Project 2 by Tiantian Zhao, Aiden Mayhood, Han Zheng, and Megan Oh

2022-10-30

```r
library(dynlm)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```r
library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ARDL)
```

```
## To cite ARDL in publications use:
##
## Kleanthis Natsiopoulos and Nickolaos G. Tzeremes (2022). "ARDL bounds test for cointegration: Replica
```

```r
library(vars)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: strucchange

## Loading required package: sandwich

## Loading required package: urca

## Loading required package: lmtest
```

```
library(dLagM)
```

```
## Loading required package: nardl

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(kableExtra)
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'

##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(fpp)
```

```
## Loading required package: forecast

##
## Attaching package: 'forecast'

## The following object is masked from 'package:dLagM':
##
##     forecast

## Loading required package: fma
```

```
##
## Attaching package: 'fma'

## The following objects are masked from 'package:MASS':
##
##     cement, housing, petrol

## Loading required package: expsmooth

## Loading required package: tseries
```

```r
library(tseries)
library(dynlm)
library(forecast)
library(reshape2)
library(readxl)
library(ggplot2)
climate <- read.csv("DailyDelhiClimateTrain.csv")
```

Question 1a

The data set consists of data from the 1st of January of 2013 until the 1st of January in 2017 in the city of Delhi, India. The data set is called 'DailyDelhiClimateTrain' and comes from Kaggle. The data set includes dates, mean temperature, humidity, wind speed, and mean pressure during this time period. Mean temperature gives an average of temperature for a particular day, and mean pressure gives an average of pressure for a particular day. The data was collected from Weather Underground API and was used meant for an exercise at PES University in Bangalore, India. The observations collected in the data are time series, as it consists of one location, Delhi, with data on weather across multiple periods of time in days. We have read the data into R and have ran libraries for certain functions that will be used in this project. Looking at the data using the is.na function, there appears to be no missing values in the data.

```r
library(dplyr)  # For manipulating data
library(tidyr)  # For making data long and wide
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths
```

```r
library(DBI)  # For database connections

dir.create("climate", showWarnings = FALSE)

con <- dbConnect(RSQLite::SQLite(), "C:\\Users\\zhao1\\OneDrive\\Desktop\\GO\\UCLA\\2022 F\\Econ 104\\Da

is.na(climate)
```

```
##         date meantemp humidity wind_speed meanpressure
##    [1,] FALSE    FALSE    FALSE      FALSE        FALSE
##    [2,] FALSE    FALSE    FALSE      FALSE        FALSE
##    [3,] FALSE    FALSE    FALSE      FALSE        FALSE
##    [4,] FALSE    FALSE    FALSE      FALSE        FALSE
##    [5,] FALSE    FALSE    FALSE      FALSE        FALSE
##    [6,] FALSE    FALSE    FALSE      FALSE        FALSE
##    [7,] FALSE    FALSE    FALSE      FALSE        FALSE
##    [8,] FALSE    FALSE    FALSE      FALSE        FALSE
##    [9,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [10,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [11,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [12,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [13,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [14,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [15,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [16,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [17,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [18,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [19,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [20,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [21,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [22,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [23,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [24,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [25,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [26,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [27,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [28,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [29,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [30,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [31,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [32,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [33,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [34,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [35,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [36,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [37,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [38,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [39,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [40,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [41,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [42,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [43,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [44,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [45,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [46,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [47,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [48,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [49,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [50,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [51,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [52,] FALSE    FALSE    FALSE      FALSE        FALSE
##   [53,] FALSE    FALSE    FALSE      FALSE        FALSE
```

```
## [1458,] FALSE    FALSE    FALSE      FALSE         FALSE
## [1459,] FALSE    FALSE    FALSE      FALSE         FALSE
## [1460,] FALSE    FALSE    FALSE      FALSE         FALSE
## [1461,] FALSE    FALSE    FALSE      FALSE         FALSE
## [1462,] FALSE    FALSE    FALSE      FALSE         FALSE
```

```r
summary(climate)
```

```
##      date              meantemp        humidity        wind_speed
##  Length:1462        Min.   : 6.00   Min.   : 13.43   Min.   : 0.000
##  Class :character   1st Qu.:18.86   1st Qu.: 50.38   1st Qu.: 3.475
##  Mode  :character   Median :27.71   Median : 62.62   Median : 6.222
##                     Mean   :25.50   Mean   : 60.77   Mean   : 6.802
##                     3rd Qu.:31.31   3rd Qu.: 72.22   3rd Qu.: 9.238
##                     Max.   :38.71   Max.   :100.00   Max.   :42.220
##   meanpressure
##  Min.   :  -3.042
##  1st Qu.:1001.580
##  Median :1008.563
##  Mean   :1011.105
##  3rd Qu.:1014.945
##  Max.   :7679.333
```

Upon research of pressure in weather, the highest ever recorded pressure was 1081.8 hPa, and the lowest ever recorded pressure was 870 hPa in a western Pacific Ocean typhoon. Therefore, we have set our range of mean pressure to be between 950 and 1081.8, as there were no significant weather events in India during our relevant period that would have caused pressure to drop below 950 or reach above 1081.8. We have replaced these outlier values with the value of the next observation in the data set. In total, 9 outliers were replaced. Now that the outliers have been removed, we can see that there is significant negative correlation between mean pressure and mean temperature.

```r
mp1 <- replace(climate$meanpressure, climate$meanpressure < 950,
    NA)
mp2 <- replace(mp1, mp1 > 1200, NA)
mp <- na.locf(mp2)
```

Below, we created time series vectors of mean temperature, humidity, wind speed, and mean pressure. We have tested for any sort of correlation between the predictors and do not believe there is any correlation between the predictors. To note, there is the potential for small negative correlation between wind speed and humidity, but this is at most a weak, negative association. Even though the correlation between mean pressure and mean temperature is low, this was because mean pressure was riddled with outliers that significantly affected the data. Once removing the outliers, the correlation became very significant.

Next, we plotted humidity against mean temperature, wind speed against mean temperature, and mean pressure against mean temperature. Humidity, wind speed, and mean temperature will be contributing factors in the overall result of mean temperature for a particular period, as mean temperature will vary depending on how much humidity there is, if there is wind, and what the value of mean pressure is.

The humidity vs. mean temperature scatterplot shows a strong, negative, linear assocation between humidity and mean temperature. There appears to be no outliers in the data based on visual analysis of the plot.

The wind speed vs. mean temperature scatterplot shows a non-associated relationship between wind speed and mean temperature. Although the line of best shows a positive linear trend, the plot doesn't really indicate any sort of relationship. There may be a few outliers present by visual analysis, off to the right of the scatterplot away from the main cluster of points on the left.

The humidity vs. mean temperature scatterplot indicates that there is a few outliers that are significantly skewing the scatterplot. Most of the data falls within a particular range, but 9 data points look like they are clear errors. These outliers will most likely need to be addressed.

The temperature vs. time plot shows seasonal trends in temperature. There is a slight increase in the mean of mean temperature over time.

The humidity vs. time plot shows seasonal trends in humidity. There is a slight decrease in the mean of humidity over time.

The wind speed vs. time plot shows seasonal trends in wind speed. There is no discernable change in the mean of wind speed over time.

The mean pressure vs. mean temperature scatterplot shows a strong, negative, linear assocation between mean pressure and mean temperature. There appears to be no outliers in the data based on visual analysis of the plot, since we removed these outliers.

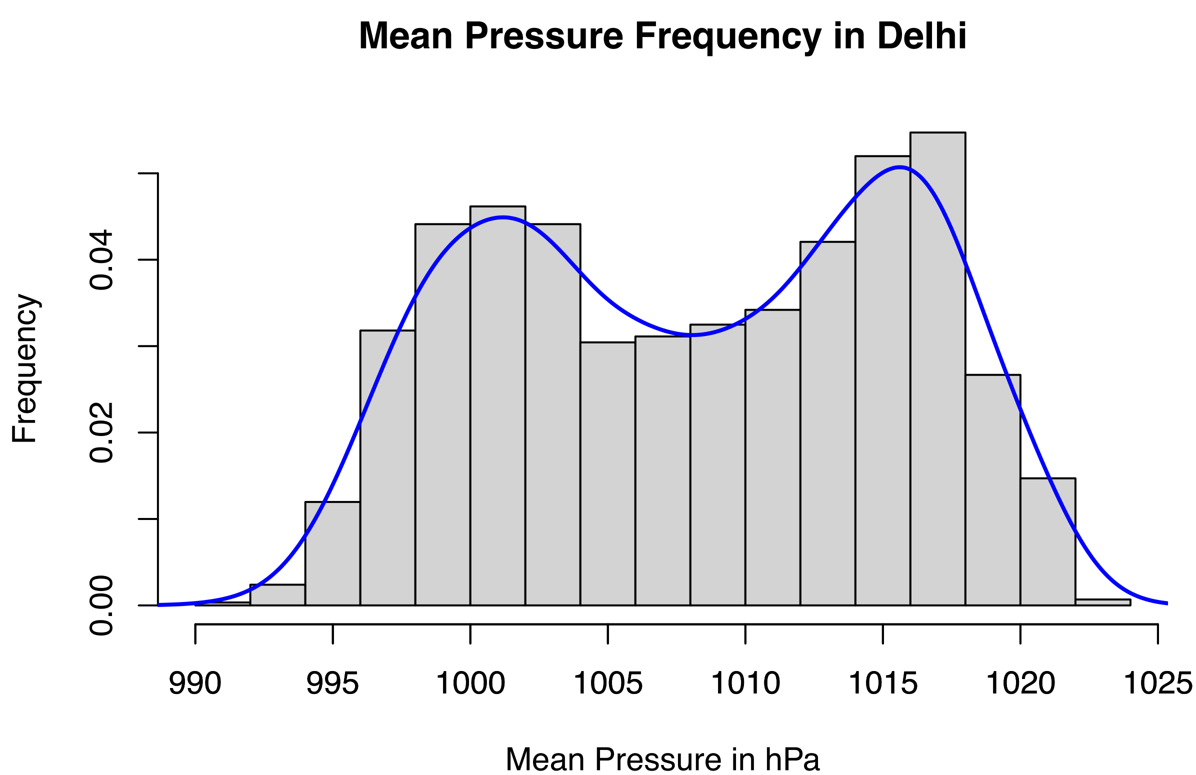
Then, we created histograms of mean temperature, humidity, wind speed, and mean pressure.

The distribution of the mean temperature histogram appears to be bimodal with a potential left skew. The highest frequency of observations is between the 30-32 degrees celcius range.

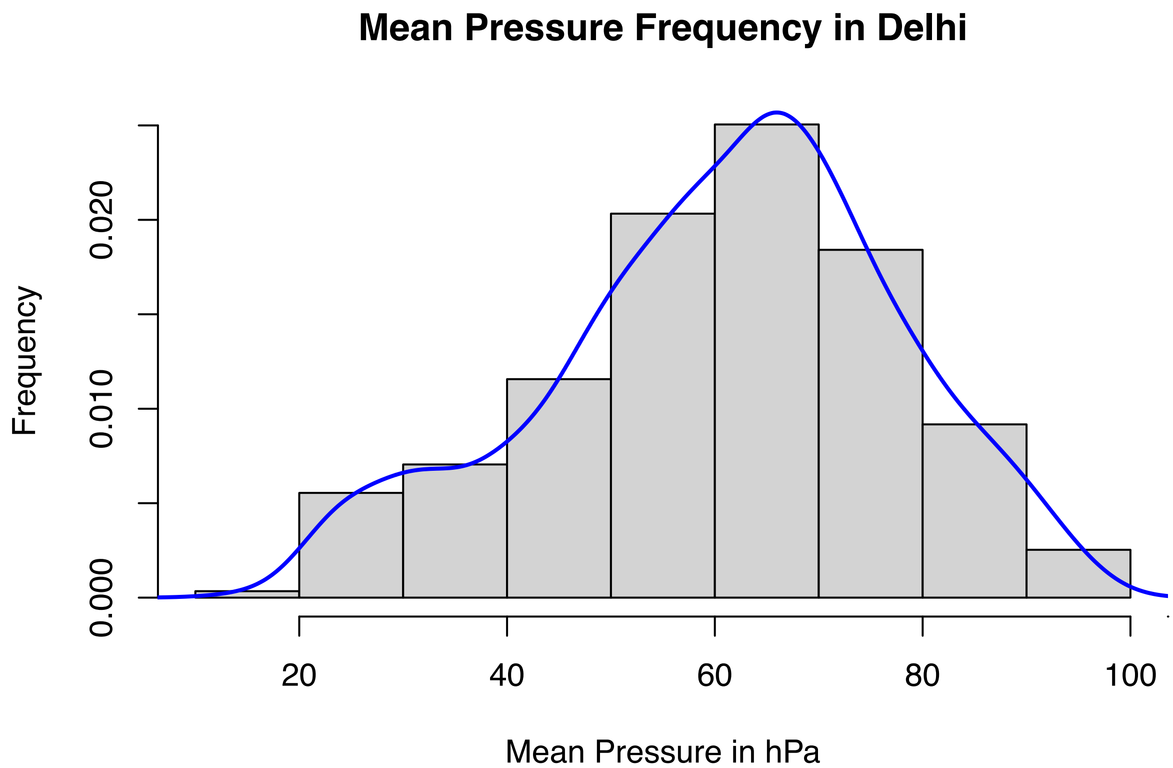
The distribution of the humidity histogram appears to be bell-shaped with no real skew. The highest frequency of observations is between the 60-70% range.

The distribution of the wind speed histogram appears to be right-skewed. This could be a result of potential outliers in the high wind speed range. The highest frequency of observations is the 5-10 kilometers per hour range.

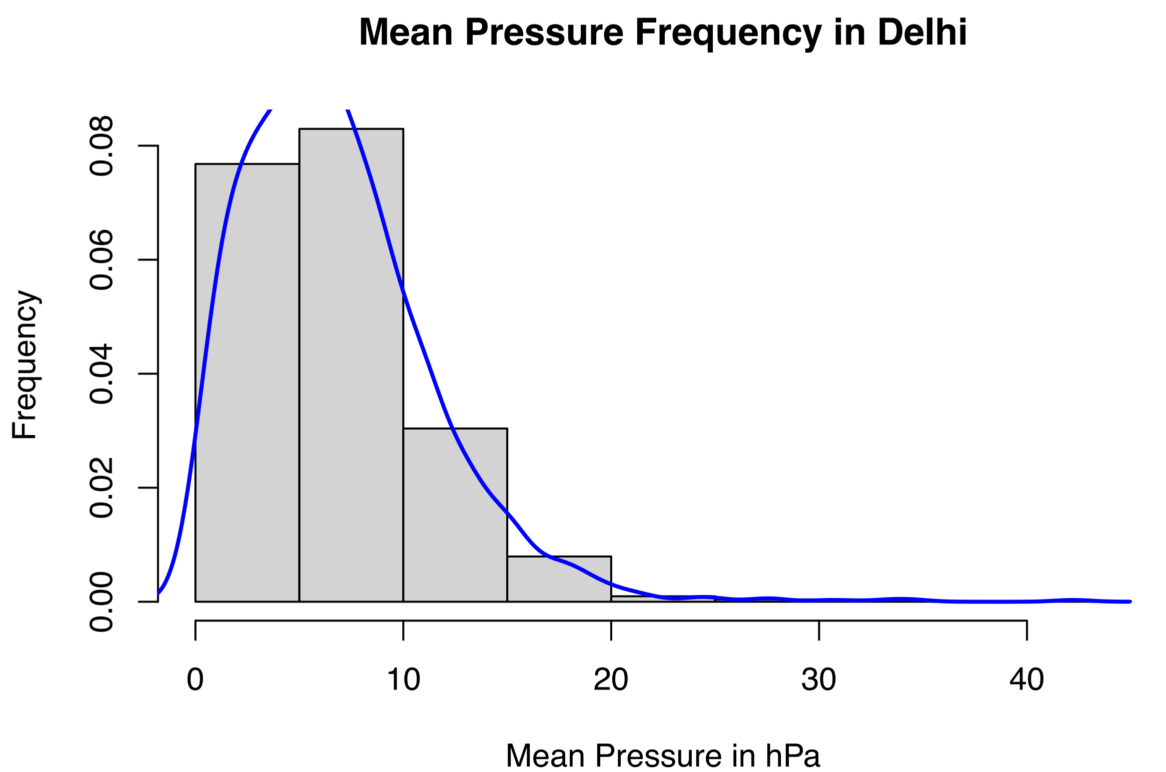
```r
mtts = ts(climate$meantemp, start = c(2013 - 1 - 1), end = c(2017 -
    1 - 1), frequency = 365)
hts = ts(climate$humidity, start = c(2013 - 1 - 1), end = c(2017 -
    1 - 1), frequency = 365)
wsts = ts(climate$wind_speed, start = c(2013 - 1 - 1), end = c(2017 -
    1 - 1), frequency = 365)
mpts = ts(mp, start = c(2013 - 1 - 1), end = c(2017 - 1 - 1),
    frequency = 365)

hist(mp, freq = FALSE, xlab = "Mean Pressure in hPa", ylab = "Frequency",
    main = "Mean Pressure Frequency in Delhi")

lines(density(mp), lwd = 2, col = "blue")
axis(1, at = seq(990, 1025, by = 5))
```




```r
hist(hts, freq = FALSE, xlab = "Mean Pressure in hPa", ylab = "Frequency",
    main = "Mean Pressure Frequency in Delhi")

lines(density(hts), lwd = 2, col = "blue")
axis(1, at = seq(990, 1025, by = 5))
```



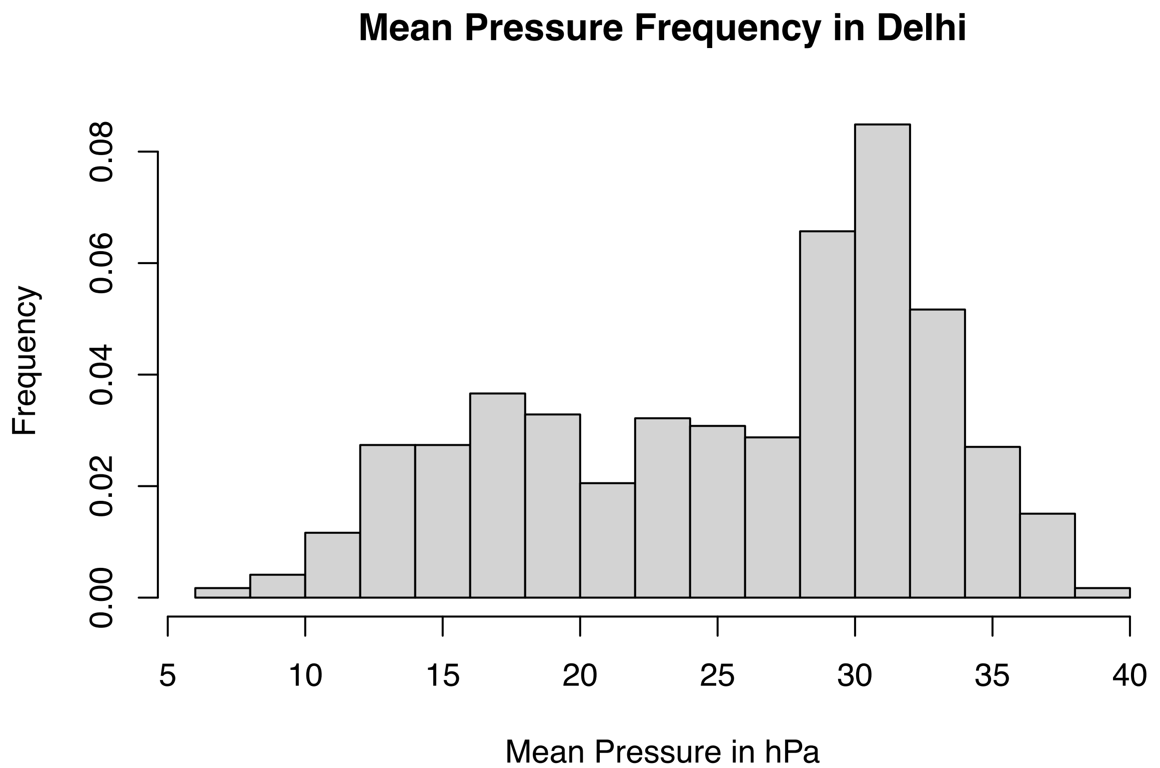

```
hist(wsts, freq = FALSE, xlab = "Mean Pressure in hPa", ylab = "Frequency",
    main = "Mean Pressure Frequency in Delhi")

lines(density(wsts), lwd = 2, col = "blue")
axis(1, at = seq(990, 1025, by = 5))
```



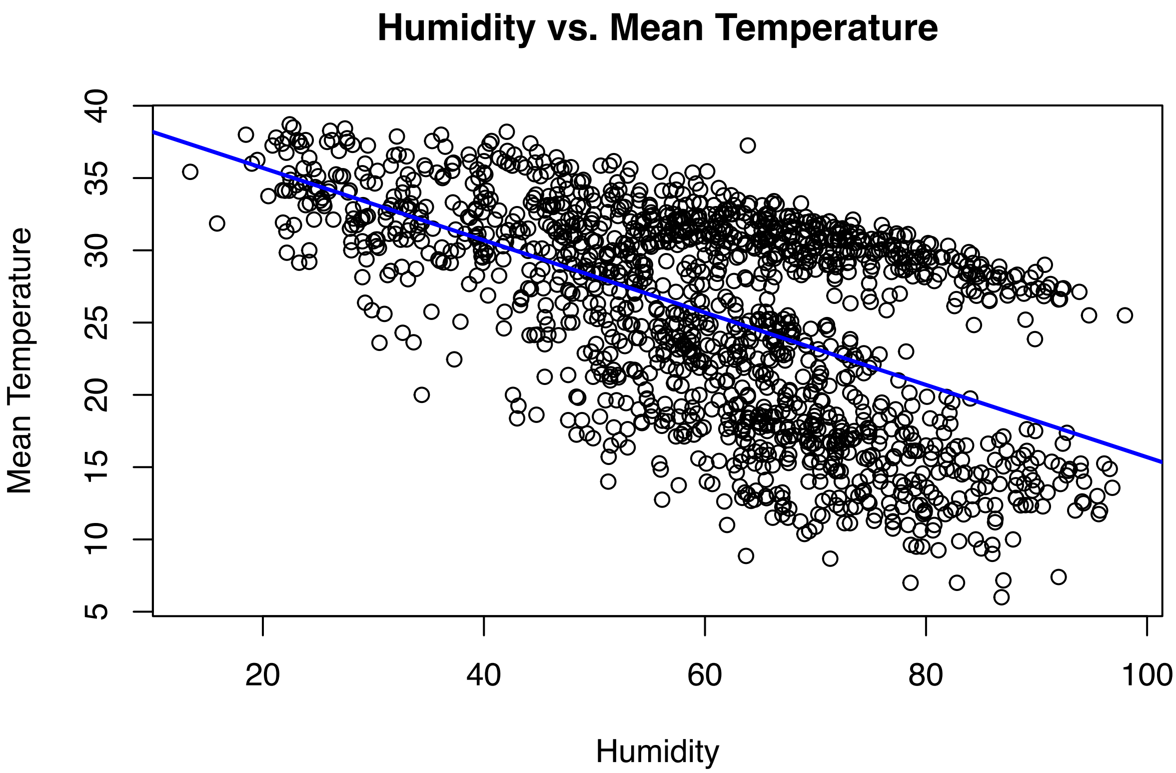

```r
hist(mtts, freq = FALSE, xlab = "Mean Pressure in hPa", ylab = "Frequency",
    main = "Mean Pressure Frequency in Delhi")

lines(density(nmtts), lwd = 2, col = "blue")
```

```
## Error in density(nmtts): object 'nmtts' not found
```

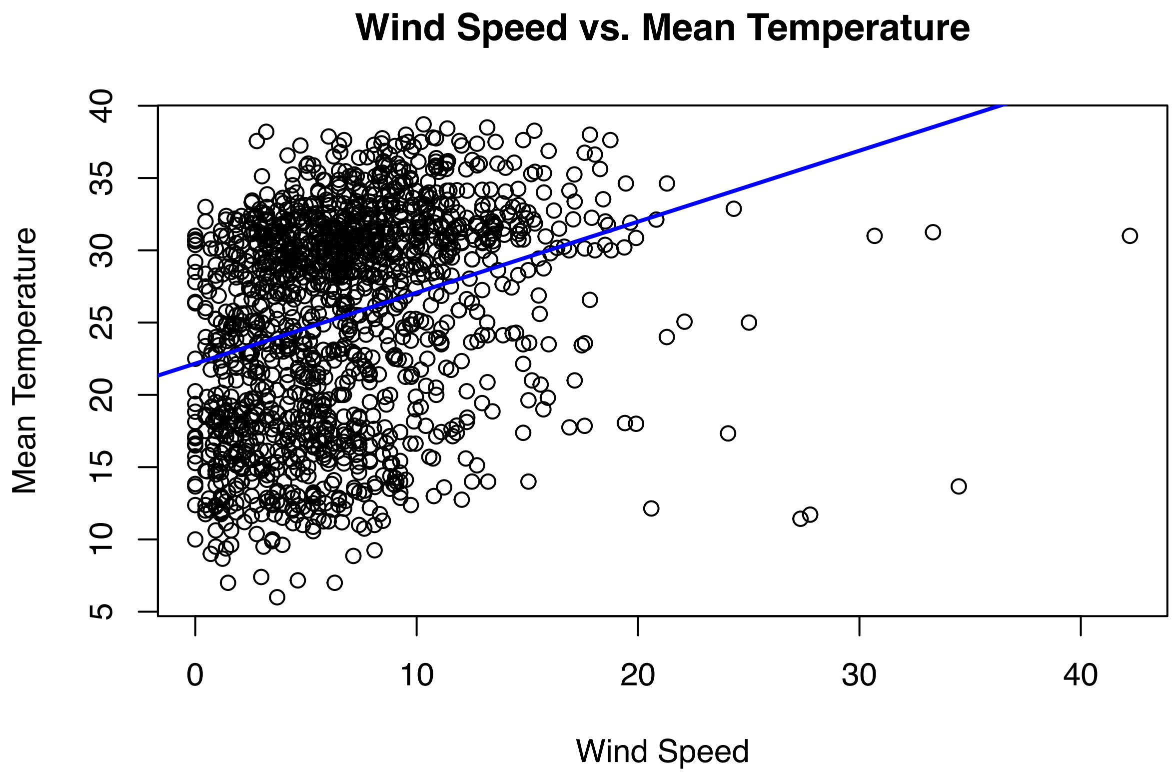
```r
axis(1, at = seq(990, 1025, by = 5))
```



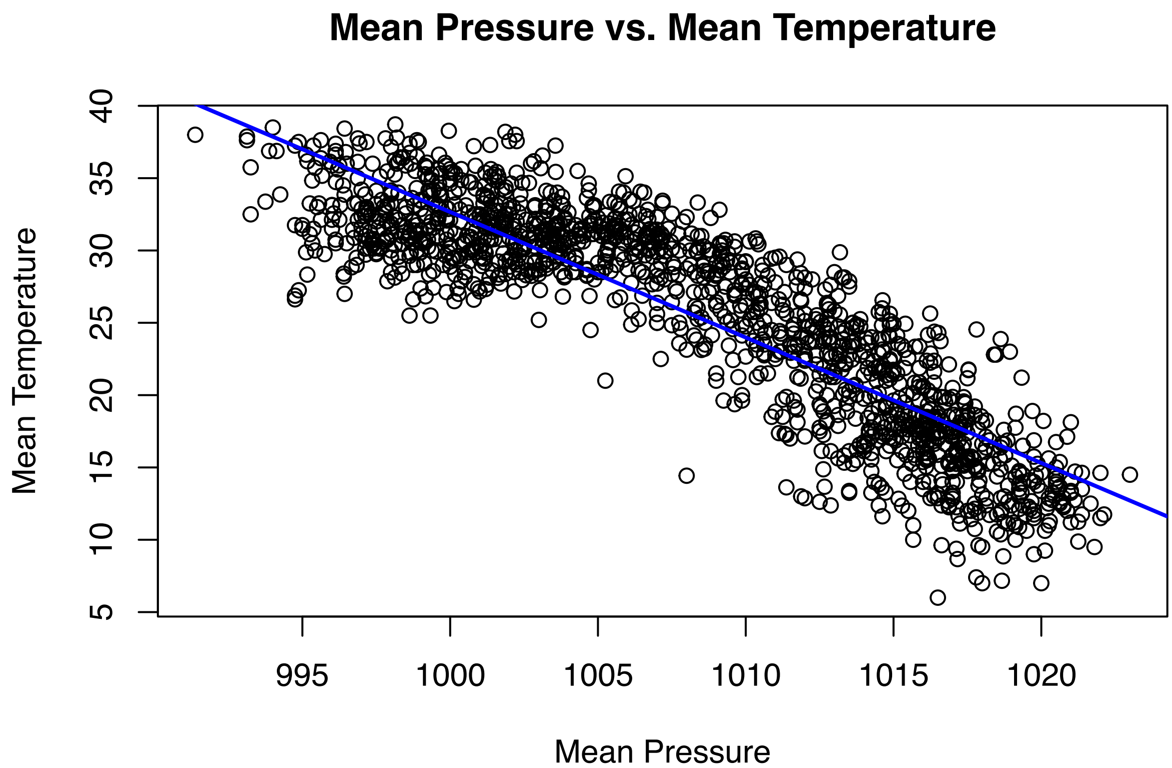

```r
reg.mod1 = lm(mtts ~ hts)
plot(hts, mtts, xlab = "Humidity", ylab = "Mean Temperature",
    main = "Humidity vs. Mean Temperature")
abline(reg.mod1, lwd = 2, col = "blue")
```



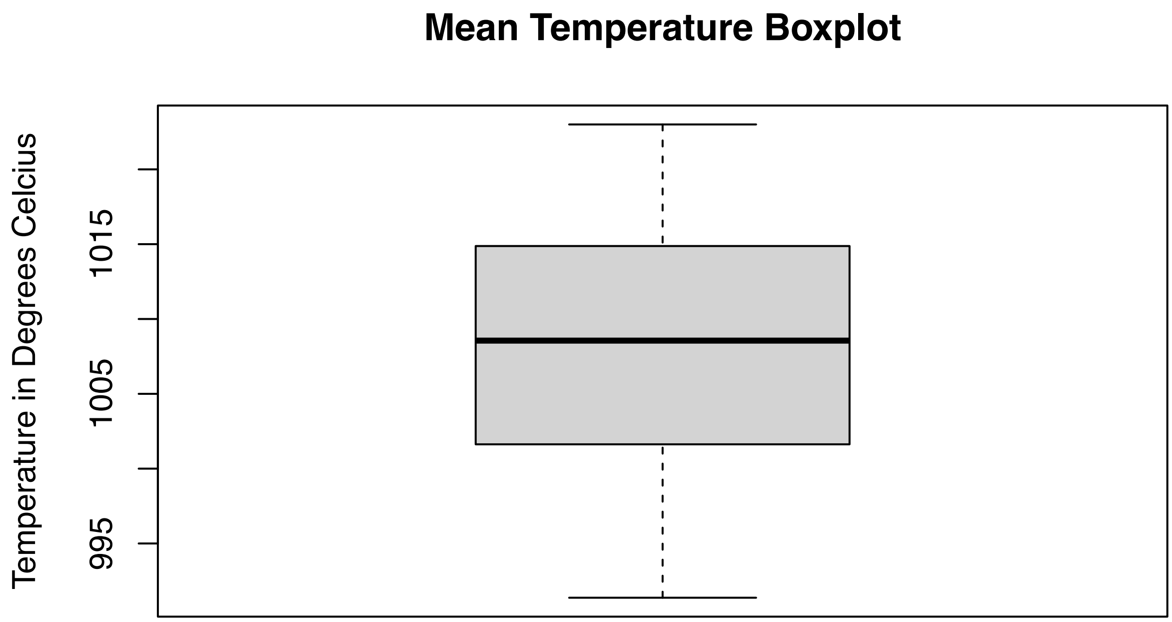

```
reg.mod2 = lm(mtts ~ wsts)
plot(wsts, mtts, xlab = "Wind Speed", ylab = "Mean Temperature",
    main = "Wind Speed vs. Mean Temperature")
abline(reg.mod2, lwd = 2, col = "blue")
```




```
reg.mod3 = lm(mtts ~ mpts)
plot(mpts, mtts, xlab = "Mean Pressure", ylab = "Mean Temperature",
    main = "Mean Pressure vs. Mean Temperature")
abline(reg.mod3, lwd = 2, col = "blue")
```



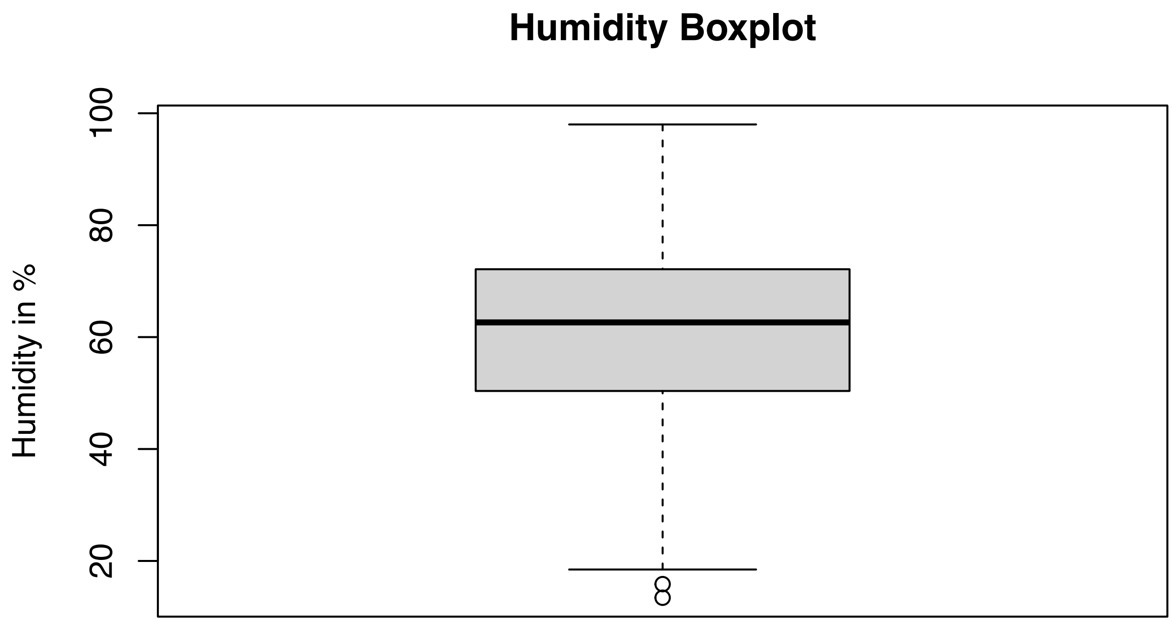

```
boxplot(mpts, main = "Mean Temperature Boxplot", ylab = "Temperature in Degrees Celcius")
```




```
summary(mpts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   991.4  1001.6  1008.6  1008.2  1014.9  1023.0
```

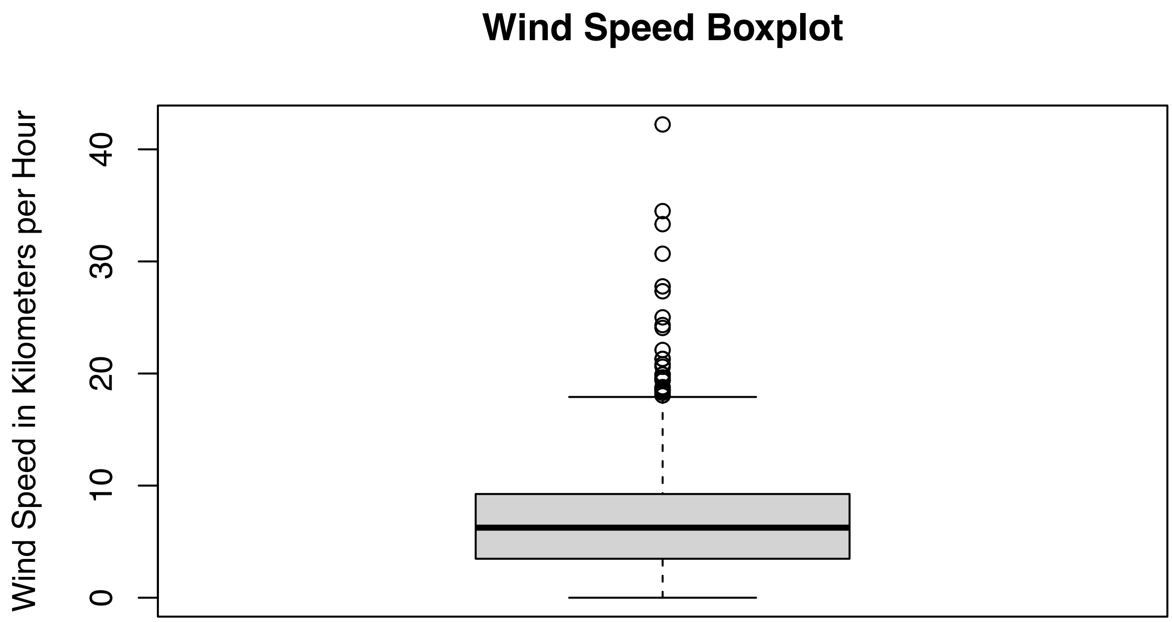
```
boxplot(hts, main = "Humidity Boxplot", ylab = "Humidity in %")
```




```
summary(hts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.43   50.38   62.62   60.74   72.12   98.00
```

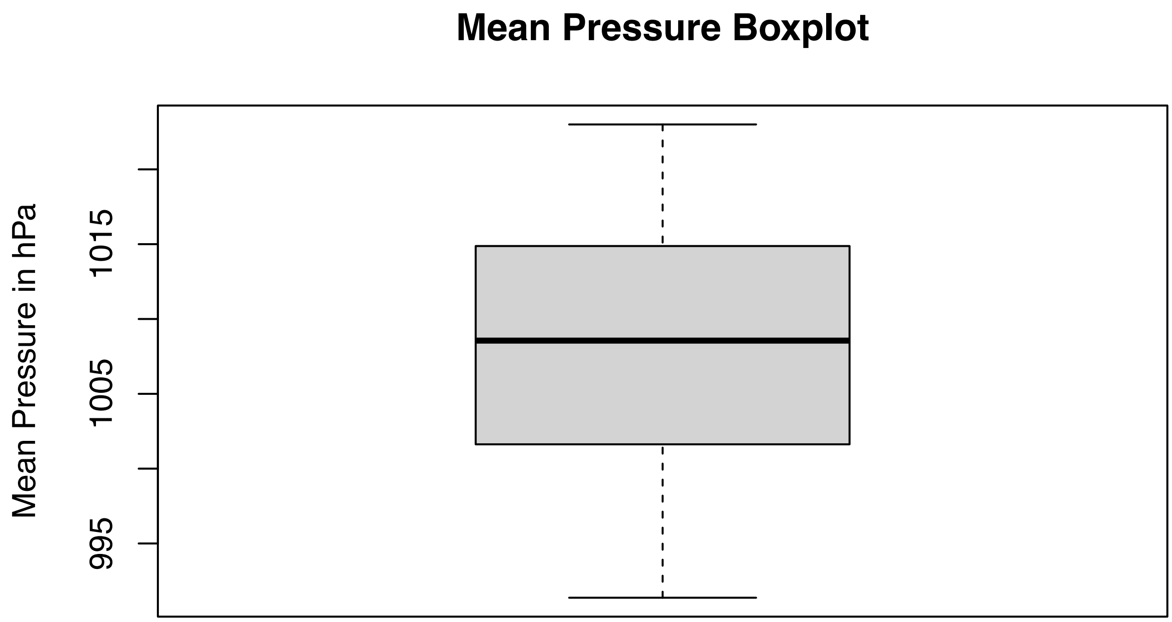
```
boxplot(wsts, main = "Wind Speed Boxplot", ylab = "Wind Speed in Kilometers per Hour")
```




```
summary(wsts)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.000    3.475    6.250    6.807    9.250   42.220
```

```
boxplot(mpts, main = "Mean Pressure Boxplot", ylab = "Mean Pressure in hPa")
```




```
summary(mpts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   991.4  1001.6  1008.6  1008.2  1014.9  1023.0
```

```
cor.test(hts, wsts)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hts and wsts
## t = -15.335, df = 1459, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4159027 -0.3275381
## sample estimates:
##        cor
## -0.3725646
```

```
cor.test(hts, mpts)
```

```
##
##  Pearson's product-moment correlation
##
```

```
## data:  hts and mpts
## t = 13.234, df = 1459, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2808101 0.3724122
## sample estimates:
##       cor
## 0.3273801
```

```
cor.test(wsts, mpts)
```

```
##
##  Pearson's product-moment correlation
##
## data:  wsts and mpts
## t = -11.675, df = 1459, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3385139 -0.2446871
## sample estimates:
##        cor
## -0.2923038
```

```
cor.test(mpts, mtts)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mpts and mtts
## t = -70.424, df = 1459, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8901824 -0.8668199
## sample estimates:
##        cor
## -0.8790278
```

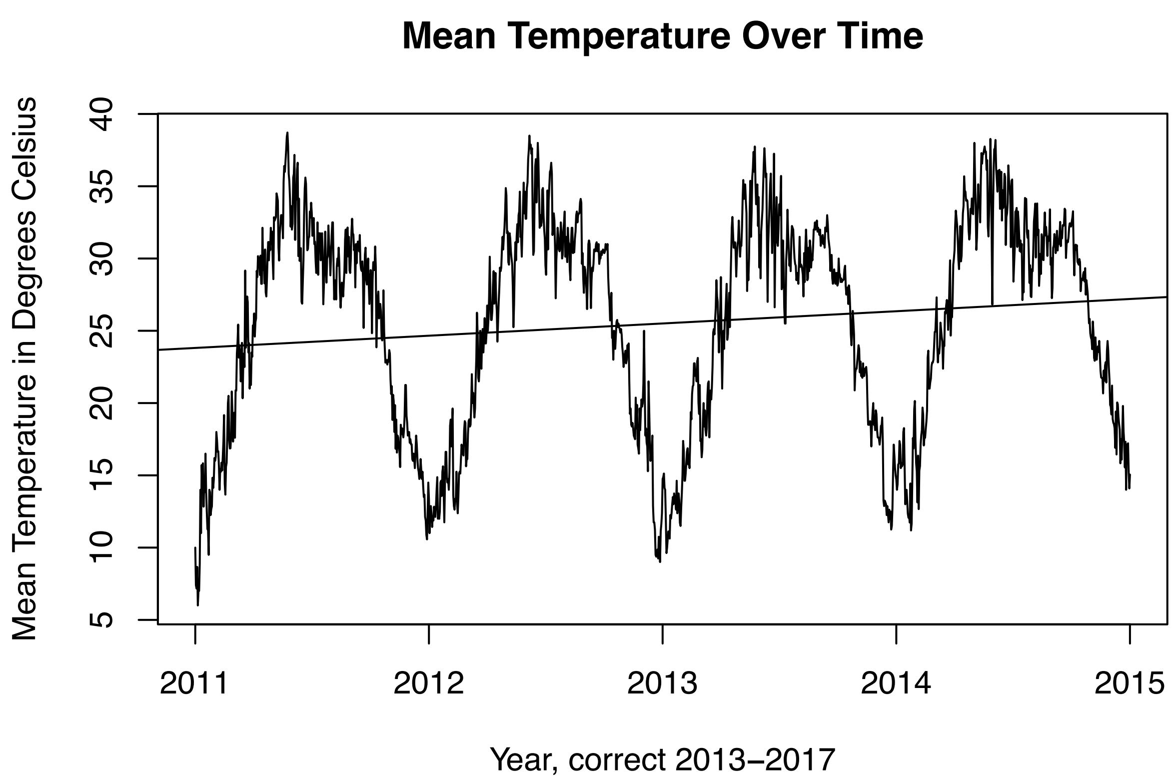
```
cor.test(hts, mtts)
```

```
##
##  Pearson's product-moment correlation
##
## data:  hts and mtts
## t = -26.533, df = 1459, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6041191 -0.5348749
## sample estimates:
##      cor
## -0.57051
```

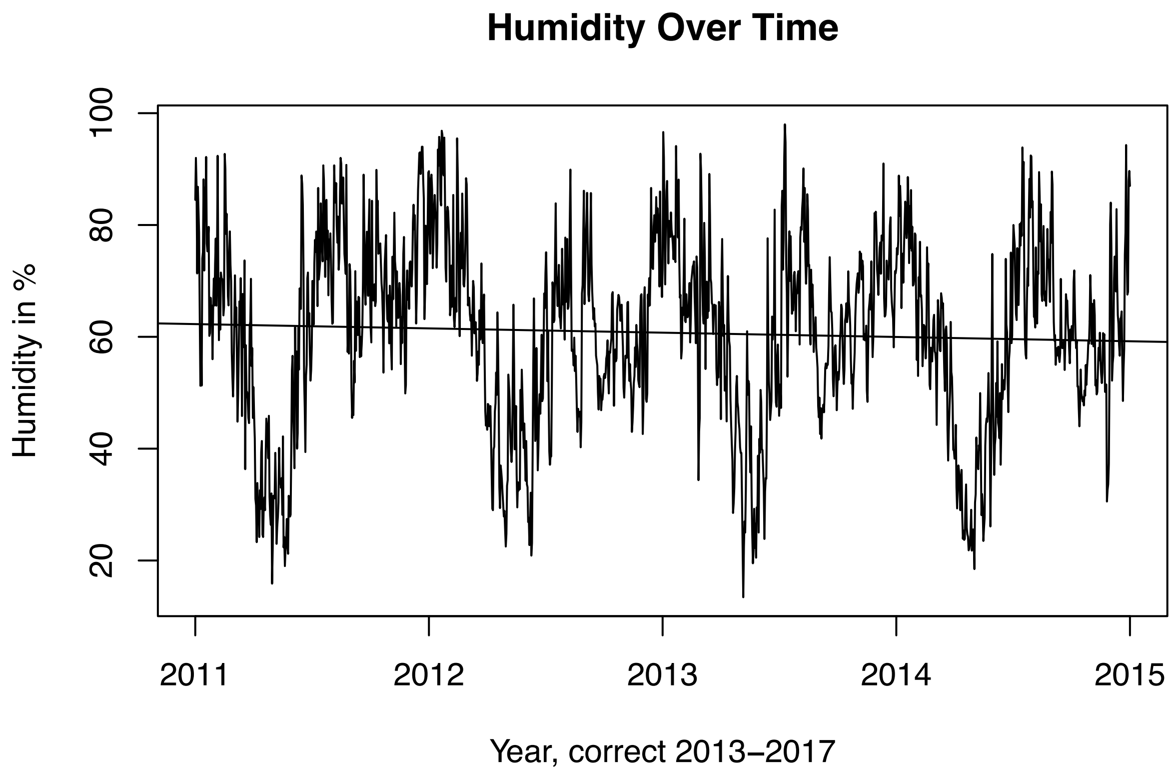
```
cor.test(wsts, mtts)
```

```
##
##  Pearson's product-moment correlation
##
## data:  wsts and mtts
## t = 12.233, df = 1459, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.257758 0.350808
## sample estimates:
##       cor
## 0.3050108
```

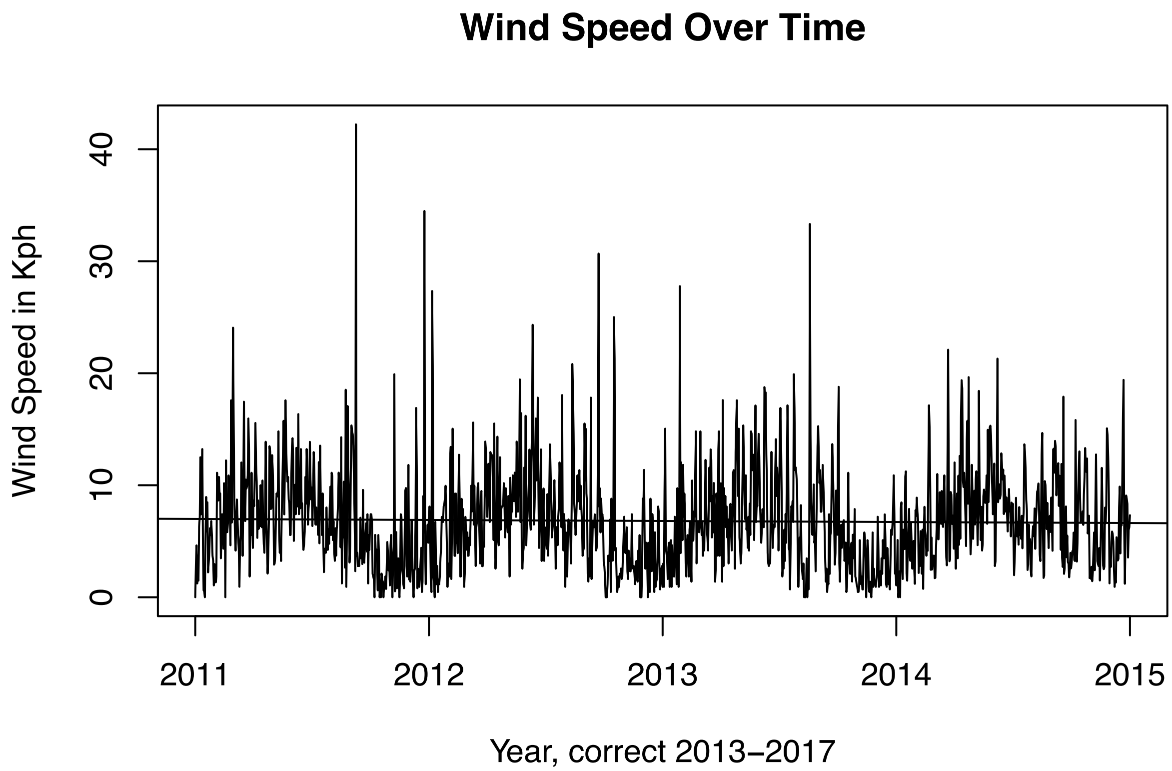
```
ts.plot(mtts, xlab = "Year, correct 2013-2017", ylab = "Mean Temperature in Degrees Celsius",
    main = "Mean Temperature Over Time")

abline(reg = lm(mtts ~ time(mtts)))
```



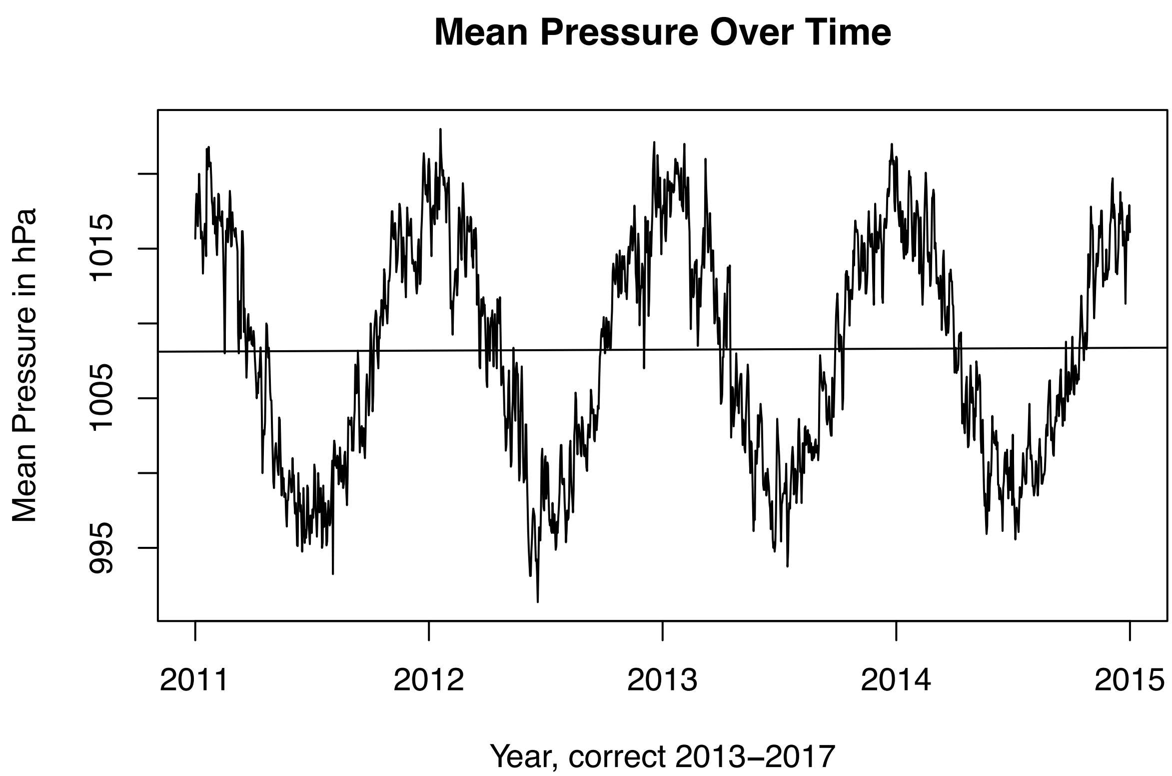

```
ts.plot(hts, xlab = "Year, correct 2013-2017", ylab = "Humidity in %",
    main = "Humidity Over Time")

abline(reg = lm(hts ~ time(hts)))
```



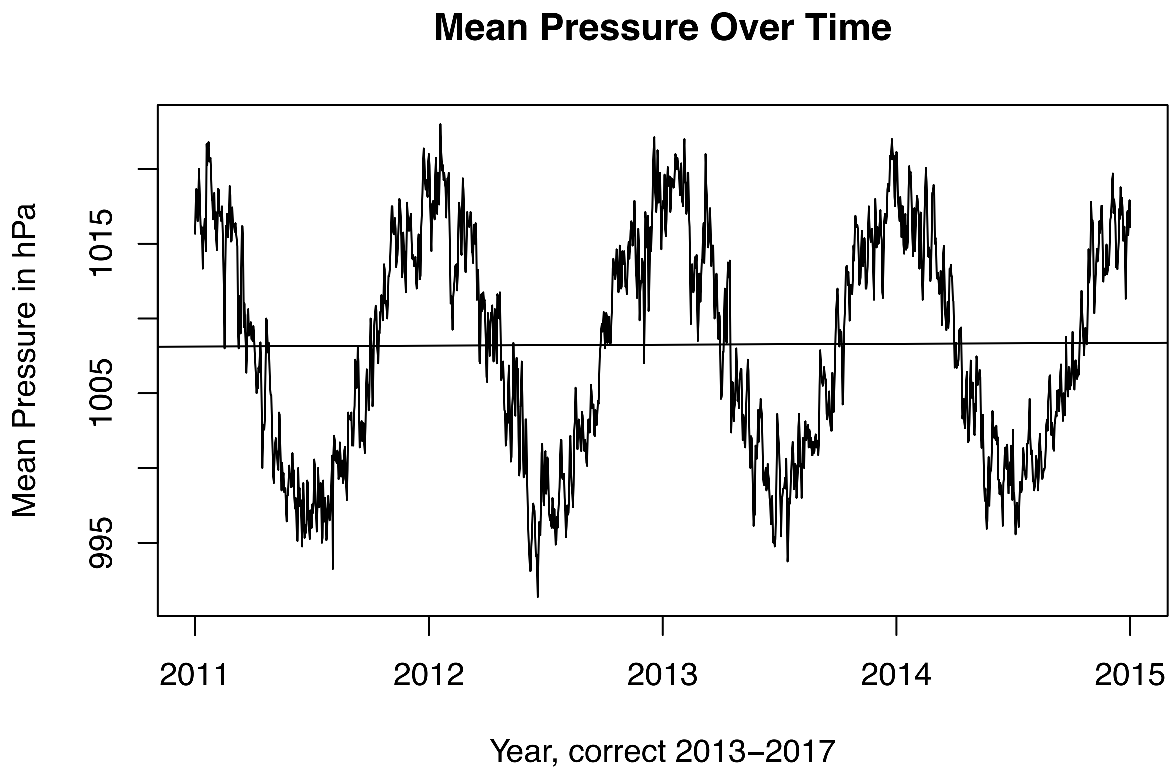

```
ts.plot(wsts, xlab = "Year, correct 2013-2017", ylab = "Wind Speed in Kph",
    main = "Wind Speed Over Time")

abline(reg = lm(wsts ~ time(wsts)))
```




```r
ts.plot(mpts, xlab = "Year, correct 2013-2017", ylab = "Mean Pressure in hPa",
    main = "Mean Pressure Over Time")

abline(reg = lm(mpts ~ time(mpts)))
```




```r
ts.plot(mpts, xlab = "Year, correct 2013-2017", ylab = "Mean Pressure in hPa",
    main = "Mean Pressure Over Time")

abline(reg = lm(mpts ~ time(mpts)))
```


Mean Pressure Over Time
Mean Pressure in hPa
995
1005
1015
2011
2012
2013
2014
2015
Year, correct 2013–2017

Question 2a

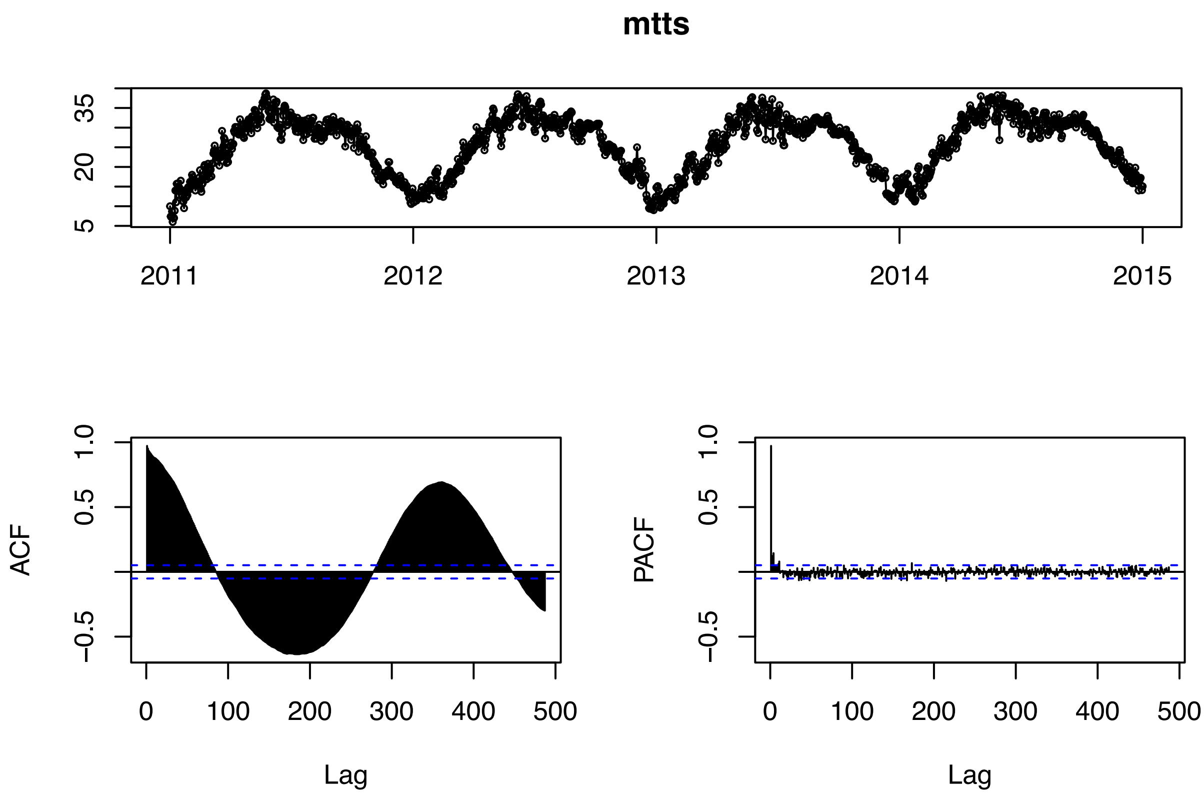
Based on tsdisplay of mean temperature, there is mean-reverting behavior. However, there seems to be seasonality based on the graph. There doesn't appear to be any trends. Intuitively, it would make sense that temperature would change with the seasons of the year. With our x-axis being time, temperature seems to be cyclical based on seasons of the year. The ACF correlogram indicates that the lags decline linearly rather than geometrically. The PACF correlogram shows decreasing autocorrelation as one gets to later lags. This resembles a possible AR model. Using the augmented Dickey-Fuller test, the p-value is greater than our alpha value of 0.05. This means that we must reject the null hypothesis, and therefore, there appears to be non-stationarity.

Based on tsdisplay of humidity, there is mean-reverting behavior. However, there seems to be seasonality based on the graph. There doesn't appear to be any trends. Intuitively, it would make sense that humidity would change with the seasons of the year. With our x-axis being time, humidity seems to be somewhat cyclical based on seasons of the year. The ACF correlogram indicates that the lags decline linearly rather than geometrically. The PACF correlogram shows decreasing autocorrelation as one gets to later lags. This resembles a possible AR model. Using the augmented Dickey-Fuller test, the p-value is less than our alpha value of 0.05. This means that we will accept the null hypothesis, and therefore, the test tells us there is stationarity.

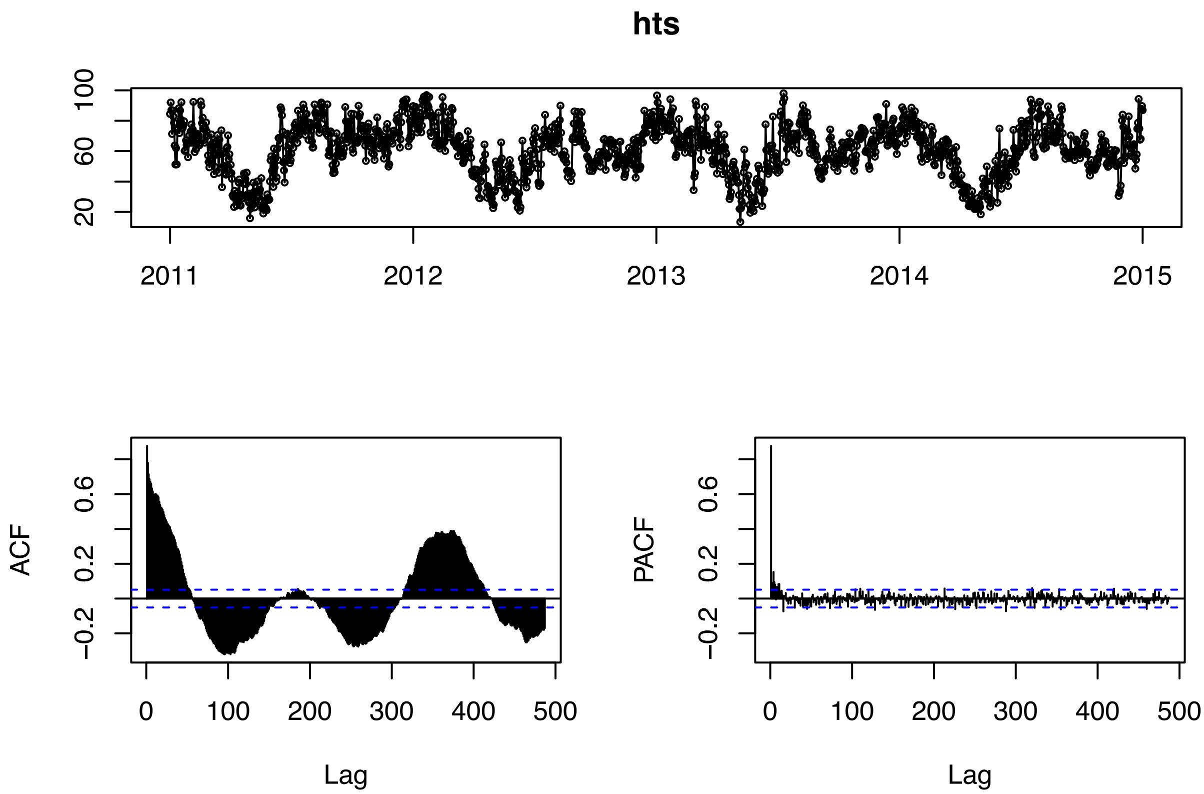
Based on tsdisplay of wind speed, there is mean-reverting behavior. However, there seems to be slight seasonality based on the graph. There doesn't appear to be any trends. Intuitively, it would make sense that wind speed would change with the seasons of the year, as wind might be higher in months where there is more potential for severe weather. With our x-axis being time, humidity seems to be somewhat cyclical based on seasons of the year. The ACF correlogram indicates that the lags are neither geometric nor linear. The PACF correlogram shows decreasing autocorrelation as one gets to later lags. This resembles a possible MA model. Using the augmented Dickey-Fuller test, the p-value is less than our alpha value of 0.05. This means that we will accept the null hypothesis, and therefore, the test tells us there is stationarity.

Based on tsdisplay mean pressure, there is mean-reverting behavior. However, there seems to be seasonality based on the graph. There doesn't appear to be any trends. Intuitively, it would make sense that pressure would change with the seasons of the year. With our x-axis being time, pressure seems to be cyclical based on seasons of the year, although these humps are very small. The ACF correlogram indicates that the lags decline linearly rather than geometrically. The PACF correlogram shows decreasing autocorrelation as one gets to later lags. This resembles a possible AR model. Using the augmented Dickey-Fuller test, the p-value is greater than our alpha value of 0.05. This means that we will reject the null hypothesis, and therefore, the test tells us there is non-stationarity.

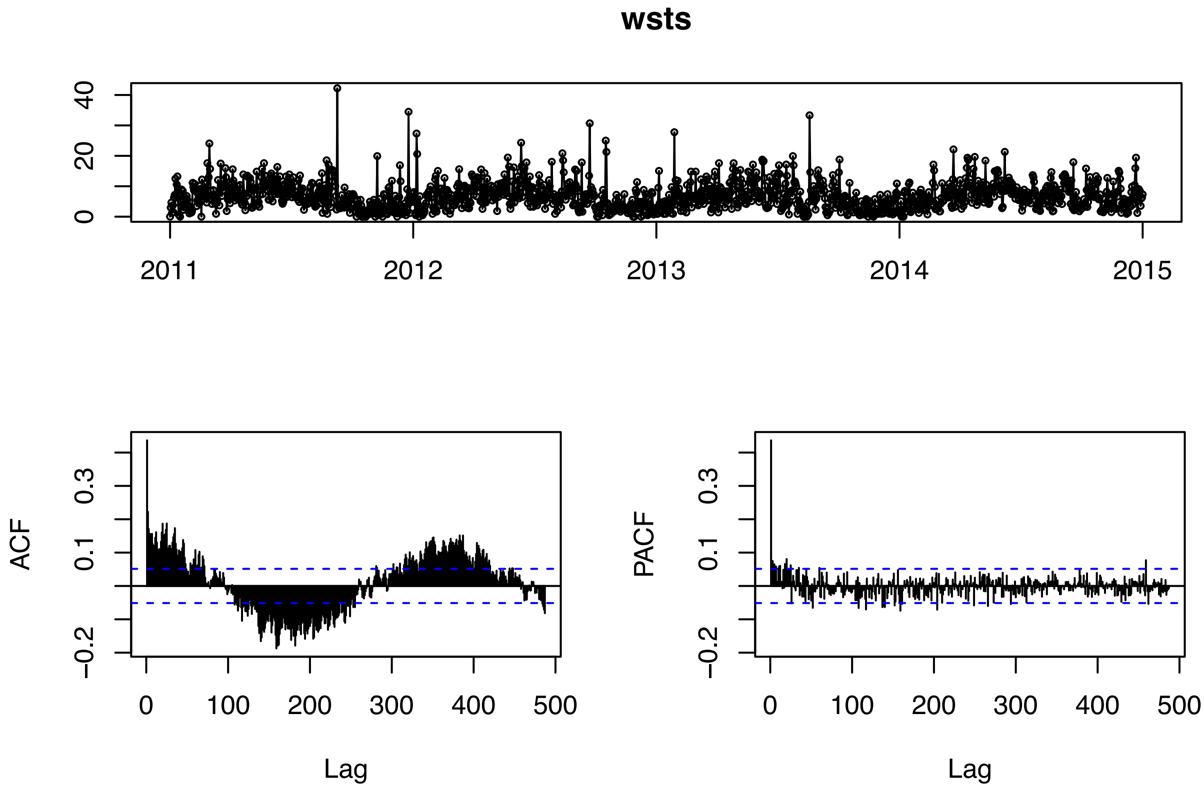
```
tsdisplay(mtts)
```



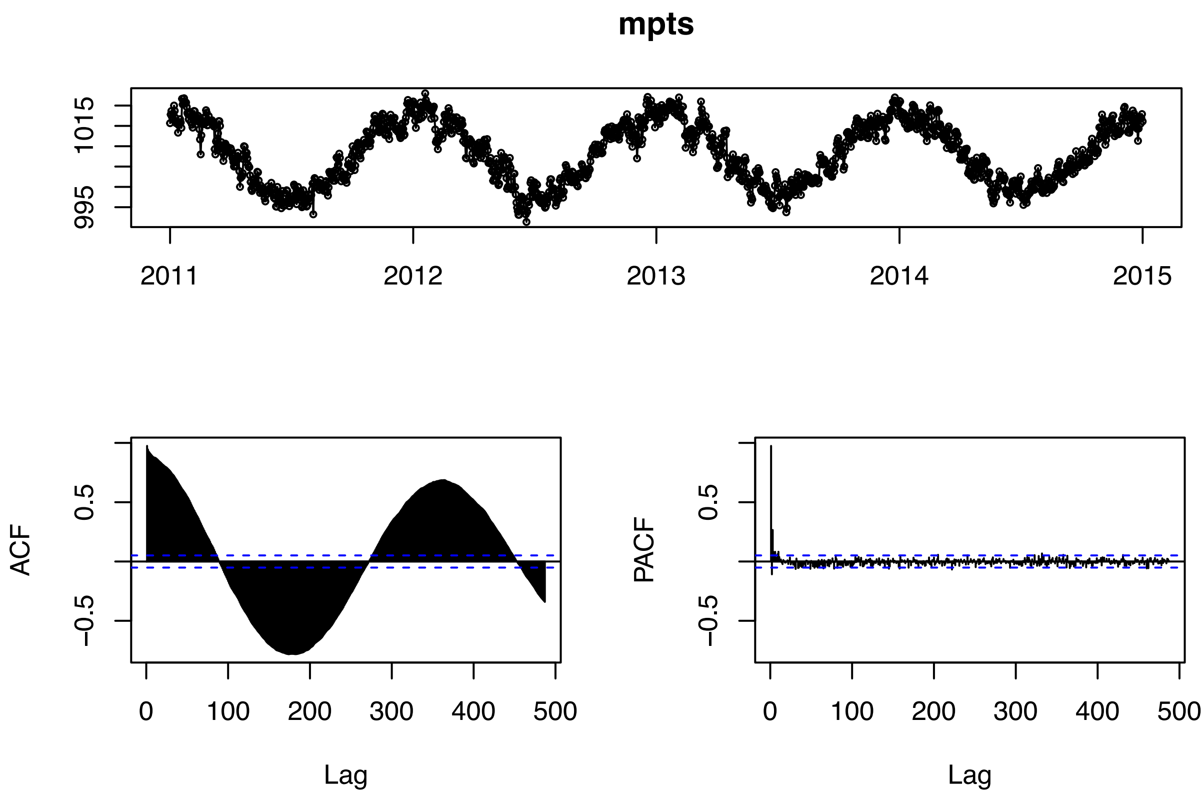

```
tsdisplay(hts)
```




```
tsdisplay(wsts)
```




```
tsdisplay(mpts)
```




```
adf.test(mtts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  mtts
## Dickey-Fuller = -2.0007, Lag order = 11, p-value = 0.578
## alternative hypothesis: stationary
```

```
adf.test(hts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  hts
## Dickey-Fuller = -3.848, Lag order = 11, p-value = 0.01659
## alternative hypothesis: stationary
```

```
adf.test(wsts)
```

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

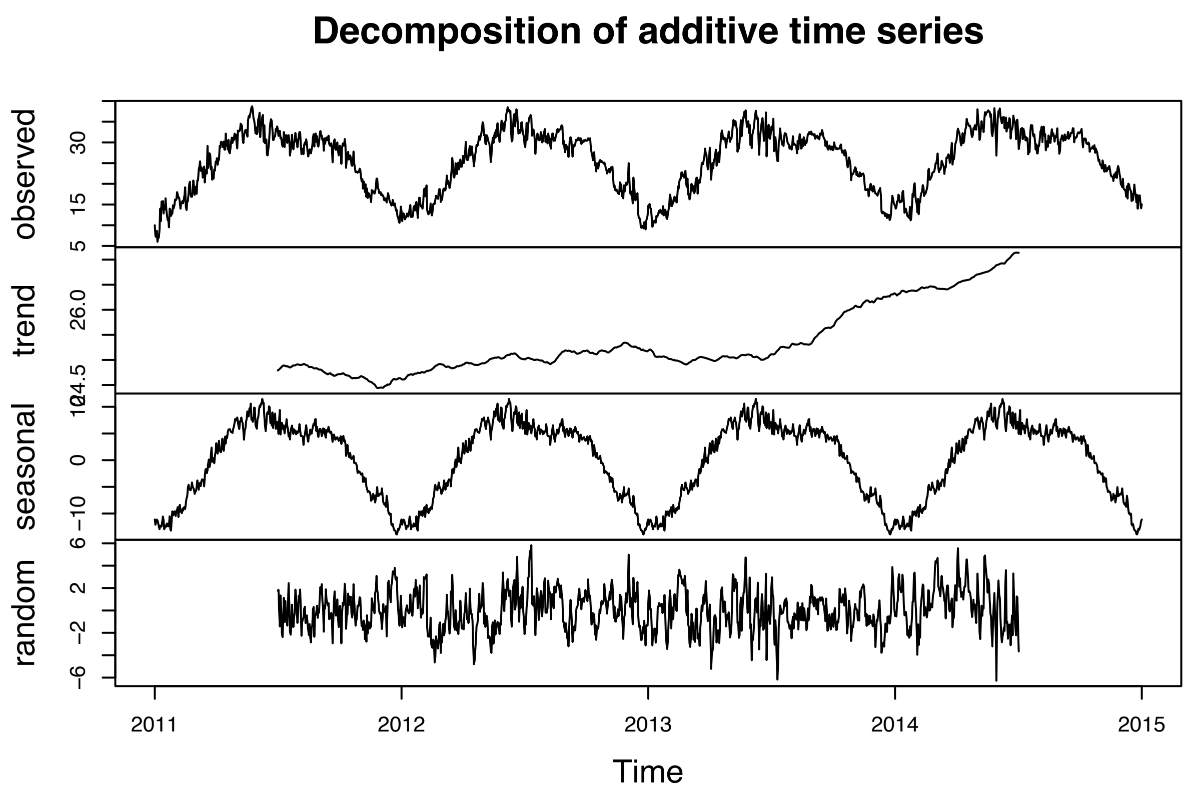
```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  wsts
## Dickey-Fuller = -7.0847, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

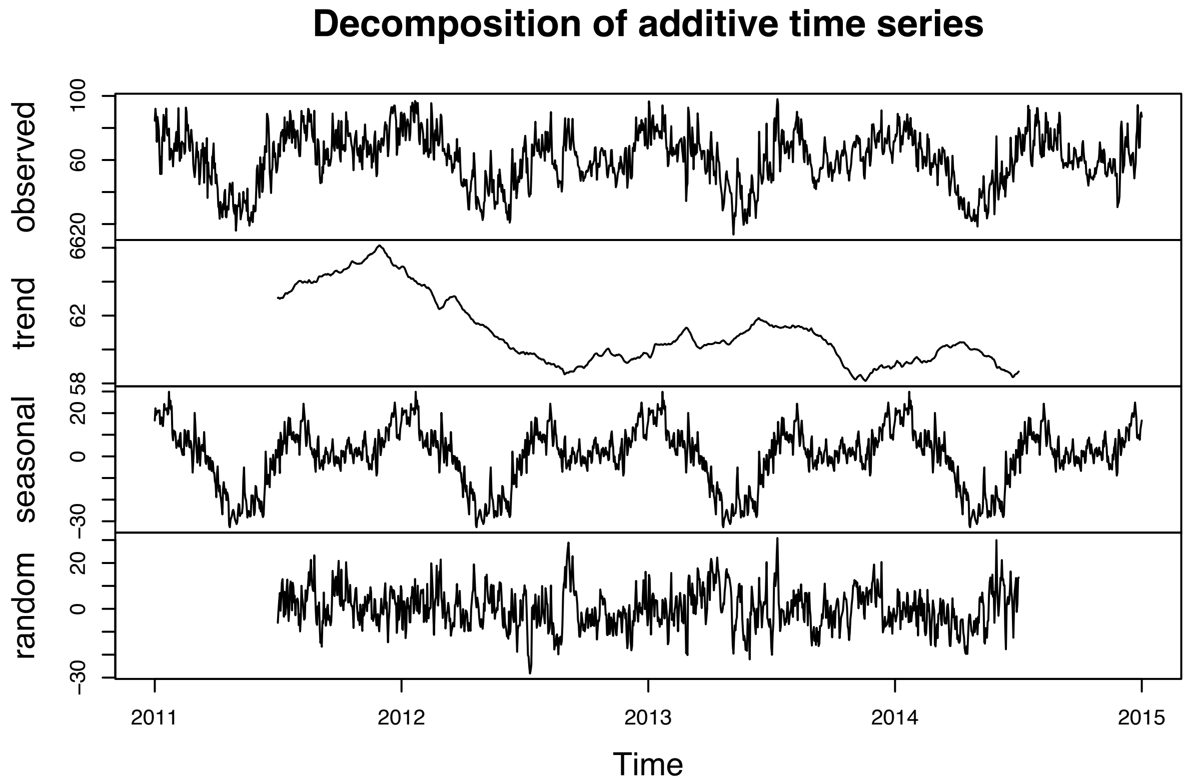
```r
adf.test(mpts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  mpts
## Dickey-Fuller = -2.1347, Lag order = 11, p-value = 0.5213
## alternative hypothesis: stationary
```

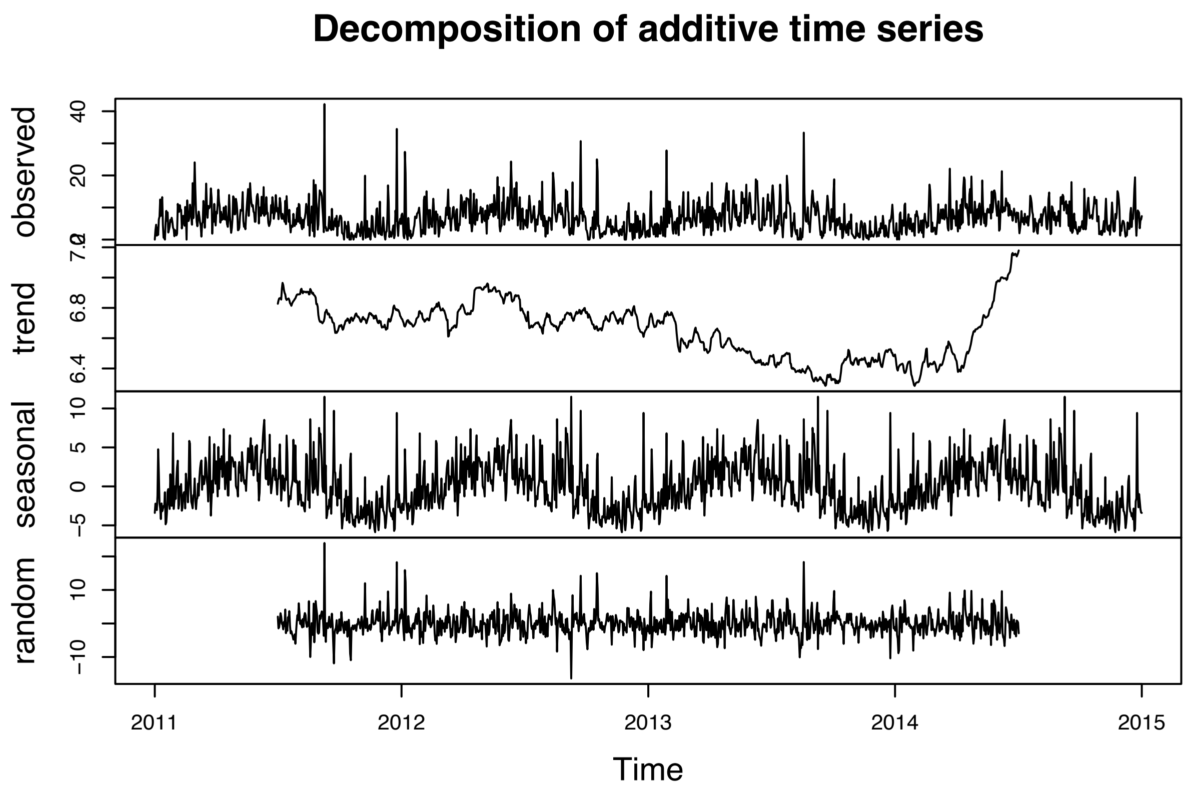
```r
# decomposed
mtts_dc <- decompose(mtts)
hts_dc <- decompose(hts)
wsts_dc <- decompose(wsts)
mpts_dc <- decompose(mpts)

plot(mtts_dc)
```




```r
plot(hts_dc)
```

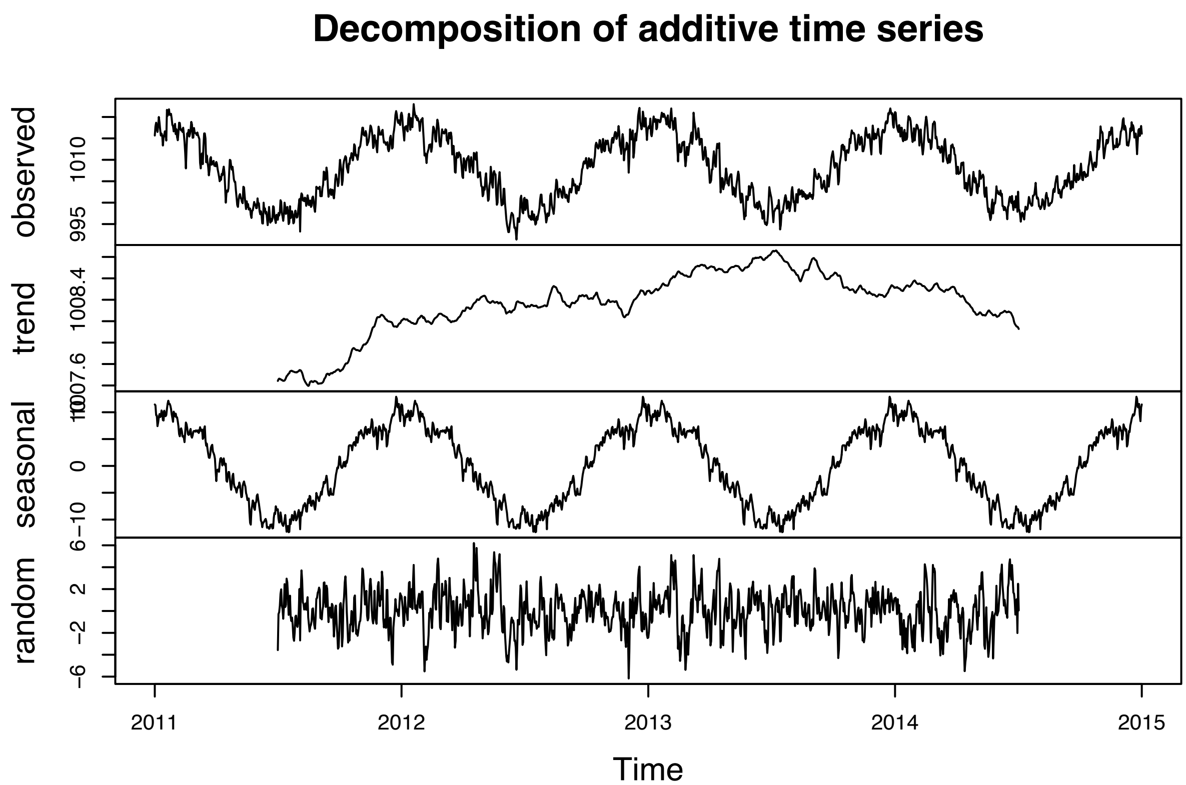

Decomposition of additive time series
observed
trend
seasonal
random
100
60
20
66
62
58
20
0
−30
20
0
−30
2011
2012
2013
2014
2015
Time

```
plot(wsts_dc)
```



Decomposition of additive time series
observed
trend
seasonal
random
Time
2011
2012
2013
2014
2015


```
plot(mpts_dc)
```


Decomposition of additive time series
observed
995
1010
trend
1007.6
1008.4
seasonal
−10
0
10
random
−6
−2
2
6
2011
2012
2013
2014
2015
Time

Question 2b

The ndiffs function will tell us how many differences we need to make our data stationary. This function suggests that we should difference once for mean temperature. This function suggests that we should difference zero times for humidity. However, we believe the data is non-stationary based on the visual of its plot over time. This function suggests that we should difference zero times. However, we believe the data is non-stationary based on the visual of its plot over time. This function suggests that we should difference zero times. However, based on the visual appearance of the data, and in conjunction with the Augmented Dickey-Fuller test, we think non-stationarity exists.

After differencing mean temperature once, there appears to be mean-reverting behavior. The seasonality in the data seems to have disappeared. There is no trend present. The ACF correlogram shows geometrically declining lags over time. The PACF correlogram shows decreasing autocorrelation as one gets to later lags, with the exception of lag 20, a relatively later lag. Using the Augmented Dickey-Fuller test, our p-value is below 0.05, so we fail to reject the null hypothesis and conclude our data is stationary.

After differencing humidity once, there appears to be mean-reverting behavior. The seasonality in the data seems to have disappeared. There is no trend present. The ACF correlogram shows geometrically declining lags over time. The PACF correlogram shows decreasing autocorrelation as one gets to later lags, but lags 1, 2, and 3 may have to be addressed. Using the Augmented Dickey-Fuller test, our p-value is below 0.05, so we fail to reject the null hypothesis and conclude our data is stationary.

After differencing wind speed once, there appears to be mean-reverting behavior. The seasonality in the data seems to have disappeared. There is no trend present. The ACF correlogram shows geometrically declining lags over time. The PACF correlogram shows decreasing autocorrelation as one gets to later lags, but lags 1 and 2 may have to be addressed. Using the Augmented Dickey-Fuller test, our p-value is below 0.05, so we fail to reject the null hypothesis and conclude our data is stationary.

After differencing mean pressure once, there appears to be mean-reverting behavior. The seasonality in the data seems to have disappeared. There is no evidence of trend. The ACF correlogram shows geometrically declining lags over time. The PACF correlogram shows decreasing autocorrelation as one gets to later lags, with the exception of lag 30, a relatively later lag. Using the Augmented Dickey-Fuller test, our p-value is below 0.05, so we fail to reject the null hypothesis that our data is stationary.

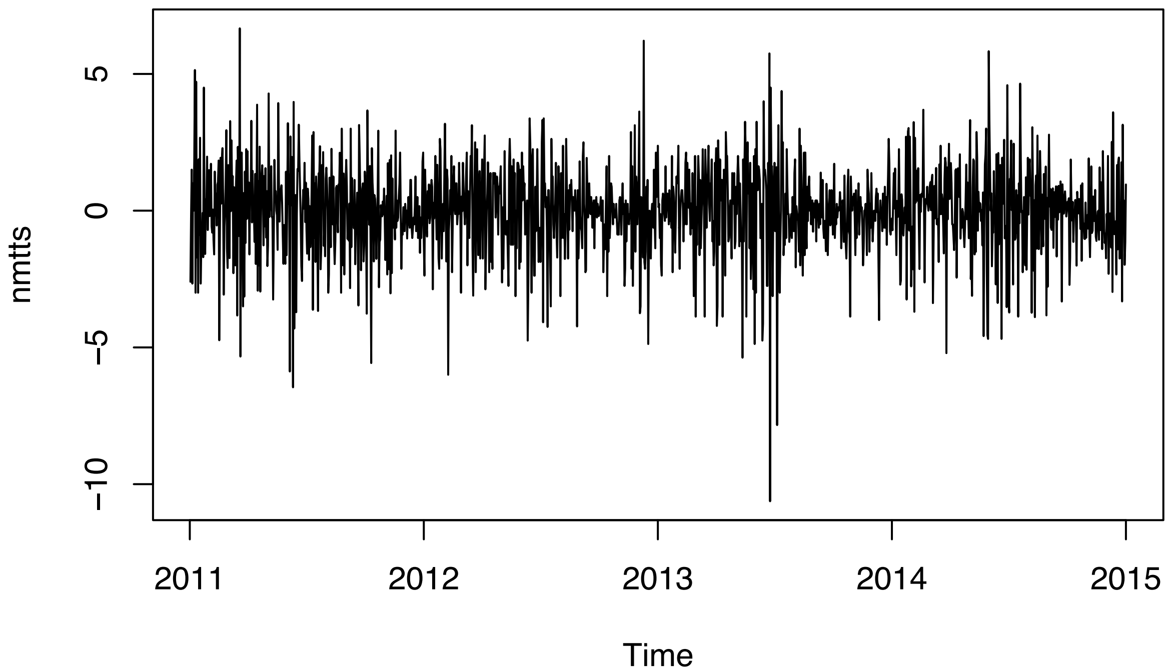
```
ndiffs(mtts)
```

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

```
nmtts <- diff(mtts, 1)
ndiffs(nmtts)
```

```
## [1] 0
```

```
nmtts_dc <- decompose(nmtts)
plot(nmtts)
```



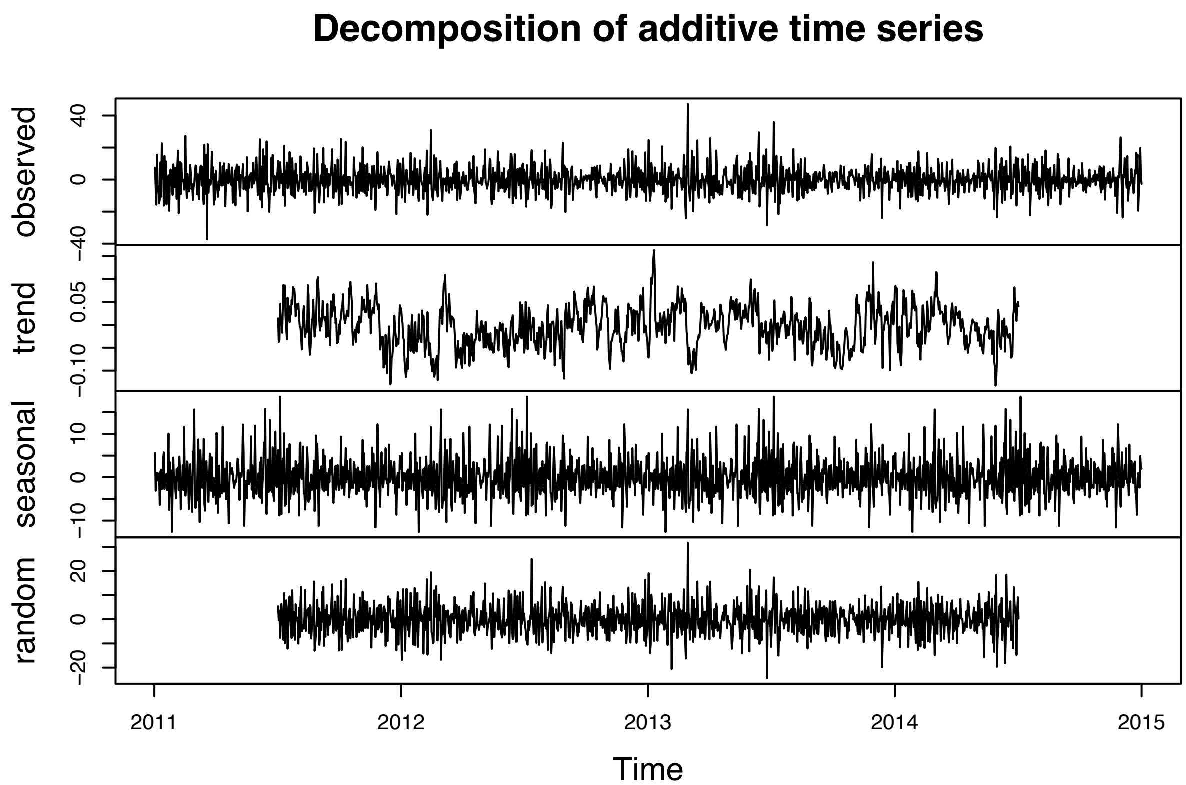

```
ndiffs(hts)
```

```
## [1] 0
```

```
nhts <- diff(hts, 1)
ndiffs(nhts)
```

```
## [1] 0
```

```
nhts_dc <- decompose(nhts)
plot(nhts_dc)
```



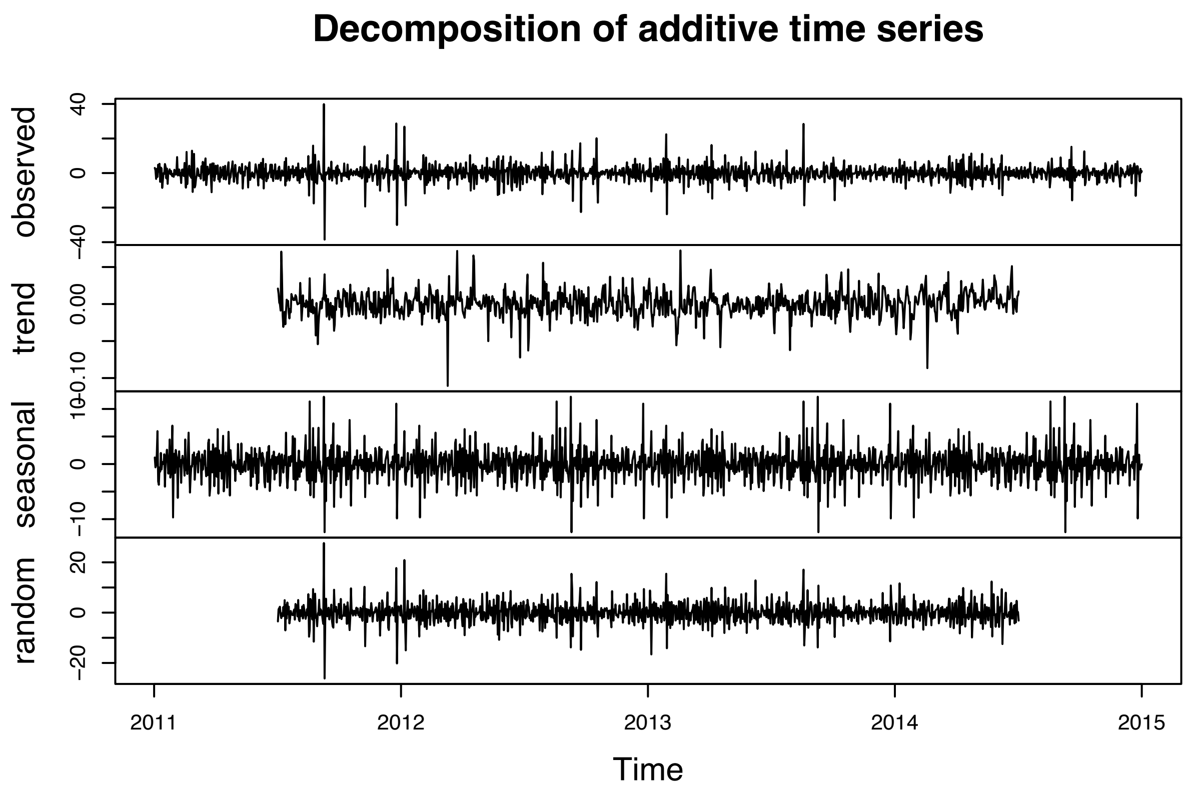

```r
ndiffs(wsts)
```

```
## [1] 0
```

```r
nwsts <- diff(wsts, 1)
ndiffs(nwsts)
```

```
## [1] 0
```

```r
nwsts_dc <- decompose(nwsts)
plot(nwsts_dc)
```



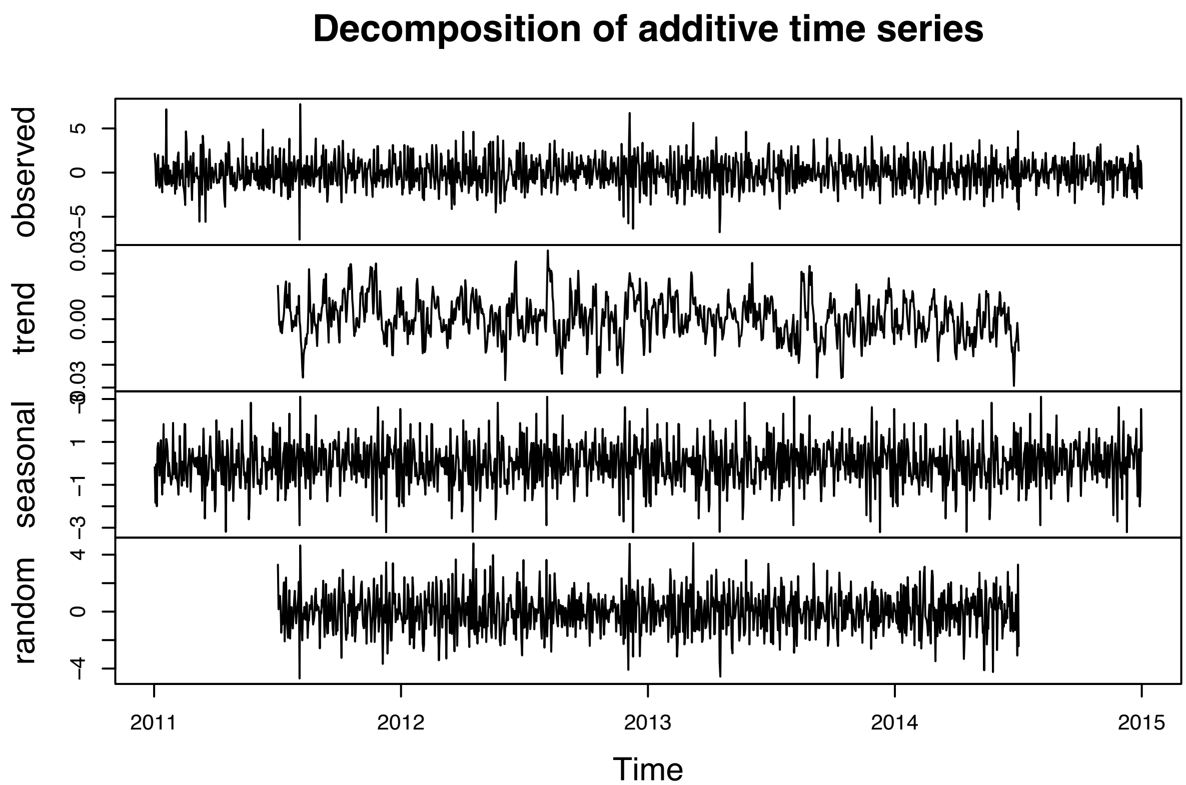

```r
ndiffs(mpts)
```

```
## [1] 0
```

```r
nmpts <- diff(mpts, 1)
ndiffs(nmpts)
```

```
## [1] 0
```

```r
nmpts_dc <- decompose(nmpts)
plot(nmpts_dc)
```

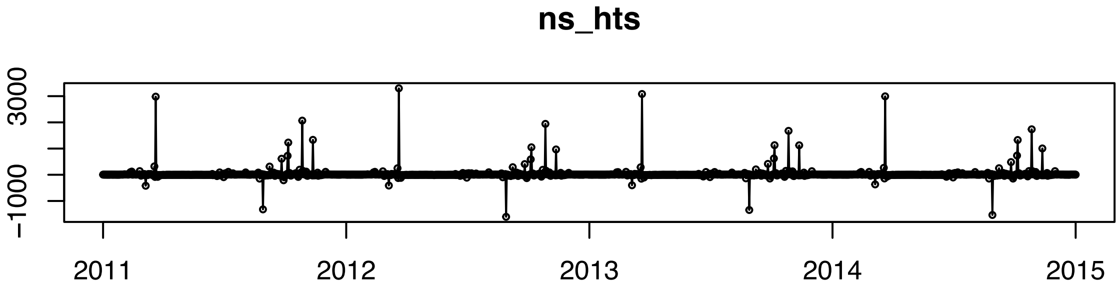


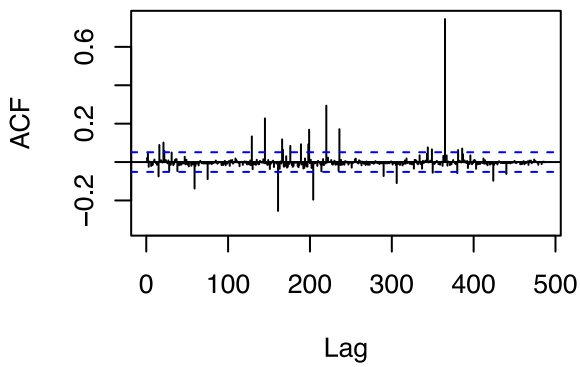

```
ns_hts = hts/hts_dc$seasonal
adf.test(ns_hts)
```

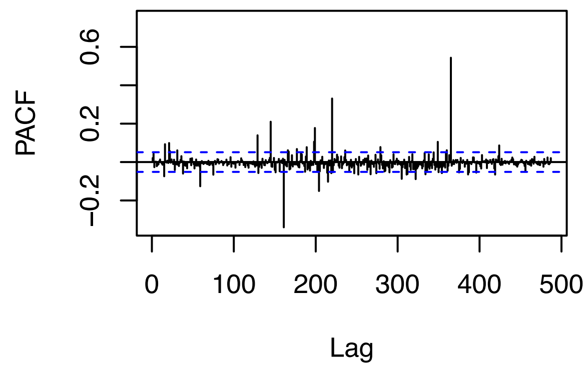
```
## Warning in adf.test(ns_hts): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ns_hts
## Dickey-Fuller = -11.114, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

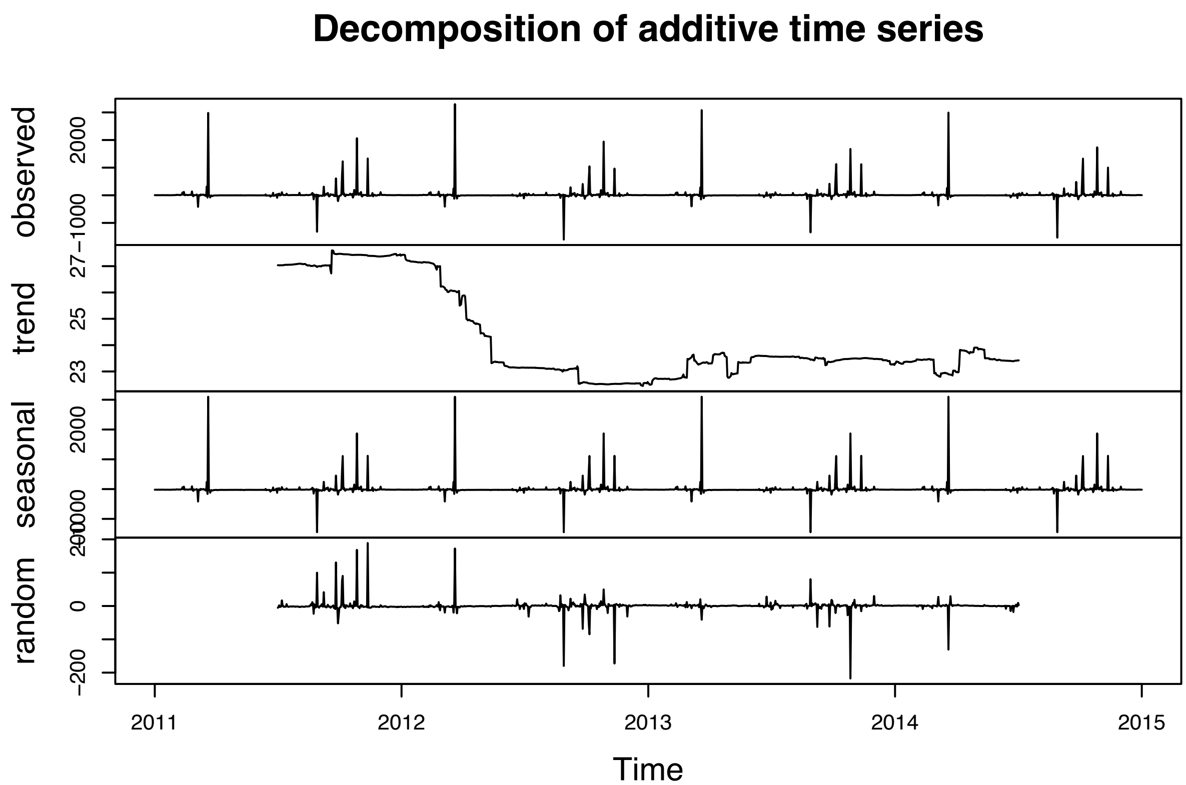
```
tsdisplay(ns_hts)
```








```
ns_hts_dc <- decompose(ns_hts)
plot(ns_hts_dc)
```



Question 3a

Looking at the AIC, it appears 20 lags would be optimal based on AIC. Looking at the BIC, it would appear 10 lags would be optimal. We chose to test up to 20 lags because based on the ACF and PACF, there were lags going past the ± 2/sqrt(T) dashed line. We didn't test further than this, as later lags were less significant. We will use the BIC results and suggest 10 lags, as this had the lowest BIC (5,532.791). The BIC tests more harshly than the AIC. We will use a lag 10 AR model, based off the results of the BIC (5,532.791).

```
m1 = dynlm(nmtts ~ L(nmtts, 1))
m2 = dynlm(nmtts ~ L(nmtts, 1:2))
m3 = dynlm(nmtts ~ L(nmtts, 1:3))
m4 = dynlm(nmtts ~ L(nmtts, 1:4))
m5 = dynlm(nmtts ~ L(nmtts, 1:5))
m6 = dynlm(nmtts ~ L(nmtts, 1:6))
m7 = dynlm(nmtts ~ L(nmtts, 1:7))
m8 = dynlm(nmtts ~ L(nmtts, 1:8))
m9 = dynlm(nmtts ~ L(nmtts, 1:9))
m10 = dynlm(nmtts ~ L(nmtts, 1:10))
m11 = dynlm(nmtts ~ L(nmtts, 1:11))
m12 = dynlm(nmtts ~ L(nmtts, 1:12))
m13 = dynlm(nmtts ~ L(nmtts, 1:13))
m14 = dynlm(nmtts ~ L(nmtts, 1:14))
m15 = dynlm(nmtts ~ L(nmtts, 1:15))
m16 = dynlm(nmtts ~ L(nmtts, 1:16))
```

```r
m17 = dynlm(nmtts ~ L(nmtts, 1:17))
m18 = dynlm(nmtts ~ L(nmtts, 1:18))
m19 = dynlm(nmtts ~ L(nmtts, 1:19))
m20 = dynlm(nmtts ~ L(nmtts, 1:20))
# summary(reg.ts)
AIC(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14,
    m15, m16, m17, m18, m19, m20)
```

```
## Warning in AIC.default(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, :
## models are not all fitted to the same number of observations
```

```
##     df      AIC
## m1   3 5597.073
## m2   4 5580.402
## m3   5 5541.012
## m4   6 5527.470
## m5   7 5519.937
## m6   8 5517.954
## m7   9 5511.712
## m8  10 5499.101
## m9  11 5488.353
## m10 12 5469.439
## m11 13 5468.579
## m12 14 5465.146
## m13 15 5462.848
## m14 16 5460.765
## m15 17 5457.592
## m16 18 5453.598
## m17 19 5451.664
## m18 20 5449.932
## m19 21 5446.995
## m20 22 5439.907
```

```r
BIC(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14,
    m15, m16, m17, m18, m19, m20)
```

```
## Warning in BIC.default(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, :
## models are not all fitted to the same number of observations
```

```
##     df      BIC
## m1   3 5612.930
## m2   4 5601.541
## m3   5 5567.433
## m4   6 5559.171
## m5   7 5556.916
## m6   8 5560.211
## m7   9 5559.244
## m8  10 5551.908
## m9  11 5546.433
## m10 12 5532.791
## m11 13 5537.201
## m12 14 5539.037
```

```
## m13 15 5542.007
## m14 16 5545.190
## m15 17 5547.282
## m16 18 5548.551
## m17 19 5551.879
## m18 20 5555.408
## m19 21 5557.730
## m20 22 5555.900
```

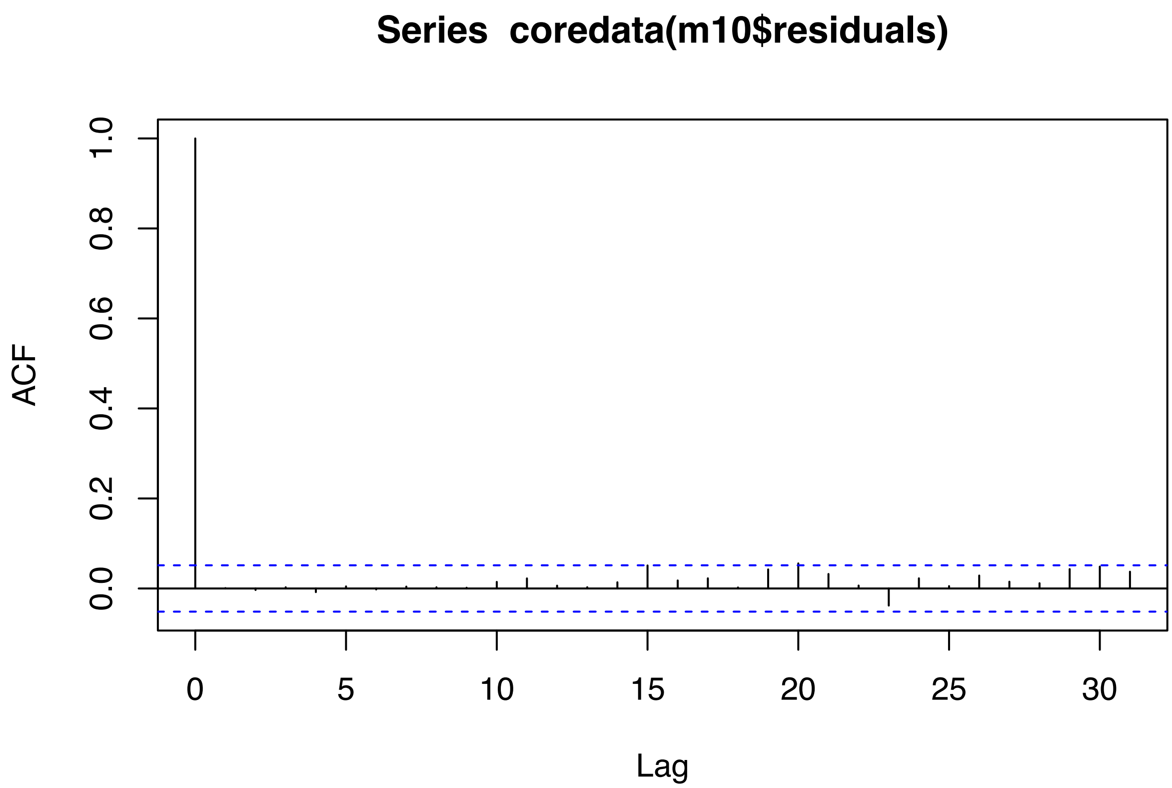
```
ar10 <- arima(nmtts, order = c(10, 0, 0))
summary(m10)
```

```
##
## Time series regression with "ts" data:
## Start = 2011(12), End = 2015(1)
##
## Call:
## dynlm(formula = nmtts ~ L(nmtts, 1:10))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1007 -0.9202  0.0714  1.0237  6.5914
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.005026   0.041715   0.120 0.904123
## L(nmtts, 1:10)1   -0.218866   0.026198  -8.354  < 2e-16 ***
## L(nmtts, 1:10)2   -0.174736   0.026720  -6.539 8.55e-11 ***
## L(nmtts, 1:10)3   -0.210881   0.026982  -7.816 1.05e-14 ***
## L(nmtts, 1:10)4   -0.119463   0.027411  -4.358 1.40e-05 ***
## L(nmtts, 1:10)5   -0.104839   0.027531  -3.808 0.000146 ***
## L(nmtts, 1:10)6   -0.061224   0.027539  -2.223 0.026361 *
## L(nmtts, 1:10)7   -0.083901   0.027422  -3.060 0.002257 **
## L(nmtts, 1:10)8   -0.044604   0.026979  -1.653 0.098491 .
## L(nmtts, 1:10)9   -0.089475   0.026645  -3.358 0.000805 ***
## L(nmtts, 1:10)10 -0.071778   0.026115  -2.749 0.006061 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.588 on 1439 degrees of freedom
## Multiple R-squared:  0.08689,    Adjusted R-squared:  0.08054
## F-statistic: 13.69 on 10 and 1439 DF,  p-value: < 2.2e-16
```

Question 3b

Based on the correlogram of the residuals of the AR(10) model, we can see that there are statistically insignificant residual correlations. This is good news and suggest no autocorrelation of our errors.

```
acf(coredata(m10$residuals))
```



Question 3b continued

We have used the auto ARDL function in order to find the best ARDL model. With these results, we have chosen a model where mean temperature is lagged nine times, humidity is lagged 20 times, wind speed is lagged five times, and mean pressure is lagged four times. This had the lowest AIC at 4,486.131.

```
nclimate <- data.frame(nmtts, nhts, nwsts, nmpts)
auto_ardl(nmtts ~ nhts + nwsts + nmpts, data = nclimate, max_order = 20)
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.name_repair` is
## Using compatibility `.name_repair`.
```

```
## $best_model
##
## Time series regression with "ts" data:
## Start = 21, End = 1460
##
## Call:
## dynlm::dynlm(formula = full_formula, data = data, start = start,
##     end = end)
##
## Coefficients:
## (Intercept)  L(nmtts, 1)  L(nmtts, 2)  L(nmtts, 3)  L(nmtts, 4)  L(nmtts, 5)
```

```
##   0.0032376   -0.1117576   -0.2044448   -0.1959901   -0.1228429   -0.0736902
## L(nmtts, 6)  L(nmtts, 7)  L(nmtts, 8)  L(nmtts, 9)         nhts   L(nhts, 1)
##  -0.0646395   -0.0510370   -0.0568898   -0.0532649   -0.1313417   -0.0085869
##  L(nhts, 2)   L(nhts, 3)   L(nhts, 4)   L(nhts, 5)   L(nhts, 6)   L(nhts, 7)
##  -0.0340826   -0.0330084   -0.0234698   -0.0112299   -0.0155586   -0.0083909
##  L(nhts, 8)   L(nhts, 9)  L(nhts, 10)  L(nhts, 11)  L(nhts, 12)  L(nhts, 13)
##  -0.0169987   -0.0112848   -0.0013777   -0.0029531   -0.0044863   -0.0006232
## L(nhts, 14)  L(nhts, 15)  L(nhts, 16)  L(nhts, 17)  L(nhts, 18)  L(nhts, 19)
##   0.0017894   -0.0035140   -0.0048975   -0.0033284   -0.0042379   -0.0086375
## L(nhts, 20)        nwsts  L(nwsts, 1)  L(nwsts, 2)  L(nwsts, 3)  L(nwsts, 4)
##  -0.0029766   -0.0462462   -0.0501536   -0.0420738   -0.0378026   -0.0294271
## L(nwsts, 5)        nmpts  L(nmpts, 1)  L(nmpts, 2)  L(nmpts, 3)  L(nmpts, 4)
##  -0.0163957   -0.2230975   -0.0359831   -0.0921994   -0.0215448   -0.0425329
##
##
## $best_order
## [1]  9 20  5  4
##
## $top_orders
##    nmtts nhts nwsts nmpts      AIC
## 1      9   20     5     4 4486.131
## 2      9   20     6     4 4486.315
## 3      9   19     5     4 4486.815
## 4     10   20     5     4 4486.833
## 5     10   20     6     4 4486.873
## 6      9   19     6     4 4486.965
## 7     10   19     5     4 4487.533
## 8     10   19     6     4 4487.539
## 9      9   20     5     5 4488.105
## 10     8   20     5     4 4488.678
## 11     9   19     5     5 4488.790
## 12    10   20     5     5 4488.800
## 13     8   20     6     4 4488.931
## 14     8   19     5     4 4489.387
## 15    10   19     5     5 4489.502
## 16     8   19     6     4 4489.606
## 17     4    4    20     4 4490.251
## 18     6   20     7     4 4490.538
## 19     7   20     7     4 4490.640
## 20     8   20     5     5 4490.673
```

```
AUTO_ARDL <- ardl(nmtts ~ nhts + nwsts + nmpts, data = nclimate,
    order = c(9, 20, 5, 4))
summary(AUTO_ARDL)
```

```
##
## Time series regression with "ts" data:
## Start = 21, End = 1460
##
## Call:
## dynlm::dynlm(formula = full_formula, data = data, start = start,
##     end = end)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -7.6860 -0.6369  0.0466  0.6869  5.7804
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0032376  0.0298239   0.109 0.913569
## L(nmtts, 1) -0.1117576  0.0267490  -4.178 3.12e-05 ***
## L(nmtts, 2) -0.2044448  0.0268738  -7.608 5.10e-14 ***
## L(nmtts, 3) -0.1959901  0.0272016  -7.205 9.45e-13 ***
## L(nmtts, 4) -0.1228429  0.0275674  -4.456 9.01e-06 ***
## L(nmtts, 5) -0.0736902  0.0264571  -2.785 0.005420 **
## L(nmtts, 6) -0.0646395  0.0263565  -2.453 0.014308 *
## L(nmtts, 7) -0.0510370  0.0257886  -1.979 0.048006 *
## L(nmtts, 8) -0.0568898  0.0252995  -2.249 0.024690 *
## L(nmtts, 9) -0.0532649  0.0253323  -2.103 0.035675 *
## nhts        -0.1313417  0.0041199 -31.880  < 2e-16 ***
## L(nhts, 1)  -0.0085869  0.0054656  -1.571 0.116387
## L(nhts, 2)  -0.0340826  0.0055414  -6.151 1.01e-09 ***
## L(nhts, 3)  -0.0330084  0.0056624  -5.829 6.90e-09 ***
## L(nhts, 4)  -0.0234698  0.0057425  -4.087 4.62e-05 ***
## L(nhts, 5)  -0.0112299  0.0057529  -1.952 0.051131 .
## L(nhts, 6)  -0.0155586  0.0056682  -2.745 0.006131 **
## L(nhts, 7)  -0.0083909  0.0056760  -1.478 0.139548
## L(nhts, 8)  -0.0169987  0.0056180  -3.026 0.002526 **
## L(nhts, 9)  -0.0112848  0.0056574  -1.995 0.046270 *
## L(nhts, 10) -0.0013777  0.0045054  -0.306 0.759803
## L(nhts, 11) -0.0029531  0.0044800  -0.659 0.509893
## L(nhts, 12) -0.0044863  0.0044325  -1.012 0.311657
## L(nhts, 13) -0.0006232  0.0044153  -0.141 0.887781
## L(nhts, 14)  0.0017894  0.0043630   0.410 0.681782
## L(nhts, 15) -0.0035140  0.0043517  -0.807 0.419523
## L(nhts, 16) -0.0048975  0.0042908  -1.141 0.253907
## L(nhts, 17) -0.0033284  0.0042114  -0.790 0.429470
## L(nhts, 18) -0.0042379  0.0040773  -1.039 0.298805
## L(nhts, 19) -0.0086375  0.0039667  -2.178 0.029609 *
## L(nhts, 20) -0.0029766  0.0038833  -0.767 0.443497
## nwsts       -0.0462462  0.0074327  -6.222 6.47e-10 ***
## L(nwsts, 1) -0.0501536  0.0085112  -5.893 4.76e-09 ***
## L(nwsts, 2) -0.0420738  0.0090589  -4.644 3.73e-06 ***
## L(nwsts, 3) -0.0378026  0.0090745  -4.166 3.29e-05 ***
## L(nwsts, 4) -0.0294271  0.0085955  -3.424 0.000636 ***
## L(nwsts, 5) -0.0163957  0.0075530  -2.171 0.030119 *
## nmpts       -0.2230975  0.0198332 -11.249  < 2e-16 ***
## L(nmpts, 1) -0.0359831  0.0206967  -1.739 0.082327 .
## L(nmpts, 2) -0.0921994  0.0215526  -4.278 2.02e-05 ***
## L(nmpts, 3) -0.0215448  0.0205267  -1.050 0.294084
## L(nmpts, 4) -0.0425329  0.0205404  -2.071 0.038570 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.132 on 1398 degrees of freedom
## Multiple R-squared:  0.5454, Adjusted R-squared:  0.532
## F-statistic:  40.9 on 41 and 1398 DF,  p-value: < 2.2e-16
```

```
AIC(AUTO_ARDL)
```

```
## [1] 4486.131
```

Question 3b continued

Here, we have removed wind speed from the model in order to test whether the slight correlation we detected at the beginning of the project would affect the model through multicollinearity. However, when removing wind speed from the model, there is still a higher AIC at 4,526.719. The model including wind speed sits at an AIC of 4,486.131. This model created without wind speed had nine lags on mean temperature, 20 lags on humidity, and four lags on mean pressure.

```
AICMatrix1 <- matrix(nrow = 20, ncol = 20)

for (i in 1:20) {
    for (j in 1:20) {

        DLM1 <- dynlm(nmtts ~ L(nhts, 1:i) + L(nmpts, 1:j), data = nclimate)
        AICMatrix1[i, j] = AIC(DLM1)
    }

}
AICMatrix1
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
##  [1,] 5610.007 5605.876 5598.179 5595.165 5591.638 5590.794 5586.944 5571.928
##  [2,] 5606.472 5605.993 5597.487 5594.532 5590.846 5590.015 5586.070 5570.758
##  [3,] 5595.223 5595.126 5592.489 5589.550 5586.146 5585.304 5581.817 5565.946
##  [4,] 5588.305 5588.379 5586.006 5587.871 5583.848 5583.001 5579.205 5564.011
##  [5,] 5576.372 5576.848 5574.987 5576.700 5577.205 5576.306 5572.667 5555.994
##  [6,] 5575.517 5575.988 5574.145 5575.852 5576.376 5578.306 5574.658 5557.979
##  [7,] 5569.613 5569.872 5567.817 5569.454 5570.156 5572.137 5573.406 5556.139
##  [8,] 5554.519 5555.262 5553.650 5555.314 5555.924 5557.907 5559.253 5555.967
##  [9,] 5548.632 5549.503 5547.300 5549.078 5549.653 5551.648 5553.050 5549.941
## [10,] 5532.125 5533.071 5530.810 5532.618 5533.072 5535.051 5536.559 5533.877
## [11,] 5529.715 5530.741 5528.695 5530.463 5530.724 5532.683 5534.178 5531.652
## [12,] 5526.677 5527.775 5525.592 5527.338 5527.583 5529.536 5531.004 5528.339
## [13,] 5520.602 5521.681 5519.438 5521.080 5521.558 5523.544 5524.808 5521.715
## [14,] 5518.202 5519.211 5516.829 5518.441 5518.887 5520.882 5522.168 5518.876
## [15,] 5513.766 5514.674 5512.311 5513.870 5514.234 5516.225 5517.347 5513.775
## [16,] 5511.467 5512.438 5510.172 5511.787 5512.096 5514.090 5515.127 5511.522
## [17,] 5508.658 5509.596 5507.255 5508.893 5509.345 5511.342 5512.332 5508.999
## [18,] 5505.305 5506.227 5503.857 5505.500 5505.936 5507.935 5508.947 5505.508
## [19,] 5500.490 5501.454 5499.150 5500.807 5501.051 5503.049 5504.055 5500.840
## [20,] 5499.622 5500.496 5498.266 5499.933 5500.167 5502.163 5503.148 5499.983
##           [,9]    [,10]    [,11]    [,12]    [,13]    [,14]    [,15]    [,16]
##  [1,] 5568.134 5559.132 5557.608 5554.715 5549.913 5547.716 5546.799 5543.927
##  [2,] 5567.174 5557.729 5556.134 5553.102 5547.774 5545.715 5544.819 5541.712
##  [3,] 5562.570 5552.300 5550.845 5548.298 5543.290 5541.657 5540.713 5537.943
##  [4,] 5560.818 5550.287 5548.916 5546.119 5540.537 5538.837 5537.774 5535.040
```

```
##  [5,] 5552.526 5541.268 5540.023 5536.575 5531.470 5529.932 5528.918 5526.763
##  [6,] 5554.525 5543.260 5542.013 5538.569 5533.434 5531.891 5530.877 5528.730
##  [7,] 5552.881 5540.673 5539.366 5535.697 5530.774 5529.235 5528.172 5526.348
##  [8,] 5552.802 5540.730 5539.319 5535.466 5530.302 5528.806 5527.782 5526.034
##  [9,] 5551.903 5539.581 5538.268 5533.974 5528.449 5526.877 5525.877 5523.880
## [10,] 5535.735 5537.173 5535.718 5531.508 5526.583 5525.112 5524.124 5522.066
## [11,] 5533.496 5534.971 5536.907 5532.444 5527.559 5526.031 5525.044 5522.896
## [12,] 5530.160 5531.679 5533.649 5533.921 5529.090 5527.524 5526.535 5524.462
## [13,] 5523.538 5525.137 5527.116 5527.401 5528.822 5527.395 5526.452 5524.144
## [14,] 5520.734 5522.292 5524.282 5524.649 5525.949 5527.702 5526.636 5524.318
## [15,] 5515.538 5517.204 5519.199 5519.379 5520.758 5522.596 5523.854 5521.638
## [16,] 5513.312 5515.006 5517.004 5517.129 5518.502 5520.335 5521.670 5523.621
## [17,] 5510.766 5512.504 5514.496 5514.522 5515.865 5517.689 5519.024 5520.958
## [18,] 5507.199 5508.992 5510.991 5510.790 5512.230 5514.037 5515.414 5517.352
## [19,] 5502.560 5504.324 5506.308 5505.894 5507.139 5508.865 5510.286 5512.177
## [20,] 5501.696 5503.452 5505.437 5505.001 5506.249 5507.968 5509.395 5511.291
##          [,17]    [,18]    [,19]    [,20]
##  [1,] 5540.687 5538.760 5535.259 5527.836
##  [2,] 5538.164 5536.287 5532.748 5525.866
##  [3,] 5534.700 5532.595 5529.218 5522.340
##  [4,] 5531.442 5529.552 5525.989 5519.182
##  [5,] 5523.115 5520.911 5517.280 5510.079
##  [6,] 5525.111 5522.909 5519.279 5512.071
##  [7,] 5522.864 5520.239 5516.733 5509.067
##  [8,] 5522.866 5520.172 5516.624 5508.796
##  [9,] 5520.840 5517.776 5514.253 5506.621
## [10,] 5519.319 5516.311 5512.496 5504.849
## [11,] 5520.194 5517.308 5513.622 5505.745
## [12,] 5521.824 5518.965 5515.390 5507.598
## [13,] 5521.300 5518.589 5515.029 5507.668
## [14,] 5521.707 5518.822 5515.349 5507.976
## [15,] 5518.959 5515.668 5512.358 5504.755
## [16,] 5520.944 5517.659 5514.356 5506.746
## [17,] 5522.900 5519.612 5516.287 5508.715
## [18,] 5519.305 5521.259 5518.098 5510.452
## [19,] 5514.149 5516.134 5516.638 5507.887
## [20,] 5513.265 5515.252 5515.794 5509.791
```

```r
which.min(AICMatrix1)
```

```
## [1] 60
```

```r
AICMatrix2 <- matrix(nrow = 20, ncol = 20)

for (i in 1:20) {
    for (j in 1:20) {

        DLM2 <- dynlm(nmtts ~ L(nwsts, 1:i) + L(nmpts, 1:j),
            data = nclimate)
        AICMatrix2[i, j] = AIC(DLM2)
    }

}
AICMatrix2
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
##  [1,] 5625.815 5623.381 5615.259 5611.176 5606.618 5605.738 5601.730 5589.632
##  [2,] 5620.283 5620.846 5612.537 5608.404 5604.080 5603.181 5598.887 5586.379
##  [3,] 5618.517 5619.153 5614.531 5610.392 5606.056 5605.156 5600.880 5588.377
##  [4,] 5613.376 5613.938 5609.369 5611.017 5606.667 5605.781 5601.555 5588.811
##  [5,] 5605.112 5605.929 5601.349 5602.774 5602.040 5601.155 5596.999 5584.446
##  [6,] 5603.257 5604.078 5599.249 5600.677 5599.727 5601.656 5597.587 5585.024
##  [7,] 5599.950 5600.748 5595.898 5597.260 5596.400 5598.370 5598.413 5585.664
##  [8,] 5588.645 5589.659 5585.052 5586.429 5585.386 5587.372 5587.638 5587.621
##  [9,] 5585.209 5586.294 5581.454 5582.854 5581.830 5583.818 5584.019 5583.788
## [10,] 5576.077 5577.013 5571.759 5573.030 5572.128 5574.112 5574.179 5574.068
## [11,] 5574.325 5575.324 5570.213 5571.440 5570.461 5572.444 5572.536 5572.439
## [12,] 5571.840 5572.856 5567.500 5568.793 5567.667 5569.656 5569.746 5569.614
## [13,] 5567.336 5568.369 5563.058 5564.260 5563.281 5565.277 5565.234 5564.991
## [14,] 5565.722 5566.674 5561.347 5562.554 5561.423 5563.413 5563.246 5563.058
## [15,] 5564.448 5565.380 5560.188 5561.399 5560.284 5562.276 5562.117 5561.917
## [16,] 5560.128 5561.170 5556.220 5557.519 5556.367 5558.358 5557.922 5557.529
## [17,] 5556.389 5557.391 5552.438 5553.804 5552.899 5554.890 5554.473 5554.169
## [18,] 5555.208 5556.214 5551.245 5552.607 5551.656 5553.652 5553.235 5552.941
## [19,] 5551.727 5552.754 5547.827 5549.225 5548.049 5550.036 5549.722 5549.455
## [20,] 5549.298 5550.299 5545.324 5546.677 5545.476 5547.451 5547.111 5546.816
##           [,9]    [,10]    [,11]    [,12]    [,13]    [,14]    [,15]    [,16]
##  [1,] 5585.565 5575.704 5573.941 5570.501 5565.321 5562.859 5561.624 5558.239
##  [2,] 5582.416 5572.095 5570.232 5566.870 5561.672 5559.566 5558.374 5554.944
##  [3,] 5584.405 5574.077 5572.206 5568.823 5563.629 5561.519 5560.344 5556.889
##  [4,] 5584.872 5574.284 5572.447 5568.938 5563.509 5561.334 5560.136 5556.890
##  [5,] 5580.738 5569.869 5568.208 5564.916 5559.733 5557.647 5556.513 5553.423
##  [6,] 5581.317 5570.693 5569.008 5565.802 5560.686 5558.622 5557.501 5554.337
##  [7,] 5582.025 5571.429 5569.812 5566.531 5561.539 5559.347 5558.228 5555.253
##  [8,] 5583.960 5573.381 5571.764 5568.507 5563.501 5561.324 5560.194 5557.225
##  [9,] 5585.782 5575.273 5573.645 5570.381 5565.311 5563.131 5561.982 5558.965
## [10,] 5576.068 5577.251 5575.631 5572.364 5567.293 5565.108 5563.958 5560.934
## [11,] 5574.439 5575.547 5577.544 5574.304 5569.221 5567.043 5565.891 5562.874
## [12,] 5571.614 5572.756 5574.743 5576.140 5571.108 5568.904 5567.756 5564.707
## [13,] 5566.991 5568.230 5570.220 5571.549 5573.037 5570.817 5569.680 5566.642
## [14,] 5565.058 5566.274 5568.264 5569.610 5571.013 5572.486 5571.362 5568.324
## [15,] 5563.917 5565.090 5567.080 5568.470 5569.875 5571.331 5573.179 5570.103
## [16,] 5559.518 5560.734 5562.731 5564.087 5565.460 5566.882 5568.805 5570.487
## [17,] 5556.163 5557.413 5559.412 5560.705 5562.058 5563.391 5565.310 5566.970
## [18,] 5554.937 5556.207 5558.206 5559.474 5560.851 5562.175 5564.106 5565.762
## [19,] 5551.450 5552.842 5554.837 5556.219 5557.622 5558.868 5560.804 5562.402
## [20,] 5548.813 5550.197 5552.182 5553.593 5554.914 5556.163 5558.125 5559.712
##          [,17]    [,18]    [,19]    [,20]
##  [1,] 5555.091 5553.626 5549.110 5542.611
##  [2,] 5551.746 5550.395 5546.100 5539.982
##  [3,] 5553.680 5552.318 5547.967 5541.806
##  [4,] 5553.531 5552.176 5547.680 5541.770
##  [5,] 5549.706 5548.361 5544.188 5537.874
##  [6,] 5550.689 5549.424 5545.396 5538.913
##  [7,] 5551.542 5550.256 5546.269 5539.566
##  [8,] 5553.537 5552.250 5548.265 5541.544
##  [9,] 5555.257 5553.996 5550.076 5543.356
## [10,] 5557.210 5555.951 5551.984 5545.199
## [11,] 5559.133 5557.867 5553.881 5547.140
```

```
## [12,] 5560.978 5559.730 5555.628 5548.916
## [13,] 5562.934 5561.688 5557.577 5550.734
## [14,] 5564.595 5563.365 5559.278 5552.353
## [15,] 5566.378 5565.149 5561.101 5554.194
## [16,] 5566.887 5565.693 5561.546 5554.708
## [17,] 5568.645 5567.471 5563.281 5556.389
## [18,] 5567.448 5569.441 5565.260 5558.336
## [19,] 5564.122 5566.084 5565.467 5558.549
## [20,] 5561.498 5563.450 5563.068 5559.188
```

```
which.min(AICMatrix2)
```

```
## [1] 385
```

```
ardl_mt.ht.mp <- dynlm((nmtts) ~ L(nmtts, 10) + L(nhts, 20) +
    L(nmpts, 3))
ardl_mt.ws.mp <- dynlm((nmtts) ~ L(nmtts, 10) + L(nwsts, 5) +
    L(nmpts, 20))


auto_ardl(nmtts ~ nhts + nmpts, data = nclimate, max_order = 20)
```

```
## $best_model
##
## Time series regression with "ts" data:
## Start = 21, End = 1460
##
## Call:
## dynlm::dynlm(formula = full_formula, data = data, start = start,
##     end = end)
##
## Coefficients:
## (Intercept)  L(nmtts, 1)  L(nmtts, 2)  L(nmtts, 3)  L(nmtts, 4)  L(nmtts, 5)
##   0.0030240   -0.0936251   -0.2137953   -0.2087550   -0.1283888   -0.0868013
## L(nmtts, 6)  L(nmtts, 7)  L(nmtts, 8)  L(nmtts, 9)         nhts   L(nhts, 1)
##  -0.0762325   -0.0620023   -0.0583820   -0.0466138   -0.1259529    0.0008959
##  L(nhts, 2)   L(nhts, 3)   L(nhts, 4)   L(nhts, 5)   L(nhts, 6)   L(nhts, 7)
##  -0.0305439   -0.0312114   -0.0215430   -0.0120898   -0.0189624   -0.0112516
##  L(nhts, 8)   L(nhts, 9)  L(nhts, 10)  L(nhts, 11)  L(nhts, 12)  L(nhts, 13)
##  -0.0182797   -0.0109633   -0.0029782   -0.0042294   -0.0047735   -0.0013835
## L(nhts, 14)  L(nhts, 15)  L(nhts, 16)  L(nhts, 17)  L(nhts, 18)  L(nhts, 19)
##   0.0005915   -0.0053076   -0.0057770   -0.0040247   -0.0048522   -0.0092936
## L(nhts, 20)        nmpts  L(nmpts, 1)  L(nmpts, 2)  L(nmpts, 3)  L(nmpts, 4)
##  -0.0028579   -0.2226937   -0.0297082   -0.0889512   -0.0152302   -0.0382587
##
##
## $best_order
## [1]  9 20  4
##
## $top_orders
##    nmtts nhts nmpts      AIC
## 1      9   20     4 4526.719
## 2     10   20     4 4527.336
```

```
## 3      9   19     4 4527.371
## 4     10   19     4 4528.005
## 5      8   20     4 4528.101
## 6      9   20     3 4528.182
## 7      8   19     4 4528.769
## 8      9   19     3 4528.772
## 9     10   20     3 4528.814
## 10    11   20     4 4529.237
## 11    10   19     3 4529.411
## 12     8   20     3 4529.443
## 13    11   19     4 4529.911
## 14     8   19     3 4530.055
## 15    11   20     3 4530.763
## 16     7   20     4 4530.911
## 17    11   19     3 4531.364
## 18     7   19     4 4531.550
## 19     7   20     3 4531.864
## 20     7   19     3 4532.463
```

```
auto_ardl(nmtts ~ nwsts + nmpts, data = nclimate, max_order = 20)
```

```
## $best_model
##
## Time series regression with "ts" data:
## Start = 21, End = 1460
##
## Call:
## dynlm::dynlm(formula = full_formula, data = data, start = start,
##     end = end)
##
## Coefficients:
##  (Intercept)   L(nmtts, 1)   L(nmtts, 2)   L(nmtts, 3)   L(nmtts, 4)
##     0.002939     -0.224471     -0.214815     -0.211498     -0.143941
##  L(nmtts, 5)   L(nmtts, 6)   L(nmtts, 7)   L(nmtts, 8)   L(nmtts, 9)
##    -0.095731     -0.061714     -0.100805     -0.043834     -0.096240
## L(nmtts, 10)         nwsts   L(nwsts, 1)   L(nwsts, 2)   L(nwsts, 3)
##    -0.070072      0.010829     -0.024105     -0.016683     -0.005049
##  L(nwsts, 4)   L(nwsts, 5)   L(nwsts, 6)   L(nwsts, 7)   L(nwsts, 8)
##    -0.009227     -0.008872      0.019597      0.015446      0.024608
##  L(nwsts, 9)  L(nwsts, 10)  L(nwsts, 11)  L(nwsts, 12)  L(nwsts, 13)
##     0.028808      0.015665      0.015706      0.004269      0.003072
## L(nwsts, 14)  L(nwsts, 15)  L(nwsts, 16)  L(nwsts, 17)  L(nwsts, 18)
##    -0.003153      0.003662      0.001415      0.008527      0.015222
## L(nwsts, 19)  L(nwsts, 20)         nmpts   L(nmpts, 1)   L(nmpts, 2)
##     0.023158      0.012232     -0.353096     -0.012625     -0.155224
##  L(nmpts, 3)   L(nmpts, 4)   L(nmpts, 5)   L(nmpts, 6)   L(nmpts, 7)
##    -0.031318     -0.088264     -0.019528     -0.025946     -0.019680
##  L(nmpts, 8)   L(nmpts, 9)
##     0.015347     -0.055765
##
##
## $best_order
## [1] 10 20  9
##
```

```
## $top_orders
##    nmtts nwsts nmpts      AIC
## 1     10    20     9 5273.866
## 2     11    20     9 5275.108
## 3     10    20    10 5275.715
## 4     10    19     9 5276.377
## 5     13    20     9 5276.642
## 6     12    20     9 5276.705
## 7     15    20     9 5276.755
## 8     11    20    10 5276.884
## 9      7    20     4 5277.527
## 10    11    19     9 5277.645
## 11    10    19    10 5278.240
## 12    13    20    10 5278.353
## 13    12    20    10 5278.465
## 14    15    20    10 5278.469
## 15     8    20     4 5278.934
## 16     7    20     5 5279.198
## 17    13    19     9 5279.341
## 18    12    19     9 5279.342
## 19    11    19    10 5279.439
## 20     7    19     4 5279.610
```

```
auto_ardl.ardl_mt.ht.mp <- ardl(nmtts ~ nhts + nmpts, data = nclimate,
    order = c(9, 20, 4))
auto_ardl.ardl_mt.ws.mp <- ardl(nmtts ~ nwsts + nmpts, data = nclimate,
    order = c(10, 20, 9))

AIC(AUTO_ARDL)
```

```
## [1] 4486.131
```

```
AIC(ardl_mt.ht.mp)
```

```
## [1] 5538.909
```

```
AIC(ardl_mt.ws.mp)
```

```
## [1] 5536.639
```

```
AIC(auto_ardl.ardl_mt.ht.mp)
```

```
## [1] 4526.719
```

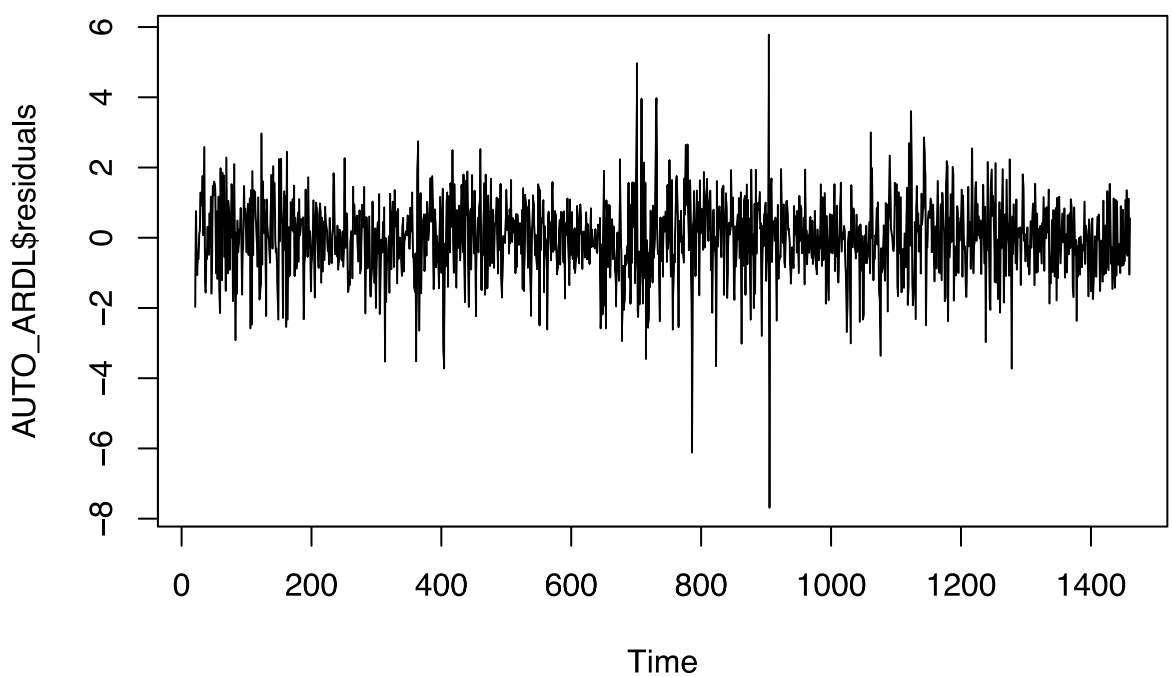
```
AIC(auto_ardl.ardl_mt.ws.mp)
```

```
## [1] 5273.866
```

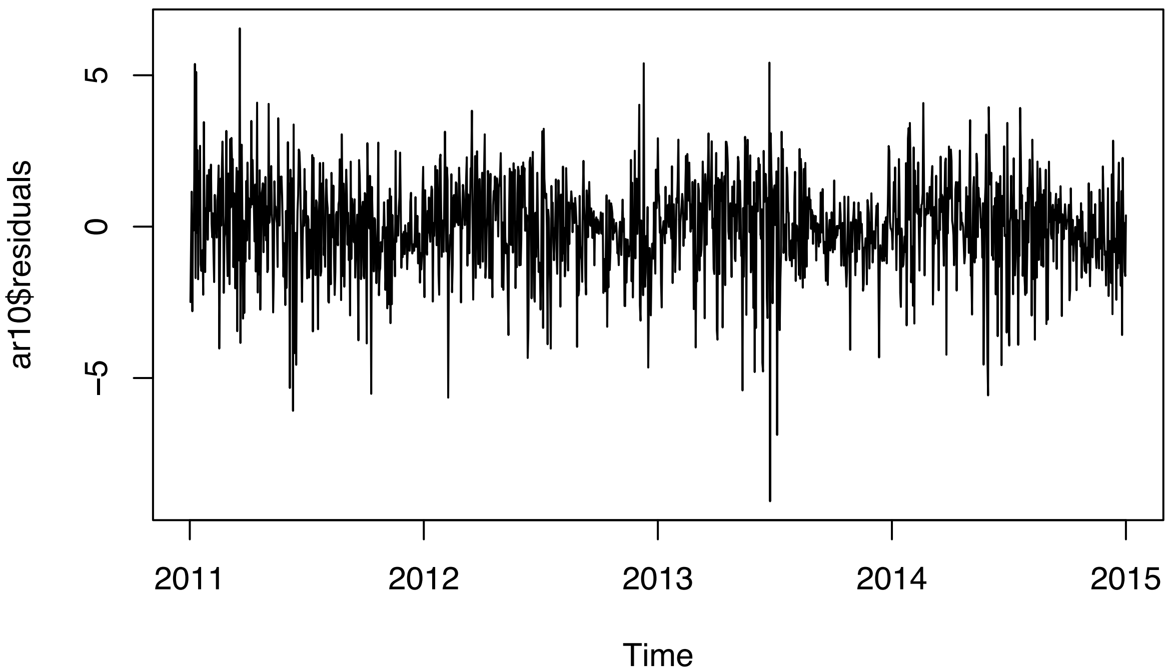
Question 4

Based on the residuals of both the AR(10) model and the ARDL(9,20,5,4) model, as well as the AIC of both the AR(10) model and the ARDL(9,20,5,4) model, we conclude that the dominant model for our data is the AR(10) model. The residuals for the AR(10) model are noisy and are less erratic than the residuals of the ARDL(9,20,5,4) model. For instance, the ARDL(9,20,5,4) model residuals appear to be smaller visually based off the residual graph, whereas the AR(10) model residuals appear to be larger and have more spread. In addition, the AIC for the ARDL(9,20,5,4) model is 4,486.131, whereas the AIC for the AR(10) model is 5,532.791. Since both of these AIC values come from the same dataset and can be compared to one another, we can conclude that the ARDL(9,20,5,4) model is the superior model compared to the AR(10) model.

```
plot(AUTO_ARDL$residuals)
```




```
plot(ar10$residuals)
```



ar10$residuals
5
0
−5
2011
2012
2013
2014
2015
Time

Question 5

Based on the results of the cross correlation function (CCF), there is very little statistical significant lags above or below the dotted line, besides in lag 3, which is an early lag with statistical significance. However, this peak is at around 0.05, which is very low all things considered.

To determine the direction of the possibility of joint correlation over time, we will use the Granger Causality test. We hypothesize that there might be joint causality between mean pressure and mean temperature, as temperature may affect pressure as well as pressure may affect mean. After the Granger Causality test with mean temperature and mean pressure, we reject the null hypothesis and conclude that mean temperature does not granger-cause mean pressure. However, we fail to reject the null hypothesis and conclude that mean pressure does granger-cause mean temperature. The first had a p-value less than 0.05, and the last had a p-value greater than 0.05

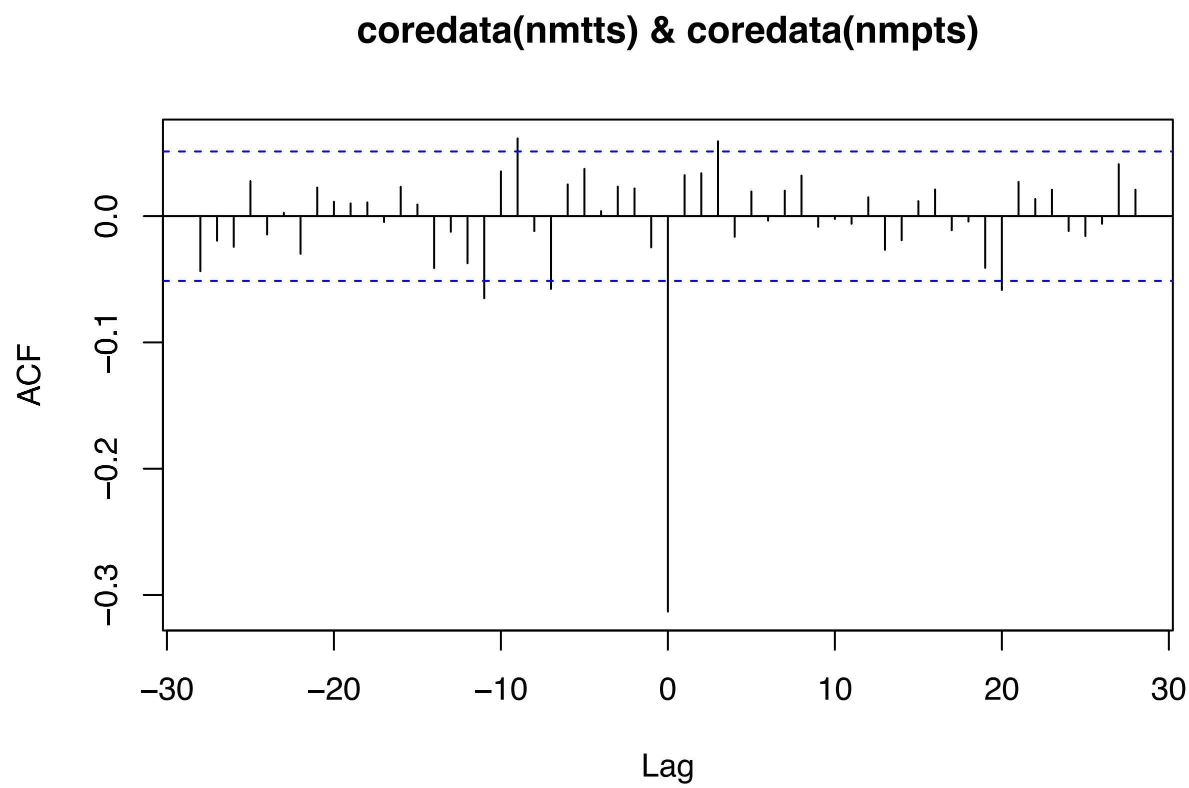
After the Granger Causality test with mean temperature and wind speed, we fail to reject the null hypothesis and conclude that wind speed granger-causes mean temperature and that mean temperature granger-causes wind speed. This is because both tests resulted in a p-value below 0.05.

After the Granger Causality test with mean temperature and humidity, we reject the null hypothesis that mean temperature granger-causes humidity. This is because the test resulted in a p-value above 0.05. However, we fail to reject the null hypothesis that humidity granger-causes mean temperature. This is because the test resulted in a p-value below 0.05.

###Based off the cumulative results of our Granger Causality tests, we might need a VAR(p) model for the relationship between mean temperature and wind speed, as mean temperature is granger-causing wind speed.

Based on the VARselect function, the Schwarz Criterion for lag 3 is the lowest. This penalizes more harshly than the AIC and is similar to the BIC. The AIC and BIC of the VAR_model are significantly higher than the AIC and BIC of the AR(10) model and the ARDL(9,20,5,4) model, and, looking at the fit of the VAR_model, there is no real fit. Therefore, we conclude that the ARDL(9,20,5,4) model remains as the superior model for our data.

```
ccf(coredata(nmtts), coredata(nmpts))
```




```
library(vars)
var_data_frame <- cbind(nmtts, nhts, nwsts, nmpts)

var_data_frame_total <- data.frame(var_data_frame)

VARselect(var_data_frame_total, lag.max = 10)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##     10      5      3     10
##
## $criteria
##                 1           2           3           4           5           6
## AIC(n)    8.488750    8.268545    8.160468    8.116044    8.084986    8.076582
## HQ(n)     8.515924    8.317458    8.231119    8.208434    8.199114    8.212450
## SC(n)     8.561569    8.399618    8.349796    8.363626    8.390822    8.440673
## FPE(n) 4859.790090 3899.275799 3499.831854 3347.765045 3245.399452 3218.259713
##                 7           8           9          10
## AIC(n)    8.065000    8.066350    8.049408    8.038704
## HQ(n)     8.222606    8.245695    8.250492    8.261526
## SC(n)     8.487345    8.546950    8.588263    8.635813
## FPE(n) 3181.223390 3185.554863 3132.079813 3098.782474
```

```
VAR_model <- VAR(var_data_frame_total, p = 3)
summary(VAR_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: nmtts, nhts, nwsts, nmpts
## Deterministic variables: const
## Sample size: 1457
## Log Likelihood: -14169.579
## Roots of the characteristic polynomial:
## 0.6551 0.6551 0.612 0.612 0.6039 0.6039 0.5591 0.5591 0.5429 0.502 0.4781 0.236
## Call:
## VAR(y = var_data_frame_total, p = 3)
##
##
## Estimation results for equation nmtts:
## ======================================
## nmtts = nmtts.l1 + nhts.l1 + nwsts.l1 + nmpts.l1 + nmtts.l2 + nhts.l2 + nwsts.l2 + nmpts.l2 + nmtts.l
##
##           Estimate Std. Error t value Pr(>|t|)
## nmtts.l1 -0.176517   0.036395  -4.850 1.37e-06 ***
## nhts.l1   0.007332   0.007267   1.009   0.3131
## nwsts.l1 -0.023100   0.010062  -2.296   0.0218 *
## nmpts.l1 -0.026381   0.028605  -0.922   0.3566
## nmtts.l2 -0.143422   0.036162  -3.966 7.67e-05 ***
## nhts.l2  -0.005143   0.007180  -0.716   0.4739
## nwsts.l2 -0.019517   0.010600  -1.841   0.0658 .
## nmpts.l2 -0.001392   0.027509  -0.051   0.9597
## nmtts.l3 -0.163131   0.036310  -4.493 7.59e-06 ***
## nhts.l3  -0.004719   0.007194  -0.656   0.5120
## nwsts.l3 -0.002358   0.009992  -0.236   0.8135
## nmpts.l3  0.014584   0.028533   0.511   0.6093
## const     0.006779   0.042302   0.160   0.8727
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.615 on 1444 degrees of freedom
## Multiple R-Squared: 0.06914, Adjusted R-squared: 0.0614
## F-statistic: 8.938 on 12 and 1444 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation nhts:
## =====================================
## nhts = nmtts.l1 + nhts.l1 + nwsts.l1 + nmpts.l1 + nmtts.l2 + nhts.l2 + nwsts.l2 + nmpts.l2 + nmtts.l3
##
##           Estimate Std. Error t value Pr(>|t|)
## nmtts.l1  0.788322   0.178640   4.413 1.10e-05 ***
## nhts.l1  -0.036053   0.035667  -1.011    0.312
## nwsts.l1  0.013184   0.049385   0.267    0.790
## nmpts.l1 -0.166382   0.140403  -1.185    0.236
## nmtts.l2 -0.123788   0.177497  -0.697    0.486
## nhts.l2  -0.183438   0.035242  -5.205 2.22e-07 ***
## nwsts.l2  0.007241   0.052029   0.139    0.889
## nmpts.l2 -0.010355   0.135024  -0.077    0.939
## nmtts.l3 -0.008312   0.178222  -0.047    0.963
```

```
## nhts.l3  -0.181570   0.035312  -5.142 3.09e-07 ***
## nwsts.l3 -0.084257   0.049043  -1.718    0.086 .
## nmpts.l3 -0.077485   0.140051  -0.553    0.580
## const     0.005281   0.207634   0.025    0.980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 7.925 on 1444 degrees of freedom
## Multiple R-Squared: 0.08386, Adjusted R-squared: 0.07624
## F-statistic: 11.01 on 12 and 1444 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation nwsts:
## ======================================
## nwsts = nmtts.l1 + nhts.l1 + nwsts.l1 + nmpts.l1 + nmtts.l2 + nhts.l2 + nwsts.l2 + nmpts.l2 + nmtts.l
##
##           Estimate Std. Error t value Pr(>|t|)
## nmtts.l1 -0.047318   0.096686  -0.489  0.62463
## nhts.l1  -0.077620   0.019304  -4.021 6.10e-05 ***
## nwsts.l1 -0.489669   0.026729 -18.320  < 2e-16 ***
## nmpts.l1  0.011344   0.075991   0.149  0.88135
## nmtts.l2  0.396237   0.096067   4.125 3.93e-05 ***
## nhts.l2   0.034159   0.019074   1.791  0.07353 .
## nwsts.l2 -0.353048   0.028160 -12.537  < 2e-16 ***
## nmpts.l2  0.025934   0.073080   0.355  0.72274
## nmtts.l3  0.295038   0.096460   3.059  0.00226 **
## nhts.l3   0.036691   0.019112   1.920  0.05508 .
## nwsts.l3 -0.190785   0.026544  -7.188 1.05e-12 ***
## nmpts.l3 -0.034311   0.075801  -0.453  0.65087
## const     0.003231   0.112379   0.029  0.97707
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.289 on 1444 degrees of freedom
## Multiple R-Squared: 0.2217,  Adjusted R-squared: 0.2152
## F-statistic: 34.27 on 12 and 1444 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation nmpts:
## ======================================
## nmpts = nmtts.l1 + nhts.l1 + nwsts.l1 + nmpts.l1 + nmtts.l2 + nhts.l2 + nwsts.l2 + nmpts.l2 + nmtts.l
##
##            Estimate Std. Error t value Pr(>|t|)
## nmtts.l1  0.0067430  0.0353781   0.191  0.84887
## nhts.l1   0.0014459  0.0070635   0.205  0.83784
## nwsts.l1 -0.0061156  0.0097803  -0.625  0.53187
## nmpts.l1  0.0883460  0.0278055   3.177  0.00152 **
## nmtts.l2 -0.0495704  0.0351517  -1.410  0.15870
## nhts.l2   0.0072817  0.0069793   1.043  0.29697
## nwsts.l2  0.0004926  0.0103038   0.048  0.96188
## nmpts.l2 -0.2956903  0.0267404 -11.058  < 2e-16 ***
## nmtts.l3  0.0024003  0.0352953   0.068  0.94579
```

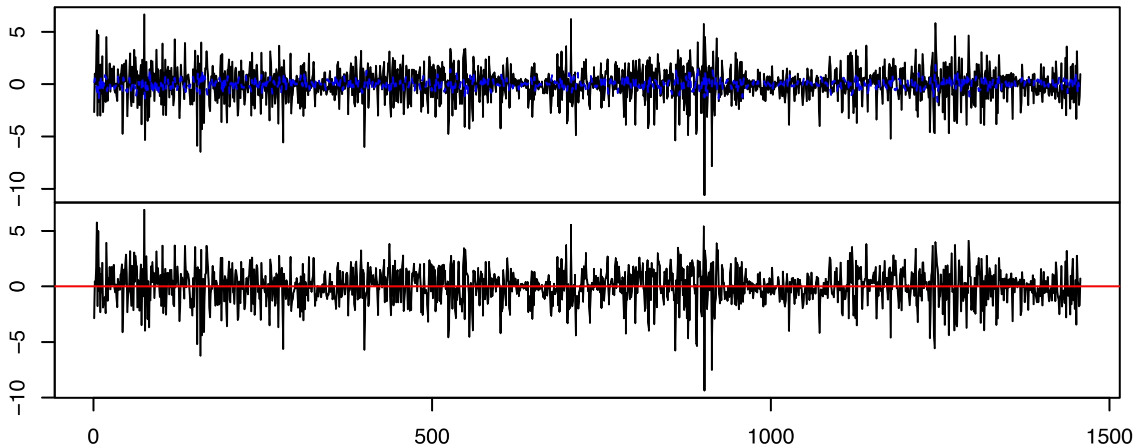
```
## nhts.l3   0.0082499  0.0069932   1.180  0.23831
## nwsts.l3 -0.0146652  0.0097126  -1.510  0.13128
## nmpts.l3 -0.0899972  0.0277360  -3.245  0.00120 **
## const    -0.0004984  0.0411202  -0.012  0.99033
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.57 on 1444 degrees of freedom
## Multiple R-Squared: 0.1037,  Adjusted R-squared: 0.09627
## F-statistic: 13.92 on 12 and 1444 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##         nmtts   nhts   nwsts   nmpts
## nmtts  2.6072 -8.444  0.4228 -0.8496
## nhts  -8.4442 62.811 -8.3500  2.3387
## nwsts  0.4228 -8.350 18.3995 -0.3789
## nmpts -0.8496  2.339 -0.3789  2.4635
##
## Correlation matrix of residuals:
##          nmtts    nhts    nwsts    nmpts
## nmtts  1.00000 -0.6599  0.06104 -0.33524
## nhts  -0.65987  1.0000 -0.24562  0.18801
## nwsts  0.06104 -0.2456  1.00000 -0.05628
## nmpts -0.33524  0.1880 -0.05628  1.00000
```

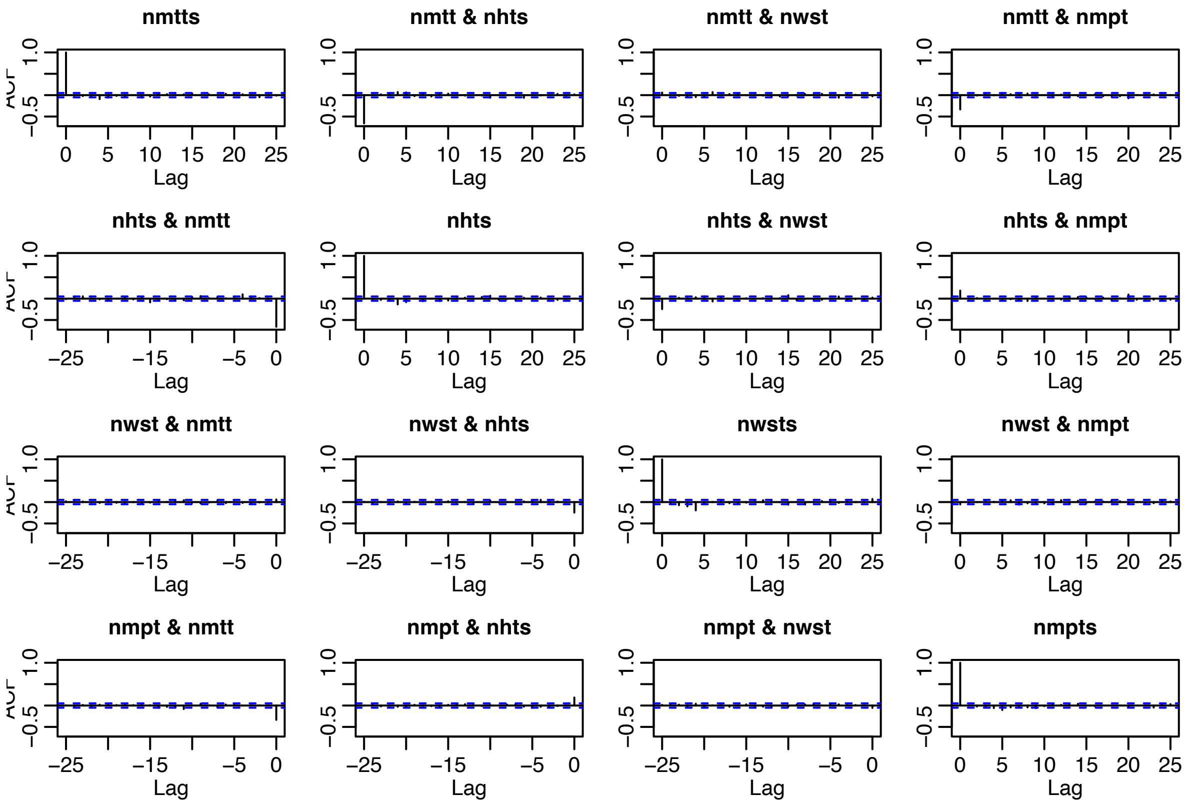
```
plot(VAR_model)
```

```
## Error in plot.new(): figure margins too large
```

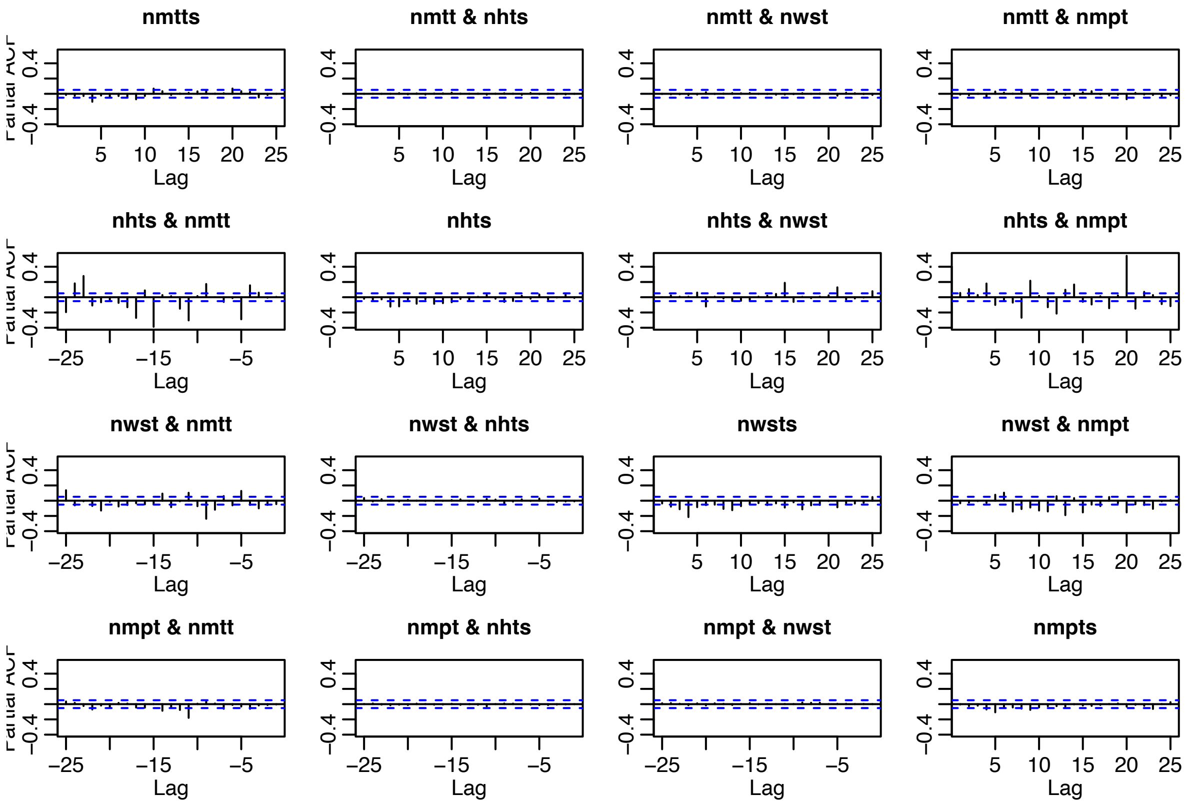
```
acf(residuals(VAR_model))
```

Diagram of fit and residuals for nmtts






```
pacf(residuals(VAR_model))
```




```
AIC(VAR_model)
```

```
## [1] 28443.16
```

```
BIC(VAR_model)
```

```
## [1] 28717.93
```

```
grangertest(nmtts ~ nmpts)
```

```
## Granger causality test
##
## Model 1: nmtts ~ Lags(nmtts, 1:1) + Lags(nmpts, 1:1)
## Model 2: nmtts ~ Lags(nmtts, 1:1)
##   Res.Df Df      F Pr(>F)
## 1   1456
## 2   1457 -1 0.5081 0.4761
```

```
grangertest(nmpts ~ nmtts)
```

```
## Granger causality test
##
## Model 1: nmpts ~ Lags(nmpts, 1:1) + Lags(nmtts, 1:1)
```

```
## Model 2: nmpts ~ Lags(nmpts, 1:1)
##   Res.Df Df      F Pr(>F)
## 1   1456
## 2   1457 -1 0.0166 0.8976
```

```
grangertest(nmtts ~ nwsts)
```

```
## Granger causality test
##
## Model 1: nmtts ~ Lags(nmtts, 1:1) + Lags(nwsts, 1:1)
## Model 2: nmtts ~ Lags(nmtts, 1:1)
##   Res.Df Df     F   Pr(>F)
## 1   1456
## 2   1457 -1 7.723 0.005522 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
grangertest(nwsts ~ nmtts)
```

```
## Granger causality test
##
## Model 1: nwsts ~ Lags(nwsts, 1:1) + Lags(nmtts, 1:1)
## Model 2: nwsts ~ Lags(nwsts, 1:1)
##   Res.Df Df      F   Pr(>F)
## 1   1456
## 2   1457 -1 8.4426 0.003721 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
grangertest(nmtts ~ nhts)
```

```
## Granger causality test
##
## Model 1: nmtts ~ Lags(nmtts, 1:1) + Lags(nhts, 1:1)
## Model 2: nmtts ~ Lags(nmtts, 1:1)
##   Res.Df Df      F Pr(>F)
## 1   1456
## 2   1457 -1 2.2796 0.1313
```

```
grangertest(nhts ~ nmtts)
```

```
## Granger causality test
##
## Model 1: nhts ~ Lags(nhts, 1:1) + Lags(nmtts, 1:1)
## Model 2: nhts ~ Lags(nhts, 1:1)
##   Res.Df Df      F    Pr(>F)
## 1   1456
## 2   1457 -1 24.862 6.896e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```