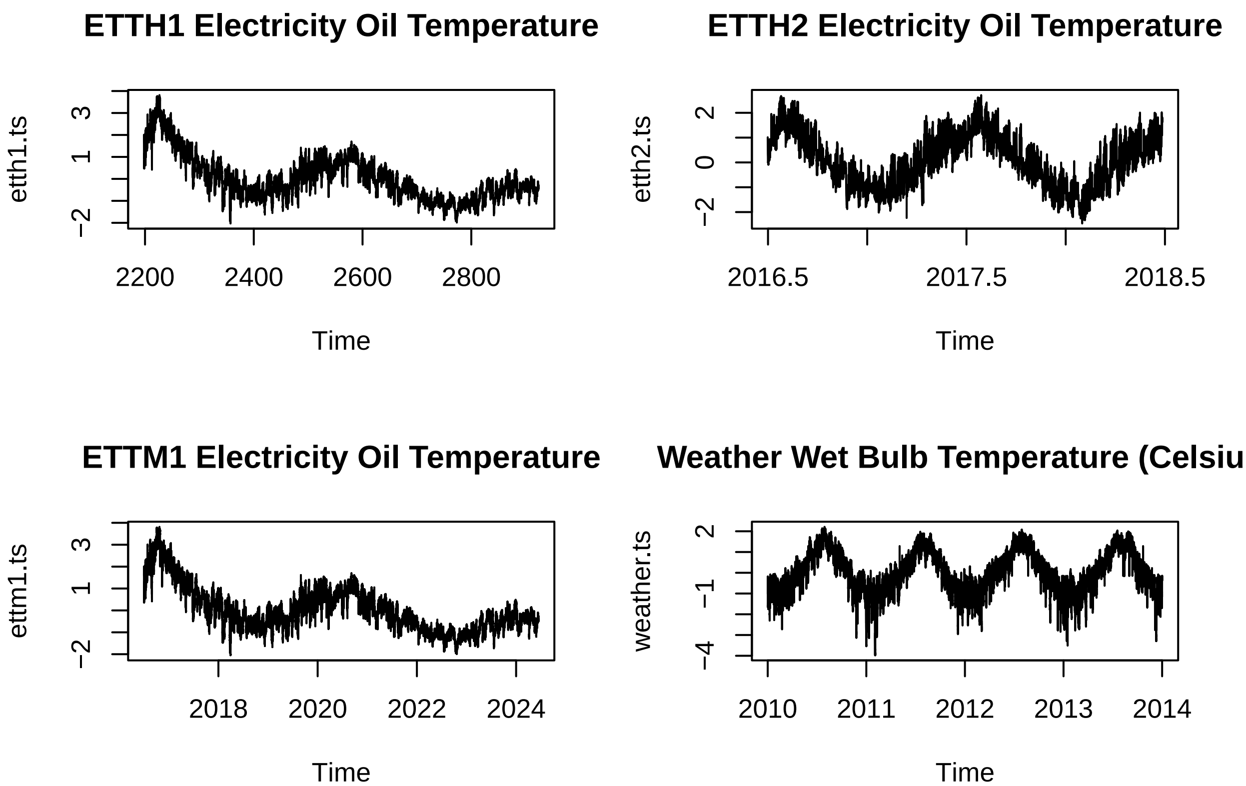
Traditional Methods 207 Final Project

Dylan Chou, Tim Yao

2024-05-30

Two Time Series Plots of Electricity and Weather

```
par(mfrow=c(2,2))
etth1 = read.csv("./data/ETTh1.csv")
etth2 = read.csv("./data/ETTh2.csv")
ettm1 = read.csv("./data/ETTm1.csv")

weather = read.csv("./data/WTH.csv")
# univariate time series
# https://stackoverflow.com/questions/33782218/how-to-create-a-time-series-of-hourly-data
first_hour_etth1 = 24*(as.Date("2016-07-01 00:00:00")-as.Date("2016-1-1 00:00:00"))
etth1.ts = ts(data=etth1$OT, start=c(2016, first_hour_etth1), freq=24)
etth1.ts = (etth1.ts-mean(etth1.ts))/sd(etth1.ts)

etth2.ts = ts(data=etth2$OT, start=c(2016, first_hour_etth1), freq=24*365)
etth2.ts = (etth2.ts-mean(etth2.ts))/sd(etth2.ts)

ettm1.ts = ts(data=ettm1$OT, start=c(2016, first_hour_etth1), freq=24*365)
ettm1.ts = (ettm1.ts-mean(ettm1.ts))/sd(ettm1.ts)

weather.ts = ts(data=weather$WetBulbCelsius, start=c(2010, 0), freq=24*365)
weather.ts = (weather.ts-mean(weather.ts))/sd(weather.ts)

plot(etth1.ts, main="ETTH1 Electricity Oil Temperature")
plot(etth2.ts, main="ETTH2 Electricity Oil Temperature")
plot(ettm1.ts, main="ETTM1 Electricity Oil Temperature")
plot(weather.ts, main="Weather Wet Bulb Temperature (Celsius)")
```

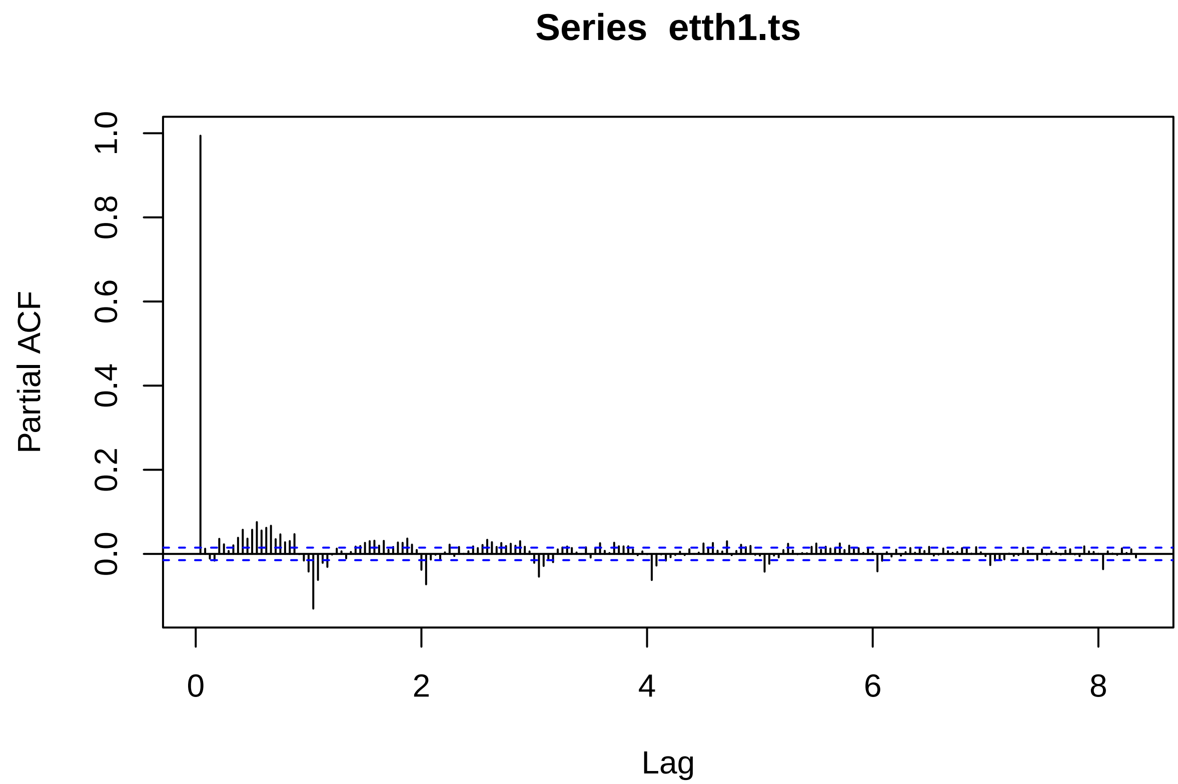


Autocorrelation of current time series

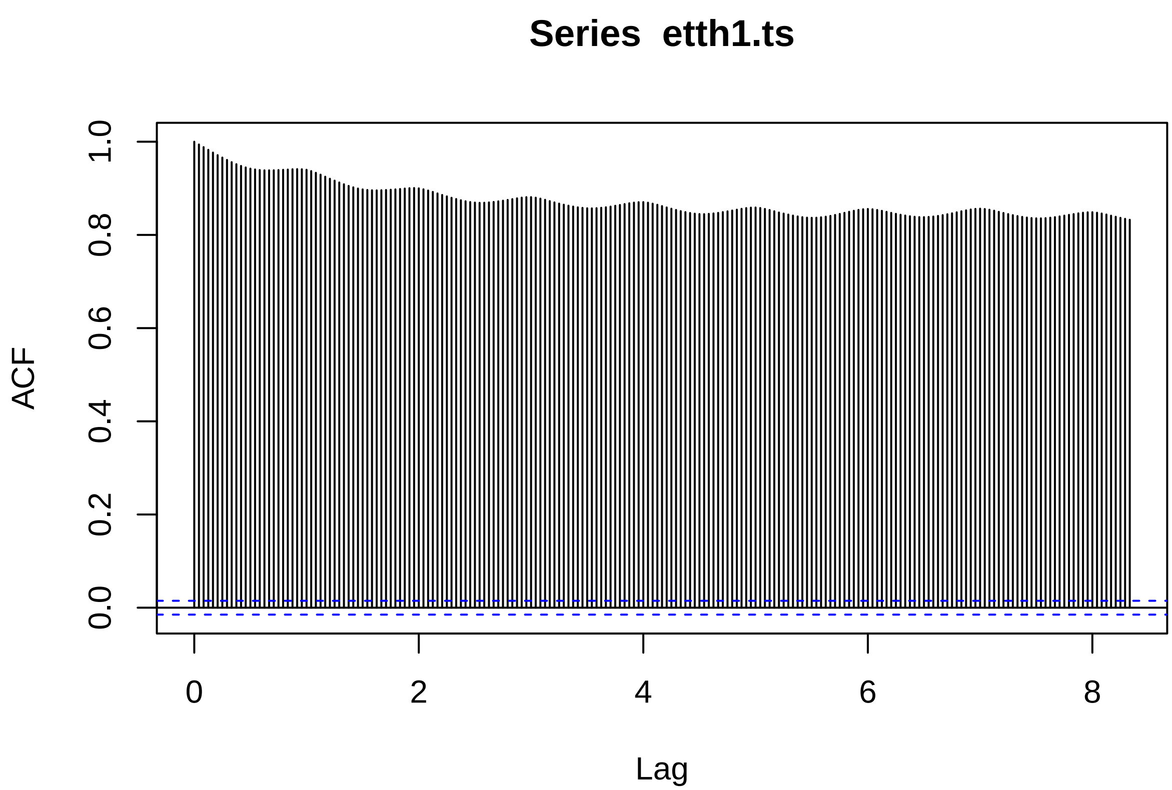
```
# first diagnose the autocorrelation of the existing time series
length(etth1.ts)
```

```
## [1] 17420
```

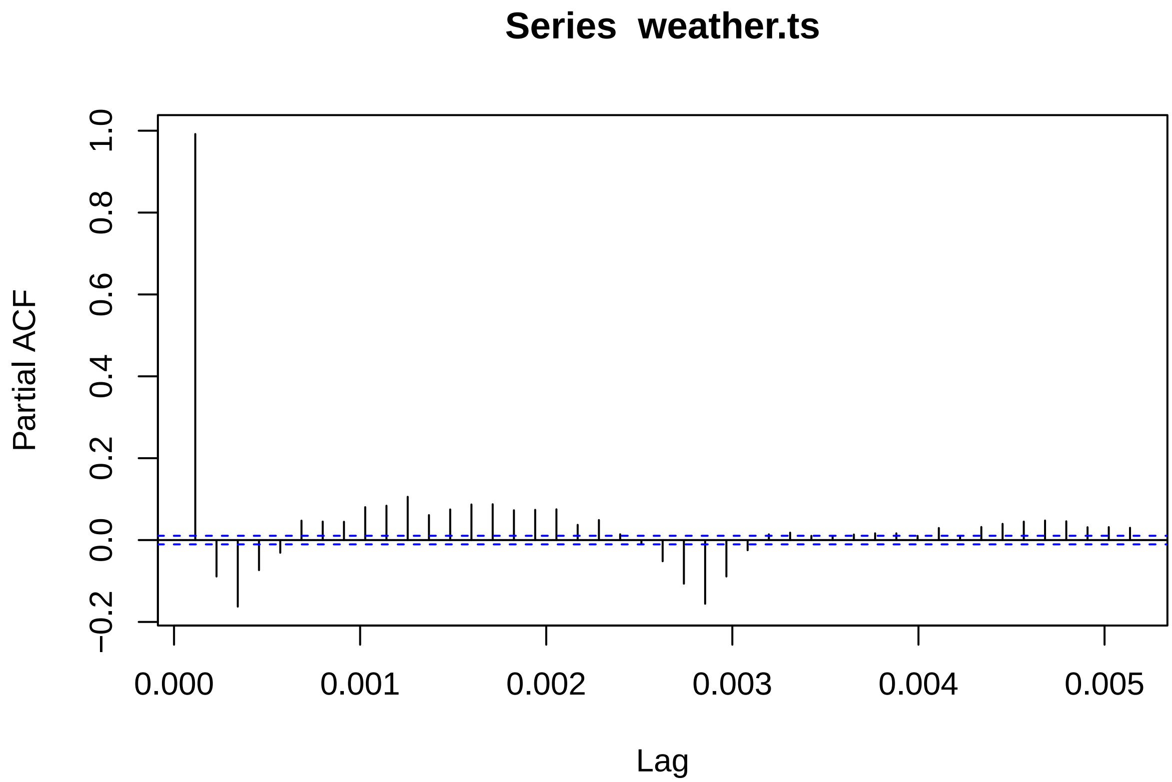
```
pacf(etth1.ts, lag.max=200)
```



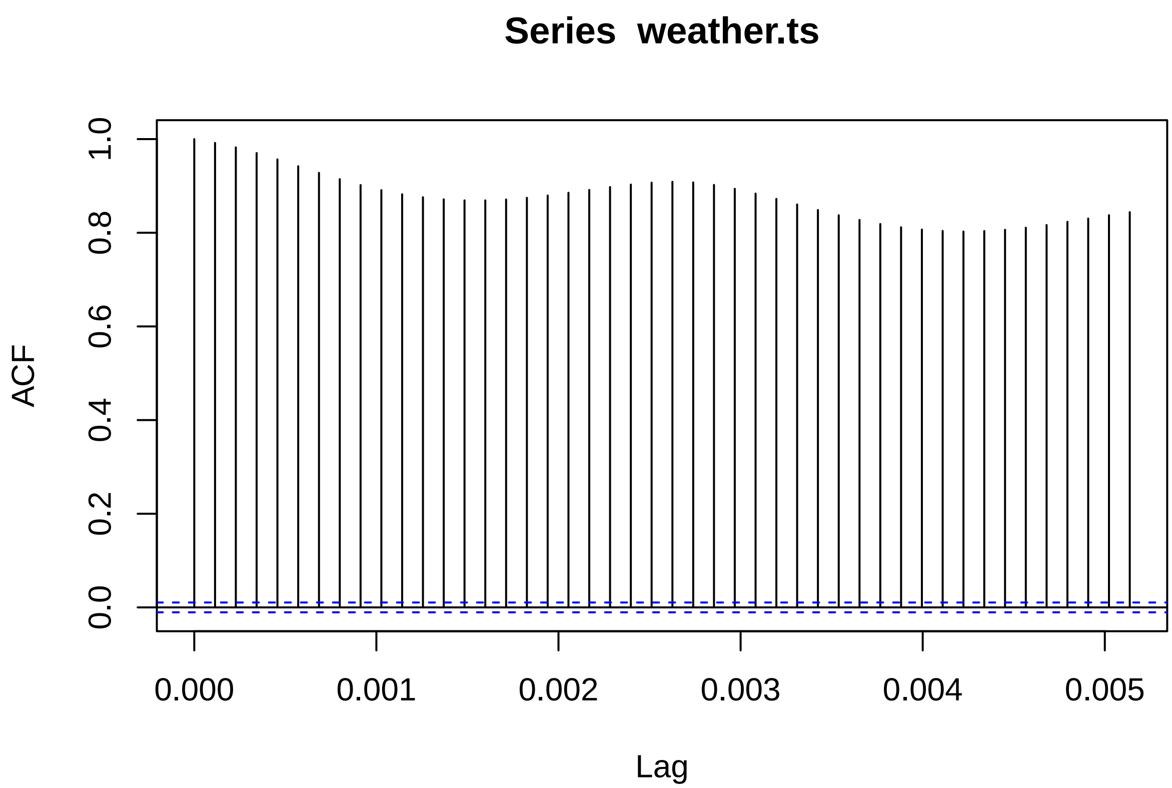

```
acf(etth1.ts, lag.max=200)
```



```
pacf(weather.ts)
```

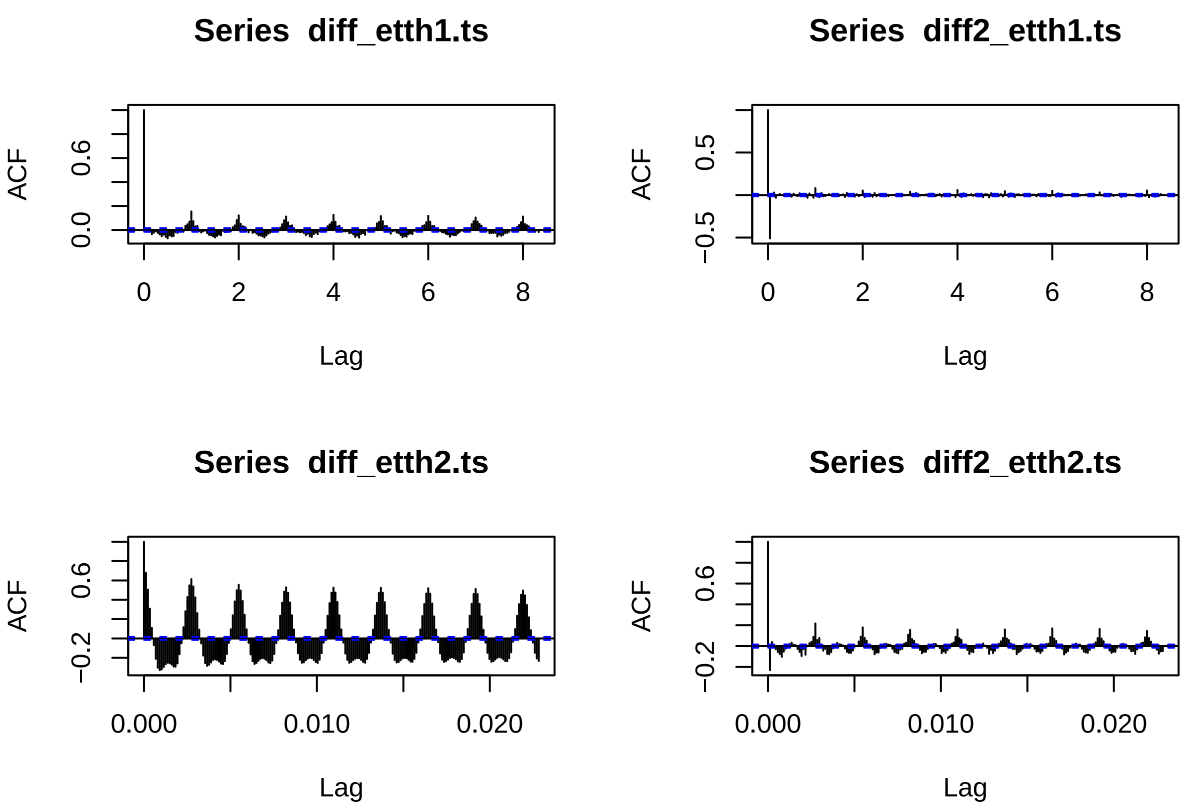



```
acf(weather.ts)
```



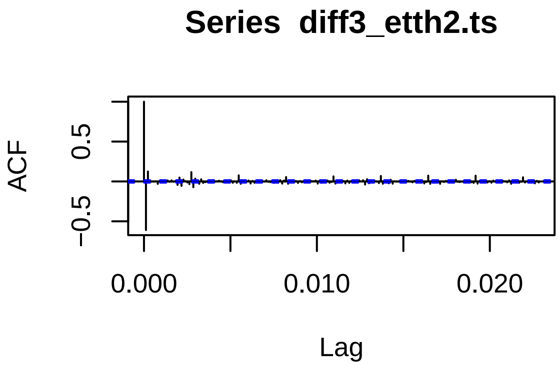
Differencing

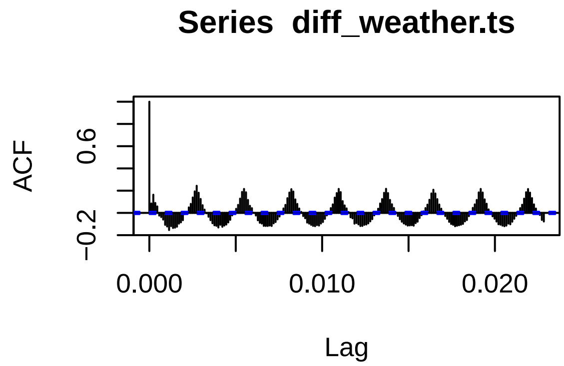
```r
par(mfrow=c(2,2))
# Electricity - trend elimination

diff_etth1.ts = diff(etth1.ts)
diff2_etth1.ts = diff(diff(etth1.ts))
acf(diff_etth1.ts, lag.max=200)
acf(diff2_etth1.ts,lag.max=200)


diff_etth2.ts = diff(etth2.ts)
diff2_etth2.ts = diff(diff(etth2.ts))
diff3_etth2.ts = diff(diff(diff(etth2.ts)))
acf(diff_etth2.ts, lag.max=200)
acf(diff2_etth2.ts,lag.max=200)
```

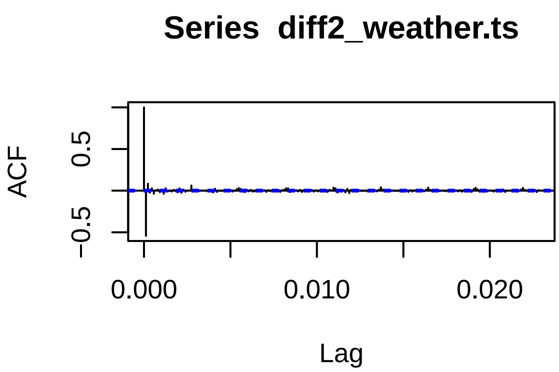


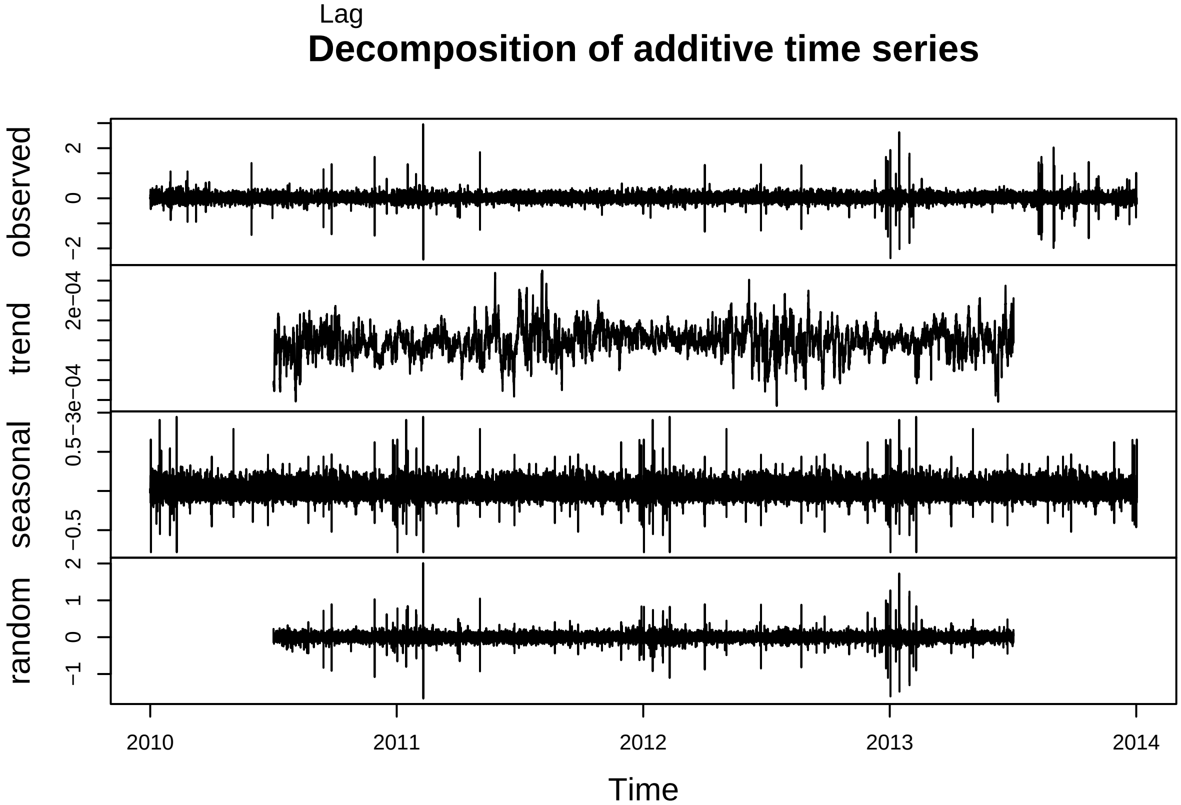

```r
acf(diff3_etth2.ts,lag.max=200)

# Weather

diff_weather.ts = diff(weather.ts)
diff2_weather.ts = diff(diff(weather.ts))
acf(diff_weather.ts,lag.max=200)
acf(diff2_weather.ts,lag.max=200)
```

```
plot(decompose(diff_weather.ts))
```









```r
# Augmented Dickey-Fuller Test

# reject null that ts is non-stationary
adf.test(weather.ts)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  weather.ts
## Dickey-Fuller = -8.1762, Lag order = 32, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test(diff_weather.ts)
```

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

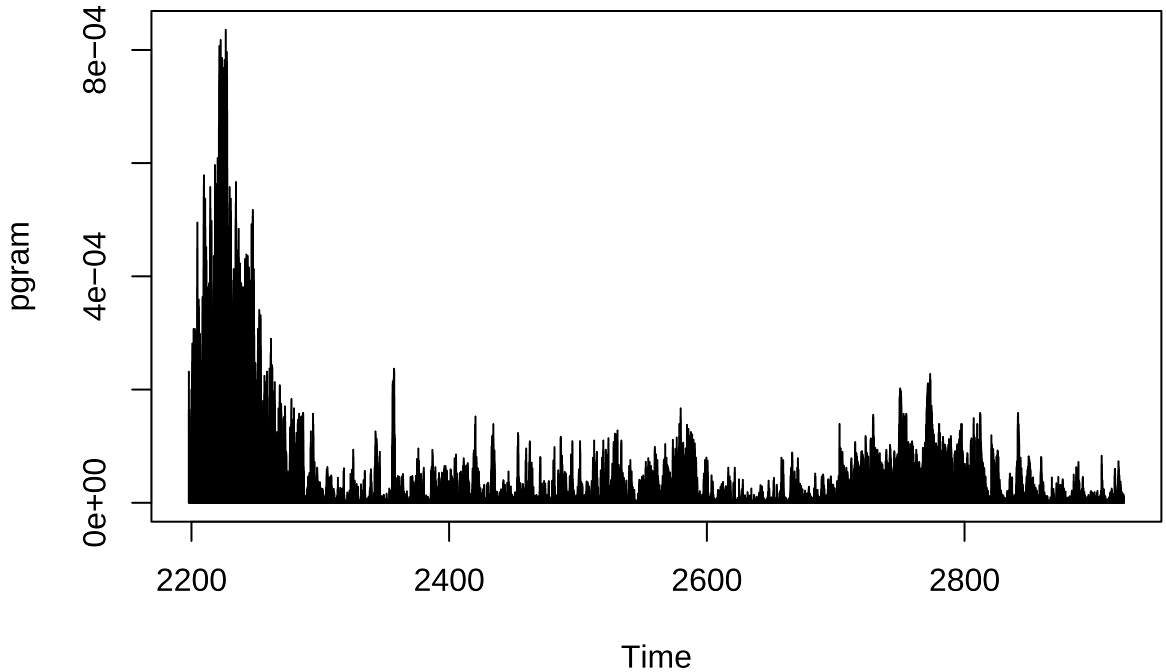
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_weather.ts
## Dickey-Fuller = -34.887, Lag order = 32, p-value = 0.01
## alternative hypothesis: stationary
```

Run some spectral analysis for seasonality detection

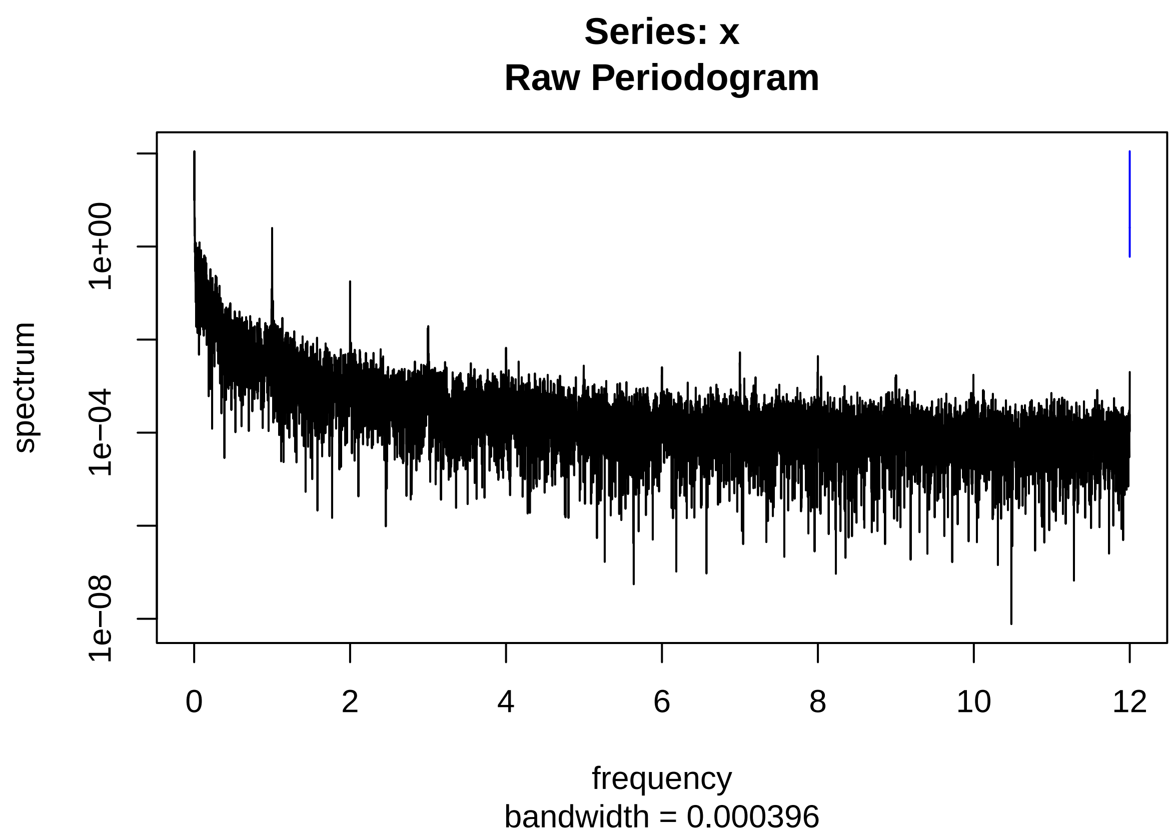
```r
pgram = abs(etth1.ts)^2/length(etth1.ts)
which.max(pgram)
```

```
## [1] 688
```

```r
plot(pgram, type = "h")
```

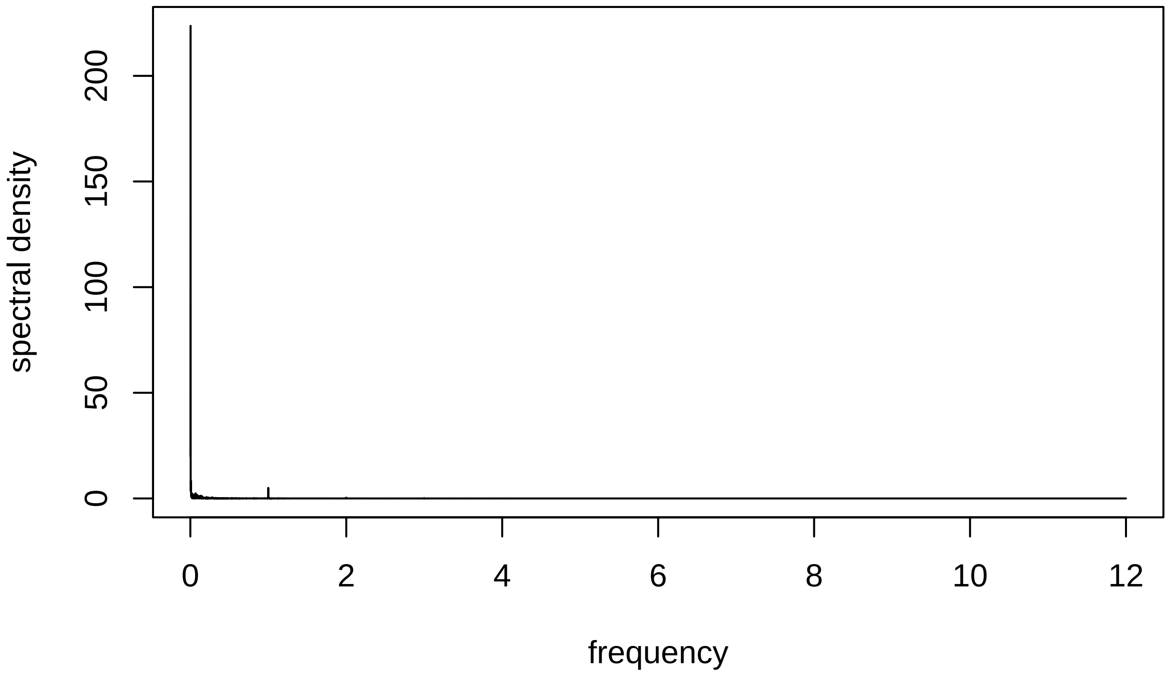



```
x.spec = spectrum(etth1.ts)
```



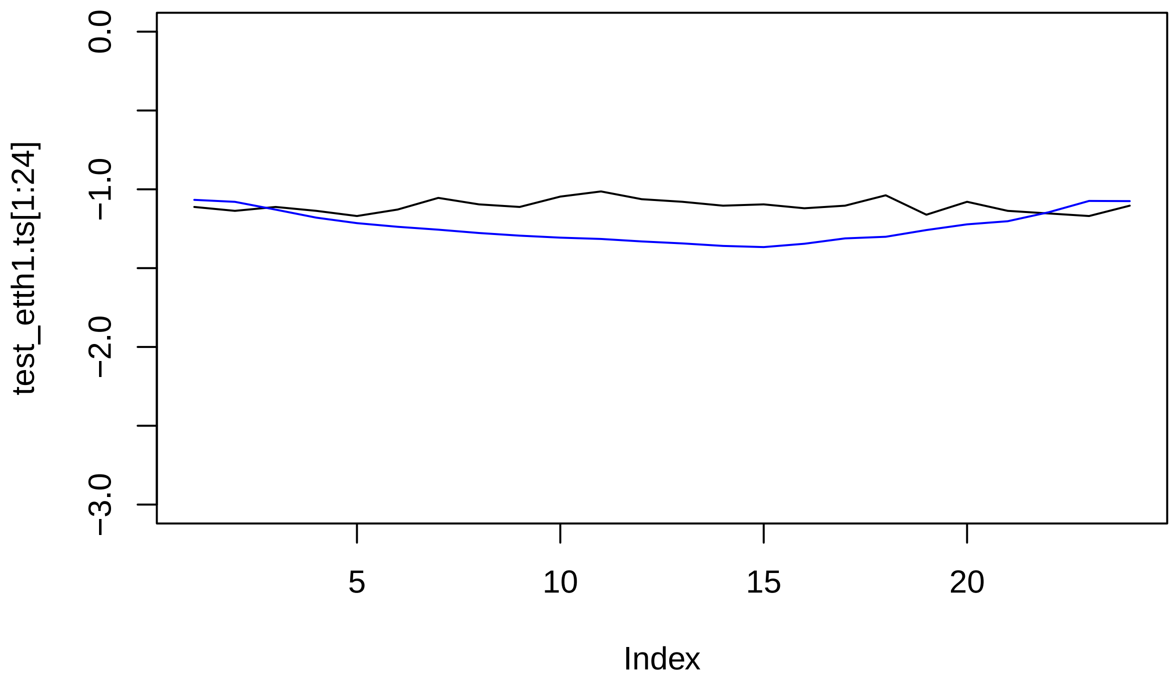

```
spx <- x.spec$freq
spy <- 2*x.spec$spec
```

```
plot (spy~spx, subset=spx<=100,xlab="frequency",ylab="spectral density",type = "l")
```

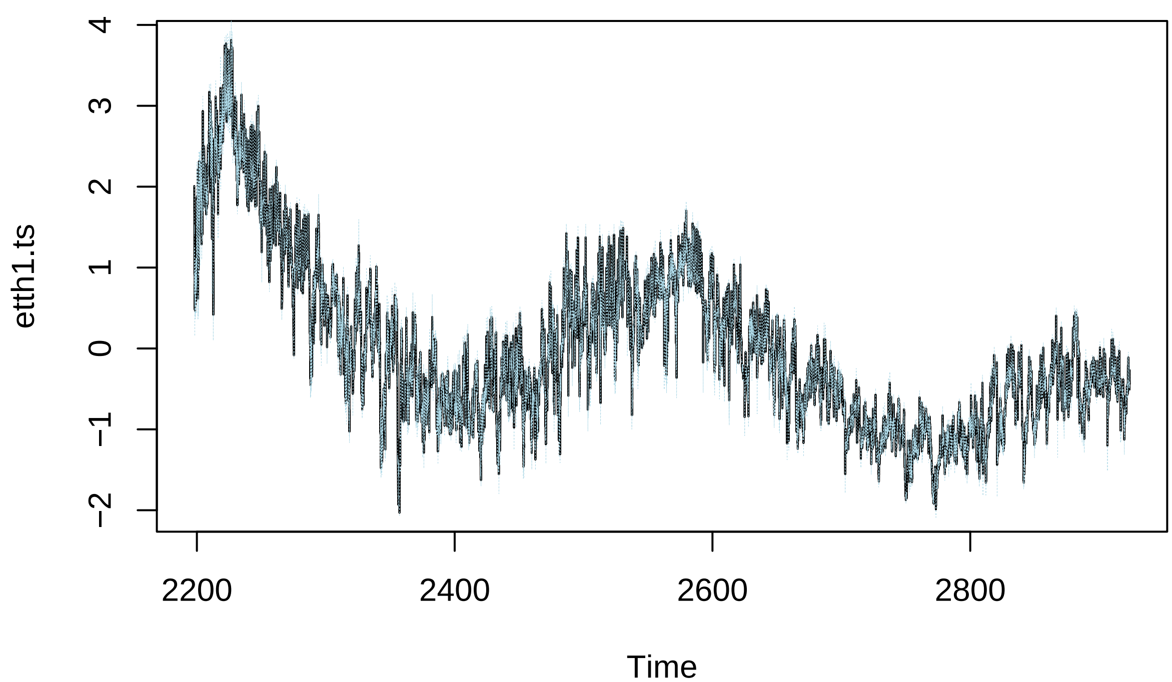


Trying different models based on preliminary plots

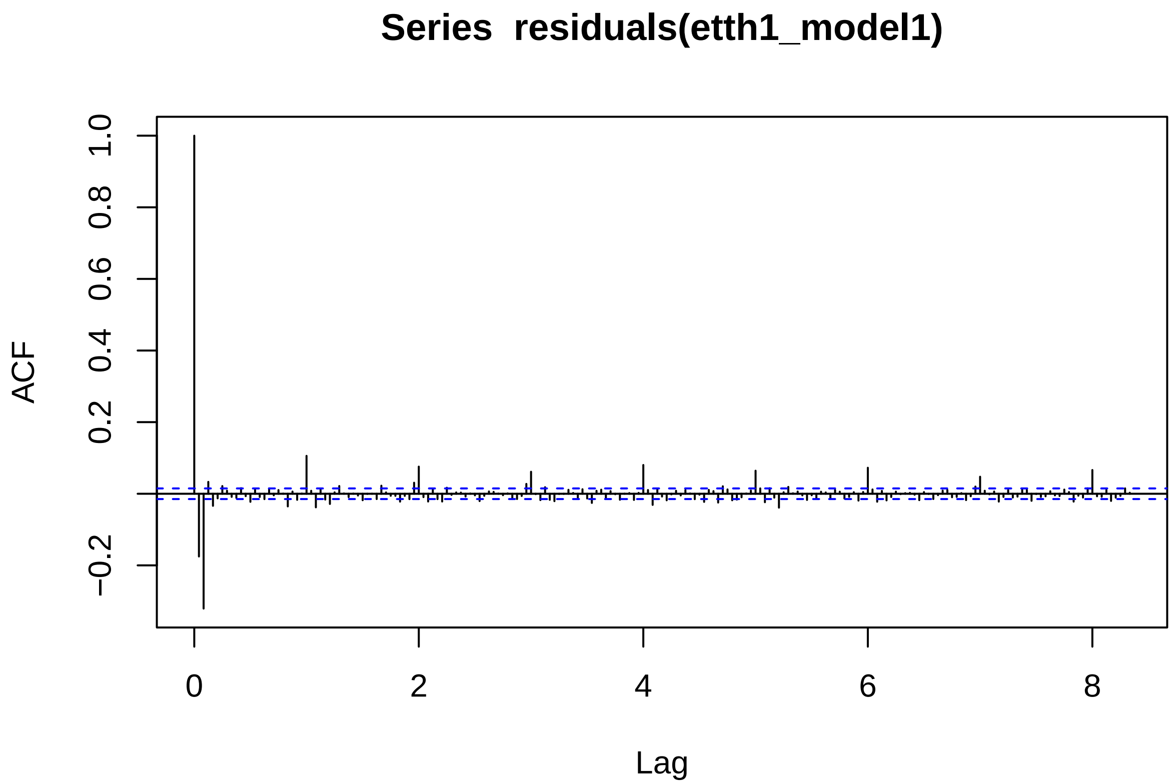
```
# KFAS
test_start = (length(etth1.ts)*0.80)+1
train_etth1.ts = ts(etth1.ts[1:(length(etth1.ts)*0.80)], start=c(2016, first_hour_etth1), frequency=24)

test_etth1.ts = etth1.ts[test_start:length(etth1.ts)]
ssmodel1 = SSModel(train_etth1.ts ~ SSMtrend(1, Q = NA) + SSMseasonal(24, sea.type = "dummy", Q = NA), H
fitssmodel1 = fitSSM(ssmodel1, inits = c(0, 0, 0))
n_ahead = 24
forecast = predict(fitssmodel1$model, n.ahead = n_ahead, interval = "prediction")
plot(test_etth1.ts[1:24],type="l",ylim=c(-3,0))
points(forecast[1:24], type="l", col="blue")
```



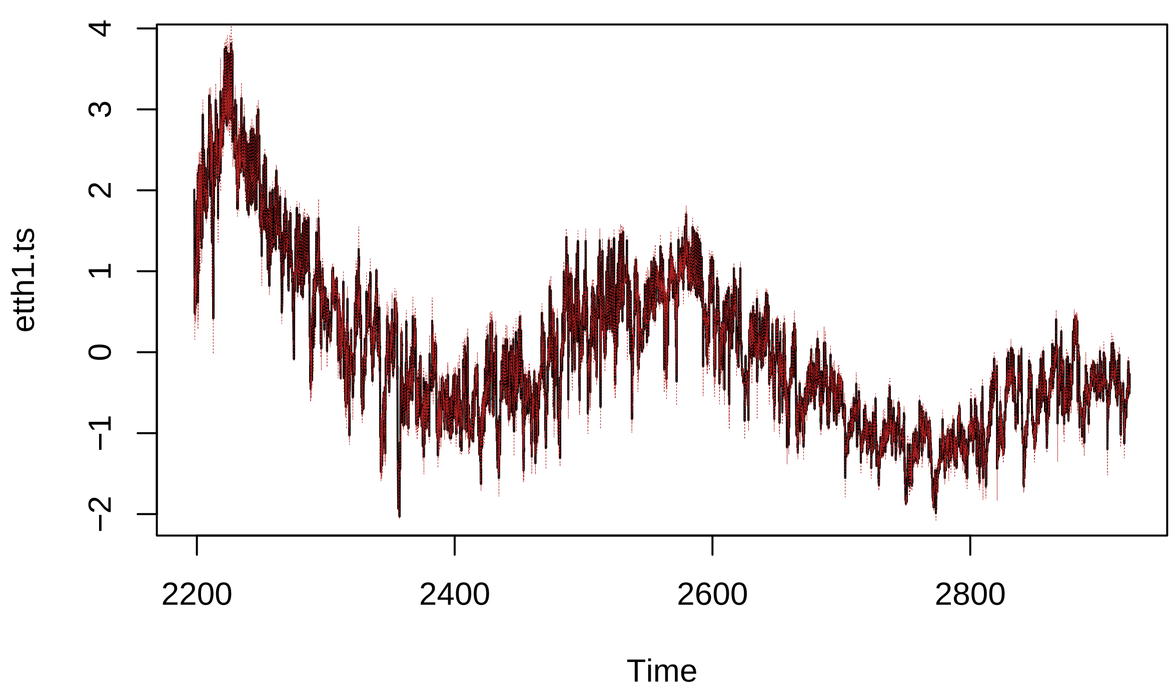

```
# ETTH1
# Ideal to difference twice to remove trend. Estimates using CSS-ML
etth1_model1 = arima(etth1.ts, order = c(1,2,0))
plot(etth1.ts)
fit1 = etth1.ts - residuals(etth1_model1)
points(fit1, type = "l", col = "lightblue", lty = 2, lwd=0.2)
```



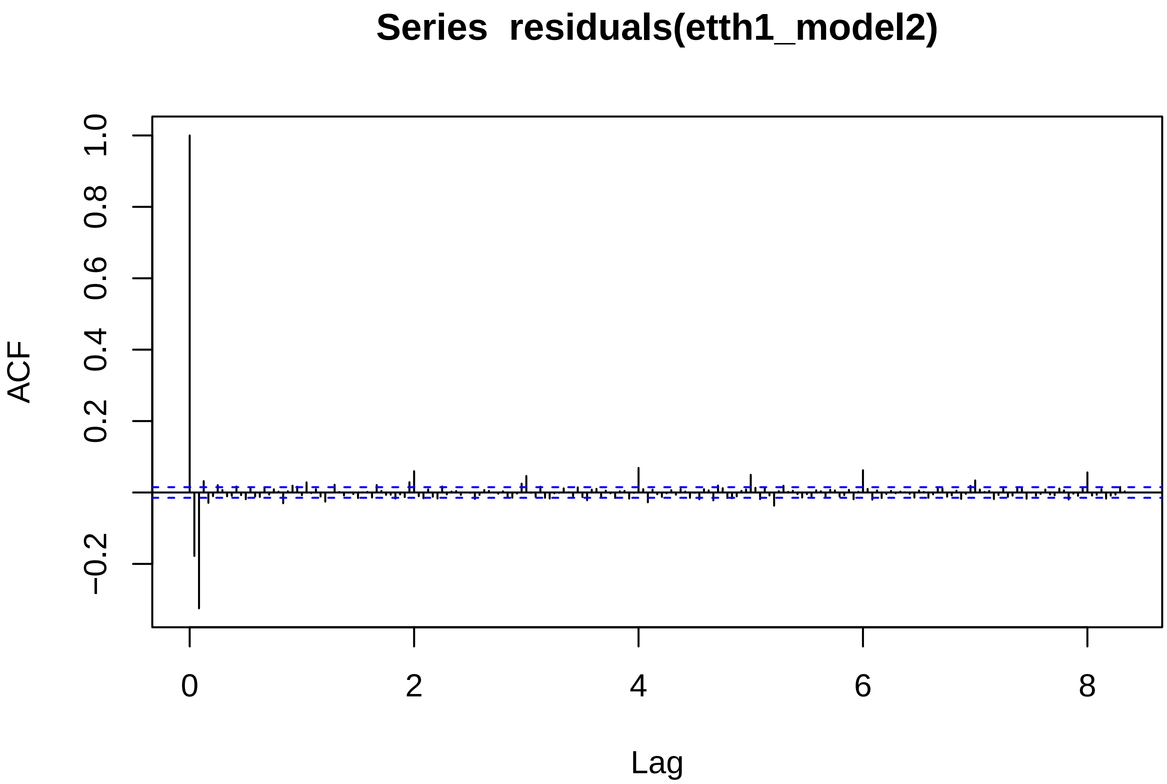

```
acf(residuals(etth1_model1), lag.max=200)
```



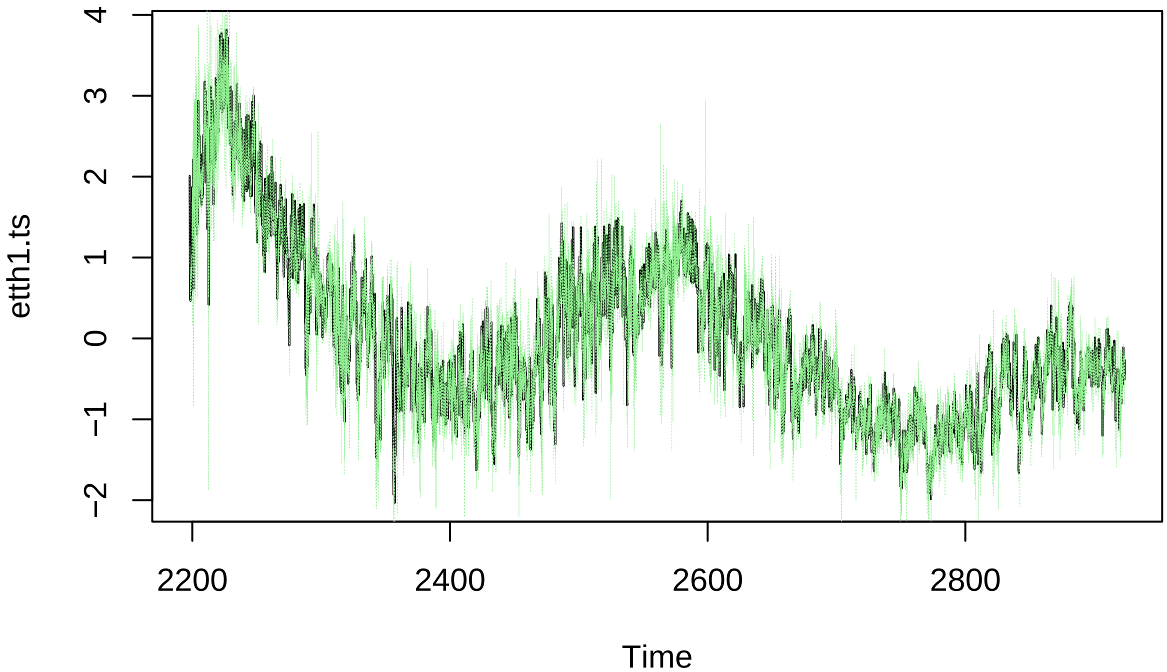

```
# seasonal component every 24 time steps (every 24 hours).
etth1_model2 = arima(etth1.ts, order = c(1,2,0), seasonal=list(order=c(1,0,0), period=24))
plot(etth1.ts)
fit2 = etth1.ts - residuals(etth1_model2)
points(fit2, type = "l", col = "firebrick", lty = 2, lwd=0.2)
```



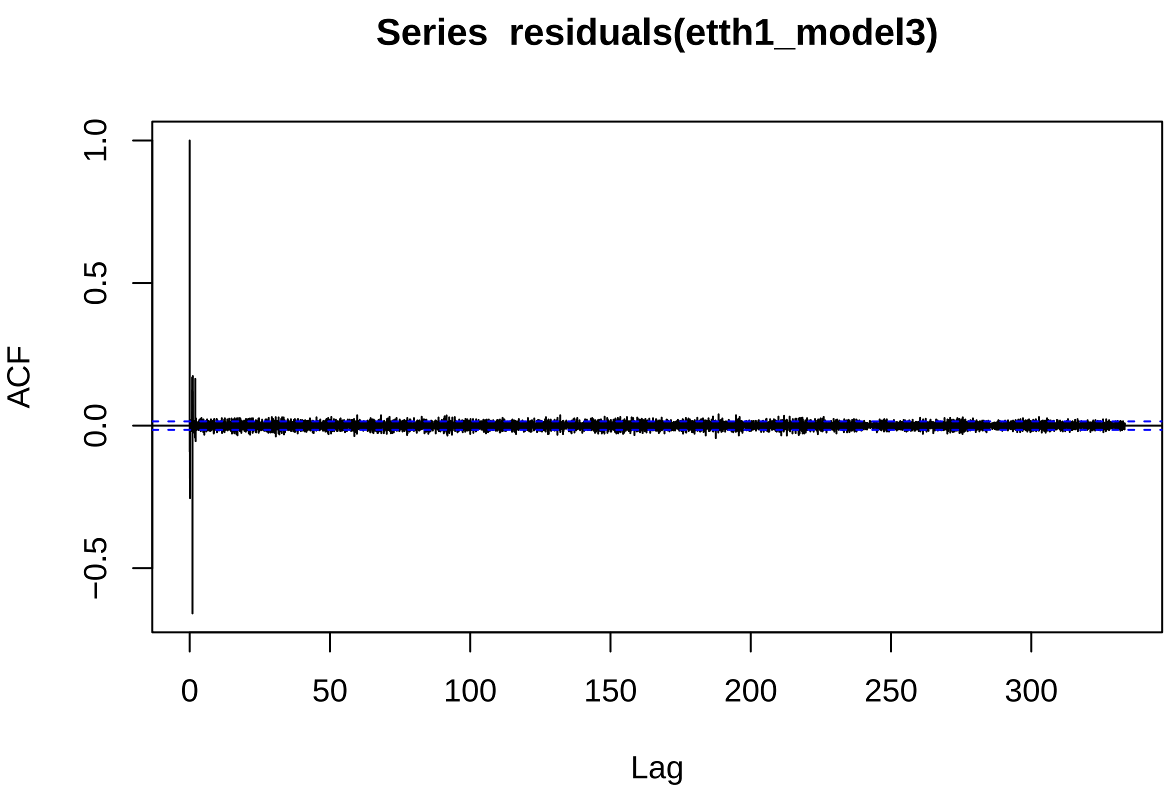
```r
acf(residuals(etth1_model2), lag.max=200)
```



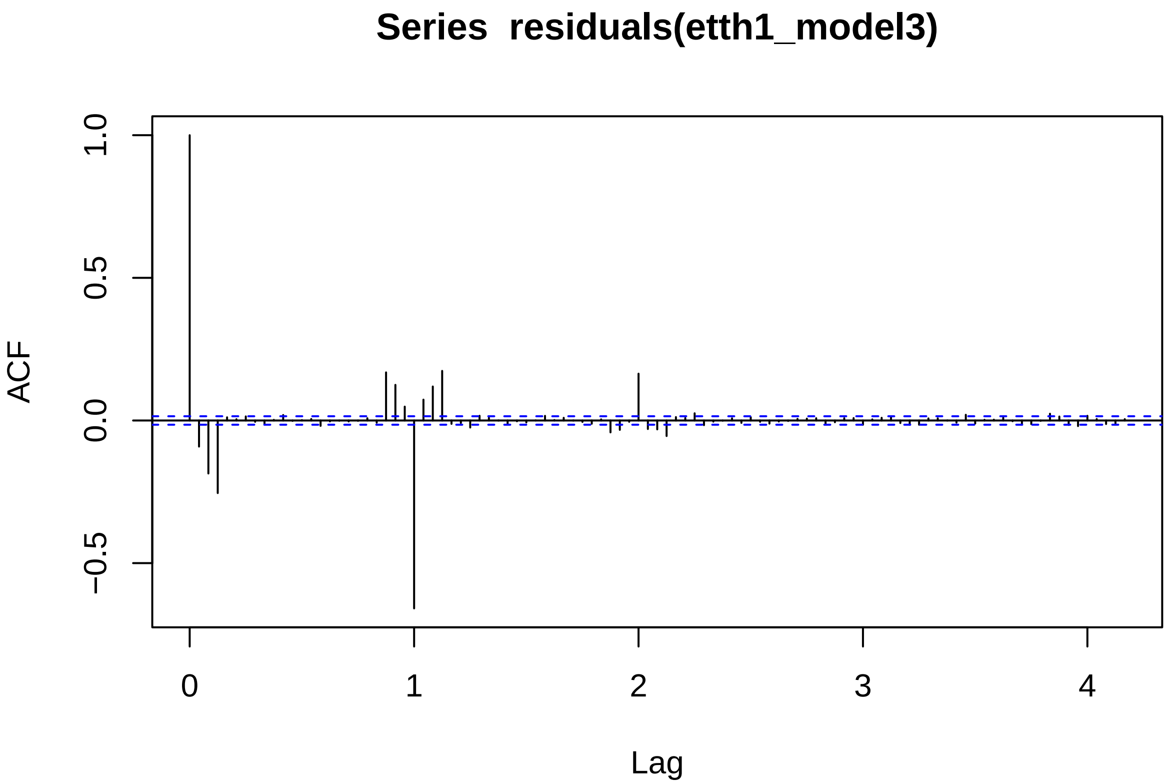

```r
# significant lag 1 and 2 components. Trying MA component
etth1_model3 = arima(etth1.ts, order = c(2,2,0), seasonal=list(order=c(0,2,0), period=24))
plot(etth1.ts)
fit3 = etth1.ts - residuals(etth1_model3)
points(fit3, type = "l", col = "lightgreen", lty = 2, lwd=0.2)
```




```
acf(residuals(etth1_model3), lag.max=8000)
```



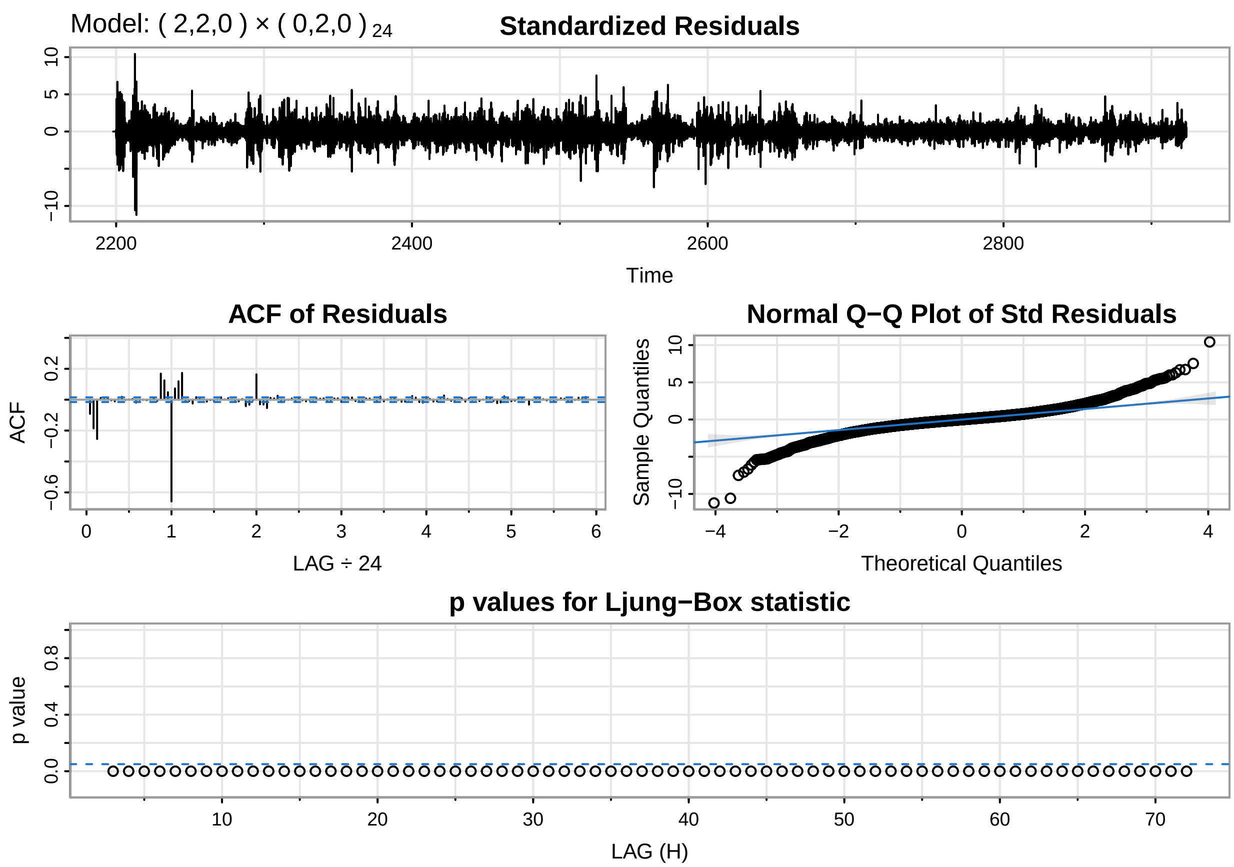
```r
acf(residuals(etth1_model3), lag.max=100)
```



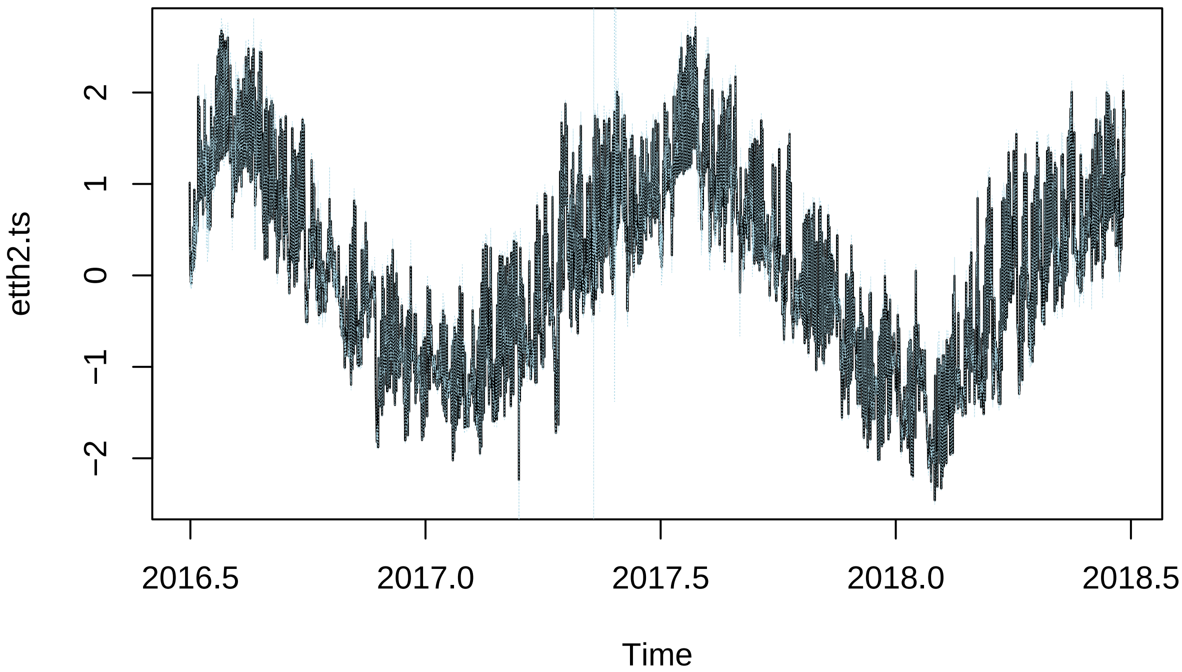

```r
# we reject the null hypothesis that lag-1 and original time series are independent.
sarima(etth1.ts,p=2,d=2,q=0,P=0,D=2,Q=0,S=24)
```

```
## initial  value -1.034497
## iter   2 value -1.190028
## iter   3 value -1.263929
## iter   4 value -1.265926
## iter   5 value -1.266788
## iter   6 value -1.266788
## iter   6 value -1.266788
## final  value -1.266788
## converged
## initial  value -1.266766
## iter   2 value -1.266774
## iter   3 value -1.266778
## iter   4 value -1.266781
## iter   5 value -1.266793
## iter   6 value -1.266796
## iter   6 value -1.266770
## final  value -1.266796
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##     Estimate     SE   t.value p.value
## ar1  -0.7193 0.0033 -216.1748       0
```

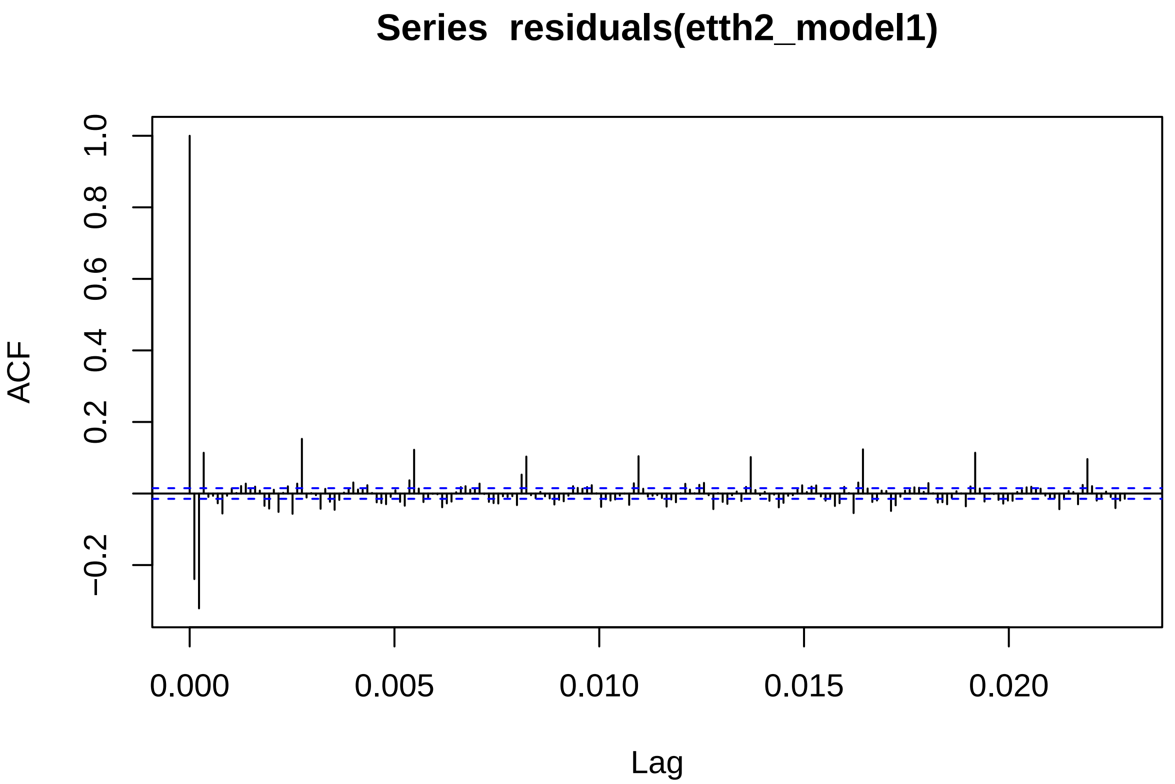
```
## ar2  -0.3546 0.0023 -154.8002       0
##
## sigma^2 estimated as 0.07936574 on 17368 degrees of freedom
##
## AIC = 0.3046304  AICc = 0.3046304  BIC = 0.3059711
##
```



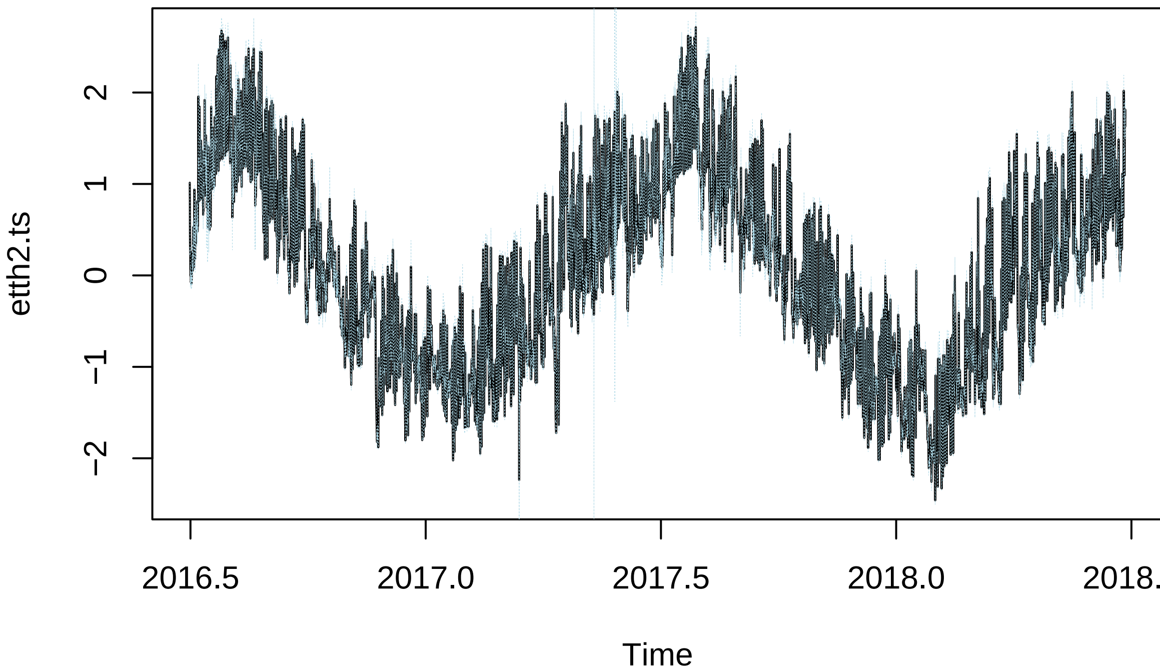

```r
# ETTh2 - 3rd order differencing. Trying an AR term due to acf plot.
etth2_model1 = arima(etth2.ts, order=c(1,3,0))
plot(etth2.ts)
etth2_fit1 = etth2.ts - residuals(etth2_model1)
points(etth2_fit1, type = "l", col = "lightblue", lty = 2, lwd=0.2)
```



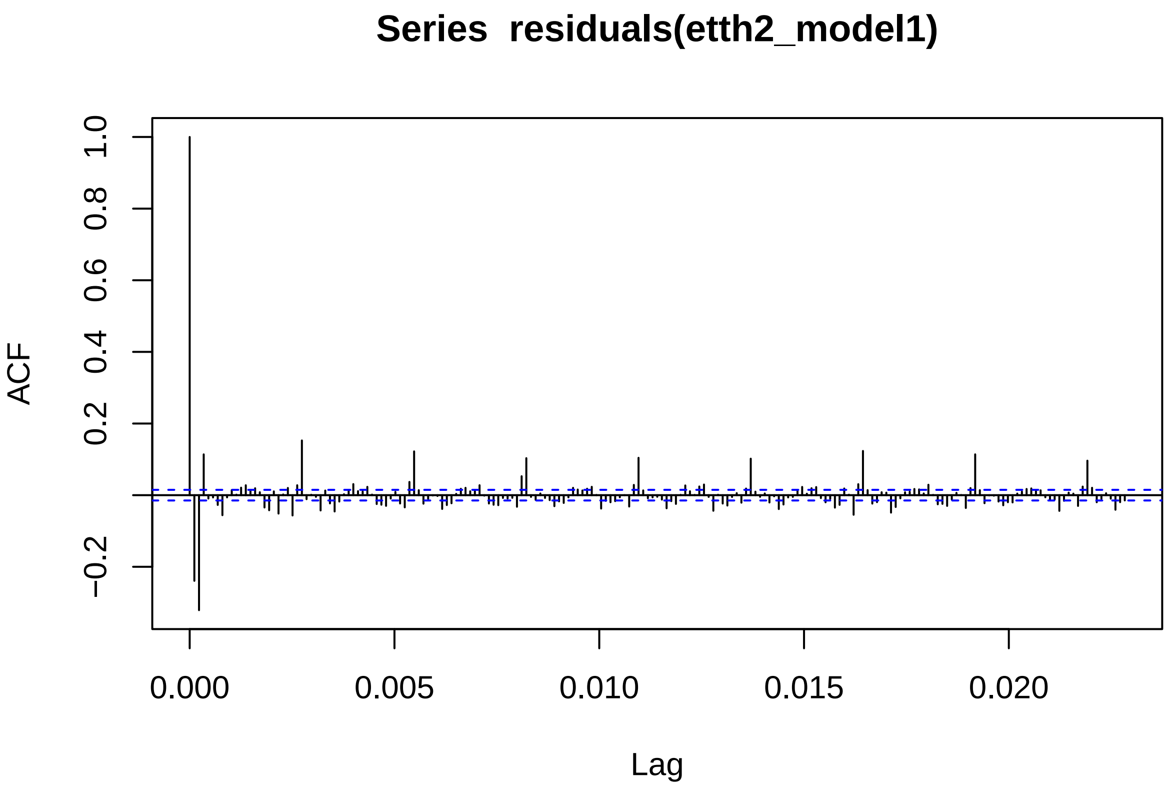

```
acf(residuals(etth2_model1), lag.max=200)
```




```
# Appears to have a seasonal component
etth2_model1 = arima(etth2.ts, order=c(1,3,0))
plot(etth2.ts)
etth2_fit1 = etth2.ts - residuals(etth2_model1)
points(etth2_fit1, type = "l", col = "lightblue", lty = 2, lwd=0.2)
```

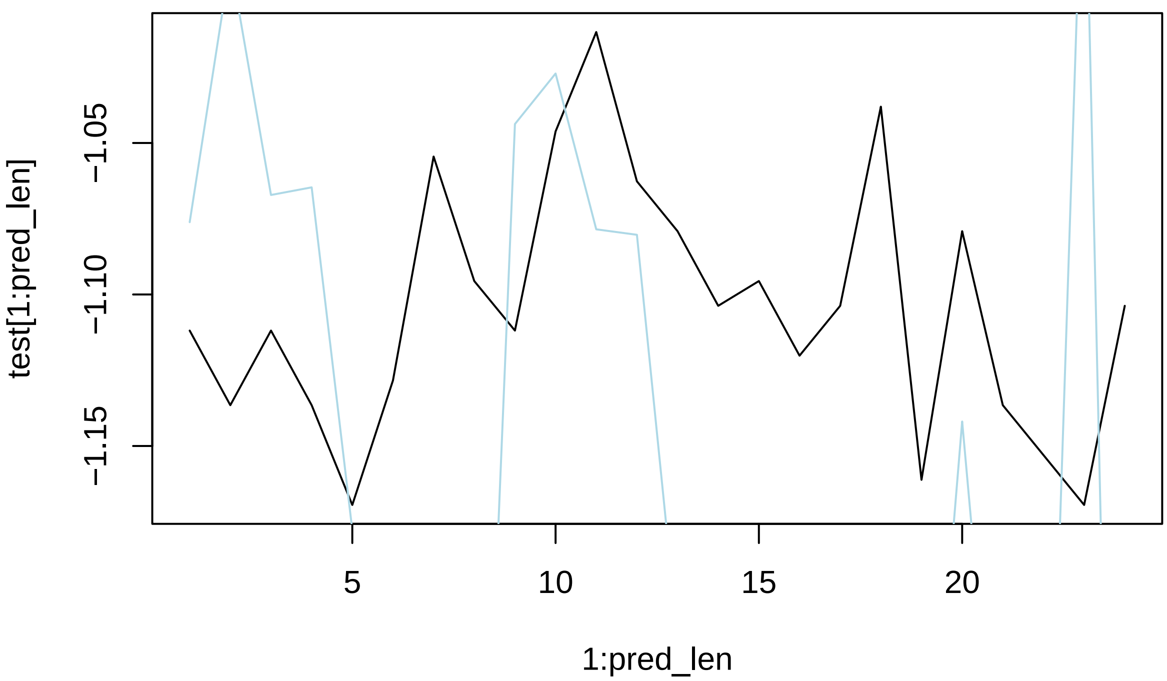


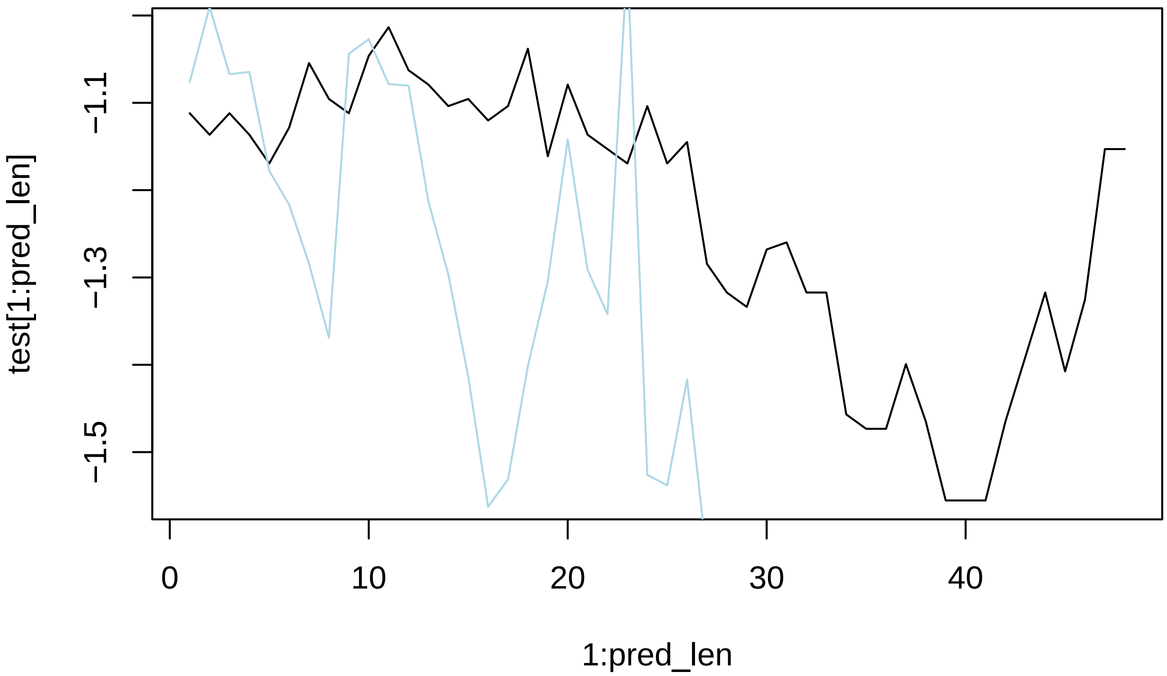

```r
acf(residuals(etth2_model1), lag.max=200)
```

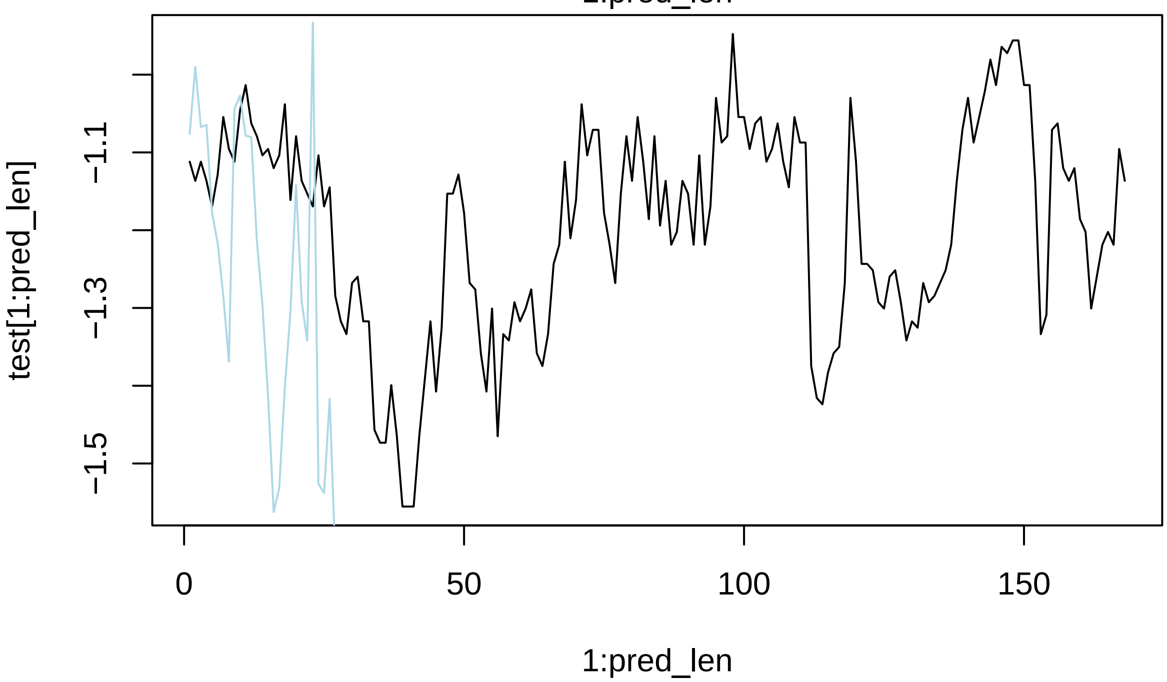


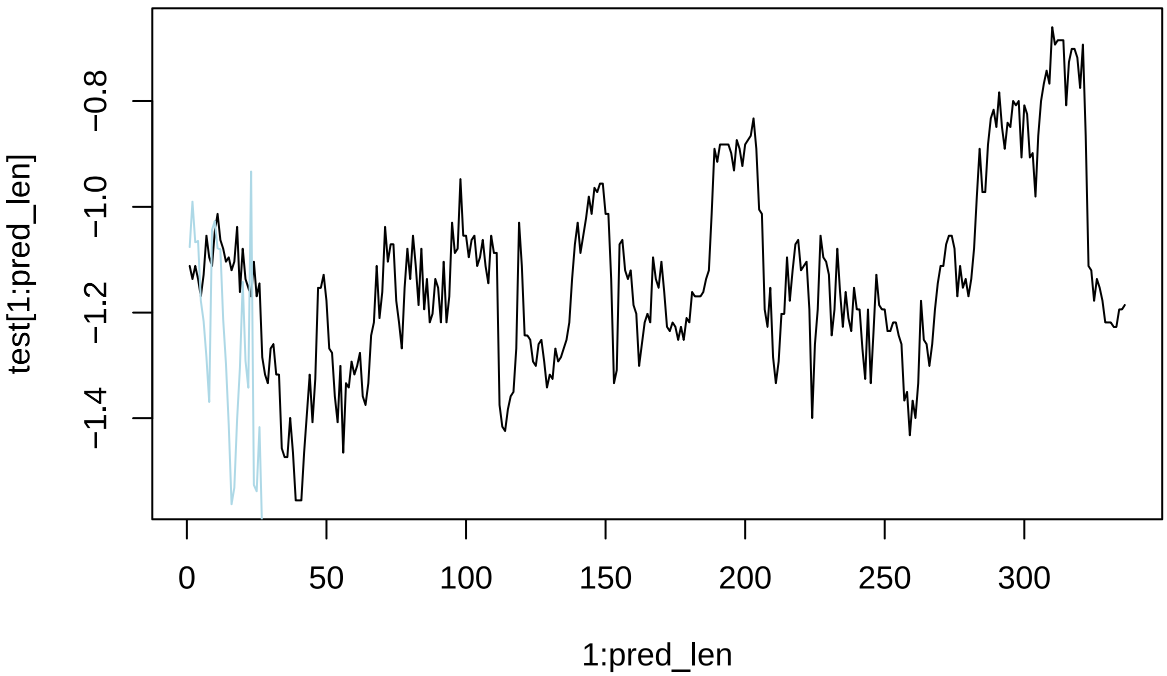

```r
# Evaluation

# Train/Validation/Test Split
get_evaluation_on_ts = function(input_ts=etth1.ts) {
  train_size = 0.8
```

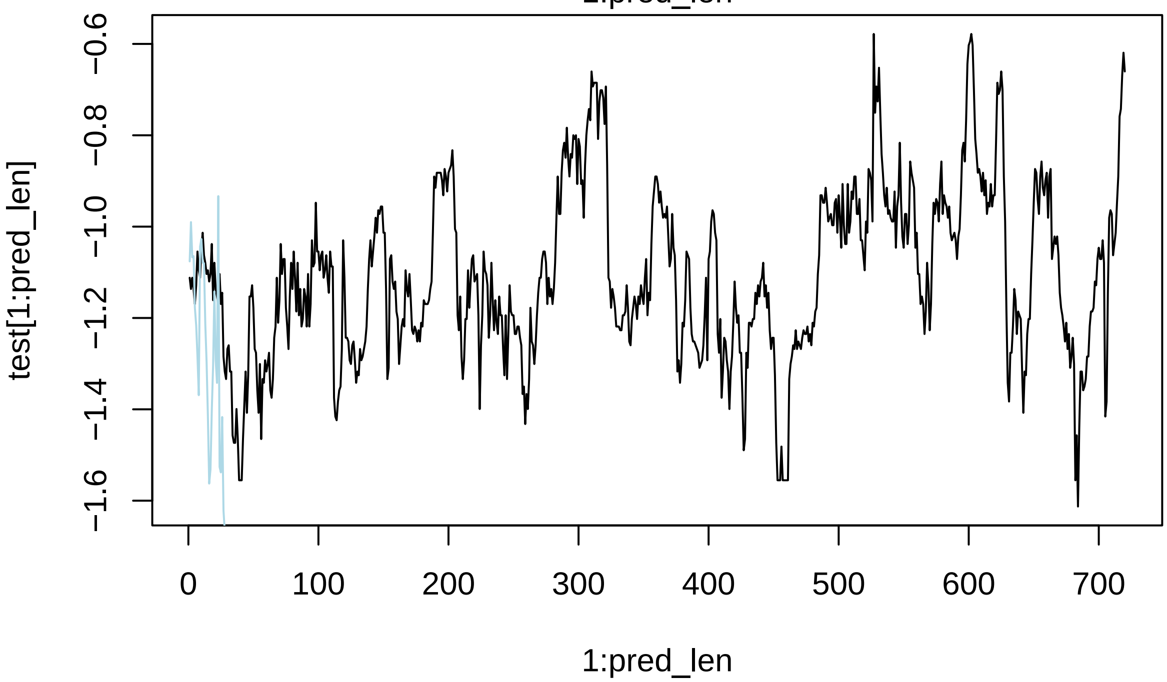
```
  test_size = 0.2
  train = input_ts[1:(train_size*length(input_ts))]
  test_start = (train_size*length(input_ts))+1
  test = input_ts[test_start:length(input_ts)]
  # may add as a parameter
  evaluated_model1 = arima(train, order = c(2,2,0), seasonal=list(order=c(0,2,0), period=24))

  res_lst = list()
  for (pred_len in c(24, 48, 168, 336, 720)) {
    curr_pred = predict(evaluated_model1, n.ahead=pred_len)$pred

    # curr_pred = curr_pred[(length(curr_pred)-(pred_len-1)):length(curr_pred)]

    plot(1:pred_len, test[1:pred_len],type="l")
    points(1:pred_len, curr_pred,type="l",col="lightblue")

    mse = mean( (test[1:pred_len]-curr_pred)^2)
    mae = mean( abs(test[1:pred_len]-curr_pred))
    res_lst[paste("mse_",pred_len,sep="")] = mse
    res_lst[paste("mae_",pred_len,sep="")] = mae
  }
  return(res_lst)
}

get_evaluation_on_ts(etth1.ts)
```




test[1:pred_len]
−1.1
−1.3
−1.5
0
10
20
30
40
1:pred_len


test[1:pred_len]
−1.1
−1.3
−1.5
0
50
100
150
1:pred_len






```
## $mse_24
## [1] 0.0483245
##
## $mae_24
## [1] 0.173118
##
## $mse_48
## [1] 0.9483854
##
## $mae_48
## [1] 0.685045
```

```
##
## $mse_168
## [1] 1132.077
##
## $mae_168
## [1] 22.81619
##
## $mse_336
## [1] 58026.52
##
## $mae_336
## [1] 161.9546
##
## $mse_720
## [1] 4901204
##
## $mae_720
## [1] 1477.134
```