W271 Lab 3 Spring 2016

Megan Jasek, Rohan Thakur, Charles Kekeh

Friday, April 22, 2016

```
# Functions for Parts 2, 3, 4
get.best.arima <- function(x.ts, maxord = c(1, 1, 1)) {
    best.aic <- 1e+08
    all.aics <- vector()
    all.models <- vector()
    n <- length(x.ts)
    for (p in 0:maxord[1]) for (d in 0:maxord[2]) for (q in 0:maxord[3]) {
        fit <- arima(x.ts, order = c(p, d, q), method = "ML")
        fit.aic <- -2 * fit$loglik + (log(n) + 1) * length(fit$coef)
        if (fit.aic < best.aic) {
            best.aic <- fit.aic
            best.fit <- fit
            best.model <- c(p, d, q)
        }
        all.aics <- c(all.aics, fit.aic)
        all.models <- c(all.models, sprintf("(%d, %d, %d)", p,
            d, q))
    }
    list(best = list(best.aic, best.fit, best.model), others = data.frame(aics = all.aics,
        models = all.models))
}

get.best.sarima <- function(x.ts, maxord = c(1, 1, 1, 1, 1, 1),
    freq) {
    best.aic <- 1e+08
    all.aics <- vector()
    all.models <- vector()
    n <- length(x.ts)
    for (p in 0:maxord[1]) for (d in 0:maxord[2]) for (q in 0:maxord[3]) for (P in 0:maxord[3]) for (D i
        fit <- arima(x.ts, order = c(p, d, q), seasonal = list(order = c(P,
            D, Q), freq), method = "CSS", optim.control = list(maxit = 10000))
        fit.aic <- -2 * fit$loglik + (log(n) + 1) * length(fit$coef)
        if (fit.aic < best.aic) {
            best.aic <- fit.aic
            best.fit <- fit
            best.model <- c(p, d, q, P, D, Q)
        }
        all.aics <- c(all.aics, fit.aic)
        all.models <- c(all.models, sprintf("(%d, %d, %d, %d, %d, %d)",
            p, d, q, P, D, Q))
    }
    list(best = list(best.aic, best.fit, best.model), others = data.frame(aics = all.aics,
        models = all.models))
}

plot.time.series <- function(x.ts, bins = 30, name) {
    str(x.ts)
```

```
    par(mfrow = c(2, 2))
    hist(x.ts, bins, main = paste("Histogram of", name, sep = " "),
        xlab = "Values")
    plot(x.ts, main = paste("Plot of", name, sep = " "), ylab = "Values",
        xlab = "Time Period")
    acf(x.ts, main = paste("ACF of", name, sep = " "))
    pacf(x.ts, main = paste("PACF of", name, sep = " "))
}

plot.residuals.ts <- function(x.mod, model_name) {
    par(mfrow = c(1, 1))
    hist(x.mod$residuals, 30, main = paste("Histogram of", model_name,
        "Residuals", sep = " "), xlab = "Values")
    par(mfrow = c(2, 2))
    plot(x.mod$residuals, fitted(x.mod), main = paste(model_name,
        "Fitted vs. Residuals", sep = " "), ylab = "Fitted Values",
        xlab = "Residuals")
    plot(x.mod$residuals, main = paste(model_name, "Residuals",
        sep = " "), ylab = paste("Residuals", sep = " "))
    acf(x.mod$residuals, main = paste("ACF of", model_name, sep = " "))
    pacf(x.mod$residuals, main = paste("PACF of", model_name,
        sep = " "))
    Box.test(x.mod$residuals, type = "Ljung-Box")
}

estimate.ar <- function(x.ts) {
    x.ar = ar(x.ts)
    print("Difference in AICs")
    print(x.ar$aic)
    print("AR parameters")
    print(x.ar$ar)
    print("AR order")
    print(x.ar$order)
    return(x.ar)
}

plot.orig.model.resid <- function(x.ts, x.mod, orig_name, model_name,
    xlim, ylim) {
    df <- data.frame(cbind(x.ts, fitted(x.mod), x.mod$residuals))
    class(df)
    stargazer(df, type = "text", title = "Descriptive Stat",
        digits = 1)

    summary(x.ts)
    summary(x.mod$residuals)
    par(mfrow = c(1, 1))
    plot.ts(x.ts, col = "red", main = paste(orig_name, "Original vs Estimated",
        model_name, "Series with Resdiauls", sep = " "), ylab = paste(orig_name,
        "Original and Estimated Values", sep = " "), xlim = xlim,
        ylim = ylim, pch = 1, lty = 2)
    par(new = T)
    plot.ts(fitted(x.mod), col = "blue", axes = T, xlab = "",
        ylab = "", xlim = xlim, ylim = ylim, lty = 1)
```

```
    leg.txt <- c(paste(orig_name, "Original Series", sep = " "),
        "Estimated Series", "Residuals")
    legend("topleft", legend = leg.txt, lty = c(2, 1, 2), col = c("red",
        "blue", "green"), bty = "n", cex = 1)
    par(new = T)
    plot.ts(x.mod$residuals, axes = F, xlab = "", ylab = "",
        col = "green", xlim = xlim, ylim = ylim, lty = 2, pch = 1,
        col.axis = "green")
    axis(side = 4, col = "green")
    mtext("Residuals", side = 4, line = 2, col = "green")
}

plot.model.forecast <- function(x.mod, mod.fcast, orig_name,
    num_steps, x, y) {
    par(mfrow = c(1, 1))
    plot(mod.fcast, main = paste(num_steps, "-Step Ahead Forecast and",
        orig_name, "Original & Estimated Series", sep = " "),
        xlab = "Time", ylab = paste(orig_name, "Original, Estimated, and Forecasted Values",
            sep = " "), xlim = x, ylim = y, lty = 2, lwd = 1.5)
    par(new = T)
    plot.ts(fitted(x.mod), col = "blue", lty = 2, lwd = 2, xlab = "",
        ylab = "", xlim = x, ylim = y)
    leg.txt <- c(paste(orig_name, "Original Series", sep = " "),
        "Estimated Series", "Forecast")
    legend("topleft", legend = leg.txt, lty = c(2, 2, 1), lwd = c(1,
        2, 2), col = c("black", "blue", "blue"), bty = "n", cex = 1)
}
```

Part 3 (25 points): Forecast the Web Search Activity for global Warming

Data Analysis

1. The time series has weekly values (630 of them) starting at 1/4/04 and ending at 1/24/16. The minimum value is -0.551 and the maximum value is 4.104.
2. Time series plot shows that the series is very persistent, The series is basically flat from 2004 to 2012. After 2012, there is a sharp trend upward. There is more volatility after 2012. There are spikes and dips which could be seasonal with a yearly frequency. The series is not stationary in the mean.
3. Histogram shows is heavily positively skewed with most values between -0.551 and -0.3.
4. ACF of the series has correlations at around 0.75 for almost 25 lags.
5. PACF drops off immediately after first lag. There are 4 points that fall outside the 95% confidence interval (blue lines) at lags 3, 5, 11 and 14. The PACF could show some signs of seasonality.

Model Selection Process

1. **Try AR models.** Use the ar() command in R to find AR(p) models or order p that potentially fit the time series. This command output a model or order 15, but looking at the difference in AICs, the AIC for the AR(1) model is not that different (only 29.85 point away) from the AIC of the AR(15), so for parsimony we will try using that one. Check if the residuals look like white noise.

- Histogram: Yes. This looks like a normal distribution.

- Fitted vs. Residuals: No. The plot does not look like an evenly distributed cloud.
- Plot: No. The plot does not look random, there is a lot of volatility on the right hand side of the graph.
- ACF: No. The ACF drops off after lag 0, but has only a few lags where the correlation comes out of the 95% confidence interval (CI)
- PACF: No. The PACF shows correlation with several values outside of the 95% CI. In summary, the residuals for this model do not look like white noise, so there is more variation that could be explained by our model. The In-Sample fit of this estimated model matches the original model very well as evidenced in the plot.

2. **Try ARIMA models.** Use the get.best.arima() function which will try models with c(p,d,q) where p=0-4, d=0-2 and q=0-2. And then we can print out a list in ascending order by AIC of the 20 models with the lowest AIC. And then inspect these models for parsimony and select one with a good AIC and a small number of parameters. The best model output from the function had an AIC of -1058.794 with parameters = c(1, 2, 2). For parsimony a model of ARIMA(1,1,1) was chosen with an AIC of -1032.364 which is not that different from the best AIC. Check if the residuals look like white noise. No, the residuals do not look like white noise. They exhibit the same characteristics as the AR(1) model from step 1. The In-Sample fit of this estimated model matches the original model very well as evidenced in the plot.

3. **Try SARIMA models.** From the plot of the original series, it looks like this series has a seasonal component with a 52-week periodicity. Use the get.best.sarima() function with parameters c(2,2,2,2,2,2). The best AIC output is -1276.817 with a model of SARIMA(1,2,2,1,0,2). For parsimony try running get.best.sarima() with c(1,1,1,1,1,1). A parsimonious model from this output is SARIMA(0,1,1,1,0,1) with AIC -1246.412 which is very close to the AIC output from c(2,2,2,2,2,2). For parsimony we will choose SARIMA(0,1,1,1,0,1) and check the residuals. No, the residuals do not look like white noise. They exhibit the same characteristics as the AR(1) model from step 1. The residuals plot exhibits evidence of time-varying volatility. The In-Sample fit of this estimated model matches the original model very well as evidenced in the plot.

4. **Try using GARCH.** Since the residuals exhibit evidence of time-varying volatility, we will try to use GARCH to model that. A GARCH model is fit with the residuals from the SARIMA(0,1,1,1,0,1) model from step 3. Looking at the residuals of the GARCH model, the square of the residuals is still not completely inside the 95% CI indicating that there is still time-varying volatility present. Since we haven't found a model with a satisfactory fit, we will look at only modeling part of the original time series.

```r
# Read in the time series data
glob.warm = read.csv("globalWarming.csv", header = TRUE)
glob.warm.ts = ts(glob.warm$data.science, start = 2004, frequency = 52)
# Print descriptive statistics
str(glob.warm.ts)
```

```
##  Time-Series [1:630] from 2004 to 2016: -0.44 -0.474 -0.423 -0.551 -0.486 -0.551 -0.453 -0.462 -0.55
```

```r
summary(glob.warm.ts)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.551000 -0.506000 -0.485000  0.000038 -0.200000  4.104000
```

```r
cbind(head(glob.warm.ts), tail(glob.warm.ts))
```

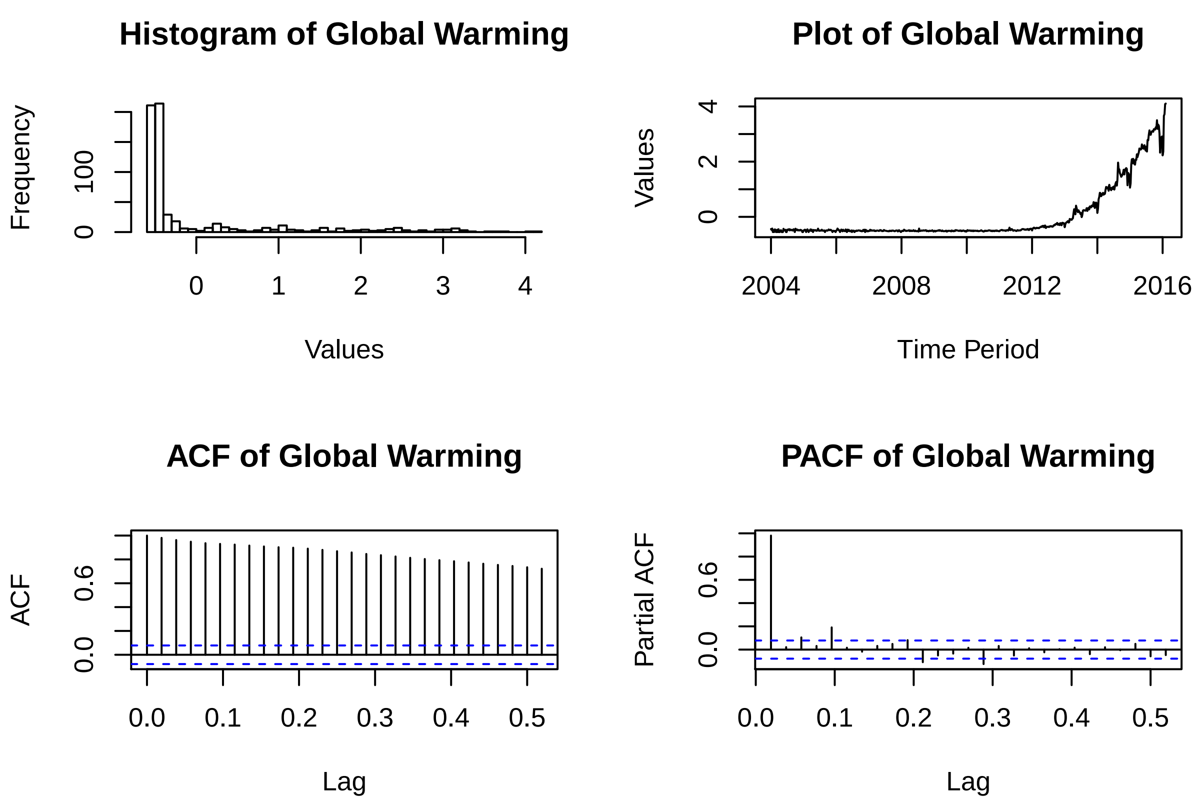
```
##        [,1]  [,2]
## [1,] -0.440 2.227
## [2,] -0.474 2.360
## [3,] -0.423 3.662
## [4,] -0.551 3.721
## [5,] -0.486 4.087
## [6,] -0.551 4.104
```

```
quantile(as.numeric(glob.warm.ts), c(0.01, 0.05, 0.1, 0.25, 0.5,
    0.75, 0.9, 0.95, 0.99))
```

```
##       1%       5%      10%      25%      50%      75%      90%      95%
## -0.55100 -0.53220 -0.51900 -0.50600 -0.48500 -0.20000  1.68410  2.48055
##      99%
##  3.28021
```

```
# Plot the time series
plot.time.series(glob.warm.ts, 50, "Global Warming")
```

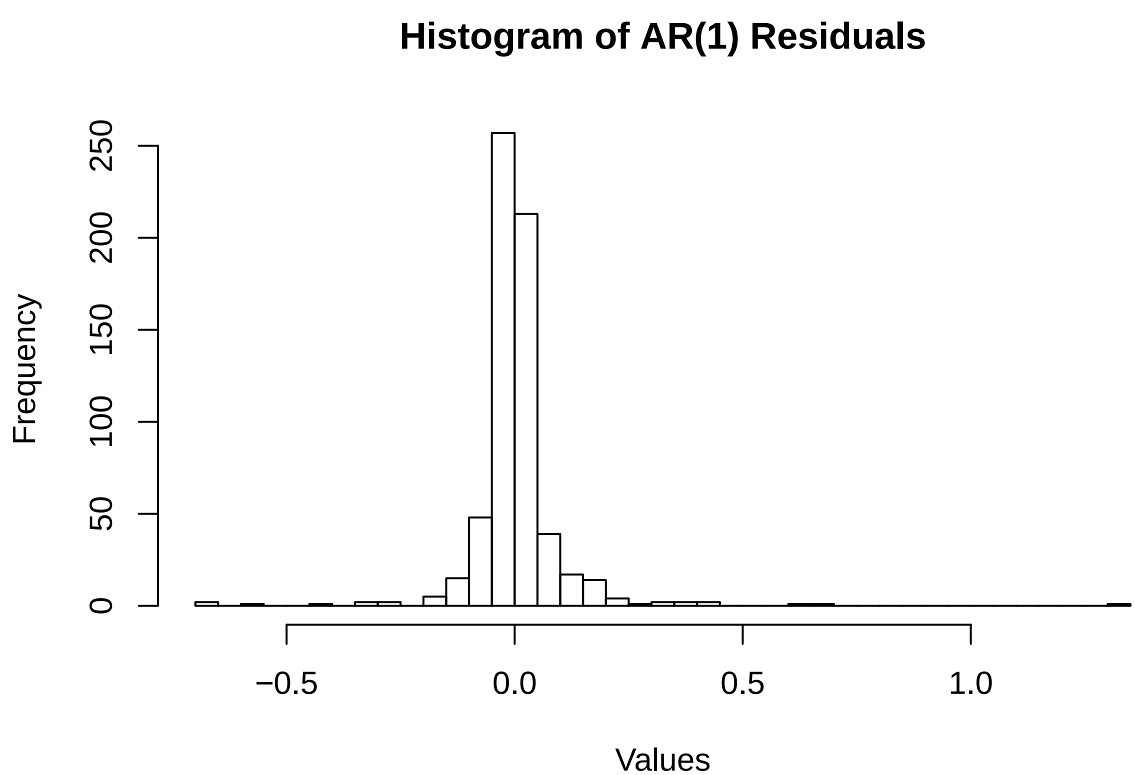
```
##  Time-Series [1:630] from 2004 to 2016: -0.44 -0.474 -0.423 -0.551 -0.486 -0.551 -0.453 -0.462 -0.55
```

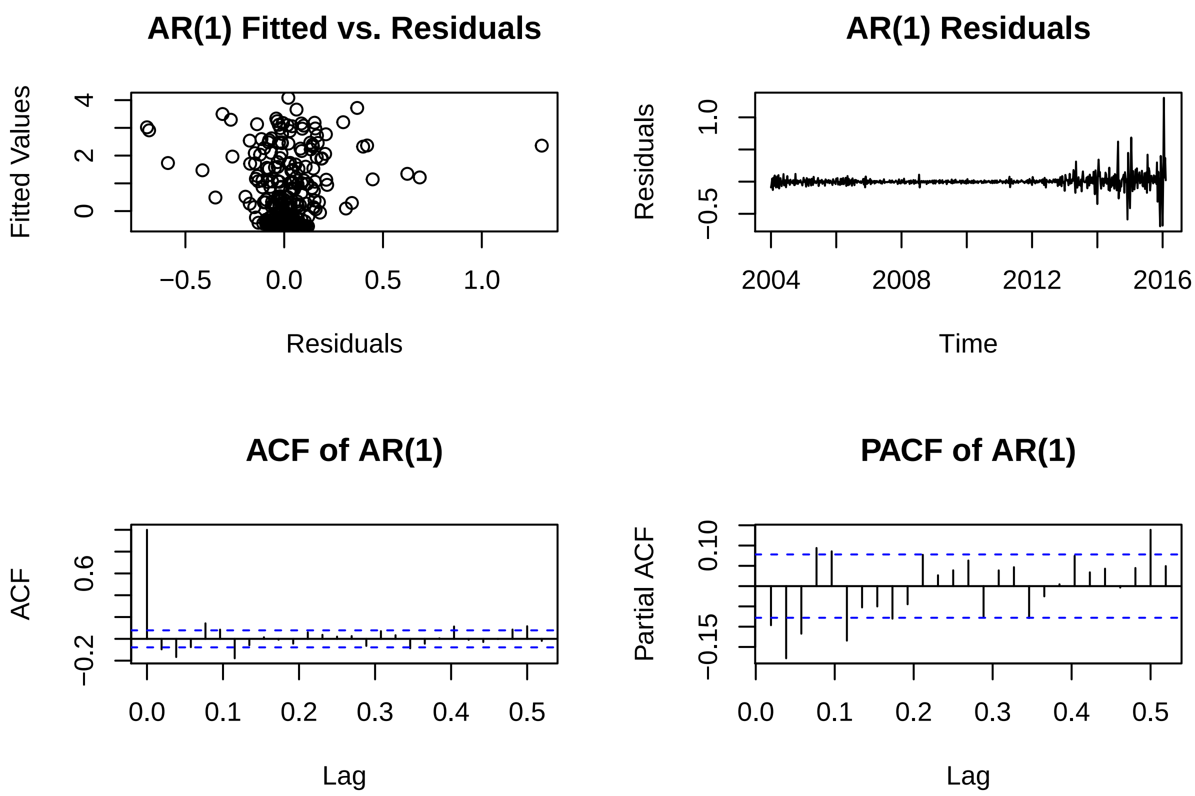



```
### 1. Try AR models
glob.warm.ar = estimate.ar(glob.warm.ts)
```

```
## [1] "Difference in AICs"
##           0           1           2           3           4           5 
## 2084.447812   29.847743   31.560248   26.579621   27.960796    6.553854 
##           6           7           8           9          10          11 
##    8.386263   10.176681   11.540473   12.035569    9.848063    4.382889 
##          12          13          14          15          16          17 
##    4.754476    5.996066    7.842039    0.000000    1.380591    1.728222 
##          18          19          20          21          22          23 
##    3.638626    5.291781    7.280008    9.104280   10.136039   11.875658 
##          24          25          26          27 
##   13.856501   14.302766   14.191716   14.781132 
## [1] "AR parameters"
##  [1]  0.944522755 -0.084770519  0.084153344 -0.171500315  0.188422207
##  [6]  0.058499722 -0.055671998 -0.008980095 -0.033122819  0.204945468
## [11] -0.084654024 -0.006099723 -0.059202741  0.132988289 -0.124502569
## [1] "AR order"
## [1] 15
```

```
glob.warm.ar1 = arima(glob.warm.ts, order = c(1, 0, 0))
# Plot the residuals
plot.residuals.ts(glob.warm.ar1, "AR(1)")
```

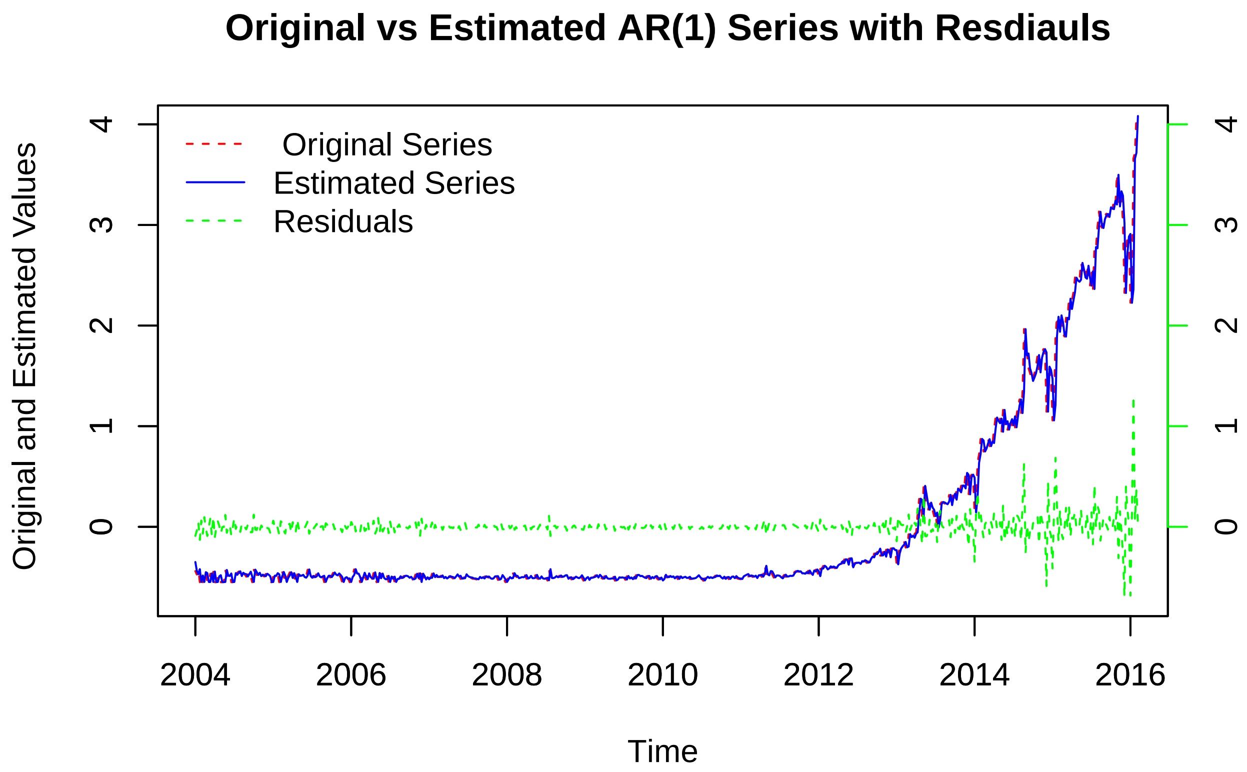




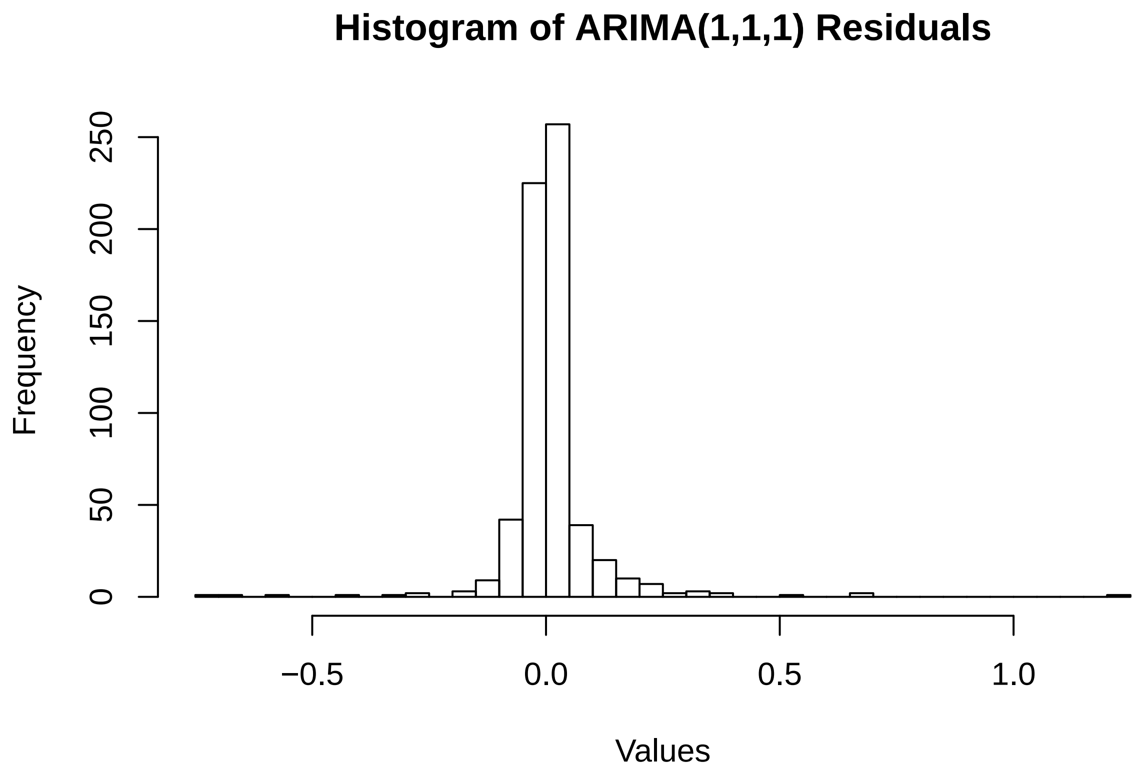

```
##
##  Box-Ljung test
##
## data:  x.mod$residuals
## X-squared = 5.8789, df = 1, p-value = 0.01532
```

```
# Plot the In-sample fit
plot.orig.model.resid(glob.warm.ts, glob.warm.ar1, "", "AR(1)",
    c(2004, 2016), c(-0.7, 4))
```

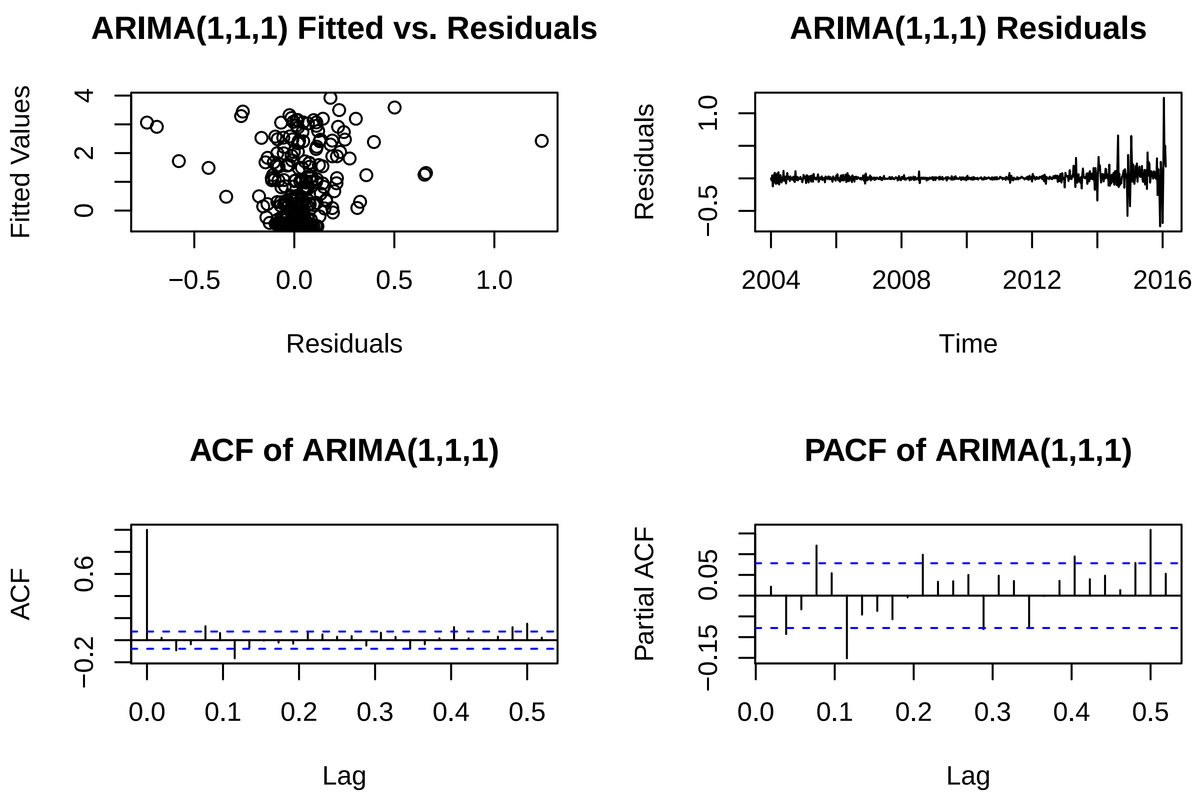
```
##
## Descriptive Stat
## ============================================
## Statistic         N   Mean  St. Dev. Min  Max
## --------------------------------------------
## x.ts             630 0.000    1.0    -0.6 4.1
## fitted.x.mod.    630 -0.01    1.0    -0.5 4.1
## x.mod.residuals 630 0.01     0.1    -0.7 1.3
## --------------------------------------------
```




```
### 2. Try ARIMA models gw.arima.best <-
### get.best.arima(glob.warm.ts, maxord=c(4,2,2)) Print the top
### 20 best models based on AIC
### gw.arima.best$others[order(gw.arima.best$others$aics)[1:20],]
glob.warm.arima = arima(glob.warm.ts, order = c(1, 1, 1))
# Plot the residuals
plot.residuals.ts(glob.warm.arima, "ARIMA(1,1,1)")
```



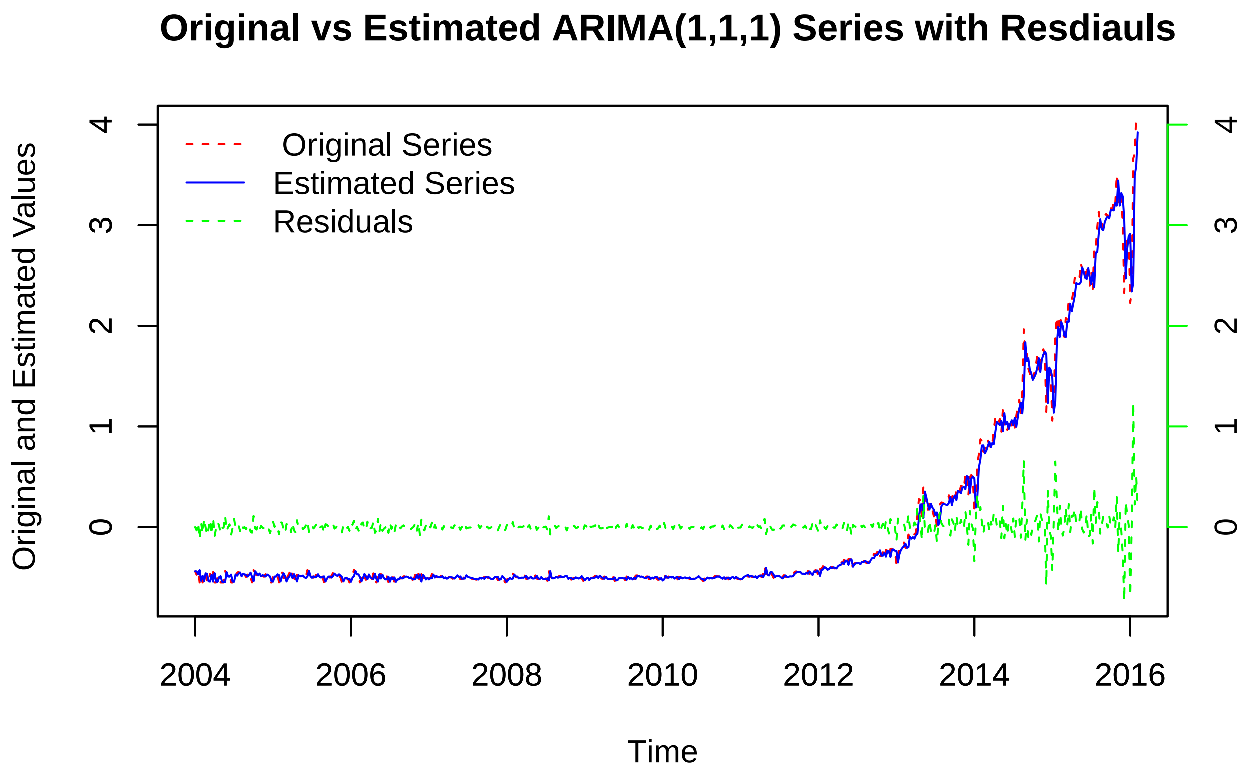
Histogram of ARIMA(1,1,1) Residuals
Frequency
0
50
100
150
200
250
−0.5
0.0
0.5
1.0
Values




```
##
##  Box-Ljung test
##
## data:  x.mod$residuals
## X-squared = 0.29725, df = 1, p-value = 0.5856
```

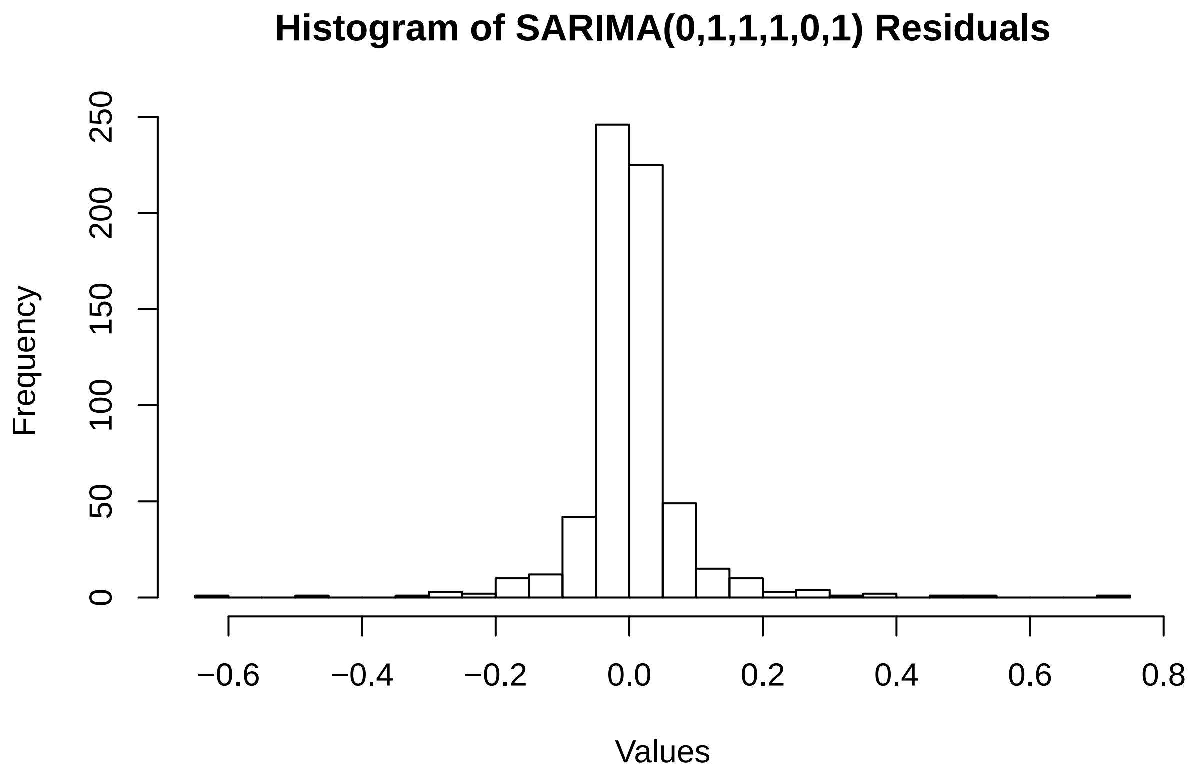
```
# Plot the In-sample fit
plot.orig.model.resid(glob.warm.ts, glob.warm.arima, "", "ARIMA(1,1,1)",
    c(2004, 2016), c(-0.7, 4))
```

```
##
## Descriptive Stat
## ===========================================
## Statistic        N   Mean  St. Dev. Min  Max
## -------------------------------------------
## x.ts             630 0.000   1.0    -0.6 4.1
## fitted.x.mod.    630 -0.01   1.0    -0.5 3.9
## x.mod.residuals  630 0.01    0.1    -0.7 1.2
## -------------------------------------------
```

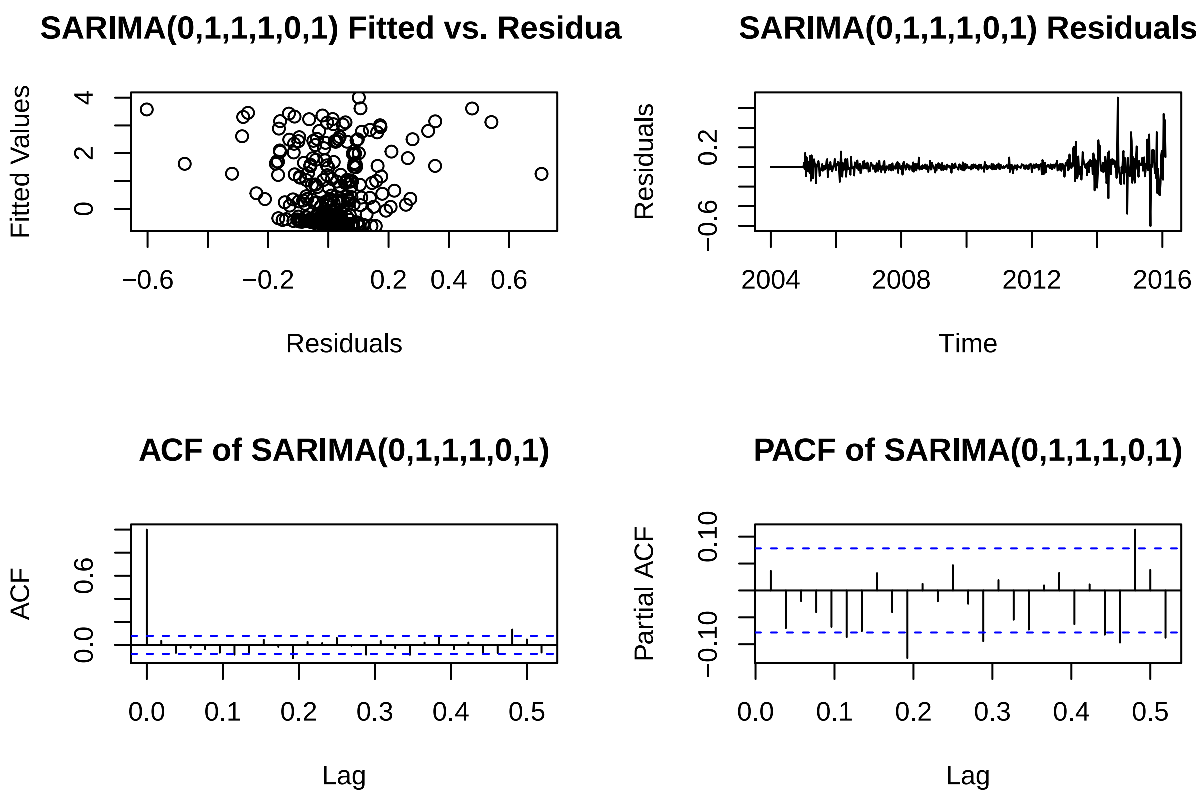



```r
### 3. Try SARIMA models gw.seas.best <-
### get.best.sarima(glob.warm.ts, maxord=c(2,2,2,2,2,2), 52)
### Print the top 20 best models based on AIC
### gw.seas.best$others[order(gw.seas.best$others$aics)[1:20],]

# gw.seas.best1 <- get.best.sarima(glob.warm.ts,
# maxord=c(1,1,1,1,1,1), 52) Print the top 20 best models
# based on AIC
# gw.seas.best1$others[order(gw.seas.best1$others$aics)[1:20],]
glob.warm.arima.seas = arima(glob.warm.ts, order = c(0, 1, 1),
    seas = list(order = c(1, 0, 1), 52), method = "CSS")
# Plot the residuals
glob.warm.arima.seas.res = glob.warm.arima.seas$residuals
plot.residuals.ts(glob.warm.arima.seas, "SARIMA(0,1,1,1,0,1)")
```

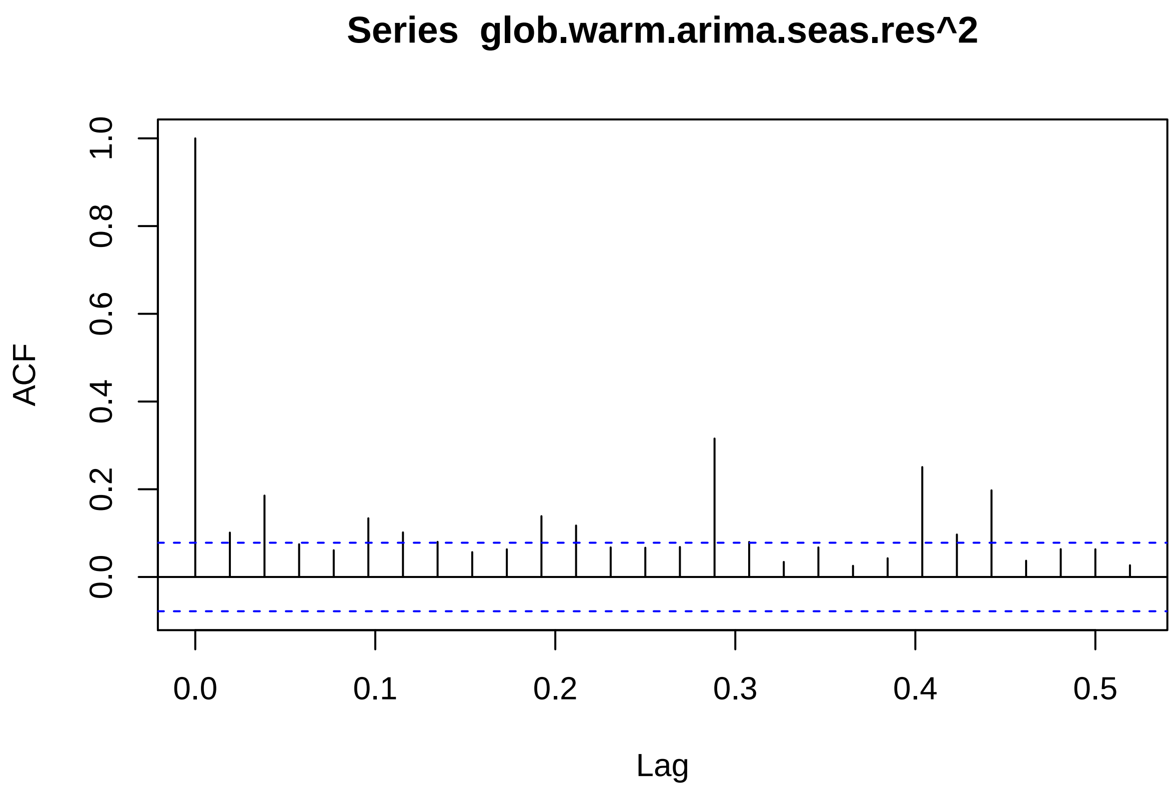


Histogram of SARIMA(0,1,1,1,0,1) Residuals
Frequency
0
50
100
150
200
250
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
Values




```
##
##  Box-Ljung test
##
## data:  x.mod$residuals
## X-squared = 0.8408, df = 1, p-value = 0.3592
```

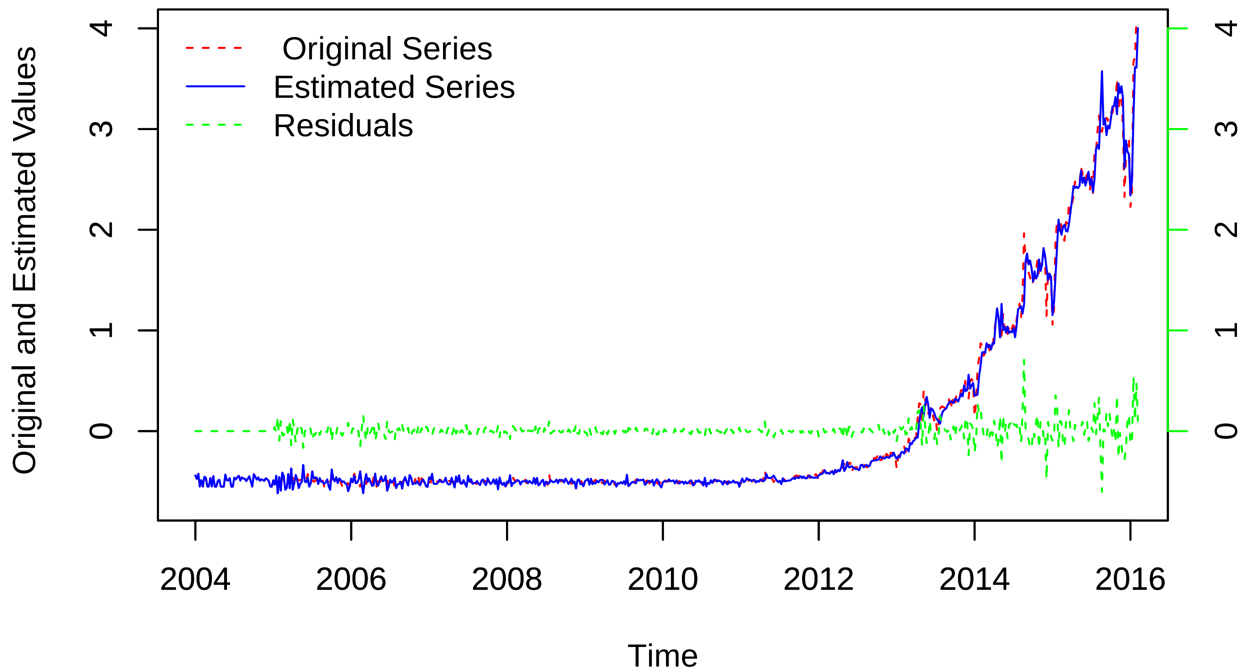
```
par(mfrow = c(1, 1))
acf(glob.warm.arima.seas.res^2)
```



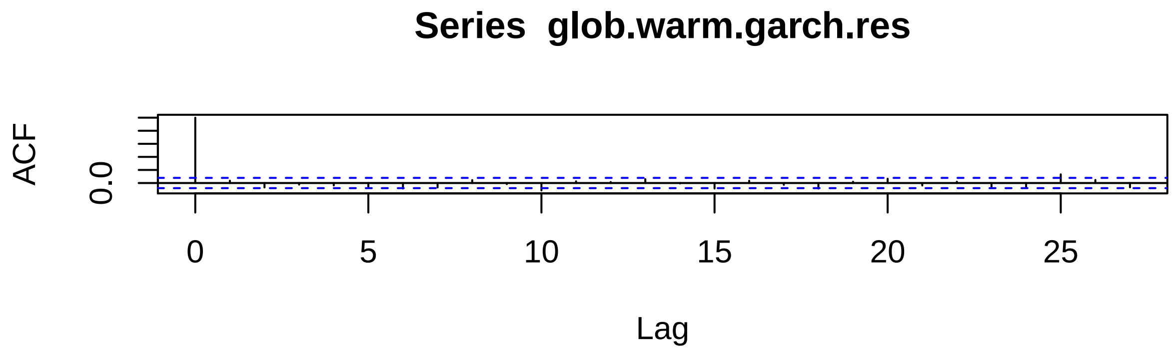

```r
# Plot the In-sample fit
plot.orig.model.resid(glob.warm.ts, glob.warm.arima.seas, "",
    "SARIMA(0,1,1,1,0,1)", c(2004, 2016), c(-0.7, 4))
```

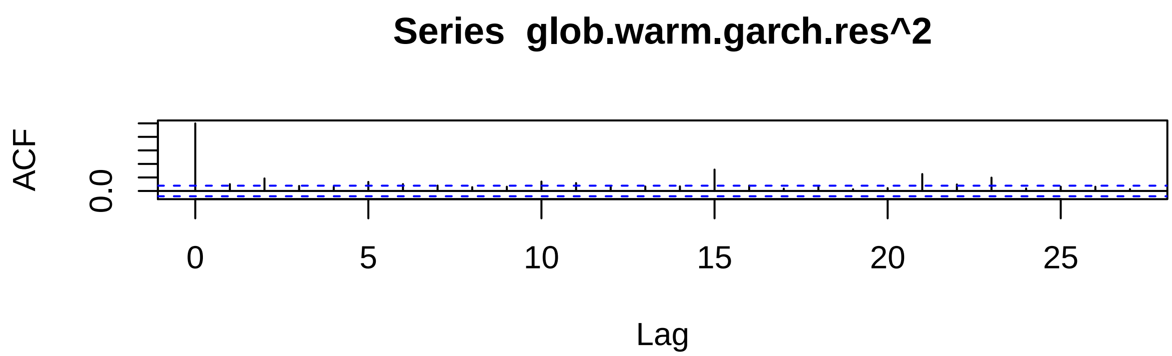
```
##
## Descriptive Stat
## ============================================
## Statistic        N   Mean   St. Dev. Min  Max
## --------------------------------------------
## x.ts            630 0.000   1.0     -0.6 4.1
## fitted.x.mod.   630 -0.01   1.0     -0.6 4.0
## x.mod.residuals 630 0.01    0.1     -0.6 0.7
## --------------------------------------------
```

Original vs Estimated SARIMA(0,1,1,1,0,1) Series with Resdiauls




```
### 4. Try using GARCH.
glob.warm.garch.fit = garchFit(~garch(1, 1), data = glob.warm.arima.seas.res,
    trace = FALSE)
glob.warm.garch.res <- glob.warm.garch.fit@residuals
par(mfrow = c(2, 1))
acf(glob.warm.garch.res)
acf(glob.warm.garch.res^2)
```





5. **Using a portion of the data.** Since no satisfactory model was found using the full data series, using a portion of the data will be considered. The data has a clear split around 2012 or 2013 where it goes from being stationary in the mean to being non-stationary in the mean. Since we are interested in forecasting this information after 2016, we will try creating a model for the latter part of the data, the part that contains the most recent information and then forecasting after that. 2012 could have been chosen, but it still had some of the non-trending data contained in it, so 2013 was chosen as a start year. We will repeat the same analysis as above. When graphs are created with this 2013-2016 series, they will be denoted by the phrase "Abrv. Original" instead of "Original"

6. **Try AR models.** Use the ar() command in R to find AR(p) models or order p that potentially fit the time series. This command output a model or order 1. Check if the residuals look like white noise.

- Histogram: Yes. This looks like a normal distribution.
- Fitted vs. Residuals: Yes. The plot looks like an evenly distributed cloud.
- Plot: Yes. The plot looks mostly like white noise. There is a little more volatility on the right hand side of the graph.
- ACF: No. The ACF drops off after lag 0, but has only a few lags where the correlation comes out of the 95% confidence interval (CI)
- PACF: No. The PACF shows correlation with a few values outside of the 95% CI. In summary, the residuals for this model do not look like white noise, so there is more variation that could be explained by our model.

7. **Try ARIMA models.** Use the get.best.arima() function to find the best model. The best model output from the function had an AIC of -25.091124 with parameters = c(0, 1, 0). An ARIMA(0,1,0) model was created. Check if the residuals look like white noise. No, the residuals do not look like white

noise. They exhibit the same characteristics as the AR(1) model from step 6. The In-Sample fit of this estimated model matches the original model very well as evidenced in the plot.

8. **Try SARIMA models.** Use the get.best.sarima() function. The best AIC output is -90.17105 with parameters c(0,1,1,1,0,1), but for parsimony we will choose a SARIMA(0,1,0,0,0,1) with an AIC of -76.45286 which is very close to the other model. Check the residuals. Yes, the residuals look basically like white noise. There is one place in the squared residuals where the value exceeds the 95% confidence interval. The In-Sample fit of this estimated model. The model now has a satisfactory fit and we will move on to backtesting and forecasting.
9. **Backtesting.** For backtesting, 10% of the values from the end of the 2013-2016 time series were withheld, in this case 10 values. The backtesting model shows mean predicted values that follow the up and down changes of the original time series, but the mean predicted values are are not as extreme as the orignal values. The seasonality of the original series is being modeled to some extent. We also note that the original series, for the most part, falls within the 95% confidence interval of the forecast, giving us confidence that this model could be used as a decent predictive model for the original time series.
10. **Forecast the model.** Using the SARIMA(0,1,0,0,0,1) model with the 2013-2016 version of the time series, we made the requested 12-step ahead forecast of the model. The forecast looks like it captures the seasonality of the model as it matches the upward trend and the seasonal volatility. We also note that all of the forecasted values are within the 80% confidence interval of the prediction.

Conclusion

MJ WILL FINISH WRITING LATE. The 2013-2016 abbreviated time series is satisfactorily modeled with a SARMIMA(x,x,x,x,x,x) model to hande trends and seasonality. We observe that the forecasts of the model are consisent with the seasonality of the original series and the forecasts are within the 95% confidence interval.

```r
### 5. Using a portion of the data Create 2013 to 2016 series
glob.warm.2013.ts = ts(glob.warm.ts[471:length(glob.warm.ts)],
    start = 2013, frequency = 52)
# Create a string label to prepend to the word 'Original' for
# proper labeling in graphs
part_name = "Abrv."
# Rename series
glob.warm.part.ts = glob.warm.2013.ts
# Descriptive statistics
str(glob.warm.part.ts)
```

```
##  Time-Series [1:160] from 2013 to 2016: -0.218 -0.196 -0.179 -0.152 -0.206 -0.198 -0.074 -0.104 -0.0
```

```r
summary(glob.warm.part.ts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.2180  0.3378  1.1370  1.4080  2.3610  4.1040
```

```r
cbind(head(glob.warm.part.ts), tail(glob.warm.part.ts))
```

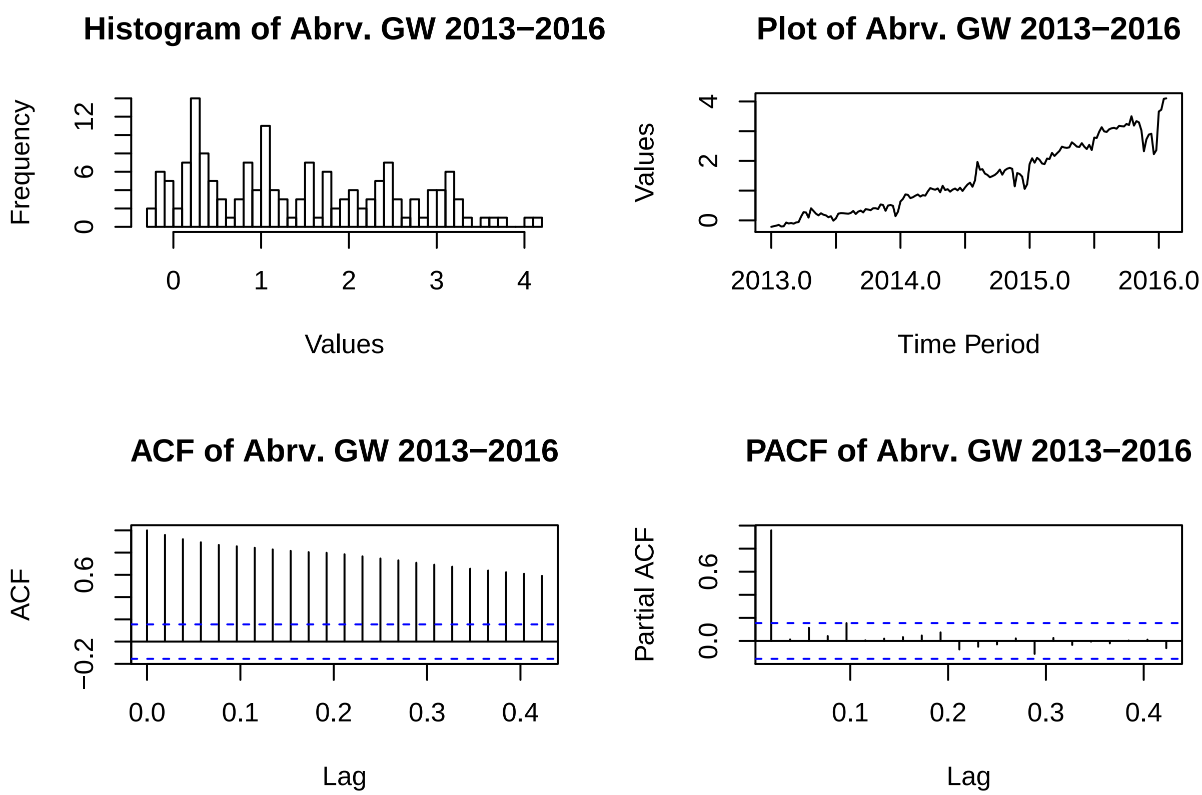
```
##        [,1]  [,2]
## [1,] -0.218 2.227
## [2,] -0.196 2.360
## [3,] -0.179 3.662
## [4,] -0.152 3.721
## [5,] -0.206 4.087
## [6,] -0.198 4.104
```

```r
quantile(as.numeric(glob.warm.part.ts), c(0.01, 0.05, 0.1, 0.25,
    0.5, 0.75, 0.9, 0.95, 0.99))
```

```
##        1%        5%       10%       25%       50%       75%       90%       95%
## -0.20128 -0.08975  0.12390  0.33775  1.13700  2.36125  3.08360  3.20790
##       99%
##   3.87106
```

```r
# Plot the time series
plot.time.series(glob.warm.part.ts, 50, "Abrv. GW 2013-2016")
```

```
##  Time-Series [1:160] from 2013 to 2016: -0.218 -0.196 -0.179 -0.152 -0.206 -0.198 -0.074 -0.104 -0.0
```




```r
### 6. Try AR models.  Use the ar function to find an ar
### estimate
glob.warm.ar = estimate.ar(glob.warm.part.ts)
```

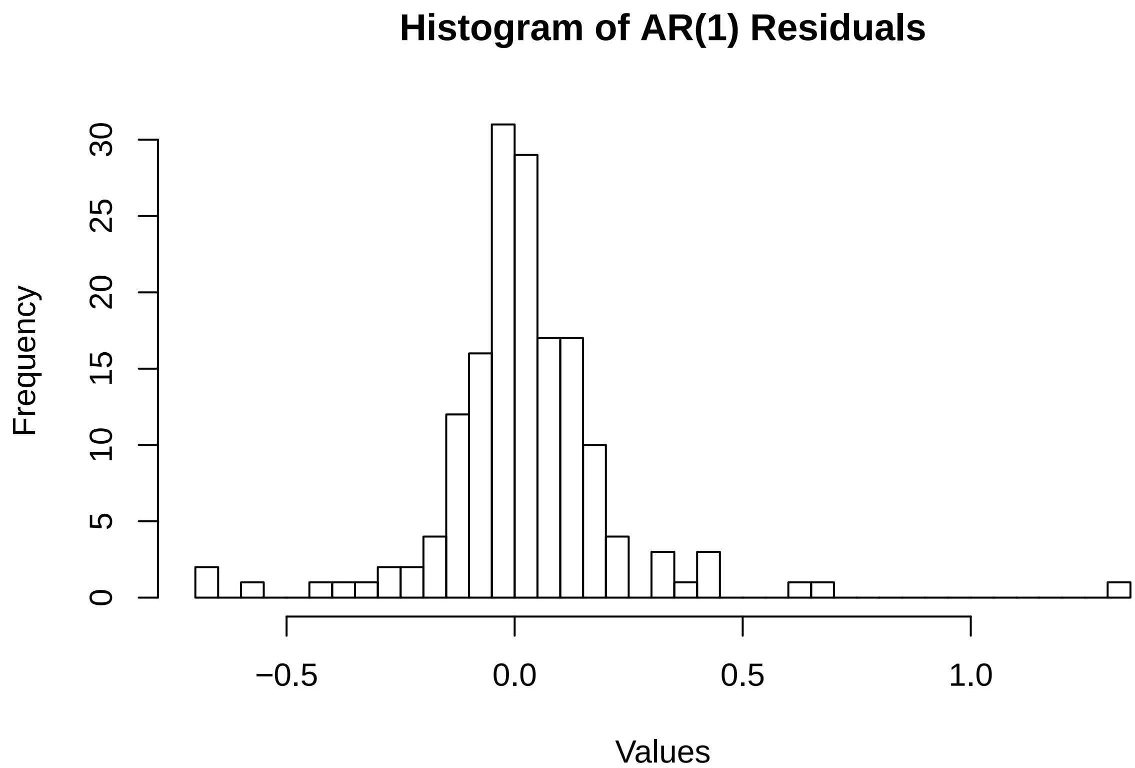
```
## [1] "Difference in AICs"
##          0          1          2          3          4          5
## 400.694524   0.000000   1.972401   1.888943   3.593698   1.706614
##          6          7          8          9         10         11
##   3.701569   5.636403   7.456583   9.096921  10.205772  11.334063
##         12         13         14         15         16         17
```

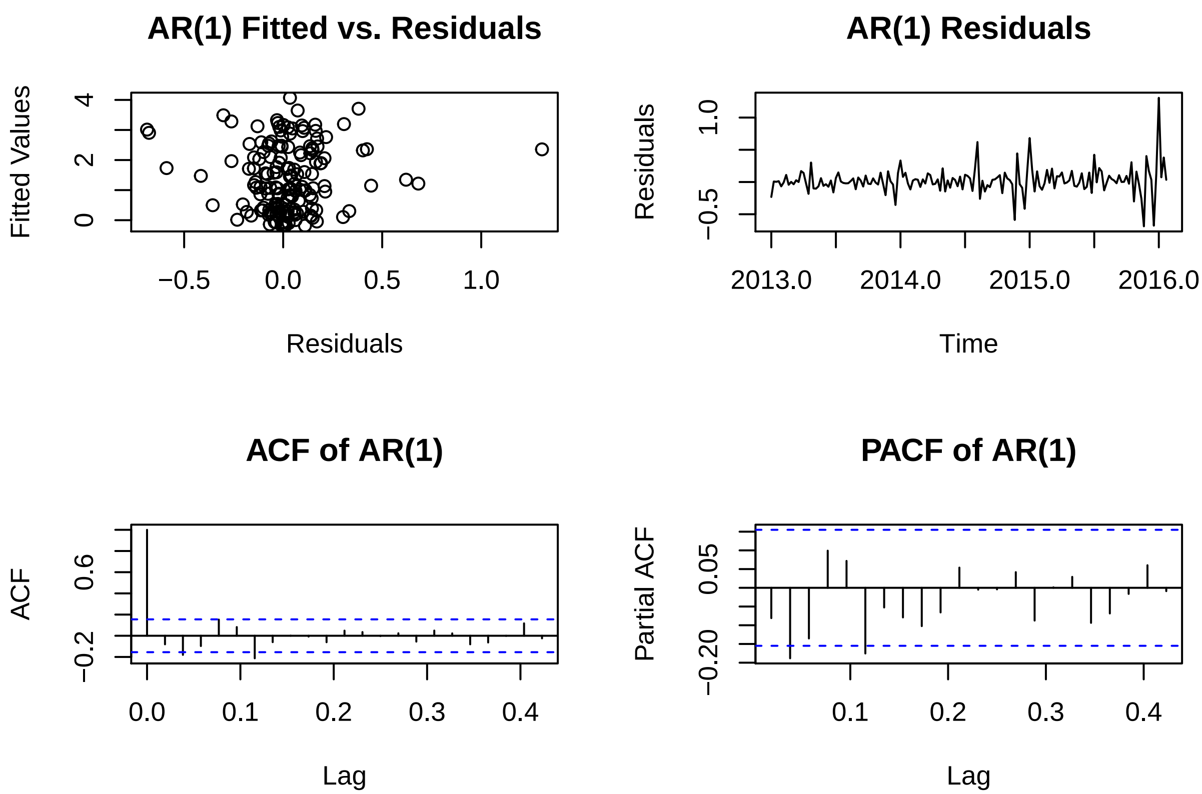
```
##  12.934238  14.792390  16.713379  16.686691  18.578162  20.389923
##         18         19         20         21         22
##  22.384765  24.318292  26.315895  28.295068  29.670844
## [1] "AR parameters"
## [1] 0.9587939
## [1] "AR order"
## [1] 1
```

```r
summary(glob.warm.ar)
```

```
##              Length Class  Mode
## order          1    -none- numeric
## ar             1    -none- numeric
## var.pred       1    -none- numeric
## x.mean         1    -none- numeric
## aic           23    -none- numeric
## n.used         1    -none- numeric
## order.max      1    -none- numeric
## partialacf    22    -none- numeric
## resid        160    ts     numeric
## method         1    -none- character
## series         1    -none- character
## frequency      1    -none- numeric
## call           2    -none- call
## asy.var.coef   1    -none- numeric
```

```r
# Create an AR(1) model
glob.warm.part.ar1 = arima(glob.warm.part.ts, order = c(1, 0,
    0))
# Plot residuals
plot.residuals.ts(glob.warm.part.ar1, "AR(1)")
```

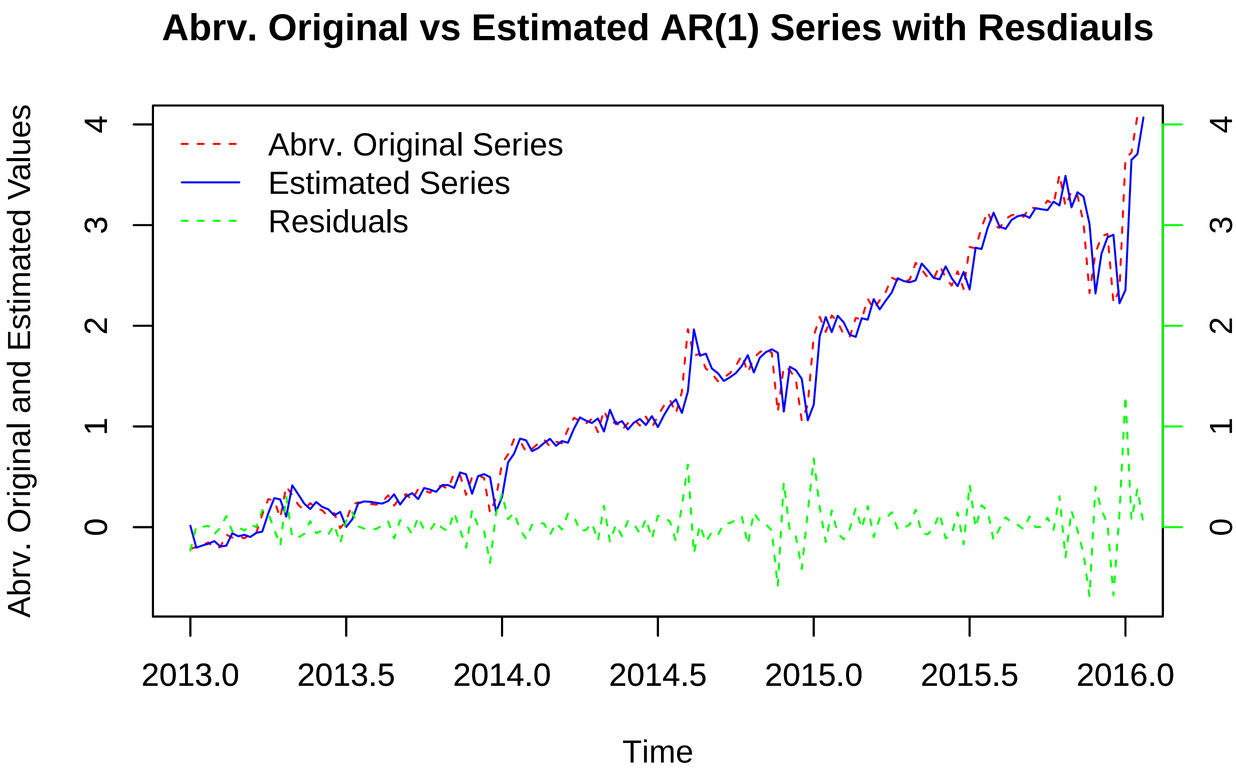

Histogram of AR(1) Residuals
Frequency
0
5
10
15
20
25
30
−0.5
0.0
0.5
1.0
Values




```
##
##  Box-Ljung test
##
## data:  x.mod$residuals
## X-squared = 1.0691, df = 1, p-value = 0.3012
```

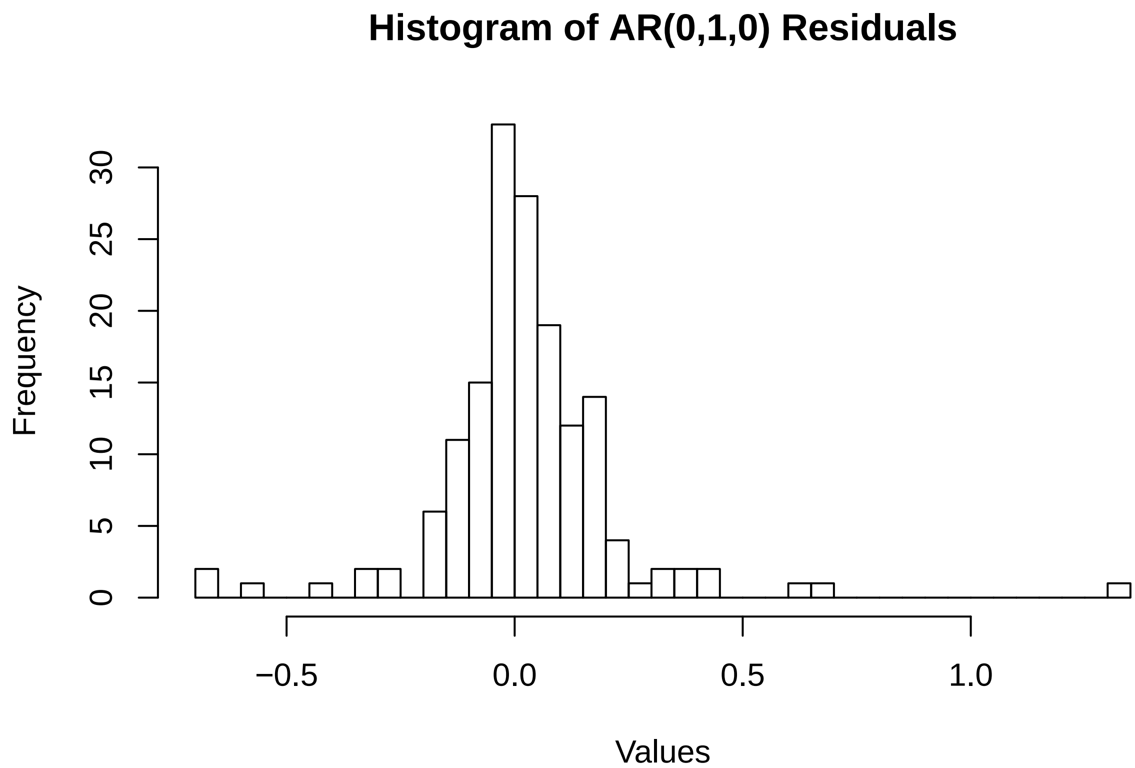
```
par(mfrow = c(2, 1))
# Plot the In-sample fit
plot.orig.model.resid(glob.warm.part.ts, glob.warm.part.ar1,
    part_name, "AR(1)", c(2013, 2016), c(-0.7, 4))
```

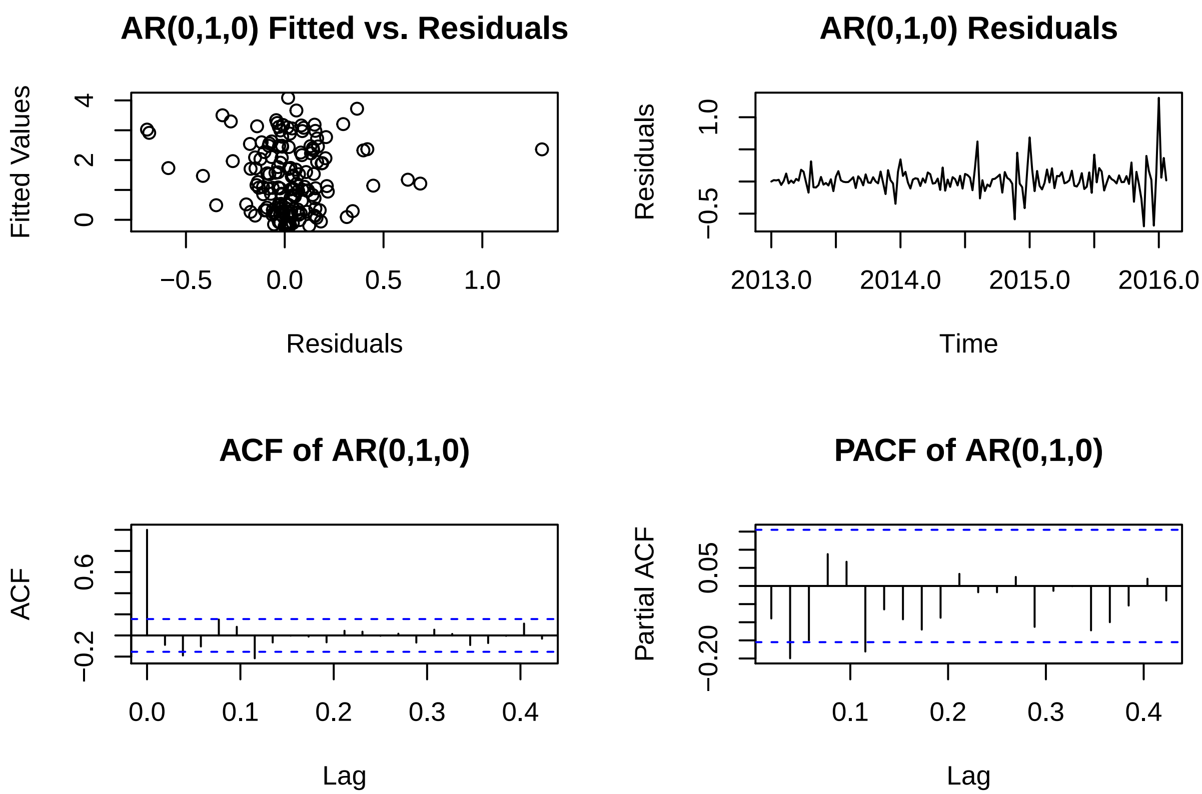
```
##
## Descriptive Stat
## ==========================================
## Statistic         N   Mean St. Dev. Min  Max
## ------------------------------------------
## x.ts              160 1.4     1.1     -0.2 4.1
## fitted.x.mod.     160 1.4     1.1     -0.2 4.1
## x.mod.residuals 160 0.02    0.2     -0.7 1.3
## ------------------------------------------
```




```
### Try ARIMA models gw.part.arima.best <-
### get.best.arima(glob.warm.part.ts, maxord=c(3,3,3)) Print
### the top 10 best models based on AIC
### gw.part.arima.best$others[order(gw.part.arima.best$others$aics)[1:10],]

# Create ARIMA(0,1,0) model
glob.warm.part.arima = arima(glob.warm.part.ts, order = c(0,
    1, 0))
# Plot residuals
plot.residuals.ts(glob.warm.part.arima, "AR(0,1,0)")
```

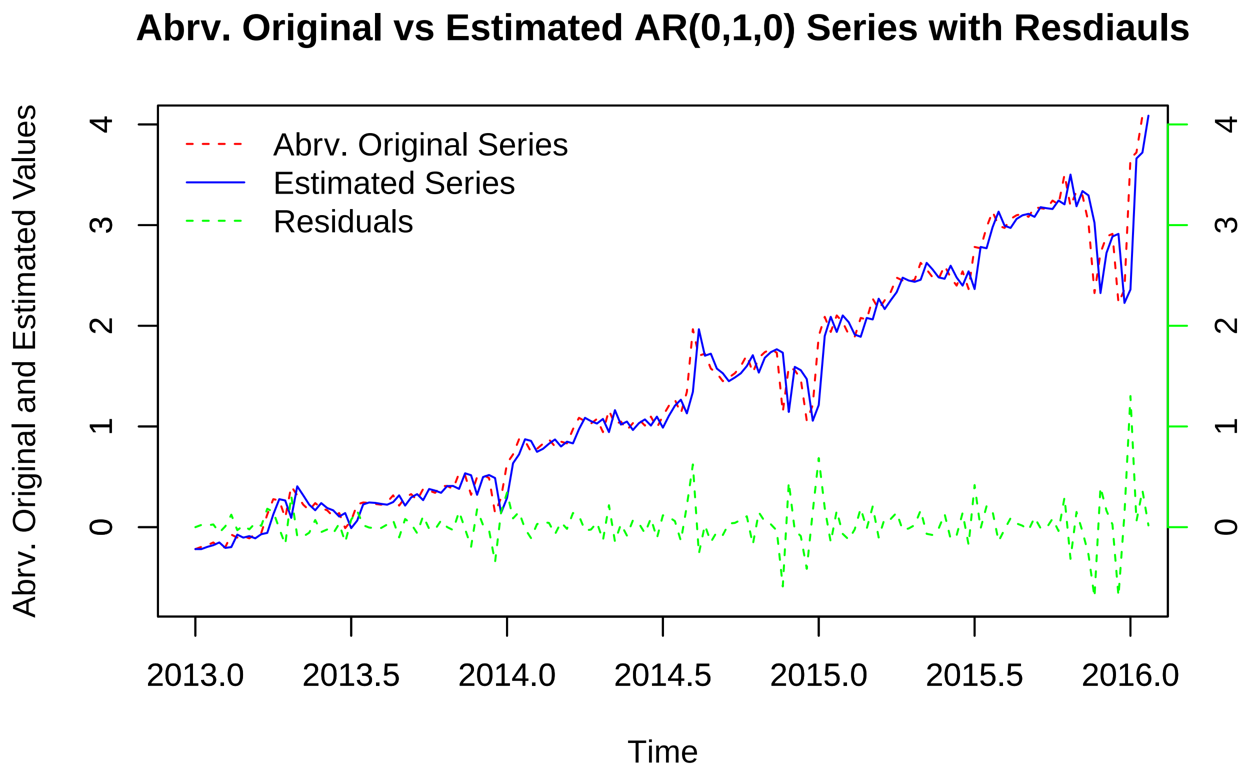

Histogram of AR(0,1,0) Residuals
Frequency
0
5
10
15
20
25
30
−0.5
0.0
0.5
1.0
Values




```
##
##  Box-Ljung test
##
## data:  x.mod$residuals
## X-squared = 1.3108, df = 1, p-value = 0.2523
```

```
# Plot In-sample fit
plot.orig.model.resid(glob.warm.part.ts, glob.warm.part.arima,
    part_name, "AR(0,1,0)", c(2013, 2016), c(-0.7, 4))
```

```
##
## Descriptive Stat
## ==========================================
## Statistic        N   Mean St. Dev. Min  Max
## ------------------------------------------
## x.ts             160 1.4   1.1     -0.2 4.1
## fitted.x.mod.    160 1.4   1.1     -0.2 4.1
## x.mod.residuals  160 0.03  0.2     -0.7 1.3
## ------------------------------------------
```



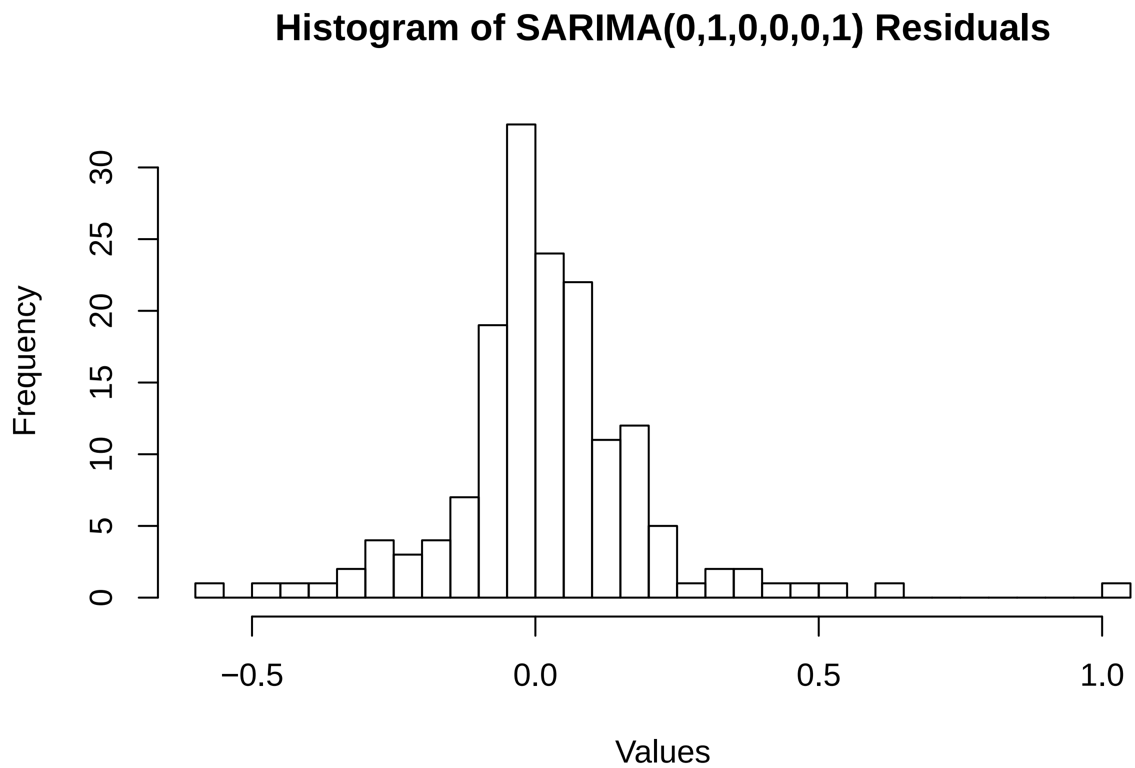

```
### Try SARIMA models
gw.part.seas.best <- get.best.sarima(glob.warm.part.ts, maxord = c(1,
    1, 1, 1, 1, 1), 52)
```

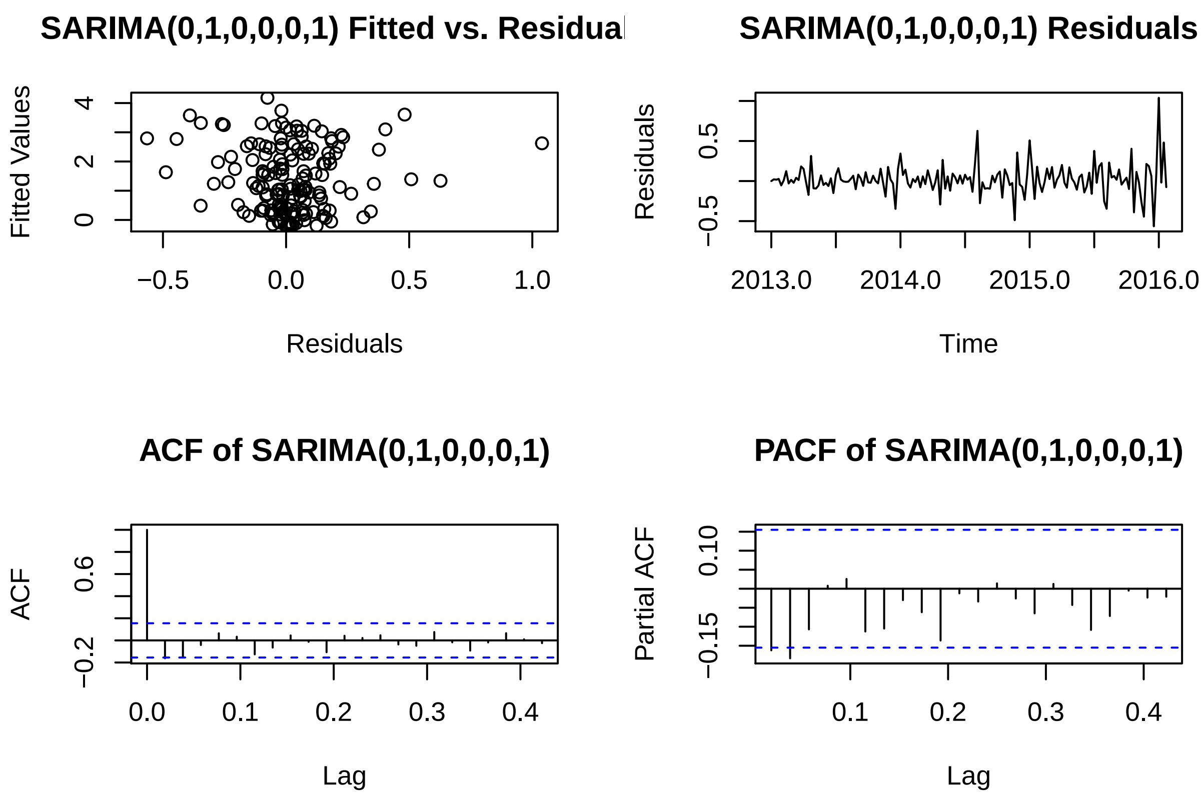
```
## Warning in arima(x.ts, order = c(p, d, q), seasonal = list(order = c(P, :
## possible convergence problem: optim gave code = 1
```

```
# Print the top 20 best models based on AIC
gw.part.seas.best$others[order(gw.part.seas.best$others$aics)[1:5],
    ]
```

```
##         aics             models
## 30 -90.17105 (0, 1, 1, 1, 0, 1)
## 54 -87.66116 (1, 1, 0, 1, 0, 1)
## 62 -84.61532 (1, 1, 1, 1, 0, 1)
## 46 -78.92479 (1, 0, 1, 1, 0, 1)
## 18 -76.45286 (0, 1, 0, 0, 0, 1)
```

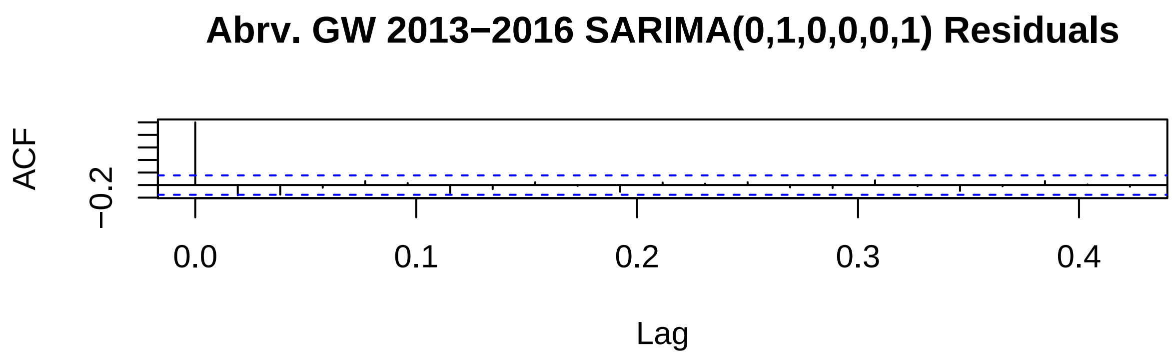
```
# Create SARIMA(0,1,0,0,0,1) model
glob.warm.part.sarima = arima(glob.warm.part.ts, order = c(0,
    1, 0), seas = list(order = c(0, 0, 1), 52), method = "CSS")
# Plot the residuals
plot.residuals.ts(glob.warm.part.sarima, "SARIMA(0,1,0,0,0,1)")
```

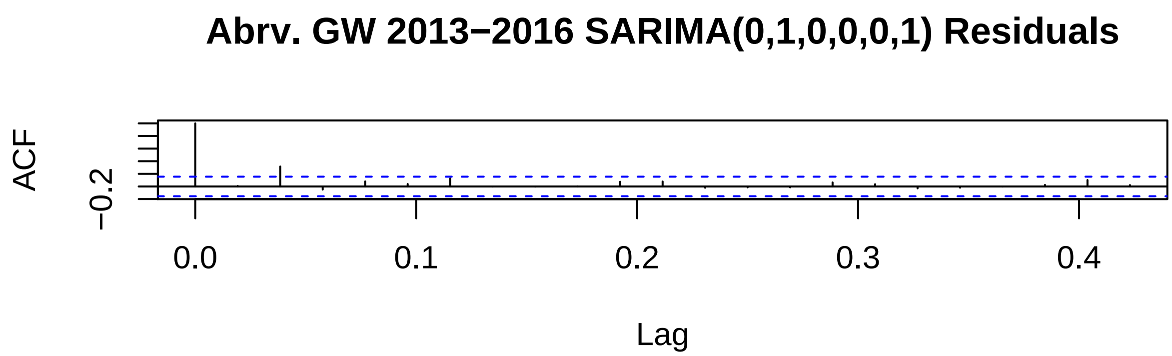

Histogram of SARIMA(0,1,0,0,0,1) Residuals
Frequency
0
5
10
15
20
25
30
−0.5
0.0
0.5
1.0
Values




```
##
##  Box-Ljung test
##
## data:  x.mod$residuals
## X-squared = 4.2839, df = 1, p-value = 0.03847
```

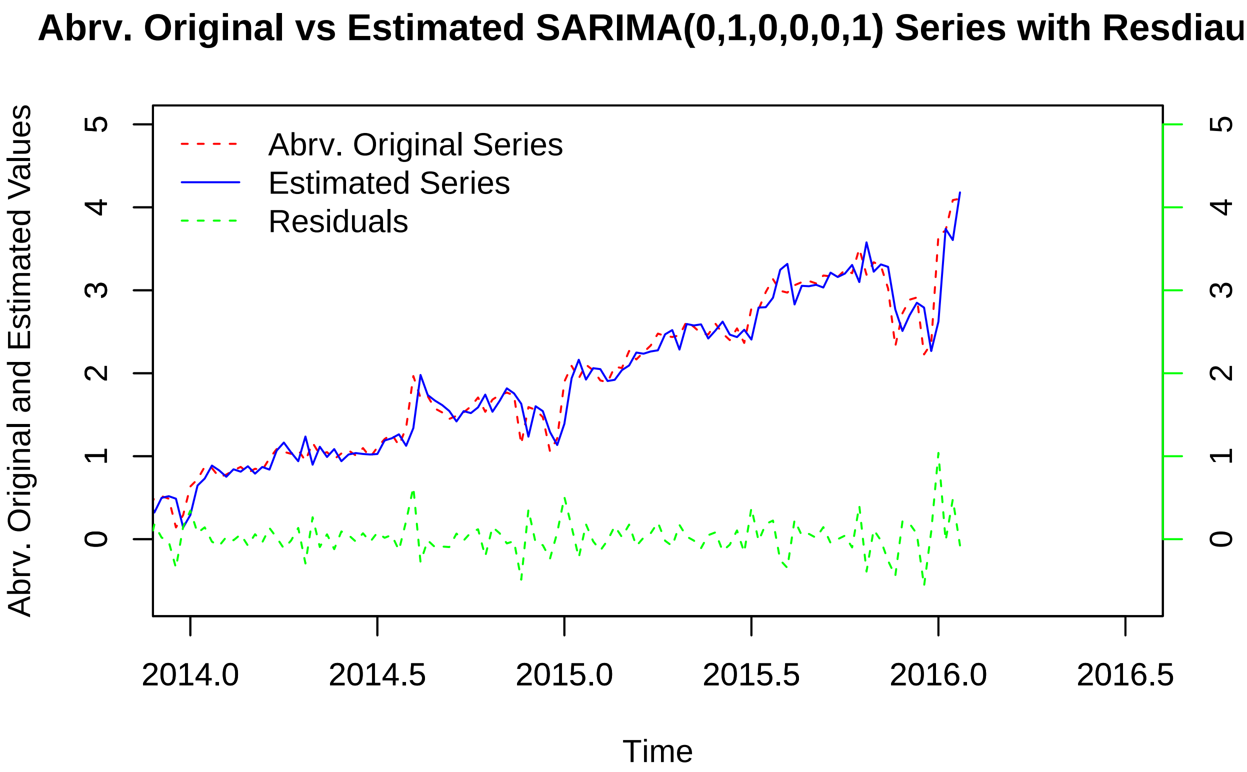
```
par(mfrow = c(2, 1))
acf(glob.warm.part.sarima$residuals, main = "Abrv. GW 2013-2016 SARIMA(0,1,0,0,0,1) Residuals")
acf(glob.warm.part.sarima$residuals^2, main = "Abrv. GW 2013-2016 SARIMA(0,1,0,0,0,1) Residuals")
```






```
# Plot the In-sample fit
plot.orig.model.resid(glob.warm.part.ts, glob.warm.part.sarima,
    part_name, "SARIMA(0,1,0,0,0,1)", c(2014, 2016.5), c(-0.7,
        5))
```

```
##
## Descriptive Stat
## ==========================================
## Statistic        N   Mean St. Dev. Min  Max
## ------------------------------------------
## x.ts            160 1.4     1.1    -0.2 4.1
## fitted.x.mod.   160 1.4     1.1    -0.2 4.2
## x.mod.residuals 160 0.02    0.2    -0.6 1.0
## ------------------------------------------
```

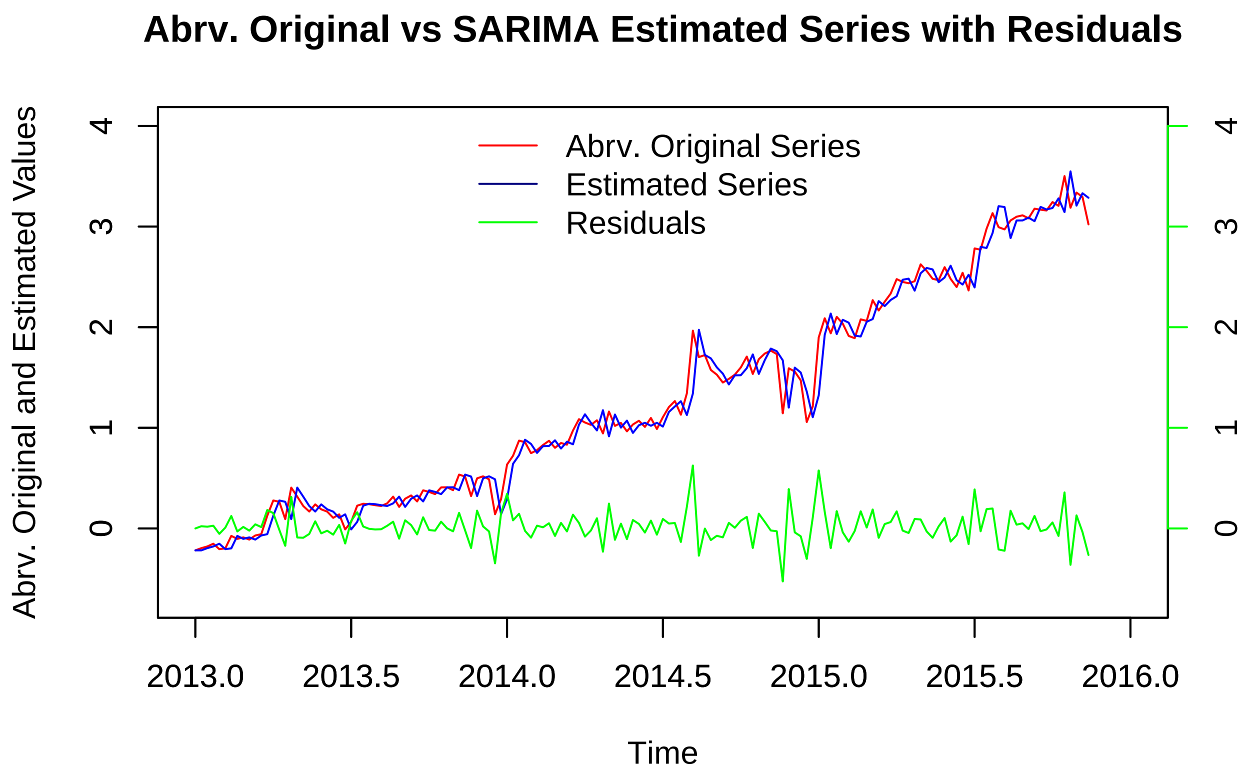



```r
#### Backtesting Create a time series cutting off the last 10%
#### of the values
glob.warm.part.bt.ts = ts(glob.warm.part.ts[1:(length(glob.warm.part.ts) -
    10)], start = 2013, frequency = 52)
# Recreate SARIMA(0,1,0,0,0,1) model
glob.warm.part.sarima.bt = arima(glob.warm.part.bt.ts, order = c(0,
    1, 0), seas = list(order = c(0, 0, 1), 52), method = "CSS")
# Combine the cut-off time series with the fitted values from
# the model and the residuals from the model
df.part = cbind(glob.warm.part.bt.ts, fitted(glob.warm.part.sarima.bt),
    glob.warm.part.sarima.bt$resid)
colnames(df.part) = c("orig_series", "fitted_vals", "resid")
head(df.part)
```

```
##      orig_series fitted_vals  resid
## [1,]      -0.218      -0.218  0.000
## [2,]      -0.196      -0.218  0.022
## [3,]      -0.179      -0.196  0.017
## [4,]      -0.152      -0.179  0.027
## [5,]      -0.206      -0.152 -0.054
## [6,]      -0.198      -0.206  0.008
```

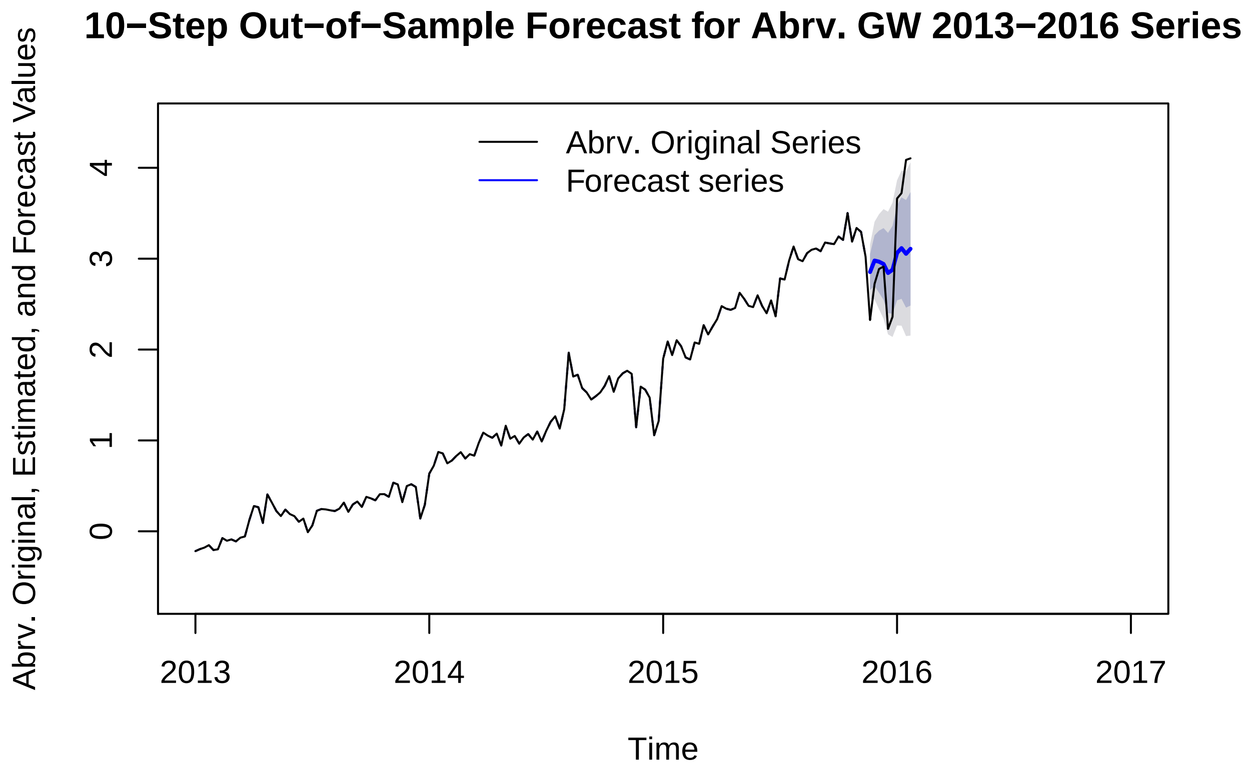
```r
# Plot the Abrv. Original and estimate series with residuals
par(mfrow = c(1, 1))
plot.ts(df.part[, "orig_series"], col = "red", main = "Abrv. Original vs SARIMA Estimated Series with Re
```

```
    ylab = "Abrv. Original and Estimated Values", xlim = c(2013,
        2016), ylim = c(-0.7, 4))
par(new = T)
plot.ts(df.part[, "fitted_vals"], col = "blue", axes = T, xlab = "",
    ylab = "", xlim = c(2013, 2016), ylim = c(-0.7, 4))
leg.txt <- c("Abrv. Original Series", "Estimated Series", "Residuals")
legend("top", legend = leg.txt, lty = 1, col = c("red", "navy",
    "green"), bty = "n", cex = 1)
par(new = T)
plot.ts(df.part[, "resid"], axes = F, xlab = "", ylab = "", col = "green",
    xlim = c(2013, 2016), ylim = c(-0.7, 4), pch = 1)
axis(side = 4, col = "green")
mtext("Residuals", side = 4, line = 2, col = "green")
```




```
# Create a forecast for backtesting and plot it
glob.warm.part.sarima.bt.fcast = forecast.Arima(glob.warm.part.sarima.bt,
    h = 10)
par(mfrow = c(1, 1))
plot(glob.warm.part.sarima.bt.fcast, lty = 2, col = "navy", main = "10-Step Out-of-Sample Forecast for
    ylab = "Abrv. Original, Estimated, and Forecast Values",
    xlim = c(2013, 2017), ylim = c(-0.7, 4.5))
par(new = T)
plot.ts(glob.warm.part.ts, axes = F, lty = 1, col = "black",
    xlim = c(2013, 2017), ylim = c(-0.7, 4.5), ylab = "")
leg.txt <- c("Abrv. Original Series", "Forecast series")
```

```r
legend("top", legend = leg.txt, lty = 1, col = c("black", "blue"),
    bty = "n", cex = 1)
```




```r
#### Forecasting - Forecast the request 12-step ahead forecast
glob.warm.part.sarima.fcast = forecast.Arima(glob.warm.part.sarima,
    h = 12)
print(str(glob.warm.part.sarima.fcast))
```

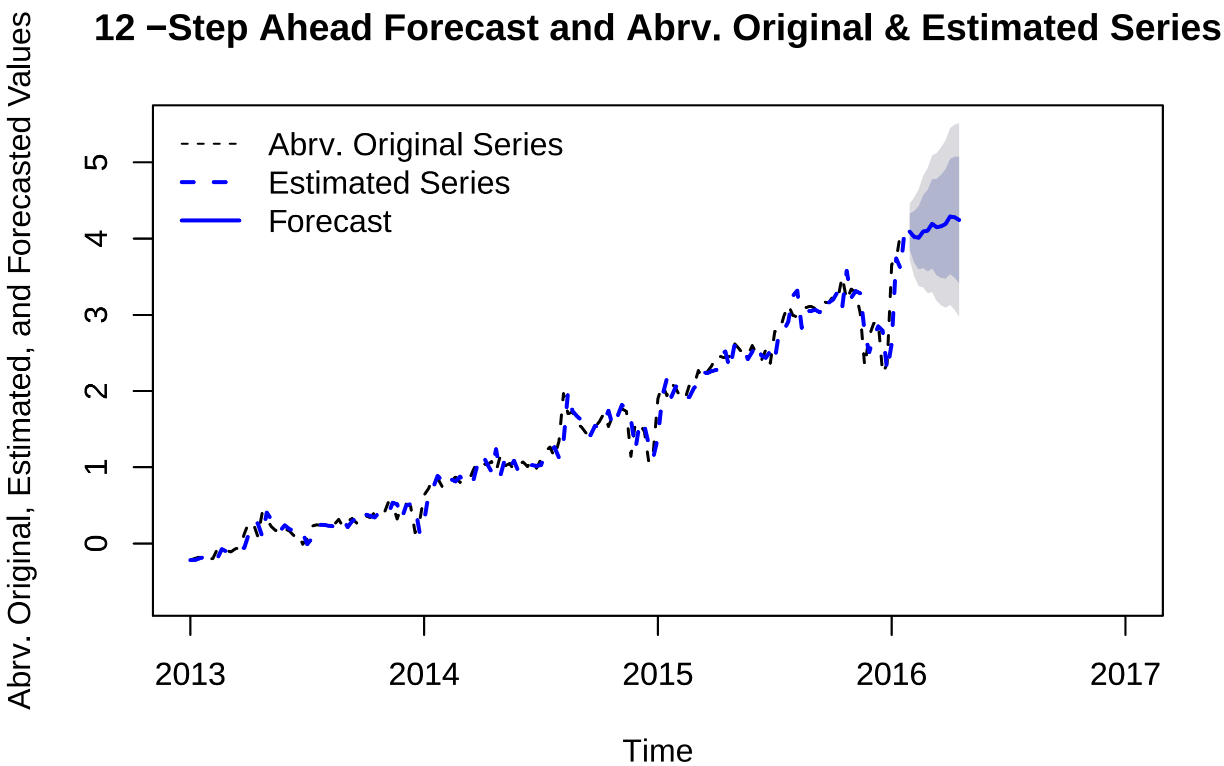
```
## List of 10
##  $ method   : chr "ARIMA(0,1,0)(0,0,1)[52]"
##  $ model    :List of 15
##   ..$ coef     : Named num 0.517
##   .. ..- attr(*, "names")= chr "sma1"
##   ..$ sigma2   : num 0.0348
##   ..$ var.coef : num [1, 1] 0.00796
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr "sma1"
##   .. .. ..$ : chr "sma1"
##   ..$ mask     : logi TRUE
##   ..$ loglik   : num 41.3
##   ..$ aic      : logi NA
##   ..$ arma     : int [1:7] 0 0 0 1 52 1 0
##   ..$ residuals: Time-Series [1:160] from 2013 to 2016: 0 0.022 0.017 0.027 -0.054 ...
##   ..$ call     : language arima(x = glob.warm.part.ts, order = c(0, 1, 0), seasonal = list(order = c
##   ..$ series   : chr "glob.warm.part.ts"
```

```
##   ..$ code     : int 0
##   ..$ n.cond   : num 1
##   ..$ nobs     : int 159
##   ..$ model    :List of 10
##   .. ..$ phi  : num(0)
##   .. ..$ theta: num [1:52] 0 0 0 0 0 0 0 0 0 0 ...
##   .. ..$ Delta: num 1
##   .. ..$ Z    : num [1:54] 1 0 0 0 0 0 0 0 0 0 ...
##   .. ..$ a    : num [1:54] 0.017 -0.0125 -0.0693 -0.0113 0.0818 ...
##   .. ..$ P    : num [1:54, 1:54] 0 0 0 0 0 0 0 0 0 0 ...
##   .. ..$ T    : num [1:54, 1:54] 0 0 0 0 0 0 0 0 0 0 ...
##   .. ..$ V    : num [1:54, 1:54] 1 0 0 0 0 0 0 0 0 0 ...
##   .. ..$ h    : num 0
##   .. ..$ Pn   : num [1:54, 1:54] 1.00 -1.23e-21 1.41e-42 7.95e-63 -1.10e-78 ...
##   ..$ x        : Time-Series [1:160] from 2013 to 2016: -0.218 -0.196 -0.179 -0.152 -0.206 -0.198 -0
##   ..- attr(*, "class")= chr "Arima"
##  $ level    : num [1:2] 80 95
##  $ mean     : Time-Series [1:12] from 2016 to 2016: 4.09 4.02 4.01 4.09 4.1 ...
##  $ lower    : num [1:12, 1:2] 3.85 3.68 3.6 3.61 3.57 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "80%" "95%"
##  $ upper    : num [1:12, 1:2] 4.33 4.36 4.43 4.57 4.64 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:2] "80%" "95%"
##  $ x        : Time-Series [1:160] from 2013 to 2016: -0.218 -0.196 -0.179 -0.152 -0.206 -0.198 -0.07
##  $ xname    : chr [1:19] "structure(c(-0.218, -0.196, -0.179, -0.152, -0.206, -0.198, -0.074, " "-0.
##  $ fitted   : Time-Series [1:160] from 2013 to 2016: -0.218 -0.218 -0.196 -0.179 -0.152 -0.206 -0.19
##  $ residuals: Time-Series [1:160] from 2013 to 2016: 0 0.022 0.017 0.027 -0.054 ...
##  - attr(*, "class")= chr "forecast"
## NULL
```

```r
print(summary(glob.warm.part.sarima.fcast$mean))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.011   4.092   4.156   4.153   4.207   4.290
```

```r
plot.model.forecast(glob.warm.part.sarima, glob.warm.part.sarima.fcast,
    part_name, "12", c(2013, 2017), c(-0.7, 5.5))
```


12 −Step Ahead Forecast and Abrv. Original & Estimated Series
Abrv. Original Series
Estimated Series
Forecast
Abrv. Original, Estimated, and Forecasted Values
0
1
2
3
4
5
2013
2014
2015
2016
2017
Time