Final Project

```
library(imputeTS)

## Warning: package 'imputeTS' was built under R version 3.3.3

library(forecast)

## Warning: package 'forecast' was built under R version 3.3.3

library(zoo)

## Warning: package 'zoo' was built under R version 3.3.3

##
## Attaching package: 'zoo'

## The following object is masked from 'package:imputeTS':
##
##     na.locf

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

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.3
```

Read in CO2 data

```
co2 <- read.csv("monthly co2.csv")
```

Explore data

```
str(co2)

## 'data.frame':    713 obs. of  7 variables:
##  $ Year               : int  1958 1958 1958 1958 1958 1958 1958 1958 1958
1958 ...
##  $ Month              : int  3 4 5 6 7 8 9 10 11 12 ...
##  $ decimal.date       : num  1958 1958 1958 1958 1959 ...
##  $ average            : num  316 317 318 -100 316 ...
##  $ interpolated       : num  316 317 318 317 316 ...
##  $ trend..season.corr.: num  315 315 315 315 315 ...
##  $ X.days             : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

head(co2)

##   Year Month decimal.date average interpolated trend..season.corr. X.days
## 1 1958     3     1958.208  315.71       315.71              314.62     -1
## 2 1958     4     1958.292  317.45       317.45              315.29     -1
## 3 1958     5     1958.375  317.50       317.50              314.71     -1
```

```
## 4 1958     6      1958.458  -99.99        317.10                   314.85      -1
## 5 1958     7      1958.542  315.86        315.86                   314.98      -1
## 6 1958     8      1958.625  314.93        314.93                   315.94      -1
```

Trim to CO2 column and data since 1966

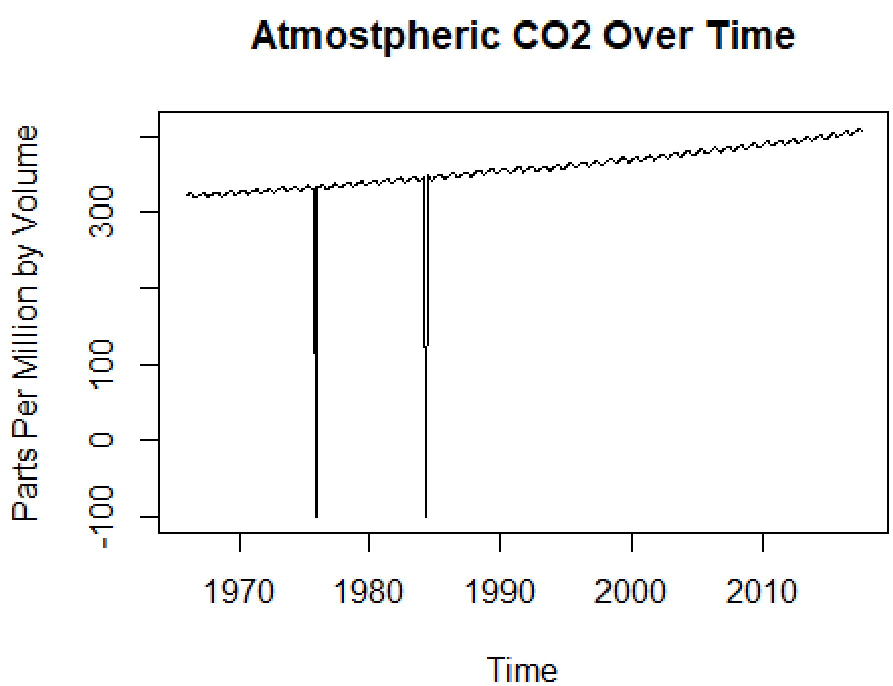
```
co2 <- co2[,c("Year", "Month", "average")]
colnames(co2) <- c("Year", "Month", "CO2")
co2 <- co2[co2$Year >= 1966, ]
co2.ts <- ts(co2$CO2, start = 1966, frequency = 12)
head(co2.ts)
```

```
##          Jan    Feb    Mar    Apr    May    Jun
## 1966 320.62 321.59 322.39 323.87 324.01 323.75
```

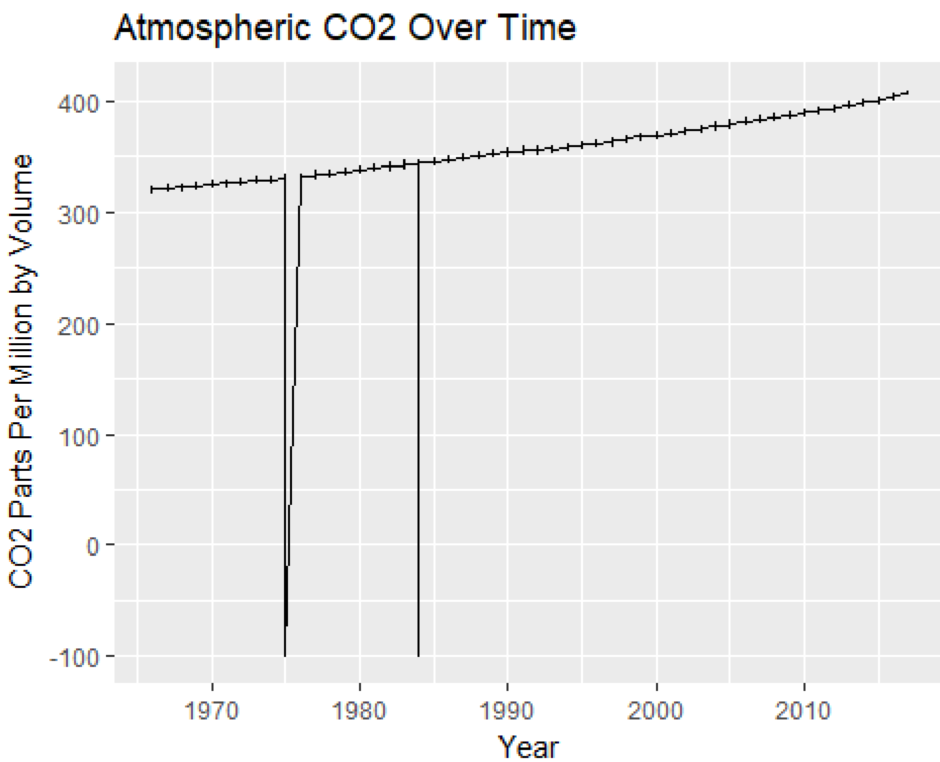
Plot time series

Observe any missing values or outliers

```
#plotNA.distribution(co2.ts)
plot.ts(co2.ts, main = "Atmostpheric CO2 Over Time", ylab = "Parts Per
Million by Volume")
```

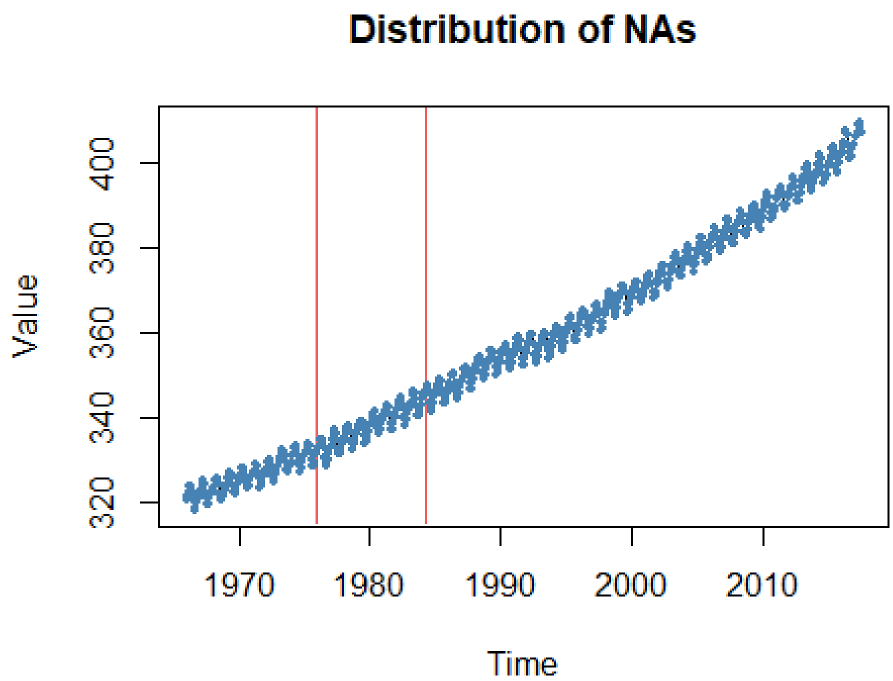



```
ggplot(co2, aes(x=Year, y=CO2)) +
  geom_line() +
  labs(y = "CO2 Parts Per Million by Volume", title = "Atmospheric CO2 Over
Time")
```



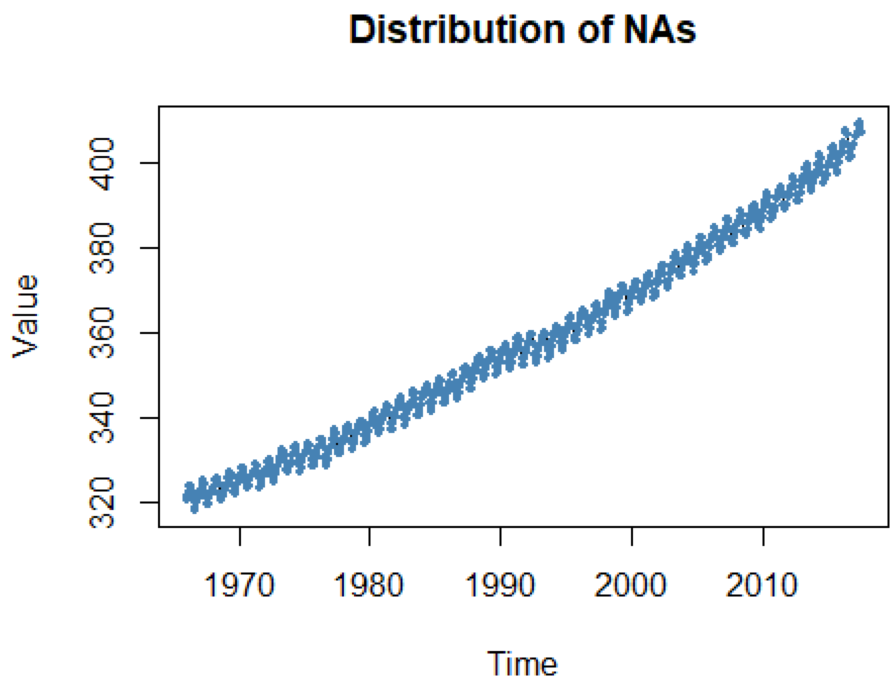
Data < 0 for atmospheric CO2 doesn't make sense so we will remove these and impute data.

```
co2.ts <- ifelse(co2.ts < 0, NA, co2.ts)

plotNA.distribution(co2.ts)
```



Impute missing values

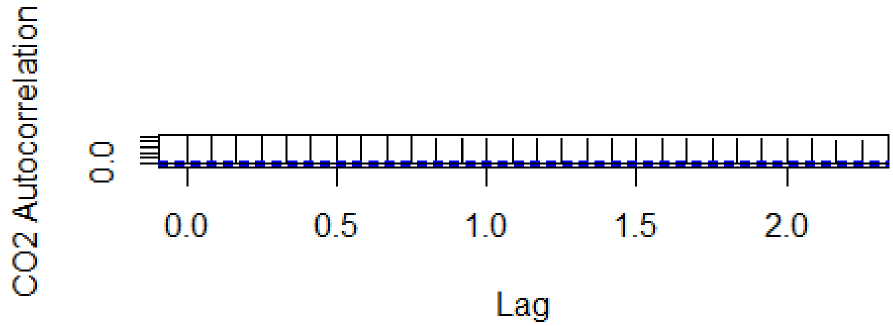
```
co2.ts.imp <- na.interpolation(co2.ts)

plotNA.distribution(co2.ts.imp)
```

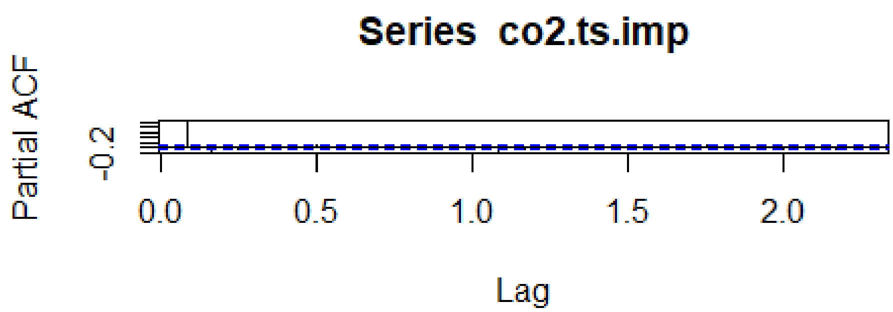


Conclusions: Plot of CO2 time series indicates positive, additive trend and annual seasonality.

ACF

```
#acf(co2.ts.imp, lag = 100)
# figure
# subplot(2,1,1)
# autocorr(co2.ts.imp)
# subplot(2,1,2)
# parcorr(co2.ts.imp)

par(mfrow=c(2,1))
plot(acf(co2.ts.imp, plot=FALSE), ylab = "CO2 Autocorrelation", main = "")
pacf(co2.ts.imp)
```

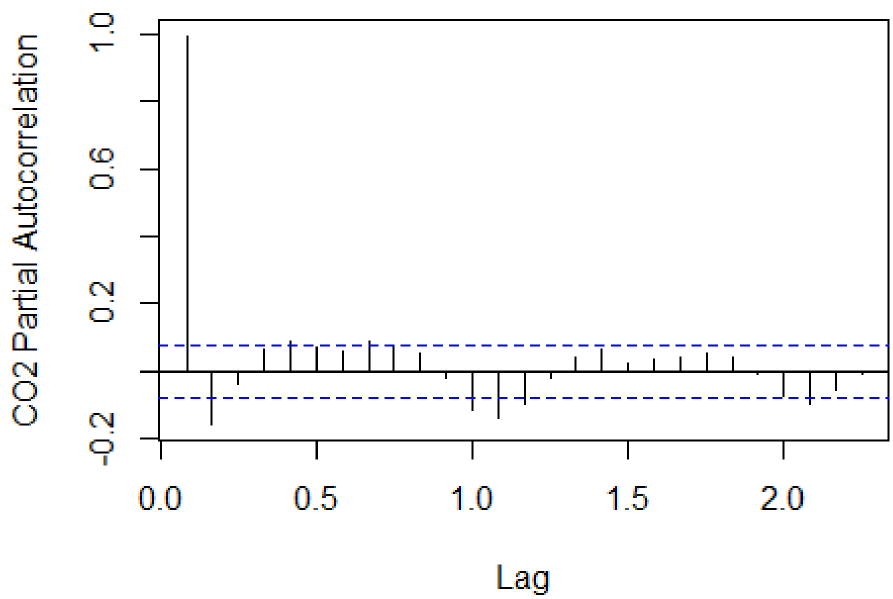




Long memory auto correlation. The persistence of high values in acf plot indicate a long term positive trend. These results are consistent with our plot of the time series.

PACF

```
pacf(co2.ts.imp,  main = "", ylab = "CO2 Partial Autocorrelation")
```



Long memory partial autocorrelation drops off around lag 20.

why doesn't x adjust with lag?

Training and test

```
co2.ts.train <- ts(co2.ts.imp[1:588], start = 1966, frequency = 12) # through
2014
co2.ts.test <- ts(co2.ts.imp[589:600], start = 2015, frequency = 12) # all of
2015
```

Seasonal ARIMA

Seasonal time series

```
sarima.mod <- auto.arima(co2.ts.train) # model through 2014
summary(sarima.mod)

## Series: co2.ts.train
## ARIMA(0,1,2)(1,1,2)[12]
##
## Coefficients:

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

##           ma1      ma2     sar1     sma1     sma2
##       -0.3543  -0.0639  -0.6315  -0.2541  -0.5608
## s.e.   0.0421   0.0423      NaN      NaN      NaN
##
```

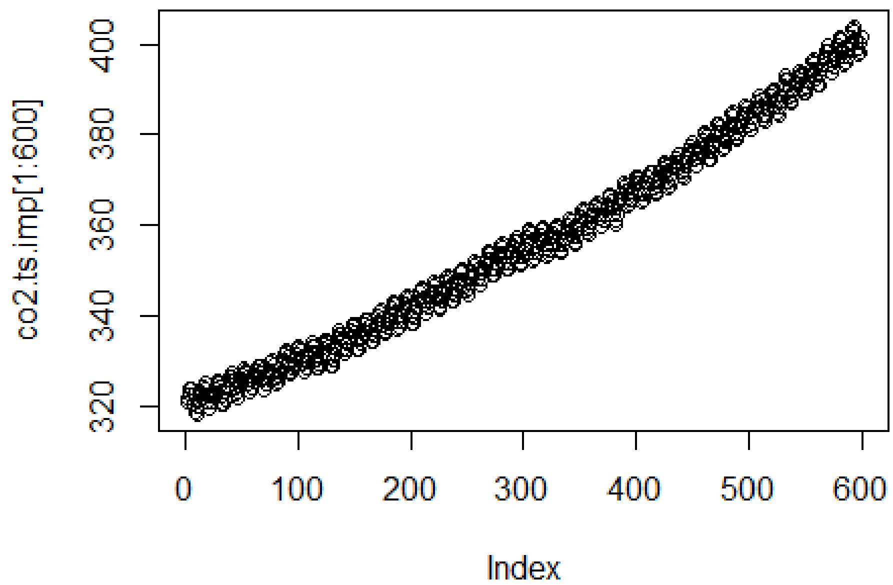
```
## sigma^2 estimated as 0.09258:  log likelihood=-131.22
## AIC=274.44   AICc=274.58   BIC=300.56
##
## Training set error measures:
##                     ME      RMSE       MAE         MPE       MAPE
## Training set 0.0220546 0.2995792 0.2327383 0.005981288 0.06555839
##                   MASE        ACF1
## Training set 0.1443602 0.008156477
```

Seasonal ARIMA Forecast

```
(sarima.2015 <- forecast(sarima.mod, h=12))
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2015       400.0454 399.6555 400.4354 399.4491 400.6418
## Feb 2015       400.8055 400.3413 401.2696 400.0956 401.5153
## Mar 2015       401.7118 401.1951 402.2284 400.9216 402.5019
## Apr 2015       403.0050 402.4407 403.5693 402.1420 403.8680
## May 2015       403.6312 403.0231 404.2394 402.7011 404.5614
## Jun 2015       402.9493 402.3002 403.5984 401.9566 403.9421
## Jul 2015       401.3714 400.6837 402.0590 400.3197 402.4230
## Aug 2015       399.2930 398.5689 400.0171 398.1856 400.4004
## Sep 2015       397.8277 397.0689 398.5865 396.6672 398.9881
## Oct 2015       397.9847 397.1927 398.7767 396.7735 399.1959
## Nov 2015       399.3993 398.5755 400.2232 398.1394 400.6593
## Dec 2015       400.8738 400.0193 401.7283 399.5669 402.1807
```

```
plot(co2.ts.imp[1:600])
lines(sarima.2015$mean, col = "green")
```



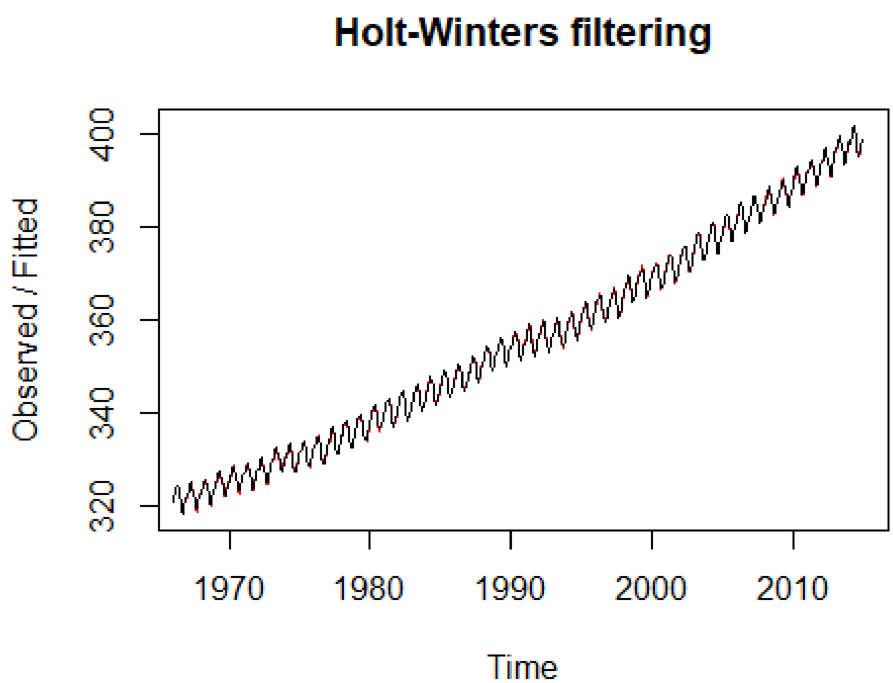
Calculate error...

Holt-Winters

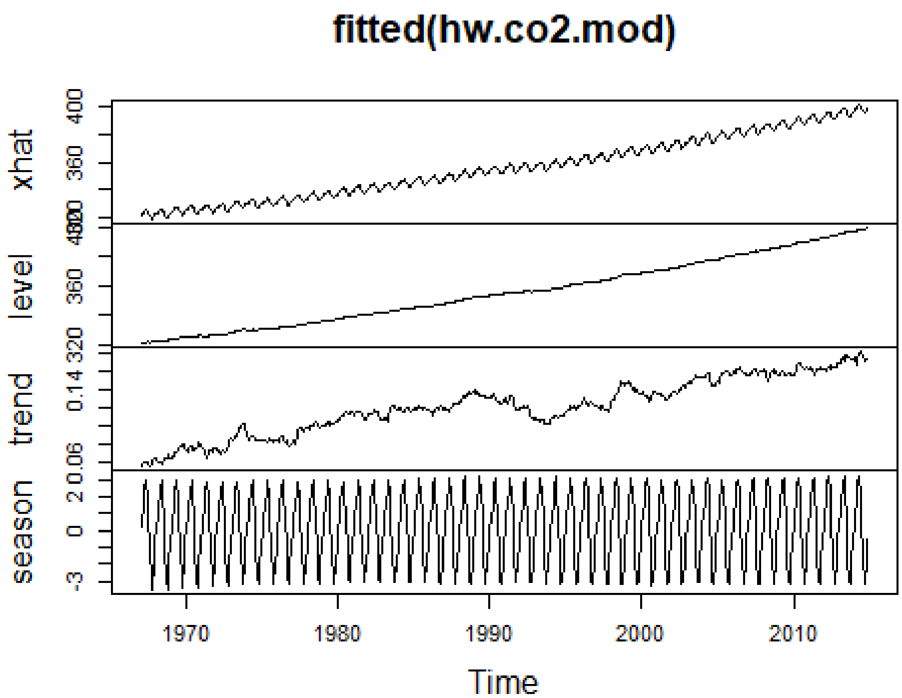
Time series with positive trend and seasonality

```
hw.co2.mod <- HoltWinters(co2.ts.train)
summary(hw.co2.mod)

##              Length Class  Mode
## fitted       2304   mts    numeric
## x             588   ts     numeric
## alpha           1   -none- numeric
## beta            1   -none- numeric
## gamma           1   -none- numeric
## coefficients   14   -none- numeric
## seasonal        1   -none- character
## SSE             1   -none- numeric
## call            2   -none- call

hw.co2.mod

## Holt-Winters exponential smoothing with trend and additive seasonal
component.
##
## Call:
## HoltWinters(x = co2.ts.train)
##
```

```
## Smoothing parameters:
##  alpha: 0.5460384
##  beta : 0.01522105
##  gamma: 0.2959677
##
## Coefficients:
##             [,1]
## a   399.4284106
## b     0.1751974
## s1    0.4492880
## s2    1.0135282
## s3    1.7472395
## s4    2.8265649
## s5    3.2747156
## s6    2.3369992
## s7    0.6081790
## s8   -1.5913489
## s9   -3.1618090
## s10  -3.1275772
## s11  -1.8776697
## s12  -0.5626363

plot(hw.co2.mod)
```




```
plot(fitted(hw.co2.mod))
```




```
# forecast
```

Read in annual temps

```
temps <- read.csv("GlobalTemperatures.csv")
#temps.ho <- read.csv("16-17_temps.csv")

names(temps)

## [1] "dt"
## [2] "LandAverageTemperature"
## [3] "LandAverageTemperatureUncertainty"
## [4] "LandMaxTemperature"
## [5] "LandMaxTemperatureUncertainty"
## [6] "LandMinTemperature"
## [7] "LandMinTemperatureUncertainty"
## [8] "LandAndOceanAverageTemperature"
## [9] "LandAndOceanAverageTemperatureUncertainty"

temps <- temps[,c("dt", "LandAndOceanAverageTemperature")]
temps$dt <- as.character(temps$dt)
```

Grab only data since 1966

```
temps <- temps[temps$dt>= "1966-01-01",]
colnames(temps) <- c("dt", "temp")
```

Streamline holdout data - 2015

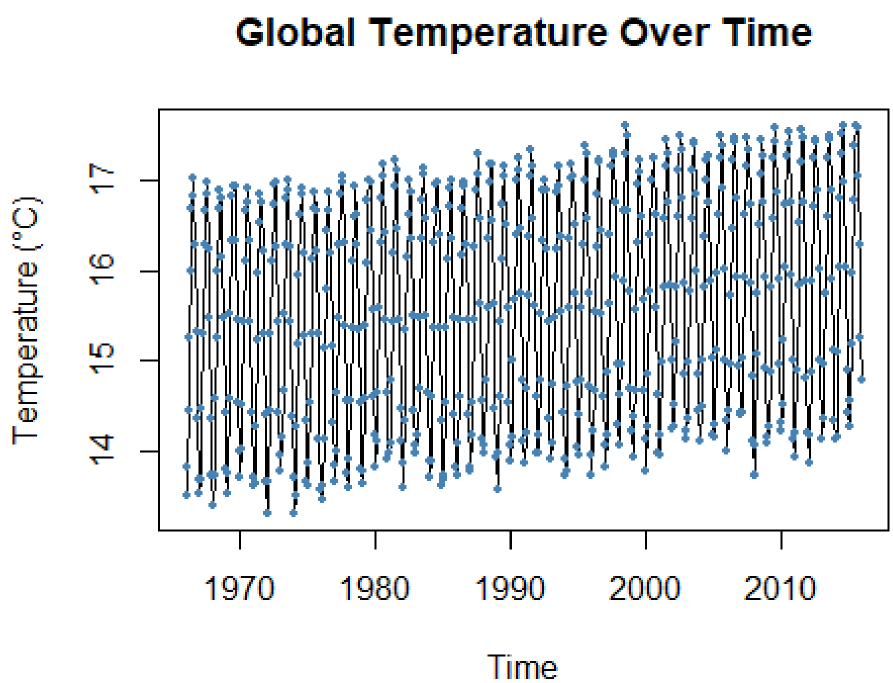
```r
# temps.ho$dt <- as.Date(as.yearmon(paste(temps.ho$year, temps.ho$month,
"01", sep = "-"))) # get date format
# temps.ho$dt <- as.character(temps.ho$dt) # match type in temps df
# temps.ho <-temps.ho[,c("dt", "raw.temp")]
# colnames(temps.ho) <- c("dt", "temp")
# temps.full <- rbind(temps, temps.ho) # make a full list
# temps.full.ts <- ts(temps.full$temp, start = 1966, frequency = 12)

temps.ts <- ts(temps$temp, start = 1966, frequency = 12)
temps.train <- temps[temps$dt<= "2014-12-01",]
temps.test <- temps[temps$dt >= "2015-01-01",]
```

Make Training Time Series

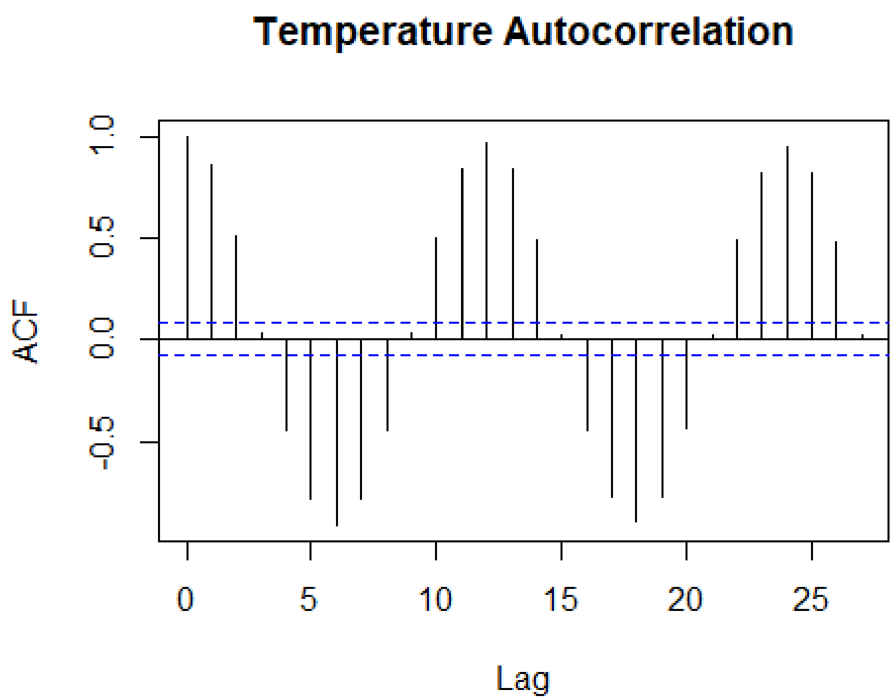
```r
temps.train.ts <- ts(temps.train$temp, start = 1966, frequency = 12)
```

Plot missing values

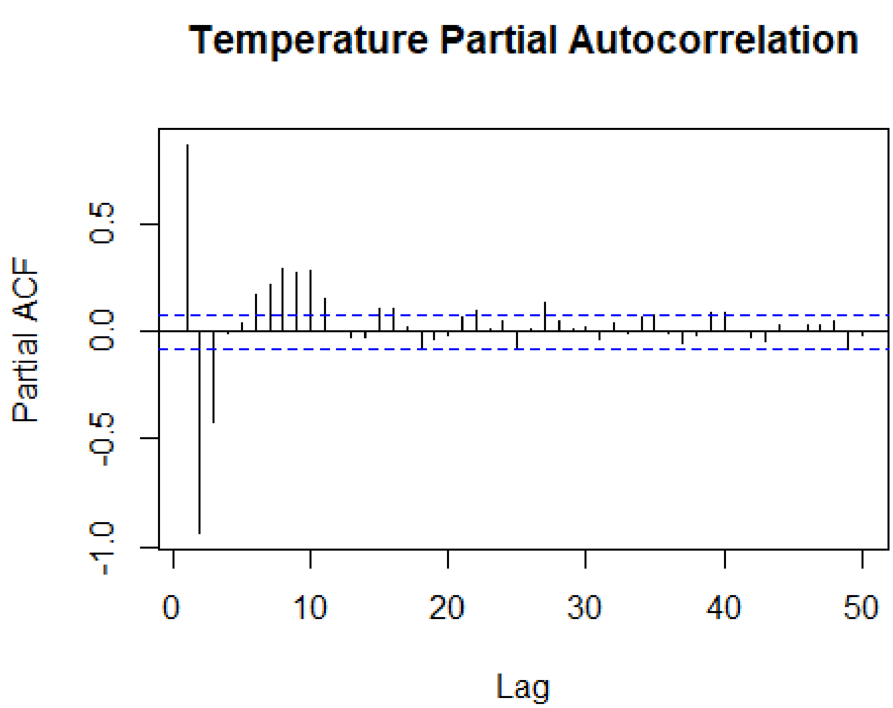
```r
plotNA.distribution(temps.ts, main = "Global Temperature Over Time", ylab =
"Temperature (°C)")
```




```r
acf(temps$temp, main = "Temperature Autocorrelation")
```




```r
pacf(temps$temp, main = "Temperature Partial Autocorrelation", lag = 50) # these are better
```



Plot both TS together

```
# library(grid)
# library(dplyr)
#
# co2$dt <- as.Date(as.yearmon(paste(co2$Year, co2$Month, "01", sep = "-")))
# get dt column in co2
# plot1 <- co2 %>%
#   select(dt, CO2) %>%
#   na.omit() %>%
#   ggplot() + geom_line(aes(x = dt, y = CO2)) + ylab("Atmospheric CO2 ppm")
+
#   theme_minimal() +
#   theme(axis.title.x = element_blank())
#
# plot2 <- temps %>%
#   select(dt, temp) %>%
#   ggplot() + geom_line(aes(x = dt, y = temp)) + ylab("Temperature (°C)") +
#   theme_minimal() +
#   theme(axis.title.x = element_blank())
#
# grid.newpage()
# grid.draw(rbind(ggplotGrob(plot1), ggplotGrob(plot2), size = "last"))

##qplot(temps.ts) + geom_line(y=temps.ts)

#ggplot(temps, aes(x=dt, y=temp)) + geom_line()
```

Seasonal ARIMA

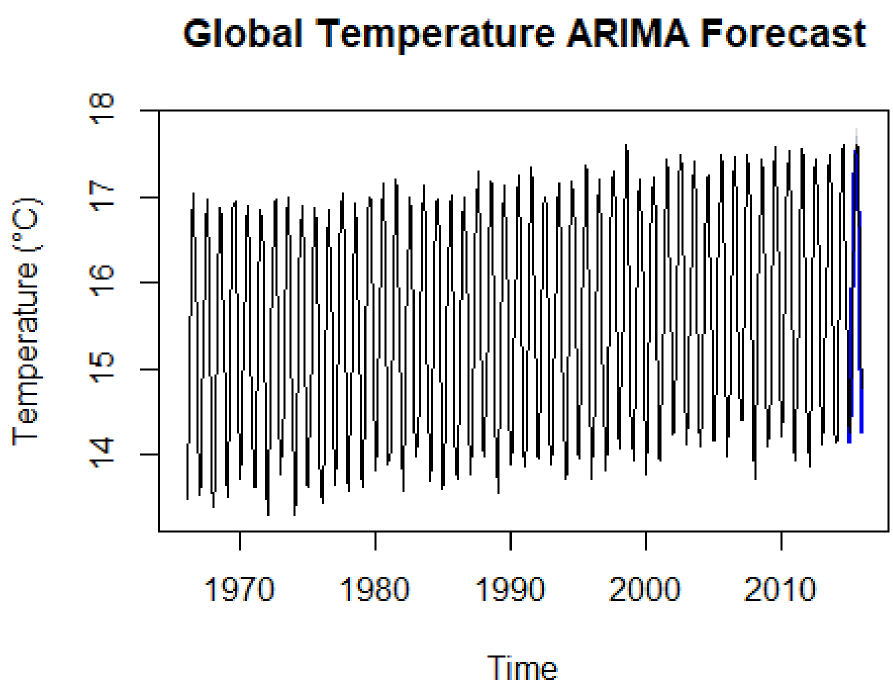
Seasonal time series

```
sarima.temps.mod <- auto.arima(temps.train.ts) # model through 2015
summary(sarima.temps.mod)
```

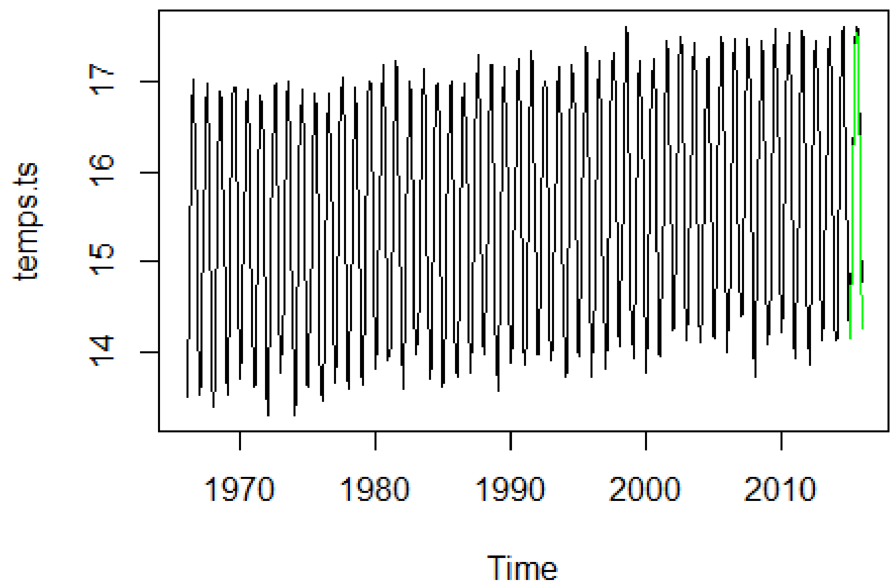
```
## Series: temps.train.ts
## ARIMA(2,0,2)(1,1,1)[12]
##
## Coefficients:
##          ar1     ar2     ma1      ma2     sar1     sma1
##       0.0556  0.7969  0.4013  -0.3348  -0.1909  -0.8239
## s.e.  0.0883  0.0862  0.1007   0.0731   0.0485   0.0352
##
## sigma^2 estimated as 0.009847:  log likelihood=507.76
## AIC=-1001.51   AICc=-1001.31   BIC=-971.02
##
## Training set error measures:
##                      ME       RMSE        MAE       MPE      MAPE
## Training set 0.01093378 0.09770333 0.07721149 0.0664059 0.5061589
##                   MASE         ACF1
## Training set 0.5345283 0.008692596
```

Seasonal ARIMA Forecast

```
(sarima.temps.2015 <- forecast(sarima.temps.mod, h=12))

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2015       14.15179 14.02462 14.27897 13.95730 14.34629
## Feb 2015       14.42931 14.28949 14.56913 14.21548 14.64314
## Mar 2015       15.06586 14.91291 15.21881 14.83194 15.29977
## Apr 2015       15.93116 15.77033 16.09200 15.68519 16.17714
## May 2015       16.66667 16.49758 16.83576 16.40807 16.92527
## Jun 2015       17.28573 17.11137 17.46009 17.01907 17.55239
## Jul 2015       17.54924 17.36942 17.72905 17.27423 17.82424
## Aug 2015       17.46753 17.28408 17.65098 17.18696 17.74810
## Sep 2015       16.86487 16.67771 17.05202 16.57864 17.15110
## Oct 2015       15.97062 15.78091 16.16033 15.68048 16.26076
## Nov 2015       14.99596 14.80369 15.18822 14.70191 15.29000
## Dec 2015       14.26590 14.07182 14.45998 13.96908 14.56272

plot(sarima.temps.2015, xlab = "Time", ylab = "Temperature (°C)", main =
"Global Temperature ARIMA Forecast")
lines(temps.ts)
```

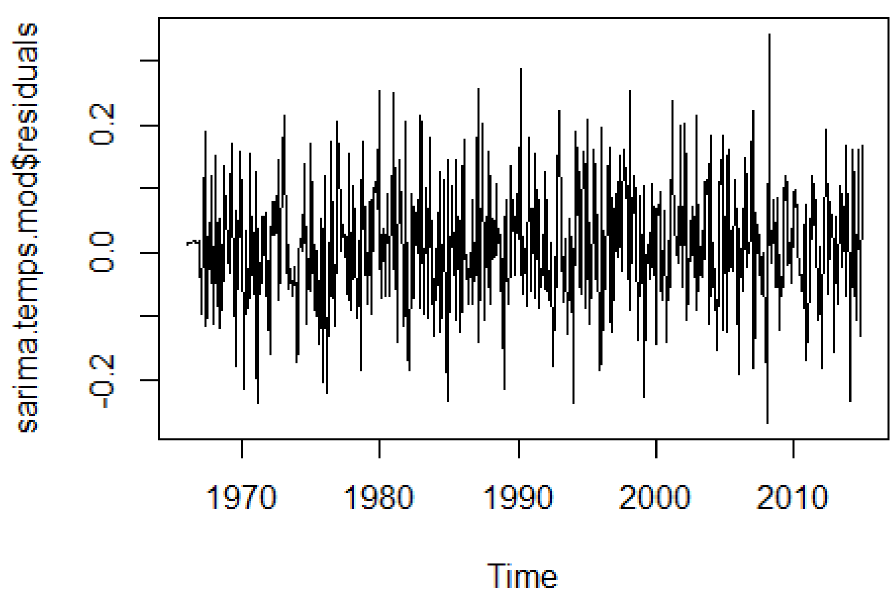



```
plot(temps.ts)
lines(sarima.temps.2015$mean, col = "green")
```

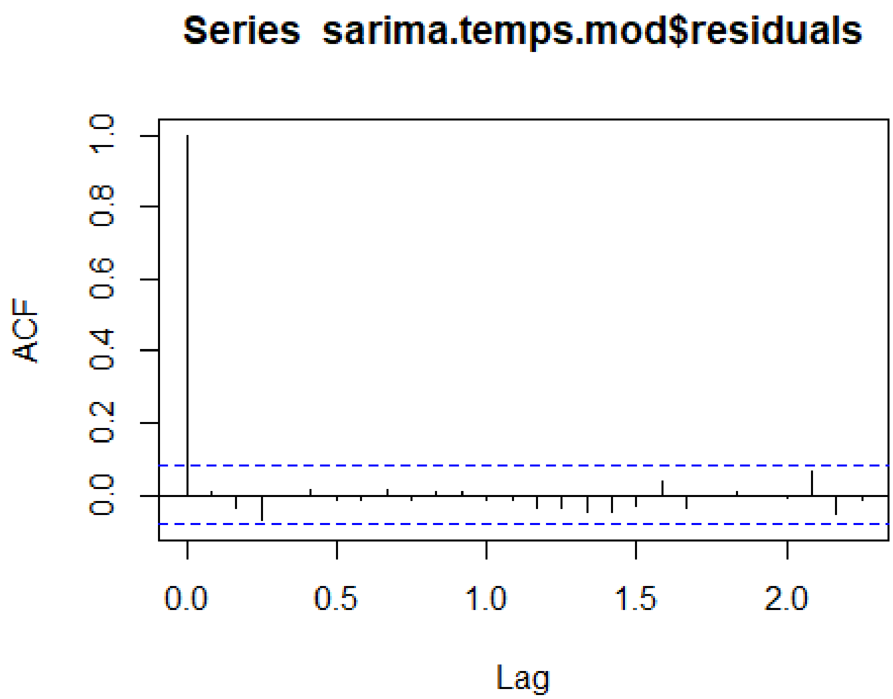


Analyze Residuals

```
plot(sarima.temps.mod$residuals)
```



```
acf(sarima.temps.mod$residuals)
```



Create function to calculate symmetric mean absolute percentage error (sMAPE) for forecast evaluation

```
sMAPE <- function(actual, estimate) {
    absDev <- abs(estimate - actual)
    return(sum(absDev/(estimate + actual))/length(actual))
}

(sMAPE.sarima.temps <- sMAPE(temps.test$temp, sarima.temps.2015$mean))

## [1] 0.005514724
```

Holt-Winters

Time series with positive trend and seasonality

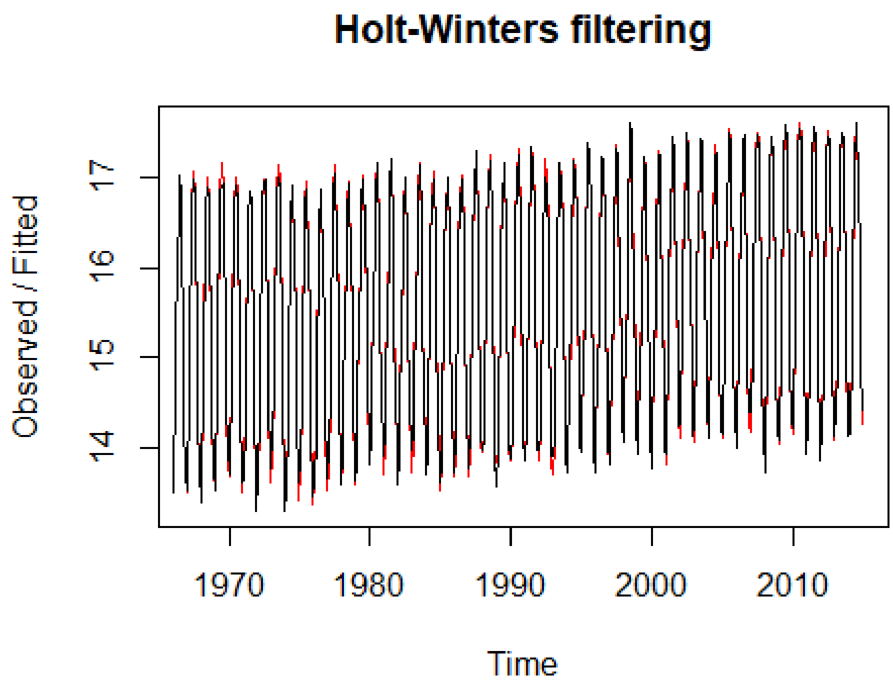
```
hw.temps.mod <- HoltWinters(temps.train.ts)
summary(hw.temps.mod)

##              Length Class  Mode
## fitted       2304   mts    numeric
## x             588   ts     numeric
## alpha           1   -none- numeric
## beta            1   -none- numeric
## gamma           1   -none- numeric
## coefficients   14   -none- numeric
```

```
## seasonal        1    -none- character
## SSE             1    -none- numeric
## call            2    -none- call
```

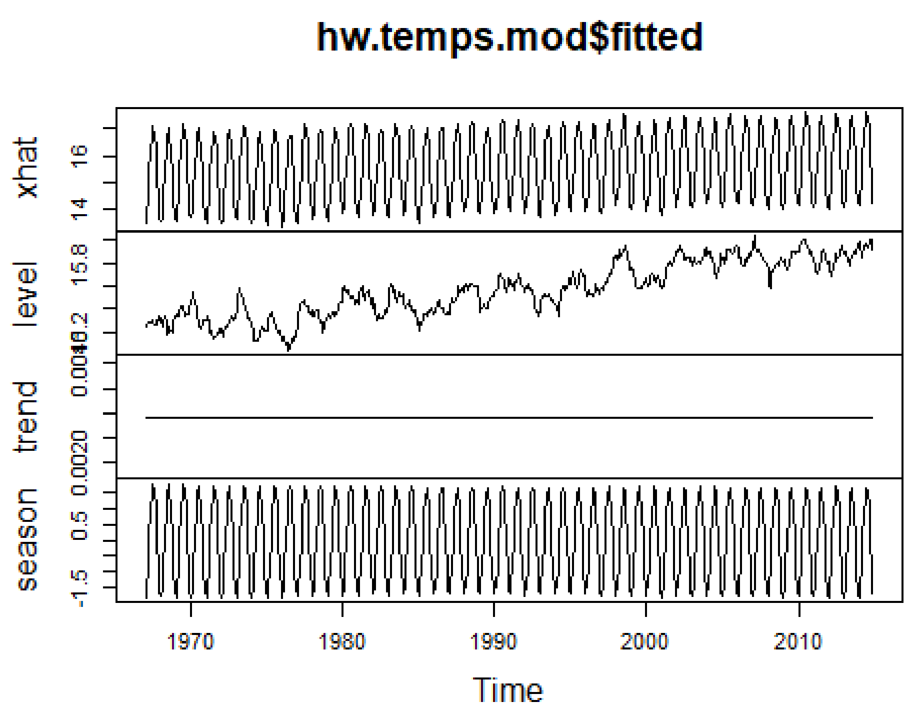
```
hw.temps.mod
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal
component.
##
## Call:
## HoltWinters(x = temps.train.ts)
##
## Smoothing parameters:
##  alpha: 0.4205287
##  beta : 0
##  gamma: 0.2717648
##
## Coefficients:
##             [,1]
## a   15.983894264
## b    0.002918124
## s1  -1.803138843
## s2  -1.557478022
## s3  -0.817346858
## s4   0.055171372
## s5   0.791421348
## s6   1.363884431
## s7   1.621384127
## s8   1.569075627
## s9   0.955293615
## s10  0.045574377
## s11 -0.948840806
## s12 -1.632386848
```

```
plot(hw.temps.mod)
```

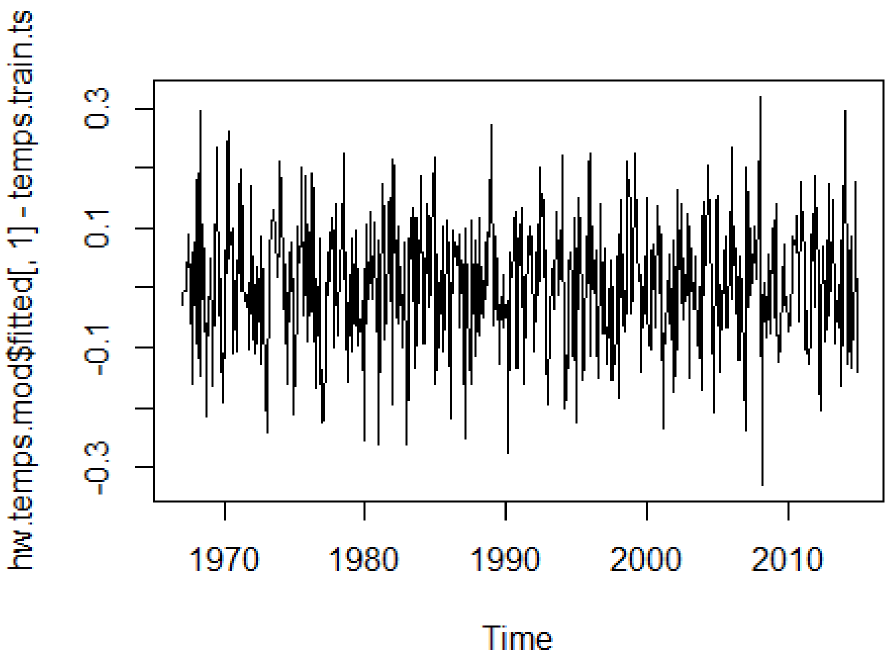



```r
plot(hw.temps.mod$fitted)
```



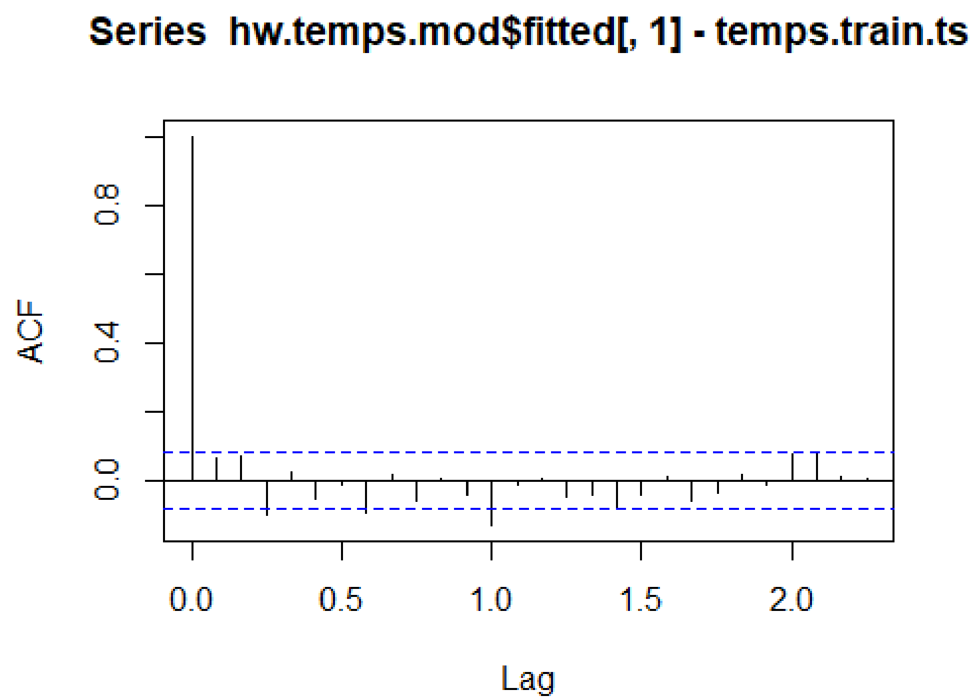
Analyzing Residuals

```r
plot(hw.temps.mod$fitted[,1]-temps.train.ts)
```



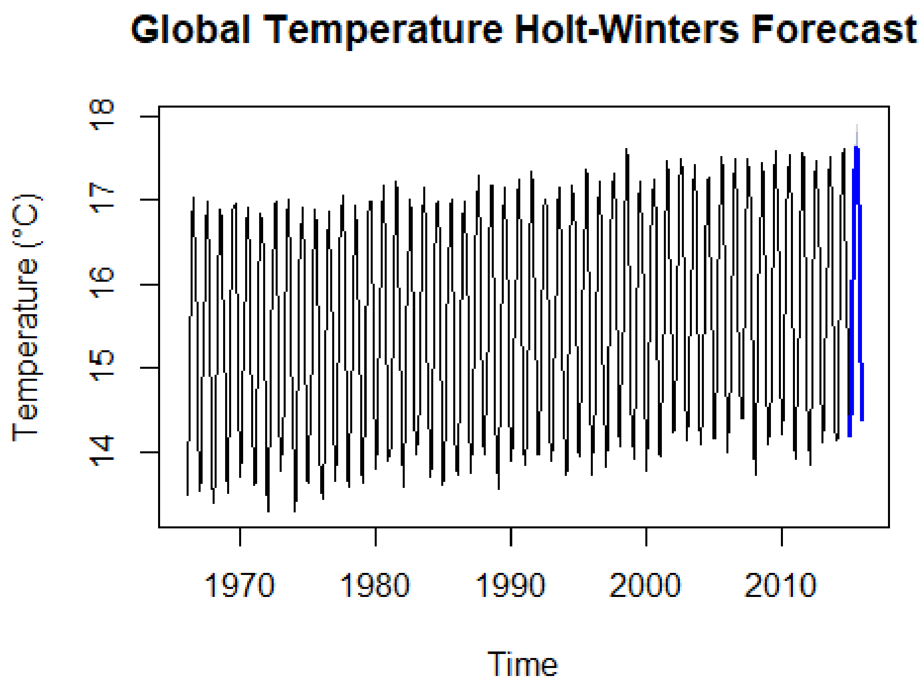
Residuals seem like white noise

```r
acf(hw.temps.mod$fitted[,1]-temps.train.ts)
```



No residual autocorrelation

```
(hw.temps.2015 <- forecast(hw.temps.mod, h=12))

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2015       14.18367 14.04888 14.31846 13.97753 14.38982
## Feb 2015       14.43225 14.28603 14.57848 14.20862 14.65588
## Mar 2015       15.17530 15.01848 15.33213 14.93546 15.41515
## Apr 2015       16.05074 15.88398 16.21749 15.79571 16.30577
## May 2015       16.78991 16.61378 16.96603 16.52055 17.05927
## Jun 2015       17.36529 17.18027 17.55031 17.08232 17.64825
## Jul 2015       17.62571 17.43220 17.81922 17.32976 17.92165
## Aug 2015       17.57631 17.37467 17.77796 17.26793 17.88470
## Sep 2015       16.96545 16.75599 17.17491 16.64511 17.28579
## Oct 2015       16.05865 15.84166 16.27564 15.72679 16.39051
## Nov 2015       15.06715 14.84288 15.29142 14.72416 15.41015
## Dec 2015       14.38652 14.15520 14.61785 14.03274 14.74030

plot(hw.temps.2015, main = "Global Temperature Holt-Winters Forecast", xlab =
"Time", ylab = "Temperature (°C)")
lines(ts(temps$temp))
```



sMAPE Calculation

```
(sMAPE.hw.2015 <- sMAPE(temps.test$temp, hw.temps.2015$mean))

## [1] 0.003446822
```