Exam2

Alex Rintamaa

2024-04-25

```r
Eventdf = read_csv("EventCostBDD.csv", skip = 1)
```

```
## Rows: 376 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): Name, Disaster, Disaster Group, Begin Date, End Date, Central Day,...
## dbl (17): Yb, Mb, Db, Ye, Me, De, Total CPI-Adjusted Cost (Millions of Dolla...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(Eventdf)
```

```
## # A tibble: 6 x 24
##   Name       Disaster `Disaster Group` `Begin Date`    Yb    Mb    Db `End Date`
##   <chr>      <chr>    <chr>            <chr>        <dbl> <dbl> <dbl> <chr>
## 1 Southern ~ Flooding SevStorm/Flood   4/10/1980     1980     4    10 4/17/1980
## 2 Hurricane~ Tropica~ Tropical Cyclone 8/7/1980      1980     8     7 8/11/1980
## 3 Central/E~ Drought  WF/Drought       6/1/1980      1980     6     1 11/30/1980
## 4 Florida F~ Freeze   Winter/Freeze    1/12/1981     1981     1    12 1/14/1981
## 5 Severe St~ Severe ~ SevStorm/Flood   5/5/1981      1981     5     5 5/10/1981
## 6 Midwest/S~ Winter ~ Winter/Freeze    1/8/1982      1982     1     8 1/16/1982
## # i 16 more variables: Ye <dbl>, Me <dbl>, De <dbl>,
## #   `Total CPI-Adjusted Cost (Millions of Dollars)` <dbl>, Deaths <dbl>,
## #   `Durn (days)` <dbl>, `Durn (Weeks)` <dbl>, `Durn (Cal Mos)` <dbl>,
## #   `Central Day` <chr>, `Day of W_yr` <dbl>, `MidPt Mo` <dbl>, W_Yr <dbl>,
## #   W_Week <dbl>, W_Mo <dbl>, SeasNum <dbl>, Season <chr>
```

```r
WF_disas <- Eventdf %>%
  dplyr::select(`Disaster Group`, `Central Day`, Season, `Total CPI-Adjusted Cost (Millions of Dollars)
  group_by(`Disaster Group`) %>%
  mutate(Date = mdy(`Central Day`)) %>%
  filter(`Disaster Group` == "Winter/Freeze")

SS_disas <- Eventdf %>%
  dplyr::select(`Disaster Group`, `Central Day`, Season, `Total CPI-Adjusted Cost (Millions of Dollars)
  group_by(`Disaster Group`) %>%
  mutate(Date = mdy(`Central Day`)) %>%
  filter(`Disaster Group` == "SevStorm/Flood")

Cyc_disas <- Eventdf %>%
  dplyr::select(`Disaster Group`, `Central Day`, Season, `Total CPI-Adjusted Cost (Millions of Dollars)
  group_by(`Disaster Group`) %>%
  mutate(Date = mdy(`Central Day`)) %>%
```

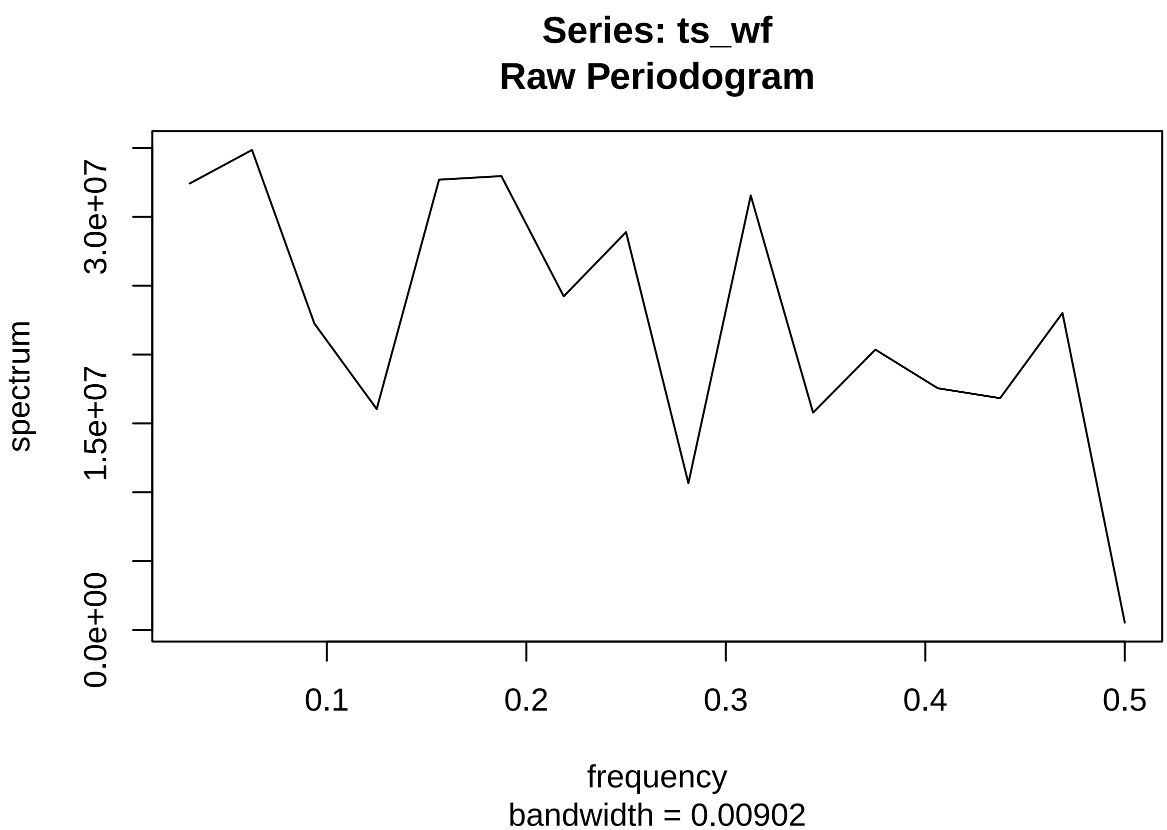
```
  filter(`Disaster Group` == "Tropical Cyclone")

Drought_disas <- Eventdf %>%
  dplyr::select(`Disaster Group`, `Central Day`, Season, `Total CPI-Adjusted Cost (Millions of Dollars)
  group_by(`Disaster Group`) %>%
  mutate(Date = mdy(`Central Day`)) %>%
  filter(`Disaster Group` == "WF/Drought")

ts_wf <- ts(WF_disas$`Total CPI-Adjusted Cost (Millions of Dollars)`, start = c(1981, 1))
ts_SS <- ts(SS_disas$`Total CPI-Adjusted Cost (Millions of Dollars)`, start = c(1980, 4))
ts_Cyc <- ts(Cyc_disas$`Total CPI-Adjusted Cost (Millions of Dollars)`, start = c(1980, 8))
ts_drought <- ts(Drought_disas$`Total CPI-Adjusted Cost (Millions of Dollars)`, start = c(1980, 8))
```

Question 1

1) (5pts) Run Periodogram on each time series based on disaster type. There are four disaster groups. Discuss your findings.

Winter Freeze Periodogram

```
# Winter Freeze Periodogram
spec.pgram(ts_wf,log="no",taper=0)
```

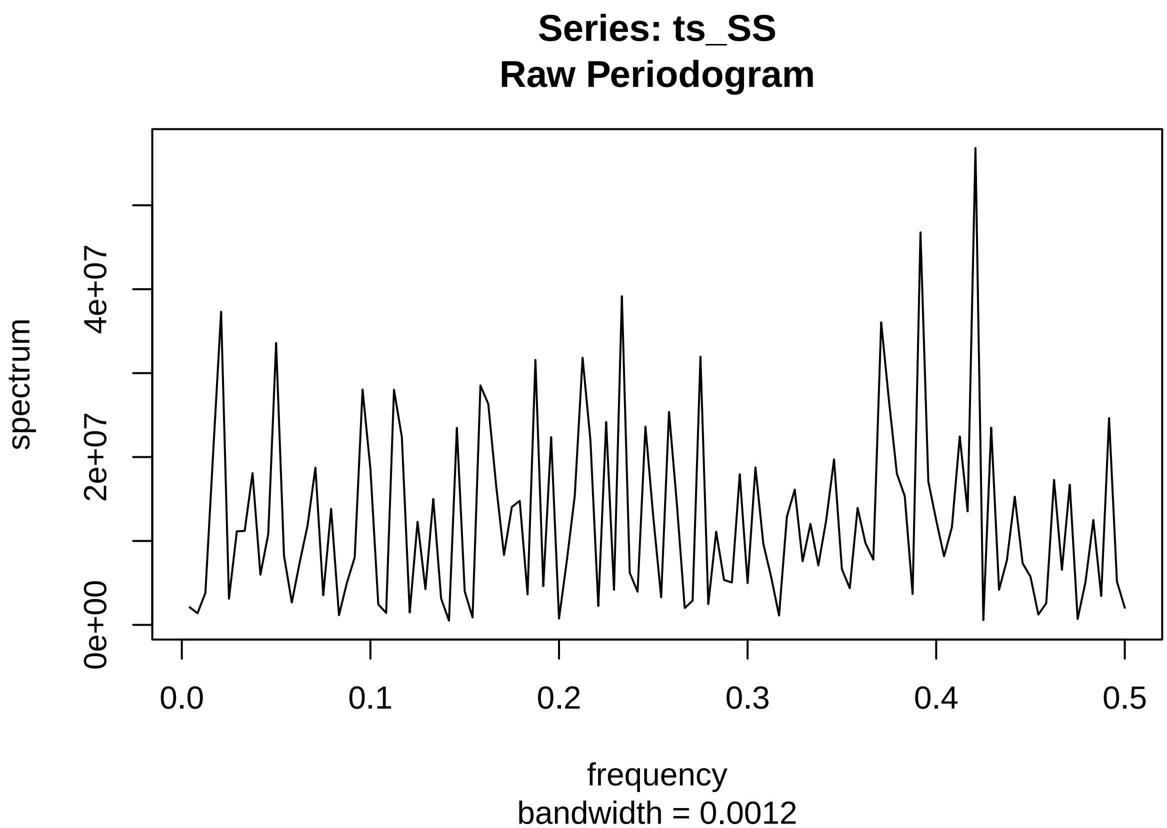


From the periodogram above we note a huge spike at frequency $\omega = 0.05$. Using the frequency above we will fit the following model $Y_t = \beta_0 + \beta_1 t + \beta_2 cos(2\pi\omega t) + \beta_3 sin(2\pi\omega t) + X_t$

I added the independent variable t to see whether there is a linear trend in disaster type. Fit lm() model and check.

Severse Storm Periodogram

```r
# Severe Storm Periodogram
spec.pgram(ts_SS,log="no",taper=0)
```

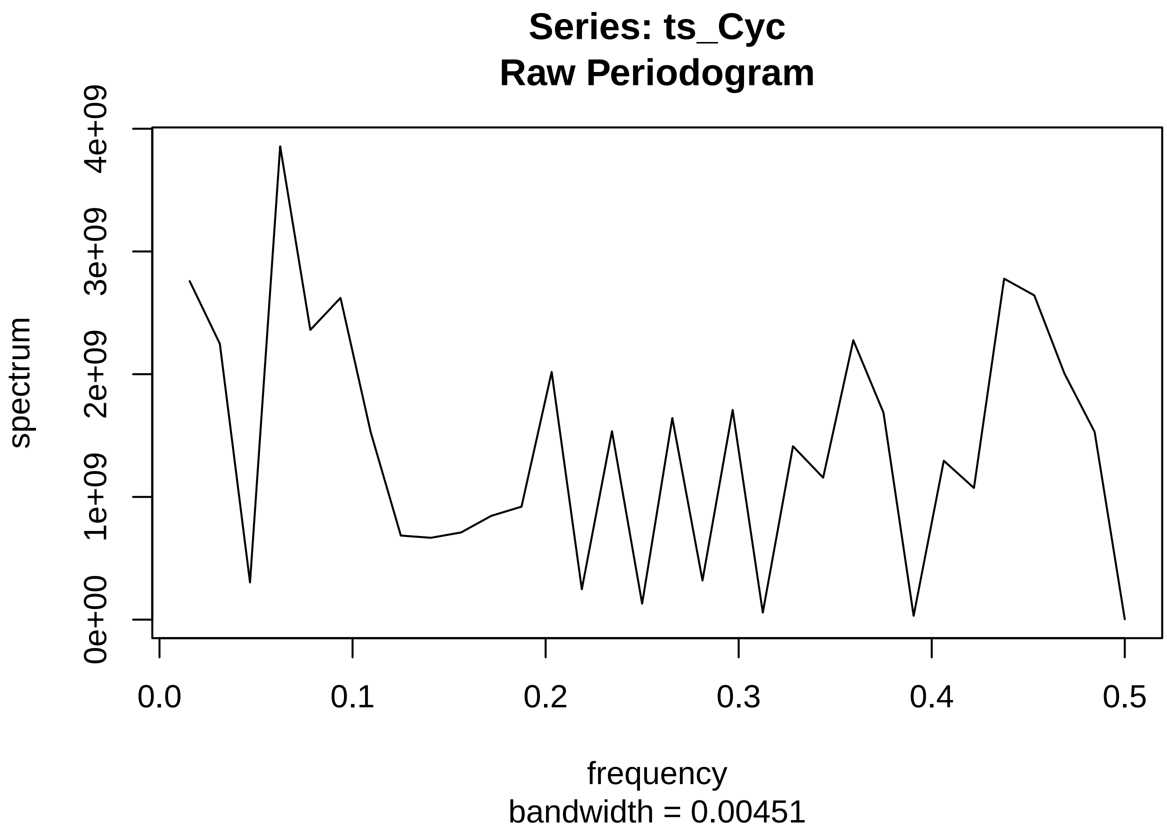


From the periodogram above we note a huge spike at frequency $\omega = 0.415$. Using the frequency above we will fit the following model $Y_t = \beta_0 + \beta_1 t + \beta_2 cos(2\pi\omega t) + \beta_3 sin(2\pi\omega t) + X_t$

I added the independent variable t to see whether there is a linear trend in disaster type. Fit lm() model and check.

Cyclone Periodogram

```r
# Cyclone Periodogram
spec.pgram(ts_Cyc,log="no",taper=0)
```

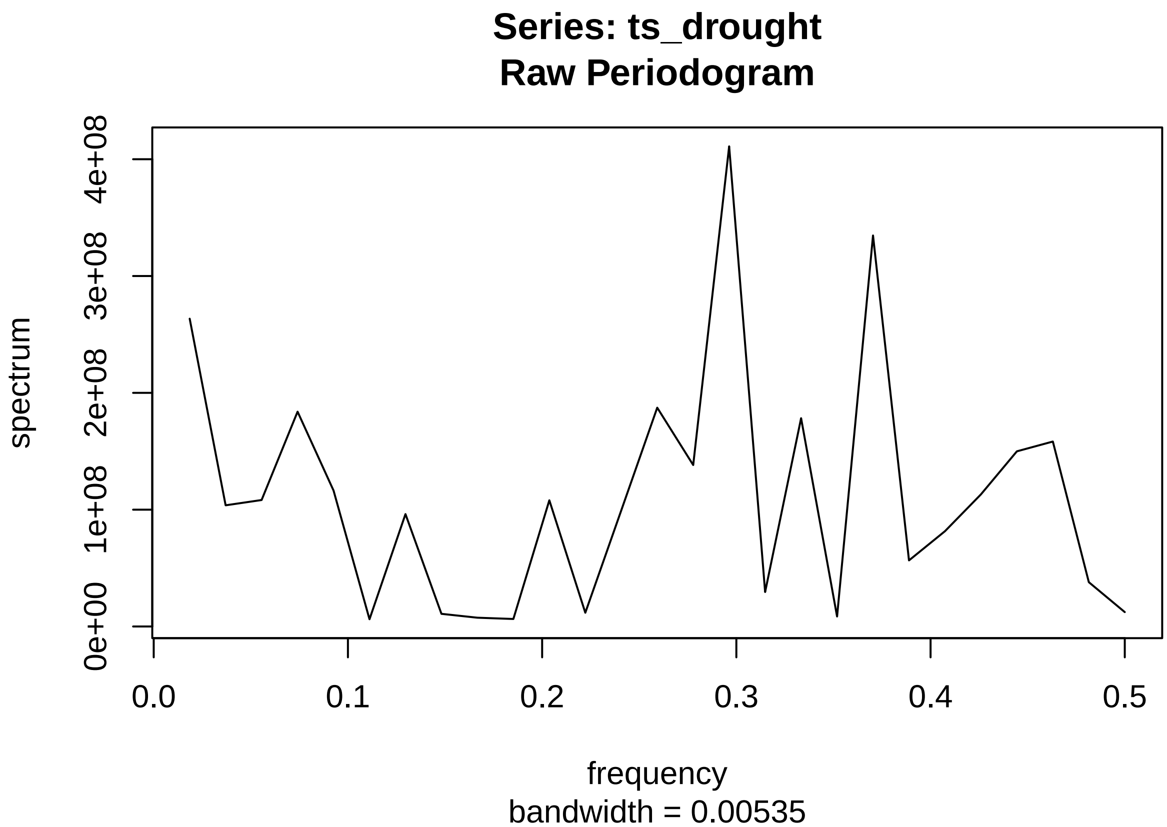


From the periodogram above we note a huge spike at frequency $\omega = 0.06$, we also notice another huge spike at 0.44, but we will analyze the fit model first, before working with the second spike Using the frequency above we will fit the following model $Y_t = \beta_0 + \beta_1 t + \beta_2 cos(2\pi\omega t) + \beta_3 sin(2\pi\omega t) + X_t$

I added the independent variable t to see whether there is a linear trend in disaster type. Fit lm() model and check.

Drought Periodogram

```
spec.pgram(ts_drought,log="no",taper=0)
```



From the periodogram above we note a huge spike at frequency $\omega = 0.3$. Using the frequency above we will fit the following model $Y_t = \beta_0 + \beta_1 t + \beta_2 cos(2\pi\omega t) + \beta_3 sin(2\pi\omega t) + X_t$

I added the independent variable t to see whether there is a linear trend in disaster type. Fit lm() model and check.

Question 2

2) (5pts) Perform a Dickey-Fuller test on each time series categorized by disaster type. Subsequently, discuss the outcomes and implications of your findings.

Dickey-Fuller Winter Freeze

```
adf.test(ts_wf)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_wf
## Dickey-Fuller = -1.2544, Lag order = 3, p-value = 0.8598
## alternative hypothesis: stationary
```

Here we see a p-value of 0.8598 this is greater than the 0.05 threshold, thus we will fail to reject the null hypothesis, and say that the Time Series where the disaster type is Winter Freeze, is non-stationary.

Dickey-Fuller SevereStorm

```r
adf.test(ts_SS)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_SS
## Dickey-Fuller = -5.713, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

Here we see a p-value <0.01 this is lower than the 0.05 threshold, thus we will reject the null hypothesis, and say that the Time Series for disaster type is SevereStorms, is stationary.

Dickey-Fuller Cyclone

```r
adf.test(ts_Cyc)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_Cyc
## Dickey-Fuller = -3.2254, Lag order = 3, p-value = 0.09177
## alternative hypothesis: stationary
```

Here we see a p-value of 0.09177 this is greater than the 0.05 threshold, thus we will fail to reject the null hypothesis, and say that the Time Series where the disaster type is Cycloe, is non-stationary.

Dickey-Fuller Drought

```r
adf.test(ts_drought)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_drought
## Dickey-Fuller = -4.6435, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

Here we see a p-value <0.01 this is lower than the 0.05 threshold, thus we will reject the null hypothesis, and say that the Time Series for disaster type is Drought, is stationary.

Question 3

3) (10pts) Based on your findings, propose the best fitting time series regression model for each disaster group. This model should include trend, sine (sin()), and cosine (cos()) components. Provide the equation of the model for each disaster group, and discuss your results. If your Periodogram reveals many spikes, focus on the most significant ones when building the model.

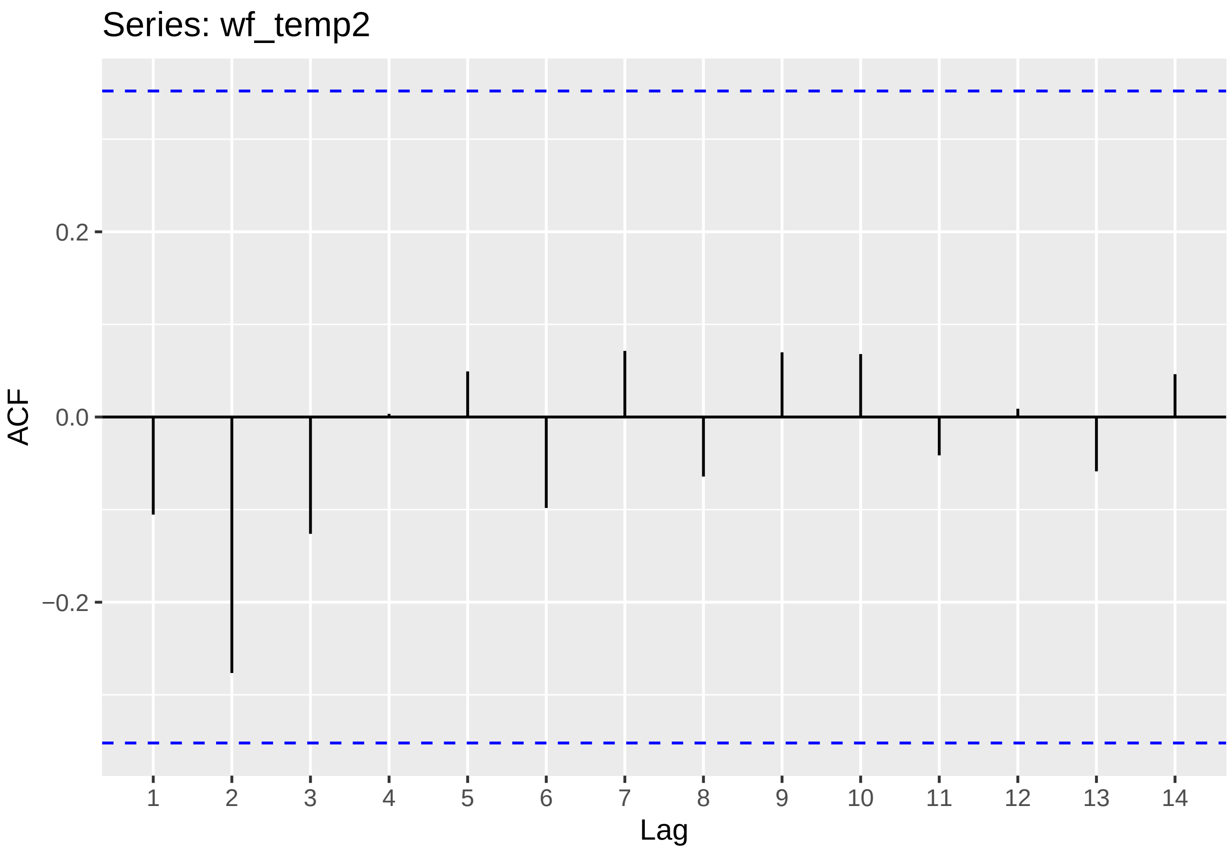
Building Model for Winter Freeze

From the periodogram above we will now make our model.

```r
t=1:length(ts_wf)
w=0.05
cs=cos(2*pi*w*t)
si=sin(2*pi*w*t)

fit_wf=lm(ts_wf~t+cs+si)
summary(fit_wf)
```

```
##
## Call:
## lm(formula = ts_wf ~ t + cs + si)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5441.5 -1417.5  -415.9   773.4 19290.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3806.47    1719.63   2.214   0.0355 *
## t              20.15      93.16   0.216   0.8303
## cs          -2974.53    1166.99  -2.549   0.0168 *
## si            105.50    1226.01   0.086   0.9321
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4565 on 27 degrees of freedom
## Multiple R-squared:  0.2042, Adjusted R-squared:  0.1157
## F-statistic: 2.309 on 3 and 27 DF,  p-value: 0.09892
```

```r
wf_resi <- fit_wf$residuals
wf_temp2 <- ts(wf_resi, start = c(1981, 1))
ggAcf(wf_temp2)
```



From the results above, we observe that the only significant term is the cosine, so we will drop the seasonal trend t and the sine from the model.

From the ACF plot we see that there are no lag values this appear to be affecting the data.

```
fit_wf_final <- lm(ts_wf ~ cs)
summary(fit_wf_final)
```

```
##
## Call:
## lm(formula = ts_wf ~ cs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5226.4 -1467.2  -319.9   761.2 19530.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4147        795   5.216 1.39e-05 ***
## cs             -3013       1110  -2.715   0.0111 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4409 on 29 degrees of freedom
## Multiple R-squared:  0.2026, Adjusted R-squared:  0.1751
## F-statistic: 7.369 on 1 and 29 DF,  p-value: 0.01106
```

```r
wf_resi_final <- fit_wf_final$residuals
```

Usually the frequencies around zero are not of our interest. Thus the model that we will use to remove the seasonal trend is $Y_t = 4147 - 3013cos(2\pi 0.3t) + X_t$

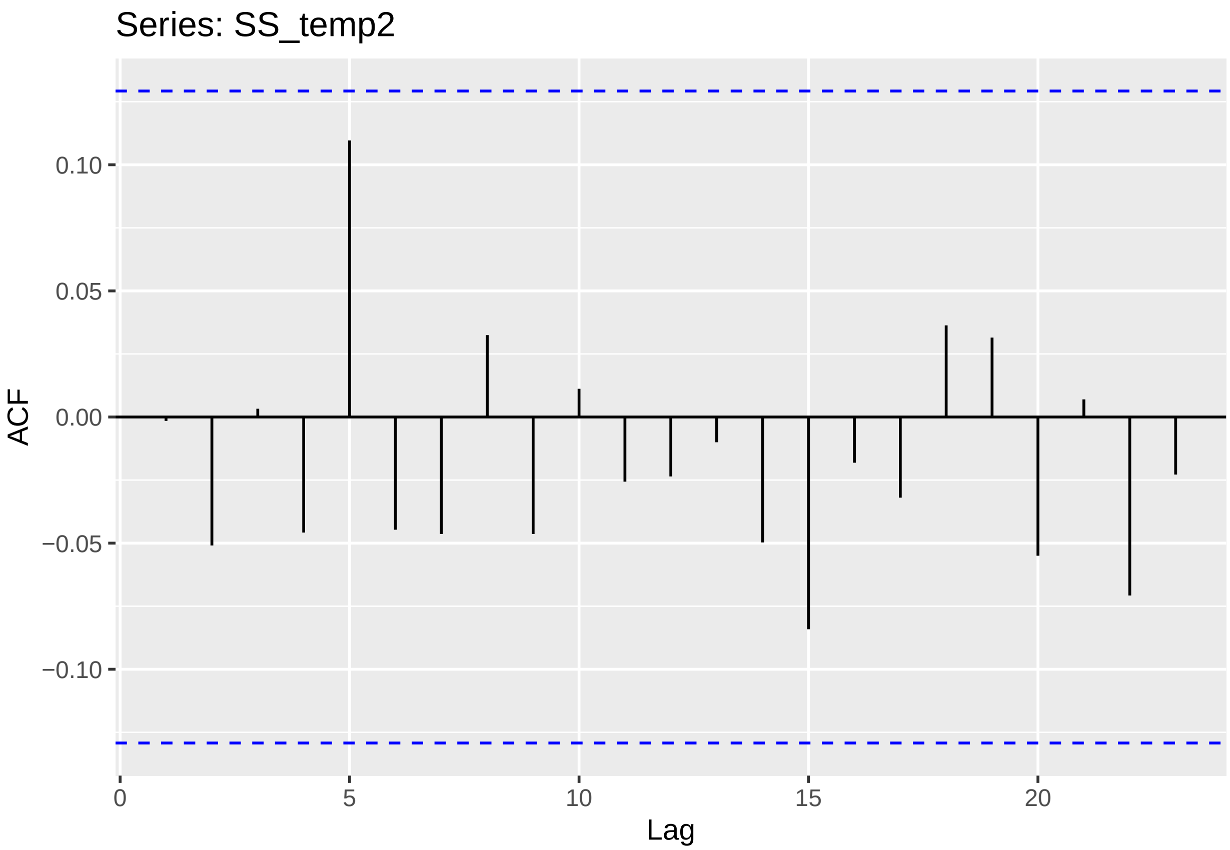
Now, we will proceed by finding the best time series model for X_t. However, first we will move on to the other disaster types.

Building Model for SevereStorm

```r
w=0.415
t=1:length(ts_SS)
cs=cos(2*pi*w*t)
si=sin(2*pi*w*t)
fit_SS=lm(ts_SS~t+cs+si)
summary(fit_SS)
```

```
##
## Call:
## lm(formula = ts_SS ~ t + cs + si)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -2694  -1409   -701    382  41066
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3369.656    468.528   7.192 9.27e-12 ***
## t             -4.681      3.517  -1.331   0.1845
## cs          -666.179    329.936  -2.019   0.0447 *
## si          -353.168    330.509  -1.069   0.2864
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3541 on 226 degrees of freedom
## Multiple R-squared:   0.03,  Adjusted R-squared:  0.01713
## F-statistic:  2.33 on 3 and 226 DF,  p-value: 0.07518
```

```r
SS_resi <- fit_SS$residuals
SS_temp2 <- ts(SS_resi, start = c(1980, 4))
ggAcf(SS_temp2)
```



From the results above, we observe that the only significant term is the cosine, so we will drop the seasonal trend t and the sine from the model.

From the ACF plot we see that there are no lag values this appear to be affecting the data.

We will now build our final model.

```
fit_ss_final <- lm(ts_SS ~ cs)
summary(fit_ss_final)
```

```
##
## Call:
## lm(formula = ts_SS ~ cs)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -2261  -1341   -732    314  41598
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2828.4      234.0  12.088   <2e-16 ***
## cs            -665.8      330.6  -2.014   0.0452 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3548 on 228 degrees of freedom
## Multiple R-squared:  0.01748,    Adjusted R-squared:  0.01317
## F-statistic: 4.056 on 1 and 228 DF,  p-value: 0.04519
```

```r
SS_resi_final <- fit_ss_final$residuals
```

Usually the frequencies around zero are not of our interest. Thus the model that we will use to remove the seasonal trend is $Y_t = 2828.4 - 665.8cos(2\pi 0.415t) + X_t$

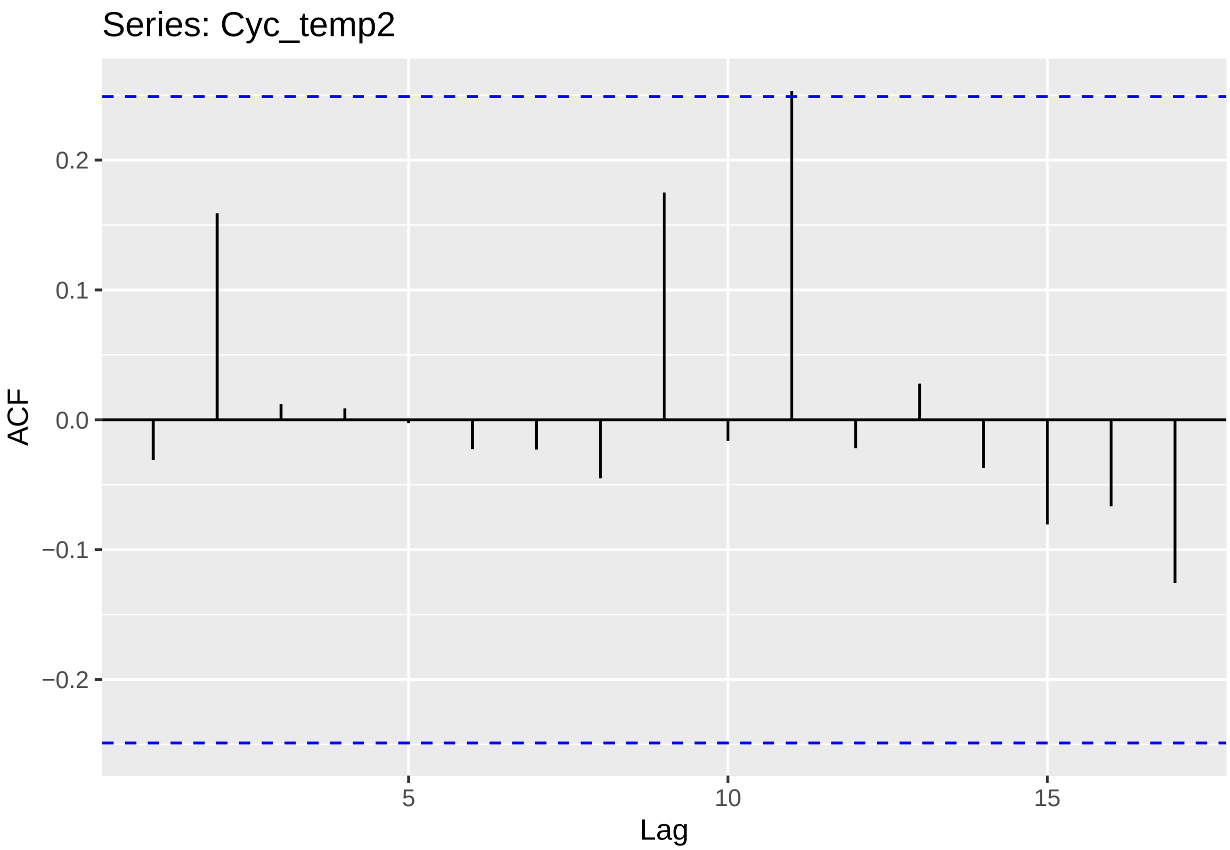
Now, we will proceed by finding the best time series model for X_t. However, first we will move on to the other disaster types.

Building Model for Cyclone Disaster Type

```r
t=1:length(ts_Cyc)
w=0.06
cs=cos(2*pi*w*t)
si=sin(2*pi*w*t)
fit_Cyc=lm(ts_Cyc~t+cs+si)
summary(fit_Cyc)
```

```
##
## Call:
## lm(formula = ts_Cyc ~ t + cs + si)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -40757 -19968  -9013   3935 171464
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13333.3     9682.3   1.377   0.1738
## t              261.7      267.6   0.978   0.3322
## cs          -14514.5     6782.1  -2.140   0.0366 *
## si           -1751.9     6735.2  -0.260   0.7957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37320 on 58 degrees of freedom
## Multiple R-squared:  0.09642,    Adjusted R-squared:  0.04968
## F-statistic: 2.063 on 3 and 58 DF,  p-value: 0.115
```

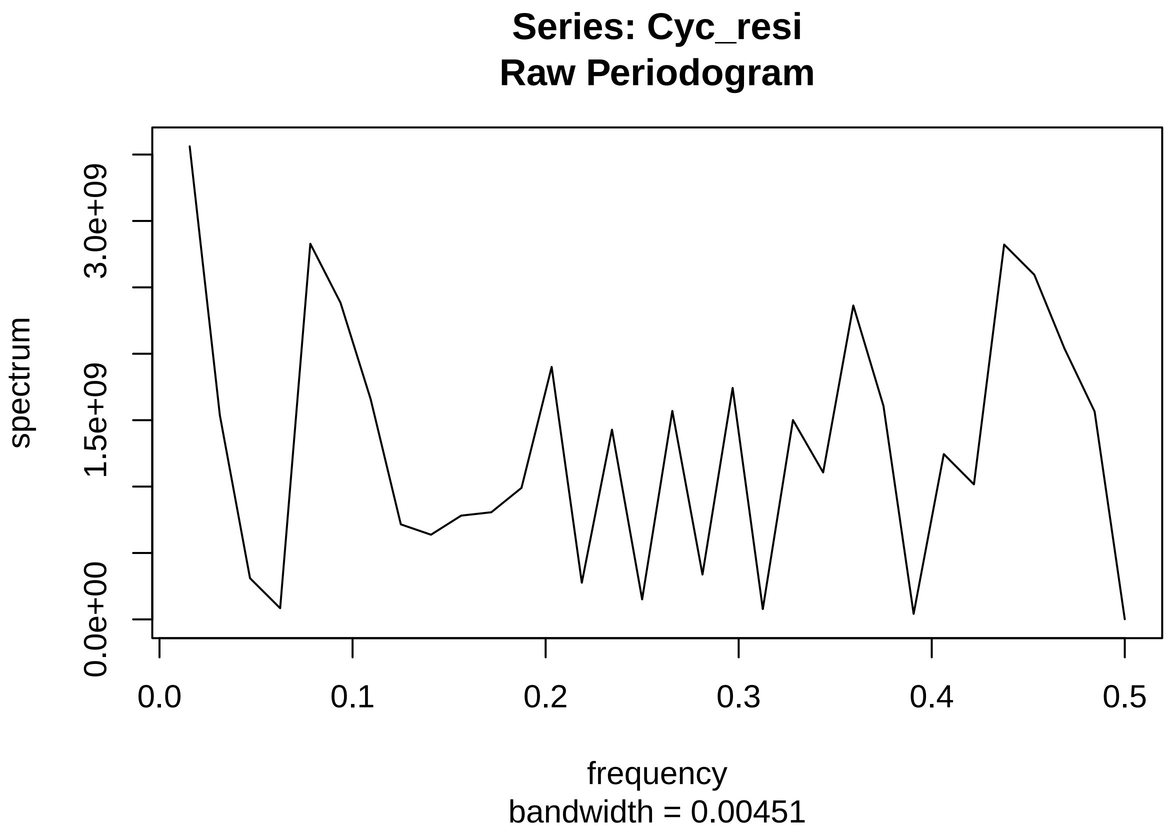
```r
Cyc_resi <- fit_Cyc$residuals
Cyc_temp2 <- ts(Cyc_resi, start = c(1980, 8))
ggAcf(Cyc_temp2)
```



From the results above, we observe that the only significant term is the cosine, so we will drop the seasonal trend t and the sine from the model.

From the ACF plot we see that lag 11 is slightly above the threshold and seems have an affect on the data so we will make a Periodogram to see any trends.

```
spec.pgram(Cyc_resi,log="no",taper=0)
```



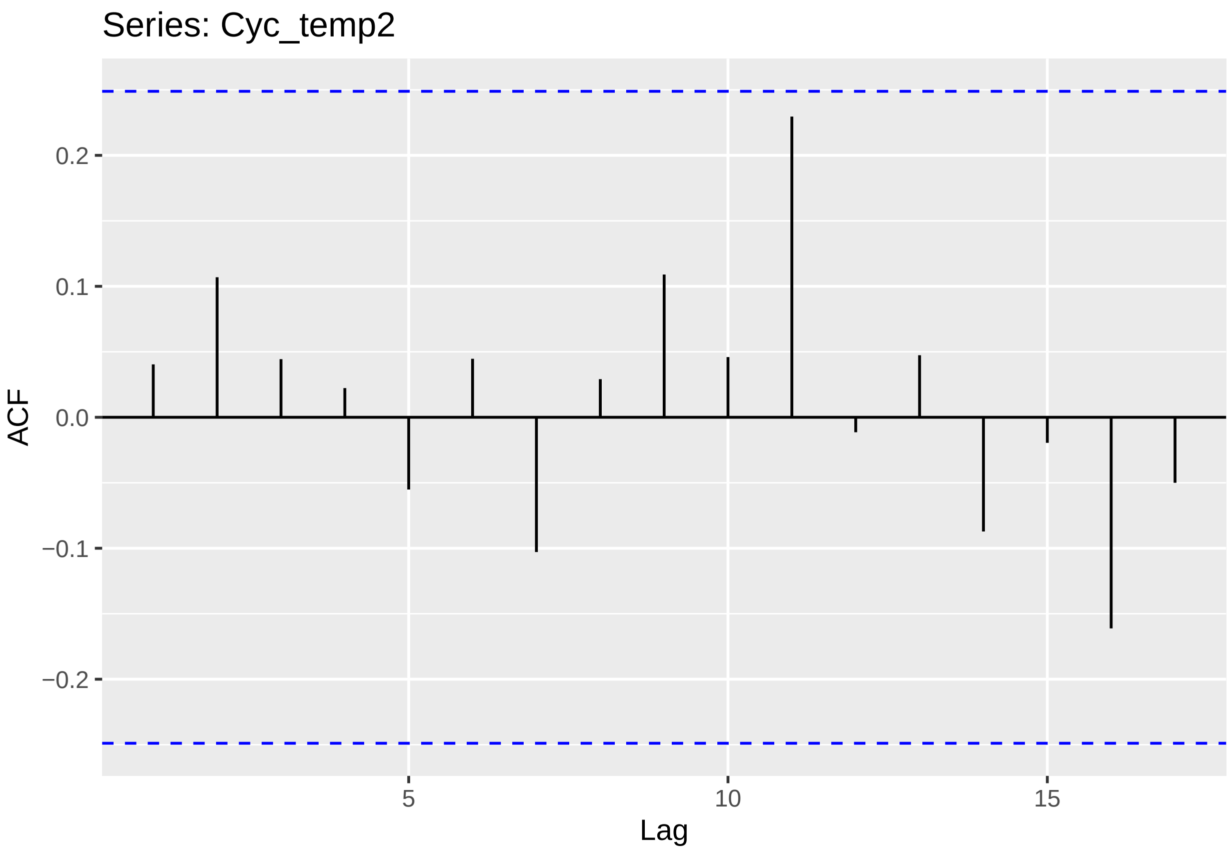
From the periodogram above we note a huge spike at frequency $\omega = 0.44$. We will now fit another model with added cosine and sine values for our second ω.

```
w1=0.44
cs1=cos(2*pi*w1*t)
si1=sin(2*pi*w1*t)
fit_Cyc=lm(ts_Cyc~t+cs+si+cs1+si1)
summary(fit_Cyc)
```

```
##
## Call:
## lm(formula = ts_Cyc ~ t + cs + si + cs1 + si1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45375 -17954  -3892   5402 157221
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13138.6     9462.2   1.389   0.1705
## t              272.0      261.6   1.040   0.3029
## cs          -14465.4     6627.2  -2.183   0.0333 *
## si           -1977.8     6581.3  -0.301   0.7649
## cs1          -2602.5     6583.0  -0.395   0.6941
## si1         -14109.0     6530.0  -2.161   0.0350 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36460 on 56 degrees of freedom
```

```
## Multiple R-squared:  0.1672, Adjusted R-squared:  0.09286
## F-statistic: 2.249 on 5 and 56 DF,  p-value: 0.0619
```

```
Cyc_resi <- fit_Cyc$residuals
Cyc_temp2 <- ts(Cyc_resi, start = c(1980, 8))
ggAcf(Cyc_temp2)
```



From the results above, we observe that there are two significant terms, those being cosine and sine1, so we will drop the seasonal trend t, sine, and cosine1 from the model.

From the ACF plot we see that there are no lag values this appear to be affecting the data.

We will now build our final model

```
fit_Cyc_final=lm(ts_Cyc~cs+si1)
summary(fit_Cyc_final)
```

```
##
## Call:
## lm(formula = ts_Cyc ~ cs + si1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -40503 -20428  -5713   6128 158665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    21604       4579   4.718 1.51e-05 ***
## cs            -15172       6502  -2.334   0.0230 *
```

```
## si1           -13852       6433  -2.153   0.0354 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35960 on 59 degrees of freedom
## Multiple R-squared:  0.1465, Adjusted R-squared:  0.1175
## F-statistic: 5.062 on 2 and 59 DF,  p-value: 0.009355
```

```
Cyc_resi_final <- fit_Cyc_final$residuals
```

Usually the frequencies around zero are not of our interest. Thus the model that we will use to remove the seasonal trend is $Y_t = 21604 - 15172cos(2\pi 0.06t) - 13852sin(2\pi 0.44t)X_t$

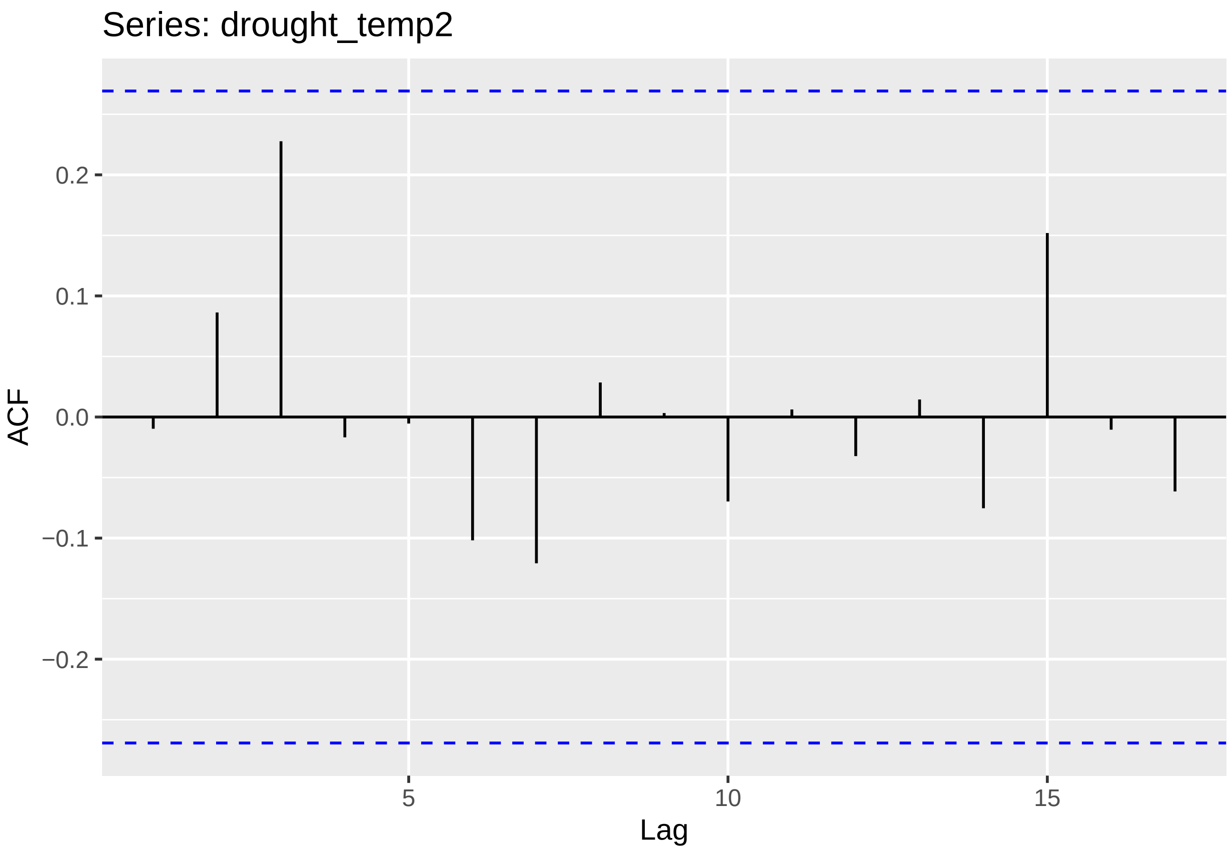
Now, we will proceed by finding the best time series model for X_t. However, first we will move on to the last disaster types.

Building model for Drought Disaster Type

```
t=1:length(ts_drought)
w=0.3
cs=cos(2*pi*w*t)
si=sin(2*pi*w*t)
fit_drought=lm(ts_drought~t+cs+si)
summary(fit_drought)
```

```
##
## Call:
## lm(formula = ts_drought ~ t + cs + si)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -13364  -5371  -1987   1780  38295
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8949.00    2804.51   3.191  0.00248 **
## t              15.26      90.41   0.169  0.86662
## cs            -69.91    1957.58  -0.036  0.97166
## si           6021.28    1951.40   3.086  0.00334 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10050 on 49 degrees of freedom
## Multiple R-squared:  0.1628, Adjusted R-squared:  0.1115
## F-statistic: 3.175 on 3 and 49 DF,  p-value: 0.03219
```

```
drought_resi <- fit_drought$residuals
drought_temp2 <- ts(drought_resi, start = c(1980, 8))
ggAcf(drought_temp2)
```



From the results above, we observe that the only significant term is the sine, so we will drop the seasonal trend t and the cosine from the model.

From the ACF plot we see that there are no lag values this appear to be affecting the data.

We will now build our final model.

```
fit_drought_final=lm(ts_drought~si)
summary(fit_drought_final)
```

```
##
## Call:
## lm(formula = ts_drought ~ si)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -13574  -5534  -2123   1730  37935
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9362       1354   6.914 7.44e-09 ***
## si              6008       1911   3.143  0.00279 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9857 on 51 degrees of freedom
## Multiple R-squared:  0.1623, Adjusted R-squared:  0.1458
## F-statistic: 9.879 on 1 and 51 DF,  p-value: 0.002785
```

```
drought_resi_final <- fit_drought_final$residuals
```

Usually the frequencies around zero are not of our interest. Thus the model that we will use to remove the seasonal trend is $Y_t = 9362 + 6008 sin(2\pi 0.3t) + X_t$

Now, we will proceed by finding the best time series model for X_t.

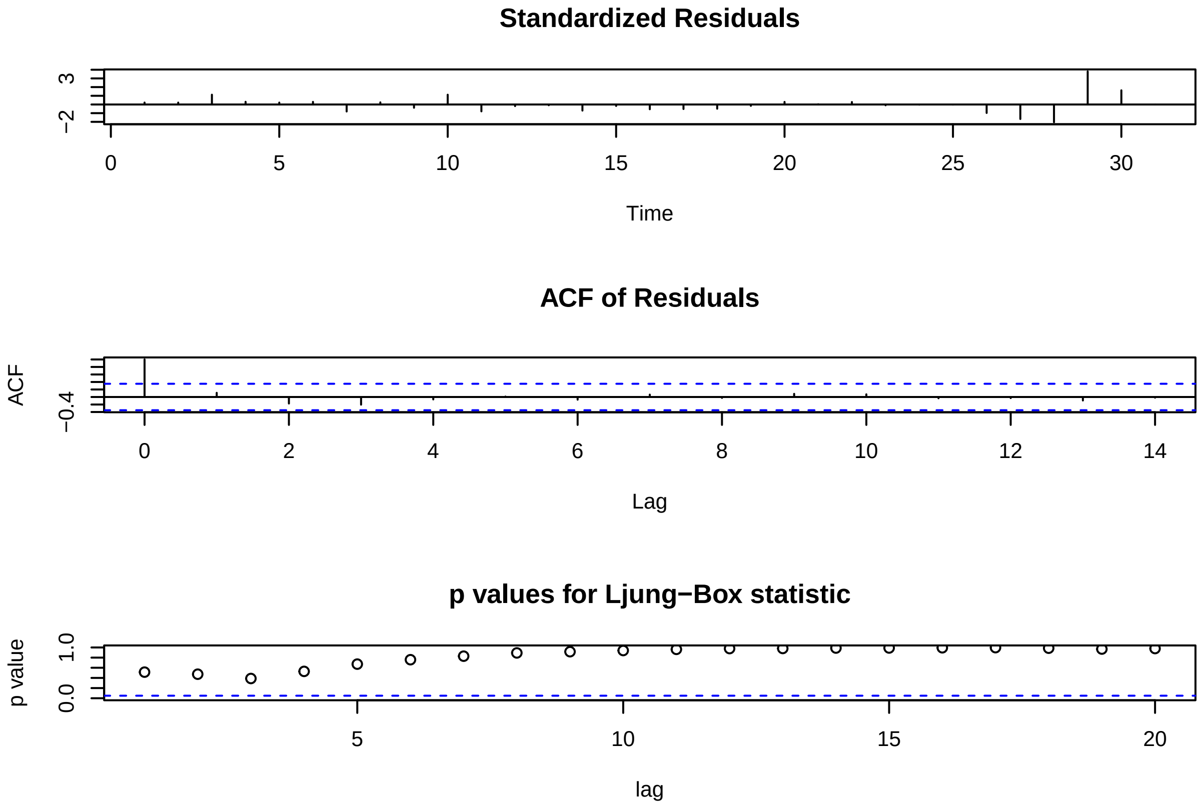
Question 4

4) (20pts) Find the best SARIMA model for each disaster group. Provide the equation of this model by disaster group. Discuss your results.

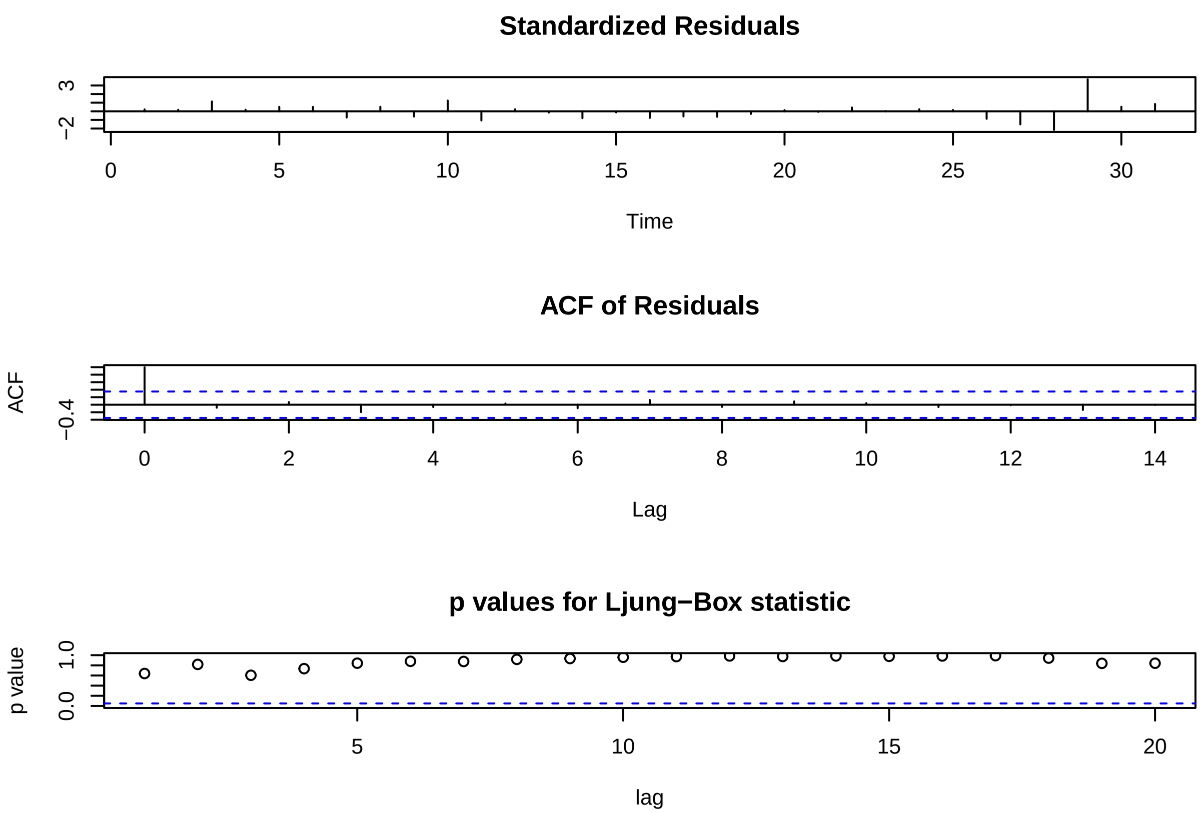
```
sarima=function(data,p,d,q,P=0,D=0,Q=0,S=-1){
  n=length(data)
  constant=1:n
  xmean=matrix(1,n,1)
  if (d>0)
  fitit=arima(data, order=c(p,d,q), seasonal=list(order=c(P,D,Q), period=S),xreg=constant,include.mean=F
  if (d<.00001)
  fitit=arima(data, order=c(p,d,q), seasonal=list(order=c(P,D,Q), period=S),xreg=xmean,include.mean=F)
  if (d+D>1)
  fitit=arima(data, order=c(p,d,q), seasonal=list(order=c(P,D,Q), period=S))
  if (S < 0) goof=20 else goof=3*S
  tsdiag(fitit,gof.lag=goof)
  k=length(fitit$coef)
  BIC=log(fitit$sigma2)+(k*log(n)/n)
  AICc=log(fitit$sigma2)+((n+k)/(n-k-2))
  AIC=log(fitit$sigma2)+((n+2*k)/n)
  list(fit=fitit, AIC=AIC, AICc=AICc, BIC=BIC)
}
```

Disaster Group Winter Freeze

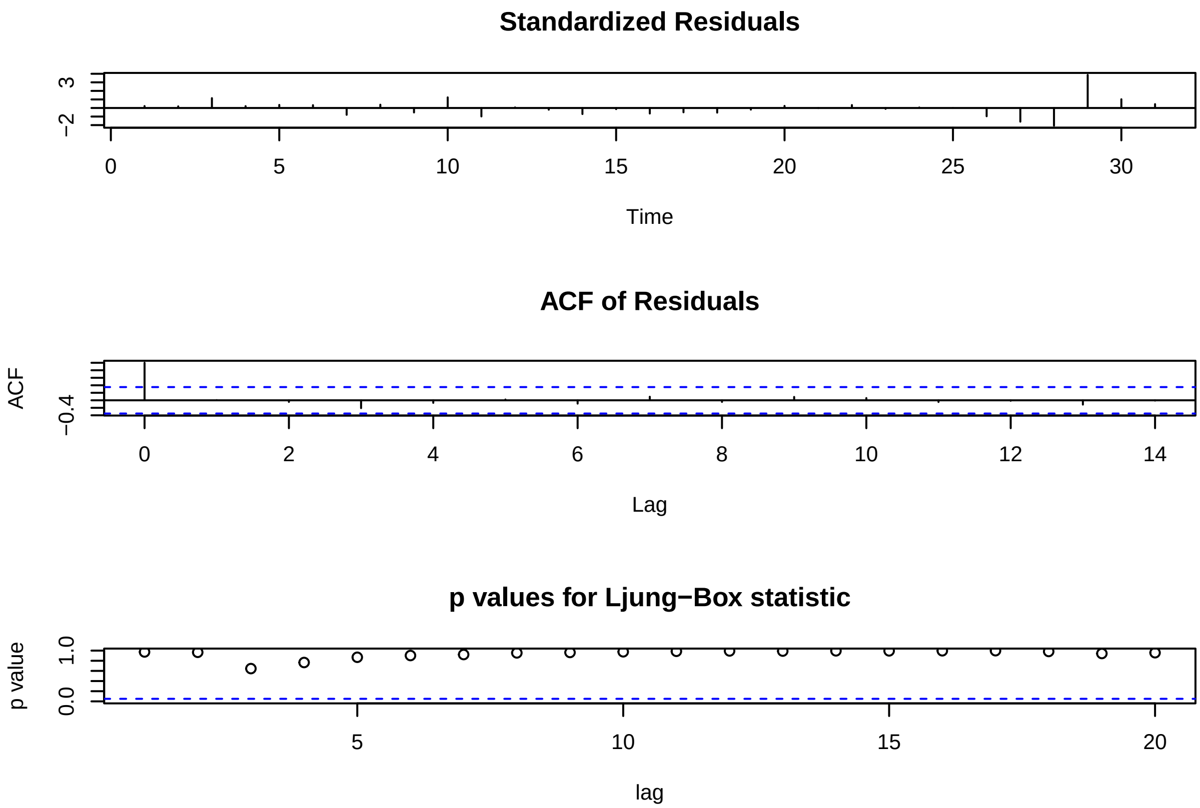
```
wf_fit1=sarima(wf_resi_final,1,0,1)
```



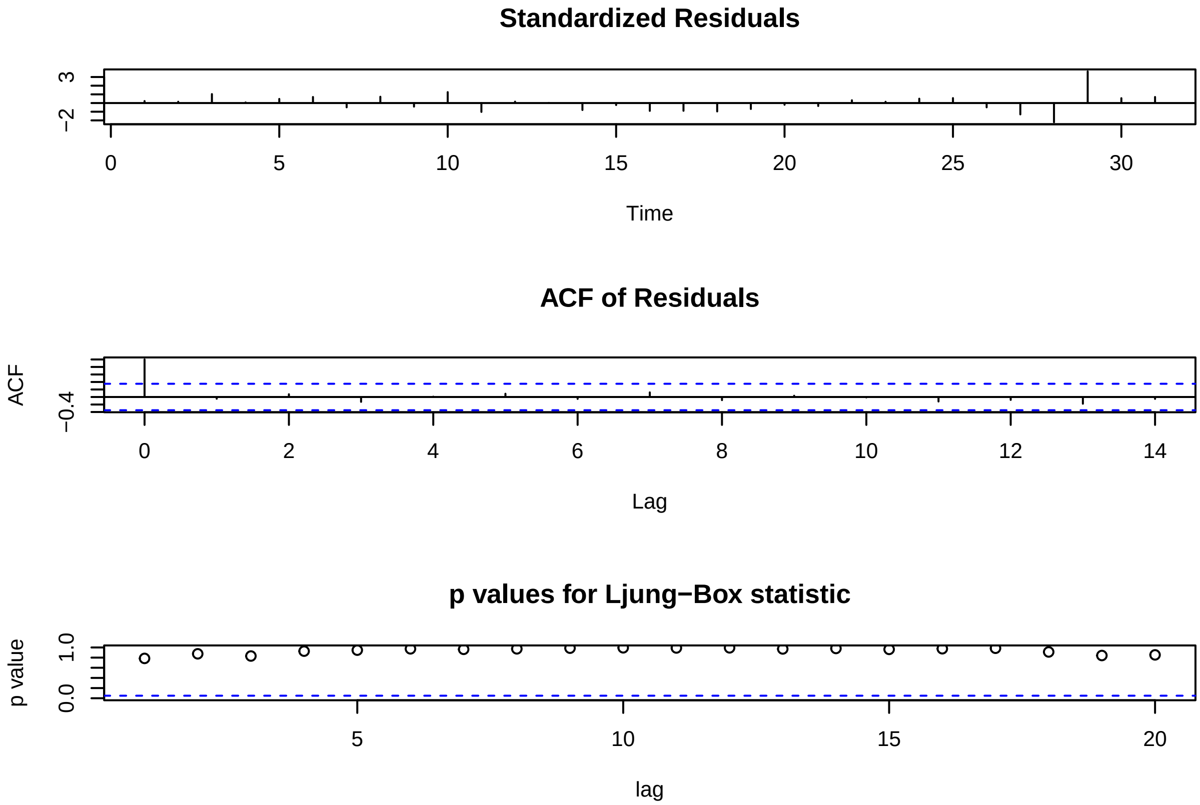

```r
wf_fit2=sarima(wf_resi_final,2,0,1)
```



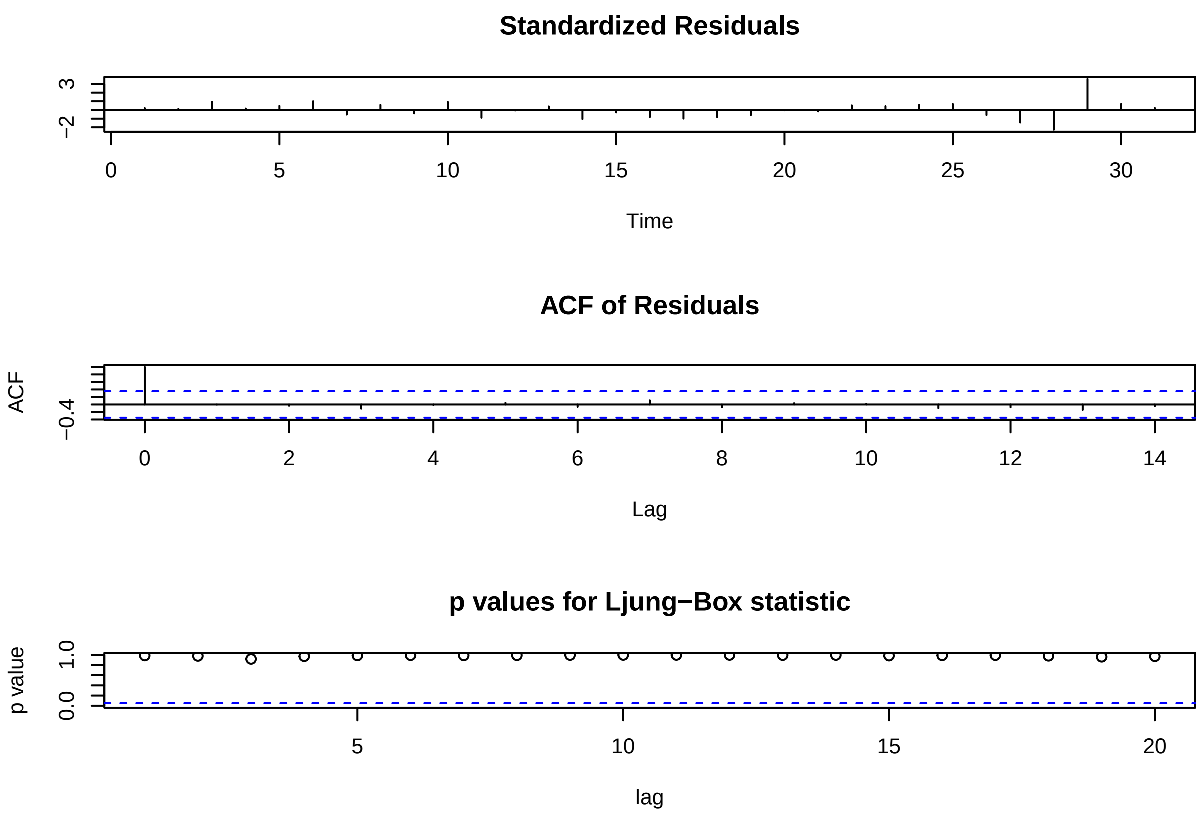
```
wf_fit3=sarima(wf_resi_final,1,0,2)
```



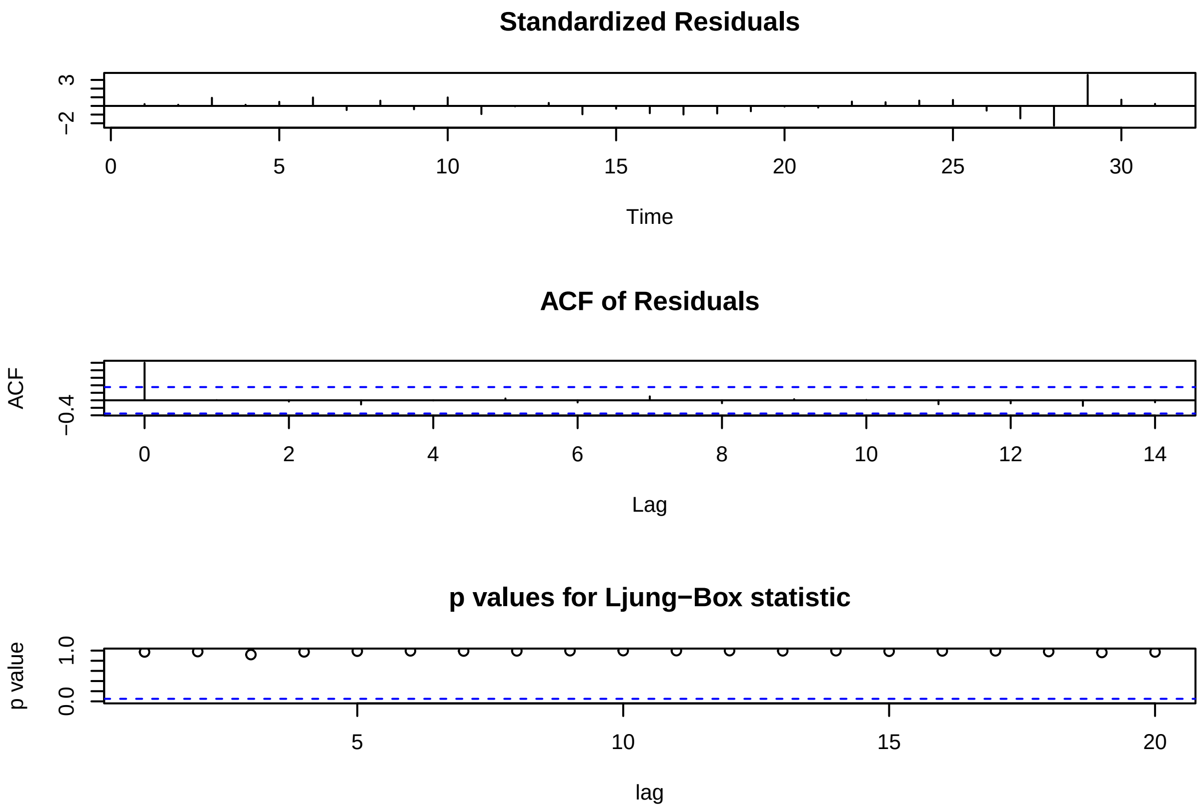

```
wf_fit4=sarima(wf_resi_final,2,0,2)
```



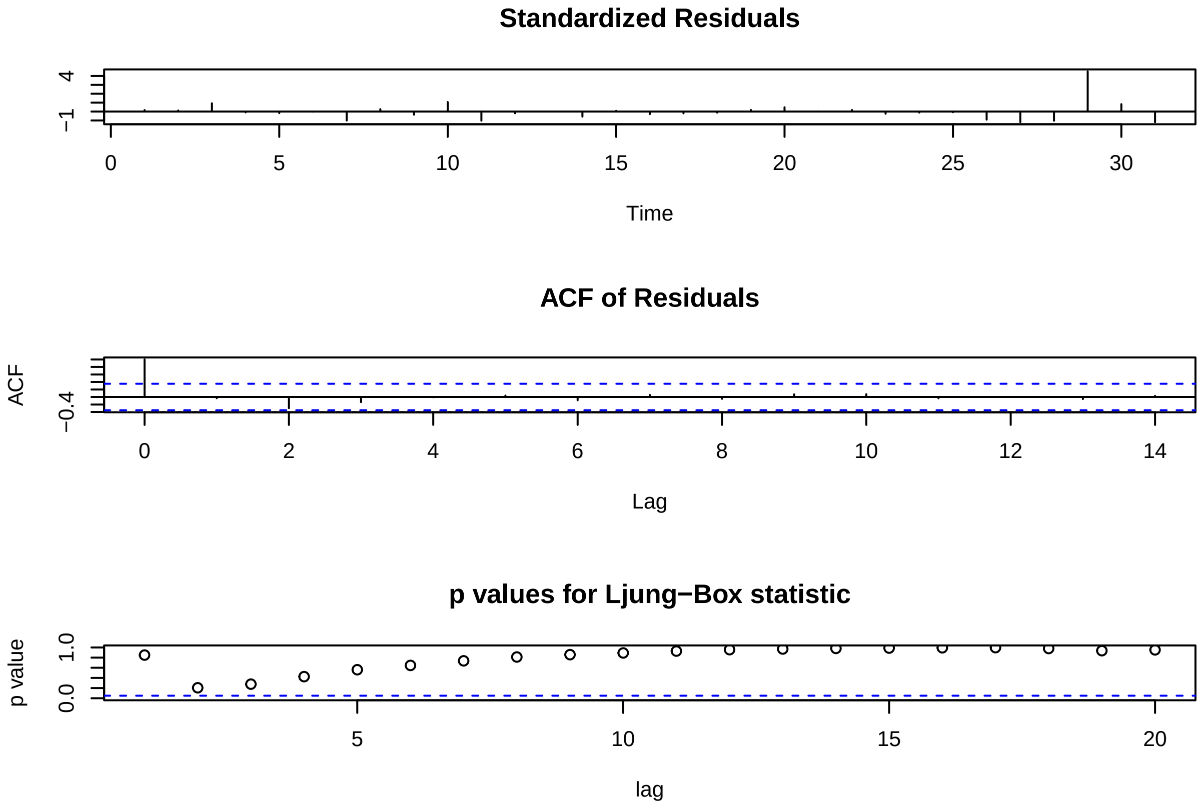

```
wf_fit5=sarima(wf_resi_final,3,0,2)
```



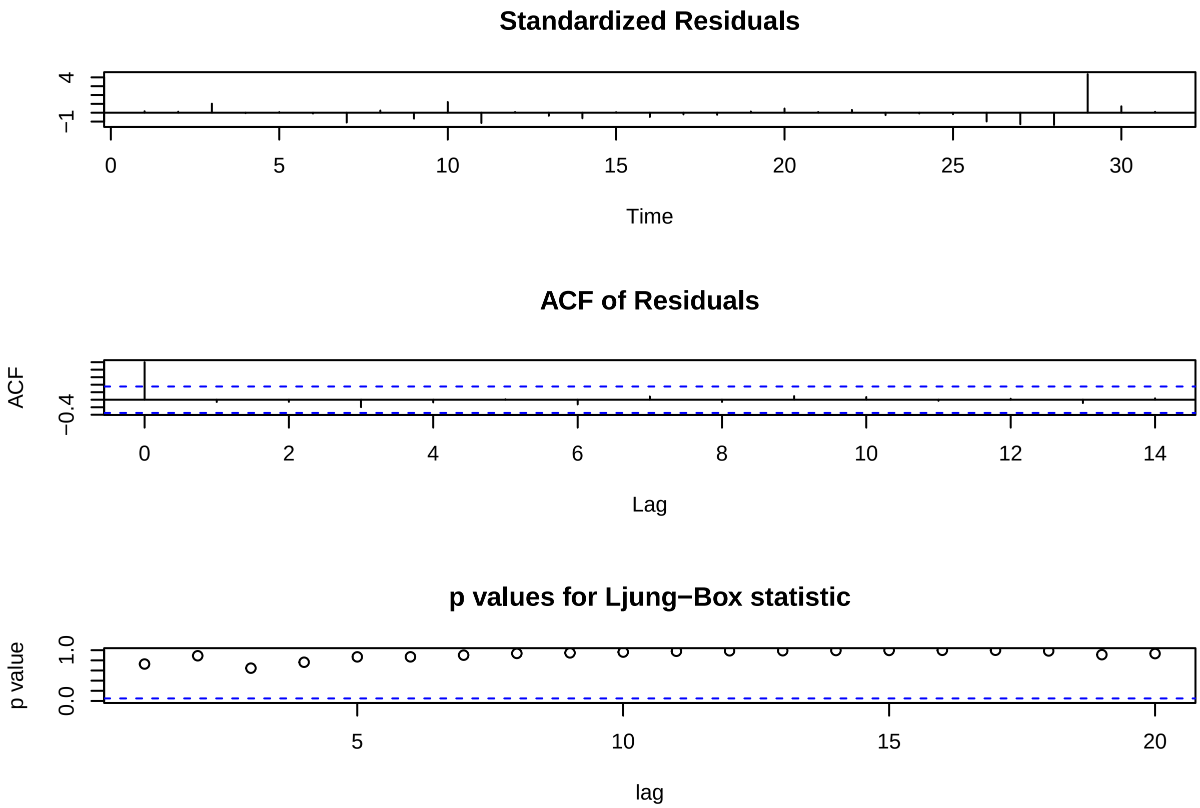
```r
wf_fit6=sarima(wf_resi_final,3,0,1)
```



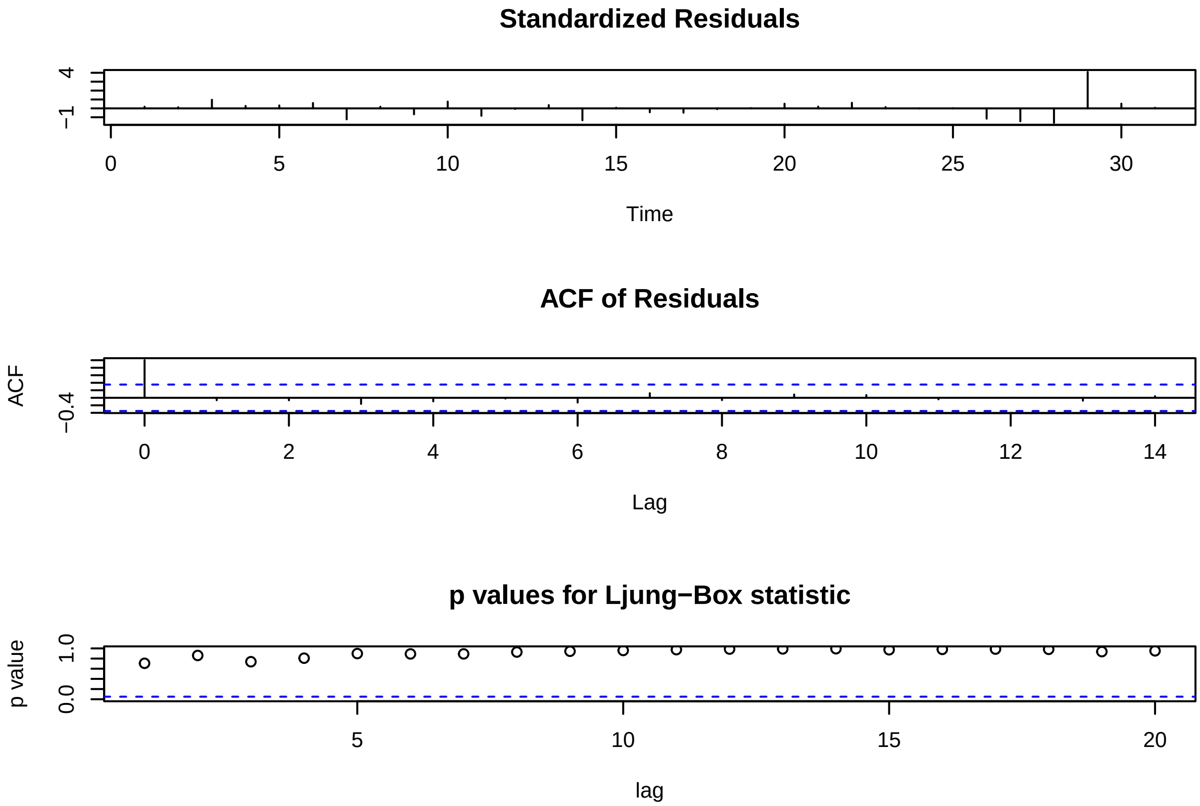

```r
wf_fit7=sarima(wf_resi_final,1,0,0)
```




```r
wf_fit8=sarima(wf_resi_final,2,0,0)
```



```
wf_fit9=sarima(wf_resi_final,3,0,0)
```



We can also loop through the possible models ARMA(p,q) up to lag 3. That is, we check an ARMA(1,0), ARMA(2,0), . . . , ARMA(1,1), ARMA(1,2), ARMA(3,1) and so on. . . .

```
#This little function extracts the
#AIC, AICc and BIC values from an Arima() fit
getAIC <- function(fit) {
  c(fit$AIC, fit$AICc, fit$BIC)
}
```

We will summarize the AIC-related results in a table and display the table

```
tab <- rbind(getAIC(wf_fit1), getAIC(wf_fit2), getAIC(wf_fit3),
             getAIC(wf_fit4), getAIC(wf_fit5), getAIC(wf_fit6),
            getAIC(wf_fit7), getAIC(wf_fit8), getAIC(wf_fit9))
colnames(tab) <- c("AIC", "AICc", "BIC")
rownames(tab) <- c("ARMA(1,1)","ARMA(2,1)","ARMA(1,2)","ARMA(2,2)",
 "ARMA(3,2)","ARMA(3,1)","AR(1)", "AR(2)", "AR(3)")
kable(tab)  #displays the table
```

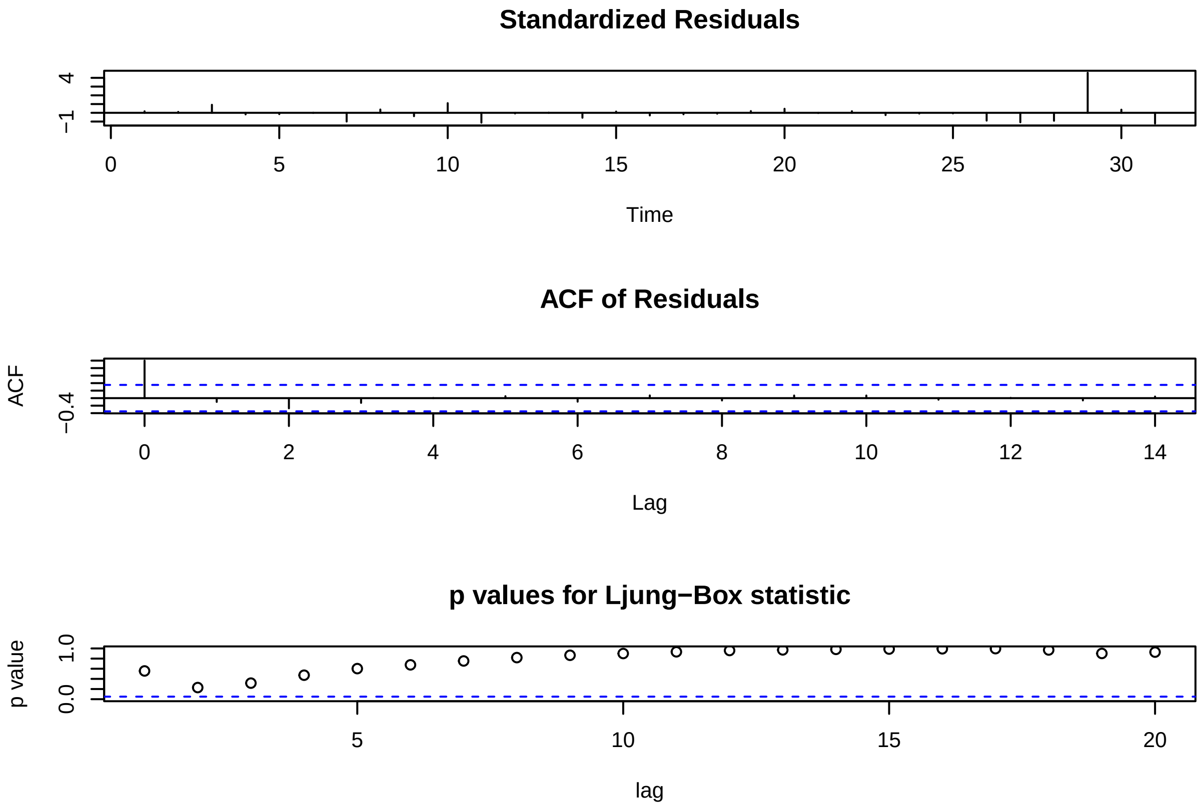
	AIC	AICc	BIC
ARMA(1,1)	17.60284	17.71698	16.74161
ARMA(2,1)	17.53477	17.67671	16.71980
ARMA(1,2)	17.60990	17.75184	16.79493
ARMA(2,2)	17.45991	17.63733	16.69120
ARMA(3,2)	17.49190	17.71350	16.76945
ARMA(3,1)	17.42725	17.60467	16.65854
AR(1)	17.83458	17.92777	16.92709

	AIC	AICc	BIC
AR(2)	17.81051	17.92466	16.94929
AR(3)	17.69702	17.83895	16.88205

From the table, using a combination of the AIC, AICc, and BIC values reported (remember when the values are within 2 of one another, they are essentially the same). We see that all of these models are the same. We want to choose the simplest model.

For now, let us consider the $AR(1)$ model for simplicity.

```
wf_fit_zero_mean=sarima(wf_resi_final,0,0,0)
```




```
ggAcf(wf_fit_zero_mean$fit$resid)
```


```
finalfit <- Arima(ts_wf, order=c(0, 0, 0))
finalfit
```

```
## Series: ts_wf
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##            mean
##       4336.5774
## s.e.   857.7928
##
## sigma^2 = 23570335:  log likelihood = -306.6
## AIC=617.2   AICc=617.63   BIC=620.07
```

Our best model comes out at as a non-zero mean ARIMA(0,0,0), this is an indication of a constant mean. Our model equation being

$$X_t = 4336.5774$$

To validate our model along with check for seasonality we will call the auto.arima() function.

```
auto.arima(ts_wf)
```

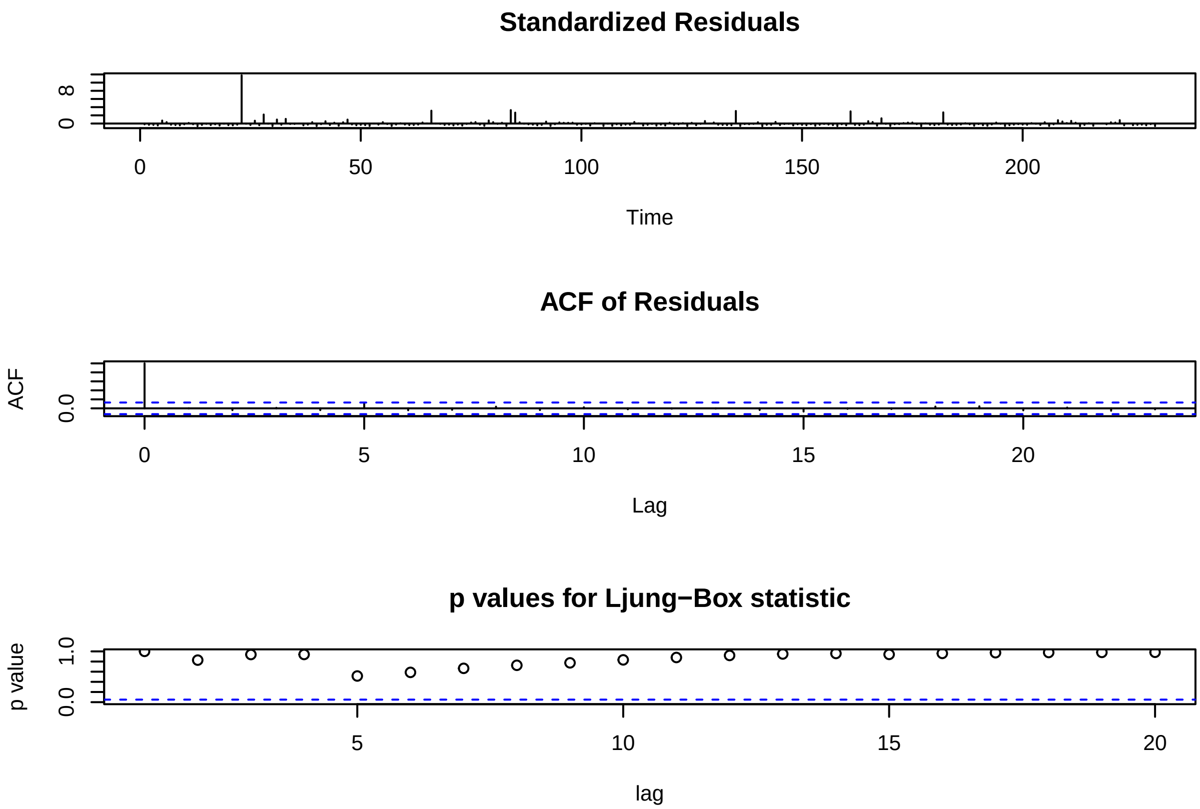
```
## Series: ts_wf
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##            mean
##       4336.5774
## s.e.   857.7928
```

```
##
## sigma^2 = 23570335:  log likelihood = -306.6
## AIC=617.2   AICc=617.63   BIC=620.07
```

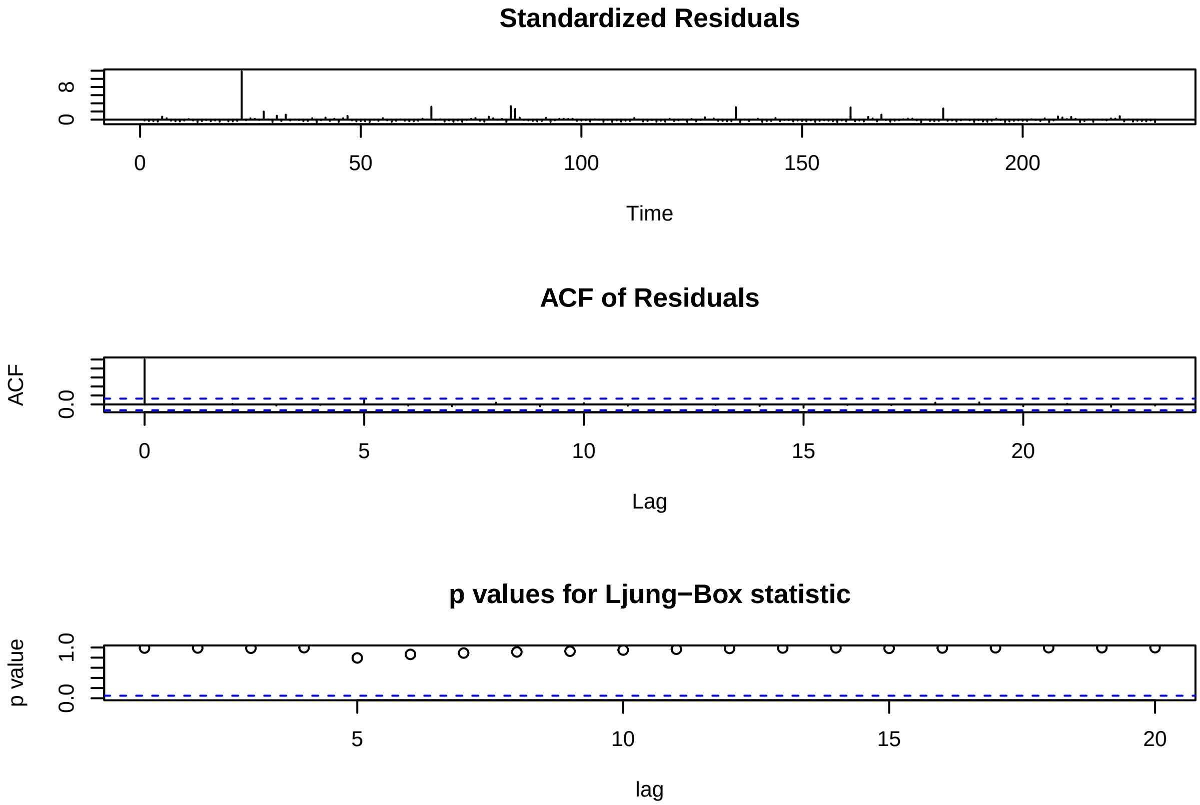
The auto.arima() function above supports our cliam for the best SARIMA model for the Winter Freeze Time Series.

Disaster Group Severe Storm

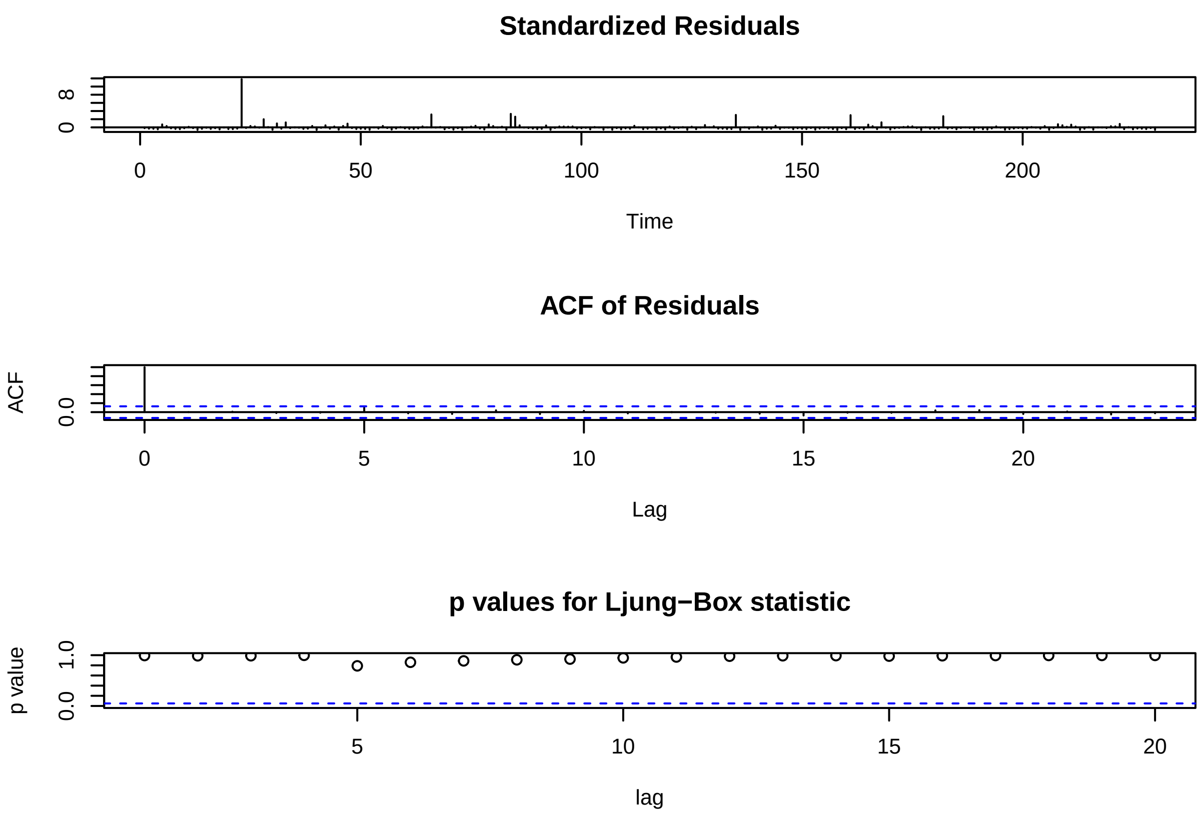
```
ss_fit1=sarima(SS_resi_final,1,0,1)
```




```
ss_fit2=sarima(SS_resi_final,2,0,1)
```

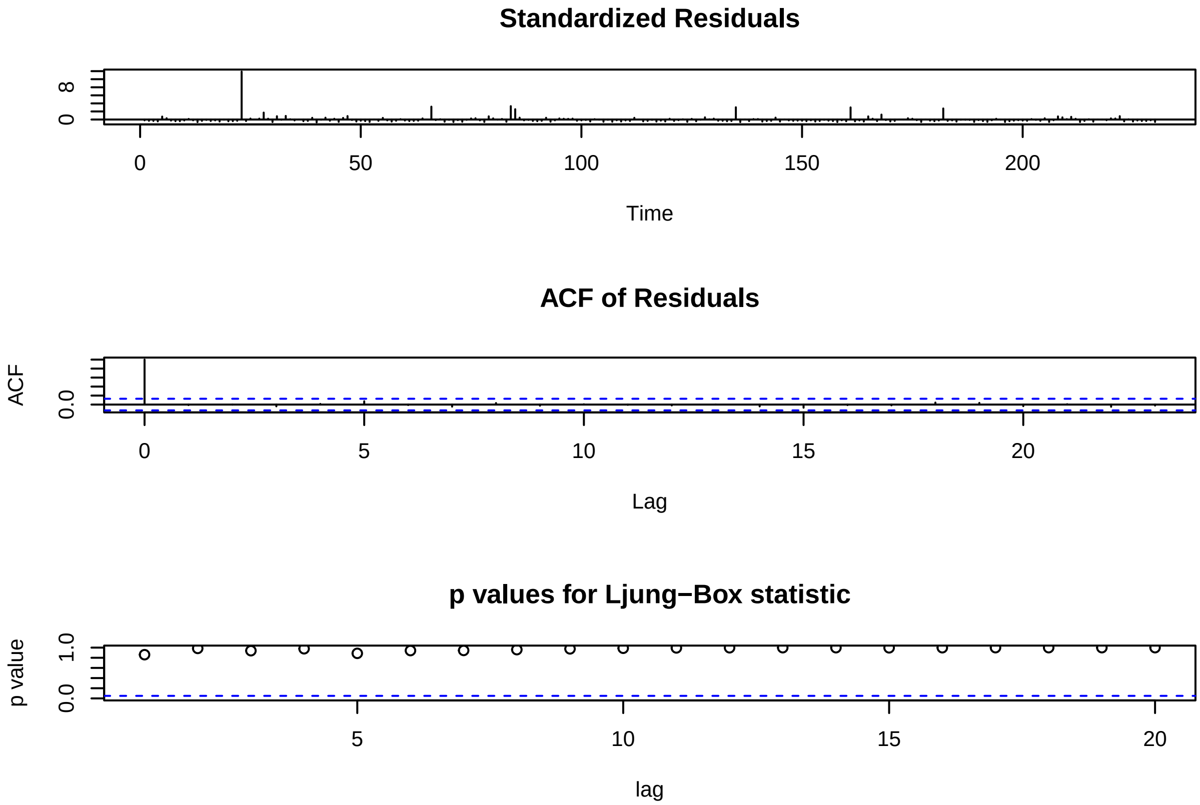



```
ss_fit3=sarima(SS_resi_final,1,0,2)
```

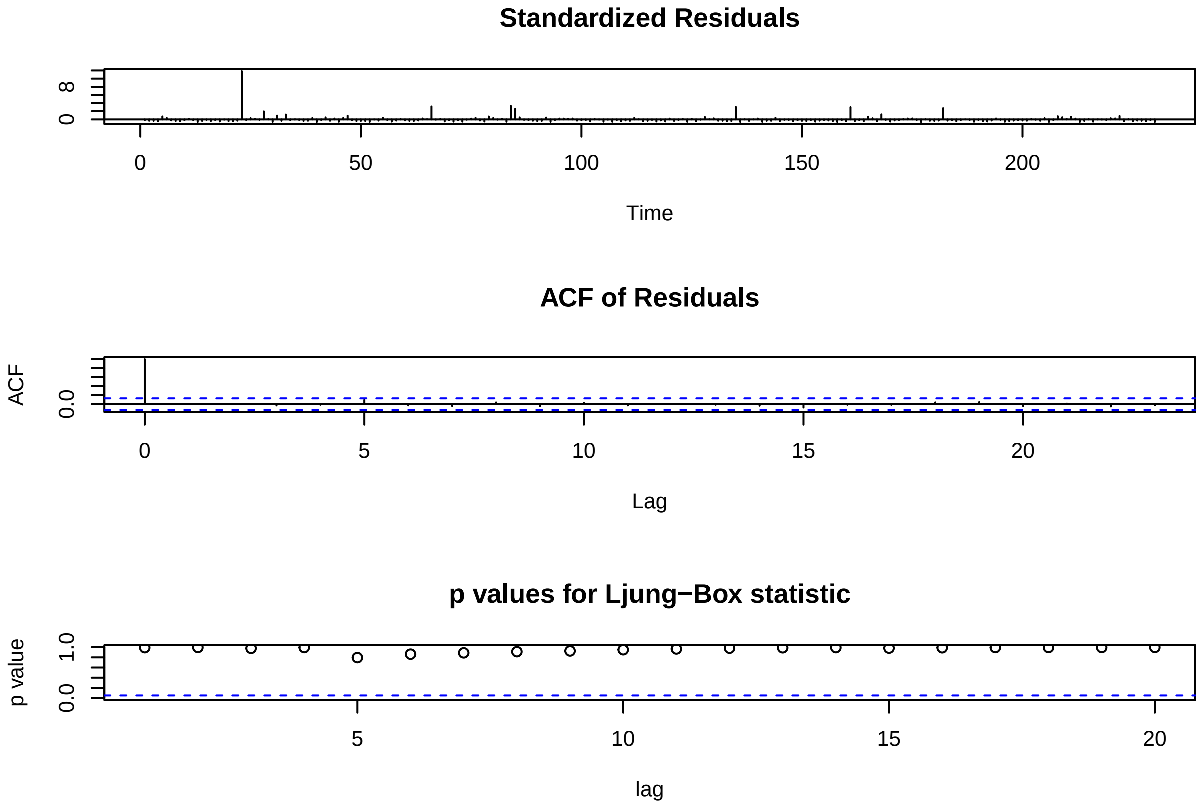


```
ss_fit4=sarima(SS_resi_final,2,0,2)
```

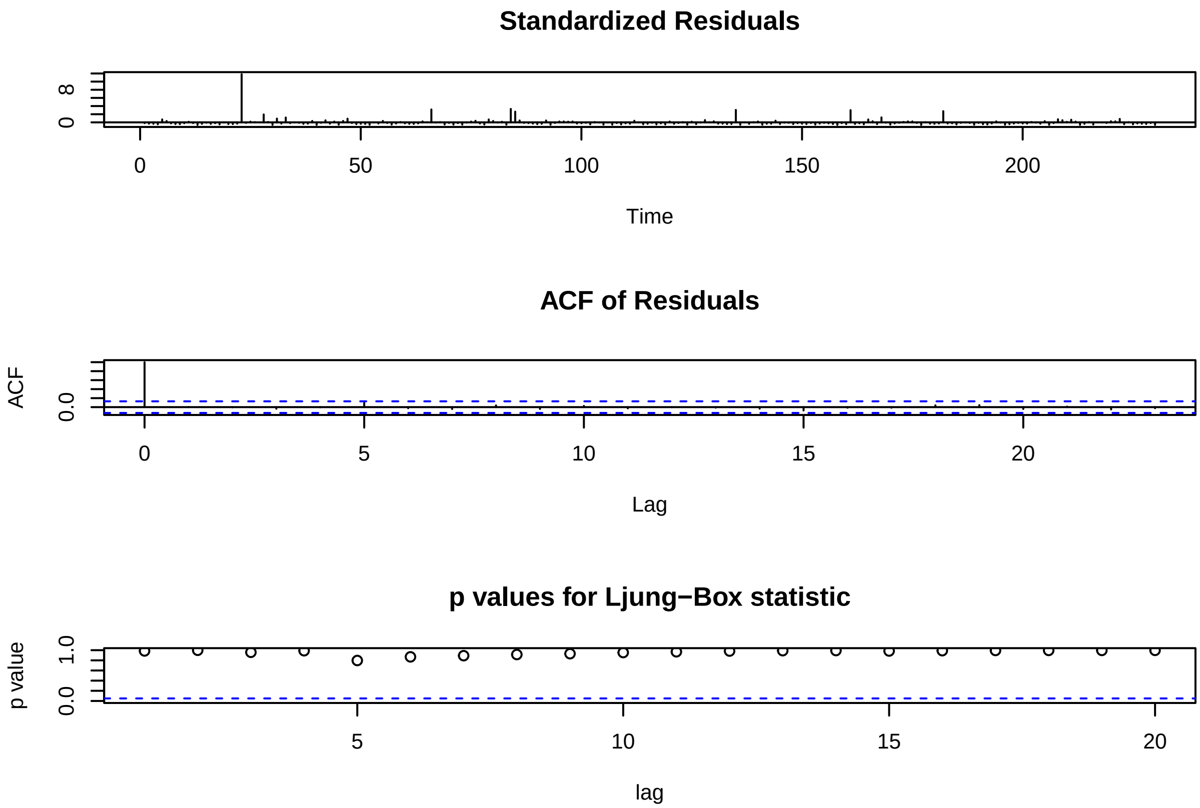
```
## Warning in arima(data, order = c(p, d, q), seasonal = list(order = c(P, :
## possible convergence problem: optim gave code = 1
```



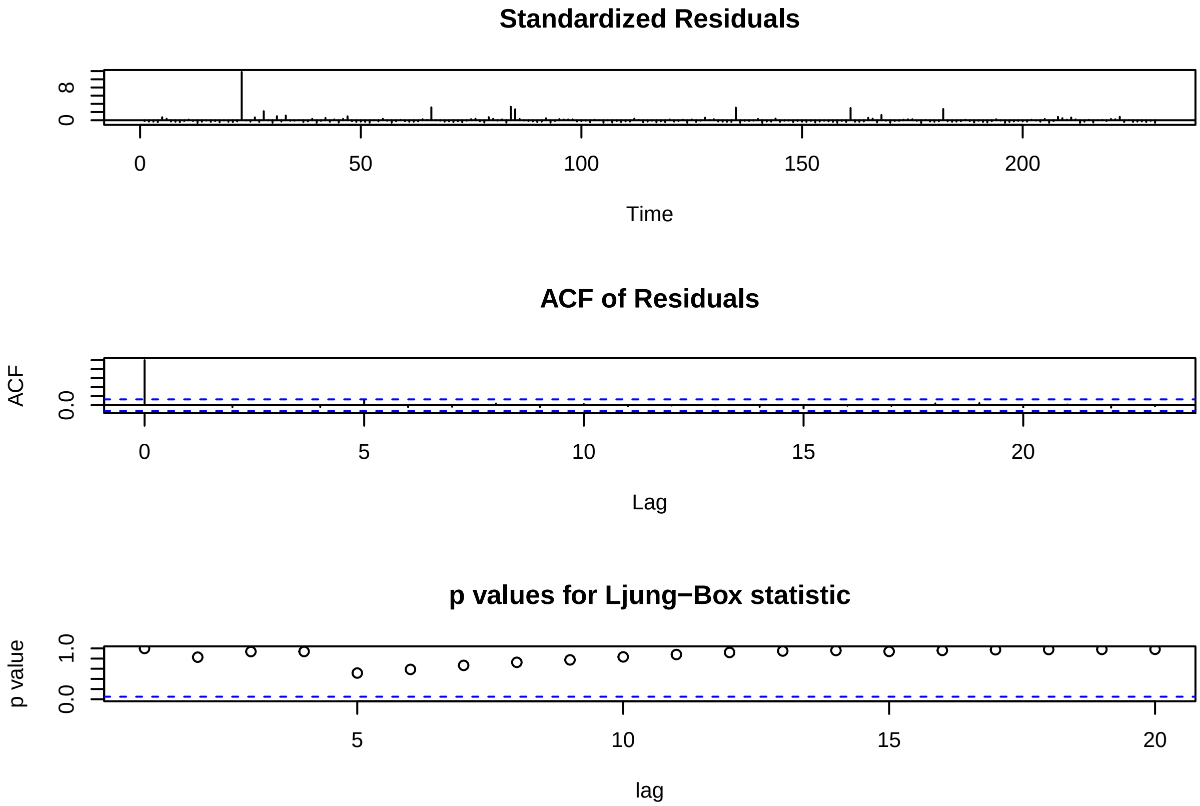

```
ss_fit5=sarima(SS_resi_final,3,0,2)
```



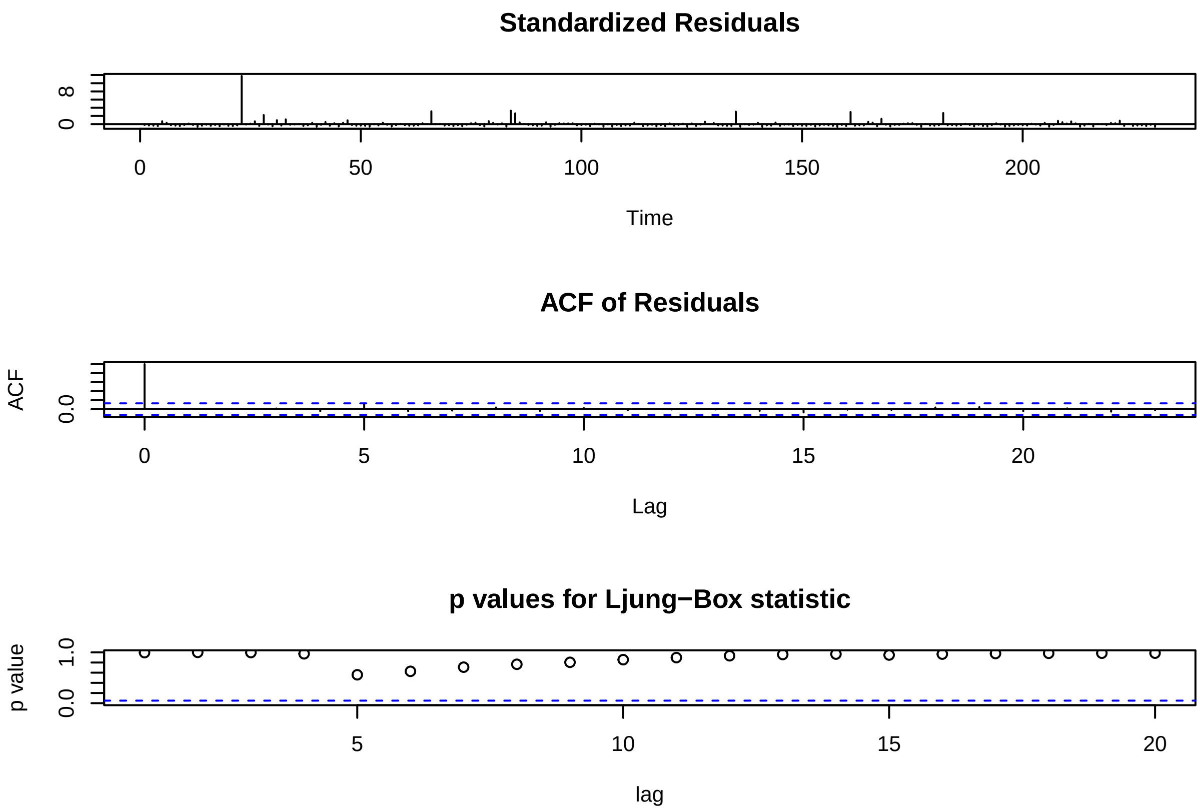

```r
ss_fit6=sarima(SS_resi_final,3,0,1)
```



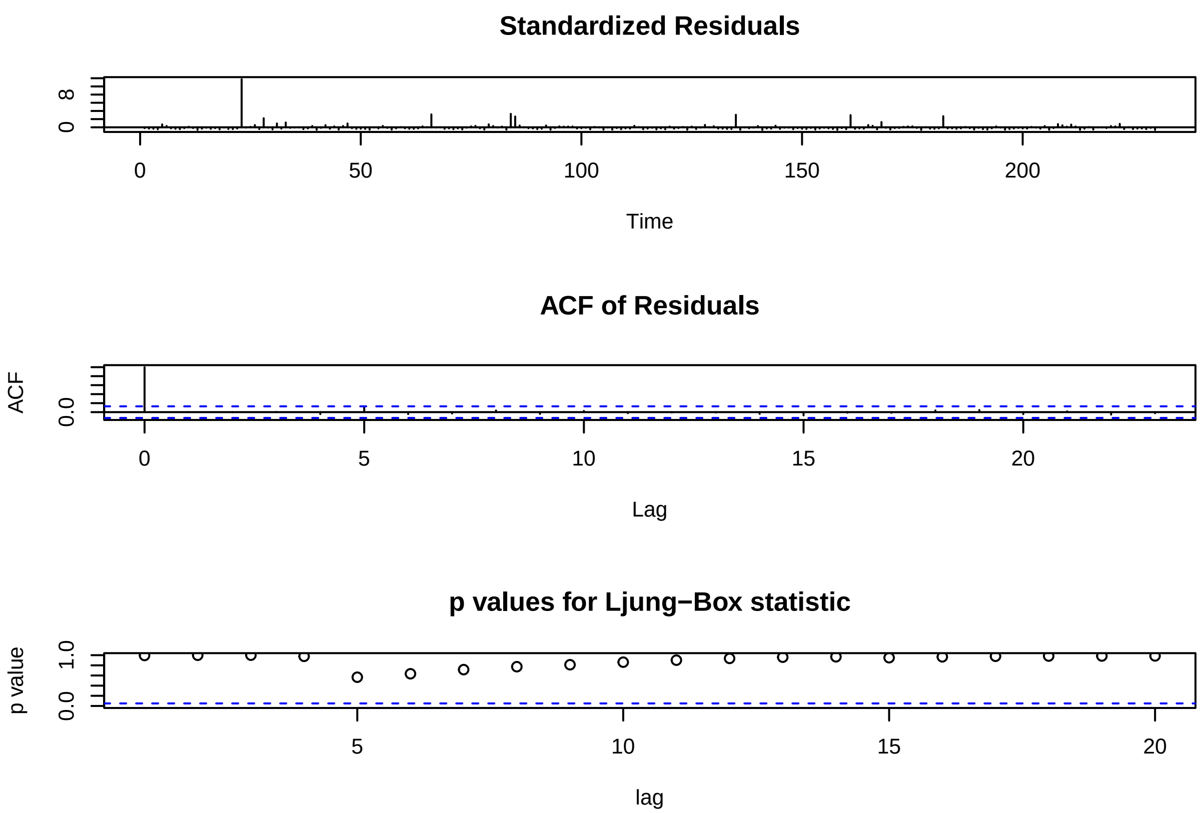
```
ss_fit7=sarima(SS_resi_final,1,0,0)
```




```
ss_fit8=sarima(SS_resi_final,2,0,0)
```




```
ss_fit9=sarima(SS_resi_final,3,0,0)
```



We can also loop through the possible models ARMA(p,q) up to lag 3. That is, we check an ARMA(1,0),

ARMA(2,0), … , ARMA(1,1), ARMA(1,2), ARMA(3,1) and so on. . . .

```
#This little function extracts the
#AIC, AICc and BIC values from an Arima() fit
getAIC <- function(fit) {
  c(fit$AIC, fit$AICc, fit$BIC)
}
```

We will summarize the AIC-related results in a table and siplay the table

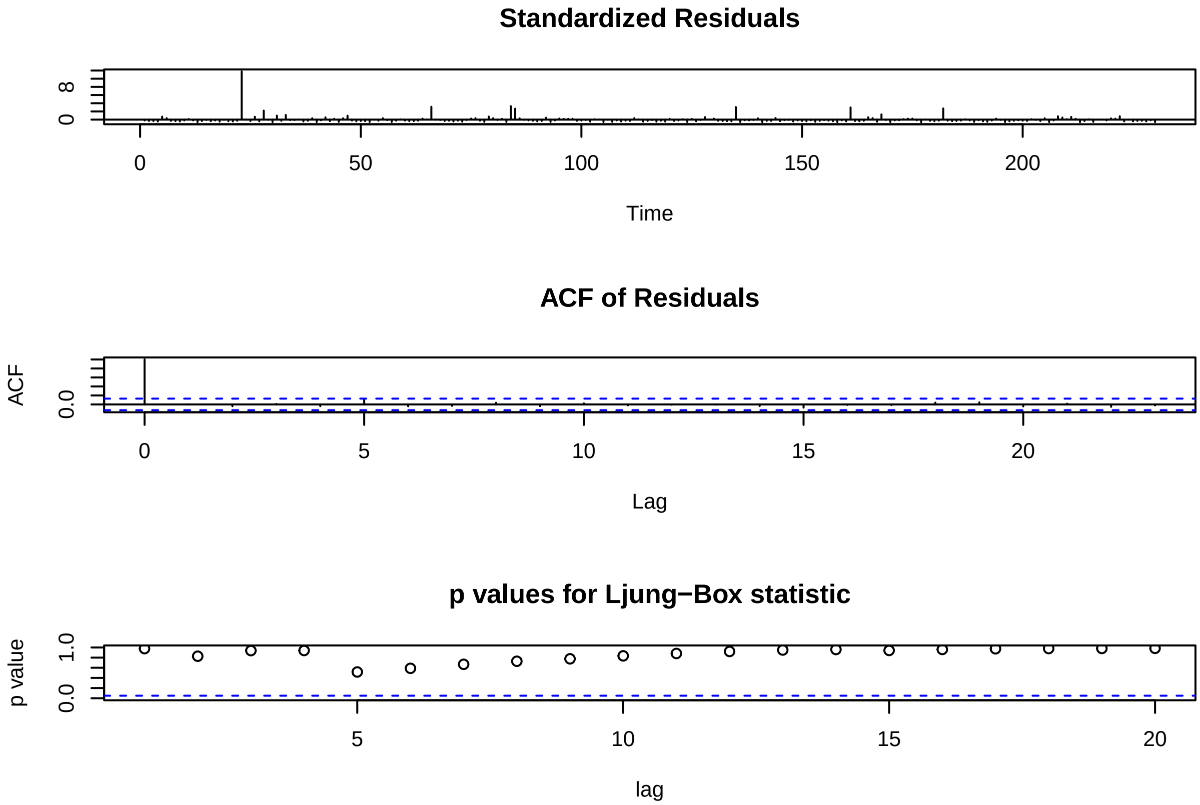
```
tab <- rbind(getAIC(ss_fit1), getAIC(ss_fit2), getAIC(ss_fit3),
             getAIC(ss_fit4), getAIC(ss_fit5), getAIC(ss_fit6),
            getAIC(ss_fit7), getAIC(ss_fit8), getAIC(ss_fit9))
colnames(tab) <- c("AIC", "AICc", "BIC")
rownames(tab) <- c("ARMA(1,1)","ARMA(2,1)","ARMA(1,2)","ARMA(2,2)",
 "ARMA(3,2)","ARMA(3,1)","AR(1)", "AR(2)", "AR(3)")
kable(tab)  #displays the table
```

	AIC	AICc	BIC
ARMA(1,1)	17.36580	17.37527	16.41065
ARMA(2,1)	17.36816	17.37802	16.42795
ARMA(1,2)	17.36828	17.37814	16.42807
ARMA(2,2)	17.36954	17.37987	16.44428
ARMA(3,2)	17.38551	17.39640	16.47520
ARMA(3,1)	17.37645	17.38678	16.45119
AR(1)	17.35711	17.36627	16.38701
AR(2)	17.36418	17.37365	16.40903
AR(3)	17.37274	17.38260	16.43254

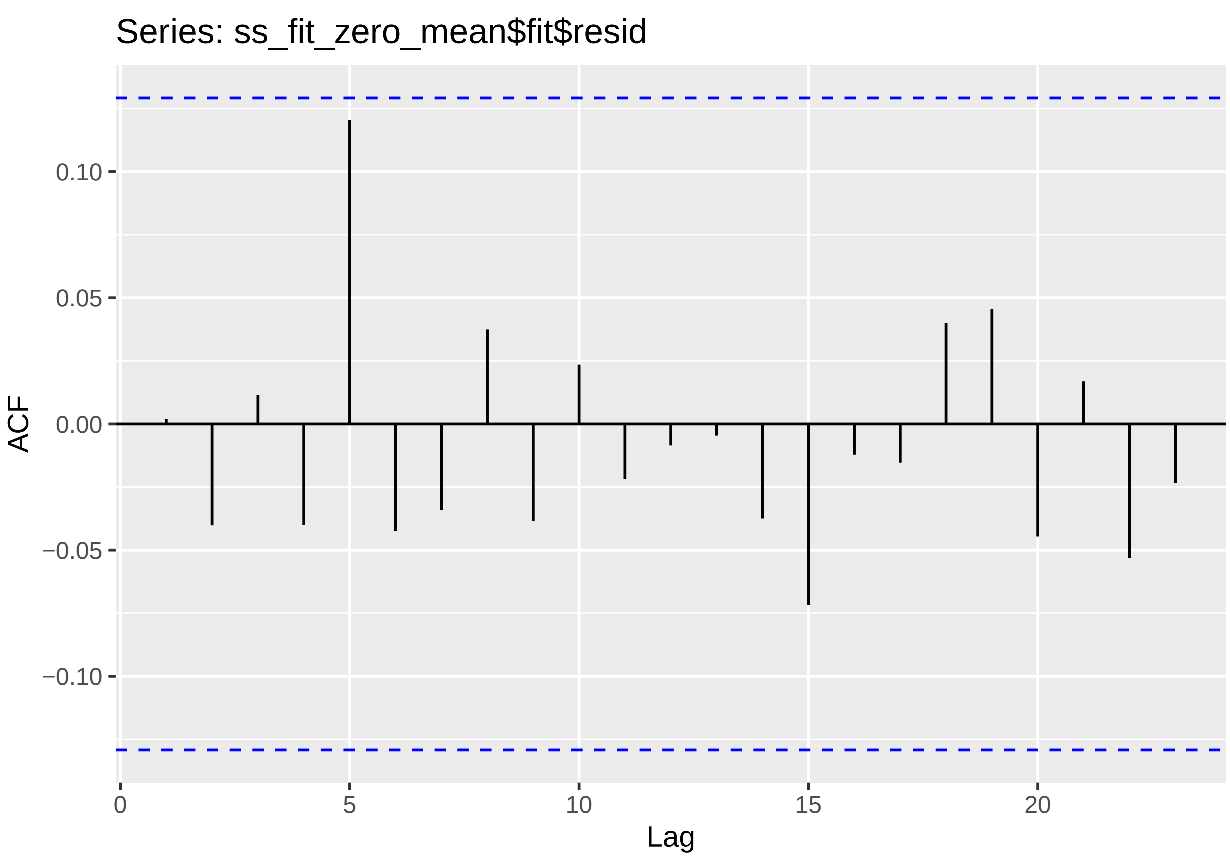
From the table, using a combination of the AIC, AICc, and BIC values reported (remember when the values are within 2 of one another, they are essentially the same). We see that all of these models are the same. We want to choose the simplest model.

For now, let us consider the $AR(1)$ model for simplicity.

```
ss_fit_zero_mean=sarima(SS_resi_final,0,0,0)
```




```r
ggAcf(ss_fit_zero_mean$fit$resid)
```



```r
finalfit <- Arima(ts_SS, order=c(0, 0, 0))
finalfit
```

```
## Series: ts_SS
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##            mean
##       2831.0648
## s.e.   235.0122
##
## sigma^2 = 12758531:  log likelihood = -2207.45
## AIC=4418.9   AICc=4418.96   BIC=4425.78
```

Our best model comes out at as a non-zero mean ARIMA(0,0,0), this is an indication of a constant mean. Our model equation being

$$X_t = 2831.0648$$

To validate our model along with check for seasonality we will call the auto.arima() function.

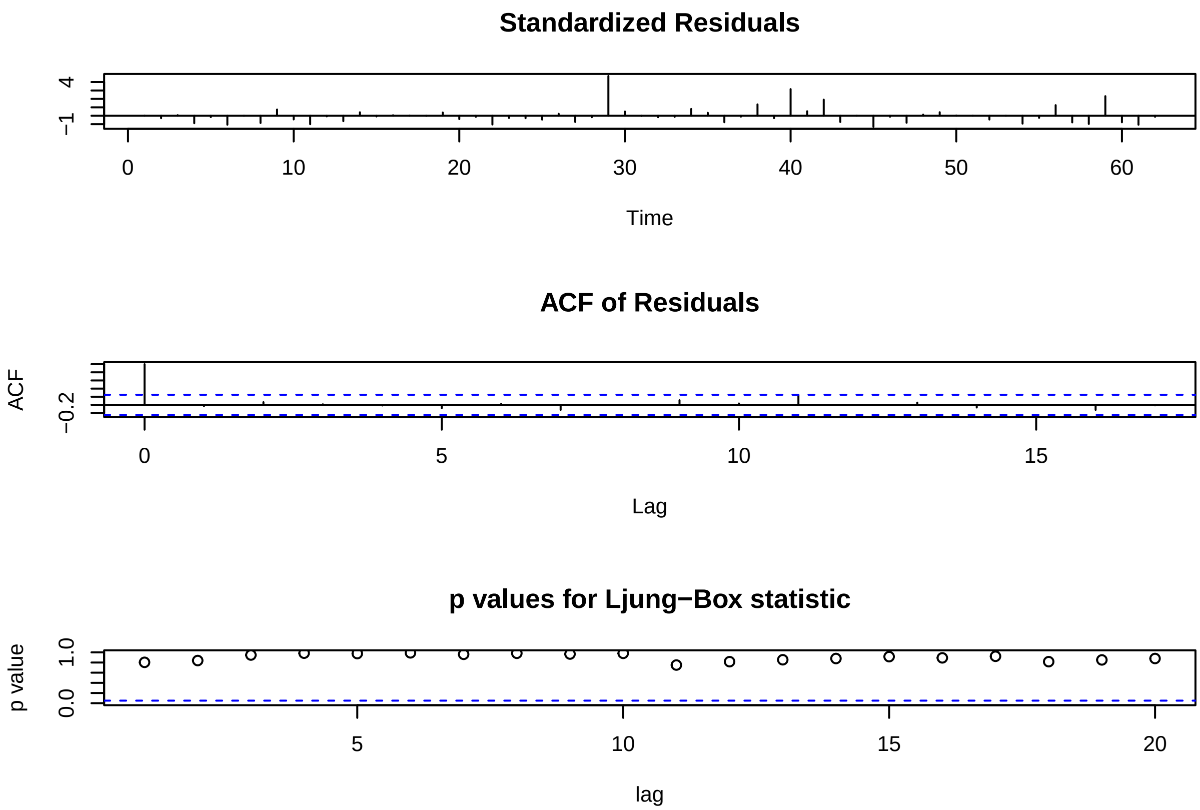
```r
auto.arima(ts_SS)
```

```
## Series: ts_SS
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##            mean
##       2831.0648
## s.e.   235.0122
##
## sigma^2 = 12758531:  log likelihood = -2207.45
## AIC=4418.9   AICc=4418.96   BIC=4425.78
```

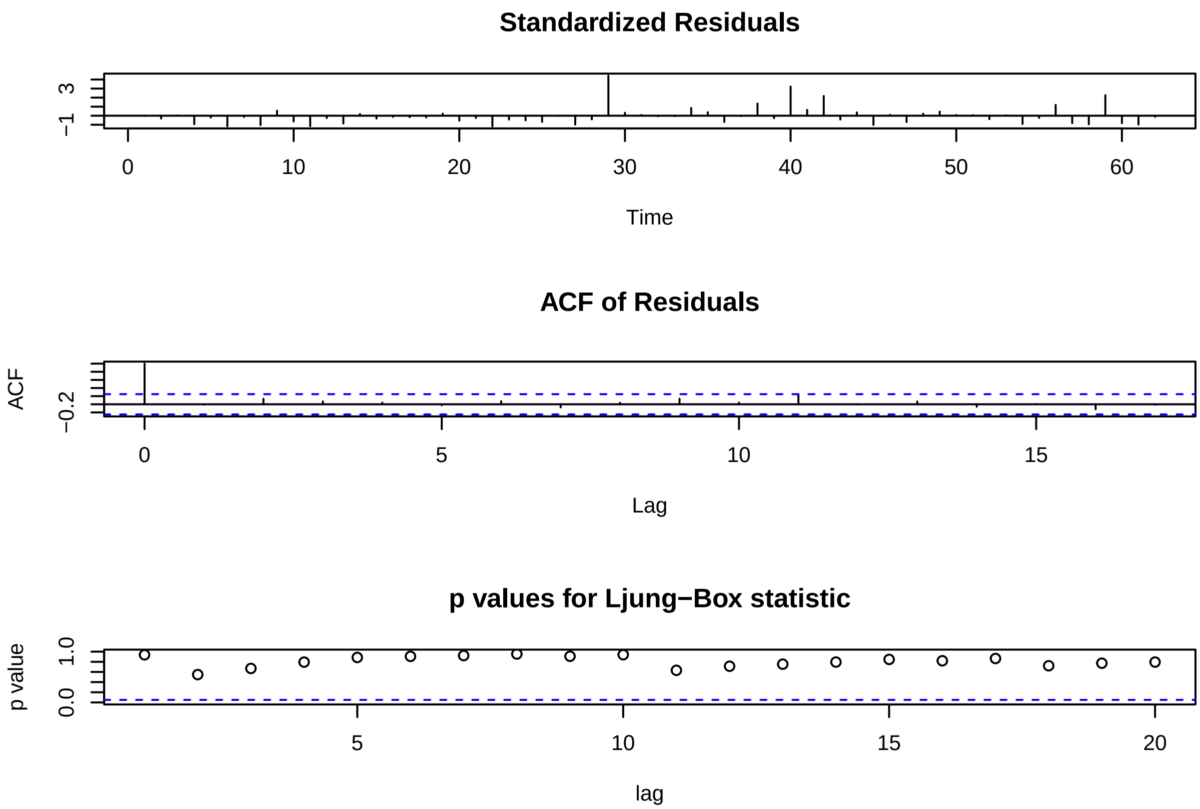
The auto.arima() function directly above supports our claim of ARIMA(0,0,0) being the best model.

Disaster Group Cyclone

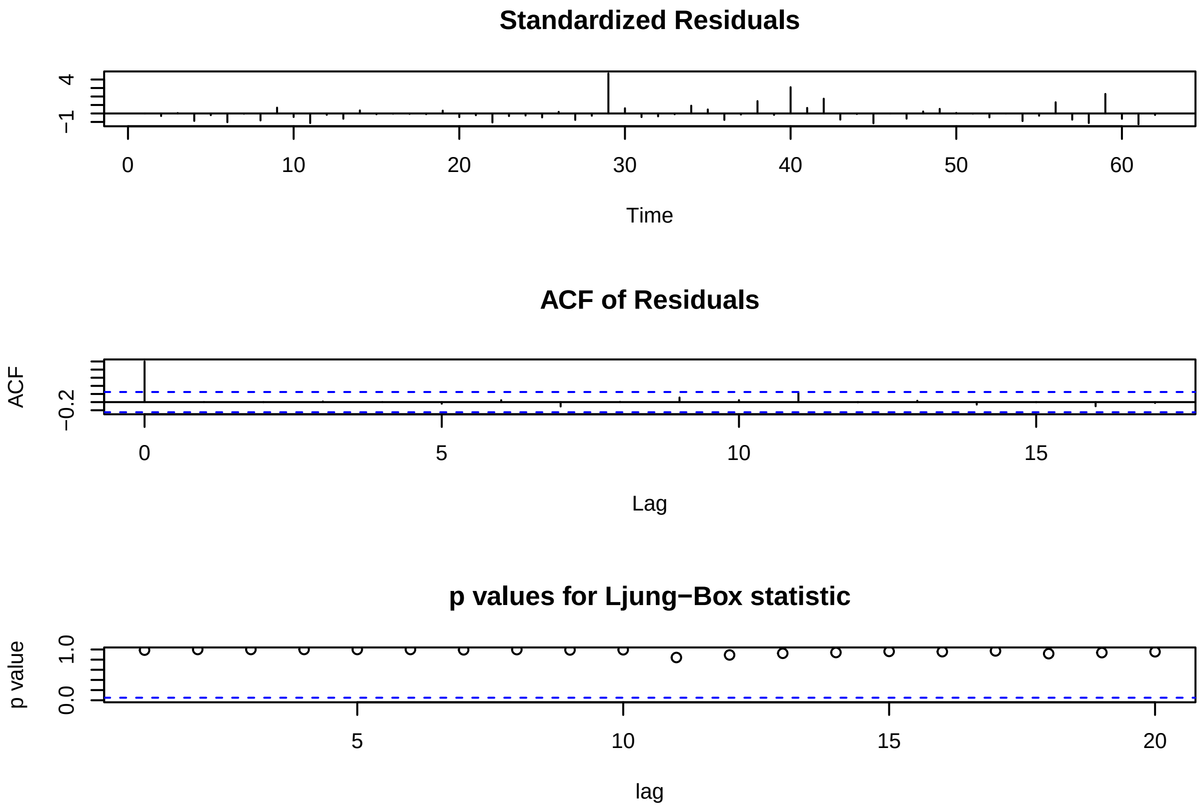
```r
cyc_fit1=sarima(Cyc_resi_final,1,0,1)
```



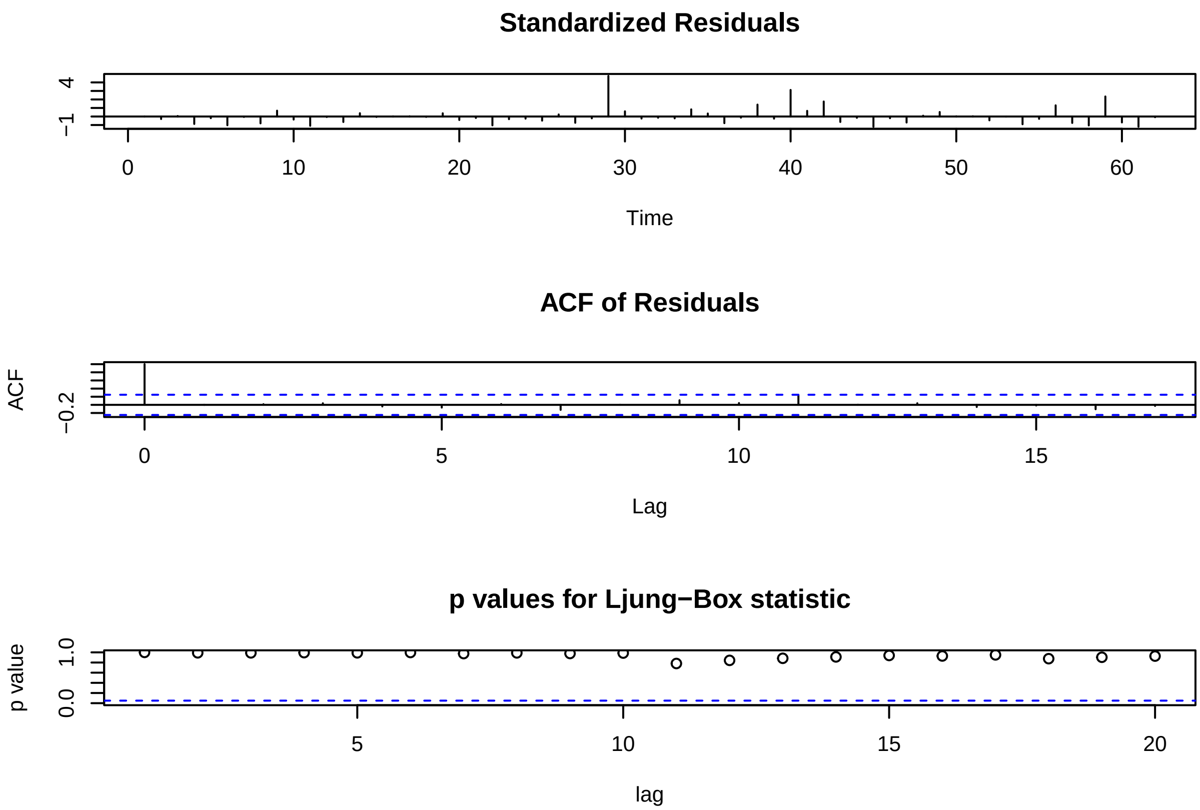
```
cyc_fit2=sarima(Cyc_resi_final,2,0,1)
```



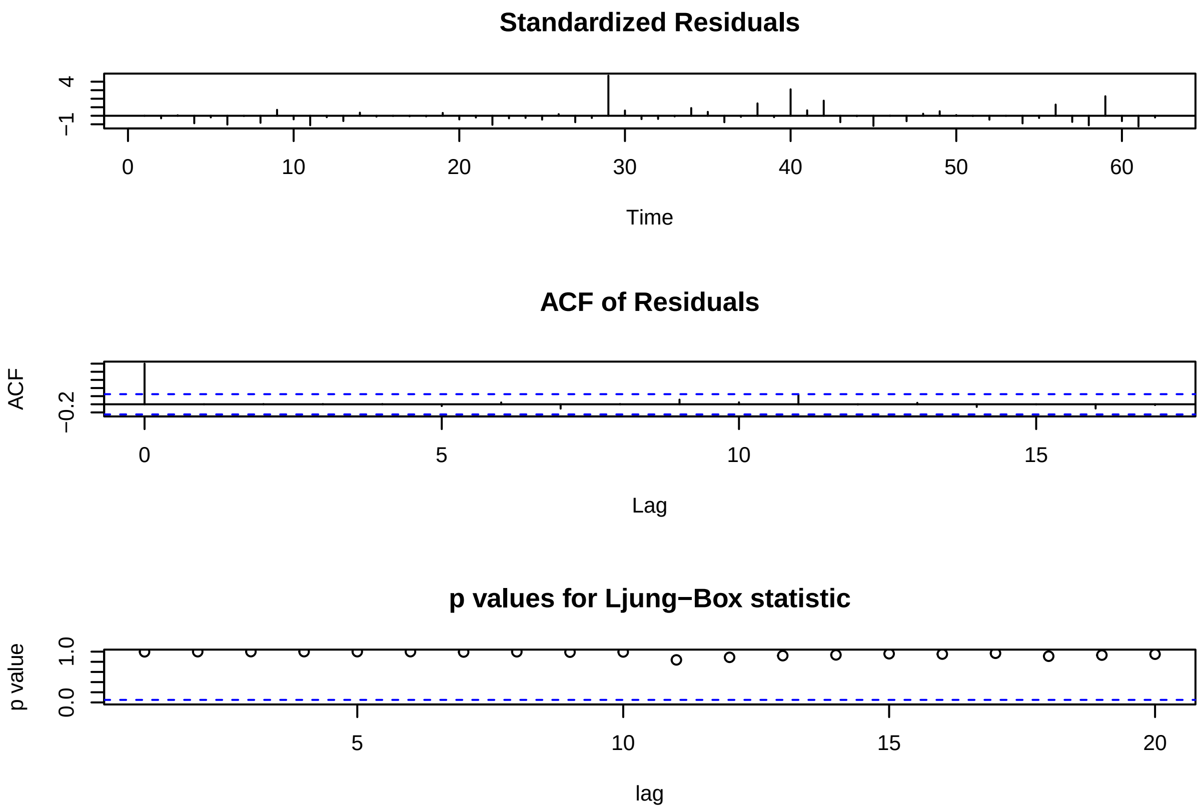
```r
cyc_fit3=sarima(Cyc_resi_final,1,0,2)
```



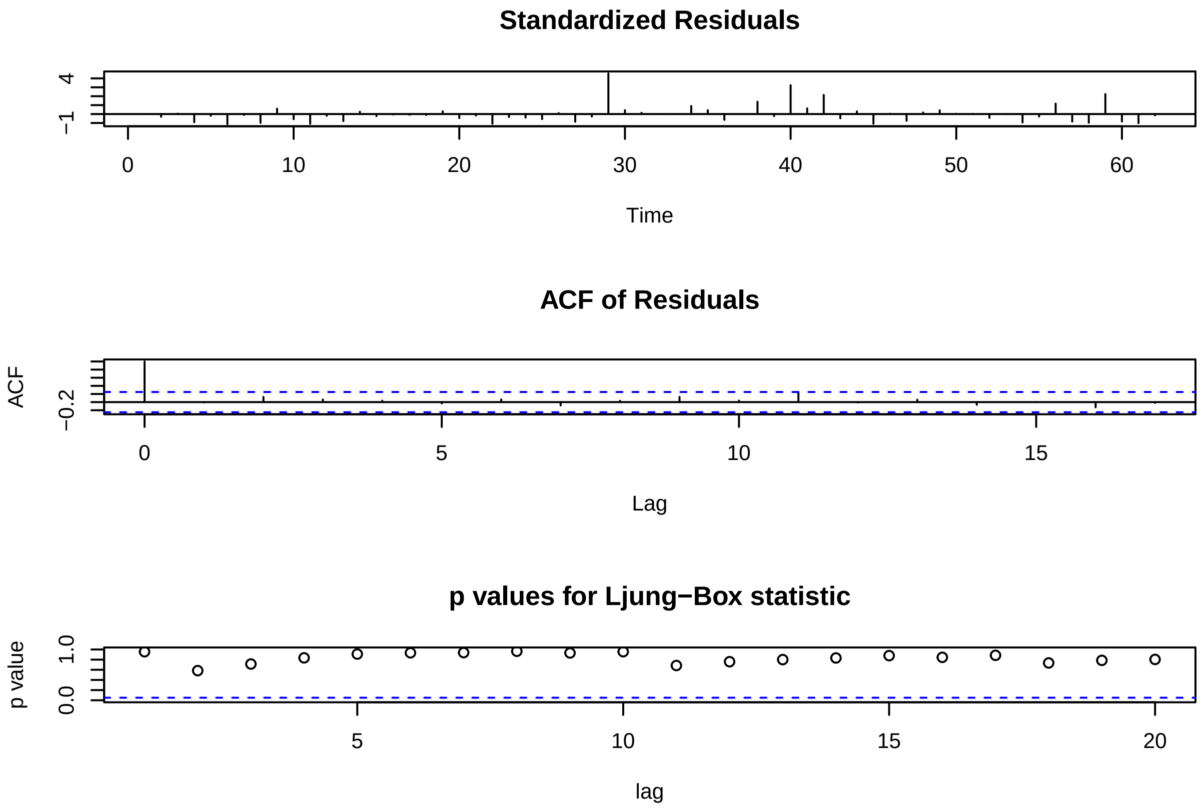

```r
cyc_fit4=sarima(Cyc_resi_final,2,0,2)
```



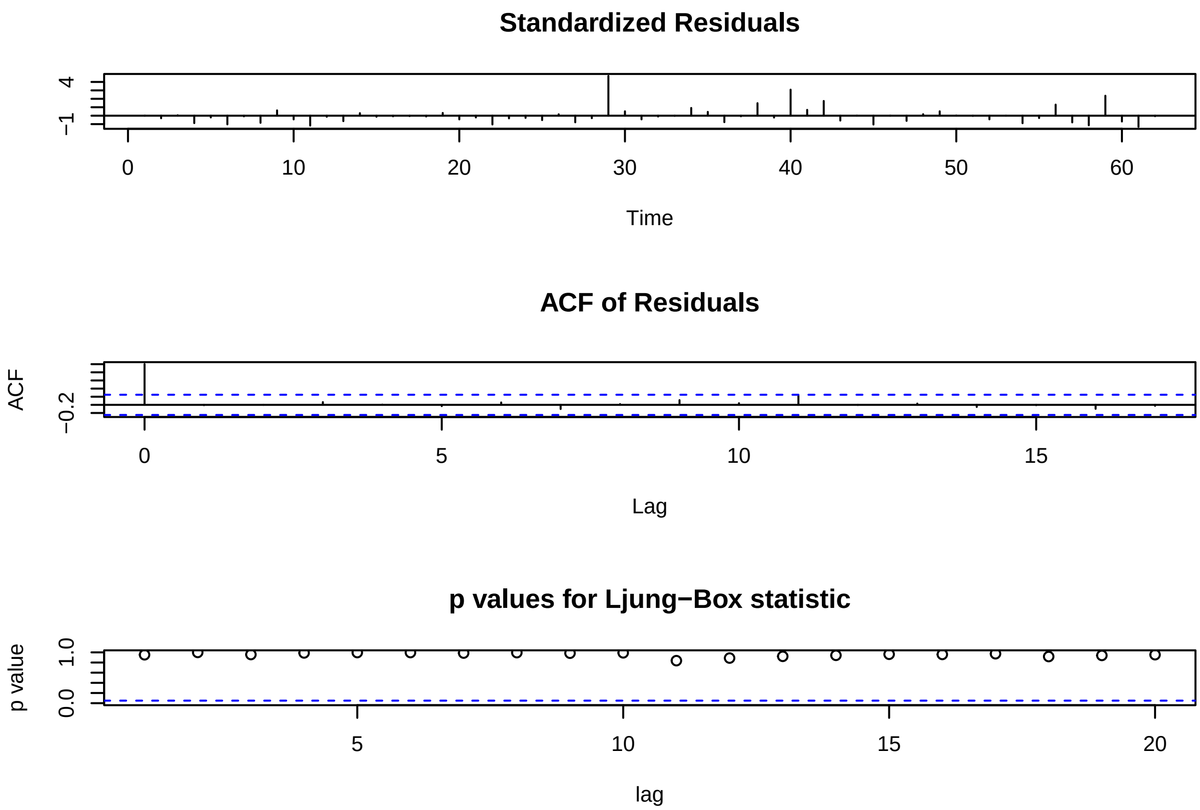

```
cyc_fit6=sarima(Cyc_resi_final,3,0,1)
```



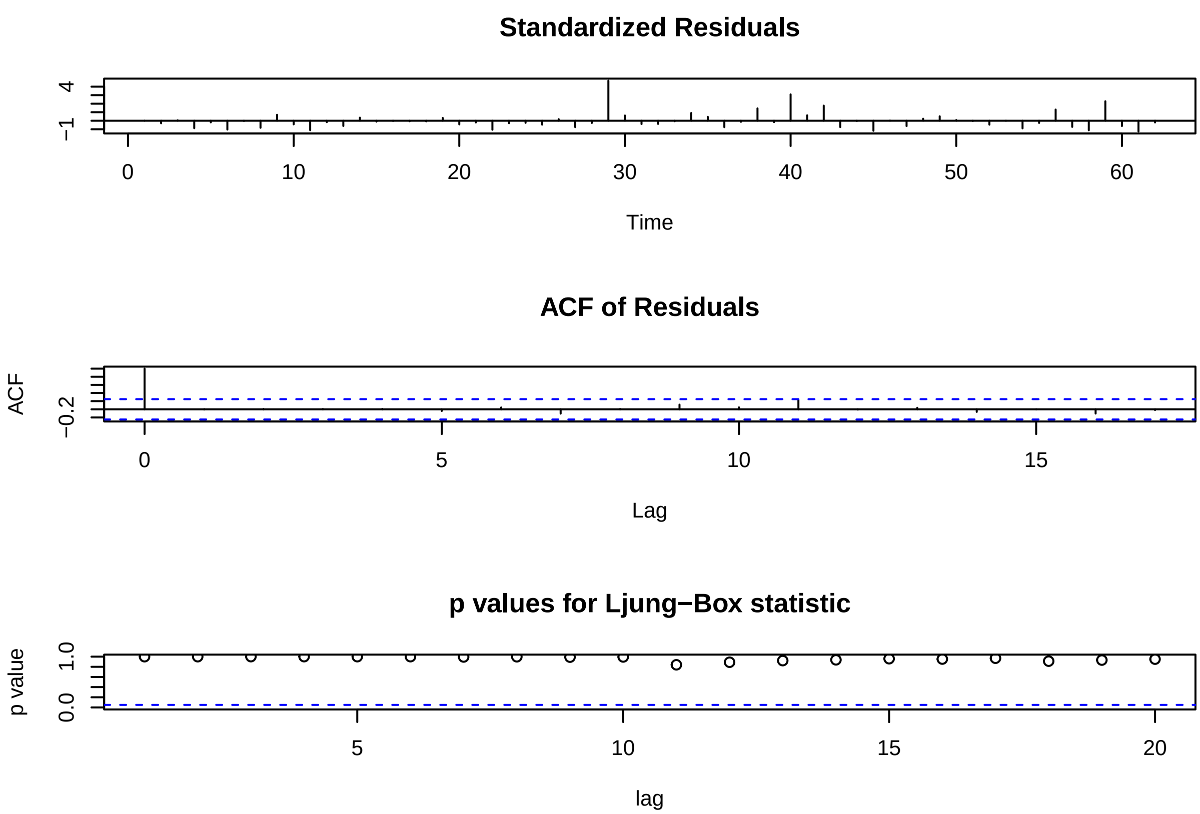
```
cyc_fit7=sarima(Cyc_resi_final,1,0,0)
```




```
cyc_fit8=sarima(Cyc_resi_final,2,0,0)
```




```
cyc_fit9=sarima(Cyc_resi_final,3,0,0)
```



We can also loop through the possible models ARMA(p,q) up to lag 3. That is, we check an ARMA(1,0),

ARMA(2,0), . . . , ARMA(1,1), ARMA(1,2), ARMA(3,1) and so on. . . .

```
#This little function extracts the
#AIC, AICc and BIC values from an Arima() fit
getAIC <- function(fit) {
  c(fit$AIC, fit$AICc, fit$BIC)
}
```

We will summarize the AIC-related results in a table and siplay the table

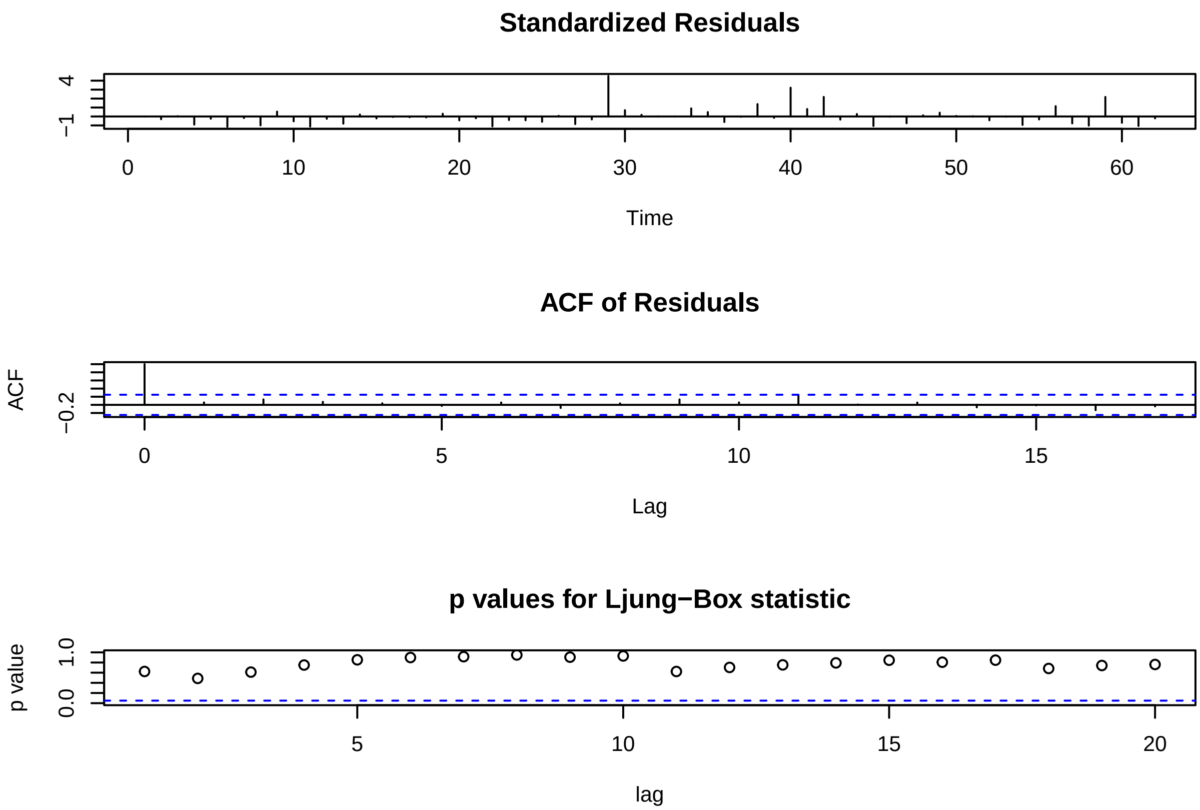
```
tab <- rbind(getAIC(cyc_fit1), getAIC(cyc_fit2), getAIC(cyc_fit3),
             getAIC(cyc_fit4), getAIC(cyc_fit6),
            getAIC(cyc_fit7), getAIC(cyc_fit8), getAIC(cyc_fit9))
colnames(tab) <- c("AIC", "AICc", "BIC")
rownames(tab) <- c("ARMA(1,1)","ARMA(2,1)","ARMA(1,2)","ARMA(2,2)","ARMA(3,1)","AR(1)", "AR(2)", "AR(3)
kable(tab)  #displays the table
```

	AIC	AICc	BIC
ARMA(1,1)	22.00729	22.05087	21.11022
ARMA(2,1)	22.04023	22.08977	21.17747
ARMA(1,2)	22.03391	22.08345	21.17114
ARMA(2,2)	22.06746	22.12435	21.23901
ARMA(3,1)	22.06624	22.12313	21.23778
AR(1)	21.99144	22.03037	21.06006
AR(2)	22.00590	22.04948	21.10882
AR(3)	22.03402	22.08356	21.17125

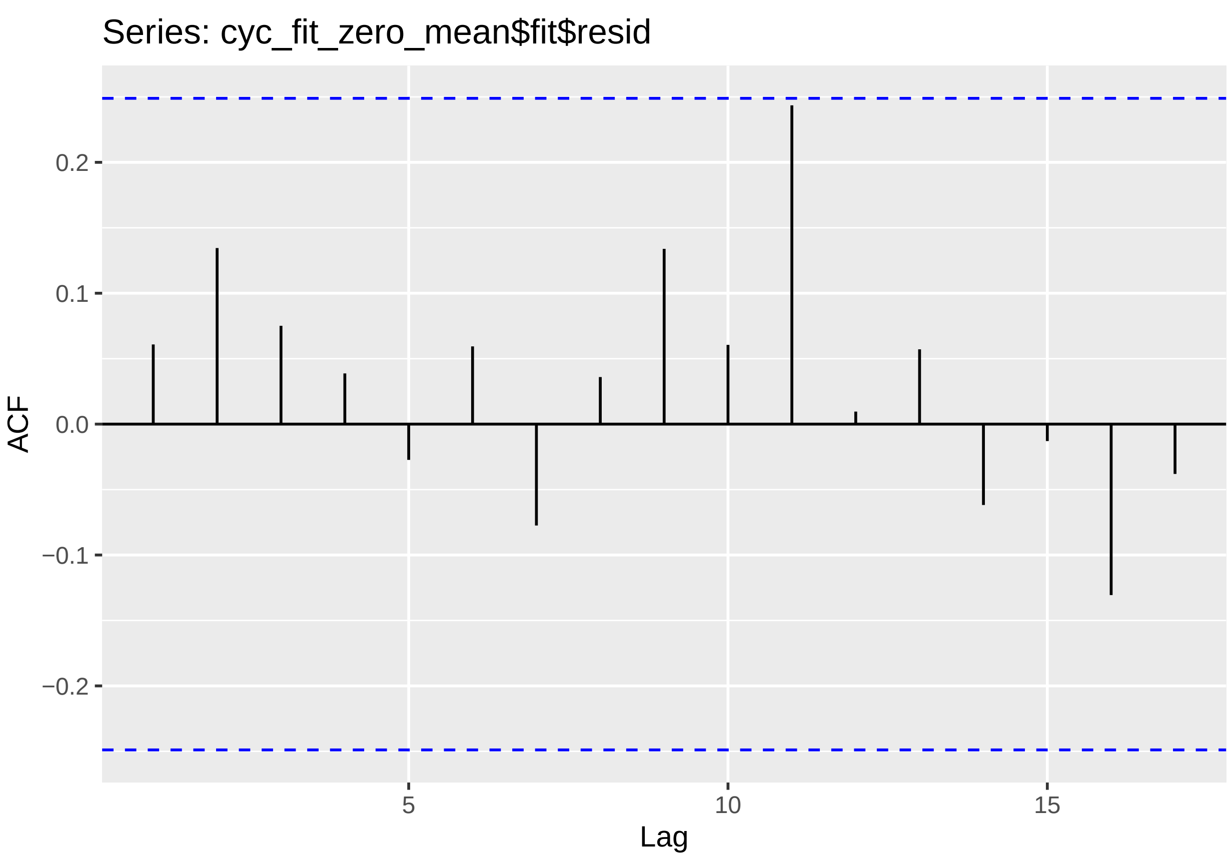
From the table, using a combination of the AIC, AICc, and BIC values reported (remember when the values are within 2 of one another, they are essentially the same). We see that all of these models are the same. We want to choose the simplest model.

For now, let us consider the $AR(1)$ model for simplicity.

```
cyc_fit_zero_mean=sarima(Cyc_resi_final,0,0,0)
```




```r
ggAcf(cyc_fit_zero_mean$fit$resid)
```



```r
finalfit <- Arima(ts_Cyc, order=c(0, 0, 0))
finalfit
```

```
## Series: ts_Cyc
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##            mean
##       22244.536
## s.e.   4822.009
##
## sigma^2 = 1.465e+09:  log likelihood = -741.73
## AIC=1487.47   AICc=1487.67   BIC=1491.72
```

Our best model comes out at as a non-zero mean ARIMA(0,0,0), this is an indication of a constant mean. Our model equation being

$$X_t = 22244.536$$

To validate our model along with check for seasonality we will call the auto.arima() function.

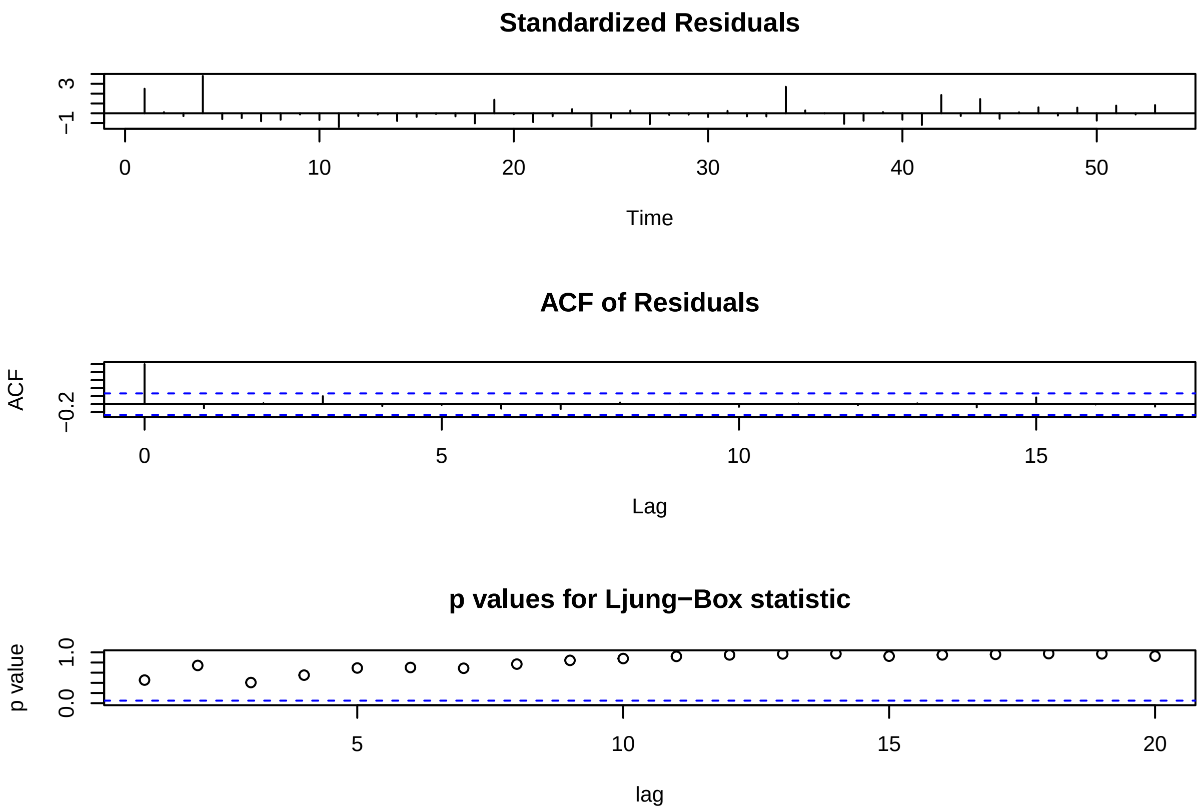
```r
auto.arima(ts_Cyc)
```

```
## Series: ts_Cyc
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##            mean
##       22244.536
## s.e.   4822.009
##
## sigma^2 = 1.465e+09:  log likelihood = -741.73
## AIC=1487.47   AICc=1487.67   BIC=1491.72
```

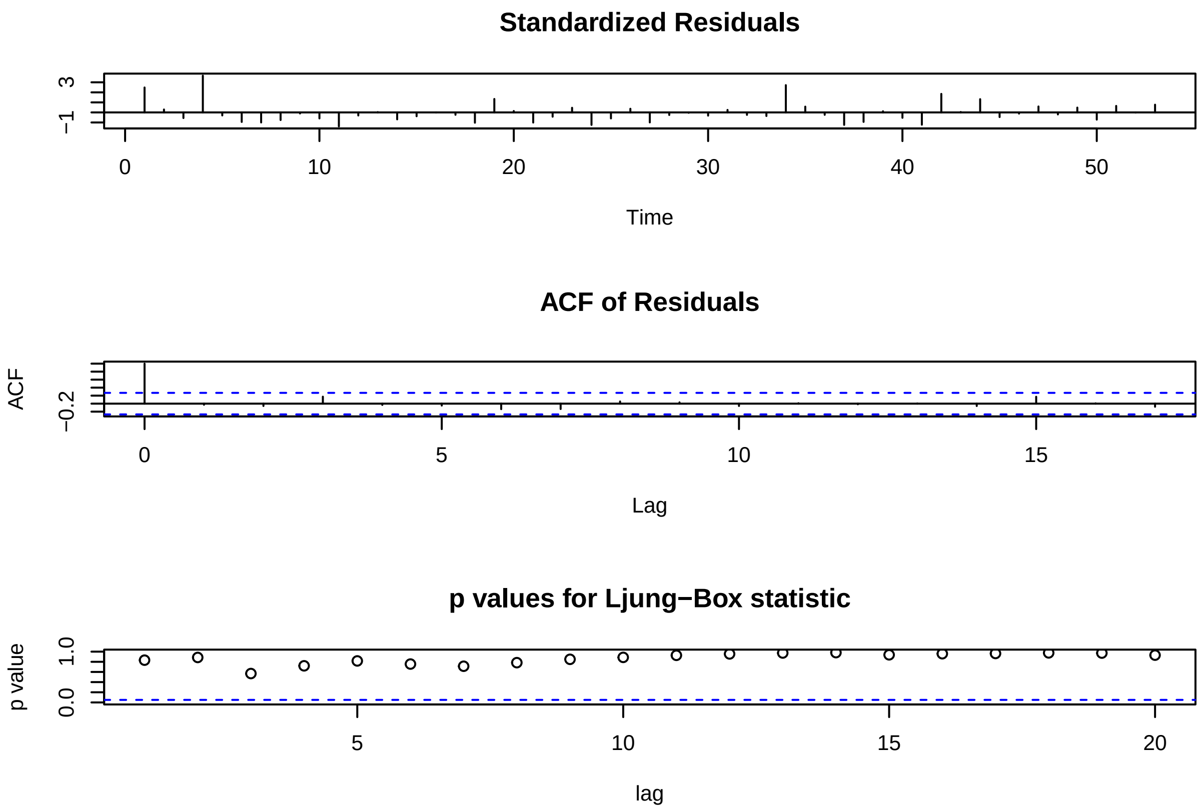
The auto.arima() function above backs up our claim of the best model being a constant model.

Disaster Group WF/Drought

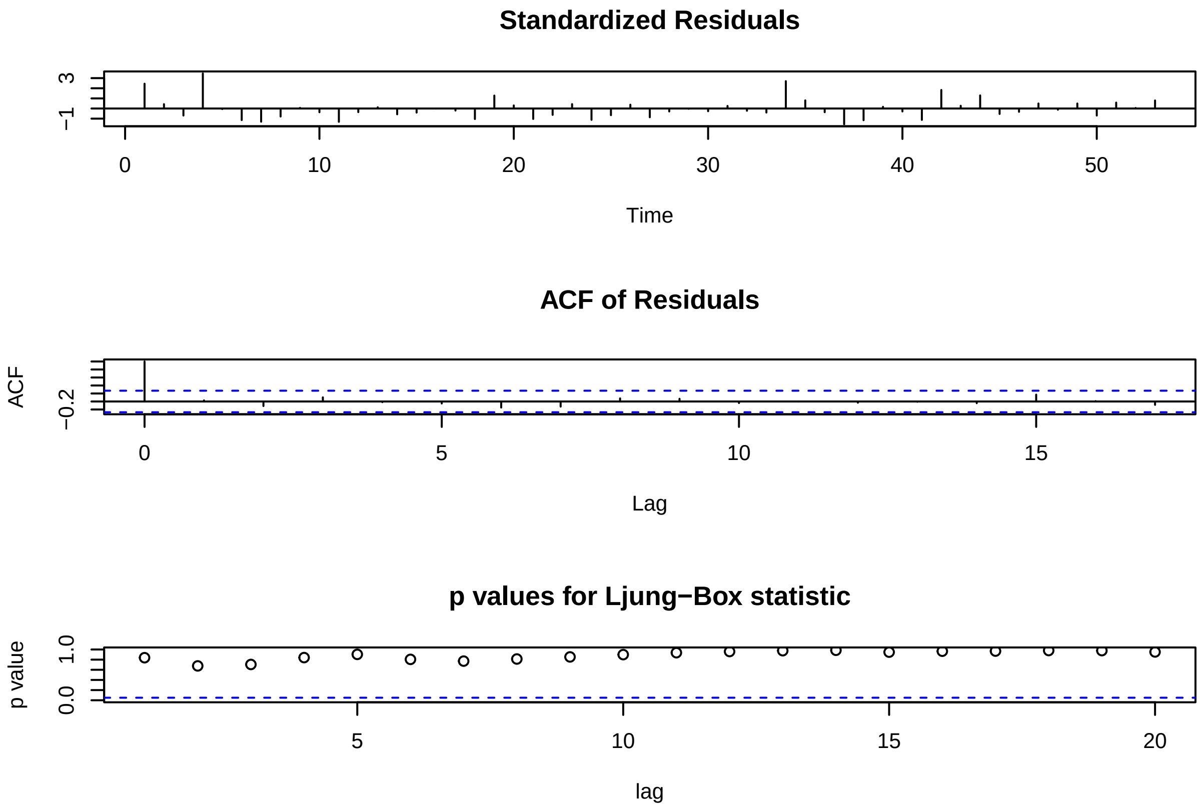
```r
drought_fit1=sarima(drought_resi_final,1,0,1)
```



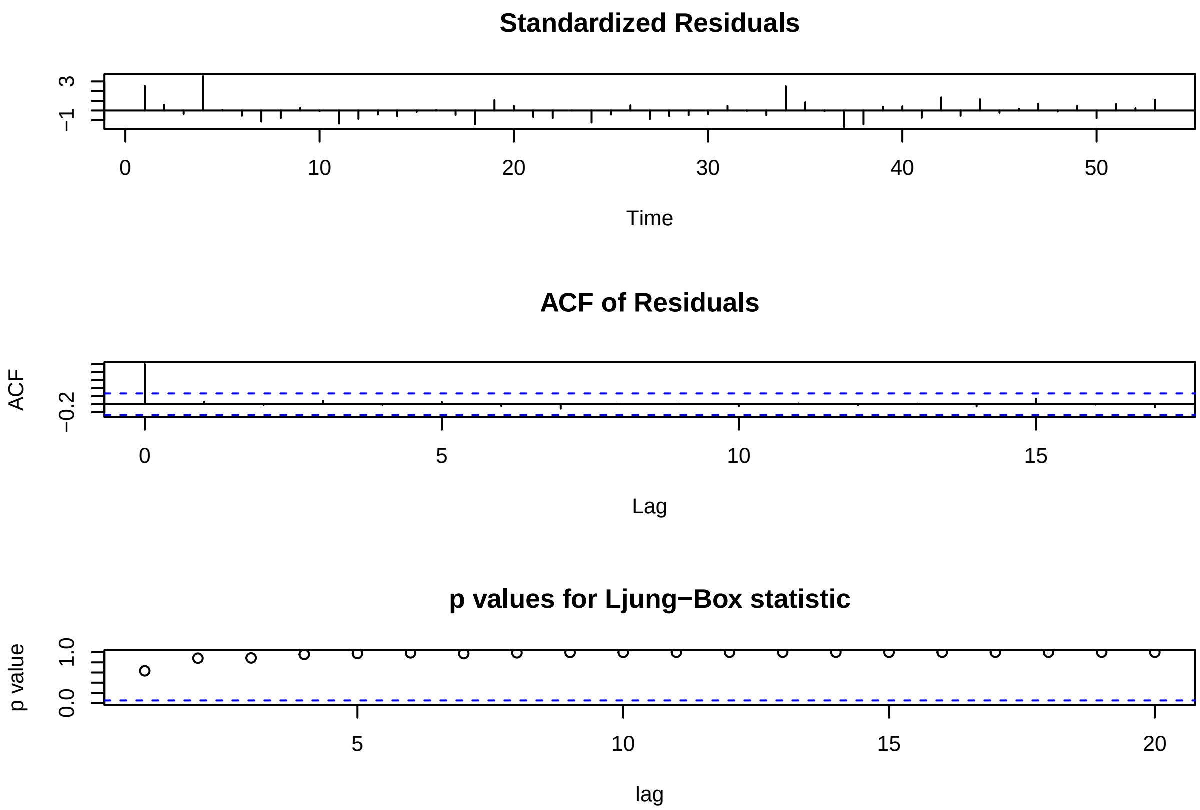

```r
drought_fit2=sarima(drought_resi_final,2,0,1)
```



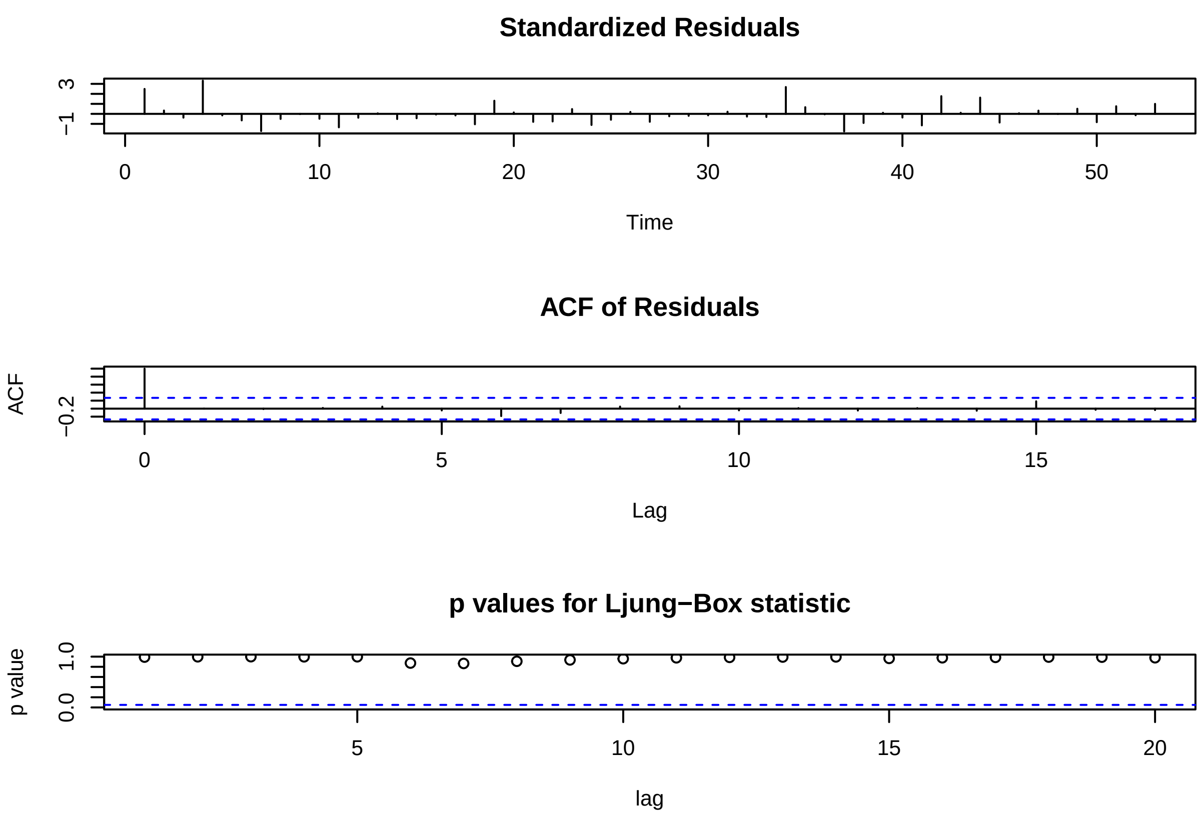
```r
drought_fit3=sarima(drought_resi_final,1,0,2)
```



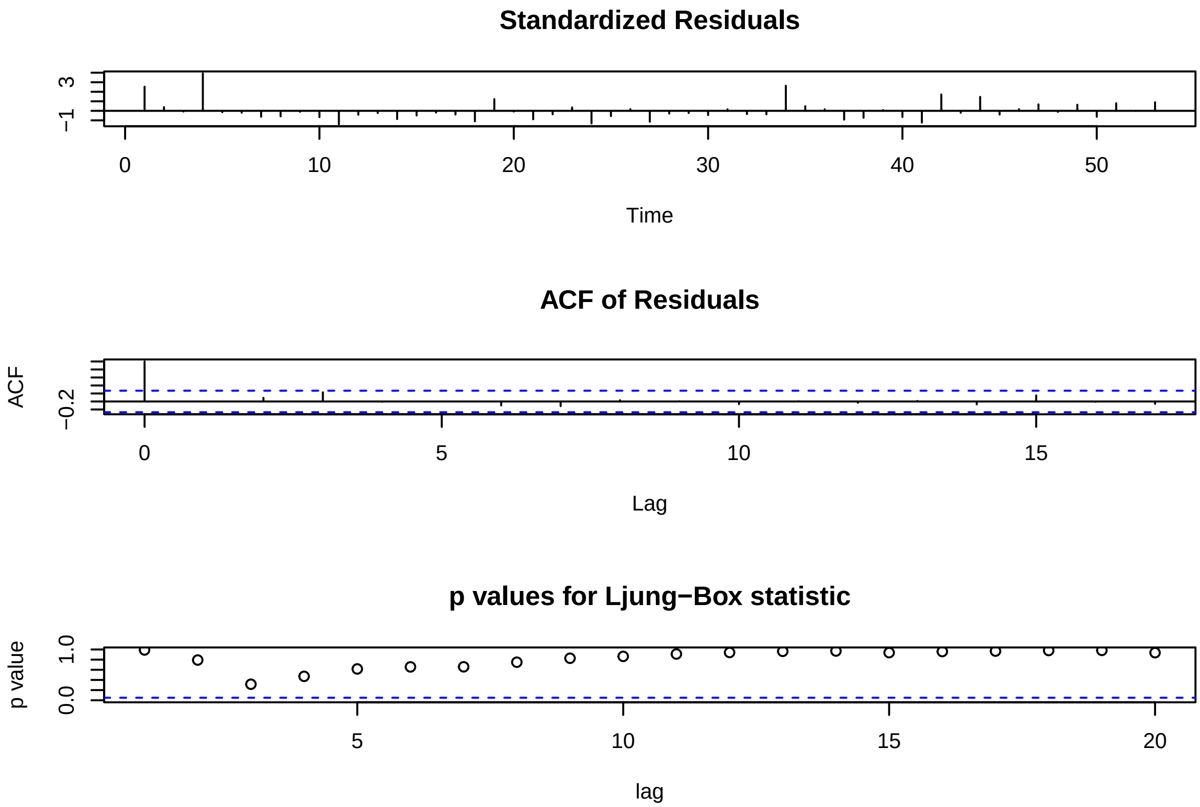

```r
drought_fit4=sarima(drought_resi_final,2,0,2)
```



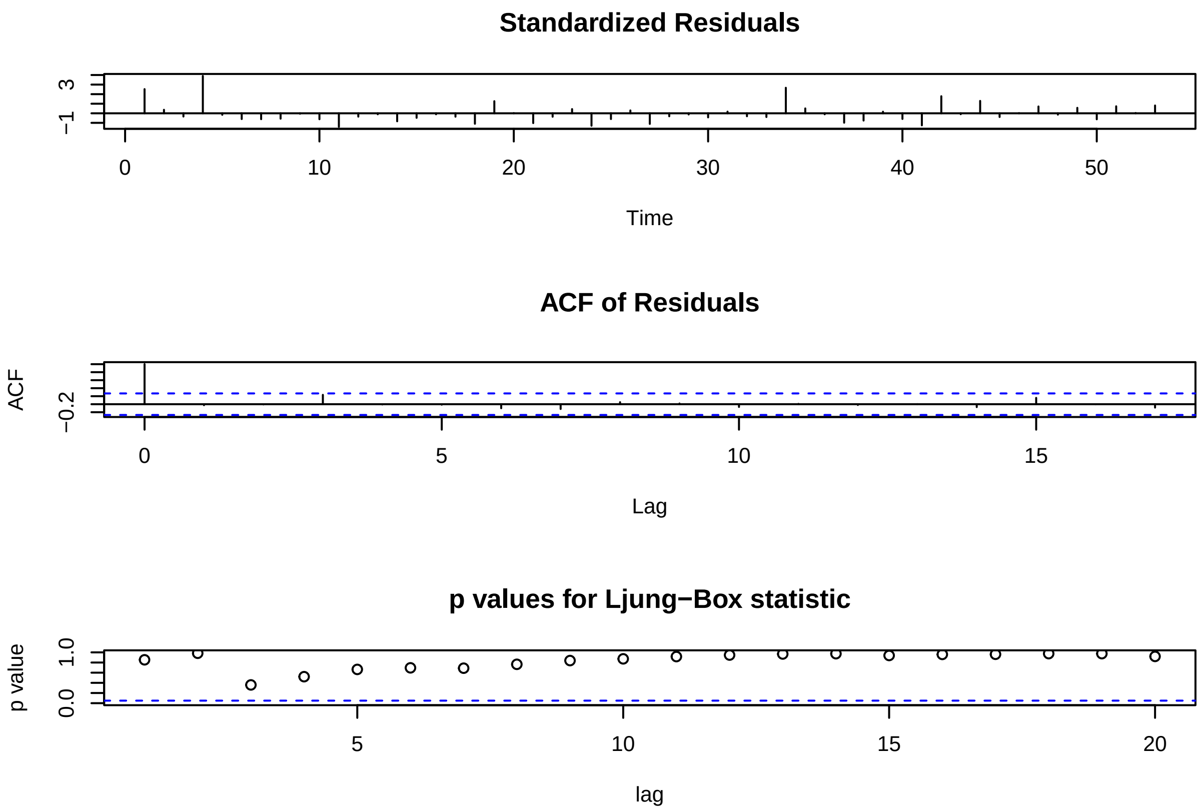

```
drought_fit6=sarima(drought_resi_final,3,0,1)
```



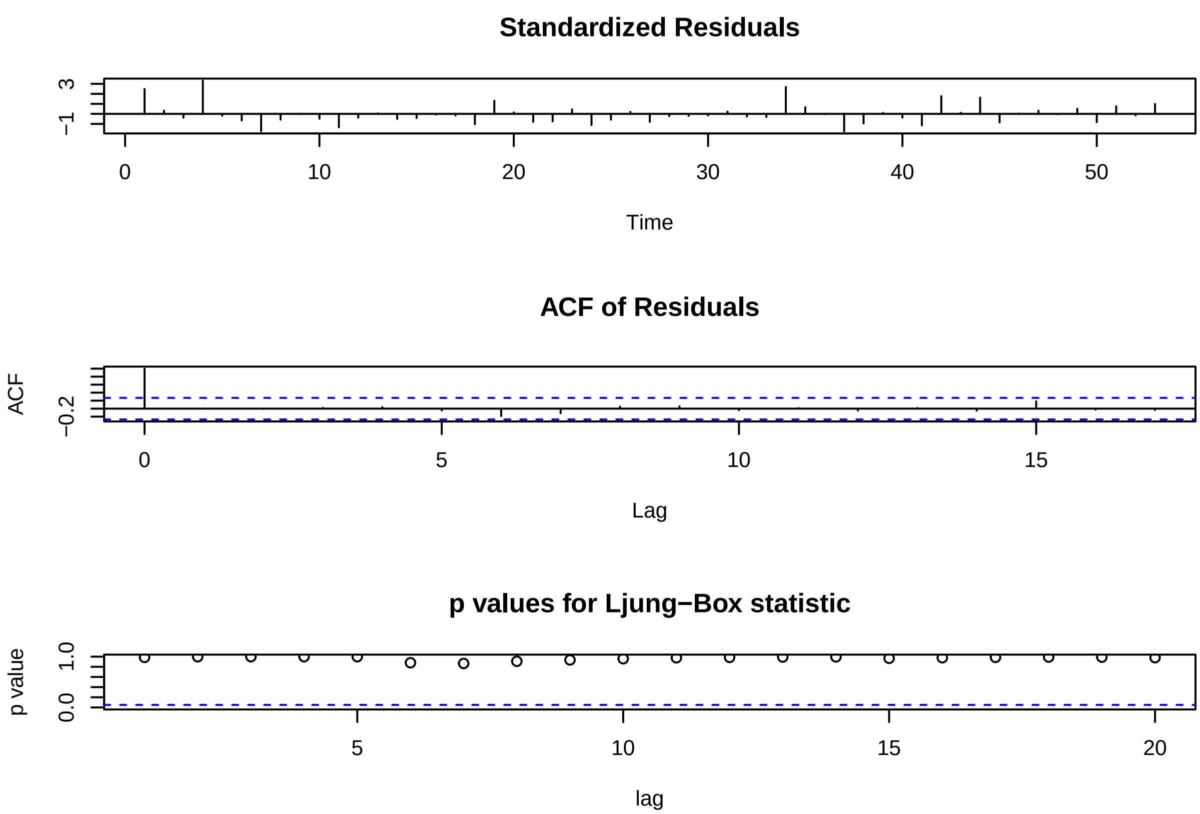
```r
drought_fit7=sarima(drought_resi_final,1,0,0)
```




```r
drought_fit8=sarima(drought_resi_final,2,0,0)
```




```
drought_fit9=sarima(drought_resi_final,3,0,0)
```



We can also loop through the possible models ARMA(p,q) up to lag 3. That is, we check an ARMA(1,0),

ARMA(2,0), … , ARMA(1,1), ARMA(1,2), ARMA(3,1) and so on. . . .

```
#This little function extracts the
#AIC, AICc and BIC values from an Arima() fit
getAIC <- function(fit) {
  c(fit$AIC, fit$AICc, fit$BIC)
}
```

We will summarize the AIC-related results in a table and siplay the table

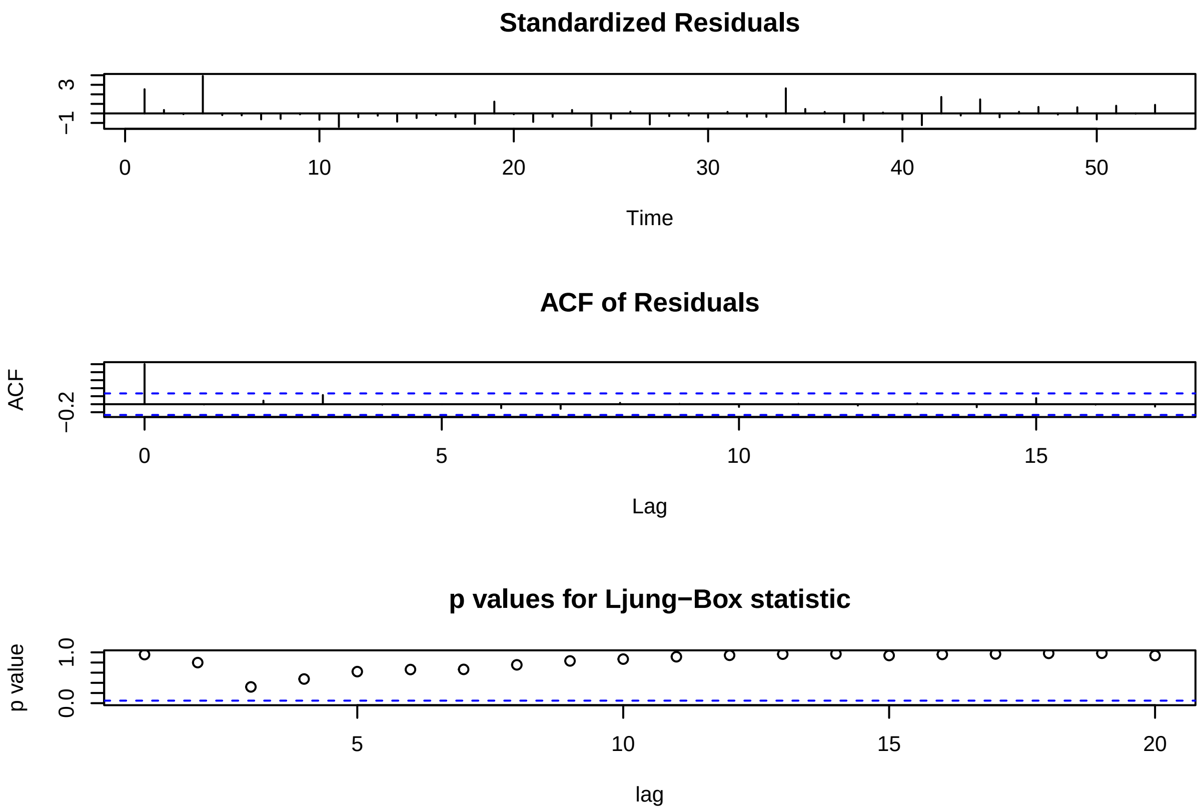
```
tab <- rbind(getAIC(drought_fit1), getAIC(drought_fit2), getAIC(drought_fit3),
             getAIC(drought_fit4), getAIC(drought_fit6),
            getAIC(drought_fit7), getAIC(drought_fit8), getAIC(drought_fit9))
colnames(tab) <- c("AIC", "AICc", "BIC")
rownames(tab) <- c("ARMA(1,1)","ARMA(2,1)","ARMA(1,2)","ARMA(2,2)","ARMA(3,1)","AR(1)", "AR(2)", "AR(3)
kable(tab)  #displays the table
```

	AIC	AICc	BIC
ARMA(1,1)	19.45575	19.50921	18.56728
ARMA(2,1)	19.47090	19.53272	18.61960
ARMA(1,2)	19.44825	19.51008	18.59696
ARMA(2,2)	19.31727	19.38946	18.50315
ARMA(3,1)	19.46553	19.53772	18.65140
AR(1)	19.42881	19.47579	18.50316
AR(2)	19.45741	19.51087	18.56894
AR(3)	19.42793	19.48975	18.57663

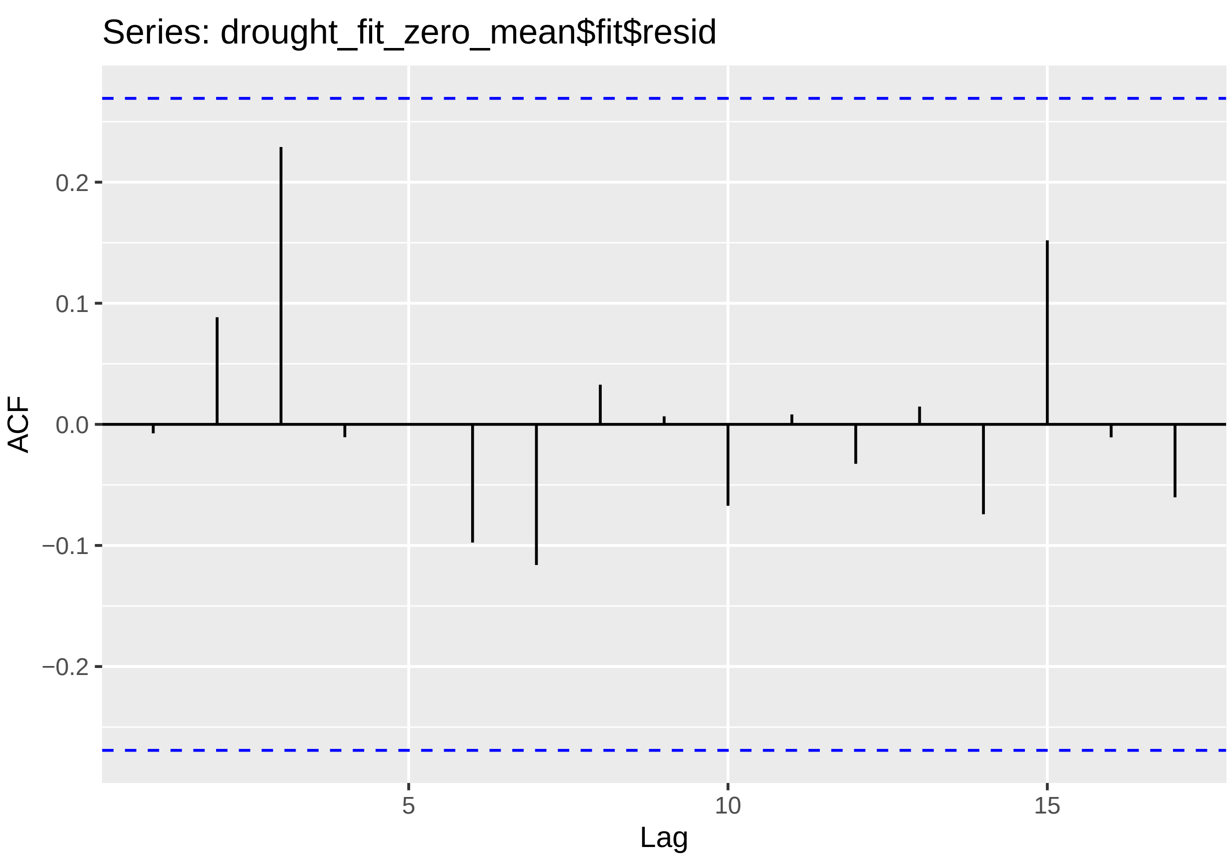
From the table, using a combination of the AIC, AICc, and BIC values reported (remember when the values are within 2 of one another, they are essentially the same). We see that all of these models are the same. We want to choose the simplest model.

For now, let us consider the $AR(1)$ model for simplicity.

```
drought_fit_zero_mean=sarima(drought_resi_final,0,0,0)
```




```r
ggAcf(drought_fit_zero_mean$fit$resid)
```



```r
finalfit <- Arima(ts_drought, order=c(0, 0, 0))
finalfit
```

```
## Series: ts_drought
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##           mean
##       9336.021
## s.e.  1451.109
##
## sigma^2 = 113749084:  log likelihood = -566.26
## AIC=1136.52   AICc=1136.76   BIC=1140.46
```

Our best model comes out at as a non-zero mean ARIMA(0,0,0), this is an indication of a constant mean. Our model equation being

$$X_t = 9336.021$$

To validate our model along with check for seasonality we will call the auto.arima() function.

```r
auto.arima(ts_drought)
```

```
## Series: ts_drought
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##           mean
##       9336.021
## s.e.  1451.109
##
## sigma^2 = 113749084:  log likelihood = -566.26
## AIC=1136.52   AICc=1136.76   BIC=1140.46
```

This code above directly validates our ARIMA(0,0,0) being the best model for the time series Drought.

5) (5pts) Refer to the seasonal means models you built for each disaster group in Exam-1. Compare and contrast those results with the findings from questions (3) and (4) above. Based on this analysis, provide recommendations with justifications on which model should be used for modeling the cost of each disaster group. Please provide a minimum of five sentences for your response.

In comparing the results of the seasonal means models I built for each disaster group in Exam-1, we can see that both models struggles with having any significant predictors besides the intercept term. However, for the models build from the periodogram in question 3, we can see there was atleast one predictor per disaster group which is more than what was seen from the models in the last project. For a winter/freeze model my recommendation would be to use the model in question 4. I would use this model because it shows the least amount of predictors and it is the simpler model for this time series and by the Dickey-Fuller test, we know that this time series is not stationary, thus I would use a non-stationary model as scene in question 4. For a SevereStorm/Flooding model, I would use the seasonal means model from exam 1, as a result of the dickey-fuller test telling us the model is stationary, and both the models done in exam 2 are non stationary models. For Cyclones/Tropical, I would choose model from question 4. From the dickey-fuller test we know this model is not stationary, which leads us to believe we should be using a model from exam 2. We use the model from question 4 as a result of the constant mean model being much simpler than a cosine/sine predictor model. For the final disaster group being WF/drought, my recommendation would be the seasonal means model from Exam 1. The dickey-fuller test here tells us this model is stationary, so we would lean with the stationary assumption from the model seen in exam 1. This model also has only 1 significant predictor making it a very simpler model to work with.

6) (5pts) Reflect on the in-class and take-home portion of the exam (a minimum of five sentences are required for full credit).

In reflecting of the in-class and the take-home portion of the exam, I have a couple of thoughts. In regards to the in-class portion of the exam, I felt it wasn't too difficult, besides how long it took to complete. For the first problem, I got very confused on how to take an expected value of the something were we never told if it was a zero or non-zero mean which made it hard for the timing aspect because I kept second guessing myself. Also I felt that question 1 was so logn that it made it to where I didn't have time to start question 2. For the take-home portion of the exam, I felt it was a little difficult to analyze the periodograms for the model equations because they looked different then ones we have seen before. Also, when finding the SARIMA models it was difficult to analyze a non-zero mean with no significant predictors, this was another thing that seems wrong, but I have done everything right up to it. I think that the take-home exam was not coding wise difficult like the first one, however, it was tedious and some of the analysis was quite difficult.