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, 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("Orivinal vs Estimated",
        model_name, "Series with Resdiauls", sep = " "), ylab = "Original and Estimated Values",
        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("Original Series", "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")
}
```

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

Data Analysis

1. The time series has weekly 630 values 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.
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.

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 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% CI.
- PACF: No. The PACF shows correlation with several values outside of the 9%% 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.

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. Inspecty 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 c(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.

3. **Try SARIMA models.** 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 c(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 c(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 c(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.

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

```
##  Time-Series [1:630] from 1 to 630: -0.44 -0.474 -0.423 -0.551 -0.486 -0.551 -0.453 -0.462 -0.551 -0
```

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

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

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

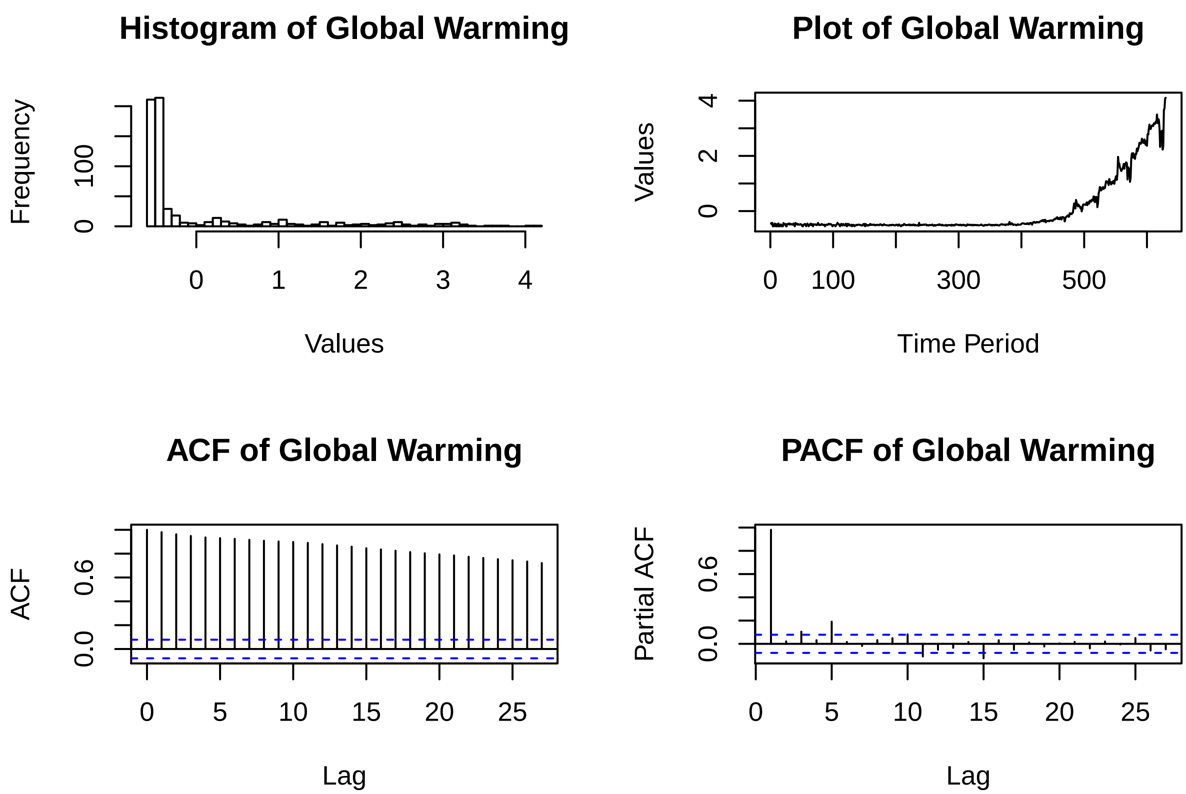
```
##        [,1]  [,2]
## [1,] -0.440 2.227
## [2,] -0.474 2.360
## [3,] -0.423 3.662
## [4,] -0.551 3.721
## [5,] -0.486 4.087
## [6,] -0.551 4.104
```

```
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")
```

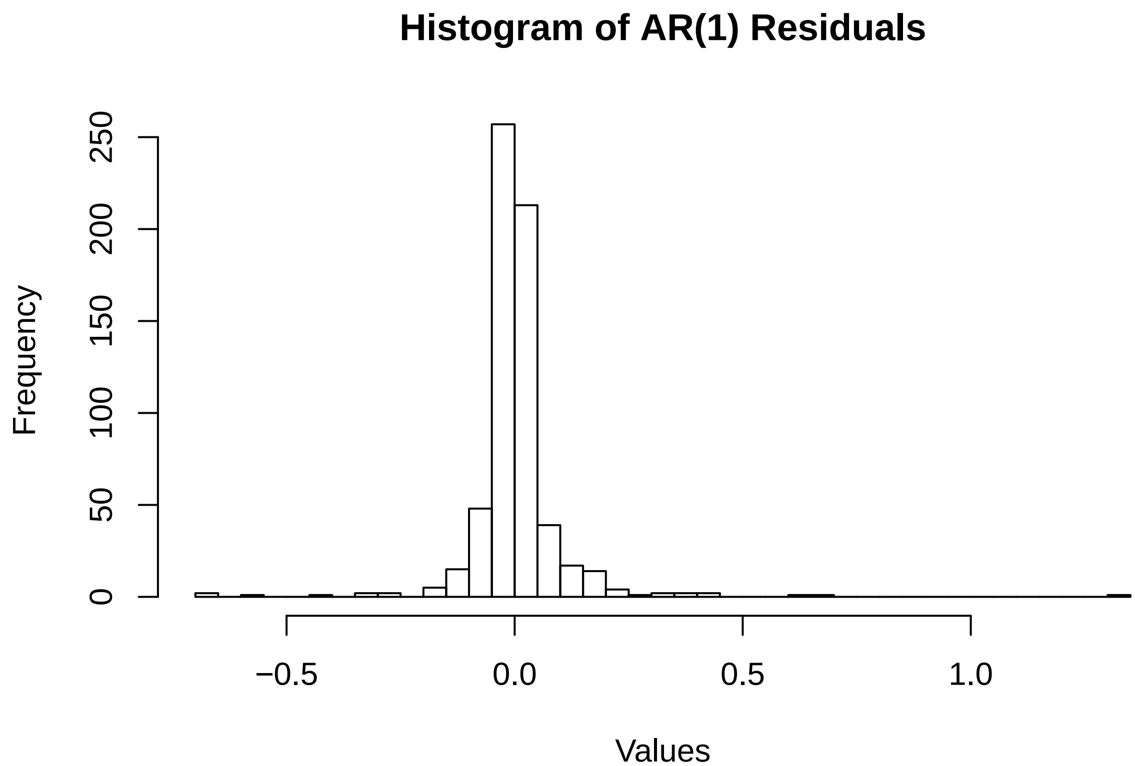

```
##  Time-Series [1:630] from 1 to 630: -0.44 -0.474 -0.423 -0.551 -0.486 -0.551 -0.453 -0.462 -0.551 -0
```

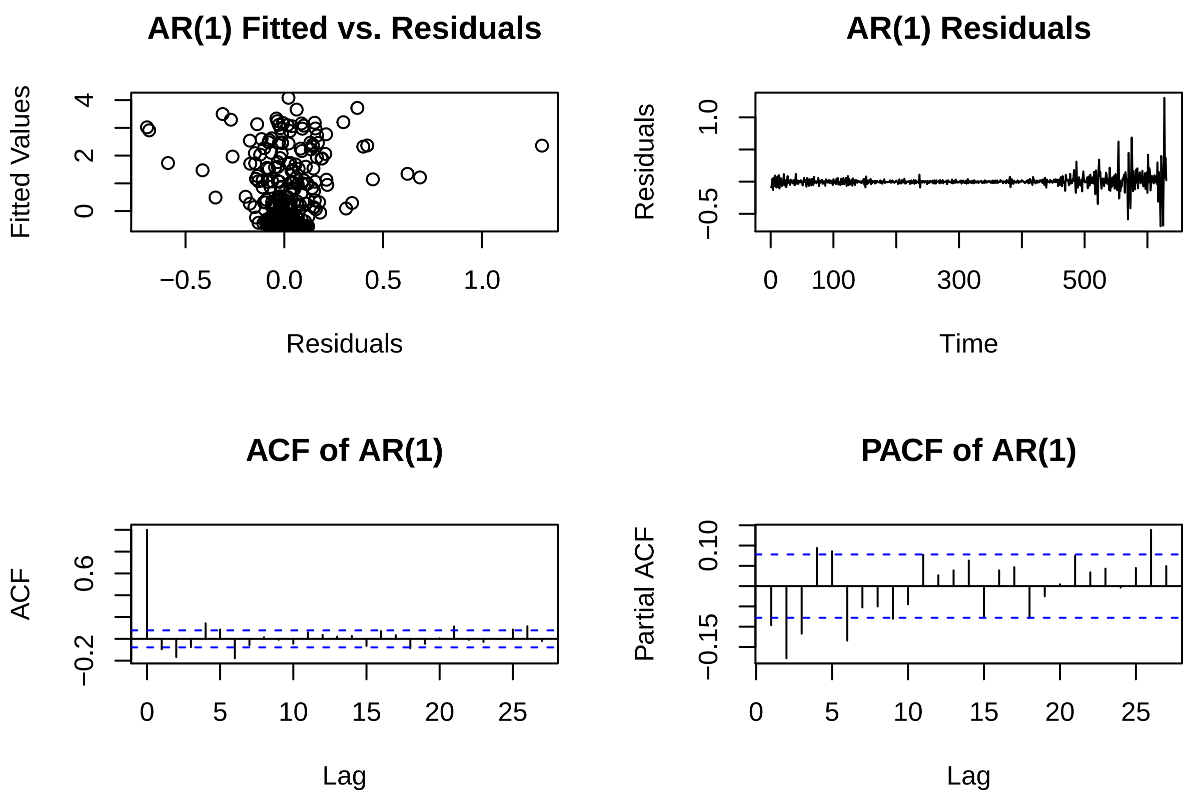



```
### Try AR models
glob.warm.ar = estimate.ar(glob.warm.ts)
```

```
## [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)")
```

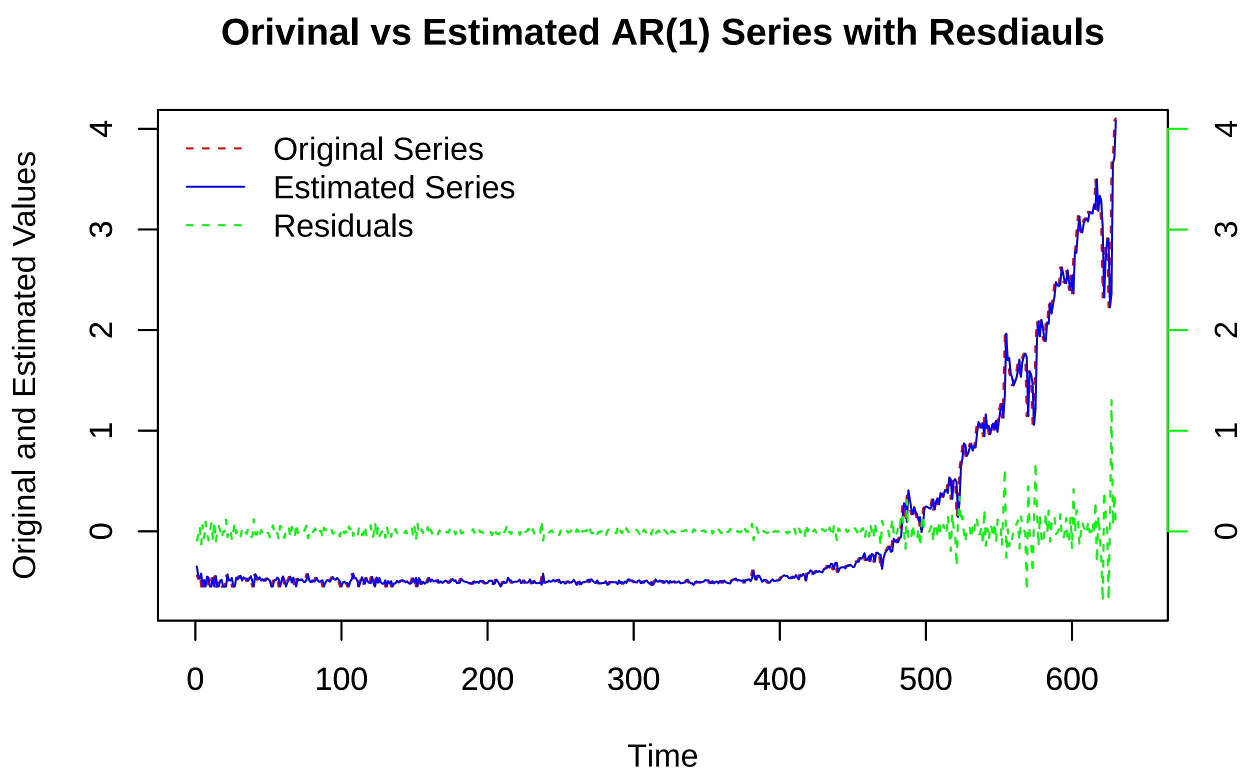

Histogram of AR(1) Residuals
Frequency
0
50
100
150
200
250
−0.5
0.0
0.5
1.0
Values



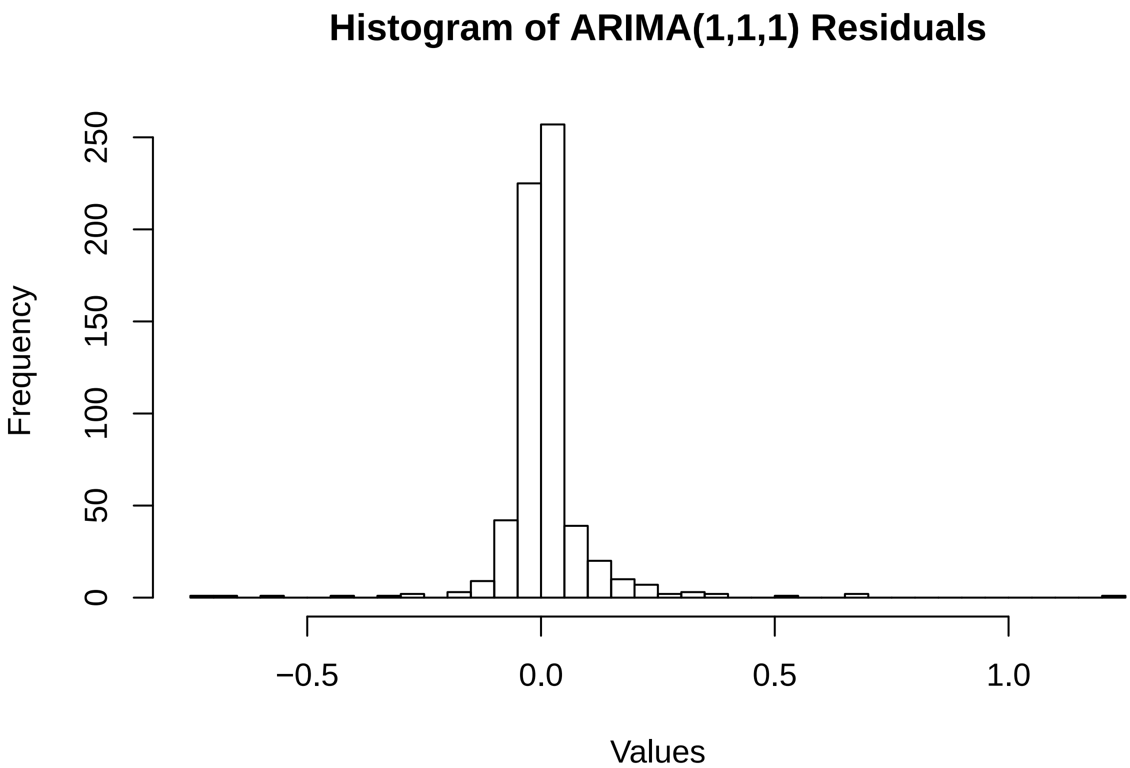

```
##
##  Box-Ljung test
##
## data:  x.mod$residuals
## X-squared = 5.8789, df = 1, p-value = 0.01532
```

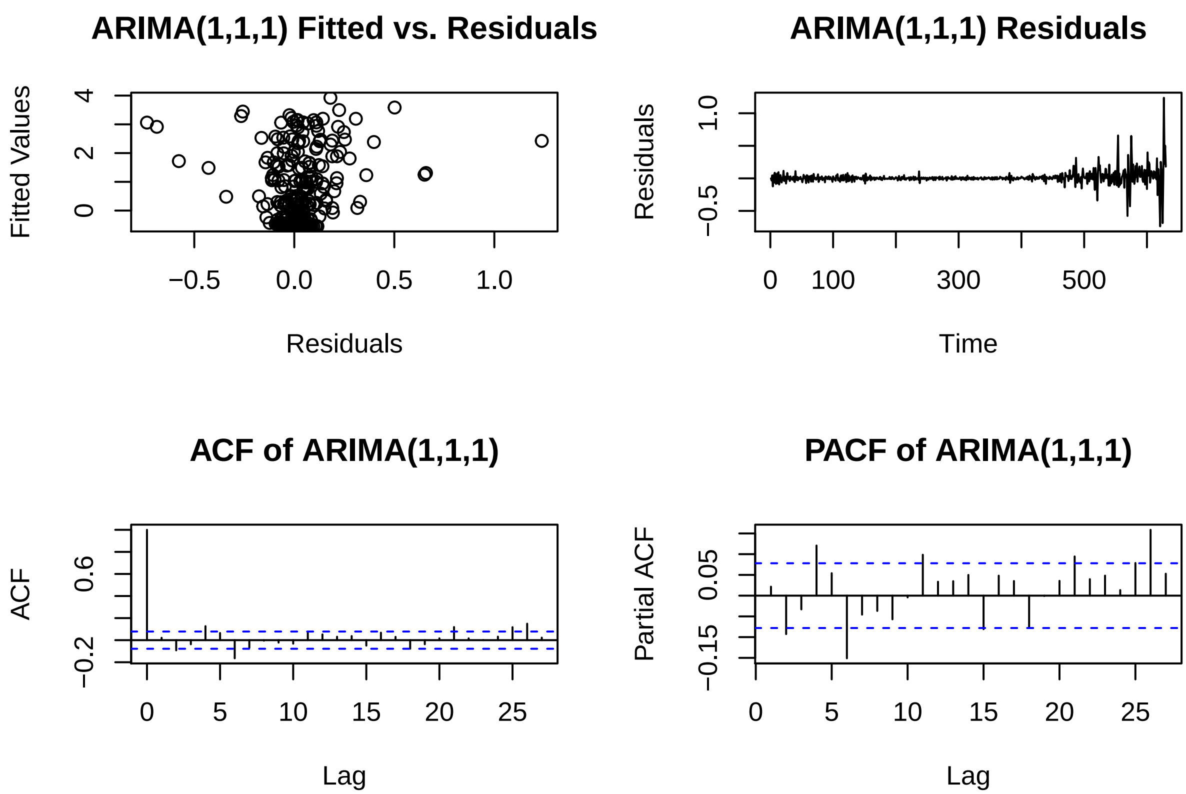
```
# Plot the In-sample fit
plot.orig.model.resid(glob.warm.ts, glob.warm.ar1, "AR(1)", c(0,
    640), c(-0.7, 4))
```

```
##
## 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
## -------------------------------------------
```




```
### 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)")
```

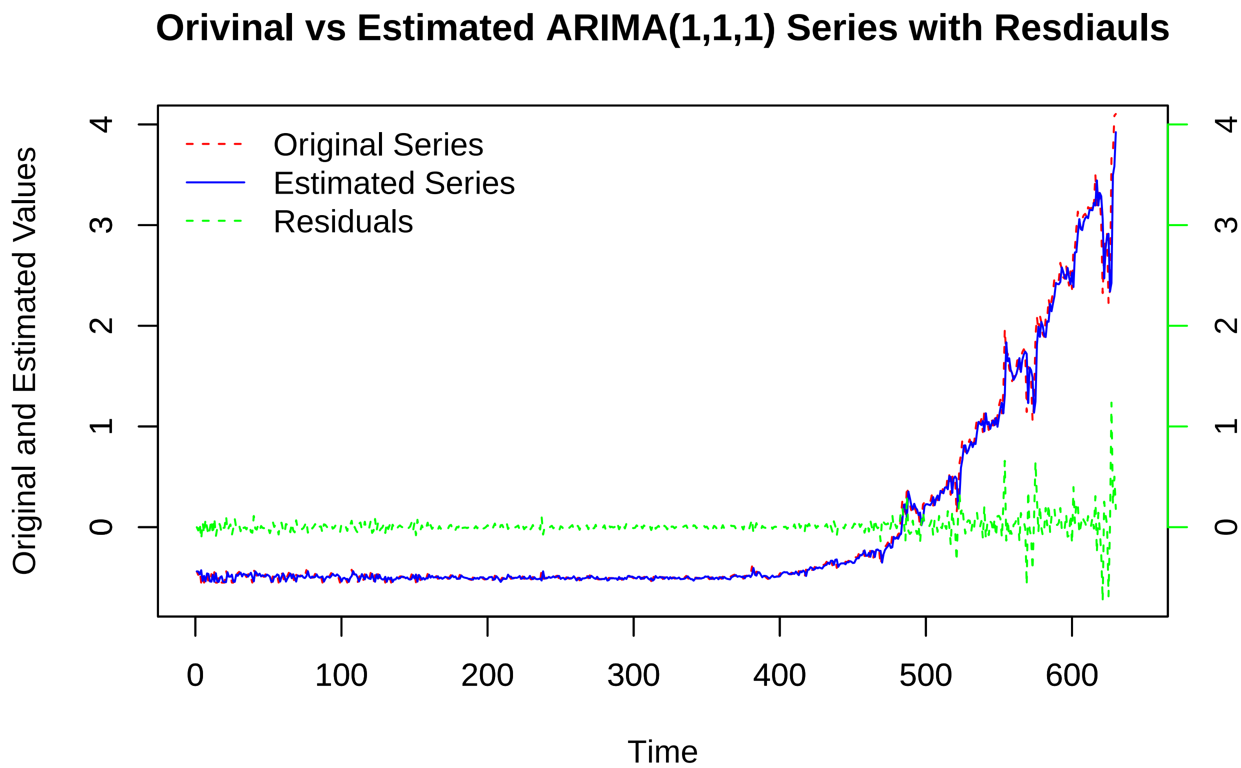

Histogram of ARIMA(1,1,1) Residuals
Frequency
0
50
100
150
200
250
−0.5
0.0
0.5
1.0
Values




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

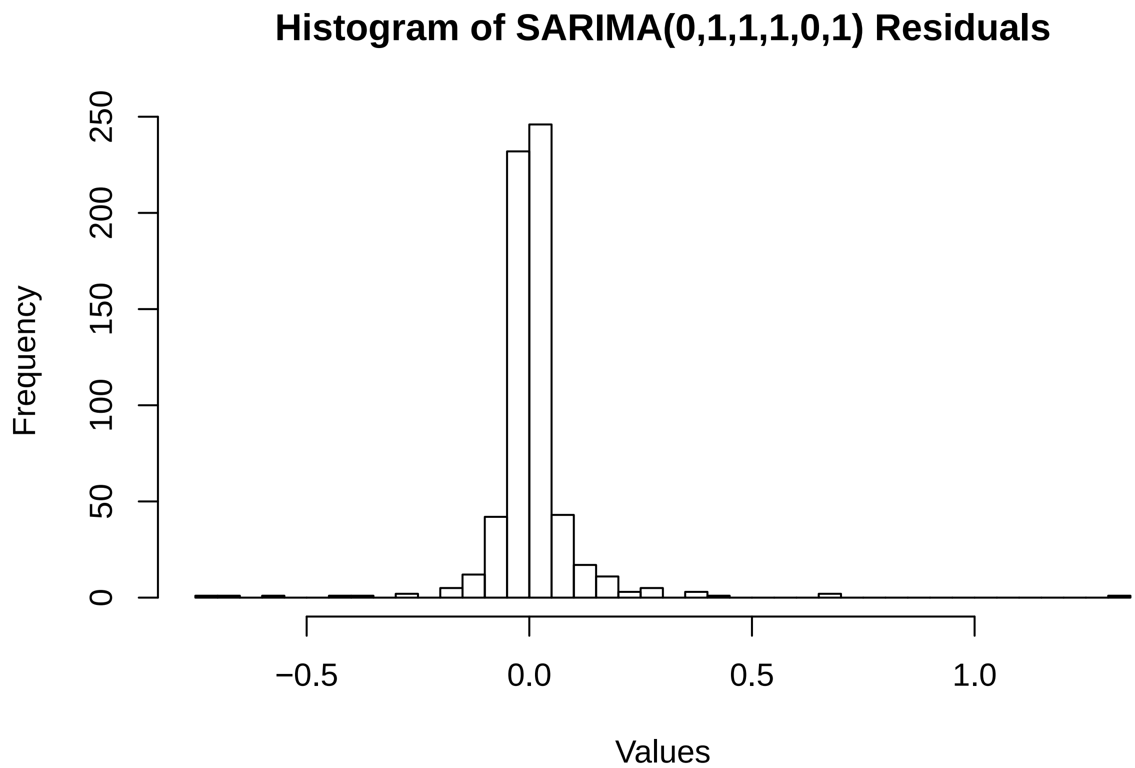
```
# Plot the In-sample fit
plot.orig.model.resid(glob.warm.ts, glob.warm.arima, "ARIMA(1,1,1)",
    c(0, 640), c(-0.7, 4))
```

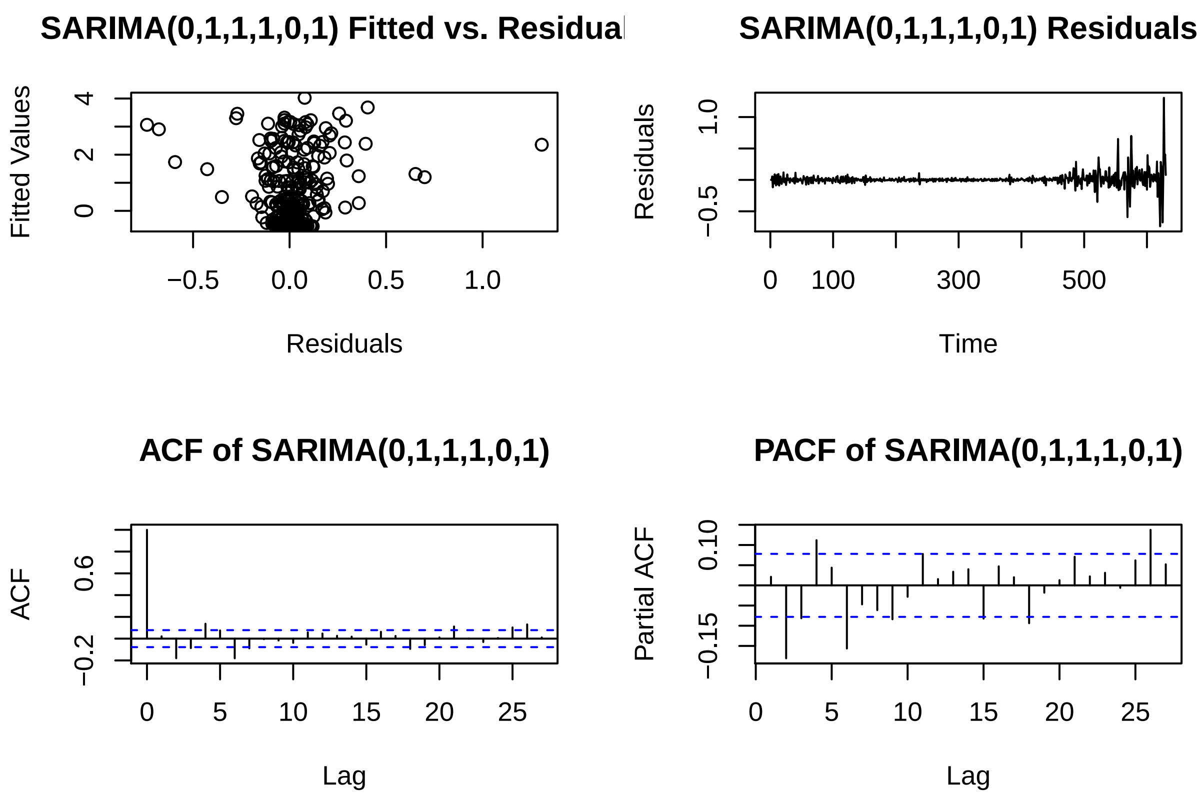
```
##
## Descriptive Stat
## ===========================================
## Statistic         N   Mean   St. Dev. Min  Max
## -------------------------------------------
## x.ts              630 0.000   1.0     -0.6 4.1
## fitted.x.mod.     630 -0.01   1.0     -0.5 3.9
## x.mod.residuals 630 0.01  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




```
### 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
plot.residuals.ts(glob.warm.arima.seas, "SARIMA(0,1,1,1,0,1)")
```

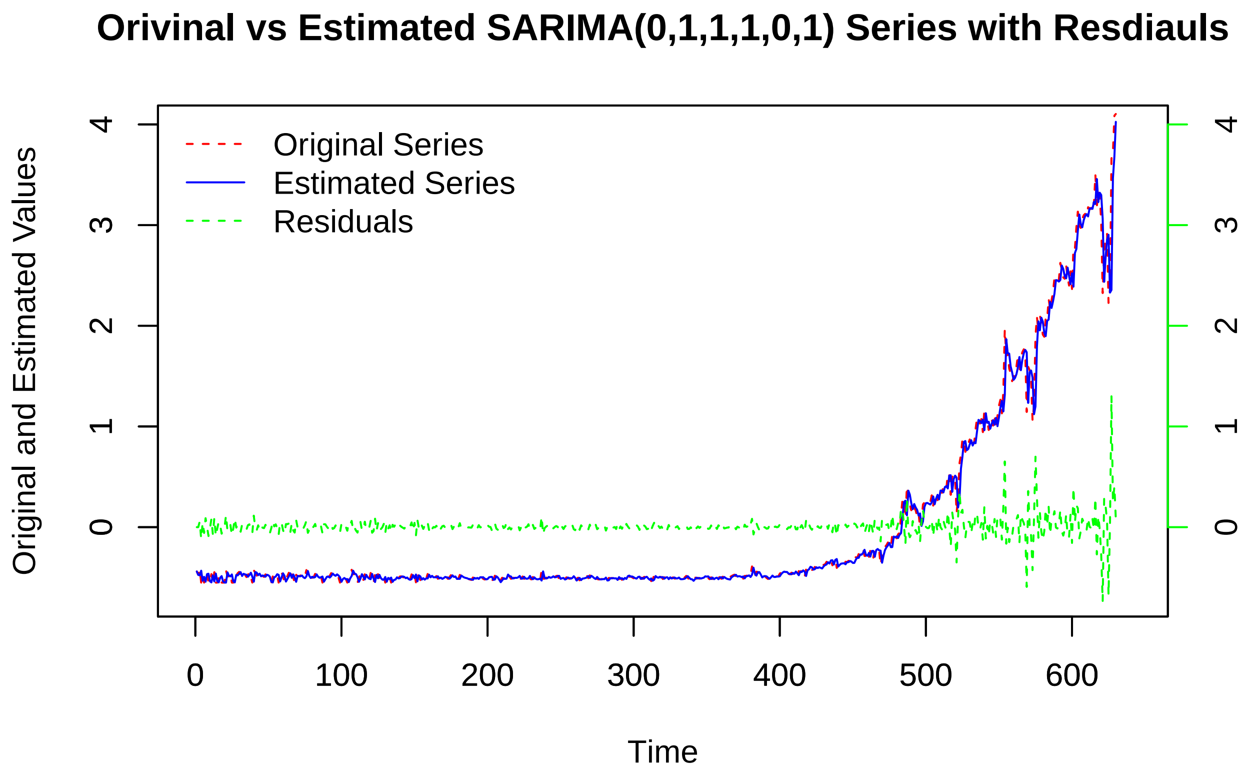

Histogram of SARIMA(0,1,1,1,0,1) Residuals
Frequency
0
50
100
150
200
250
−0.5
0.0
0.5
1.0
Values




```
##
##  Box-Ljung test
##
## data:  x.mod$residuals
## X-squared = 0.28653, df = 1, p-value = 0.5925
```

```
# Plot the In-sample fit
plot.orig.model.resid(glob.warm.ts, glob.warm.arima.seas, "SARIMA(0,1,1,1,0,1)",
    c(0, 640), 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     -0.7 1.3
## --------------------------------------------
```




```
# ts.plot(cbind(glob.warm.ts,
# predict(glob.warm.arima.seas,12)$pred),lty=1:2)
```