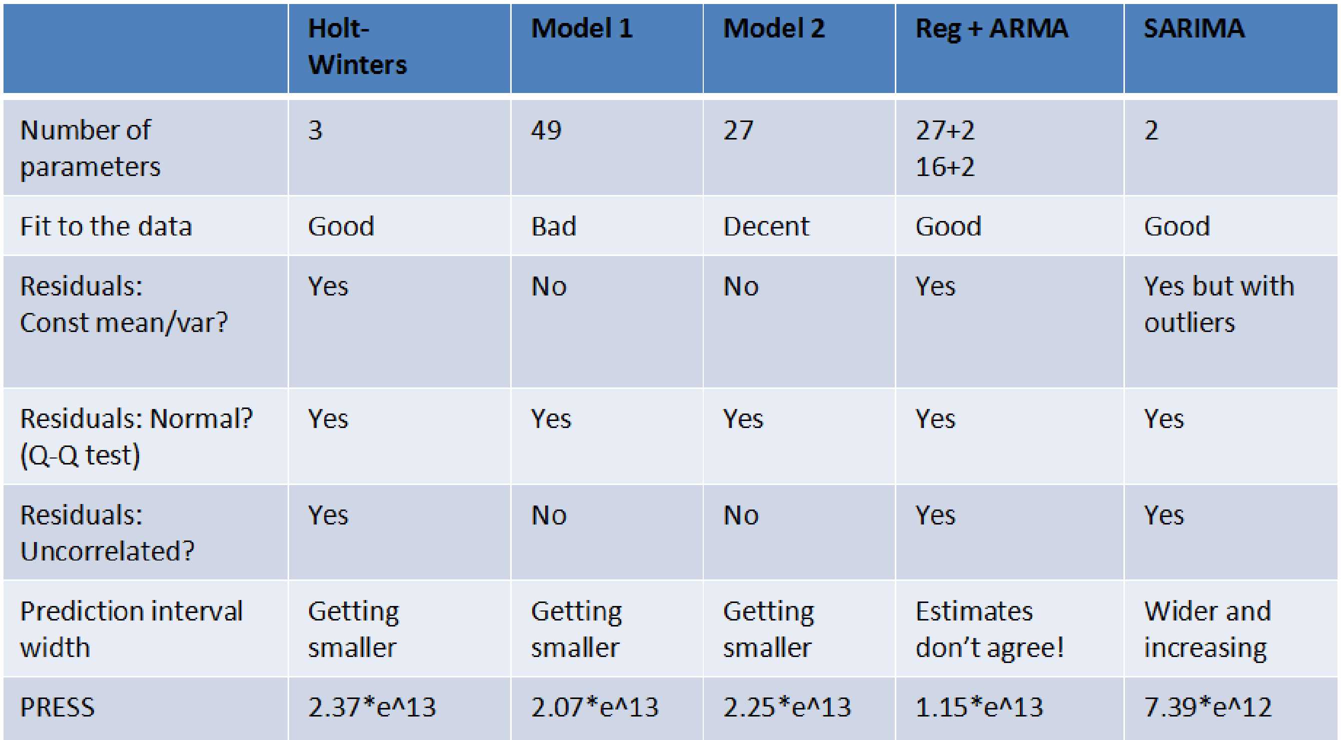
Prediction

- Now we test the prediction ability of the model:



- The model performs no bad in prediction: all testing sets point falls in 95% prediction interval with a decent PRESS value 9.3127e+12. However, the model is obviously overestimate the trend, and the width of interval is increasing.

Statistical Conclusion

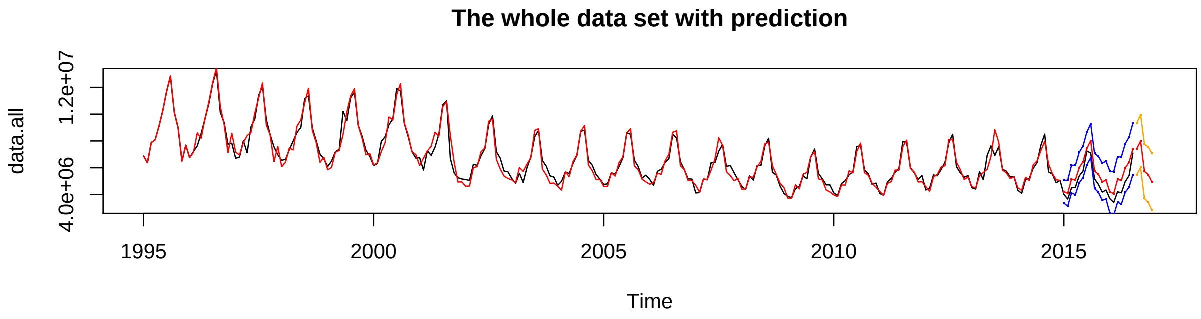


	Holt-Winters	Model 1	Model 2	Reg + ARMA	SARIMA
Number of parameters	3	49	27	27+2 16+2	2
Fit to the data	Good	Bad	Decent	Good	Good
Residuals: Const mean/var?	Yes	No	No	Yes	Yes but with outliers
Residuals: Normal? (Q-Q test)	Yes	Yes	Yes	Yes	Yes
Residuals: Uncorrelated?	Yes	No	No	Yes	Yes
Prediction interval width	Getting smaller	Getting smaller	Getting smaller	Estimates don't agree!	Wider and increasing
PRESS	2.37*e^13	2.07*e^13	2.25*e^13	1.15*e^13	7.39*e^12

Based on the comparison above, the SARIMA model is the best model, considering both model fitting and prediction intervals. Since we are focusing on future predictions the lowest PRESS is the deciding factor for choosing the model.

Conclusion

The model (SARIMA) chosen predicts border crossings fairly accurately when tested against the data from Jan 2015 - Jul 2016. It captures both the seasonal and temporal trend of the data. The predictions are, however, consistently higher than the actual number of US Canada border crossings of the testing set and is over estimated. Therefore we expect our future predictions to be over estimated as well.



The next five prediction : Aug.: 7410112 Sept.:7992489 Oct.: 5732052 Nov.:5487692 Dec.:4938829

The model is simplistic and concise, with minimal parameters. It would be easy to implement. However, as more time passes, if no new data is added, the prediction intervals increase. Therefore, in order for the model predictions to be of value, continuous updating the dataset is required.

Reference

"BTS | Border Crossing/Entry Data: Query Detailed Statistics." BTS | Border Crossing/Entry Data: Query Detailed Statistics. US Bureau of Transportation, n.d. Web. 27 Nov. 2016. https://transborder.bts.gov/programs/international/transborder/TBDR_BC/TBDR_BCQ.html.

Appendix - RCode

```
mydata <- read.table("JustinBieberIsMyHero_data.txt", header=T)
data.all <- ts(mydata$All_Border[0:259], start=c(1995, 1), frequency = 12)
data.training <-ts(mydata$All_Border[0:241], start = c(1995, 1), frequency =12)
data.test <-ts(mydata$All_Border[242:259], start = c(2016, 1), frequency =12)

par(mfcol=c(1,1))
plot(data.all)
abline(v=2015, col="blue")
abline(v=2001.7,col = "red")

plot(decompose(data.training))


turning.point.test <- function(ts)
{
  n <- length(ts)
  mu <- 2*(n-2)/3
  var <- (16*n-29)/90
  x <- embed(ts,3)
  test.sum <- sum((x[,2] > x[,1] & x[,2] > x[,3]) | (x[,2] < x[,1] & x[,2] < x[,3])) # this is a boolea
  test.stat <- abs(test.sum-mu)/sqrt(var)
  p.value <- 2*(1-pnorm(test.stat))
  structure(list(test.sum=test.sum, mu=mu,var=var,test.stat=test.stat,p.value=p.value,reject.null=(test
}
turning.point.test(data.training)

qqnorm(data.training)
qqline(data.training)

par(mfcol=c(2,1))


values <- decompose(data.training)$random[!is.na(decompose(data.training)$random)]

par(mfrow=c(1,2))
acf(values)
pacf(values)

data <- read.table("JustinBieberIsMyHero_data.txt")
colnames(data) <- as.character(unlist(data[1,]))
data = data[-1, ]

buffalo <- ts(data$Buffalo_Ppl, start=c(1995, 1), end=c(2016, 7), frequency=12)
all <- ts(data$All_Border, start=c(1995, 1), end=c(2016, 7), frequency=12)

## Moving Average Filter
MAsmooth <- function(series, q)
{
  c = 1/(2*q+1) # constant to multiply by
  series.MA <- series # starting point Xt
  for(i in 1:q){ series.MA <- series.MA + lag(series, k=-i) + lag(series, k=i) } # adding Xt-i and Xt+i
  series.MA <- series.MA*c # multiplying by the constant
  return(series.MA)
}

plot(all)
lines(MAsmooth(all, 2), col="red")
lines(MAsmooth(all, 3), col="dark green")


par(mfcol=c(2,2))
model1 <-HoltWinters(all, seasonal = c("additive")) #Additive HW method with trend
model2 <-HoltWinters(all,seasonal = c("multiplicative")) #Multiplicative HW method with trend
model3 <-HoltWinters(all,beta=F, seasonal = c("additive")) #Additive HW without trend
model4 <-HoltWinters(all, beta=F,seasonal = c("multiplicative")) #Multiplicative HW without trend
plot(model1, predict(model1,n.ahead=12),main = "HW Additive Seasonal with Trend Parameter")
plot(model2, predict(model2,n.ahead=12),main = "HW Multiplicative Seasonal with Trend Parameter")
plot(model3, predict(model3,n.ahead=12),main = "HW Additive Seasonal without Trend Parameter")
plot(model4, predict(model4,n.ahead=12),main = "HW Multiplicative Seasonal without Trend Parameter")

model1$SSE
model2$SSE
model3$SSE
model4$SSE

runs.test <- function(ts)
{
n <- length(ts)
n.pos <- sum(ts > 0)
n.neg <- n - n.pos
ts.binary <- ts
ts.binary[ts > 0] <- 1
ts.binary[ts < 0] <- -1
n.runs <- sum(abs(diff(ts.binary)) > 0) + 1
mu <- 2*n.pos*n.neg/n + 1
var <- (mu-1)*(mu-2)/(n-1)
test.stat <- abs(n.runs-mu)/sqrt(var)
p.value=2*(1-pnorm(test.stat))
structure(list(n.runs=n.runs,n.pos=n.pos,n.neg=n.neg,mu=mu,var=var,test.stat=test.stat,p.value=p.value,
}

difference.sign.test <- function(ts)
{
n <- length(ts)
mu <- (n-1)/2
var <- (n+1)/12
test.sum <- sum(diff(ts) > 0)
test.stat <- abs(test.sum-mu)/sqrt(var)
p.value <- 2*(1-pnorm(test.stat))
structure(list(test.sum=test.sum,mu=mu,var=var,test.stat=test.stat,p.value=p.value,reject.null=(test.sta
}

turning.point.test <- function(ts)
{
```

```
n <- length(ts)
mu <- 2*(n-2)/3
var <- (16*n-29)/90
x <- embed(ts,3)
test.sum <- sum((x[,2] > x[,1] & x[,2] > x[,3]) | (x[,2] < x[,1] & x[,2] < x[,3])) # this is a boolean
test.stat <- abs(test.sum-mu)/sqrt(var)
p.value <- 2*(1-pnorm(test.stat))
structure(list(test.sum=test.sum, mu=mu,var=var,test.stat=test.stat,p.value=p.value,reject.null=(test.s
}

res3<-resid(model3)

ts.plot(res3,main="Residuals for Additive HW(Without Trend)",ylab="Resid")

qqnorm(res3, main="QQ plot for Additive HW(Without Trend)")
qqline(res3)

par(mfrow=c(1,2))
acf(res3,main="ACF for Additive HW(Without Trend)")
pacf(res3,main="ACF for Additive HW(Without Trend)")

runs.test(res3)
shapiro.test(res3)
difference.sign.test(res3)
turning.point.test(res3)

par(mfcol=c(1,1))
predict_s <- predict(model3, 12, prediction.interval = T, level = 0.90)
plot(model3,predict_s, main="Forecast for Additive HW(without trend)")

library(MASS)
data <- read.table("JustinBieberIsMyHero_data.txt",header = TRUE)
data <- data[,3]
data.total <- ts(data,start=c(1995,1),frequency = 12)
data.test <- ts(data[241:259],start = c(2015,1),frequency = 12)
data.train <- ts(data[1:240],start = c(1995,1),frequency = 12)
time <- time(data.train)
time2 <- time^2
month <- as.factor(cycle(data.train))
chg = c(rep(0,80),rep(1,160))
time2 <- time^2
reg1 <- lm(data.train~time*month*chg)
reg2 <- lm(data.train~time*month*chg+time2)
reg3 <- lm(data.train~time*month + time*chg+time2)
t.test2 <- time(data.test)^2
t.test <- time(data.test)
month.test <- as.factor(cycle(data.test))
chg.test <- rep(1,19)

plot(data.train)
points(time, reg1$fitted, type='l', col="red")
plot(reg1$residuals, type="l")
points(reg1$residuals)
```

```
abline(0,0)

plot(data.train)
points(time, reg2$fitted, type='l', col="red")
plot(reg2$residuals, type="l")
points(reg2$residuals)
abline(0,0)
#AIC=6248.1

plot(data.train)
points(time, reg3$fitted, type='l', col="red")
#AIC=6237.57

par(mfcol=c(2,2))
plot(reg3$residuals, type="l")
points(reg3$residuals)
abline(0,0)
plot(reg3$fitted, reg3$residuals)
qqnorm(reg3$residuals)
qqline(reg3$residuals)
acf(reg3$residuals)

par(mfcol=c(1,2))
reg1.pred <- predict(reg1,newdata = list(time=t.test,month=month.test,chg=chg.test),interval = "predicti
plot(data.test,main="Model 1")
points(t.test, reg1.pred[,1], type='l', col="red")
points(t.test, reg1.pred[,2], type='l', col="blue")
points(t.test, reg1.pred[,3], type='l', col="blue")
points(t.test, reg1.pred[,1], col="red")
points(t.test, reg1.pred[,2], col="blue")
points(t.test, reg1.pred[,3], col="blue")
reg3.pred <- predict(reg3,newdata = list(time=t.test,month=month.test,chg=chg.test,time2=t.test2),inter
plot(data.test,main="Model 2")
points(t.test, reg3.pred[,1], type='l', col="red")
points(t.test, reg3.pred[,2], type='l', col="blue")
points(t.test, reg3.pred[,3], type='l', col="blue")
points(t.test, reg3.pred[,1], col="red")
points(t.test, reg3.pred[,2], col="blue")
points(t.test, reg3.pred[,3], col="blue")

data<-read.table("JustinBieberIsMyHero_data.txt",header = T)
data.all <- ts(data$All_Border, start=1995, frequency=12)
data.train <-ts(data$All_Border[1:240],start = 1995, frequency=12)
data.test <-ts(data$All_Border[241:259],start = 2015, frequency=12)
resdiags <- function(res) # you give this function a vector containing residuals from a model
{
  par(mfcol=c(2,2)) # splits the view to show 4 plots
  ts.plot(res) # time series plot of residuals
  points(res) # points to make counting runs easier
  abline(h=mean(res)) # mean line
  qqnorm(res) #qq plot
  qqline(res)
  acf(res) #acf
  acf(res, type="partial") #pacf
}

t<-time(data.train)-1995
t2 <- t^2
mth <- as.factor(cycle(data.train))
chg = c(rep(0,80),rep(1,160))
data.reg<-lm(data.train~t2 +t*mth+ chg*t)
data.reg0<-lm(data.train~t+t2+mth)
par(mfrow = c(1,2))
acf(data.reg$residuals,main = "ACF for Model 2") #acf
acf(data.reg$residuals, main  = "PACF for Model 2",type="partial")

par(mfrow=c(2,1))

X <- model.matrix(data.reg)
data.regarma <- arima(data.train, order=c(1,0,0), xreg=X[,2:27])
data.regarma.fit <- data.train - data.regarma$res # making fitted values
plot(data.train,main="Fitting of Model 2 + MA(1)")
points(t+1995, data.regarma.fit, type="l", col="red")


data.regarma2 <- arima(data.train, order=c(1,0,1), xreg=X[,2:27])
data.regarma.fit2 <- data.train - data.regarma2$res # making fitted values
plot(data.train,main="Fitting of Model 2 + ARMA(1,1)")
points(t+1995, data.regarma.fit2, type="l", col="red")

resdiags(data.regarma$res)


resdiags(data.regarma2$res)

##Correlation is almost in accetable area, but some out of 95% with seasonal pattern.
##Have to use SARIMA

t.test <- time(data.test) - 1995
t2.test <- t.test^2
mth.test <- as.factor(cycle(data.test))
chg.test <- rep(1, 19)
temp <- lm(data.test~t2.test +t.test*mth.test+ chg.test*t.test)
X.test <- model.matrix(temp)
pred.regarma <- predict(data.regarma, n.ahead=19, newxreg=X.test[,2:27])
plot(data.test,ylim=c(1e+6,10e+6),main="Prediction Interval for Model2 with ARMA(1,1) Res")
points(t.test+1995,temp$fitted,col = "blue",pty = 19)
points(pred.regarma$pred, type='l', col="red")
points(pred.regarma$pred + 1.96*pred.regarma$se, type='l', col="blue")
points(pred.regarma$pred - 1.96*pred.regarma$se, type='l', col="blue")
points(pred.regarma$pred, col="red")
points(pred.regarma$pred + 1.96*pred.regarma$se, col="blue")
points(pred.regarma$pred - 1.96*pred.regarma$se, col="blue")
#sum((data.test - pred.regarma$pred)^2) # PRESS
### Some data is out of bound. Predictablity is worse than linear prediction
```

```
t<-time(data.train)-1995
t2 <- t^2
mth <- as.factor(cycle(data.train))
chg = c(rep(0,80),rep(1,160))
data.reg<-lm(data.train~t+t2+mth + chg*t)
X <- model.matrix(data.reg)
data.regarma <- arima(data.train, order=c(1,0,1), xreg=X[,2:15])
data.regarma.fit <- data.train - data.regarma$res # making fitted values

t.test <- time(data.test) - 1995
t2.test <- t.test^2
mth.test <- as.factor(cycle(data.test))
chg.test <- rep(1, 19)
temp <- lm(data.test~t.test+t2.test+mth.test+chg.test*t.test)
X.test <- model.matrix(temp)
pred.regarma <- predict(data.regarma, n.ahead=19, newxreg=X.test[,2:15])
plot(data.test,ylim=c(1e+6,10e+6),main="Prediction for Simplified Linear model + ARMA(1,1) Res")
points(t.test+1995,temp$fitted,col = "blue",pty = 19)
points(pred.regarma$pred, type='l', col="red")
points(pred.regarma$pred + 1.96*pred.regarma$se, type='l', col="blue")
points(pred.regarma$pred - 1.96*pred.regarma$se, type='l', col="blue")
points(pred.regarma$pred, col="red")
points(pred.regarma$pred + 1.96*pred.regarma$se, col="blue")
points(pred.regarma$pred - 1.96*pred.regarma$se, col="blue")

summary(temp)$coef

par(mfrow=c(1,1))
plot(data.train,main="Fitting of Model 2 + ARMA(1,1)")
points(t+1995, data.regarma.fit, type="l", col="red")


resdiags(data.regarma$res)
#data.reg0<-lm(data.train~t+t2+mth)
#plot(data.reg$residuals)
#par(mfrow=c(2,1))

data<-read.table("JustinBieberIsMyHero_data.txt",header = T)
data.all <- ts(data$All_Border, start=1995, frequency=12)
data.train <-ts(data$All_Border[1:240],start = 1995, frequency=12)
data.test <-ts(data$All_Border[241:259],start = 2015, frequency=12)
resdiags <- function(res) # you give this function a vector containing residuals from a model
{
  par(mfcol=c(2,2)) # splits the view to show 4 plots
  ts.plot(res) # time series plot of residuals
  points(res) # points to make counting runs easier
  abline(h=mean(res)) # mean line
  qqnorm(res) #qq plot
  qqline(res)
  acf(res) #acf
  acf(res, type="partial") #pacf
}
```

```
#Differencing the data
par(mfcol=c(3,1))
plot(diff(data.train, lag=12), main = "Figure 1 Seasonal D = 1")
plot(diff(diff(data.train, lag=12)), main = "Figure 2 Seasonal D = 1, ordinary d = 1")
plot(diff(diff(data.train, lag=12), differences=2), main = "Figure 3 Seasonal D = 1, ordinary d = 2")
#After one seasonal difference, the pattern is removed but still a drifting trend After adding an ordin

data.diff <- diff(diff(data.train, lag=12), differences=2)
par(mfcol=c(1,2))
acf(data.diff, lag.max=36)
acf(data.diff, type="p", lag.max=36)

knitr::include_graphics("crop1.png")

diff.arma1 <- arima(data.diff,order=c(1,0,2),seasonal=list(order=c(0,0,1), period=12))
diff.arma2 <- arima(data.diff,order=c(1,0,2),seasonal=list(order=c(1,0,1), period=12))
diff.arma3 <- arima(data.diff,order=c(1,0,2),seasonal=list(order=c(1,0,2), period=12))

resdiags(diff.arma1$res)

resdiags(diff.arma2$res)

resdiags(diff.arma3$res)

data.sarima <- arima(data.train,order=c(1,2,2),seasonal=list(order=c(0,1,1),frequency=12))
data.sarima.fit <-data.train - data.sarima$res # creating fitted values
plot(data.train, main = "Fitting of SARIMA(1,2,2)x(0,1,1)_12")
points(data.sarima.fit, type="l", col="red")

pred.sarima <- predict(data.sarima, n.ahead=19)
plot(data.test,ylim=c(1e+6,10e+6),main = "Prediction Inverval for SARIMA")
points(pred.sarima$pred, type='l', col="red")
points(pred.sarima$pred + 1.96*pred.sarima$se, type='l', col="blue")
points(pred.sarima$pred - 1.96*pred.sarima$se, type='l', col="blue")
points(pred.sarima$pred, col="red")
points(pred.sarima$pred + 1.96*pred.sarima$se, col="blue")
points(pred.sarima$pred - 1.96*pred.sarima$se, col="blue")
#sum((data.test - pred.sarima$pred)^2) # PRESS

knitr::include_graphics("table.png")

plot(data.all,ylim=c(3e+6,13e+6),xlim =c(1995,2017),main = "The whole data set with prediction")
pred.sarima1 <- predict(data.sarima, n.ahead=23)
pred.val <- pred.sarima1$pred[19:23]
pred.se <- pred.sarima1$se[19:23]
points(data.sarima.fit, type="l", col="red")
points(pred.sarima$pred, type='l', col="red")
points(pred.sarima$pred + 1.96*pred.sarima$se, type='l', col="blue")
points(pred.sarima$pred - 1.96*pred.sarima$se, type='l', col="blue")
points(pred.sarima$pred, col="red",pch = ".")
points(pred.sarima$pred + 1.96*pred.sarima$se, col="blue",pch = ".")
points(pred.sarima$pred - 1.96*pred.sarima$se, col="blue",pch = ".")
t<-c(2016.583,2016.667,2016.750,2016.833,2016.917)
```

```r
points(t,pred.val, type='l', col="red")
points(t,pred.val + 1.96*pred.se, type='l', col="orange")
points(t,pred.val - 1.96*pred.se, type='l', col="orange")
points(t,pred.val, col="red",pch = ".")
points(t,pred.val + 1.96*pred.se, col="orange",pch = ".")
points(t,pred.val - 1.96*pred.se, col="orange",pch = ".")
#sum((data.test - pred.sarima$pred)^2) # PRESS
```