TSProject_Kai

Kai Li

August 19, 2019

Load required libraries

```
# load libraries
library(caret)
library(TSA)
library(pls)
library(forecast)
library(tseries)
library(vars)
library(MASS)
library(fpp2)
```

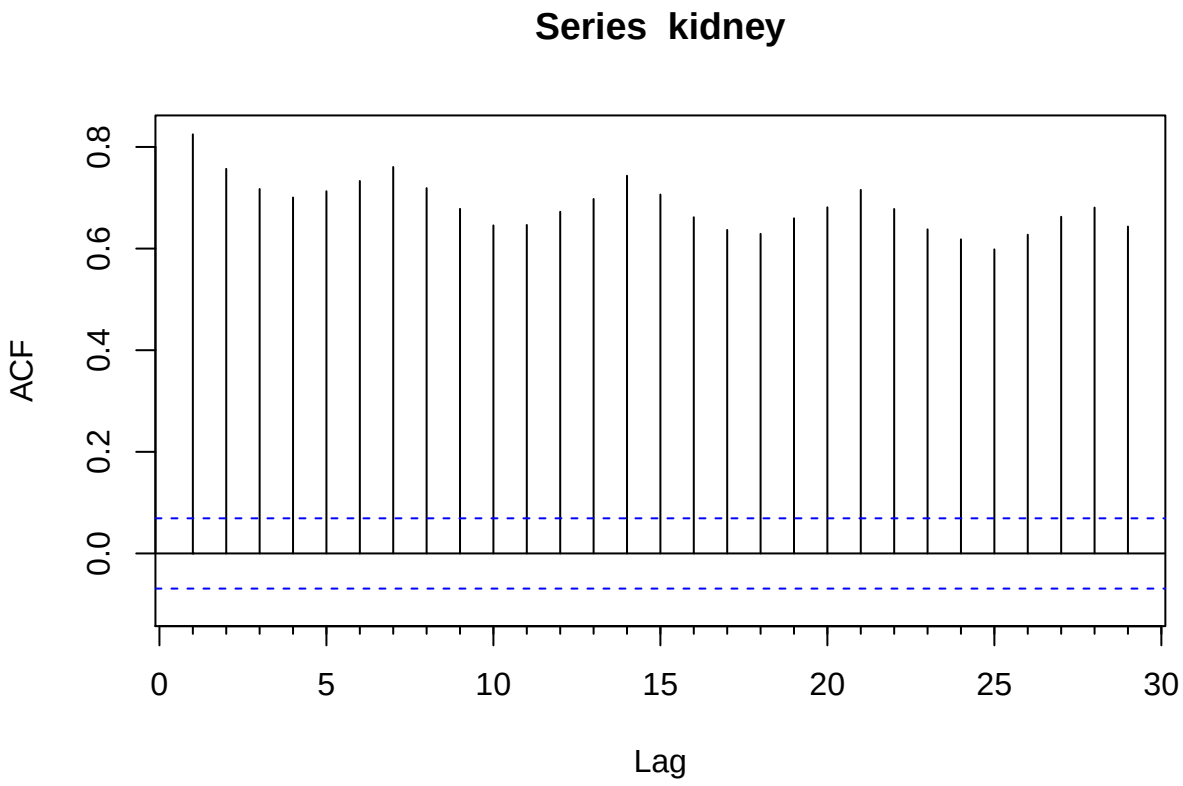
Read data from the data selected from META data file

```
df = readRDS("~/GitHub/TimeSeries-Project/TSProject/new_train.rds")

df1 = df[,10:12]
append = as.numeric(df1[,1])
kidney = as.numeric(df1[,3])
```

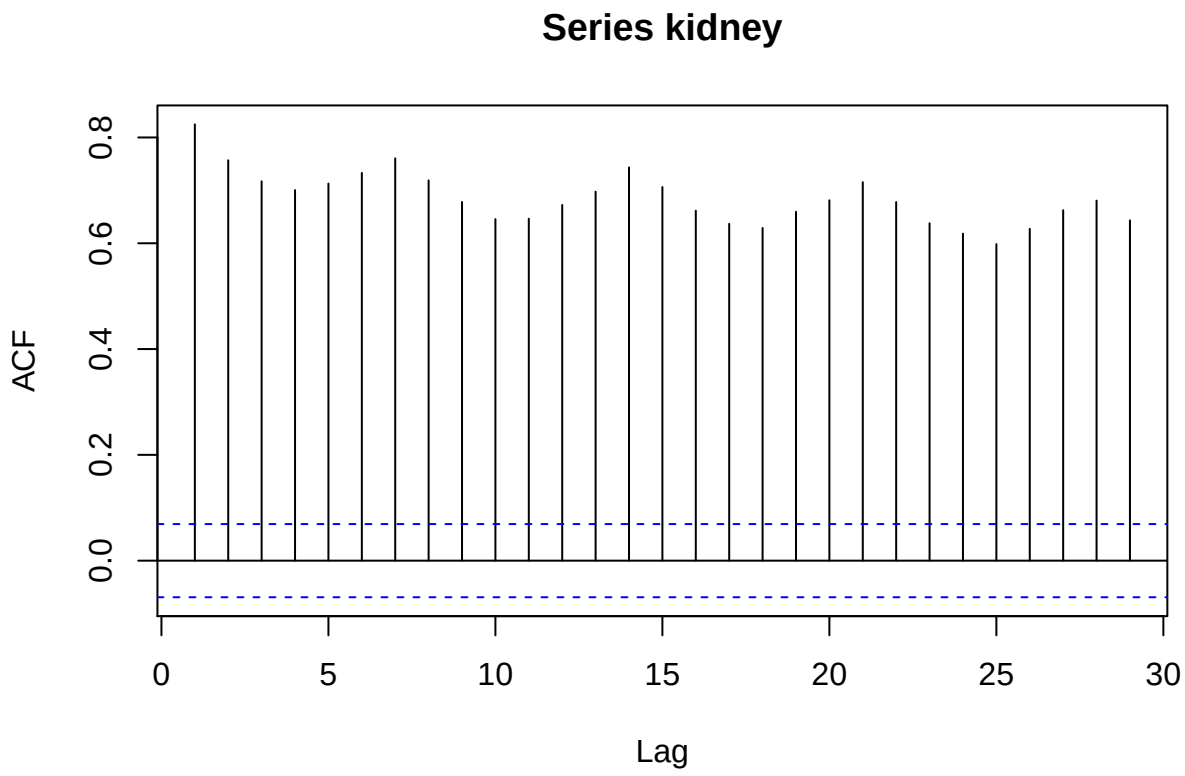
For this data set, the main target is the Kidney Stone Disease and want to see if it is possible to improve prediction by including anther disease which is appendicitis in this case. These two data set have a very similar trend in general from the data selection process.

Time Series plot for the Kidney Stone

```
Acf(kidney)
```




```
acf(kidney)
```



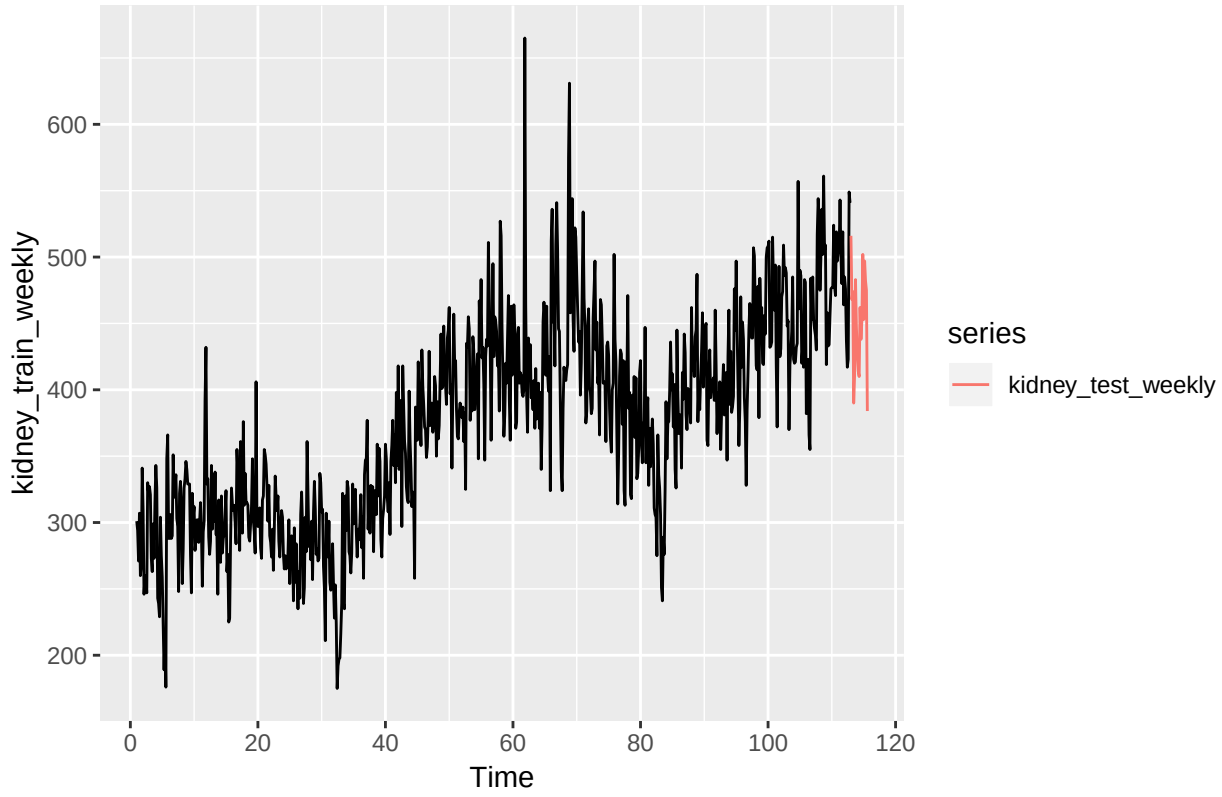
From the time series plot, we can see the magnitude of the data is various along with time. Also, the ACF plot shows wave in every 7 spikes. Therefore, I choose the frequency 7 in the following analysis.

Split train and test data set

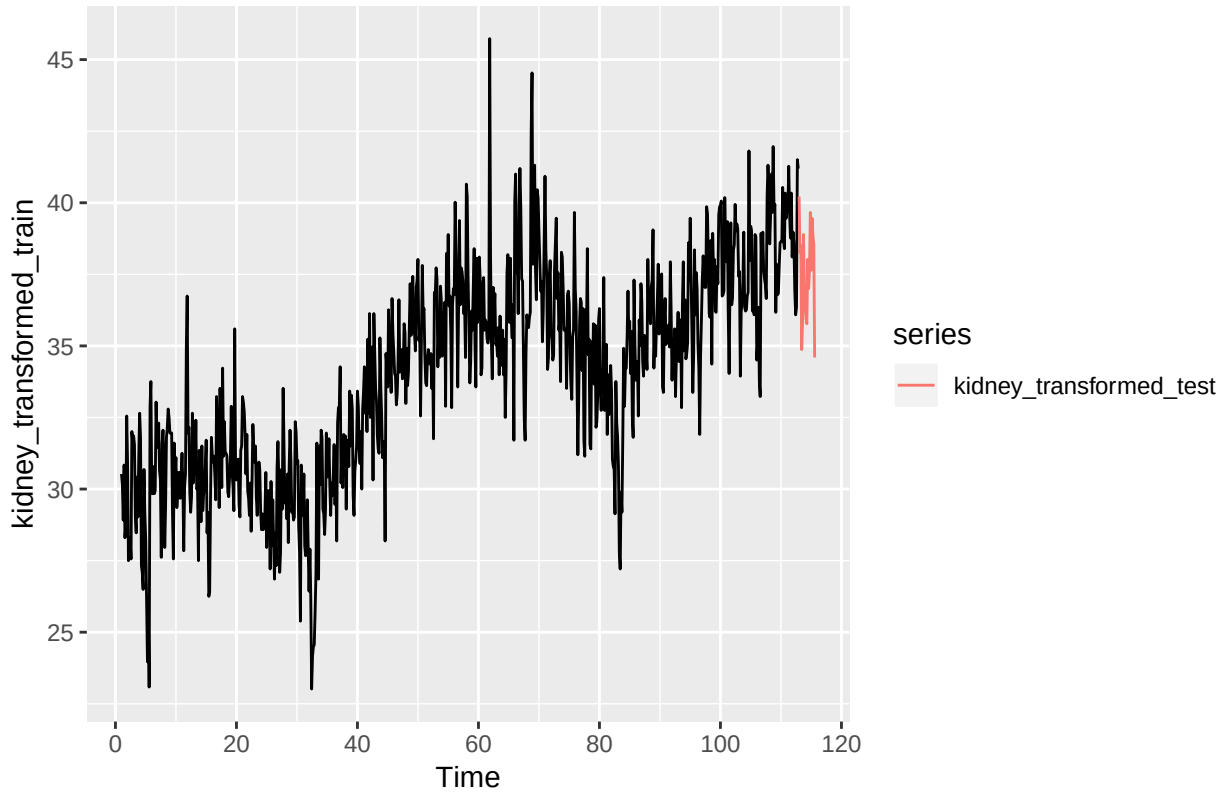
```
## Use the last 19 data points as the test data set

append_weekly = ts(append, frequency = 7)
kidney_weekly = ts(kidney, frequency = 7)


append_train_weekly = window(append_weekly, start = c(1,1), end = c(112,7))
append_test_weekly  = window(append_weekly, start = c(113,1), end = c(115,5))


kidney_train_weekly = window(kidney_weekly, start = c(1,1), end = c(112,7))
kidney_test_weekly  = window(kidney_weekly, start = c(113,1), end = c(115,5))


kidney_diff_train = window(diff(kidney_weekly), start = c(1,2), end = c(112,7))
kidney_diff_test = window(diff(kidney_weekly), start = c(113,1), end = c(115,5))


append_diff_train = window(diff(append_weekly), start = c(1,2), end = c(112,7))
append_diff_test = window(diff(append_weekly), start = c(113,1), end = c(115,5))
```

BoxCox Transformation

```
##
append_lambda = BoxCox.lambda(append_train_weekly)
append_transformed_train = BoxCox(append_train_weekly, lambda = append_lambda)
append_transformed_test = BoxCox(append_test_weekly, lambda = append_lambda)

kidney_lambda = BoxCox.lambda(kidney_train_weekly)
kidney_transformed_train = BoxCox(kidney_train_weekly, lambda = kidney_lambda)

autoplot(kidney_train_weekly)+
  autolayer(kidney_test_weekly)
```




```
kidney_transformed_test = BoxCox(kidney_test_weekly, lambda = kidney_lambda)

autoplot(kidney_transformed_train)+
  autolayer(kidney_transformed_test)
```



Stationary Test for Appendicitis

```
adf.test(append_transformed_train) # p-value need to be less than 0.05
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  append_transformed_train
## Dickey-Fuller = -3.6756, Lag order = 9, p-value = 0.02541
## alternative hypothesis: stationary
```

```
kpss.test(append_transformed_train) # p-value need to be greater than 0.05
```

```
## Warning in kpss.test(append_transformed_train): p-value smaller than
## printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  append_transformed_train
## KPSS Level = 6.8808, Truncation lag parameter = 6, p-value = 0.01
```

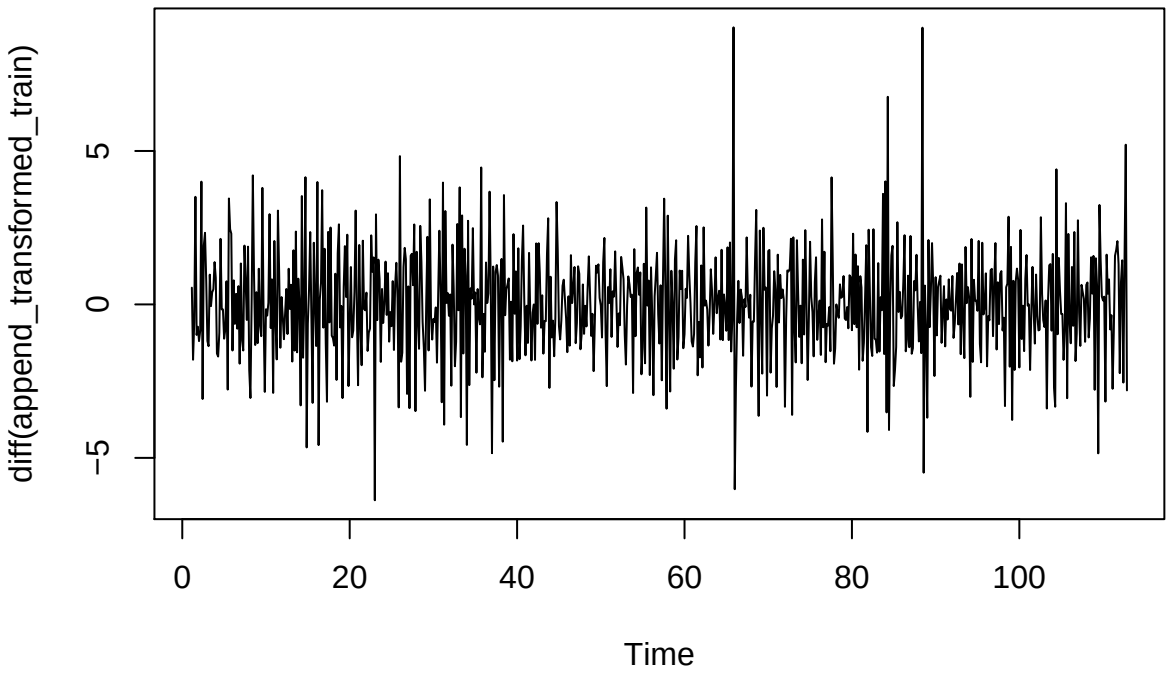
```r
adf.test(diff(append_transformed_train)) # p-value need to be less than 0.05
```

```
## Warning in adf.test(diff(append_transformed_train)): p-value smaller than
## printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  diff(append_transformed_train)
## Dickey-Fuller = -11.839, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(diff(append_transformed_train)) # p-value need to be greater than 0.05
```

```
## Warning in kpss.test(diff(append_transformed_train)): p-value greater than
## printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  diff(append_transformed_train)
## KPSS Level = 0.03249, Truncation lag parameter = 6, p-value = 0.1
```

```r
plot(diff(append_transformed_train))
```



```r
adf.test(diff(append_train_weekly)) # p-value need to be less than 0.05
```

```
## Warning in adf.test(diff(append_train_weekly)): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(append_train_weekly)
## Dickey-Fuller = -11.905, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(diff(append_train_weekly)) # p-value need to be greater than 0.05
```

```
## Warning in kpss.test(diff(append_train_weekly)): p-value greater than
## printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(append_train_weekly)
## KPSS Level = 0.024293, Truncation lag parameter = 6, p-value = 0.1
```

Stationary Test for Kidney Stone

```r
adf.test(kidney_transformed_train)  # p-value need to be less than 0.05
```

```
## Warning in adf.test(kidney_transformed_train): p-value smaller than printed
## p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  kidney_transformed_train
## Dickey-Fuller = -4.4762, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(kidney_transformed_train) # p-value need to be greater than 0.05y
```

```
## Warning in kpss.test(kidney_transformed_train): p-value smaller than
## printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  kidney_transformed_train
## KPSS Level = 7.8046, Truncation lag parameter = 6, p-value = 0.01
```

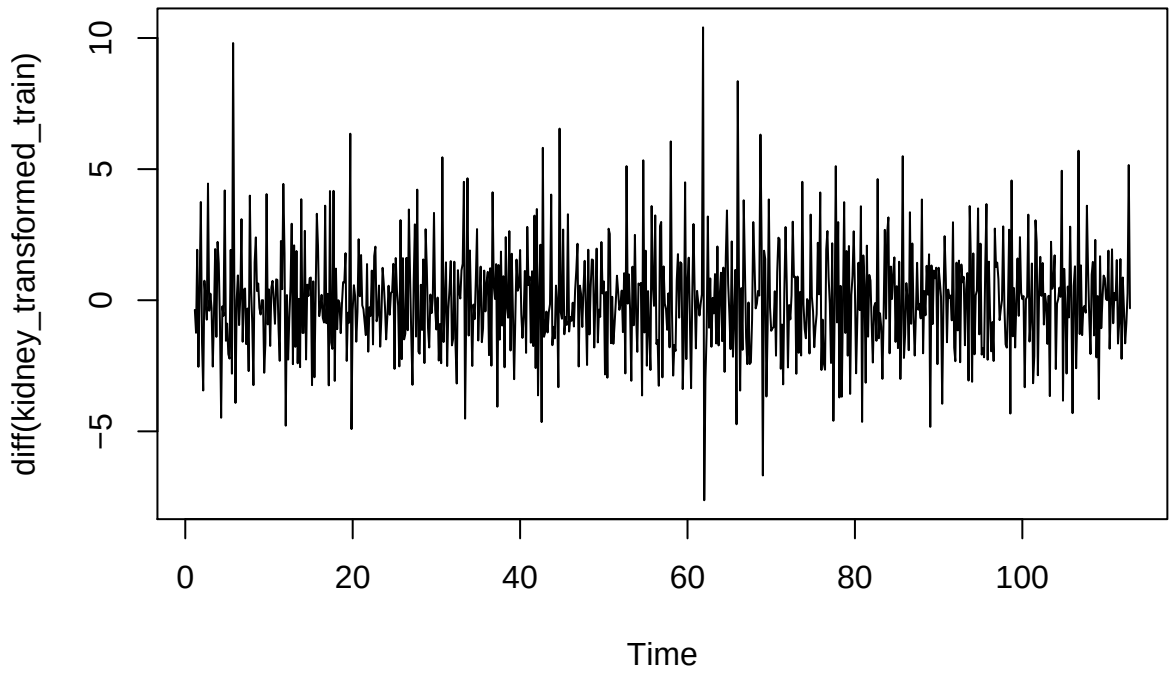
```r
adf.test(diff(diff(kidney_transformed_train)))   # p-value need to be less than 0.05
```

```
## Warning in adf.test(diff(diff(kidney_transformed_train))): p-value smaller
## than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(kidney_transformed_train))
## Dickey-Fuller = -16.924, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(diff(diff(kidney_transformed_train)))  # p-value need to be greater than 0.05
```

```
## Warning in kpss.test(diff(diff(kidney_transformed_train))): p-value greater
## than printed p-value

##
##  KPSS Test for Level Stationarity
##
## data:  diff(diff(kidney_transformed_train))
## KPSS Level = 0.0058048, Truncation lag parameter = 6, p-value =
## 0.1
```

```r
plot(diff(kidney_transformed_train))
```



```r
adf.test(diff(kidney_train_weekly))  # p-value need to be less than 0.05
```

```
## Warning in adf.test(diff(kidney_train_weekly)): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(kidney_train_weekly)
## Dickey-Fuller = -11.637, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

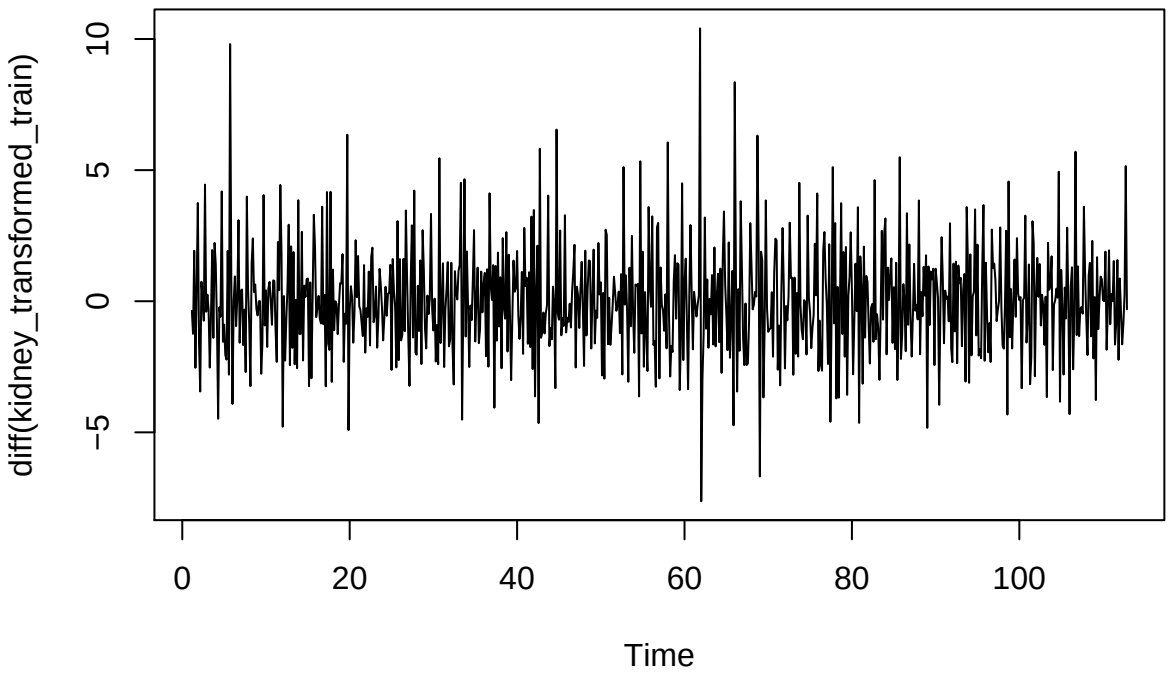
```r
kpss.test(diff(kidney_train_weekly)) # p-value need to be greater than 0.05
```

```
## Warning in kpss.test(diff(kidney_train_weekly)): p-value greater than
## printed p-value
```

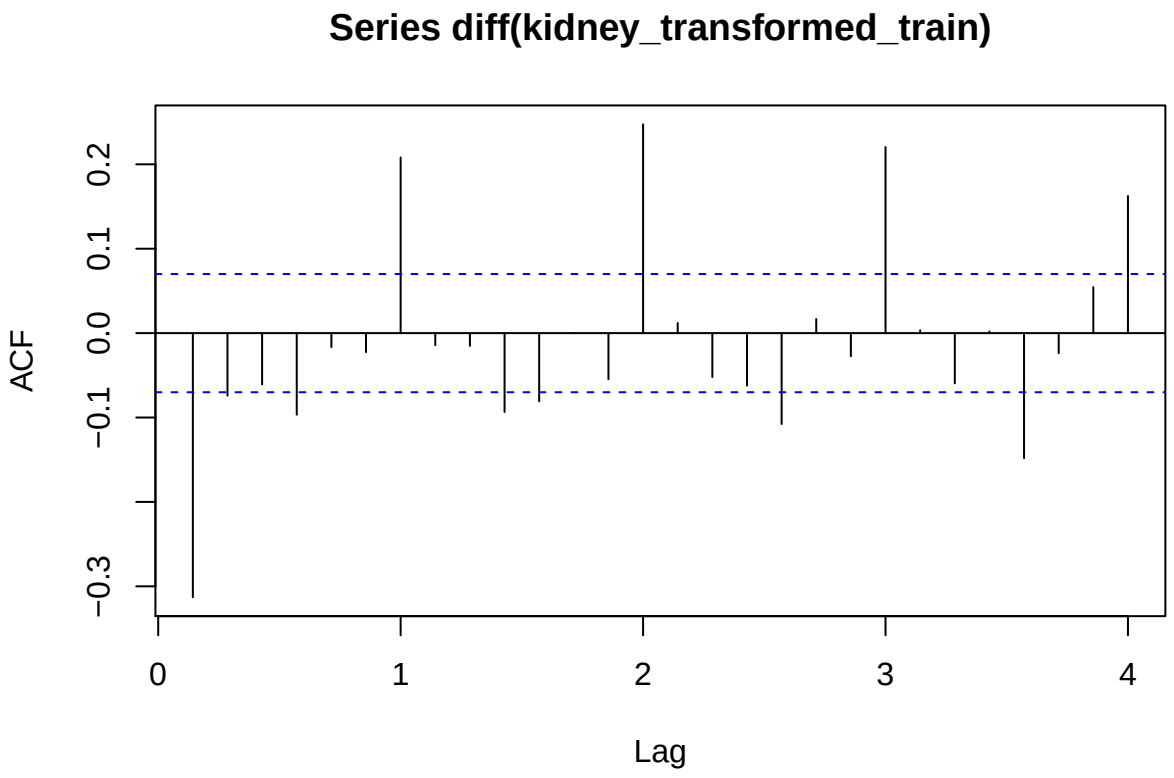
```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(kidney_train_weekly)
## KPSS Level = 0.016881, Truncation lag parameter = 6, p-value = 0.1
```

TS plot, ACF plot and pacf plot

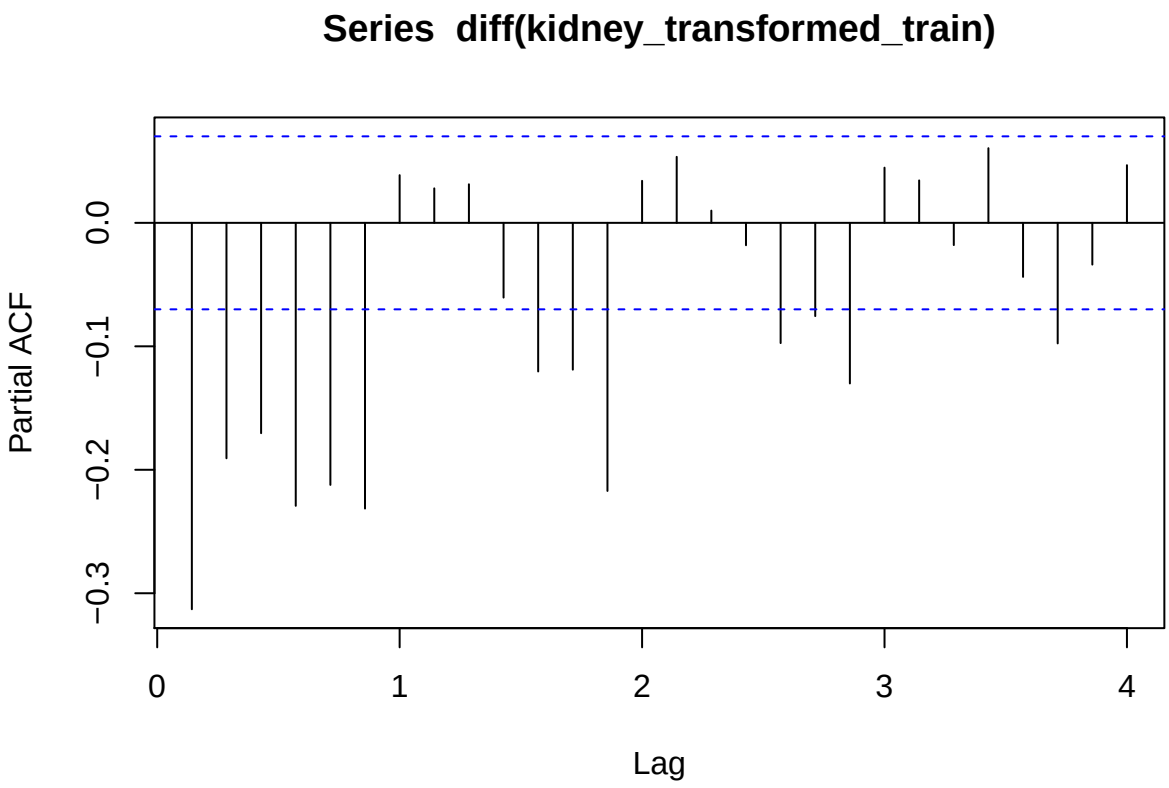
```r
ts.plot(diff(kidney_transformed_train))
```




```
acf(diff(kidney_transformed_train))
```




```
pacf(diff(kidney_transformed_train))
```



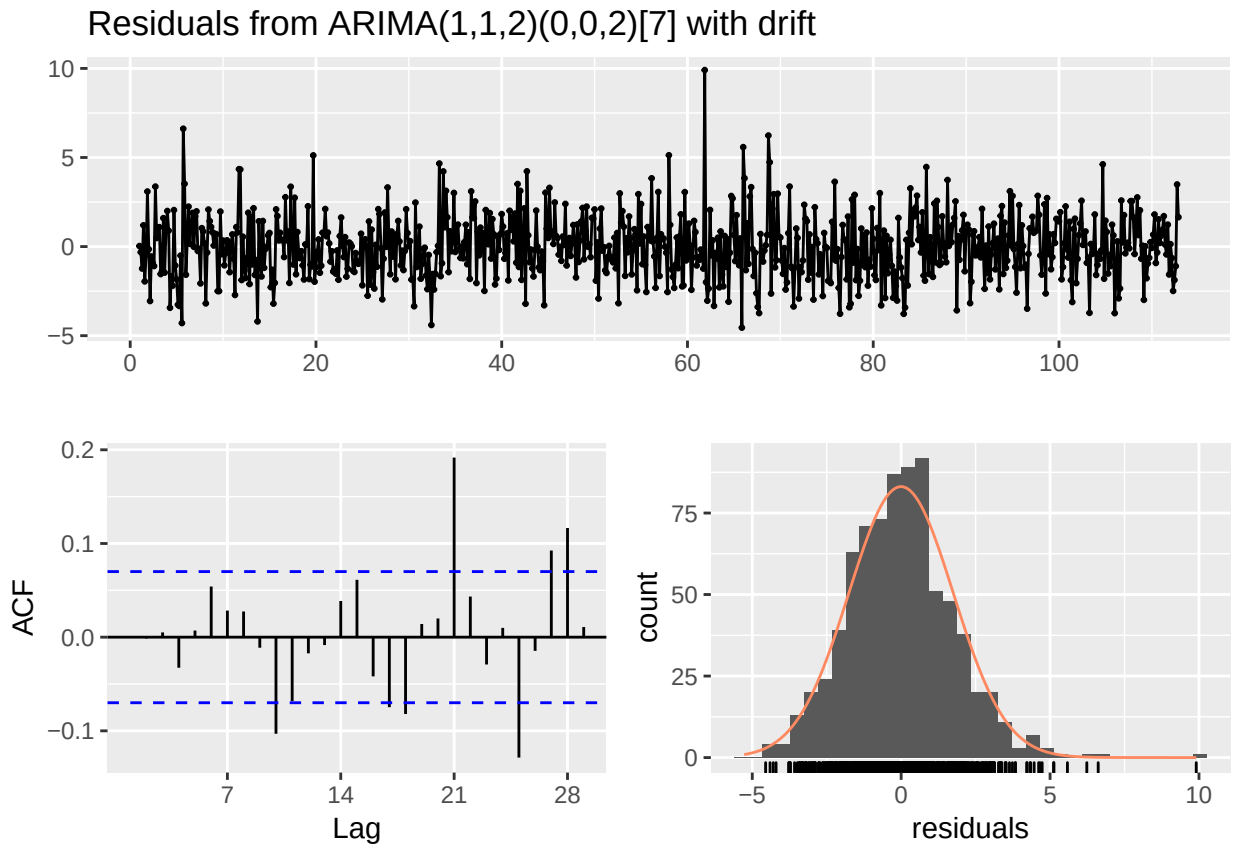
Model Selection

Auto Arima model

```
# After 1st differencing, the data set is stationary. Thus, set d = 1, and there is strong seasonality

arima_atuo_kidney = auto.arima(kidney_transformed_train, d = 1, seasonal = T)
arima_atuo_kidney
```

```
## Series: kidney_transformed_train
## ARIMA(1,1,2)(0,0,2)[7] with drift
##
## Coefficients:
##          ar1      ma1     ma2   sma1    sma2   drift
##       0.5558  -1.1193  0.1599  0.153  0.1829  0.0121
## s.e.  0.0837   0.0959  0.0865  0.037  0.0343  0.0078
##
## sigma^2 estimated as 3.106:  log likelihood=-1552.57
## AIC=3119.13   AICc=3119.28   BIC=3151.77
```

```
# Ljung-Box test need to be greater than 0.05
checkresiduals(arima_atuo_kidney)
```



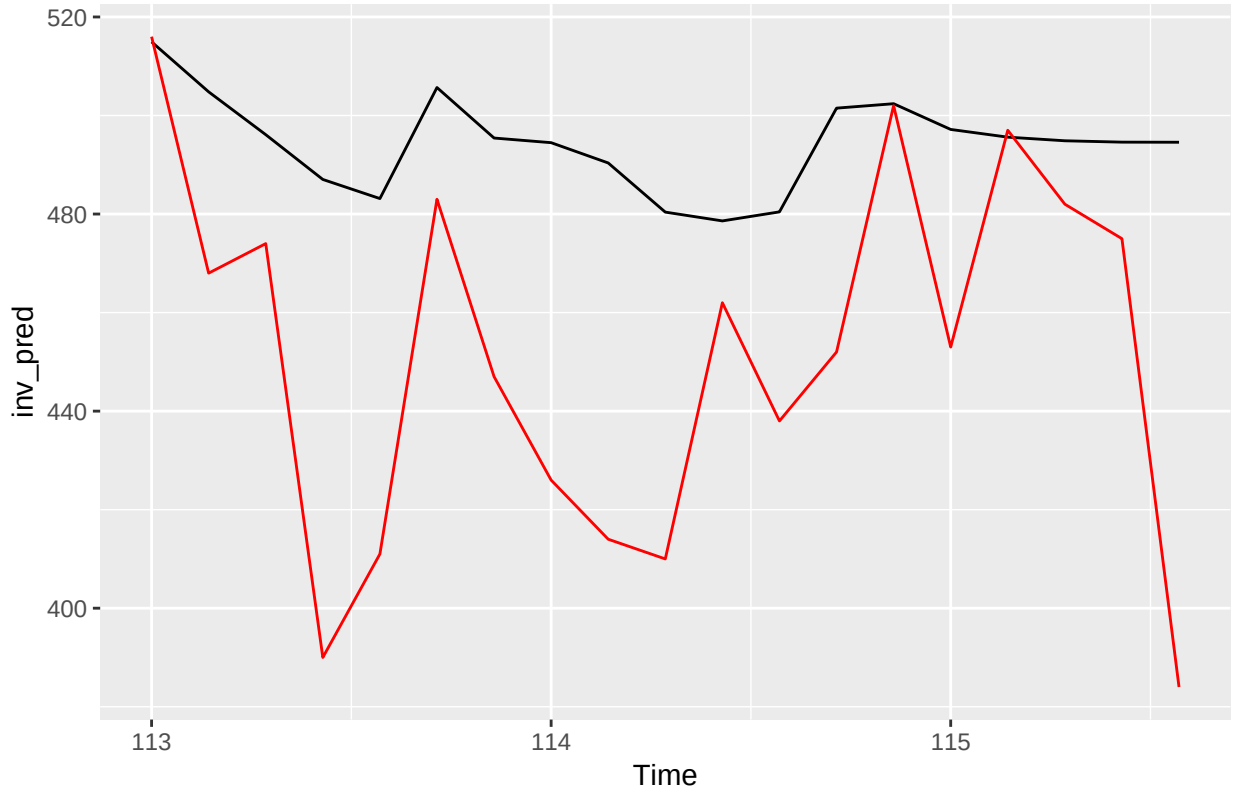

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,2)(0,0,2)[7] with drift
## Q* = 18.212, df = 8, p-value = 0.01969
##
## Model df: 6.   Total lags used: 14
```

```
pred = forecast(arima_atuo_kidney, h = 19)
inv_pred = InvBoxCox(pred$mean, lambda = kidney_lambda)
acc_atuo = accuracy(inv_pred, kidney_test_weekly)
acc_atuo
```

```
##                 ME     RMSE      MAE       MPE     MAPE       ACF1
## Test set -42.53157 53.33778 42.78991 -10.0707 10.12185 0.07131722
##          Theil's U
## Test set  1.251208
```

```
autoplot(inv_pred, include = 100)+
  autolayer(kidney_test_weekly,series = "Auto_ARIMA", color = "red")
```

```
## Warning: Ignoring unknown parameters: include
```



Try Self-defined SARIMA model

```
eacf(kidney_transformed_train)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x o o x o o x o o x x  o  o  x
## 2 x x o x o o x x o o x  o  o  x
## 3 x x o x o o x o o o x  o  o  x
## 4 x x x o o o x x o o o  o  o  x
## 5 x x x x o x x x o o o  o  o  x
## 6 x o x x x x o o o o o  o  x  x
## 7 x x x x x x o x o x o  o  o  o
```

Try different combinations of AR and MA, also for the seasonal part.

```
arima_kidney = Arima(kidney_transformed_train, order = c(1,1,2), seasonal = c(1, 0, 2))
arima_kidney
```

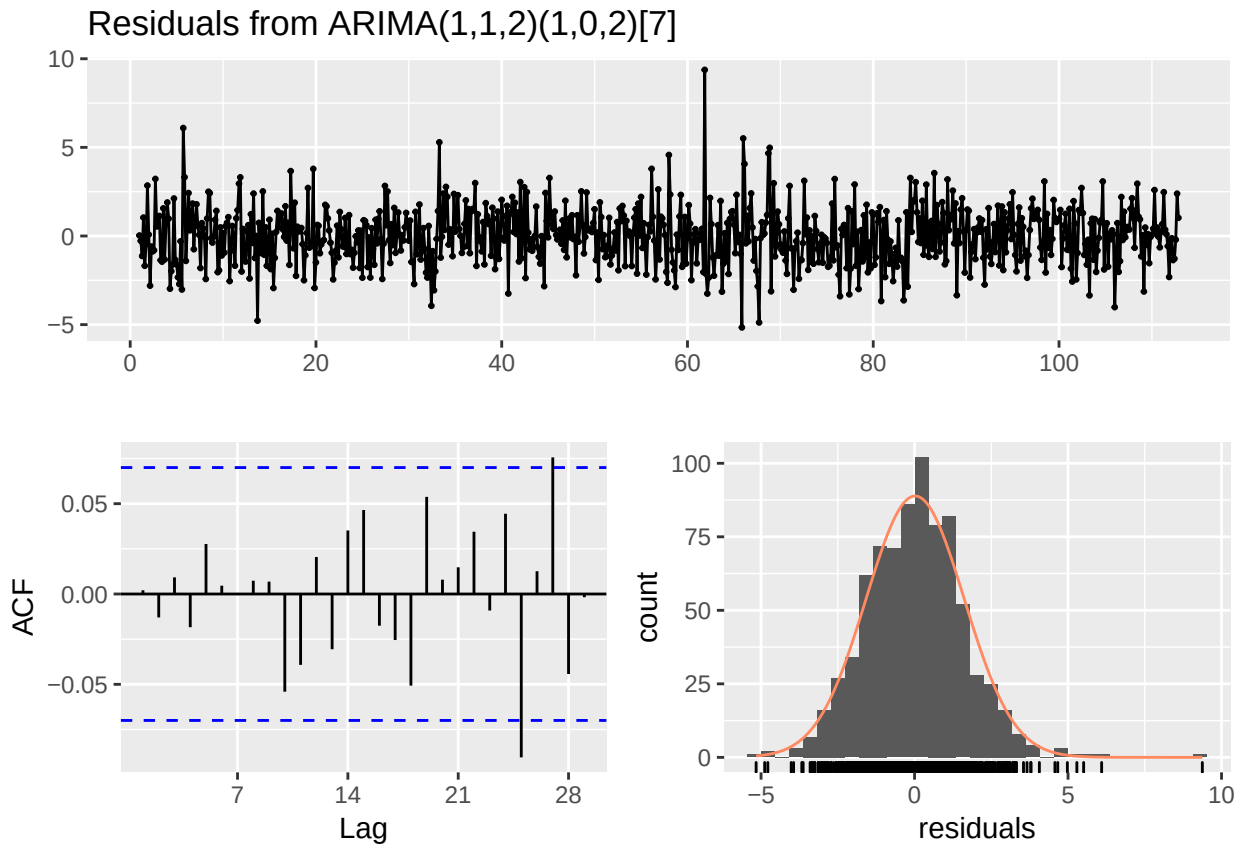
```
## Series: kidney_transformed_train
## ARIMA(1,1,2)(1,0,2)[7]
##
```

```
## Coefficients:
##          ar1      ma1     ma2    sar1     sma1    sma2
##       0.6717  -1.2592  0.3070  0.9999  -1.0045  0.0127
## s.e.  0.0900   0.1052  0.0896  0.0002   0.0365  0.0360
##
## sigma^2 estimated as 2.576:  log likelihood=-1488.91
## AIC=2991.82   AICc=2991.96   BIC=3024.46
```

```
arima_kidney$coef
```

```
##        ar1        ma1        ma2       sar1       sma1       sma2
##  0.6717310 -1.2591675  0.3070262  0.9998947 -1.0045181  0.0127447
```

```
# Ljung-Box test need to be greater than 0.05
checkresiduals(arima_kidney)
```



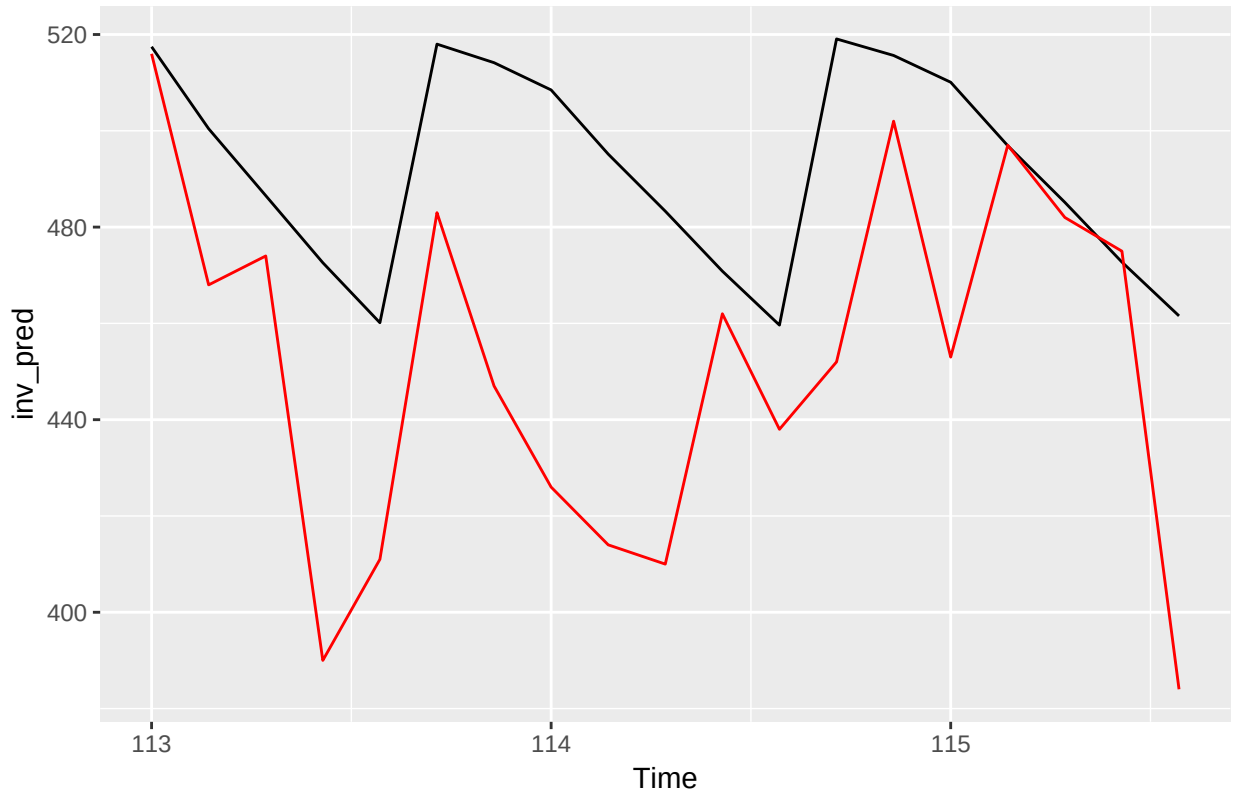

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,2)(1,0,2)[7]
## Q* = 6.8031, df = 8, p-value = 0.558
##
## Model df: 6.   Total lags used: 14
```

```
pred = forecast(arima_kidney, h = 19)
inv_pred = InvBoxCox(pred$mean, lambda = kidney_lambda)
acc_arima = accuracy(inv_pred, kidney_test_weekly)
acc_arima
```

```
##                  ME     RMSE      MAE       MPE     MAPE      ACF1
## Test set -40.20631 51.02664 40.45492 -9.446347 9.498593 0.1304068
##          Theil's U
## Test set  1.198112
```

```
autoplot(inv_pred, include = 100)+
  autolayer(kidney_test_weekly,series = "SARIMA", color = "red")
```

```
## Warning: Ignoring unknown parameters: include
```



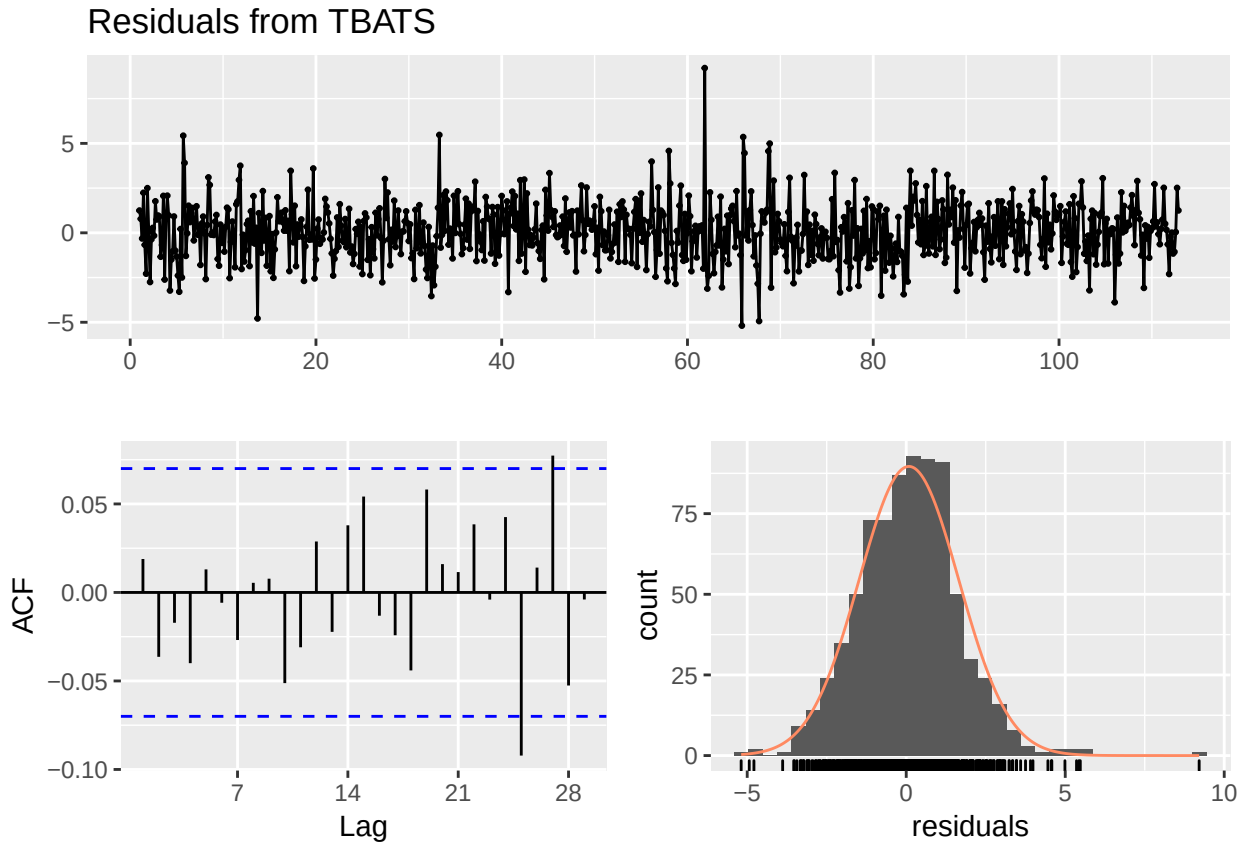
TBATS model

```
tbats_kidney = tbats(kidney_train_weekly)
tbats_kidney
```

```
## TBATS(0.481, {0,0}, 0.8, {<7,3>})
```

```
##
## Call: tbats(y = kidney_train_weekly)
##
## Parameters
##   Lambda: 0.480946
##   Alpha: 0.4605426
##   Beta: -0.07999926
##   Damping Parameter: 0.8
##   Gamma-1 Values: 0.0001873843
##   Gamma-2 Values: 0.0002385076
##
## Seed States:
##            [,1]
## [1,] 28.1540281
## [2,]  0.2504756
## [3,]  0.9413988
## [4,] -0.4615508
## [5,]  0.1948228
## [6,] -0.5084875
## [7,] -0.1570779
## [8,]  0.3630236
## attr(,"lambda")
## [1] 0.4809458
##
## Sigma: 1.571592
## AIC: 10764.55
```

```
checkresiduals(tbats_kidney)
```



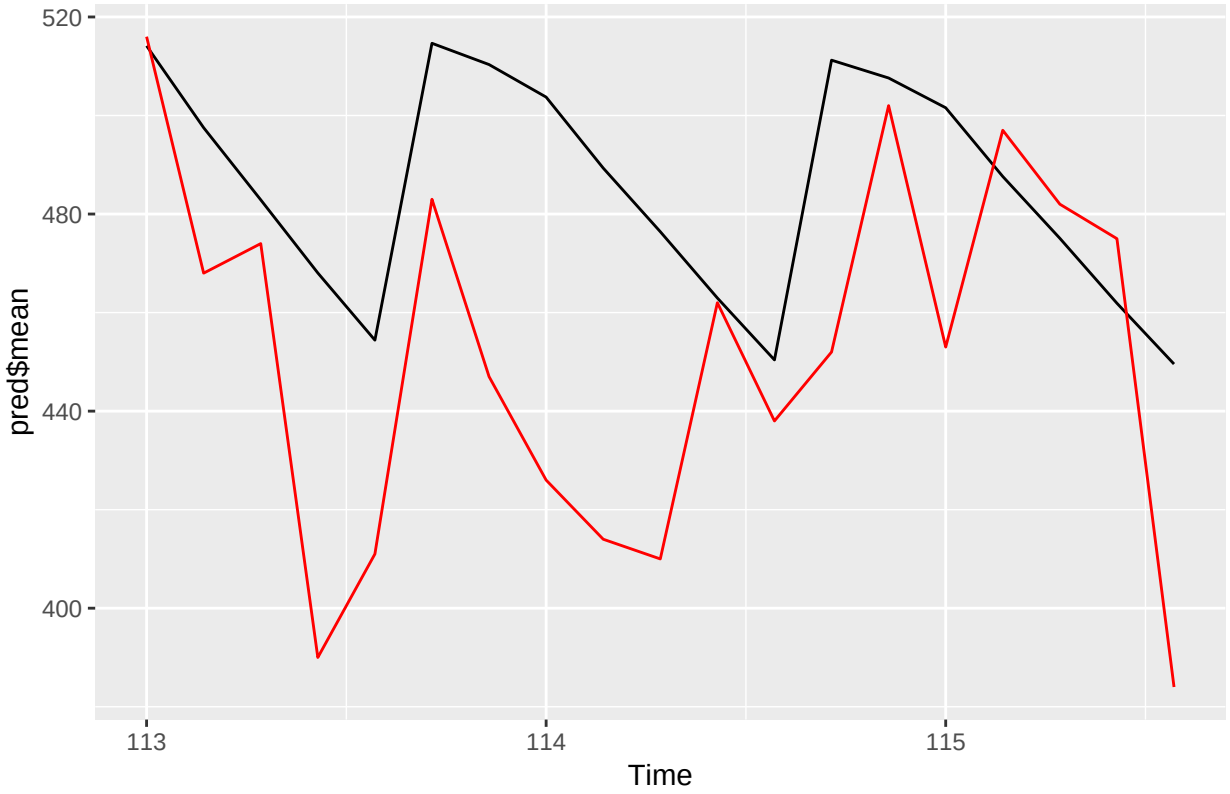

```
##
##  Ljung-Box test
##
## data:  Residuals from TBATS
## Q* = 11.645, df = 3, p-value = 0.008704
##
## Model df: 14.   Total lags used: 17
```

```
pred = forecast(tbats_kidney, h = 19)

acc_tbats = accuracy(pred, kidney_test_weekly)
acc_tbats
```

```
##                      ME     RMSE      MAE         MPE     MAPE      MASE
## Training set   2.525601 34.43044 26.27422 -0.09849957 7.158403 0.6676173
## Test set     -33.441559 46.30050 36.73305 -7.93327374 8.611517 0.9333718
##                    ACF1 Theil's U
## Training set 0.01528974        NA
## Test set     0.17807984  1.087163
```

```
autoplot(pred$mean, include = 100)+
  autolayer(kidney_test_weekly,series = "tbats", color = "red")
```

```
## Warning: Ignoring unknown parameters: include
```



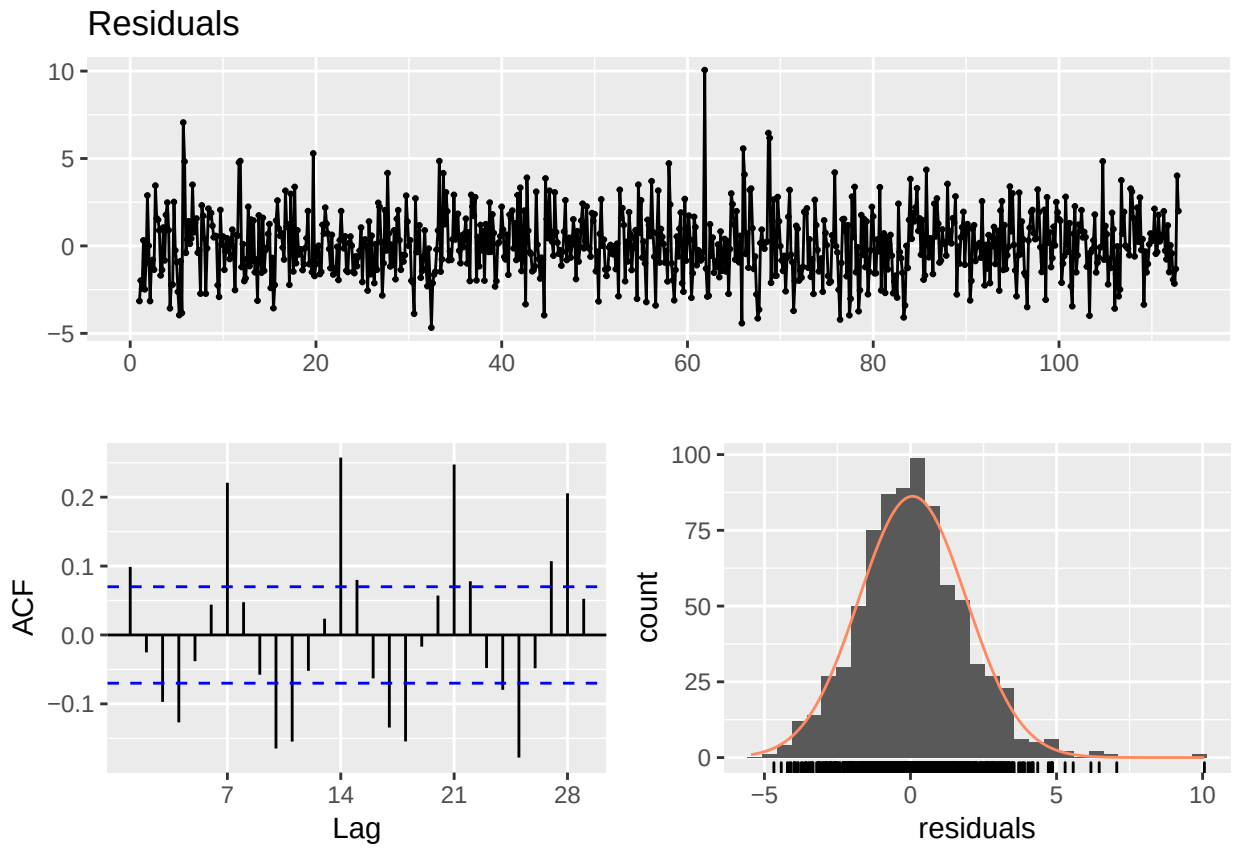
ARFIMA model

```
model_arfima = arfima(kidney_transformed_train)
summary(model_arfima)
```

```
##
## Call:
##   arfima(y = kidney_transformed_train)
##
## Coefficients:
##        Estimate Std. Error z value Pr(>|z|)
## d       0.32543    0.02080   15.65   <2e-16 ***
## ar.ar1  0.99578    0.02826   35.24   <2e-16 ***
## ma.ma1  0.95970    0.01089   88.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## sigma[eps] = 1.848145
## [d.tol = 0.0001221, M = 100, h = 1.681e-05]
## Log likelihood: -1594 ==> AIC = 3196.742 [4 deg.freedom]
```

```
checkresiduals(model_arfima)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```

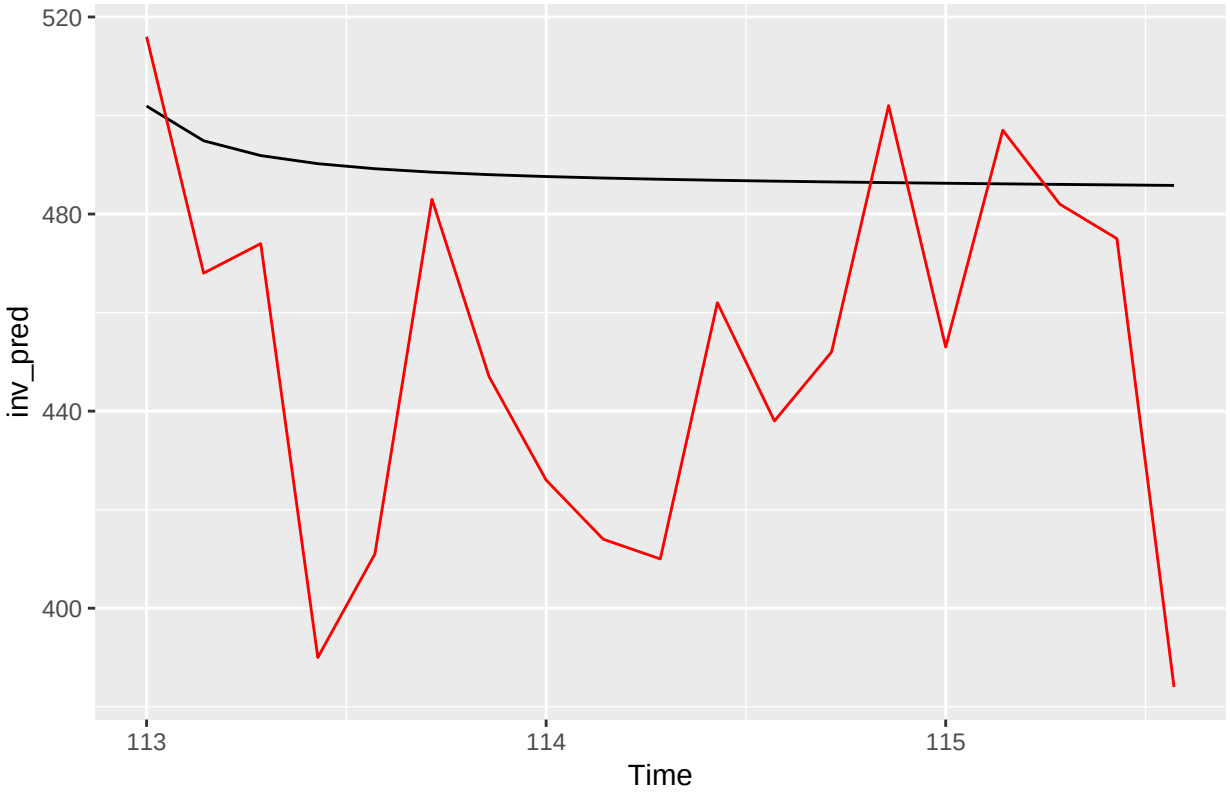



```
pred = forecast(model_arfima, h = 19)
inv_pred = InvBoxCox(pred$mean, lambda = kidney_lambda)

acc_arfima = accuracy(inv_pred, kidney_test_weekly)
acc_arfima
```

```
##                ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -36.7926 51.48498 41.06491 -8.873037 9.718373 0.1511233  1.216226
```

```
autoplot(inv_pred, include = 10)+
  autolayer(kidney_test_weekly,series = "ARFIMA", color = "red")
```

```
## Warning: Ignoring unknown parameters: include
```

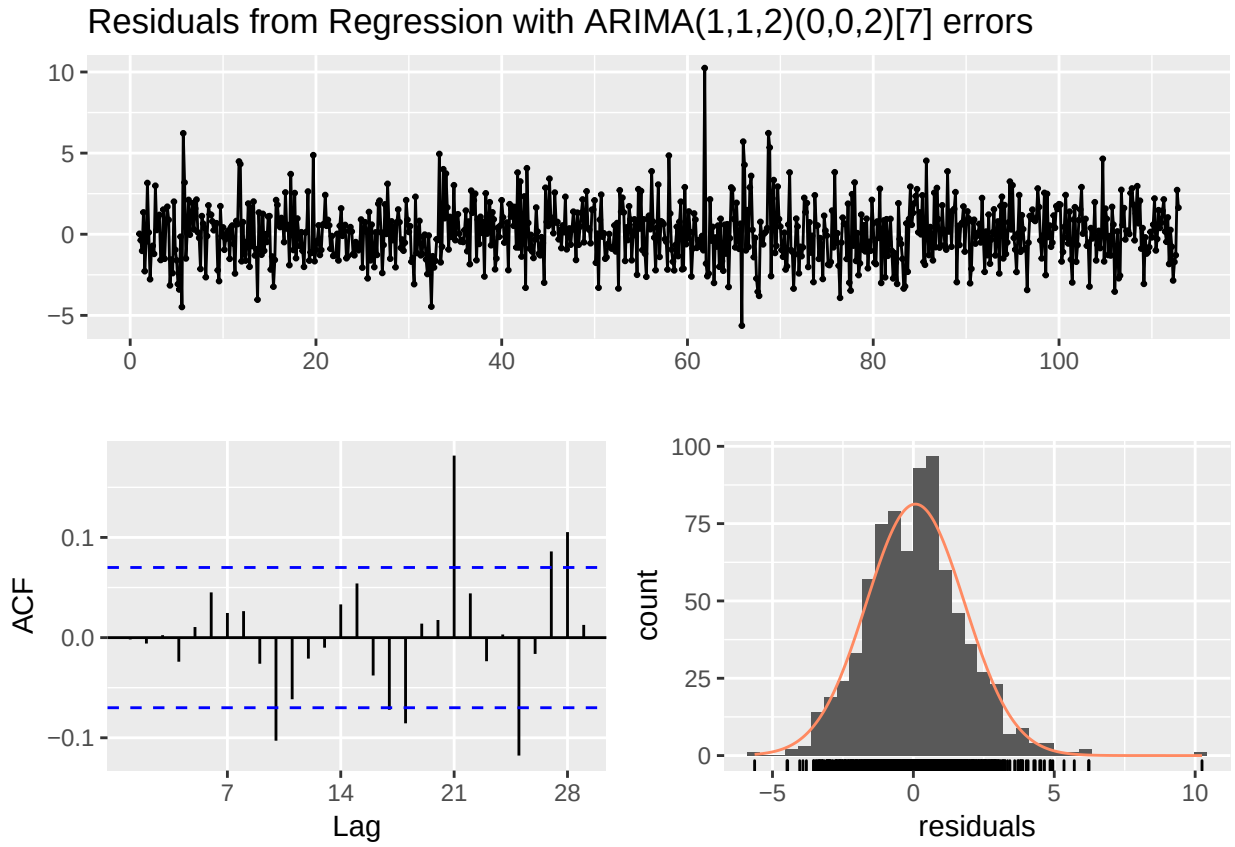


Regression model with Time Series Residuals

```
model_xreg = auto.arima(kidney_transformed_train, xreg = append_transformed_train  ,seasonal = TRUE)
model_xreg
```

```
## Series: kidney_transformed_train
## Regression with ARIMA(1,1,2)(0,0,2)[7] errors
##
## Coefficients:
##          ar1      ma1     ma2    sma1    sma2    xreg
##       0.5202  -1.1029  0.1503  0.1448  0.1793  0.1368
## s.e.  0.0957   0.1070  0.0957  0.0372  0.0345  0.0424
##
## sigma^2 estimated as 3.074:  log likelihood=-1548.47
## AIC=3110.93   AICc=3111.08   BIC=3143.57
```

```
checkresiduals(model_xreg)
```



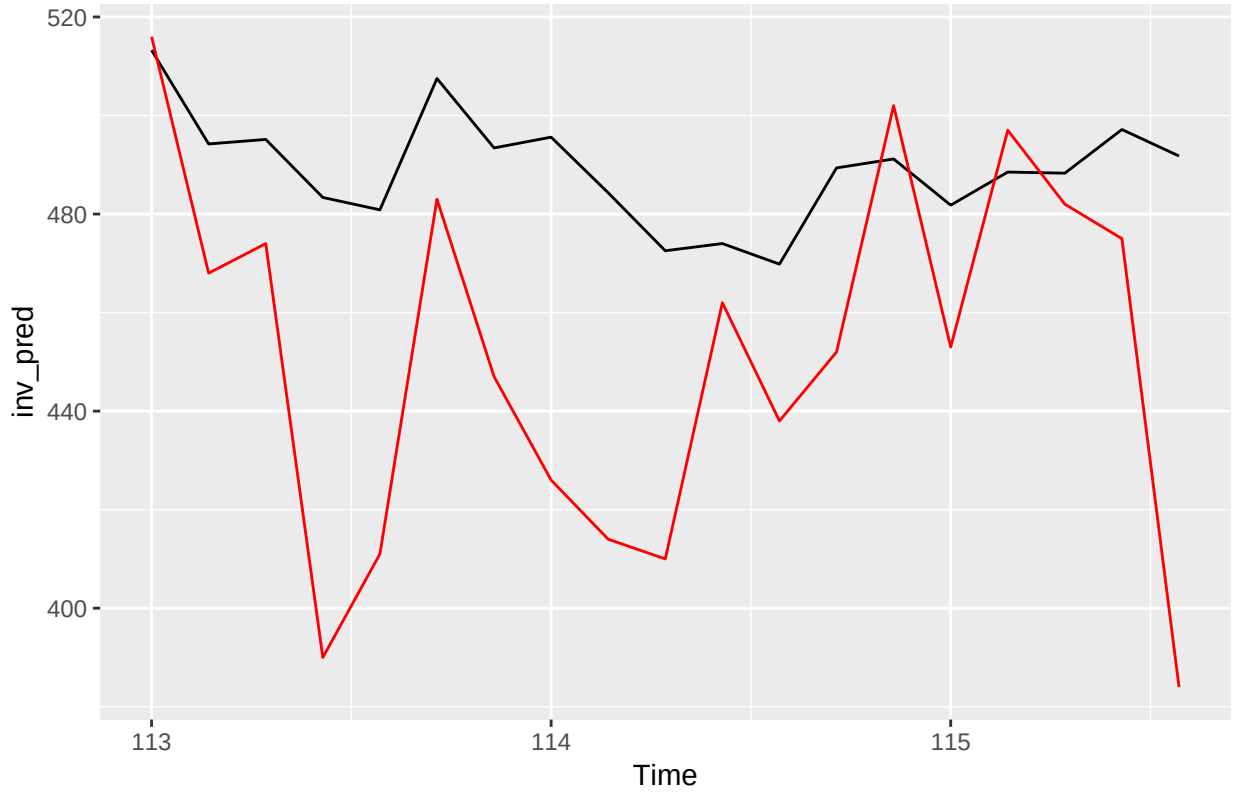

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,1,2)(0,0,2)[7] errors
## Q* = 16.525, df = 8, p-value = 0.03545
##
## Model df: 6.   Total lags used: 14
```

```
pred = forecast(model_xreg, xreg = append_transformed_test, h = 19)
inv_pred = InvBoxCox(pred$mean, lambda = kidney_lambda)


acc_xreg = accuracy(inv_pred, kidney_test_weekly)
acc_xreg
```

```
##                ME     RMSE     MAE       MPE     MAPE      ACF1 Theil's U
## Test set -37.2725 49.69068 39.5923 -8.903901 9.366244 0.2104222  1.167475
```

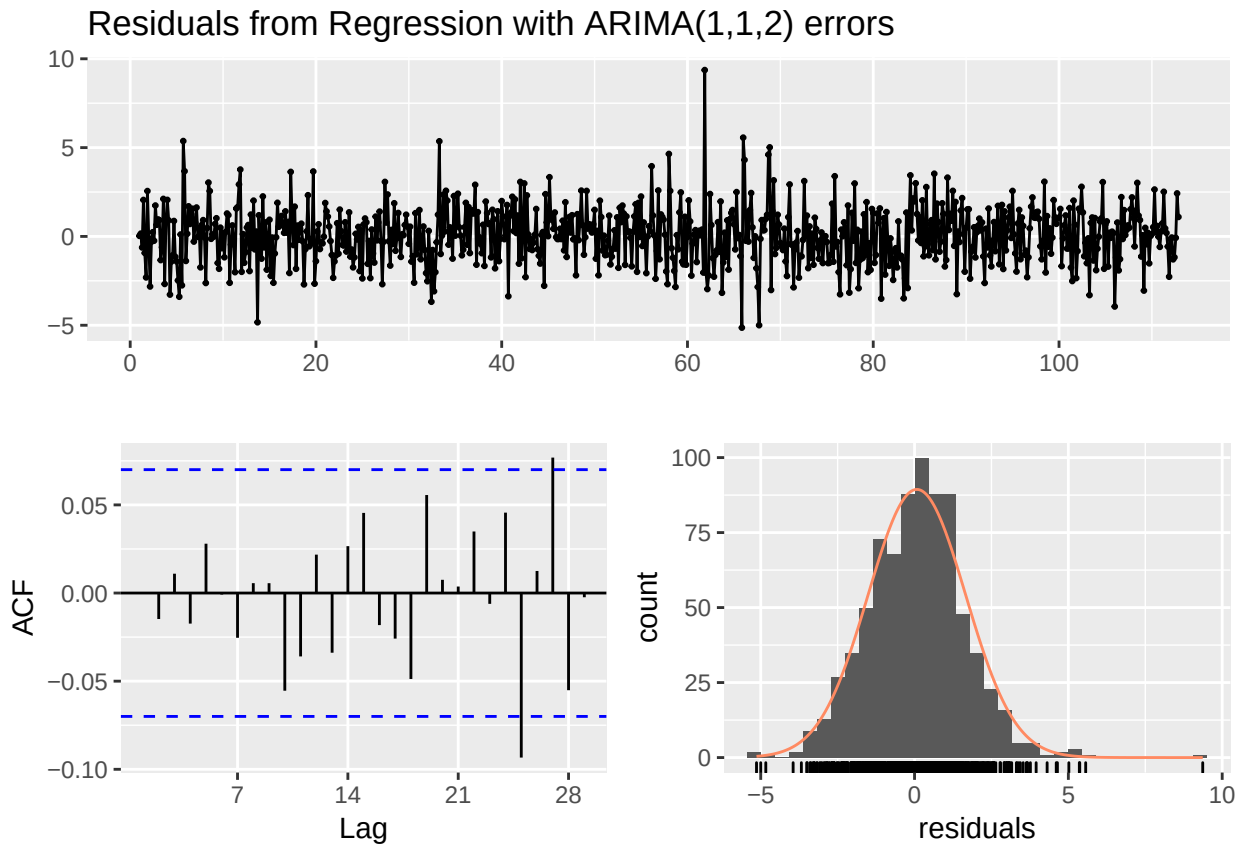
```
autoplot(inv_pred, include = 10)+
  autolayer(kidney_test_weekly,series = "XREG", color = "red")
```

```
## Warning: Ignoring unknown parameters: include
```



Fourier Transformation for Dynamic Harmonic Regression

```
harmonics = fourier(kidney_transformed_train, K =3)
model_fourier = auto.arima(kidney_transformed_train, xreg = harmonics, seasonal = FALSE)

checkresiduals(model_fourier)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,1,2) errors
## Q* = 7.0225, df = 5, p-value = 0.219
##
## Model df: 9.   Total lags used: 14
```

```
summary(model_fourier)
```

```
## Series: kidney_transformed_train
## Regression with ARIMA(1,1,2) errors
##
## Coefficients:
##          ar1      ma1     ma2    S1-7    C1-7     S2-7    C2-7     S3-7
##       0.6723  -1.2588  0.3094  0.4168  0.9855  -0.4158  0.2570  -0.2429
## s.e.  0.0877   0.1025  0.0870  0.0824  0.0823   0.0655  0.0655   0.0619
##          C3-7
##       -0.3319
## s.e.   0.0619
##
## sigma^2 estimated as 2.55:  log likelihood=-1473.51
## AIC=2967.02   AICc=2967.31   BIC=3013.65
##
## Training set error measures:
```

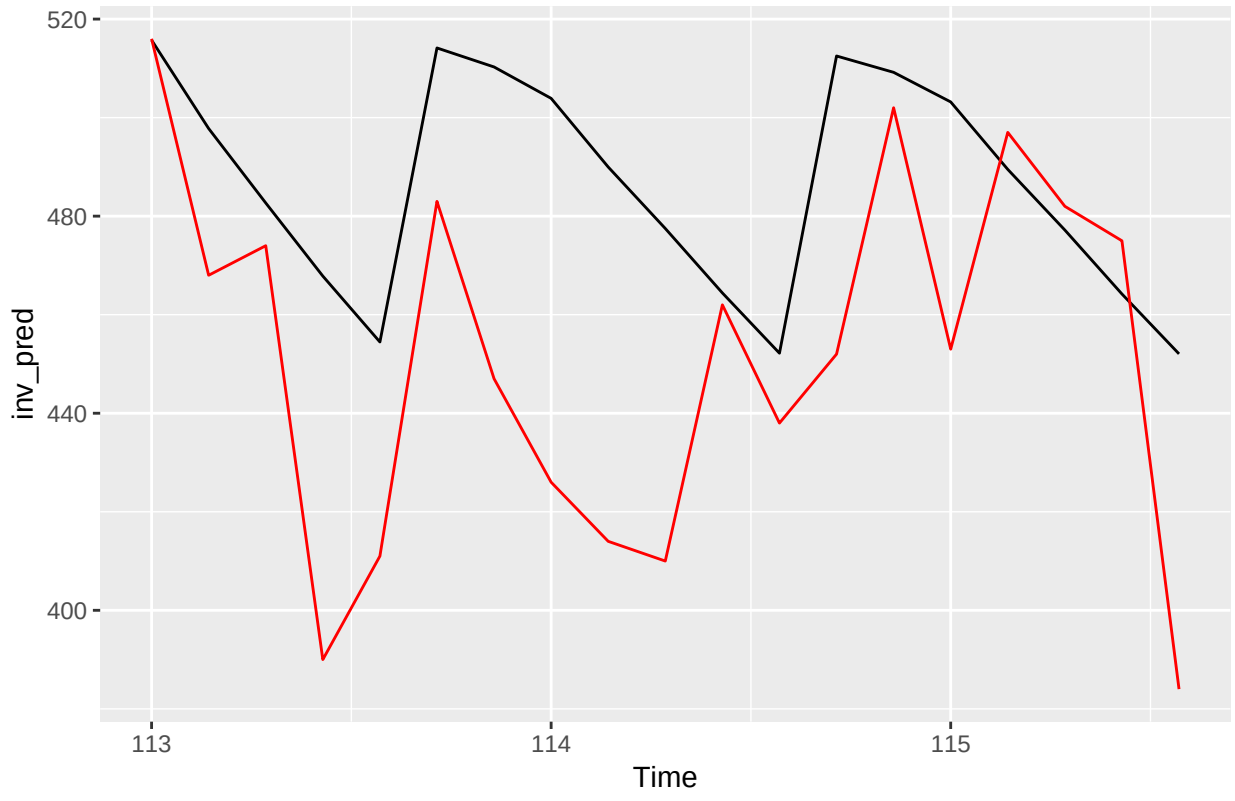
```
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 0.07584121 1.586781 1.228984 0.01459153 3.660009 0.6645907
##                        ACF1
## Training set -0.0003133193
```

```
new_harmonics = fourier(kidney_transformed_train, K =3, h = 19)
pred = forecast(model_fourier, xreg = new_harmonics)

inv_pred = InvBoxCox(pred$mean, lambda = kidney_lambda)
acc_fourier = accuracy(inv_pred, kidney_test_weekly)
acc_fourier
```

```
##                  ME     RMSE      MAE       MPE     MAPE      ACF1
## Test set -34.45598 46.76135 36.92016 -8.156342 8.665715 0.1593996
##          Theil's U
## Test set  1.097786
```

```
autoplot(inv_pred, include = 10)+
  autolayer(kidney_test_weekly,series = "SARIMA", color = "red")
```

```
## Warning: Ignoring unknown parameters: include
```



VAR model

```
var_data = window(ts.union(diff(kidney_transformed_train), diff(append_transformed_train)))

model_var = VAR(y = var_data, p = 2)
model_var
```

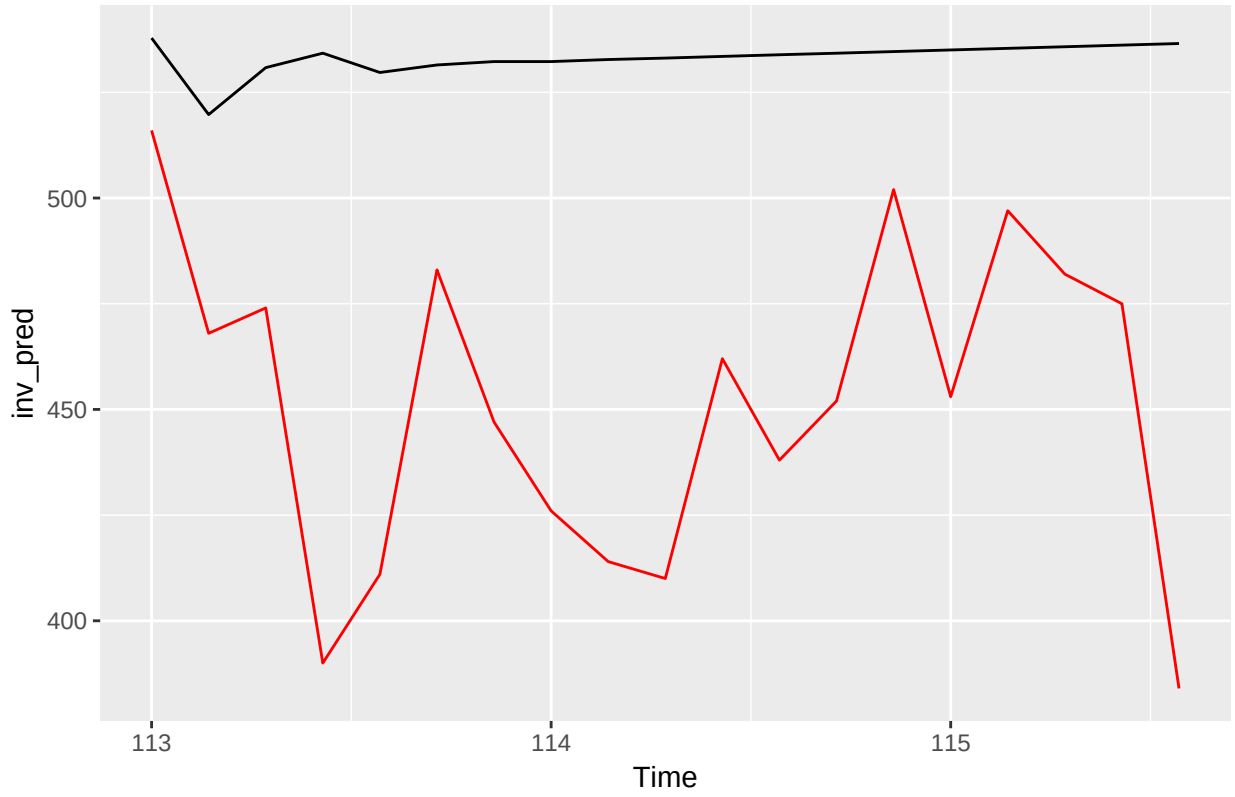
```
##
## VAR Estimation Results:
## ========================
##
## Estimated coefficients for equation diff.kidney_transformed_train.:
## ====================================================================
## Call:
## diff.kidney_transformed_train. = diff.kidney_transformed_train..l1 + diff.append_transformed_train..l
##
## diff.kidney_transformed_train..l1 diff.append_transformed_train..l1
##                        -0.41395943                         0.23504505
## diff.kidney_transformed_train..l2 diff.append_transformed_train..l2
##                        -0.20570354                         0.27970057
##                              const
##                         0.01420989
##
##
## Estimated coefficients for equation diff.append_transformed_train.:
## ====================================================================
## Call:
## diff.append_transformed_train. = diff.kidney_transformed_train..l1 + diff.append_transformed_train..l
##
## diff.kidney_transformed_train..l1 diff.append_transformed_train..l1
##                        -0.04124463                        -0.42430464
## diff.kidney_transformed_train..l2 diff.append_transformed_train..l2
##                        -0.08589516                        -0.19455270
##                              const
##                         0.03255207
```

```
pred = forecast(model_var, h = 19)
pred_kidney = pred$forecast$diff.kidney_transformed_train.$mean
pred_kidney = ts(pred_kidney, start = c(113,1), end = c(115,5), frequency = 7)

act_pred = append(kidney_transformed_train[784],pred_kidney)
act_pred = cumsum(act_pred)
act_pred = act_pred[2:20]
inv_pred = InvBoxCox(act_pred, lambda = kidney_lambda)
inv_pred = ts(inv_pred, start = c(113,1), end = c(115,5), frequency = 7)
accuracy(inv_pred, kidney_test_weekly)
```

```
##                  ME     RMSE      MAE      MPE    MAPE      ACF1 Theil's U
## Test set -81.34758 89.36197 81.34758 -18.8218 18.8218 0.1610737  2.101848
```

```
autoplot(inv_pred, include = 100)+
  autolayer(kidney_test_weekly,series = "SARIMA", color = "red")
```

```
## Warning: Ignoring unknown parameters: include
```



I exclude the VAR model because it doesn't have any prediciton power shown above.

Model Comparasion

```
aicc <- cbind(arima_atuo_kidney$aicc, arima_kidney$aicc, tbats_kidney$AIC,
              summary(model_arfima)$aic, model_fourier$aicc , model_xreg$aicc)
aicc
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 3119.277 2991.963 10764.55 3196.742 2967.307 3111.077
```

```
#sort(aicc)

rmse <- c(acc_atuo[,2], acc_arima[,2], acc_tbats[2,2],
          acc_arfima[,2], acc_fourier[,2], acc_xreg[,2])
rmse
```

```
## [1] 53.33778 51.02664 46.30050 51.48498 46.76135 49.69068
```

```
data <- rbind(aicc,rmse)
comparison <- as.data.frame(t(data),row.name=c('Auto_arima','SARIMA','TBATS','ARFIMA','Fourier', 'Regres
colnames(comparison)<-c('AICC', 'RMSE')
comparison[order(comparison$RMSE),]
```

```
##                                   AICC     RMSE
## TBATS                         10764.550 46.30050
## Fourier                        2967.307 46.76135
## Regression with arima error    3111.077 49.69068
## SARIMA                         2991.963 51.02664
## ARFIMA                         3196.742 51.48498
## Auto_arima                     3119.277 53.33778
```

By Comparing all models' performance. TBATS has the best performance in fitting by showing the lowest RMSE. However, it has very high AICC, so I consider Fourier Transformation for Dynamic Harmonic Regression has the best performance.