Toronto Temperature Forecasting

Group D

Member:

Yunhong Sun 20489589 :Motivation and Introduction, data research
Yilin Zhu 20374185:General analysis of Data
Tianlu Zhu 20476679: Regression, Smoothing Method, Box-Jenkins Methods, Appendix
Songzhen Lu 20453733 Regression, Smoothing Method, Box-Jenkins Methods, Appendix
Xiaoxin Chen 20494581: Statistical Conclusion and General Conclusion

yun

Contents

1 Motivation and introduction:

When first looking for the project topic, the aspectsthe team cared about the most were the application and analysis of data. The idea was trying to look for data that was not directly affected by humans (i.e. policy changes). In this way, the first idea that came to mind was local temperature. There were only two variables: time and temperature, which provided convenient and efficient ways for future analysis.

2 Data:

The data was searched from bloomberg, and was split into daily temperature from March 5th, 1995 to March 4th, 2015. Since not every year has 29 days in February, and also for predicting futher data conveniently, the data of February 29this dropped to ensure there are 365 days every year.

There was one problem in the data: all the temperatures are approximated to two decimal places except for the data from 2012, the data from 2012 was rounded up to integers. But since there was twenty years of data, this will not be a problem for further analysis.

The team divided the data into two parts: the data from 1995 to 2012 as the training part, and the remaining two years as the testing part.

Figure 1 named “20 year daily temp. in Toronto” shows the whole plot of the dataset, which shows very clear seasonality: high temperature in several months and low temperature in several months. But it seems there is no obvious trend of the temperature based on this plot.

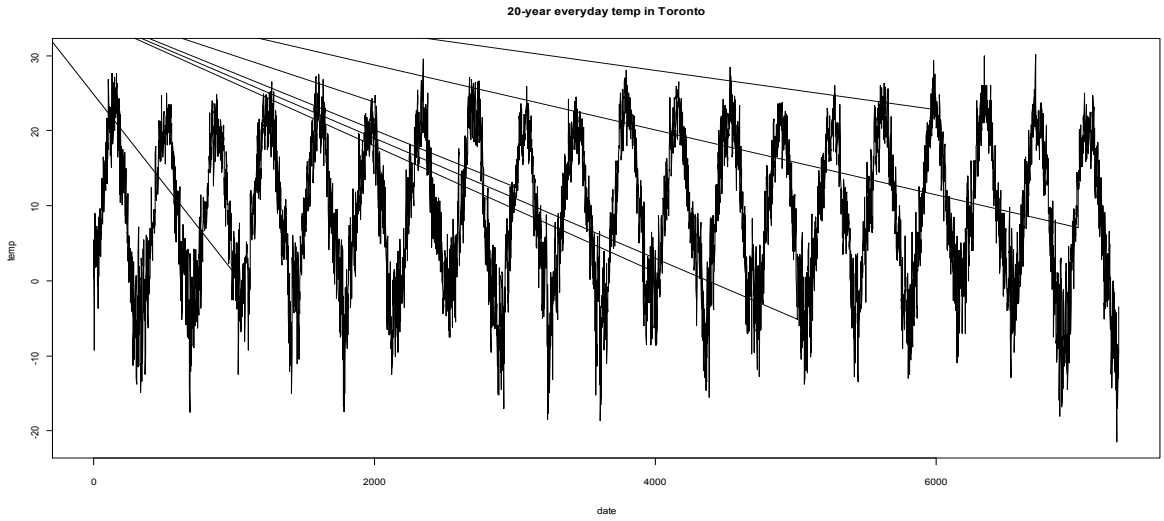


Figure 1

For clearly watching one-year temperature change, the team plotted the Figure 2 named “20 years’ same day temp. avg. in Toronto”. It was done by calculating the average temperature of the same day over 20 years, and showing the 365-day average temperature in one picture. From this graph, there is a clear seasonality.

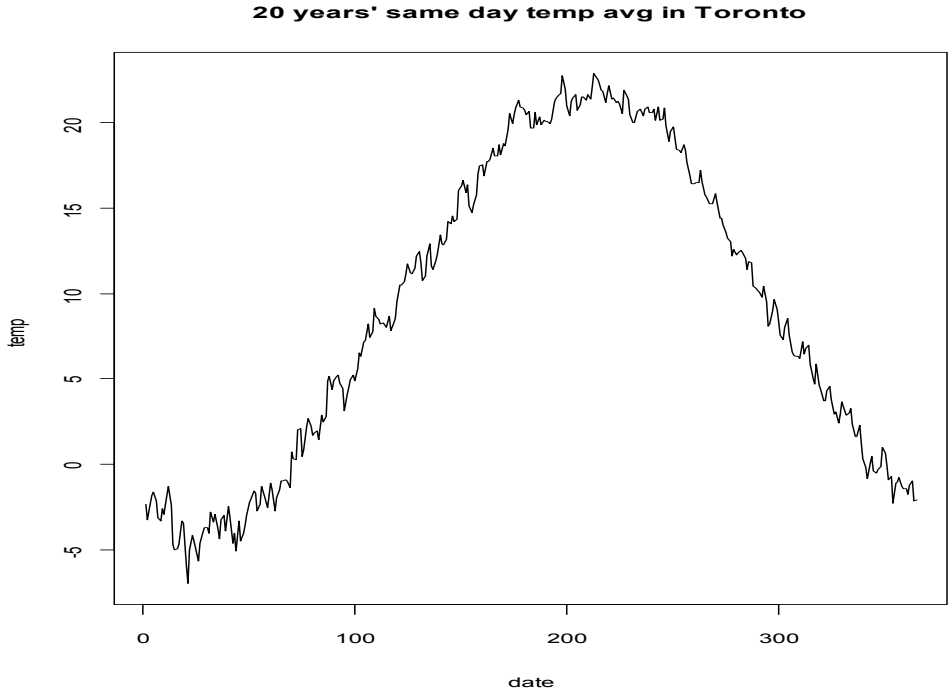


Figure 2

Since in the whole plot graph, it is hard to see the trend of the data, the Figure 3--"lowest and highest avg. temp. over 20 years" has been plotted. After calculating the average temperature of every month, conclusion is made that the extreme temperatures occur in July, August, January and February. An average temperature of the hottest two months (July and August) and coldest two months (January and February) over 20 years has been plotted (Figure 3), in order to indicate if there is any trend. It could be seen that neither the highest temperature nor lowest temperature have a clear trend in this figure, but there were some fluctuations. Therefore, the team would still consider the trend in the further analysis.

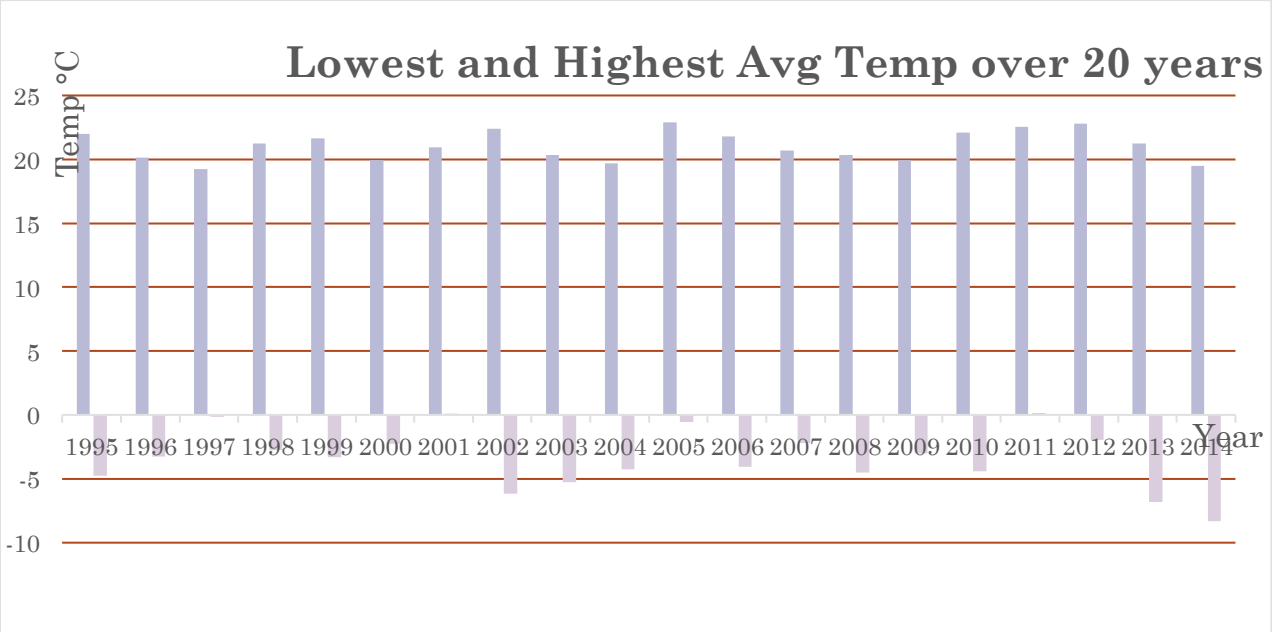


Figure 3

3 Regression

3.1 Linear Regression

First, the data is fitted by monthly factor model. The residual plots and other analysis are shown as below:

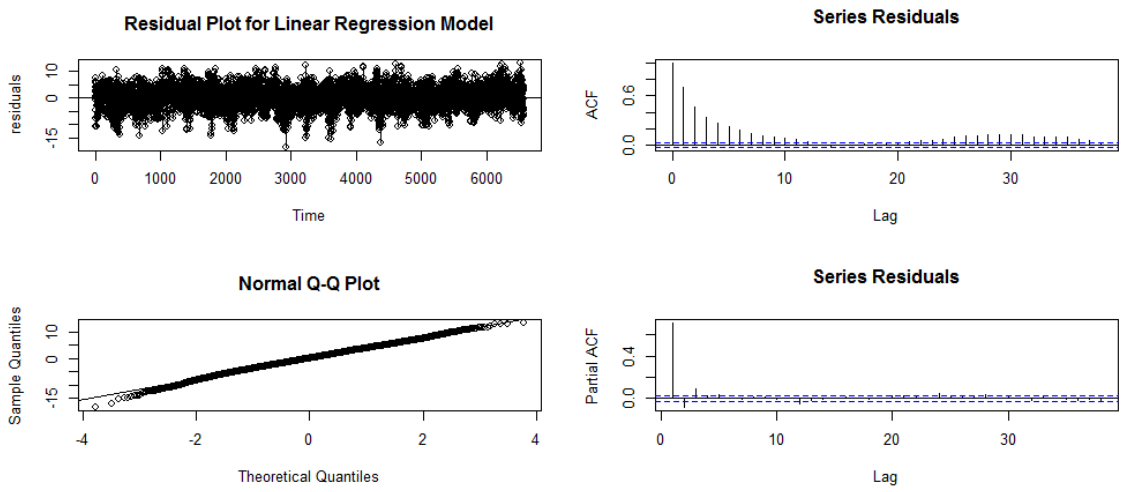


Figure 4

The residuals look random, but acf and pacf plots suggest the non-stationality of the linear model residuals. The Q-Q Plot seems to have a good fit, which does not violate the normality assumption.

Next, a yearly increasing trend is added to the model. Figure 5 illustrates the new residual analysis:

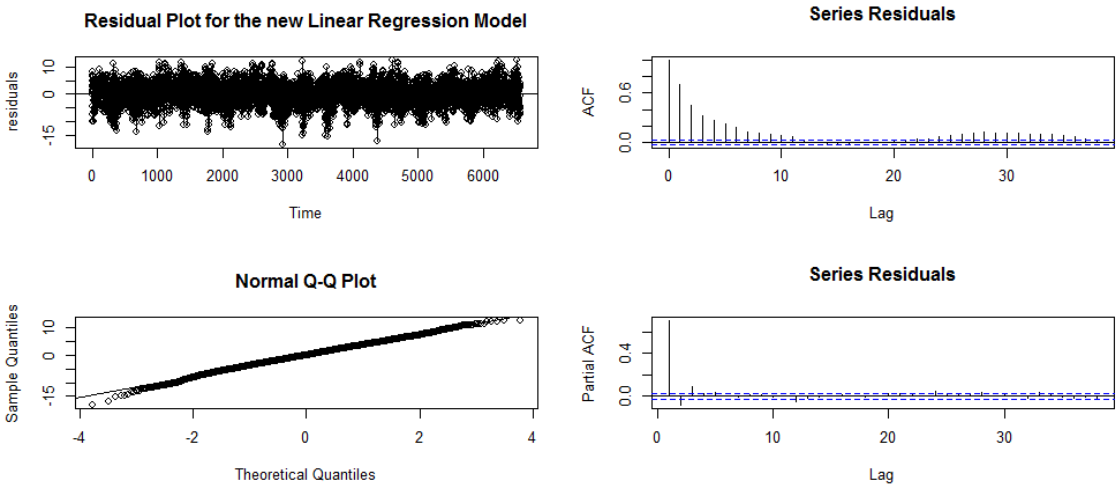


Figure 5

It does not seem to have a big improvement on the model, but by performing residual tests, the result is barely passed. Therefore the new trend is included in the model.

3.2 Other Models and Transformations

To fit the data better by regression model, square-root and log transformation are considered. However, by carefully applying residual tests, these transformations do not improve the model a lot, and in model summaries, the p-value for intercepts are both more than 0.05. Thus these two transformations are not applied in the model.

In addition, the daily effect factor is being considered to be added to the model. Nevertheless, the improvement is too little to be contained into the regression model. The detailed analysis can be found in appendix, but this parameter is not in the final model as well.

3.3 Data fitting by Linear Regression Model

In order to remove the trend with time, linear regression is applied to remove the seasonality of the data. The linear model with monthly and yearly factors is picked by comparing how well the data is fitted among a few models.

Using the linear model to predict the temperature from March 5, 2013 to March 4, 2015, 88.77% of the testing set data are in the 95% prediction interval. Figure 6 can illustrate the fitness of this model. The red line is the estimated value, while the orange lines are the upper and lower bound of the prediction interval. Most of the data are included in the prediction intervals.

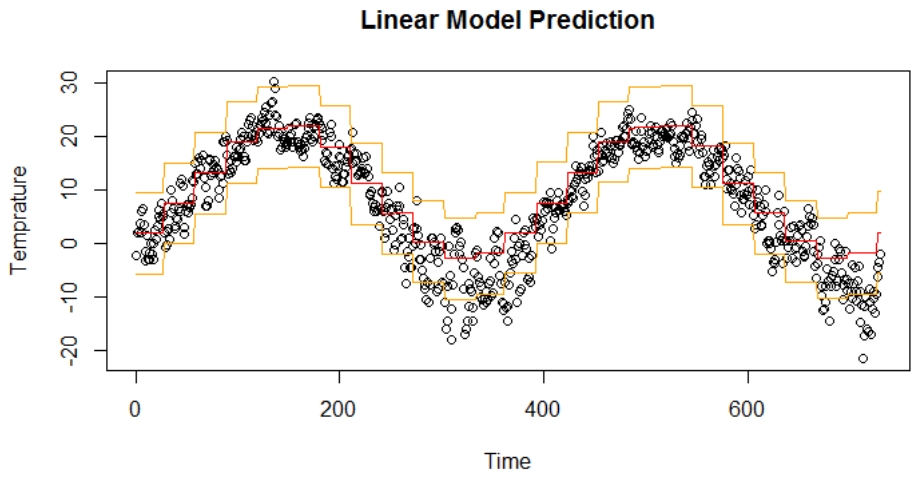
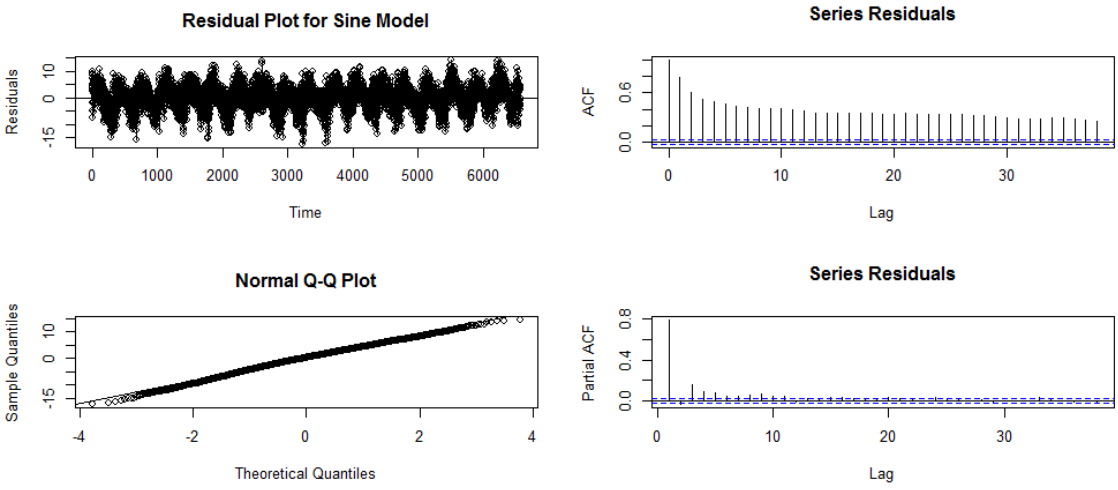


Figure 6

3.4 Sine Model

Because of its seasonality, a sine model is proposed by the Professor. However, after fitting the model into the data, the team rejects the suggestion.



The residual analysis can prove this argument. There is clear seasonality in the residuals, which by inspection the residual plot is quite seasonal after fitting the sine model. Moreover, the ACF plot indicates strong correlation between data points, and the Q-Q plot shows light tail on the one side while heavy tail on the other. In conclusion, the sine model is not considered in the final model. This feature can also be confirmed by the original data plot. The ideal sinusoidal plot has smooth curves

when changing directions, but in this data set it is pointier at the local maximum and local minimum point. As a result, the sine model is not used in the final model.

4 Smoothing method

4.1 Holt-Winter Model (addictive)

Since there are some 0 values (as 0 temperature) in the data set, the multiplicative model cannot be applied here. Fitting the Holt-Winter Model, the sum of squared error of prediction (SSE) is 71097.16, suggesting a large sum of errors in this model.

This affect can also be determined from Figure 7. Same as before, the red line shows the estimated values by the Holt-Winter model, which follows the overall trend of the data but is not fitting the trend quite closely. Meanwhile, the variance is quite large, resulting in a wide prediction interval, which tends to go beyond the scope of the plot quickly.

4.2 Data Fitting by Holt-Winter Model

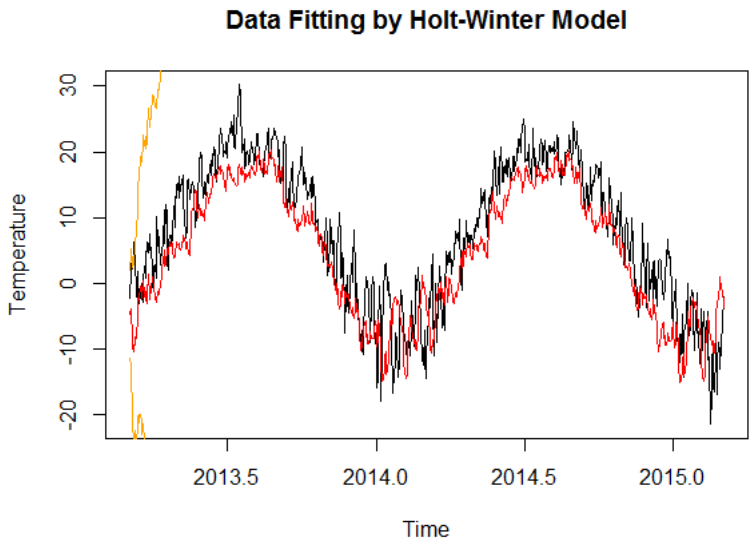


Figure 7

5 Box-Jenkins models

5.1 ARIMA/SARIMA Models

Clearly, the data contains seasonality as temperature heavily depends on the day of the year. On the other hand, there are seasonality trends other than lag=365, which suggests the relationship between the same day of different years. By close analysis the data is only differenced by lag=365, which the decision process can be found in appendix. The proposed model is ARIMA(2,1,2), and the resulted residual plots are shown as below:

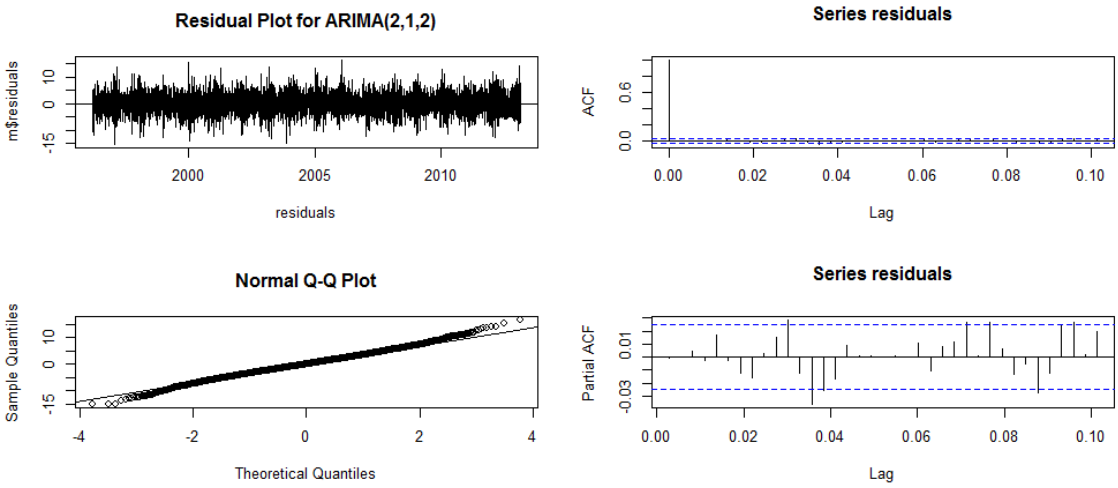


Figure 8

The ACF and PACF plots show only one significant spike at lag=0, which is pretty good compared to other graphs. Moreover, the Q-Q plot also claims normal residuals after fitting this model.

5.2 ARCH/GARCH Model

The ARCH/GARCH models are considered to combine with the regression model, but they do not do a better job as the proposed ARIMA model above. Therefore it is not included in the final model. The analysis process in the appendix will provide detailed information about the reason of rejecting the model.

5.3 Data Fitting by Linear Regression & ARIMA model

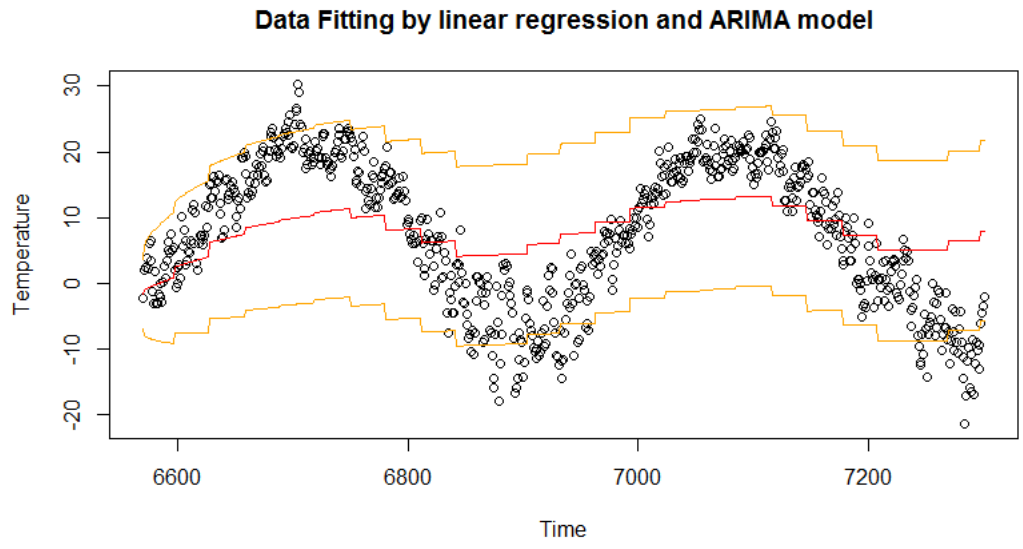


Figure 9

The forecast by the combination of linear regression and ARIMA model shows a smoother graph compared to the previous "linear regression only" model. It includes more data in its prediction interval. Hence it is used as the final model for predicting.

6 Statistical Conclusions

Models analysis

	AIC	PRESS	FIT TO DATA
regression model[reg5]	36459.2	16639.72	88.77%
ARIMA(2,2)[mod4]	33754.54	125400.8	48.08%
HOLT-WINTER [hw]	NA	23672.51	99.59%
regression with arima [reg.arma]	31639.66	55668.46	88.36%

Judging from the table, it is reasonable the Holt-Winter model as the final model since it has a relatively small PRESS value and the percentage of predicted data fitting the test data is relatively high. The Holt-Winter model should be use to predict the temperature in Toronto from March 5th, 2015 to March 5th, 2016, which is shown in Figure 10.

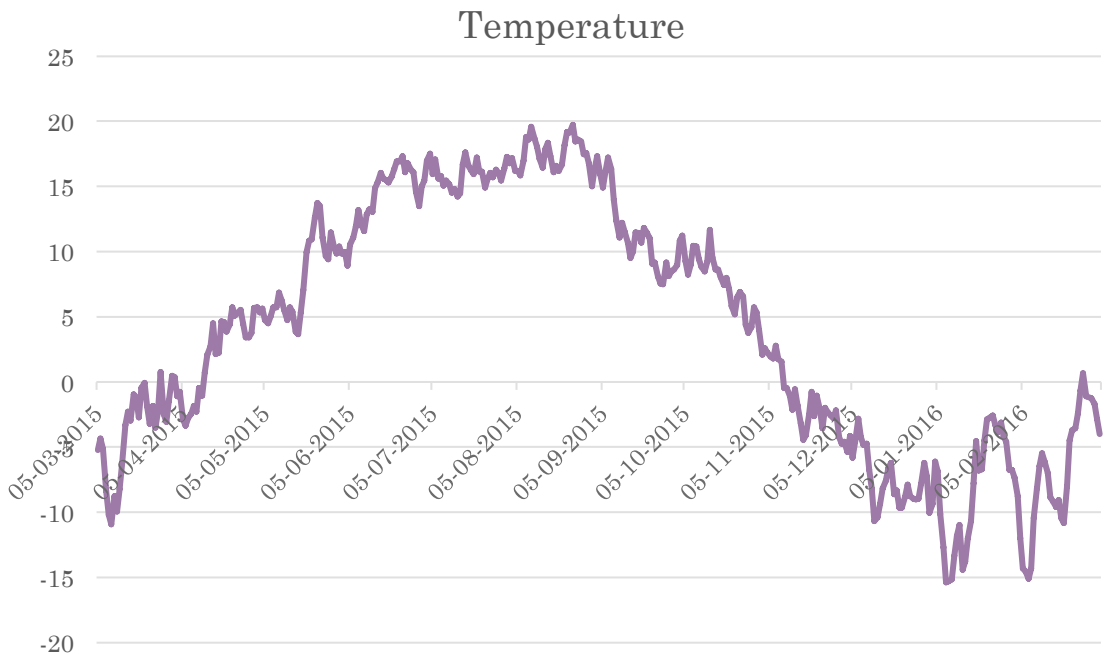


Figure 10

7 General Conclusions

A closer look should be taken to different aspects of our forecasting model concerning the prediction error. A visual analysis can be done by looking at the graph of how well the predicted data fit the tested data. Through analysis of properties of different temperature models, the best model to predict the temperature can be chosen. The trend of the predicted temperature is similar with the trend of the temperature in previous years. It was found that predicted data was slightly different from the real data judging from the temperature in March in 2015. An opinion is hold that only considering time variable is not enough for making a precise prediction. To improve the forecasting model, other natural factors are needed to be considered, such as latitude, altitude, cloud cover, distance from sea, ocean currents and so on. After all, a reasonable forecasting model is finalized by minimizing errors.

Appendix

Model Fitting Process

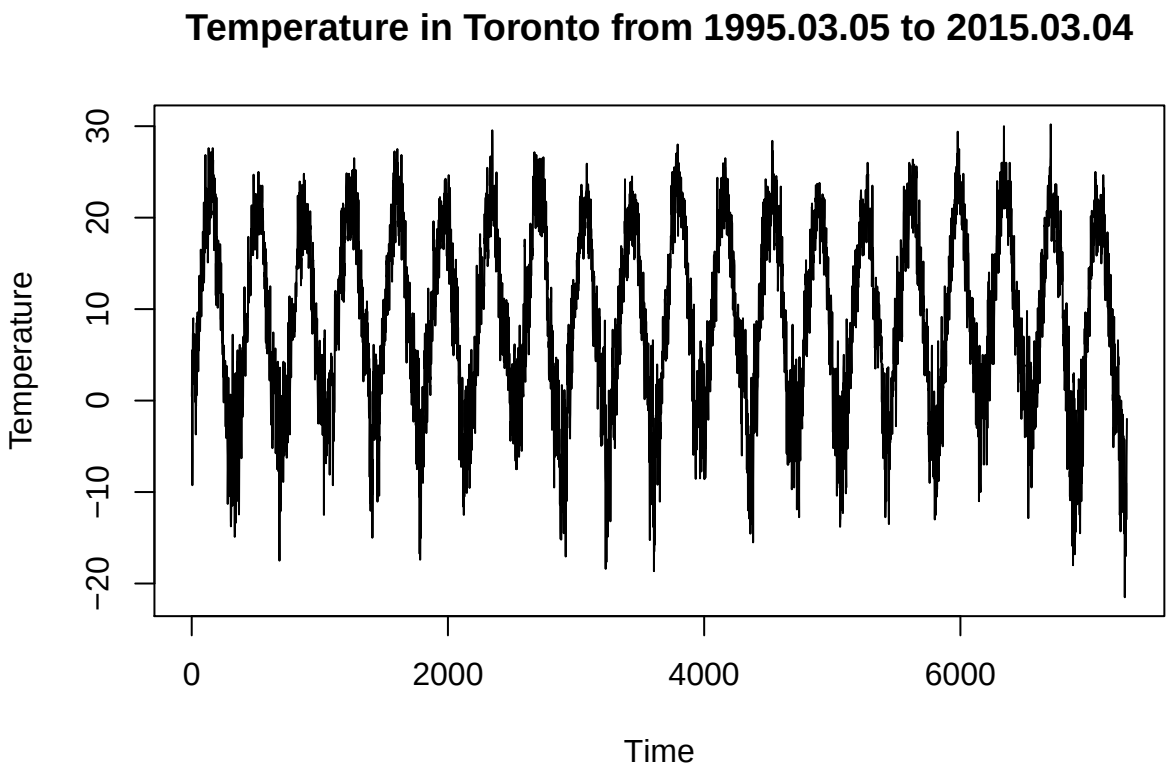
This part will lead the reader analyzing the data step by step.

1. General Analysis

```
setwd("~/Documents/UW/(4A) 2015 winter/STAT 443/group project")
library(gdata,quietly=T)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
##
## The following object is masked from 'package:stats':
##
##     nobs
##
## The following object is masked from 'package:utils':
##
##     object.size
```

```
trt<-read.xls("trt(f).xls")
ts.plot(trt$temp,ylab="Temperature",
        main="Temperature in Toronto from 1995.03.05 to 2015.03.04")
```



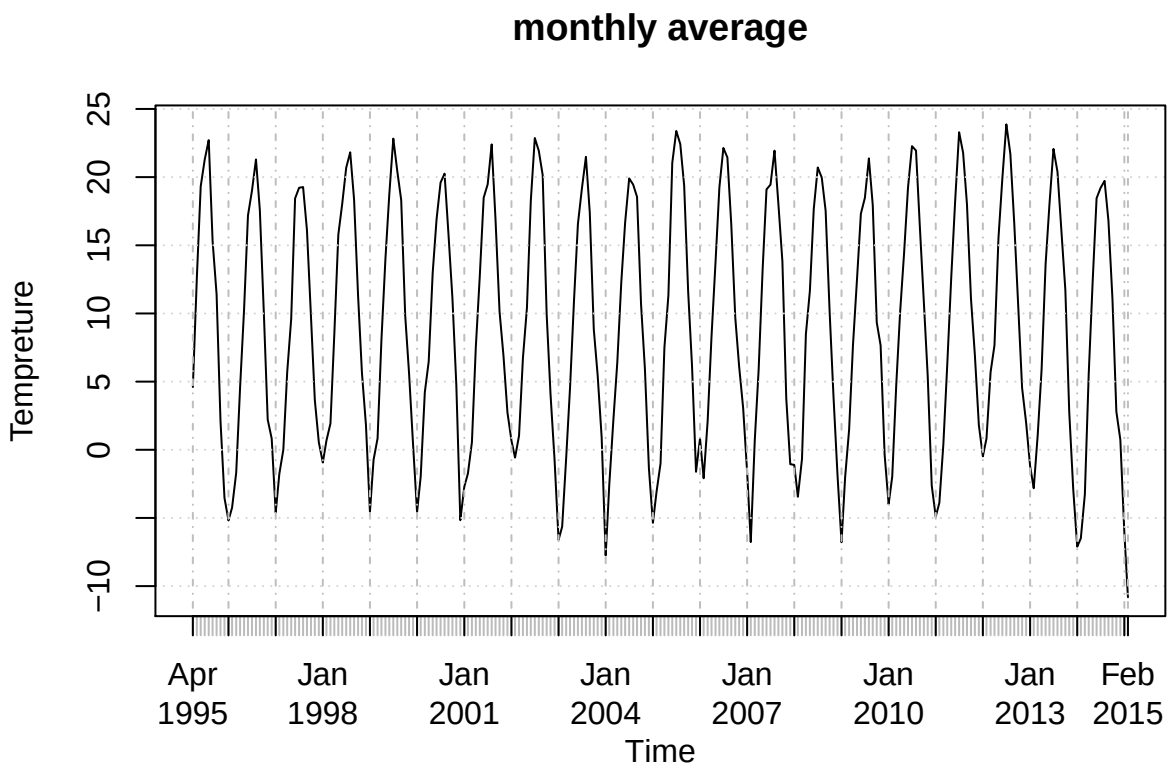
Observations: (1) no straight trend in this plot (more analysis on trends below) (2) a clear seasonality for 1 year

Next plot monthly average:

```
dat<-data.frame(date=as.Date(trt$date),temp=trt[2][,1]) # convert date factor to date format
library(xts,quietly=T)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

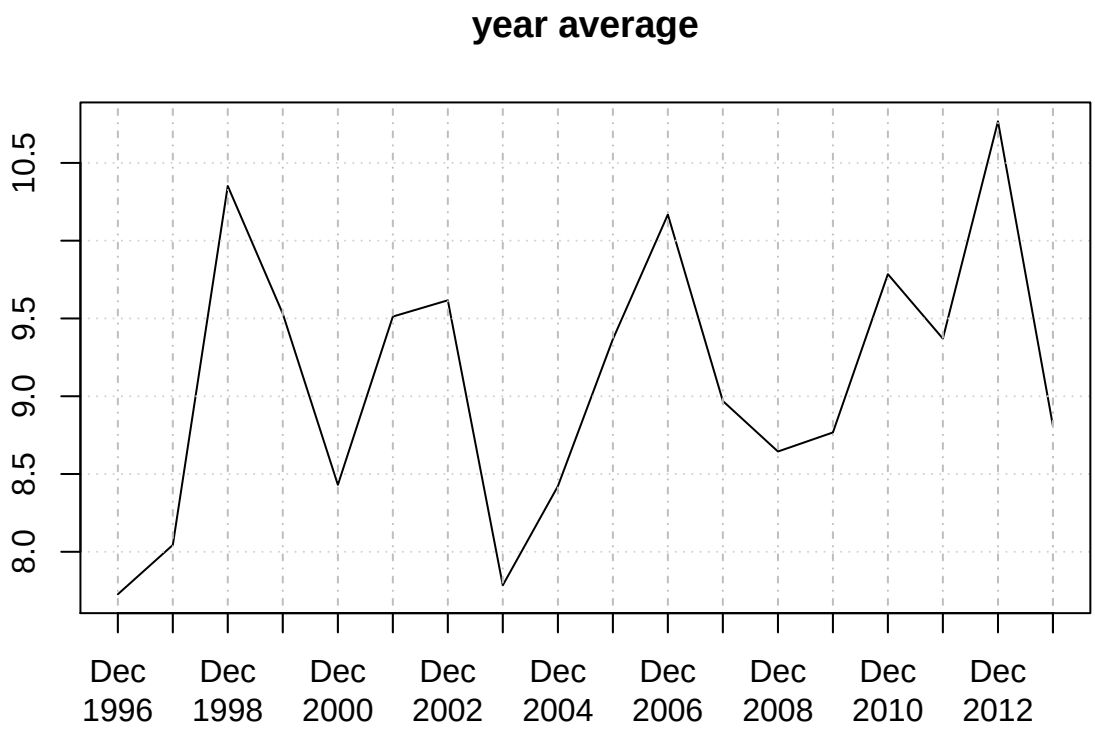
```
dat.xts<-xts(x=dat$temp,order.by=dat$date)
mavg<-apply.monthly(dat.xts,mean)
plot(mavg[2:(length(mavg)-1)],main="monthly average",xlab="Time",ylab="Tempreture") ## first and last m
```



Since the first and last months are not complete, they are removed in the graph. As above, a clear seasonality can be observed.

For yearly average

```
yavg<-apply.yearly(dat.xts,mean)
plot(yavg[2:19],main= "year average") ## same reason as above for remove
```



There is a slight increasing trend with large variance, which might scatter the linear trend in analysis.

Seperate data to train and test set.

```
rm(dat,dat.xts,mavg,yavg)
trt.train<-trt[2][,1][1:6570]
trt.test<-trt[2][,1][6571:7300]
```

Basic idea: (1) 10% (730) data for test, and 90% (6570) data for train (2) It is assumed to have 365 days in a year, which means the Feb. 29th data for 2012, 2008, 2004, 2000, 1996 are removed. (3) In year 2012, all temperature have been rounded to integer (from database). It would affect the model build too little to be statistically significance since the data set is huge. (4) Try linear model, time series model, and combined model to find a good fit.

2. linear model

The month indicator is the only variable considered here:

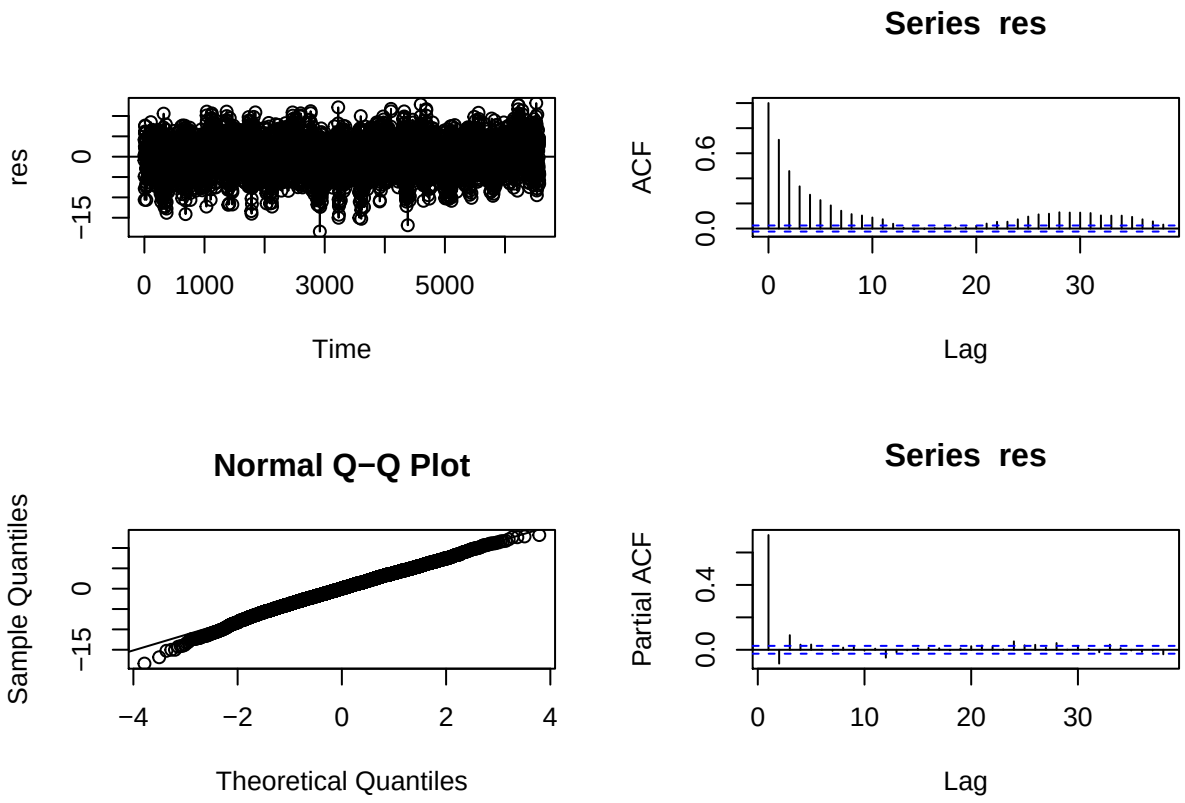
```
fa=NULL # create a factor for month indicator
jan=rep(1,31)
feb=rep(2,28)
mar=rep(3,31)
apr=rep(4,30)
may=rep(5,31)
jun=rep(6,30)
jul=rep(7,31)
```

```
aug=rep(8,31)
sep=rep(9,30)
oct=rep(10,31)
nov=rep(11,30)
dec=rep(12,31)
for (i in 1:21)
{
  fa=c(fa,jan,feb,mar,apr,may,jun,jul,aug,sep,oct,nov,dec)
}
fa = fa[64:7363]
mon=as.factor(fa)
mon.train<-mon[1:6570]
mon.test<-mon[6571:7300]
rm(mon,jan,feb,mar,apr,may,jun,jul,aug,sep,oct,nov,dec,i,fa)
reg1<-lm(trt.train~mon.train) # first regression - only about seasonality
summary(reg1)
```

```
##
## Call:
## lm(formula = trt.train ~ mon.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4033  -2.4652   0.0571   2.6857  13.1813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.3813     0.1647  -20.53  < 2e-16 ***
## mon.train2    0.9468     0.2391    3.96 7.59e-05 ***
## mon.train3    4.7346     0.2330   20.32  < 2e-16 ***
## mon.train4   10.3155     0.2349   43.92  < 2e-16 ***
## mon.train5   15.9876     0.2330   68.63  < 2e-16 ***
## mon.train6   21.7336     0.2349   92.53  < 2e-16 ***
## mon.train7   24.3619     0.2330  104.58  < 2e-16 ***
## mon.train8   24.6955     0.2330  106.01  < 2e-16 ***
## mon.train9   20.8776     0.2349   88.88  < 2e-16 ***
## mon.train10  13.9428     0.2330   59.85  < 2e-16 ***
## mon.train11   8.4305     0.2349   35.89  < 2e-16 ***
## mon.train12   3.1479     0.2330   13.51  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.891 on 6558 degrees of freedom
## Multiple R-squared:  0.8348, Adjusted R-squared:  0.8345
## F-statistic:  3013 on 11 and 6558 DF,  p-value: < 2.2e-16
```

```
resdiags <- function(res)
{
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
}
resdiags(reg1$residuals)
```



Comments: (1) The residual look random, and the Q-Q plot confirms that. (2) There are a few spikes from lag=1 to 10, which suggests correlation between data

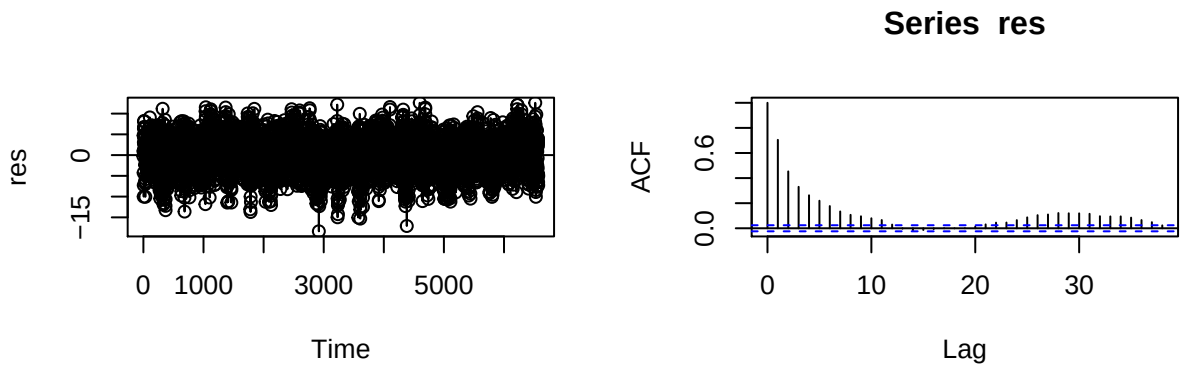
The year effect is included in the model:

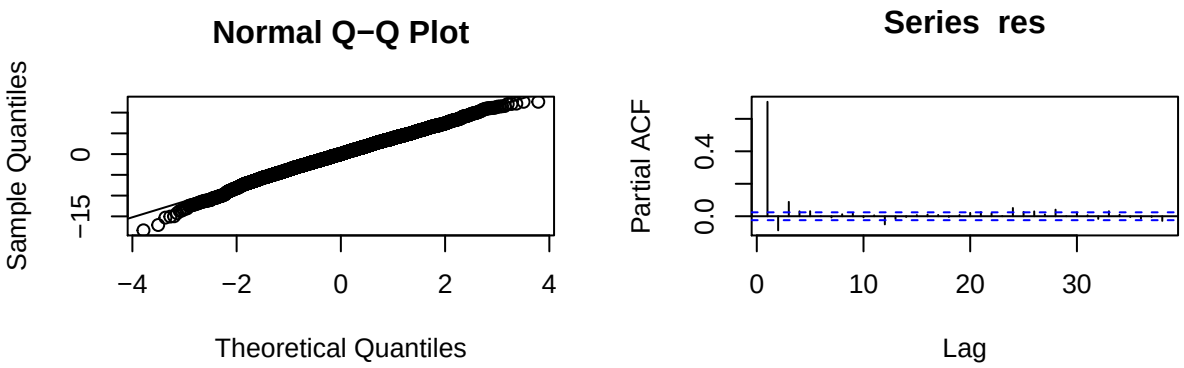
```
## construct a year set
yff<-function(){
  st<-rep(1995,302)
  for (i in 1:19){
    st<-c(st,rep((i+1995),365))
  }
  st<-c(st,rep(2015,63))
  st
  }
yf.train<-yff()[1:6570]
yf.test<-yff()[6571:7300]
reg2<-lm(trt.train~yf.train+mon.train)
summary(reg2)
```

```
##
```

```
## Call:
## lm(formula = trt.train ~ yf.train + mon.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3602  -2.4936   0.0333   2.6690  12.5988
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.407e+02  1.847e+01  -7.620 2.89e-14 ***
## yf.train     6.852e-02  9.213e-03   7.437 1.16e-13 ***
## mon.train2   9.468e-01  2.381e-01   3.976 7.09e-05 ***
## mon.train3   4.794e+00  2.321e-01  20.652  < 2e-16 ***
## mon.train4   1.038e+01  2.341e-01  44.356  < 2e-16 ***
## mon.train5   1.606e+01  2.322e-01  69.152  < 2e-16 ***
## mon.train6   2.180e+01  2.341e-01  93.128  < 2e-16 ***
## mon.train7   2.443e+01  2.322e-01 105.219  < 2e-16 ***
## mon.train8   2.476e+01  2.322e-01 106.656  < 2e-16 ***
## mon.train9   2.095e+01  2.341e-01  89.472  < 2e-16 ***
## mon.train10  1.401e+01  2.322e-01  60.346  < 2e-16 ***
## mon.train11  8.499e+00  2.341e-01  36.304  < 2e-16 ***
## mon.train12  3.216e+00  2.322e-01  13.853  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.875 on 6557 degrees of freedom
## Multiple R-squared:  0.8362, Adjusted R-squared:  0.8359
## F-statistic:  2789 on 12 and 6557 DF,  p-value: < 2.2e-16
```

```
resdiags(reg2$residuals)
```





The Q-Q plot, SACF and SPACF are not improved (same problem as above), but the SSE is decreased and the p-value for year is less than 0.05, so the year effect is contained in the model.

3. Other transformations

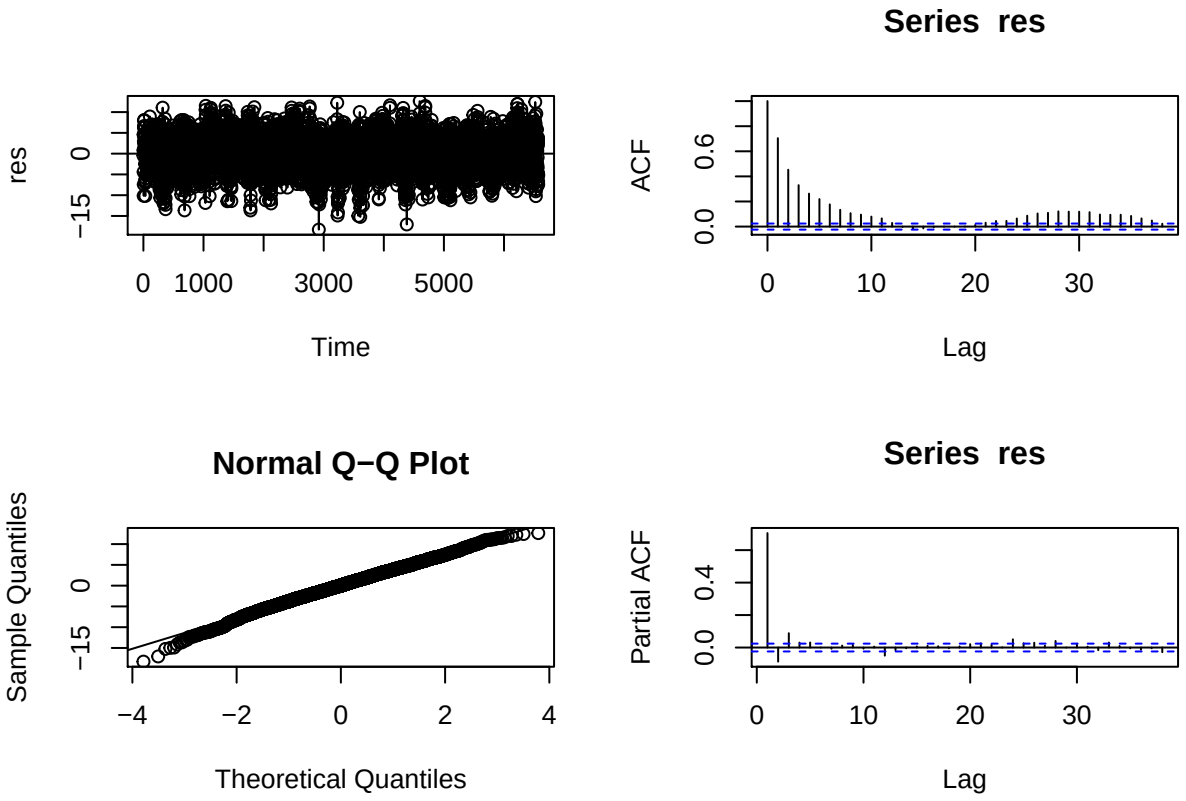
Consider the square root of year

```
ysr<-sqrt(yf.train)
reg3<-lm(trt.train~yf.train+ysr+mon.train)
summary(reg3)
```

```
##
## Call:
## lm(formula = trt.train ~ yf.train + ysr + mon.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.2831  -2.4901   0.0349   2.6614  12.6446
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46419.2955 31615.9197   1.468    0.142
## yf.train       23.3060    15.7791   1.477    0.140
## ysr         -2080.3254  1412.6148  -1.473    0.141
## mon.train2      0.9468     0.2381   3.976 7.08e-05 ***
```

```
## mon.train3      4.7959     0.2321  20.661  < 2e-16 ***
## mon.train4     10.3859     0.2341  44.367  < 2e-16 ***
## mon.train5     16.0580     0.2322  69.165  < 2e-16 ***
## mon.train6     21.8040     0.2341  93.143  < 2e-16 ***
## mon.train7     24.4323     0.2322 105.235  < 2e-16 ***
## mon.train8     24.7659     0.2322 106.672  < 2e-16 ***
## mon.train9     20.9480     0.2341  89.486  < 2e-16 ***
## mon.train10    14.0132     0.2322  60.358  < 2e-16 ***
## mon.train11     8.5009     0.2341  36.314  < 2e-16 ***
## mon.train12     3.2183     0.2322  13.862  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.875 on 6556 degrees of freedom
## Multiple R-squared:  0.8362, Adjusted R-squared:  0.8359
## F-statistic:  2575 on 13 and 6556 DF,  p-value: < 2.2e-16
```

```
resdiags(reg3$residuals)
```



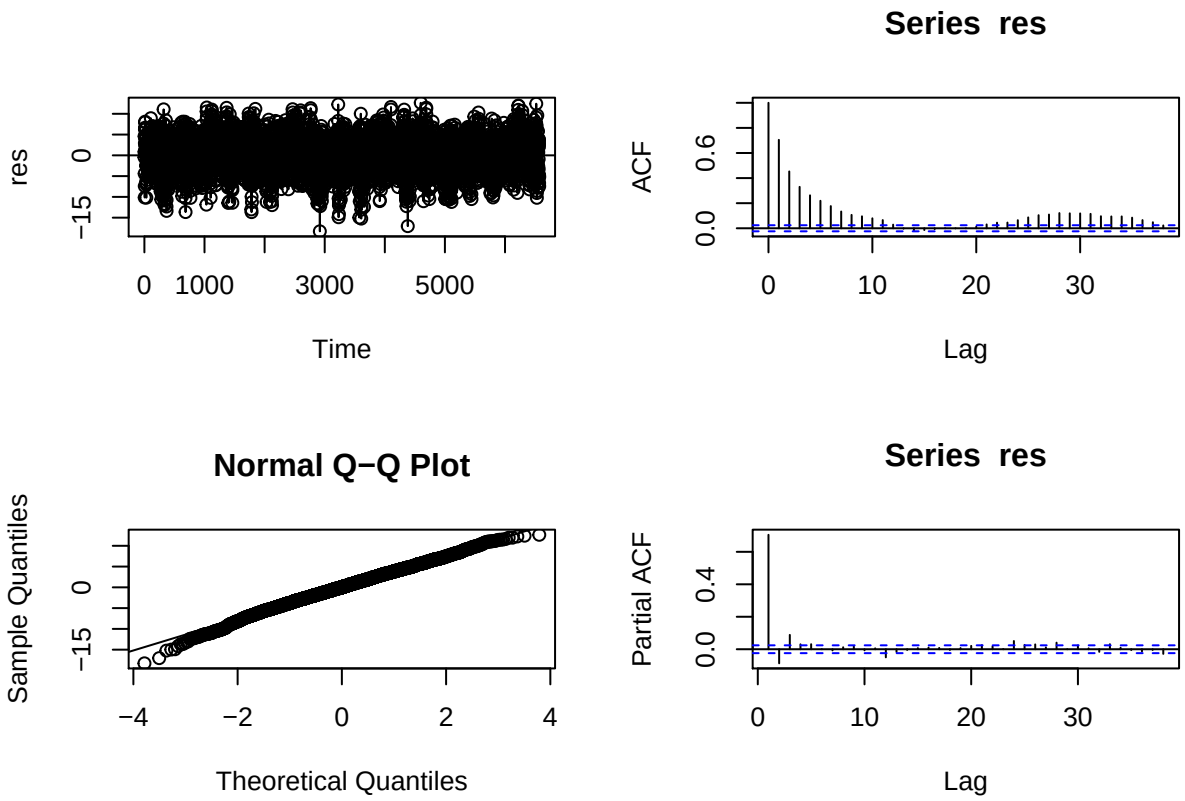
Comment: square root is not useful, which can be determined from the statistics summary. The p-value of intercept, year, and square root of year are more than 0.05, so the square root of year is removed.

Try log of year:

```
yrl<-log(yf.train)
reg4<-lm(trt.train~yf.train+yrl+mon.train)
summary(reg4)
```

```
##
## Call:
## lm(formula = trt.train ~ yf.train + yrl + mon.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.2832  -2.4901   0.0349   2.6614  12.6444
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.533e+05  1.044e+05   1.469    0.142
## yf.train     1.167e+01  7.889e+00   1.479    0.139
## yrl         -2.324e+04  1.581e+04  -1.470    0.142
## mon.train2   9.468e-01  2.381e-01   3.976 7.08e-05 ***
## mon.train3   4.796e+00  2.321e-01  20.661  < 2e-16 ***
## mon.train4   1.039e+01  2.341e-01  44.367  < 2e-16 ***
## mon.train5   1.606e+01  2.322e-01  69.165  < 2e-16 ***
## mon.train6   2.180e+01  2.341e-01  93.143  < 2e-16 ***
## mon.train7   2.443e+01  2.322e-01 105.235  < 2e-16 ***
## mon.train8   2.477e+01  2.322e-01 106.672  < 2e-16 ***
## mon.train9   2.095e+01  2.341e-01  89.486  < 2e-16 ***
## mon.train10  1.401e+01  2.322e-01  60.358  < 2e-16 ***
## mon.train11  8.501e+00  2.341e-01  36.314  < 2e-16 ***
## mon.train12  3.218e+00  2.322e-01  13.862  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.875 on 6556 degrees of freedom
## Multiple R-squared:  0.8362, Adjusted R-squared:  0.8359
## F-statistic:  2575 on 13 and 6556 DF,  p-value: < 2.2e-16
```

```
resdiags(reg4$residuals)
```



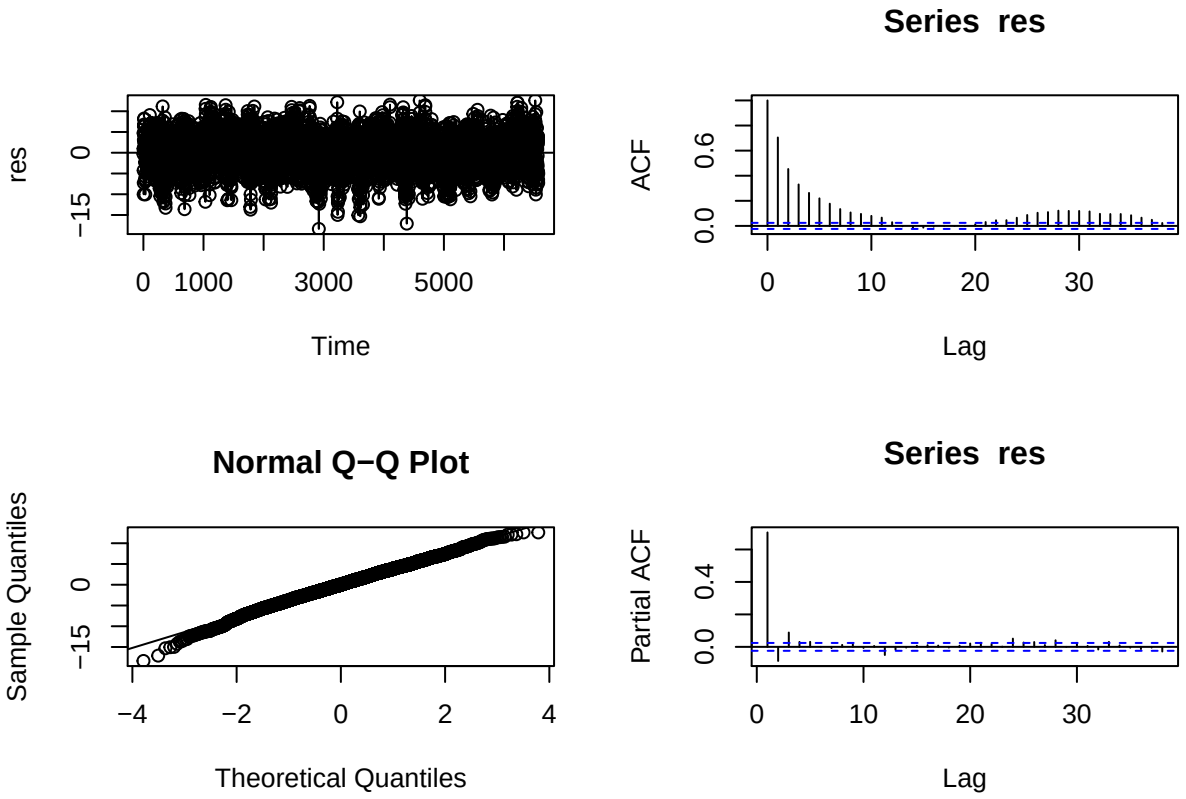
Comment: log of year is not useful, same reason as square root discuss.

```
t<-time(trt$date)
t.train<-t[1:6570]
t.test<-t[6571:7300]
reg5<-lm(trt.train~t.train+mon.train)
summary(reg5)
```

```
##
## Call:
## lm(formula = trt.train ~ t.train + mon.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3577  -2.4934   0.0337   2.6694  12.6000
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.024e+00  1.854e-01 -21.705  < 2e-16 ***
## t.train      1.878e-04  2.524e-05   7.441 1.13e-13 ***
## mon.train2   9.413e-01  2.381e-01   3.953 7.81e-05 ***
## mon.train3   4.783e+00  2.321e-01  20.609  < 2e-16 ***
## mon.train4   1.037e+01  2.340e-01  44.299  < 2e-16 ***
## mon.train5   1.603e+01  2.321e-01  69.086  < 2e-16 ***
## mon.train6   2.177e+01  2.340e-01  93.055  < 2e-16 ***
## mon.train7   2.440e+01  2.320e-01 105.135  < 2e-16 ***
```

```
## mon.train8   2.472e+01  2.320e-01 106.555  < 2e-16 ***
## mon.train9   2.090e+01  2.339e-01  89.339  < 2e-16 ***
## mon.train10  1.396e+01  2.320e-01  60.169  < 2e-16 ***
## mon.train11  8.442e+00  2.339e-01  36.088  < 2e-16 ***
## mon.train12  3.154e+00  2.320e-01  13.593  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.875 on 6557 degrees of freedom
## Multiple R-squared:  0.8362, Adjusted R-squared:  0.8359
## F-statistic:  2789 on 12 and 6557 DF,  p-value: < 2.2e-16
```

```
resdiags(reg5$residuals)
```




```
library(lawstat)
```

```
## Loading required package: mvtnorm
## Loading required package: VGAM
## Loading required package: stats4
## Loading required package: splines
## Loading required package: Kendall
## Loading required package: Hmisc
## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
```

```
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'Hmisc'
##
## The following object is masked from 'package:gdata':
##
##     combine
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
runs.test(reg5$residuals)
```

```
##
##  Runs Test - Two sided
##
## data:  reg5$residuals
## Standardized Runs Statistic = -41.5302, p-value < 2.2e-16
```

```
library(randtests)
```

```
##
## Attaching package: 'randtests'
##
## The following object is masked from 'package:lawstat':
##
##     runs.test
```

```
difference.sign.test(reg5$residuals)
```

```
##
##  Difference Sign Test
##
## data:  reg5$residuals
## statistic = 1.8162, n = 6570, p-value = 0.06934
## alternative hypothesis: nonrandomness
```

```
turning.point.test(reg5$residuals)
```

```
##
##  Turning Point Test
##
## data:  reg5$residuals
## statistic = -29.1668, n = 6570, p-value < 2.2e-16
## alternative hypothesis: non randomness
```

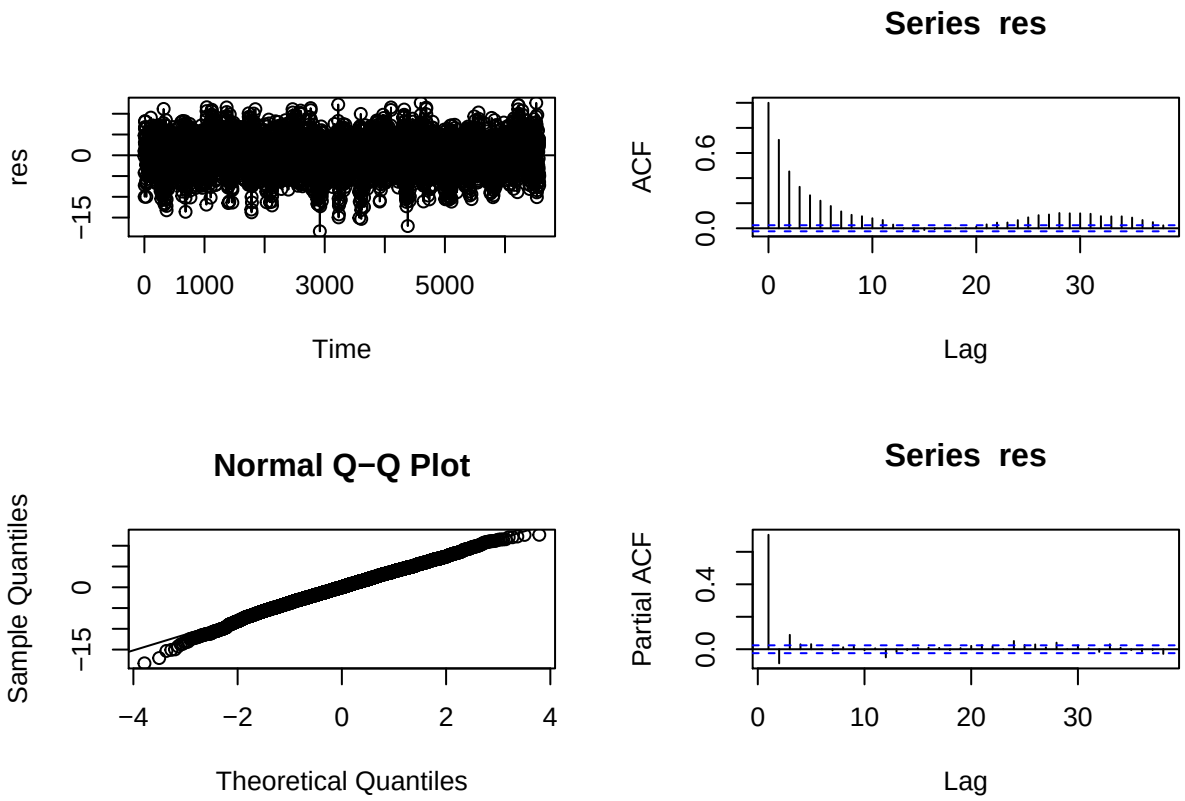
Comments: (1) Since the length of data set is too large, shapiro test cannot be performed here. (2) Other tests show a good result.

If the day effect is considered:

```
days<-function(){
  a=c(1:31)
  b=c(1:30)
  c=c(1:28)
  d=c(a,c,a,b,a,b,a,a,b,a,b,a)
  for (i in 1:21){
     d=c(d,d)
  }
d[64:7363]
}
days.train=days()[1:6570]
days.test=days()[6571:7300]
reg6<-lm(trt.train~days.train+mon.train+yf.train)
summary(reg6)
```

```
##
## Call:
## lm(formula = trt.train ~ days.train + mon.train + yf.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3033  -2.4831   0.0303   2.6688  12.6335
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.409e+02  1.847e+01  -7.626 2.76e-14 ***
## days.train   4.378e-03  5.442e-03   0.805    0.421
## mon.train2   9.534e-01  2.383e-01   4.001 6.38e-05 ***
## mon.train3   4.794e+00  2.321e-01  20.652  < 2e-16 ***
## mon.train4   1.039e+01  2.341e-01  44.361  < 2e-16 ***
## mon.train5   1.606e+01  2.322e-01  69.151  < 2e-16 ***
## mon.train6   2.180e+01  2.341e-01  93.129  < 2e-16 ***
## mon.train7   2.443e+01  2.322e-01 105.217  < 2e-16 ***
## mon.train8   2.476e+01  2.322e-01 106.654  < 2e-16 ***
## mon.train9   2.095e+01  2.341e-01  89.473  < 2e-16 ***
## mon.train10  1.401e+01  2.322e-01  60.344  < 2e-16 ***
## mon.train11  8.501e+00  2.341e-01  36.310  < 2e-16 ***
## mon.train12  3.216e+00  2.322e-01  13.853  < 2e-16 ***
## yf.train     6.855e-02  9.214e-03   7.440 1.14e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.875 on 6556 degrees of freedom
## Multiple R-squared:  0.8362, Adjusted R-squared:  0.8359
## F-statistic:  2575 on 13 and 6556 DF,  p-value: < 2.2e-16
```

```
resdiags(reg6$residuals)
```



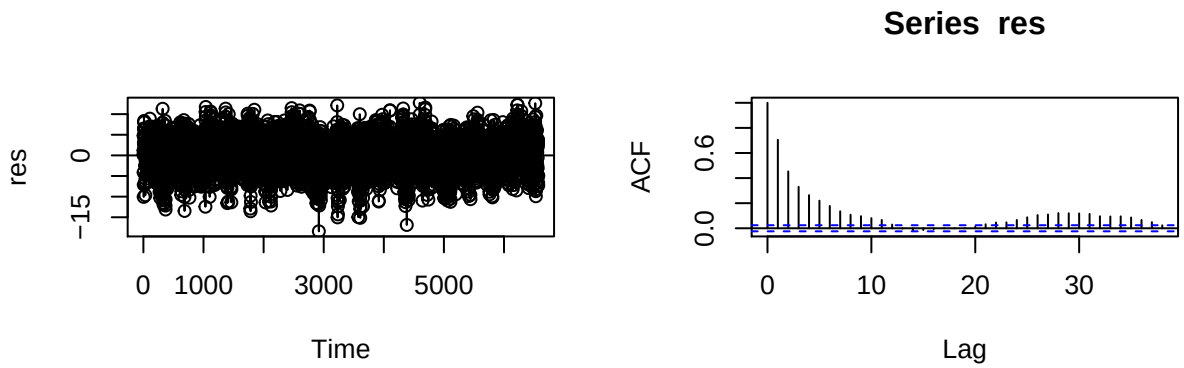
It is clear that the day effect is very small. Convert day to factor:

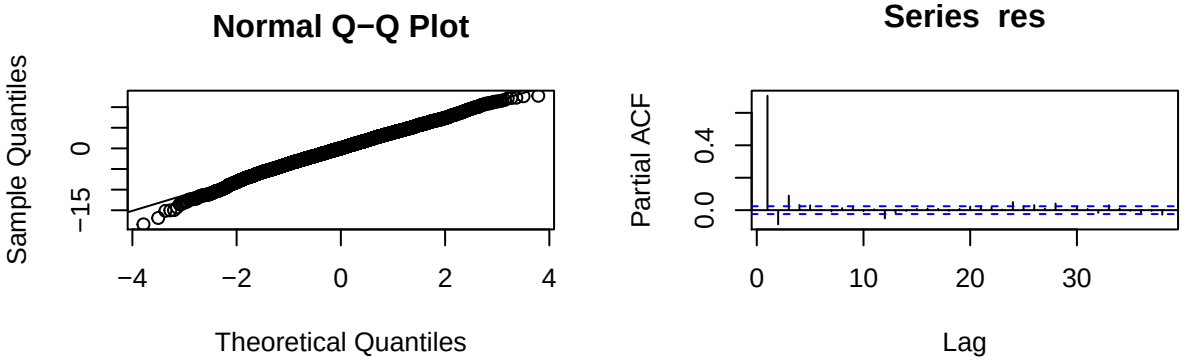
```
dayf<-as.factor(days.train)
reg7<-reg6<-lm(trt.train~dayf+mon.train+yf.train)
summary(reg7)
```

```
##
## Call:
## lm(formula = trt.train ~ dayf + mon.train + yf.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4059  -2.4749   0.0247   2.6634  12.7471
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.402e+02  1.849e+01  -7.582 3.88e-14 ***
## dayf2       -3.035e-01  3.733e-01  -0.813   0.4162
## dayf3       -3.813e-01  3.733e-01  -1.021   0.3071
## dayf4       -4.946e-01  3.733e-01  -1.325   0.1852
## dayf5       -8.420e-01  3.733e-01  -2.256   0.0241 *
## dayf6       -6.451e-01  3.733e-01  -1.728   0.0840 .
## dayf7       -4.857e-01  3.733e-01  -1.301   0.1933
## dayf8       -5.753e-01  3.733e-01  -1.541   0.1233
## dayf9       -3.780e-01  3.733e-01  -1.013   0.3112
## dayf10      -5.690e-01  3.733e-01  -1.524   0.1275
```

```
## dayf11      -5.810e-01  3.733e-01  -1.556   0.1197
## dayf12      -4.443e-01  3.733e-01  -1.190   0.2340
## dayf13      -4.691e-01  3.733e-01  -1.257   0.2089
## dayf14      -1.924e-01  3.733e-01  -0.515   0.6063
## dayf15      -3.559e-01  3.733e-01  -0.953   0.3405
## dayf16      -5.497e-01  3.733e-01  -1.473   0.1409
## dayf17      -5.919e-01  3.733e-01  -1.586   0.1129
## dayf18      -5.663e-01  3.733e-01  -1.517   0.1293
## dayf19      -2.729e-01  3.733e-01  -0.731   0.4648
## dayf20      -5.020e-01  3.733e-01  -1.345   0.1787
## dayf21      -5.877e-01  3.733e-01  -1.574   0.1154
## dayf22      -6.404e-01  3.733e-01  -1.716   0.0863 .
## dayf23      -5.680e-01  3.733e-01  -1.522   0.1281
## dayf24      -6.016e-01  3.733e-01  -1.612   0.1071
## dayf25      -3.510e-01  3.733e-01  -0.940   0.3472
## dayf26      -2.931e-01  3.733e-01  -0.785   0.4324
## dayf27      -4.395e-01  3.733e-01  -1.177   0.2391
## dayf28      -2.917e-01  3.733e-01  -0.781   0.4346
## dayf29      -1.633e-01  3.820e-01  -0.428   0.6690
## dayf30      -2.981e-01  3.820e-01  -0.780   0.4351
## dayf31       1.988e-01  4.368e-01   0.455   0.6491
## mon.train2   9.832e-01  2.390e-01   4.113 3.95e-05 ***
## mon.train3   4.794e+00  2.324e-01  20.630  < 2e-16 ***
## mon.train4   1.040e+01  2.346e-01  44.345  < 2e-16 ***
## mon.train5   1.606e+01  2.324e-01  69.078  < 2e-16 ***
## mon.train6   2.182e+01  2.346e-01  93.009  < 2e-16 ***
## mon.train7   2.443e+01  2.324e-01 105.107  < 2e-16 ***
## mon.train8   2.476e+01  2.324e-01 106.542  < 2e-16 ***
## mon.train9   2.097e+01  2.346e-01  89.360  < 2e-16 ***
## mon.train10  1.401e+01  2.324e-01  60.281  < 2e-16 ***
## mon.train11  8.520e+00  2.346e-01  36.311  < 2e-16 ***
## mon.train12  3.216e+00  2.324e-01  13.838  < 2e-16 ***
## yf.train     6.847e-02  9.223e-03   7.423 1.29e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.879 on 6527 degrees of freedom
## Multiple R-squared:  0.8366, Adjusted R-squared:  0.8355
## F-statistic: 795.6 on 42 and 6527 DF,  p-value: < 2.2e-16
```

```
resdiags(reg7$residuals)
```





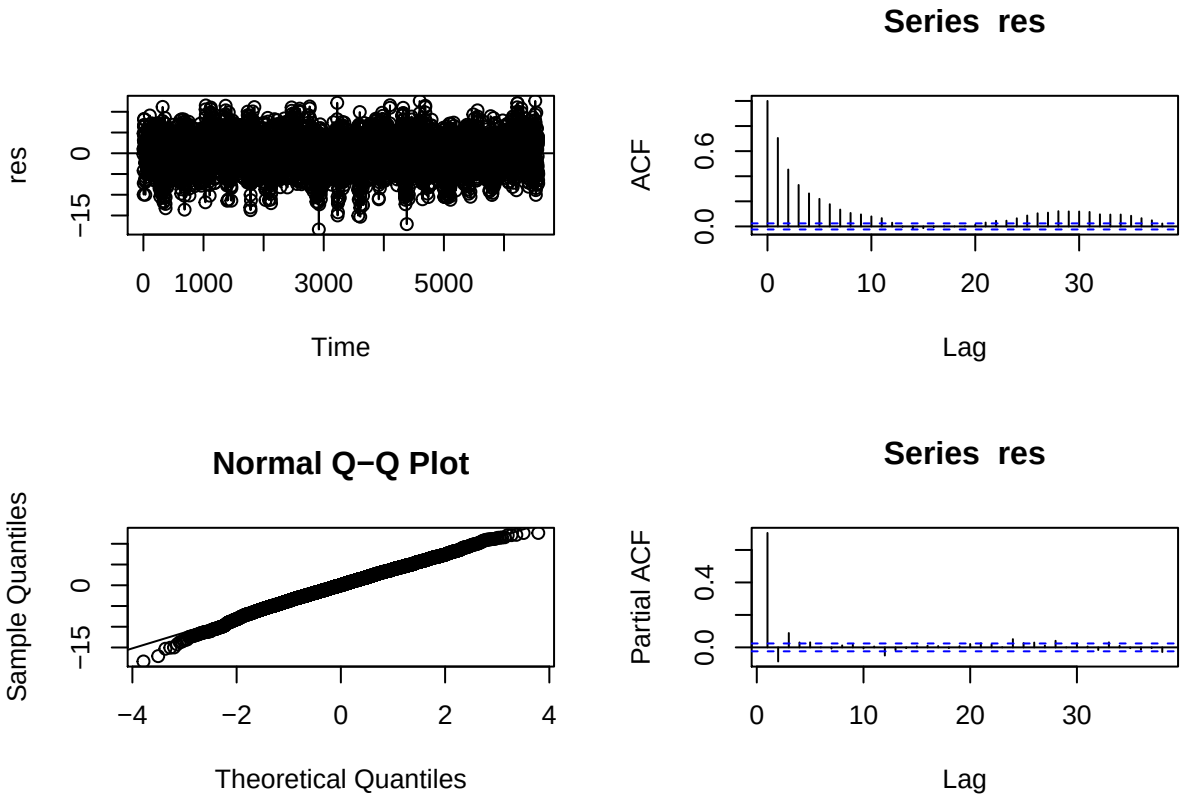
The model is still not improved, so the day variable is removed. Consider the square root of temperature:

```
t2<-sqrt(t.train)
reg8<-lm(trt.train~t.train+t2+mon.train)
summary(reg8)
```

```
##
## Call:
## lm(formula = trt.train ~ t.train + t2 + mon.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3676  -2.4927   0.0336   2.6680  12.6126
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.0781445  0.3308755 -12.325  < 2e-16 ***
## t.train      0.0001633  0.0001263   1.293    0.196
## t2           0.0024909  0.0125430   0.199    0.843
## mon.train2   0.9412200  0.2381534   3.952 7.83e-05 ***
## mon.train3   4.7846525  0.2322257  20.603  < 2e-16 ***
## mon.train4  10.3685295  0.2341344  44.285  < 2e-16 ***
## mon.train5  16.0346245  0.2321543  69.069  < 2e-16 ***
## mon.train6  21.7747082  0.2340391  93.039  < 2e-16 ***
## mon.train7  24.3970254  0.2320846 105.121  < 2e-16 ***
## mon.train8  24.7247106  0.2320617 106.544  < 2e-16 ***
```

```
## mon.train9  20.9009640  0.2339702  89.332  < 2e-16 ***
## mon.train10 13.9603196  0.2320328  60.165  < 2e-16 ***
## mon.train11  8.4421655  0.2339500  36.085  < 2e-16 ***
## mon.train12  3.1537709  0.2320199  13.593  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.875 on 6556 degrees of freedom
## Multiple R-squared:  0.8362, Adjusted R-squared:  0.8359
## F-statistic:  2574 on 13 and 6556 DF,  p-value: < 2.2e-16
```

```
resdiags(reg8$residuals)
```



The quare root of t is not good.

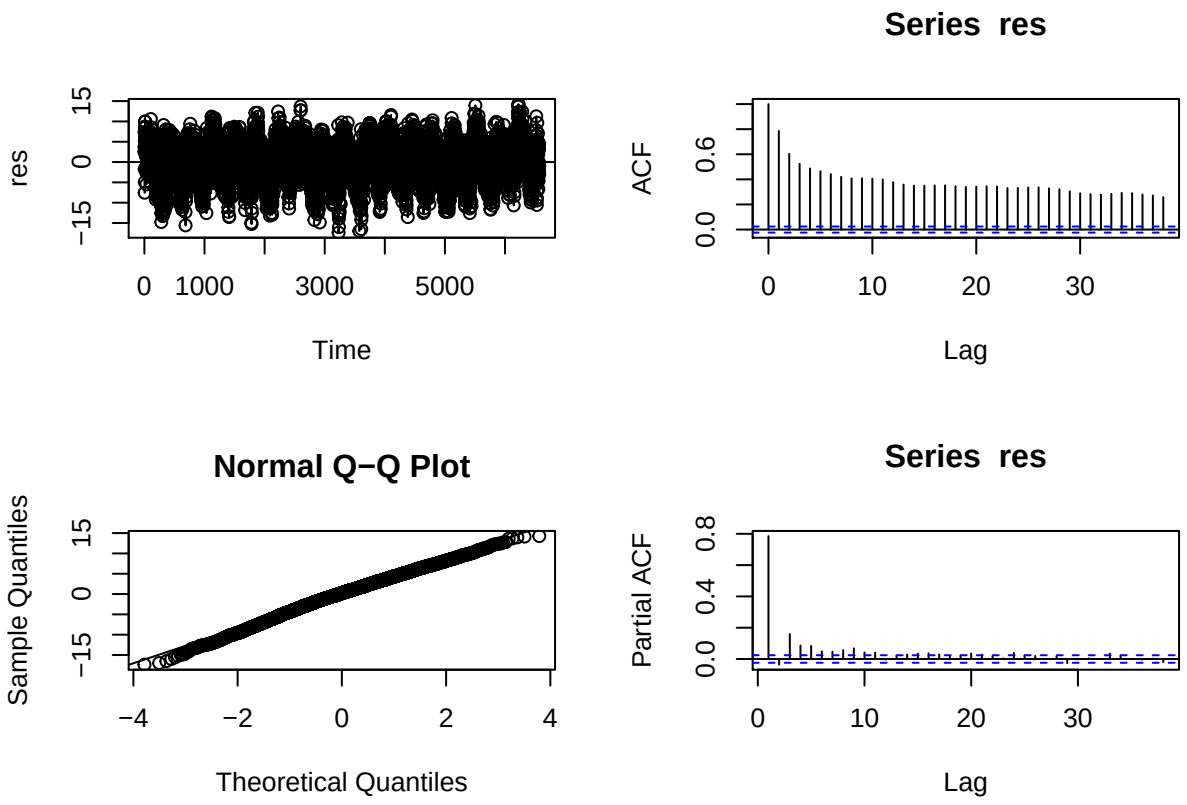
4. Sine regression model

```
a<-sin(2*pi/365*(t.train-70))
sinm<-lm(trt.train~a)
summary(sinm)
```

```
##
## Call:
```

```
## lm(formula = trt.train ~ a)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.3560  -2.7574   0.2407   3.0345  14.2635
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.11246    0.05421   168.1   <2e-16 ***
## a           12.01593    0.07667   156.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.394 on 6568 degrees of freedom
## Multiple R-squared:  0.789,  Adjusted R-squared:  0.789
## F-statistic: 2.456e+04 on 1 and 6568 DF,  p-value: < 2.2e-16
```

```
resdiags(sinm$residuals)
```

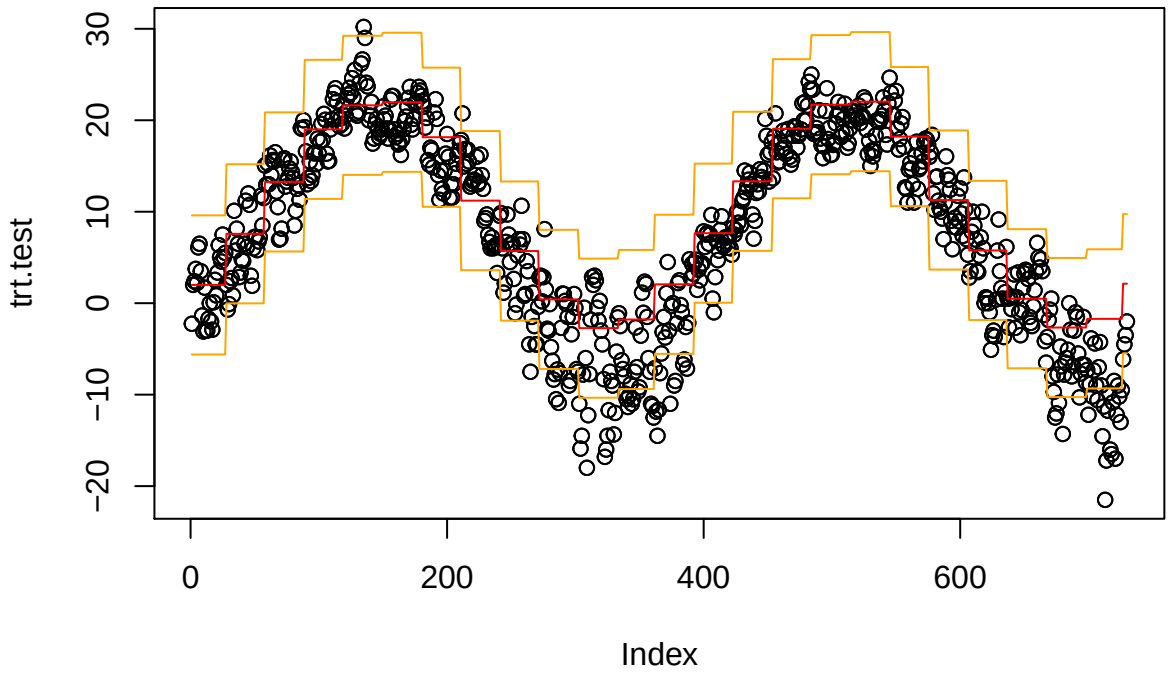


Therefore the model reg5 is chosen as the linear regression model, and it is used to fit the testing set in the next part.

```
reg5.pre<-predict(reg5,newdata=list(t.train=t.test,mon.train=mon.test),interval="prediction")
pre.result<-(trt.test>reg5.pre[,"lwr"] & trt.test<reg5.pre[,"upr"])
table(pre.result)
```

```
## pre.result
## FALSE  TRUE
##    82   648
```

```
par(mfcol=c(1,1))
plot(trt.test)
points(trt.test)
points(reg5.pre[,"fit"], type="l", col="red")
points(reg5.pre[,"lwr"], type="l", col="orange")
points(reg5.pre[,"upr"], type="l", col="orange")
```



88.77% of test data is in 95% predicted interval

```
PRESS.reg5<-sum((trt.test-reg5.pre[,"fit"])^2)
PRESS.reg5
```

```
## [1] 16639.72
```

```
AIC(reg5)
```

```
## [1] 36459.2
```

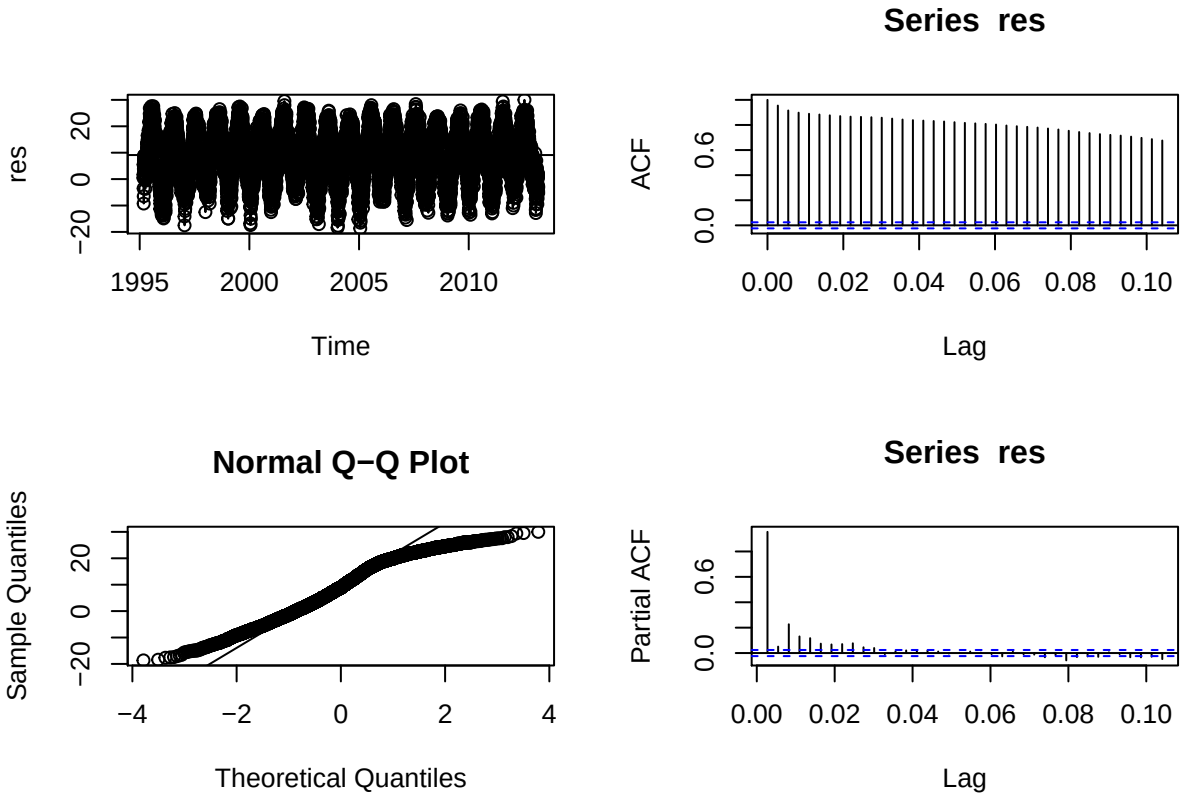
5. Time series

SARIMA

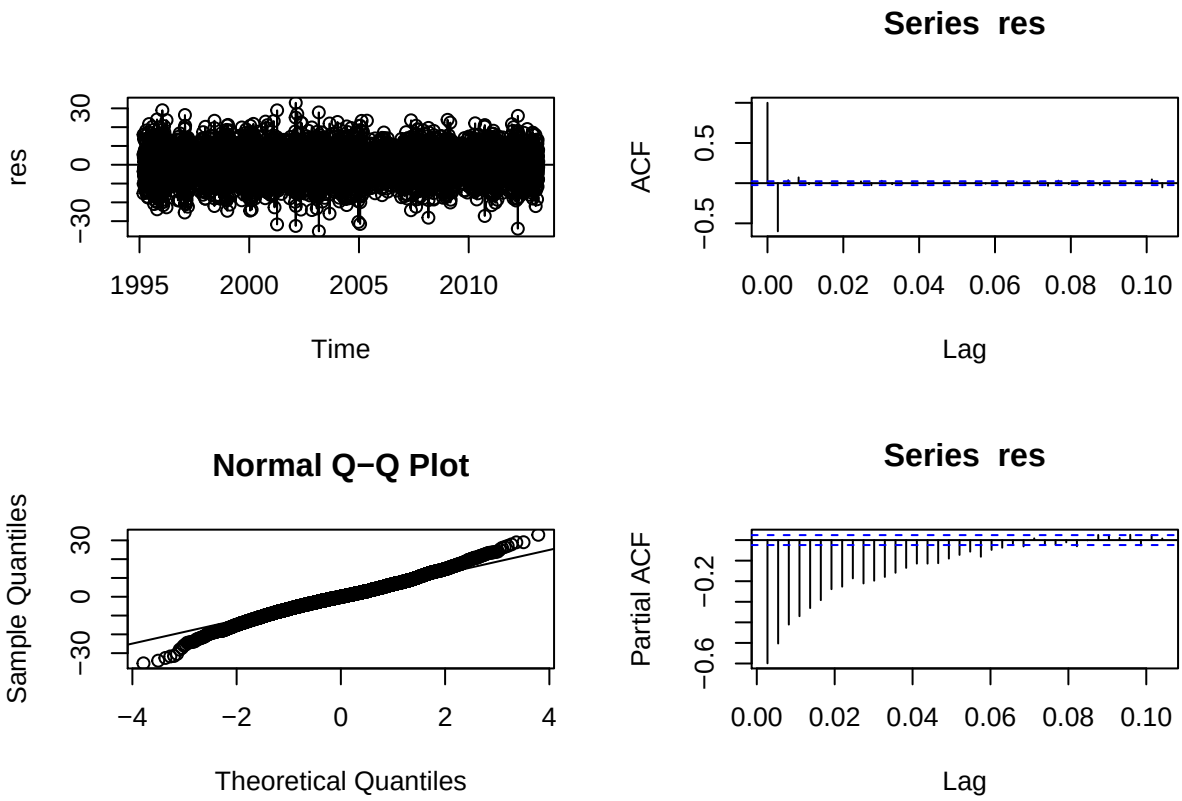
```
trt.ts<-ts(trt.train,start=c(1995,64),frequency=365)
trt.testts<-ts(trt.test,start=c(2013,64),frequency=365)
library(forecast)
```

```
## Loading required package: timeDate
## This is forecast 5.9
```

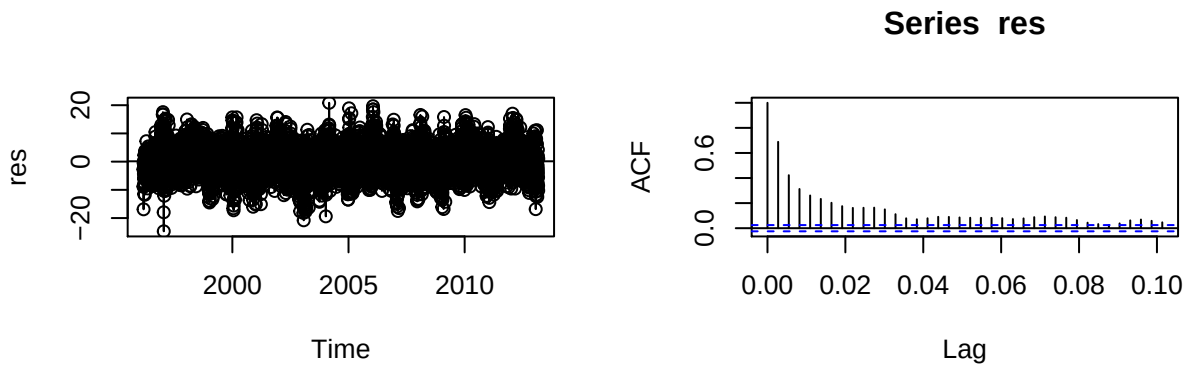
```
par(mfcol=c(1,1))
## we can see there is a yearly seasonality in this series.
resdiags(trt.ts)
```

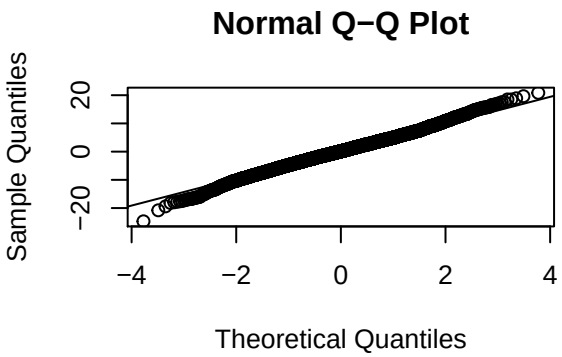


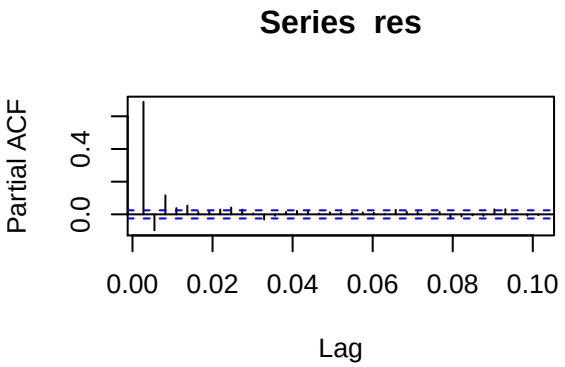

```
## too bad, try diff 1
resdiags(diff(trt.ts,diff=3))
```



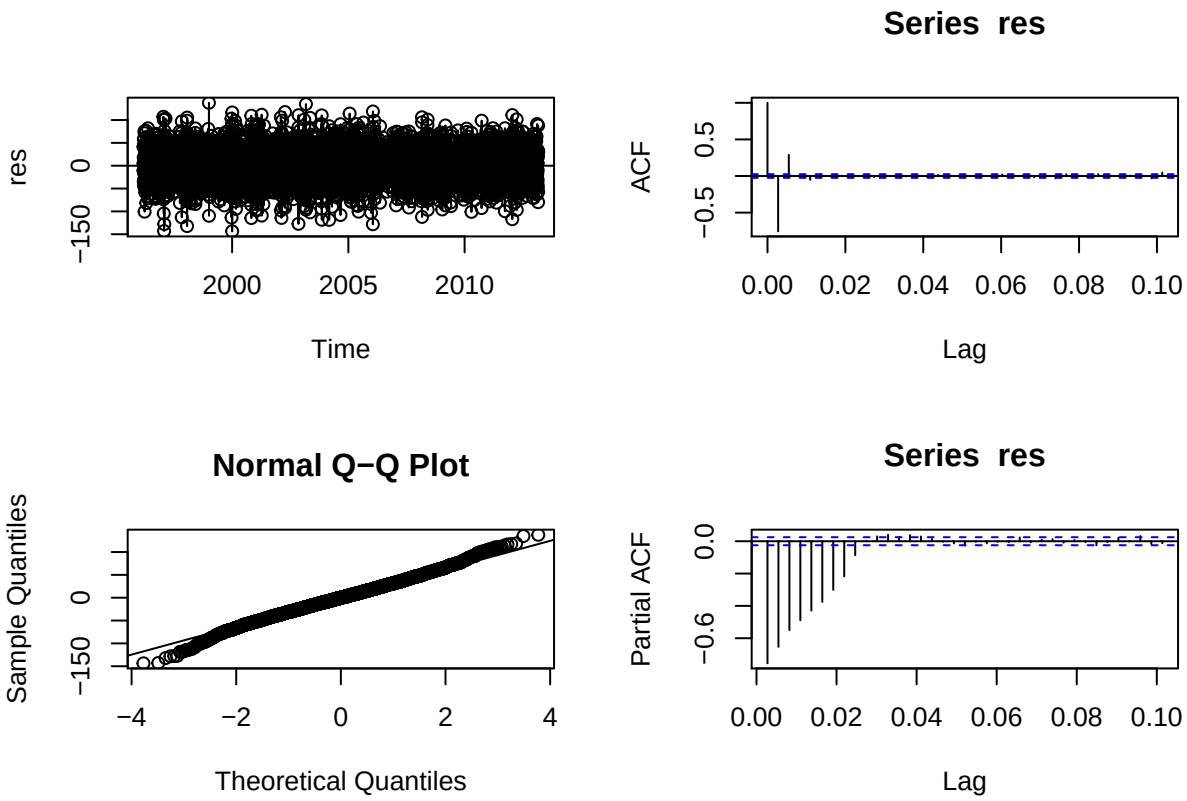

```
## qq plot and acf look better.
resdiags(diff(trt.ts,lag=365))
```








```
## pacf is better, but acf is worse, try more difference
resdiags(diff(diff(trt.ts,lag=365),diff=5))
```

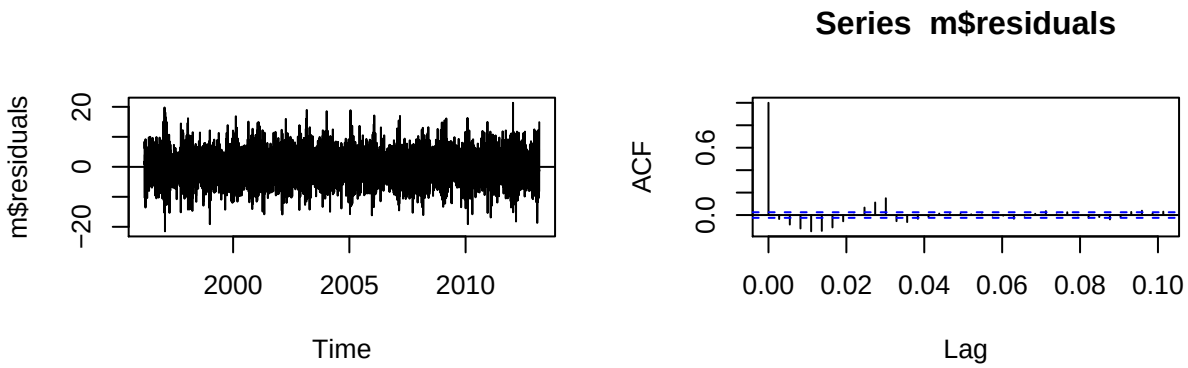


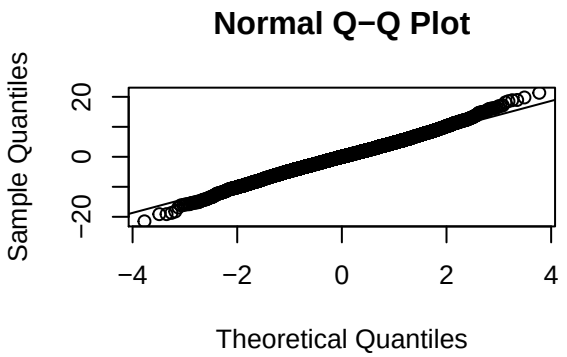

```
## looks good. try arima(9,0,3)
trt.try<-diff(diff(trt.ts,lag=365),diff=5)
mod1<-arima(trt.try,order=c(9,0,3),method="ML")
```

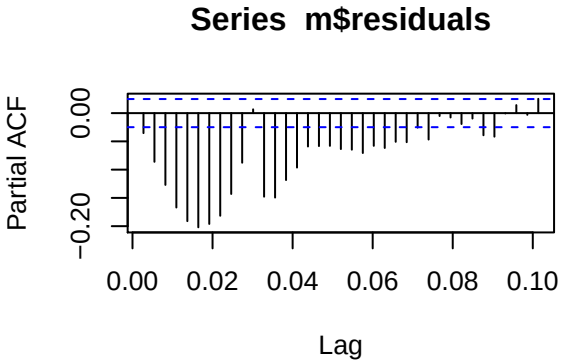
```
## Warning in log(s2): NaNs produced
```

```
## Warning in log(s2): NaNs produced
```

```
resd<-function(m){
  par(mfcol=c(2,2))
  plot(m$residuals)
  abline(h=mean(m$residuals))
  qqnorm(m$residuals)
  qqline(m$residuals)
  acf(m$residuals)
  acf(m$residuals,type="partial")
}

resd(mod1)
```



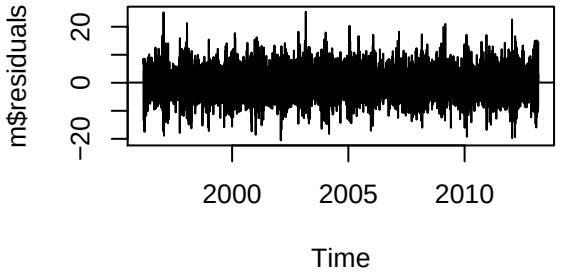


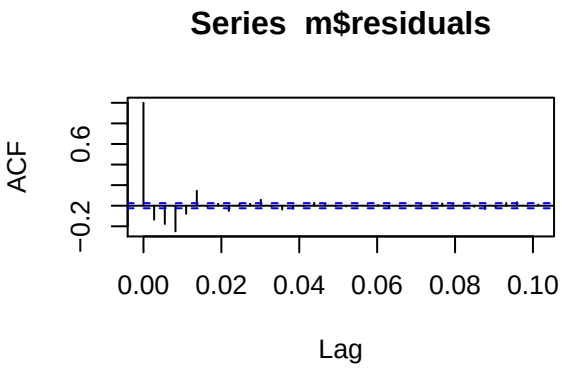


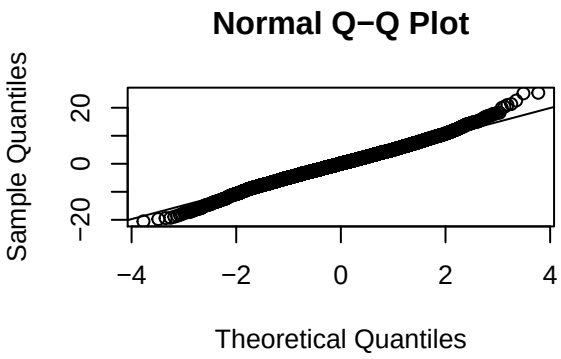

```
## too bad, try auto.arima
auto.arima(trt.try)
```

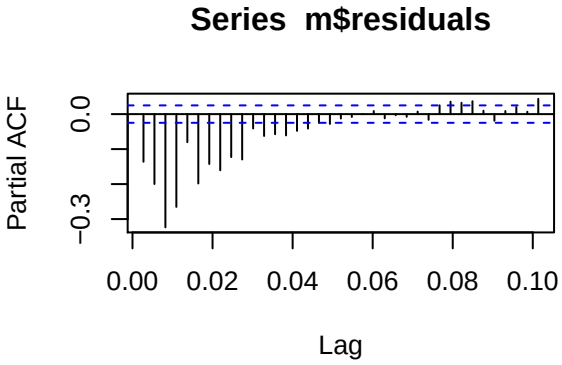
```
## Series: trt.try
## ARIMA(3,0,4) with non-zero mean
##
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

##           ar1      ar2      ar3      ma1    ma2      ma3     ma4
##       -0.3680  -0.3748  -0.2642  -3.1463  3.565  -1.6761  0.2585
## s.e.   0.0067   0.0039   0.0063      NaN    NaN      NaN  0.0078
##       intercept
##           0e+00
## s.e.      1e-04
##
## sigma^2 estimated as 25.04:  log likelihood=-18793.27
## AIC=37604.54   AICc=37604.57   BIC=37665.13
```

```
## get arima(3,0,4)
mod2<-arima(trt.try,order=c(3,0,4),method="ML")
resd(mod2)
```

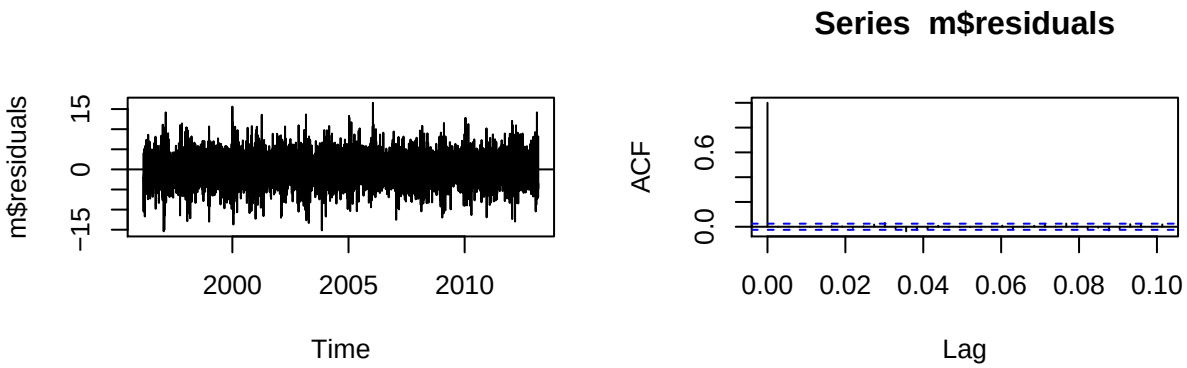


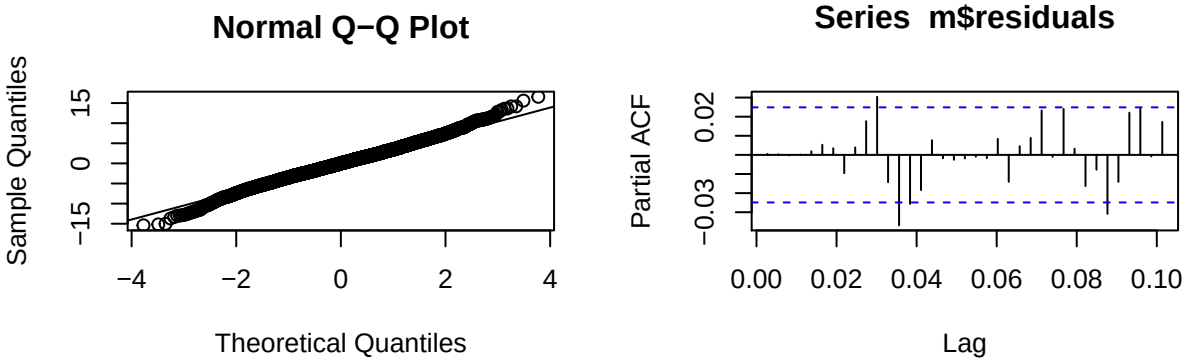







```
## too bad, so we just need one diff
trt.diff<-diff(trt.ts,lag=365)
## difference with lag 365 (yearly seasonality)
mod3<-arima(trt.diff,order=c(3,0,9),method="ML")
## do residuals diagnose
resd(mod3)
```

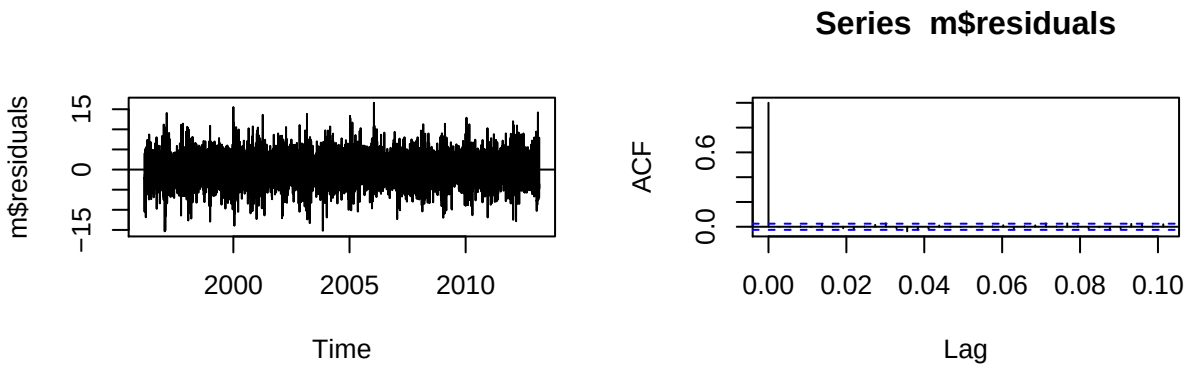


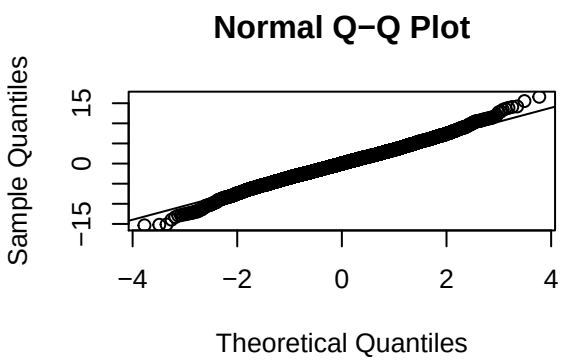


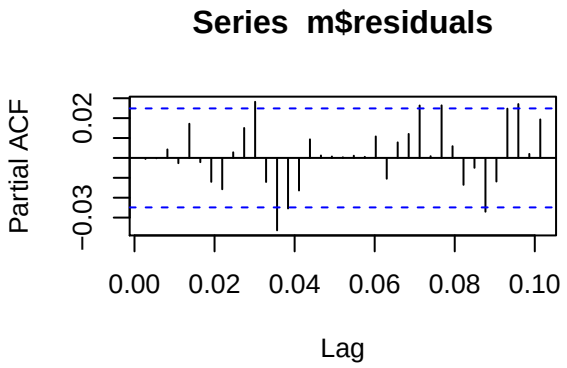

```
## good. try other model
auto.arima(trt.diff)
```

```
## Series: trt.diff
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2
##       1.3027  -0.3479  -0.5455  -0.2504
## s.e.  0.0539   0.0417   0.0534   0.0179
##
## sigma^2 estimated as 13.46:  log likelihood=-16871.41
## AIC=33752.82   AICc=33752.83   BIC=33786.49
```

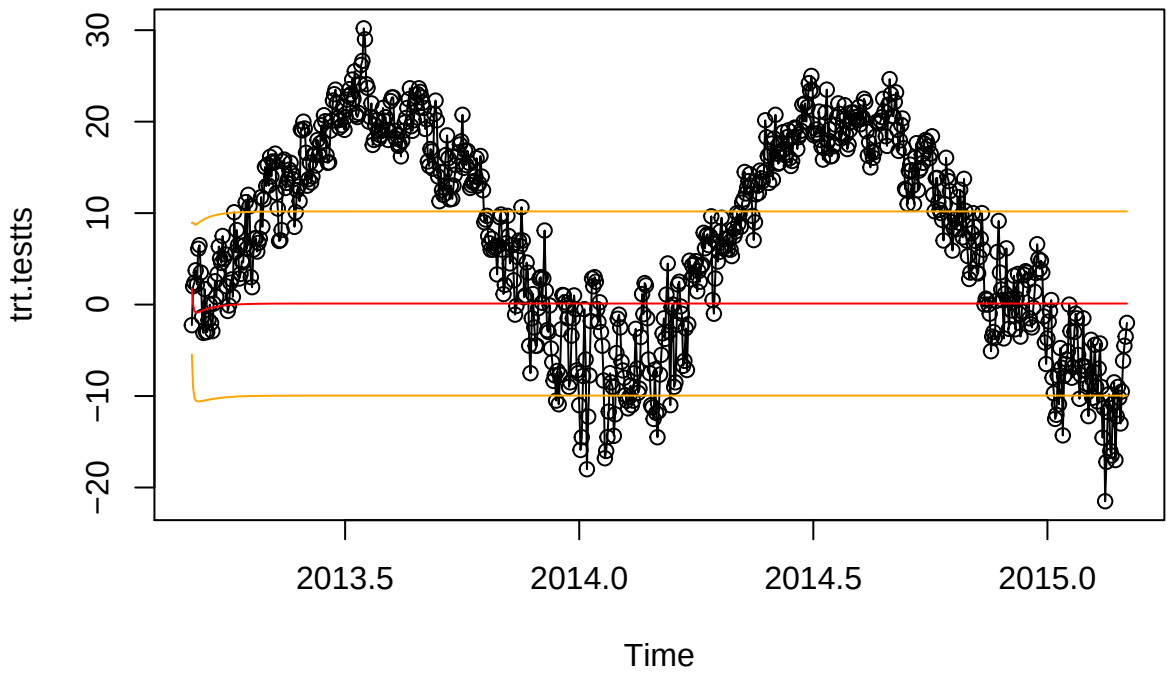
```
## find arima(p,d,q), get (2,0,2)
mod4<-arima(trt.diff,order=c(2,0,2),method="ML")
resd(mod4)
```








```
## looks perfect, just q-q plot have heavy tails.
## predict for next 2 year, test it.
pred.mod4<-predict(mod4,n.ahead=730,se.fit=TRUE)
par(mfcol=c(1,1))
plot(trt.testts)
points(trt.testts)
points(pred.mod4$pred, type="l", col="red")
points(pred.mod4$pred + 1.96*pred.mod4$se, type="l", col="orange")
points(pred.mod4$pred - 1.96*pred.mod4$se, type="l", col="orange")
```




```
pre.result2<-(trt.test>pred.mod4$pred - 1.96*pred.mod4$se & trt.test<pred.mod4$pred + 1.96*pred.mod4$se)
table(pre.result2)
```

```
## pre.result2
## FALSE  TRUE
##   379   351
```

```
PRESS.mod4<-sum((trt.test-pred.mod4$pred)^2)
PRESS.mod4
```

```
## [1] 125400.8
```

```
AIC(mod4)
```

```
## [1] 33754.54
```

Seasonal Holt-Winters Models

```
trt.hw<-HoltWinters(trt.ts)
trt.hw$SSE
```

```
## [1] 71097.76
```

```
PI.hw<-predict(trt.hw,730,prediction.interval=TRUE)
PI.hw
```

```
## Time Series:
## Start = c(2013, 64)
## End = c(2015, 63)
## Frequency = 365
##                   fit        upr        lwr
## 2013.173  -4.71044218   1.924546  -11.34543
## 2013.175  -3.85128196   4.508973  -12.21154
## 2013.178  -4.59410513   5.191837  -14.38005
## 2013.181  -7.83680795   3.192039  -18.86566
## 2013.184  -9.64883316   2.496383  -21.79405
## 2013.186 -10.39096617   2.776307  -23.55824
## 2013.189  -8.24617342   5.869346  -22.36169
## 2013.192  -9.42284568   5.581110  -24.42680
## 2013.195  -7.72908662   8.113561  -23.57173
## 2013.197  -5.69289518  10.946224  -22.33201
## 2013.200  -2.79931872  14.599851  -20.19849
## 2013.203  -1.77448835  16.352891  -19.90187
## 2013.205  -2.43050626  16.396939  -21.25795
## 2013.208  -0.42559487  19.076802  -19.92799
## 2013.211  -0.67992281  19.474836  -20.83468
## 2013.214  -2.21796782  18.568689  -23.00462
## 2013.216   0.02835517  21.428260  -21.37155
## 2013.219   0.45227731  22.448339  -21.54378
## 2013.222  -1.37952273  21.196959  -23.95600
## 2013.225  -2.71885563  20.423493  -25.86120
## 2013.227  -1.31282726  22.381879  -25.00753
## 2013.230  -3.01498488  21.219493  -27.24946
## 2013.233  -1.73760749  23.024879  -26.50009
## 2013.236   1.29802017  26.577489  -23.98145
## 2013.238  -1.95524342  23.830844  -27.74133
## 2013.241  -2.55928667  23.723657  -28.84223
## 2013.244  -0.94567221  25.824907  -27.71625
## 2013.247   1.00050231  28.249993  -26.24899
## 2013.249   0.90683063  28.626959  -26.81330
## 2013.252  -0.57147644  27.611432  -28.75439
## 2013.255  -0.25451192  28.383699  -28.89272
## 2013.258  -2.34997827  26.736409  -31.43637
## 2013.260  -2.83947466  26.688288  -32.36724
## 2013.263  -2.26030166  27.702334  -32.22294
## 2013.266  -1.89810422  28.493183  -32.28939
## 2013.268  -1.30563063  29.508346  -32.11961
## 2013.271  -1.75387061  29.477075  -32.98482
## 2013.274   0.04648280  31.688903  -31.59594
## 2013.277  -0.54372384  31.504889  -32.59234
## 2013.279   1.24559465  33.695315  -31.20413
## 2013.282   2.58077635  35.426707  -30.26515
## 2013.285   3.23111421  36.468532  -30.00630
## 2013.288   5.00768790  38.632036  -28.61666
## 2013.290   2.63306139  36.639937  -31.37381
## 2013.293   2.76482016  37.149968  -31.62033
```

```
## 2013.296   5.15886413  39.918168  -29.60044
## 2013.299   5.13274360  40.262218  -29.99673
## 2013.301   4.37912153  39.874907  -31.11666
## 2013.304   4.91938165  40.777736  -30.93897
## 2013.307   6.23199971  42.449294  -29.98529
## 2013.310   5.62217831  42.194889  -30.95053
## 2013.312   5.86582224  42.790529  -31.05888
## 2013.315   6.04793549  43.321315  -31.22544
## 2013.318   4.95069204  42.569512  -32.66813
## 2013.321   3.91317827  41.874295  -34.04794
## 2013.323   3.94285100  42.243206  -34.35750
## 2013.326   4.26168201  42.898297  -34.37493
## 2013.329   6.18534659  45.155320  -32.78463
## 2013.332   6.25012960  45.550634  -33.05037
## 2013.334   5.86671305  45.494991  -33.76156
## 2013.337   6.12221120  46.075574  -33.83115
## 2013.340   5.25513355  45.530958  -35.02069
## 2013.342   5.03549300  45.631217  -35.56023
## 2013.345   5.53005357  46.443176  -35.38307
## 2013.348   6.25573775  47.483816  -34.97234
## 2013.351   6.21559937  47.756245  -35.32505
## 2013.353   7.37296642  49.223845  -34.47791
## 2013.356   6.80147807  48.960307  -35.35735
## 2013.359   6.03365136  48.498197  -36.43089
## 2013.362   5.28034347  48.048421  -37.48773
## 2013.364   6.23846271  49.307933  -36.83101
## 2013.367   5.90961194  49.278380  -37.45916
## 2013.370   4.37644954  48.042464  -39.28957
## 2013.373   4.16670185  48.127954  -39.79455
## 2013.375   5.86492868  50.119448  -38.38959
## 2013.378   7.55115257  52.097008  -36.99470
## 2013.381  10.48163307  55.316932  -34.35367
## 2013.384  11.35596753  56.478854  -33.76692
## 2013.386  11.46298046  56.871632  -33.94567
## 2013.389  13.13350938  58.826140  -32.55912
## 2013.392  14.25592251  60.230777  -31.71893
## 2013.395  13.97433893  60.229696  -32.28102
## 2013.397  11.58766744  58.121836  -34.94650
## 2013.400  10.18406481  56.995385  -36.62726
## 2013.403   9.94393102  57.030771  -37.14291
## 2013.405  11.97475174  59.335509  -35.38601
## 2013.408  10.85076226  58.483861  -36.78234
## 2013.411  10.35742962  58.261322  -37.54646
## 2013.414  10.92604620  59.099210  -37.24712
## 2013.416  10.38369456  58.824633  -38.05724
## 2013.419  10.47200923  59.179250  -38.23523
## 2013.422   9.42161555  58.393711  -39.55048
## 2013.425  11.04473471  60.280259  -38.19079
## 2013.427  11.60207545  61.099628  -37.89548
## 2013.430  12.41194186  62.170142  -37.34626
## 2013.433  13.71917367  63.736664  -36.29832
## 2013.436  12.53006177  62.805504  -37.74538
## 2013.438  12.07341741  62.605495  -38.45866
## 2013.441  13.35323979  64.140657  -37.43418
```

```
## 2013.444  13.73184941  64.773328  -37.30963
## 2013.447  13.55083437  64.845116  -37.74345
## 2013.449  15.39231866  66.938164  -36.15353
## 2013.452  15.83872187  67.634908  -35.95746
## 2013.455  16.55859810  68.603922  -35.48673
## 2013.458  16.08994375  68.383218  -36.20333
## 2013.460  16.02161229  68.561667  -36.51844
## 2013.463  15.79818441  68.583866  -36.98750
## 2013.466  16.26976635  69.299936  -36.76040
## 2013.468  16.83140683  70.104944  -36.44213
## 2013.471  17.45286976  70.968667  -36.06293
## 2013.474  17.42770306  71.184669  -36.32926
## 2013.477  17.84495546  71.842012  -36.15210
## 2013.479  16.58236612  70.818451  -37.65372
## 2013.482  17.27949555  71.753561  -37.19457
## 2013.485  16.81968266  71.530692  -37.89133
## 2013.488  16.60887829  71.555810  -38.33805
## 2013.490  15.09061371  70.272460  -40.09123
## 2013.493  13.97609090  69.391855  -41.43967
## 2013.496  15.51589406  71.164594  -40.13281
## 2013.499  15.95024911  71.830913  -39.93041
## 2013.501  17.48235674  73.594025  -38.62931
## 2013.504  18.01710192  74.358829  -38.32462
## 2013.507  16.45370348  73.024553  -40.11715
## 2013.510  17.57472248  74.373770  -39.22432
## 2013.512  16.10491730  73.131250  -40.92142
## 2013.515  16.30919736  73.561912  -40.94352
## 2013.518  15.56227178  73.040478  -41.91593
## 2013.521  15.94495739  73.647774  -41.75786
## 2013.523  15.67436680  73.600922  -42.25219
## 2013.526  15.02766363  73.177097  -43.12177
## 2013.529  15.30436839  73.675829  -43.06709
## 2013.532  14.75821537  73.350862  -43.83443
## 2013.534  14.97767454  73.790675  -43.83533
## 2013.537  17.21388883  76.246421  -41.81864
## 2013.540  18.14111388  77.392364  -41.11014
## 2013.542  17.09081565  76.559979  -42.37835
## 2013.545  16.74090640  76.427188  -42.94538
## 2013.548  16.44249555  76.345108  -43.46012
## 2013.551  17.75299270  77.871158  -42.36517
## 2013.553  16.66785994  77.000808  -43.66509
## 2013.556  16.64363905  77.190608  -43.90333
## 2013.559  15.43732415  76.197560  -45.32291
## 2013.562  16.12366686  77.096423  -44.84909
## 2013.564  16.55190516  77.736444  -44.63263
## 2013.567  16.20235112  77.597942  -45.19324
## 2013.570  16.79307213  78.398993  -44.81285
## 2013.573  16.55634749  78.371881  -45.25919
## 2013.575  15.96434103  77.988780  -46.06010
## 2013.578  16.88163003  79.114273  -45.35101
## 2013.581  17.76970580  80.209858  -44.67045
## 2013.584  17.28522020  79.932195  -45.36175
## 2013.586  17.68813771  80.541254  -45.16498
## 2013.589  16.68153750  79.740122  -46.37705
```

```
## 2013.592  16.67515693  79.938542  -46.58823
## 2013.595  16.35442266  79.821947  -47.11310
## 2013.597  17.46678966  81.137799  -46.20422
## 2013.600  19.31607058  83.189917  -44.55778
## 2013.603  19.10275881  83.178800  -44.97328
## 2013.605  20.07638411  84.353984  -44.20122
## 2013.608  19.21426009  83.692789  -45.26427
## 2013.611  18.56206633  83.240900  -46.11677
## 2013.614  17.64250345  82.521023  -47.23602
## 2013.616  16.92472816  82.002321  -48.15286
## 2013.619  18.36104512  83.637104  -46.91501
## 2013.622  18.86547252  84.339396  -46.60845
## 2013.625  17.81406794  83.485260  -47.85712
## 2013.627  16.59575316  82.463623  -49.27212
## 2013.630  17.10154874  83.165511  -48.96241
## 2013.633  16.68088710  82.940361  -49.57859
## 2013.636  17.18670853  83.641119  -49.26770
## 2013.638  18.67558052  85.324358  -47.97320
## 2013.641  19.68335365  86.525932  -47.15922
## 2013.644  19.63226003  86.668080  -47.40356
## 2013.647  20.25726676  87.485772  -46.97124
## 2013.649  18.97588851  86.396529  -48.44475
## 2013.652  19.10120137  86.713430  -48.51103
## 2013.655  18.95595465  86.759231  -48.84732
## 2013.658  17.95298336  85.946771  -50.04080
## 2013.660  18.05702299  86.240789  -50.12674
## 2013.663  17.29601490  85.669231  -51.07720
## 2013.666  15.52253438  84.084678  -53.03961
## 2013.668  16.85488105  85.605433  -51.89567
## 2013.671  17.82564115  86.764086  -51.11280
## 2013.674  16.40192356  85.527750  -52.72390
## 2013.677  15.40732232  84.720025  -53.90538
## 2013.679  16.66512905  86.164205  -52.83395
## 2013.682  17.71157767  87.396528  -51.97337
## 2013.685  16.85098403  86.721315  -53.01935
## 2013.688  14.47885979  84.534080  -55.57636
## 2013.690  12.87593162  83.115555  -57.36369
## 2013.693  11.60309143  82.026635  -58.82045
## 2013.696  12.71167207  83.318657  -57.89531
## 2013.699  12.01649211  82.806443  -58.77346
## 2013.701  11.20565276  82.178098  -59.76679
## 2013.704  10.04490411  81.199375  -61.10957
## 2013.707  10.51645779  81.852490  -60.81957
## 2013.710  11.96230634  83.479440  -59.55483
## 2013.712  11.95601726  83.653794  -59.74176
## 2013.715  11.20155231  83.079518  -60.67641
## 2013.718  12.31901571  84.376720  -59.73869
## 2013.721  11.92075115  84.157747  -60.31624
## 2013.723  11.55368922  83.969533  -60.86215
## 2013.726   9.59201460  82.186265  -63.00224
## 2013.729   9.67438151  82.446601  -63.09784
## 2013.732   8.58991595  81.539671  -64.35984
## 2013.734   8.07764295  81.204502  -65.04922
## 2013.737   7.99095389  81.294490  -65.31258
```

```
## 2013.740   9.68055137  83.160339  -63.79924
## 2013.742   8.65357806  82.309196  -65.00204
## 2013.745   8.98309671  82.814125  -64.84793
## 2013.748   9.14672298  83.152747  -64.85930
## 2013.751   9.48327971  83.663886  -64.69733
## 2013.753  11.33043581  85.685215  -63.02434
## 2013.756  11.71985980  86.248405  -62.80869
## 2013.759   9.78118465  84.483091  -64.92072
## 2013.762   8.76649446  83.641361  -66.10837
## 2013.764   9.49808856  84.545516  -65.54934
## 2013.767  10.96992748  86.189521  -64.24967
## 2013.770  10.89821589  86.289582  -64.49315
## 2013.773   9.91896061  85.481708  -65.64379
## 2013.775   9.31634312  85.050085  -66.41740
## 2013.778   9.00588950  84.910240  -66.89846
## 2013.781   9.73957533  85.814152  -66.33500
## 2013.784  12.18242482  88.426848  -64.06200
## 2013.786  10.07765214  86.491544  -66.33624
## 2013.789   9.14224527  85.725231  -67.44074
## 2013.792   9.15356873  85.905275  -67.59814
## 2013.795   8.53620224  85.456260  -68.38386
## 2013.797   7.96198881  85.050030  -69.12605
## 2013.800   8.51942258  85.775082  -68.73624
## 2013.803   7.64619651  85.069111  -69.77672
## 2013.805   6.35791247  83.947721  -71.23190
## 2013.808   5.68551700  83.441862  -72.07083
## 2013.811   7.04283251  84.965359  -70.87969
## 2013.814   7.43599037  85.524343  -70.65236
## 2013.816   7.09187276  85.345701  -71.16196
## 2013.819   4.91176217  83.330717  -73.50719
## 2013.822   4.26203786  82.845772  -74.32170
## 2013.825   4.71595718  83.464126  -74.03221
## 2013.827   6.25024568  85.162506  -72.66201
## 2013.830   5.82485718  84.900869  -73.25115
## 2013.833   4.44641662  83.685842  -74.79301
## 2013.836   2.60112688  82.003629  -76.80137
## 2013.838   3.10986215  82.675106  -76.45538
## 2013.841   2.76394647  82.491601  -76.96371
## 2013.844   2.45642917  82.346164  -77.43331
## 2013.847   2.30192695  82.353413  -77.74956
## 2013.849   3.27622493  83.489137  -76.93669
## 2013.852   2.25624708  82.630261  -78.11777
## 2013.855   2.04314394  82.577937  -78.49165
## 2013.858   0.04376731  80.739019  -80.65148
## 2013.860   0.04631535  80.901708  -80.80908
## 2013.863  -0.54327330  80.471943  -81.55849
## 2013.866  -1.61909196  79.555634  -82.79382
## 2013.868  -0.06004756  81.273875  -81.39397
## 2013.871  -1.27622285  80.216585  -82.76903
## 2013.874  -2.65041956  79.000964  -84.30180
## 2013.877  -3.92313231  77.886521  -85.73279
## 2013.879  -3.58867758  78.378939  -85.55629
## 2013.882  -2.03282699  80.092449  -84.15810
## 2013.885  -0.26223475  82.020398  -82.54487
```

```
## 2013.888  -2.10080567  80.338885  -84.54050
## 2013.890  -0.55290721  82.043541  -83.14936
## 2013.893  -1.60094677  81.151963  -84.35386
## 2013.896  -3.03671396  79.872362  -85.94579
## 2013.899  -1.44863452  81.616314  -84.51358
## 2013.901  -1.88755170  81.332978  -85.10808
## 2013.904  -2.08161286  81.294207  -85.45743
## 2013.907  -2.23600279  81.294819  -85.76682
## 2013.910  -1.65028026  82.035256  -85.33582
## 2013.912  -3.70767554  80.132290  -87.54764
## 2013.915  -4.20641358  79.787697  -88.20052
## 2013.918  -4.08131202  80.066661  -88.22928
## 2013.921  -4.87038693  79.431168  -89.17194
## 2013.923  -3.62803357  80.826824  -88.08289
## 2013.926  -5.29547988  79.312403  -89.90336
## 2013.929  -4.08572410  80.674907  -88.84636
## 2013.932  -2.28635767  82.626747  -87.19946
## 2013.934  -3.73724377  81.328062  -88.80255
## 2013.937  -4.34073632  80.876498  -89.55797
## 2013.940  -4.20028137  81.168611  -89.56917
## 2013.942  -6.21922467  79.301057  -91.73951
## 2013.945  -7.64930473  78.022099  -93.32071
## 2013.948 -10.16571284  75.656546  -95.98797
## 2013.951  -9.88686269  76.085988  -95.85971
## 2013.953  -8.85416335  77.269015  -94.97734
## 2013.956  -7.67254384  78.600700  -93.94579
## 2013.959  -7.06101518  79.362034  -93.48406
## 2013.962  -6.07430779  80.498287  -92.64690
## 2013.964  -5.69034076  81.031542  -92.41222
## 2013.967  -8.08521244  78.785702  -94.95613
## 2013.970  -7.78629163  79.233399  -94.80598
## 2013.973  -9.11362543  78.054588  -96.28184
## 2013.975  -9.12835309  78.188130  -96.44484
## 2013.978  -8.31973915  79.144762  -95.78424
## 2013.981  -7.37491426  80.237356  -94.98718
## 2013.984  -8.27164118  79.488148  -96.03143
## 2013.986  -8.46479660  79.442265  -96.37186
## 2013.989  -8.48188203  79.572205  -96.53597
## 2013.992  -8.42120157  79.779666  -96.62207
## 2013.995  -7.32818525  81.019219  -95.67559
## 2013.997  -5.68264332  82.811055  -94.17634
## 2014.000  -6.71380157  81.925949  -95.35355
## 2014.003  -9.52172145  79.263842  -98.30728
## 2014.005  -8.84680779  80.084329  -97.77794
## 2014.008  -5.62009994  83.456372  -94.69657
## 2014.011  -6.31470628  82.906864  -95.53628
## 2014.014  -9.62016247  79.746271  -98.98660
## 2014.016 -12.17858374  77.332478 -101.68965
## 2014.019 -14.86169853  74.793759 -104.51716
## 2014.022 -14.79538890  75.004232 -104.59501
## 2014.025 -14.61063764  75.332915 -104.55419
## 2014.027 -12.81527029  77.271984 -102.90252
## 2014.030 -11.28962875  78.941099 -101.52036
## 2014.033 -10.44994336  79.924030 -100.82392
```

```
## 2014.036 -13.90739574  76.609596 -104.42439
## 2014.038 -13.25870932  77.401076 -103.91849
## 2014.041 -11.47328631  79.329068 -102.27564
## 2014.044 -10.17283306  80.771866 -101.11753
## 2014.047  -7.27591917  83.810902  -98.36274
## 2014.049  -4.03608609  87.192637  -95.26481
## 2014.052  -6.28370650  85.086697  -97.65411
## 2014.055  -6.20441224  85.307453  -97.71628
## 2014.058  -3.97024094  87.682867  -95.62335
## 2014.060  -2.35069252  89.443442  -94.14483
## 2014.063  -2.20720352  89.727740  -94.14215
## 2014.066  -2.07736969  89.998168  -94.15291
## 2014.068  -2.77776298  89.438155  -94.99368
## 2014.071  -3.36127965  88.994805  -95.71736
## 2014.074  -2.60860190  89.887436  -95.10464
## 2014.077  -3.70390546  88.931875  -96.33969
## 2014.079  -4.07261027  88.702703  -96.84792
## 2014.082  -6.23937908  86.675257  -99.15401
## 2014.085  -6.24325001  86.810500  -99.29700
## 2014.088  -6.84318698  86.349469 -100.03584
## 2014.090  -8.26643438  85.064921 -101.59779
## 2014.093 -11.48747593  81.982373 -104.95733
## 2014.096 -13.81046126  79.797677 -107.41860
## 2014.099 -14.00928105  79.736942 -107.75550
## 2014.101 -14.58043994  79.303665 -108.46454
## 2014.104 -13.80671511  80.215069 -107.82850
## 2014.107  -9.92296068  84.236302 -104.08222
## 2014.110  -7.64923519  86.647305 -101.94578
## 2014.112  -5.95673911  88.476880 -100.39036
## 2014.115  -4.93611751  89.634381  -99.50662
## 2014.118  -5.59824220  89.108938 -100.30542
## 2014.121  -6.46150486  88.382160 -101.30517
## 2014.123  -8.34401683  86.635937 -103.32397
## 2014.126  -8.63039030  86.485656 -103.74644
## 2014.129  -9.10786676  86.144079 -104.35981
## 2014.132  -8.52604647  86.861604 -103.91370
## 2014.134  -9.92663711  85.596526 -105.44980
## 2014.137 -10.30048169  85.358002 -105.95897
## 2014.140  -7.49765177  88.295962 -103.29127
## 2014.142  -3.97898837  91.949564  -99.90754
## 2014.145  -3.19354563  92.869756  -99.25685
## 2014.148  -3.06104986  93.136813  -99.25891
## 2014.151  -1.92071727  94.411518  -98.25295
## 2014.153  -0.14034842  96.326073  -96.60677
## 2014.156   1.15781696  97.758237  -95.44260
## 2014.159  -0.56408838  96.170145  -97.29832
## 2014.162  -0.74904918  96.118813  -97.61691
## 2014.164  -1.17089372  95.830413  -98.17220
## 2014.167  -2.37821676  94.756352  -99.51279
## 2014.170  -3.50494634  93.762701 -100.77259
## 2014.173  -4.96593050  92.501174 -102.43303
## 2014.175  -4.10677027  93.492959 -101.70650
## 2014.178  -4.84959345  92.882582 -102.58177
## 2014.181  -8.09229626  89.772145 -105.95674
```

```
## 2014.184  -9.90432147  88.092207 -107.90085
## 2014.186 -10.64645448  87.481984 -108.77489
## 2014.189  -8.50166174  89.758510 -106.76183
## 2014.192  -9.67833399  88.713394 -108.07006
## 2014.195  -7.98457493  90.538533 -106.50768
## 2014.197  -5.94838349  92.705930 -104.60270
## 2014.200  -3.05480703  95.730538 -101.84015
## 2014.203  -2.02997666  96.886227 -100.94618
## 2014.205  -2.68599457  96.360894 -101.73288
## 2014.208  -0.68108318  98.496318  -99.85848
## 2014.211  -0.93541112  98.372331 -100.24315
## 2014.214  -2.47345613  96.964457 -101.91137
## 2014.216  -0.22713314  99.340780  -99.79505
## 2014.219   0.19678900  99.894533  -99.50095
## 2014.222  -1.63501104  98.192394 -101.46242
## 2014.225  -2.97434394  96.982555 -102.93124
## 2014.227  -1.56831558  98.517910 -101.65454
## 2014.230  -3.27047320  96.944911 -103.48586
## 2014.233  -1.99309580  98.351282 -102.33747
## 2014.236   1.04253186 101.515737  -99.43067
## 2014.238  -2.21073174  98.391136 -102.81260
## 2014.241  -2.81477498  97.915590 -103.54514
## 2014.244  -1.20116053  99.657539 -102.05986
## 2014.247   0.74501400 101.731885 -100.24186
## 2014.249   0.65134231 101.766222 -100.46354
## 2014.252  -0.82696475 100.415762 -102.06969
## 2014.255  -0.51000023 100.860412 -101.88041
## 2014.258  -2.60546658  98.892471 -104.10340
## 2014.260  -3.09496297  98.530340 -104.72027
## 2014.263  -2.51578997  99.236718 -104.26830
## 2014.266  -2.15359253  99.725962 -104.03315
## 2014.268  -1.56111894 100.445325 -103.56756
## 2014.271  -2.00935892 100.123816 -104.14253
## 2014.274  -0.20900551 102.050743 -102.46875
## 2014.277  -0.79921216 101.586954 -103.18538
## 2014.279   0.99010634 103.502534 -101.52232
## 2014.282   2.32528804 104.963821 -100.31325
## 2014.285   2.97562590 105.740111  -99.78886
## 2014.288   4.75219959 107.642482  -98.13808
## 2014.290   2.37757307 105.393499 -100.63835
## 2014.293   2.50933185 105.650748 -100.63208
## 2014.296   4.90337582 108.170130  -98.36338
## 2014.299   4.87725529 108.269196  -98.51469
## 2014.301   4.12363321 107.640608  -99.39334
## 2014.304   4.66389334 108.305752  -98.97797
## 2014.307   5.97651139 109.743104  -97.79008
## 2014.310   5.36668999 109.257867  -98.52449
## 2014.312   5.61033393 109.625945  -98.40528
## 2014.315   5.79244718 109.932344  -98.34745
## 2014.318   4.69520373 108.959239  -99.56883
## 2014.321   3.65768996 108.045715 -100.73034
## 2014.323   3.68736269 108.199231 -100.82451
## 2014.326   4.00619370 108.641758 -100.62937
## 2014.329   5.92985828 110.688973  -98.82926
```

```
## 2014.332   5.99464128 110.877161  -98.88788
## 2014.334   5.61122474 110.617005  -99.39456
## 2014.337   5.86672289 110.995618  -99.26217
## 2014.340   4.99964524 110.251512 -100.25222
## 2014.342   4.78000469 110.154699 -100.59469
## 2014.345   5.27456525 110.771944 -100.22281
## 2014.348   6.00024944 111.620171  -99.61967
## 2014.351   5.96011106 111.702433  -99.78221
## 2014.353   7.11747811 112.982059  -98.74710
## 2014.356   6.54598976 112.532688  -99.44071
## 2014.359   5.77816305 111.886839 -100.33051
## 2014.362   5.02485516 111.255368 -101.20566
## 2014.364   5.98297440 112.335185 -100.36924
## 2014.367   5.65412363 112.127892 -100.81965
## 2014.370   4.12096122 110.716150 -102.47423
## 2014.373   3.91121353 110.627684 -102.80526
## 2014.375   5.60944037 112.447055 -101.22817
## 2014.378   7.29566426 114.254285  -99.66296
## 2014.381  10.22614476 117.305636  -96.85335
## 2014.384  11.10047921 118.300704  -96.09975
## 2014.386  11.20749215 118.528315  -96.11333
## 2014.389  12.87802107 120.319307  -94.56326
## 2014.392  14.00043420 121.562047  -93.56118
## 2014.395  13.71885062 121.400657  -93.96296
## 2014.397  11.33217913 119.134045  -96.46969
## 2014.400   9.92857650 117.850368  -97.99321
## 2014.403   9.68844271 117.730027  -98.35314
## 2014.405  11.71926343 119.880507  -96.44198
## 2014.408  10.59527395 118.876045  -97.68550
## 2014.411  10.10194131 118.502109  -98.29823
## 2014.414  10.67055789 119.189990  -97.84887
## 2014.416  10.12820624 118.766772  -98.51036
## 2014.419  10.21652092 118.974089  -98.54105
## 2014.422   9.16612724 118.042569  -99.71031
## 2014.425  10.78924640 119.784431  -98.20594
## 2014.427  11.34658714 120.460386  -97.76721
## 2014.430  12.15645355 121.388738  -97.07583
## 2014.433  13.46368536 122.814326  -95.88696
## 2014.436  12.27457345 121.743443  -97.19430
## 2014.438  11.81792910 121.404900  -97.76904
## 2014.441  13.09775147 122.802697  -96.60719
## 2014.444  13.47636110 123.299154  -96.34643
## 2014.447  13.29534606 123.235860  -96.64517
## 2014.449  15.13683035 125.194940  -94.92128
## 2014.452  15.58323355 125.758813  -94.59235
## 2014.455  16.30310979 126.596034  -93.98981
## 2014.458  15.83445544 126.244599  -94.57569
## 2014.460  15.76612398 126.293364  -94.76112
## 2014.463  15.54269610 126.186907  -95.10152
## 2014.466  16.01427804 126.775338  -94.74678
## 2014.468  16.57591851 127.453703  -94.30187
## 2014.471  17.19738144 128.191768  -93.79701
## 2014.474  17.17221475 128.283082  -93.93865
## 2014.477  17.58946715 128.816692  -93.63776
```

```
## 2014.479  16.32687781 127.670339  -95.01658
## 2014.482  17.02400724 128.483583  -94.43557
## 2014.485  16.56419435 128.139765  -95.01138
## 2014.488  16.35338998 128.044834  -95.33805
## 2014.490  14.83512540 126.642323  -96.97207
## 2014.493  13.72060258 125.643435  -98.20223
## 2014.496  15.26040574 127.298752  -96.77794
## 2014.499  15.69476080 127.848503  -96.45898
## 2014.501  17.22686842 129.495888  -95.04215
## 2014.504  17.76161360 130.145792  -94.62256
## 2014.507  16.19821517 128.697435  -96.30100
## 2014.510  17.31923417 129.933377  -95.29491
## 2014.512  15.84942899 128.578378  -96.87952
## 2014.515  16.05370905 128.897348  -96.78993
## 2014.518  15.30678347 128.264995  -97.65143
## 2014.521  15.68946907 128.762138  -97.38320
## 2014.523  15.41887849 128.605888  -97.76813
## 2014.526  14.77217532 128.073411  -98.52906
## 2014.529  15.04888008 128.464226  -98.36647
## 2014.532  14.50272705 128.032069  -99.02662
## 2014.534  14.72218622 128.365410  -98.92104
## 2014.537  16.95840052 130.715392  -96.79859
## 2014.540  17.88562557 131.756271  -95.98502
## 2014.542  16.83532733 130.819514  -97.14886
## 2014.545  16.48541808 130.583032  -97.61220
## 2014.548  16.18700724 130.397937  -98.02392
## 2014.551  17.49750439 131.821637  -96.82663
## 2014.553  16.41237163 130.849595  -98.02485
## 2014.556  16.38815074 130.938353  -98.16205
## 2014.559  15.18183584 129.844906  -99.48123
## 2014.562  15.86817855 130.644005  -98.90765
## 2014.564  16.29641685 131.184890  -98.59206
## 2014.567  15.94686281 130.947872  -99.05415
## 2014.570  16.53758382 131.651018  -98.57585
## 2014.573  16.30085918 131.526610  -98.92489
## 2014.575  15.70885272 131.046810  -99.62910
## 2014.578  16.62614172 132.076197  -98.82391
## 2014.581  17.51421749 133.076262  -98.04783
## 2014.584  17.02973188 132.703656  -98.64419
## 2014.586  17.43264940 133.218346  -98.35305
## 2014.589  16.42604919 132.323411  -99.47131
## 2014.592  16.41966861 132.428587  -99.58925
## 2014.595  16.09893435 132.219303 -100.02143
## 2014.597  17.21130135 133.443013  -99.02041
## 2014.600  19.06058227 135.403531  -97.28237
## 2014.603  18.84727049 135.301349  -97.60681
## 2014.605  19.82089580 136.385999  -96.74421
## 2014.608  18.95877178 135.634793  -97.71725
## 2014.611  18.30657802 135.093413  -98.48026
## 2014.614  17.38701514 134.284559  -99.51053
## 2014.616  16.66923984 133.677387 -100.33891
## 2014.619  18.10555681 135.224203  -99.01309
## 2014.622  18.60998420 135.839025  -98.61906
## 2014.625  17.55857962 134.897912  -99.78075
```

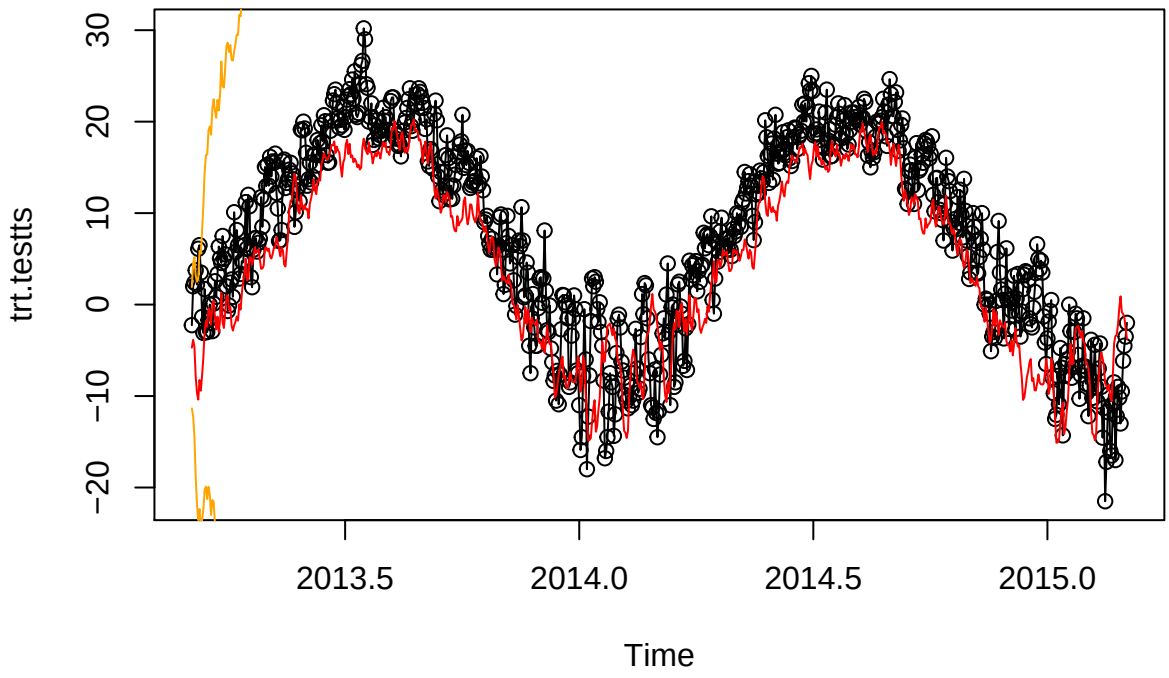
```
## 2014.627  16.34026485 133.789784 -101.10925
## 2014.630  16.84606043 134.405664 -100.71354
## 2014.633  16.42539878 134.094984 -101.24419
## 2014.636  16.93122022 134.710684 -100.84824
## 2014.638  18.42009221 136.309332  -99.46915
## 2014.641  19.42786534 137.426779  -98.57105
## 2014.644  19.37677172 137.485258  -98.73171
## 2014.647  20.00177844 138.219735  -98.21618
## 2014.649  18.72040020 137.047726  -99.60693
## 2014.652  18.84571306 137.282307  -99.59088
## 2014.655  18.70046634 137.246228  -99.84530
## 2014.658  17.69749504 136.352324 -100.95733
## 2014.660  17.80153468 136.565331 -100.96226
## 2014.663  17.04052659 135.913190 -101.83214
## 2014.666  15.26704606 134.248477 -103.71438
## 2014.668  16.59939274 135.689492 -102.49071
## 2014.671  17.57015284 136.768821 -101.62852
## 2014.674  16.14643525 135.453574 -103.16070
## 2014.677  15.15183401 134.567344 -104.26368
## 2014.679  16.40964073 135.933425 -103.11414
## 2014.682  17.45608936 137.088049 -102.17587
## 2014.685  16.59549571 136.335533 -103.14454
## 2014.688  14.22337148 134.071389 -105.62465
## 2014.690  12.62044331 132.576344 -107.33546
## 2014.693  11.34760312 131.411290 -108.71608
## 2014.696  12.45618376 132.627560 -107.71519
## 2014.699  11.76100380 132.039973 -108.51797
## 2014.701  10.95016445 131.336631 -109.43630
## 2014.704   9.78941580 130.283283 -110.70445
## 2014.707  10.26096947 130.862143 -110.34020
## 2014.710  11.70681803 132.415201 -109.00157
## 2014.712  11.70052894 132.516027 -109.11497
## 2014.715  10.94606400 131.868582 -109.97645
## 2014.718  12.06352740 133.092971 -108.96592
## 2014.721  11.66526284 132.801537 -109.47101
## 2014.723  11.29820091 132.541212 -109.94481
## 2014.726   9.33652629 130.686180 -112.01313
## 2014.729   9.41889320 130.875097 -112.03731
## 2014.732   8.33442763 129.897087 -113.22823
## 2014.734   7.82215464 129.491177 -113.84687
## 2014.737   7.73546558 129.510758 -114.03983
## 2014.740   9.42506306 131.306532 -112.45641
## 2014.742   8.39808975 130.385644 -113.58946
## 2014.745   8.72760840 130.821155 -113.36594
## 2014.748   8.89123467 131.090682 -113.30821
## 2014.751   9.22779140 131.533048 -113.07747
## 2014.753  11.07494750 133.485922 -111.33603
## 2014.756  11.46437148 133.980972 -111.05223
## 2014.759   9.52569634 132.147832 -113.09644
## 2014.762   8.51100615 131.238587 -114.21657
## 2014.764   9.24260025 132.075535 -113.59033
## 2014.767  10.71443916 133.652637 -112.22376
## 2014.770  10.64272758 133.686100 -112.40064
## 2014.773   9.66347230 132.811928 -113.48498
```

```
## 2014.775   9.06085481 132.314305 -114.19260
## 2014.778   8.75040119 132.108756 -114.60795
## 2014.781   9.48408702 132.947258 -113.97908
## 2014.784  11.92693651 135.494834 -111.64096
## 2014.786   9.82216383 133.494700 -113.85037
## 2014.789   8.88675696 132.663843 -114.89033
## 2014.792   8.89808042 132.779628 -114.98347
## 2014.795   8.28071392 132.266635 -115.70521
## 2014.797   7.70650050 131.796707 -116.38371
## 2014.800   8.26393427 132.458339 -115.93047
## 2014.803   7.39070820 131.689224 -116.90781
## 2014.805   6.10242415 130.504964 -118.30012
## 2014.808   5.43002869 129.936505 -119.07645
## 2014.811   6.78734420 131.397671 -117.82298
## 2014.814   7.18050205 131.894592 -117.53359
## 2014.816   6.83638445 131.654152 -117.98138
## 2014.819   4.65627386 129.577632 -120.26508
## 2014.822   4.00654955 129.031414 -121.01831
## 2014.825   4.46046887 129.588753 -120.66782
## 2014.827   5.99475737 131.226376 -119.23686
## 2014.830   5.56936887 130.904237 -119.76550
## 2014.833   4.19092831 129.628960 -121.24710
## 2014.836   2.34563857 127.886750 -123.19547
## 2014.838   2.85437384 128.498480 -122.78973
## 2014.841   2.50845815 128.255475 -123.23856
## 2014.844   2.20094085 128.050784 -123.64890
## 2014.847   2.04643863 127.999024 -123.90615
## 2014.849   3.02073662 129.075981 -123.03451
## 2014.852   2.00075877 128.158578 -124.15706
## 2014.855   1.78765562 128.047967 -124.47266
## 2014.858  -0.21172101 126.150999 -126.57444
## 2014.860  -0.20917296 126.255873 -126.67422
## 2014.863  -0.79876161 125.768528 -127.36605
## 2014.866  -1.87458027 124.794869 -128.54403
## 2014.868  -0.31553587 126.455992 -127.08706
## 2014.871  -1.53171116 125.341813 -128.40524
## 2014.874  -2.90590787 124.069530 -129.88135
## 2014.877  -4.17862062 122.898650 -131.25589
## 2014.879  -3.84416590 123.334856 -131.02319
## 2014.882  -2.28831530 124.992376 -129.56901
## 2014.885  -0.51772306 126.864556 -127.90000
## 2014.888  -2.35629398 125.127493 -129.84008
## 2014.890  -0.80839552 126.776818 -128.39361
## 2014.893  -1.85643508 125.830125 -129.54299
## 2014.896  -3.29220227 124.495623 -131.08003
## 2014.899  -1.70412283 126.184888 -129.59313
## 2014.901  -2.14304002 125.847076 -130.13316
## 2014.904  -2.33710118 125.754041 -130.42824
## 2014.907  -2.49149110 125.700597 -130.68358
## 2014.910  -1.90576857 126.387187 -130.19872
## 2014.912  -3.96316385 124.430579 -132.35691
## 2014.915  -4.46190189 124.032549 -132.95635
## 2014.918  -4.33680033 124.258280 -132.93188
## 2014.921  -5.12587524 123.569756 -133.82151
```

```
## 2014.923  -3.88352189 124.912582 -132.67963
## 2014.926  -5.55096820 123.345530 -134.44747
## 2014.929  -4.34121241 124.655601 -133.33803
## 2014.932  -2.54184598 126.555206 -131.63890
## 2014.934  -3.99273208 125.204480 -133.18994
## 2014.937  -4.59622463 124.701070 -133.89352
## 2014.940  -4.45576968 124.941530 -133.85307
## 2014.942  -6.47471298 123.022515 -135.97194
## 2014.945  -7.90479304 121.692285 -137.50187
## 2014.948 -10.42120115 119.275651 -140.11805
## 2014.951 -10.14235100 119.654199 -139.93890
## 2014.953  -9.10965166 120.786519 -139.00582
## 2014.956  -7.92803215 122.067683 -137.92375
## 2014.959  -7.31650349 122.778680 -137.41169
## 2014.962  -6.32979611 123.864780 -136.52437
## 2014.964  -5.94582907 124.348063 -136.23972
## 2014.967  -8.34070076 122.052432 -138.73383
## 2014.970  -8.04177994 122.450519 -138.53408
## 2014.973  -9.36911375 121.222275 -139.96050
## 2014.975  -9.38384140 121.306562 -140.07424
## 2014.978  -8.57522746 122.214116 -139.36457
## 2014.981  -7.63040257 123.257806 -138.51861
## 2014.984  -8.52712950 122.459870 -139.51413
## 2014.986  -8.72028491 122.365431 -139.80600
## 2014.989  -8.73737034 122.446987 -139.92173
## 2014.992  -8.67668988 122.606235 -139.95961
## 2014.995  -7.58367356 123.797745 -138.96509
## 2014.997  -5.93813163 125.541707 -137.41797
## 2015.000  -6.96928988 124.608895 -138.54748
## 2015.003  -9.77720976 121.899248 -141.45367
## 2015.005  -9.10229610 122.672362 -140.87695
## 2015.008  -5.87558825 125.997196 -137.74837
## 2015.011  -6.57019459 125.400643 -138.54103
## 2015.014  -9.87565078 122.193168 -141.94447
## 2015.016 -12.43407205 119.732655 -144.60080
## 2015.019 -15.11718684 117.147375 -147.38175
## 2015.022 -15.05087721 117.311448 -147.41320
## 2015.025 -14.86612596 117.593891 -147.32614
## 2015.027 -13.07075860 119.486877 -145.62839
## 2015.030 -11.54511706 121.110066 -144.20030
## 2015.033 -10.70543167 122.047227 -143.45809
## 2015.036 -14.16288405 118.687179 -147.01295
## 2015.038 -13.51419763 119.433198 -146.46159
## 2015.041 -11.72877462 121.315883 -144.77343
## 2015.044 -10.42832137 122.713526 -143.57017
## 2015.047  -7.53140748 125.707560 -140.77037
## 2015.049  -4.29157440 129.044442 -137.62759
## 2015.052  -6.53919482 126.893800 -139.97219
## 2015.055  -6.45990055 127.070002 -139.98980
## 2015.058  -4.22572926 129.401011 -137.85247
## 2015.060  -2.60618083 131.117327 -136.32969
## 2015.063  -2.46269184 131.357513 -136.28290
## 2015.066  -2.33285800 131.583975 -136.24969
## 2015.068  -3.03325129 130.980139 -137.04664
```

```
## 2015.071  -3.61676797 130.493111 -137.72665
## 2015.074  -2.86409021 131.342208 -137.07039
## 2015.077  -3.95939377 130.343254 -138.26204
## 2015.079  -4.32809859 130.070830 -138.72703
## 2015.082  -6.49486739 128.000273 -140.99001
## 2015.085  -6.49873832 128.092546 -141.09002
## 2015.088  -7.09867529 127.588683 -141.78603
## 2015.090  -8.52192270 126.261442 -143.30529
## 2015.093 -11.74296424 123.136338 -146.62227
## 2015.096 -14.06594957 120.909222 -149.04112
## 2015.099 -14.26476936 120.806203 -149.33574
## 2015.101 -14.83592825 120.330778 -150.00263
## 2015.104 -14.06220342 121.200169 -149.32458
## 2015.107 -10.17844899 125.179521 -145.53642
## 2015.110  -7.90472350 127.548777 -143.35822
## 2015.112  -6.21222743 129.336737 -141.76119
## 2015.115  -5.19160582 130.452755 -140.83597
## 2015.118  -5.85373051 129.885959 -141.59342
## 2015.121  -6.71699317 129.117959 -142.55194
## 2015.123  -8.59950514 127.330642 -144.52965
## 2015.126  -8.88587861 127.139397 -144.91115
## 2015.129  -9.36335508 126.756983 -145.48369
## 2015.132  -8.78153478 127.433800 -144.99687
## 2015.134 -10.18212542 126.128139 -146.49239
## 2015.137 -10.55597000 125.849158 -146.96110
## 2015.140  -7.75314009 128.746786 -144.25307
## 2015.142  -4.23447669 132.360181 -140.82913
## 2015.145  -3.44903395 133.240290 -140.13836
## 2015.148  -3.31653817 133.467387 -140.10046
## 2015.151  -2.17620558 134.702255 -139.05467
## 2015.153  -0.39583673 136.577094 -137.36877
## 2015.156   0.90232865 137.969664 -136.16501
## 2015.159  -0.81957669 136.342099 -137.98125
## 2015.162  -1.00453750 136.251414 -138.26049
## 2015.164  -1.42638203 135.923780 -138.77654
## 2015.167  -2.63370507 134.810603 -140.07801
## 2015.170  -3.76043466 133.777955 -141.29882
```

```
plot(trt.testts)
points(trt.testts)
points(PI.hw[,1], type="l", col="red")
points(PI.hw[,2], type="l", col="orange")
points(PI.hw[,3], type="l", col="orange")
```




```
PRESS.hw<-sum((trt.test-PI.hw[,1])^2)
PRESS.hw
```

```
## [1] 23672.51
```

```
pre.result3<-(trt.test<PI.hw[,2] & trt.test>PI.hw[,3])
table(pre.result3)
```

```
## pre.result3
## FALSE  TRUE
##     3   727
```

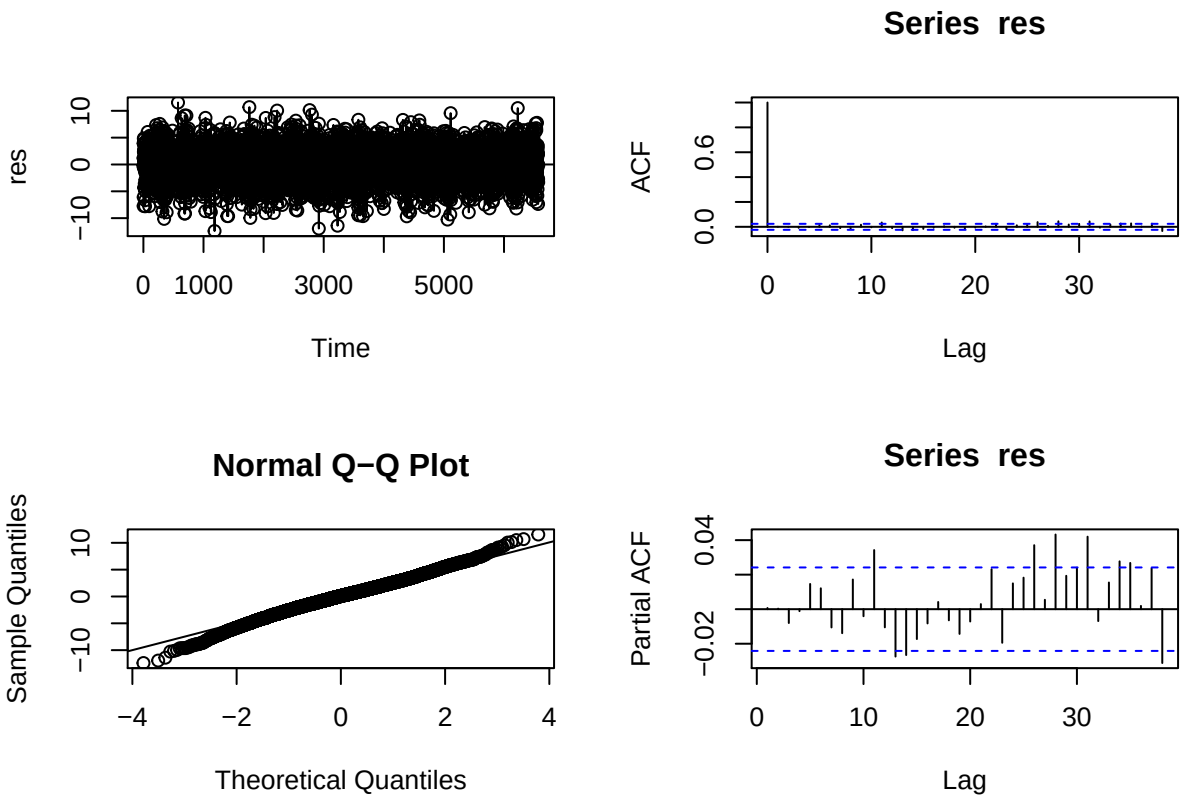
regression with arima residual

```
auto.arima(reg5$residuals)
```

```
## Series: reg5$residuals
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2
##       1.1363  -0.2604  -0.3669  -0.1734
## s.e.  0.0878   0.0614   0.0872   0.0174
##
```

```
## sigma^2 estimated as 7.414:  log likelihood=-15903.76
## AIC=31817.52   AICc=31817.53   BIC=31851.47
```

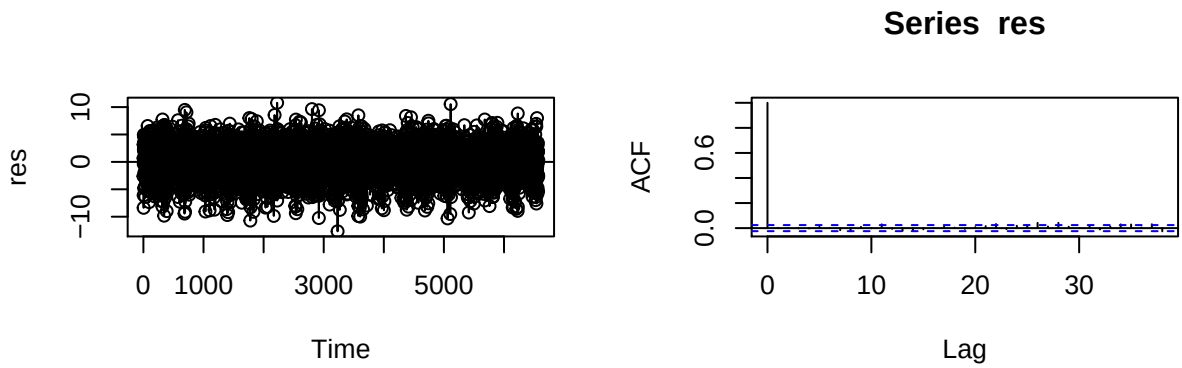
```
## get arima(2,0,2)
res.arma <- arima(reg5$residuals, order=c(2,0,2), method="ML")
resdiags(res.arma$residuals)
```

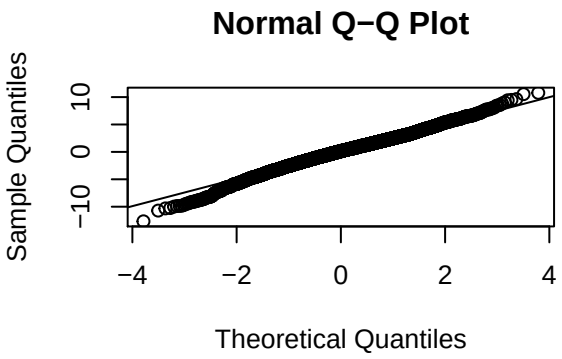


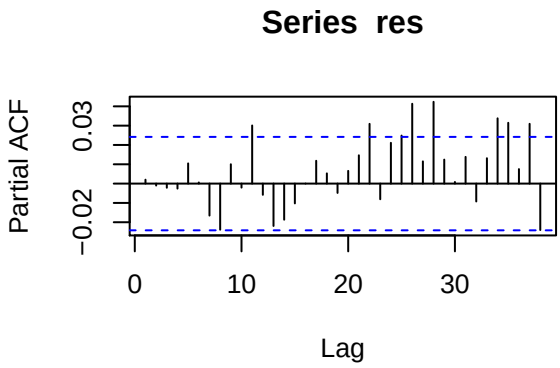

```
X <- model.matrix(reg5)[,2:12]
reg.arma <- arima(trt.train, order=c(2,0,2), xreg=X, method="ML")
```

```
## Warning in arima(trt.train, order = c(2, 0, 2), xreg = X, method = "ML"):
## possible convergence problem: optim gave code = 1
```

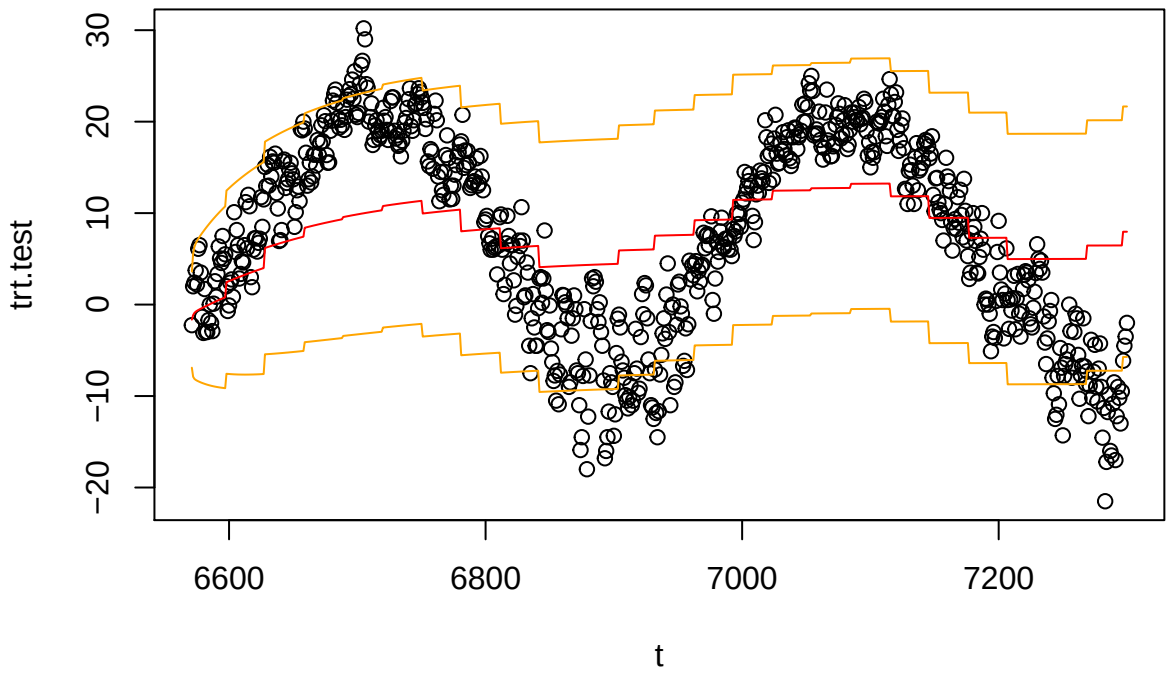
```
resdiags(reg.arma$residuals)
```








```
## prediction
newX <- model.matrix(reg5)[1:730,2:12]
newX[,1]=newX[,1]+6570
newXpred.arma <- predict(reg.arma, n.ahead=730, newxreg=newX, se.fit=TRUE)
par(mfcol=c(1,1))
t=c(6571:7300)
plot(t,trt.test)
points(trt.test)
points(newXpred.arma$pred, type="l", col="red")
points(newXpred.arma$pred + 1.96*newXpred.arma$se, type="l", col="orange")
points(newXpred.arma$pred - 1.96*newXpred.arma$se, type="l", col="orange")
```




```
PRESS.regar<-sum((trt.test-newXpred.arma$pred)^2)
PRESS.regar
```

```
## [1] 55668.46
```

```
AIC(reg.arma)
```

```
## [1] 31639.66
```

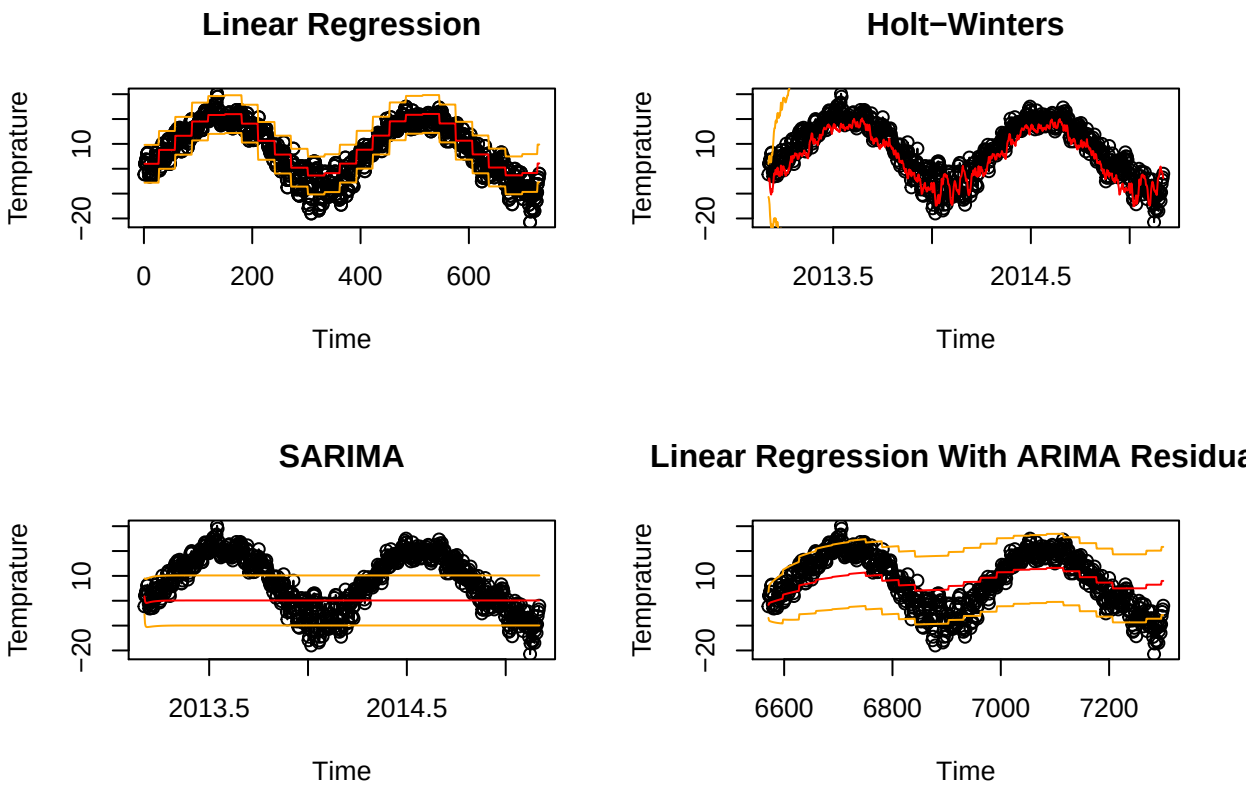
```
pre.result4<-(trt.test<newXpred.arma$pred + 1.96*newXpred.arma$se & trt.test>newXpred.arma$pred - 1.96*
table(pre.result4)
```

```
## pre.result4
## FALSE  TRUE
##    85   645
```

As above four types of model, the combination of ARIMA and linear model is selected for forecast.

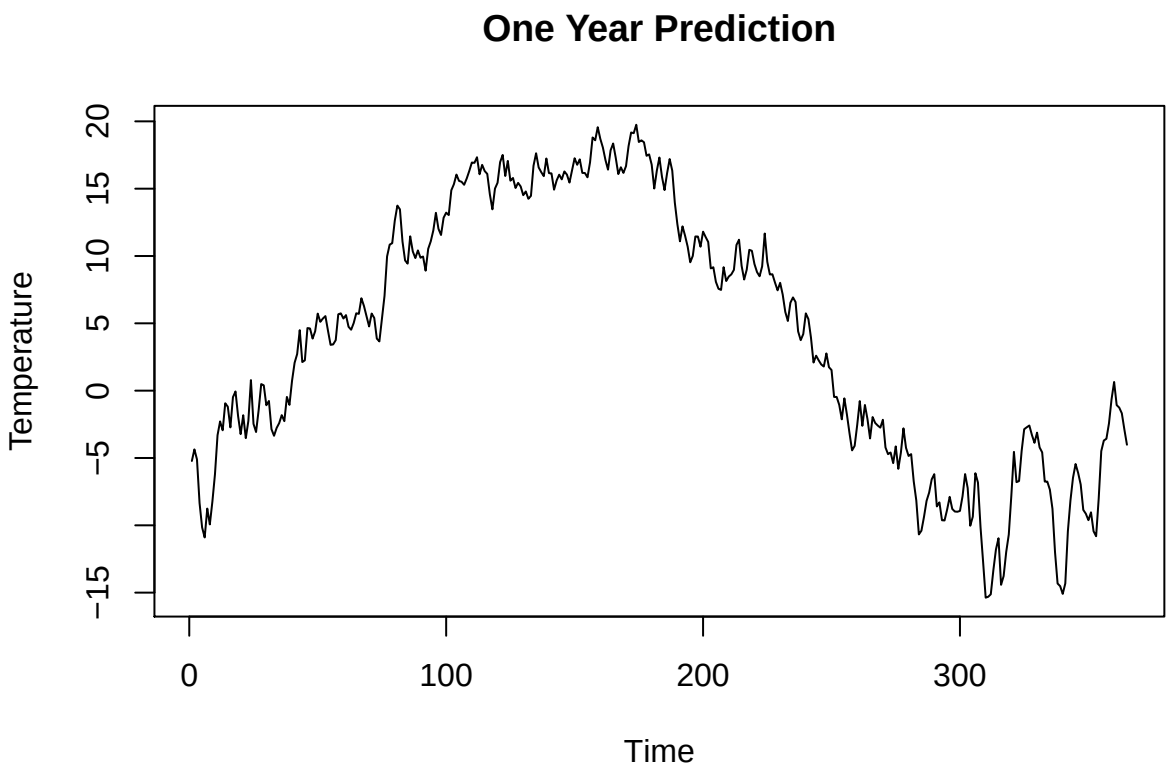
```
par(mfcol=c(2,2))

plot(trt.test,main="Linear Regression",xlab="Time",ylab="Temprature")
points(trt.test)
points(reg5.pre[,"fit"], type="l", col="red")
```

```
points(reg5.pre[,"lwr"], type="l", col="orange")
points(reg5.pre[,"upr"], type="l", col="orange")

plot(trt.testts,main="SARIMA",xlab="Time",ylab="Temprature")
points(trt.testts)
points(pred.mod4$pred, type="l", col="red")
points(pred.mod4$pred + 1.96*pred.mod4$se, type="l", col="orange")
points(pred.mod4$pred - 1.96*pred.mod4$se, type="l", col="orange")

plot(trt.testts,main="Holt-Winters",xlab="Time",ylab="Temprature")
points(trt.testts)
points(PI.hw[,1], type="l", col="red")
points(PI.hw[,2], type="l", col="orange")
points(PI.hw[,3], type="l", col="orange")

plot(t,trt.test,main="Linear Regression With ARIMA Residuals",xlab="Time",ylab="Temprature")
points(trt.test)
points(newXpred.arma$pred, type="l", col="red")
points(newXpred.arma$pred + 1.96*newXpred.arma$se, type="l", col="orange")
points(newXpred.arma$pred - 1.96*newXpred.arma$se, type="l", col="orange")
```



We select Holt-Winters model for next one year prediction.

```
oneyrpre<-predict(trt.hw,1095,prediction.interval=TRUE)
par(mfcol=c(1,1))
ts.plot(oneyrpre[,1][731:1095],main="One Year Prediction",ylab="Temperature")
```




```
pre.data<-oneyrpre[,1][731:1095]
```