project3

Adhrit Srivastav, eid: Ams22362

11/26/2021

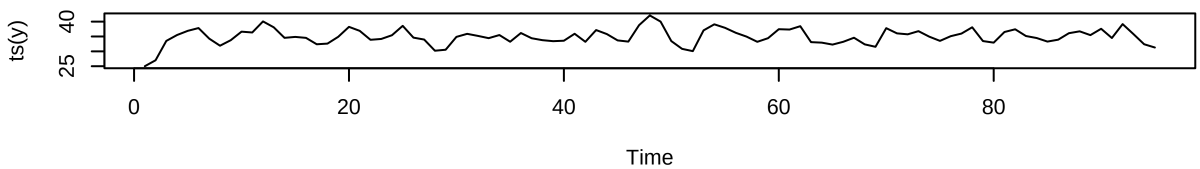
Problem 1

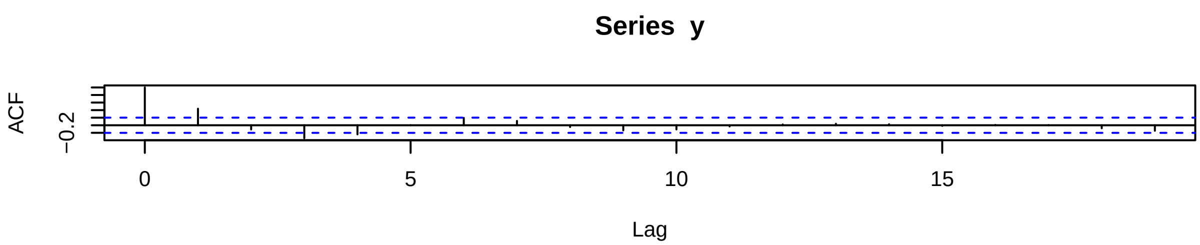
```
library(forecast)
```

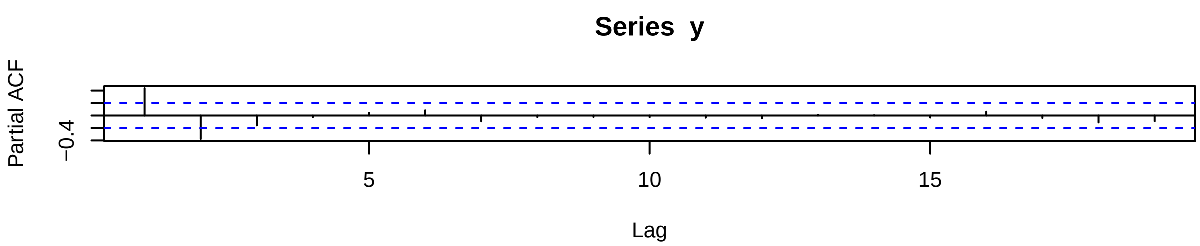
```
## Warning: package 'forecast' was built under R version 4.1.2
```

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

```
viscocity = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/exam3/Viscosity.csv")
attach(viscocity)


#Step 0 Graph
# data appears stationary bc no linear decay in the ACF
# cuts after lag 2 in the PACF
par(mfrow=c(3,1))
plot(ts(y))
acf(y)
pacf(y)
```



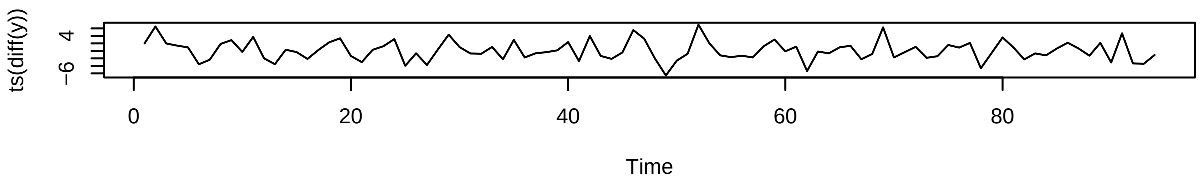


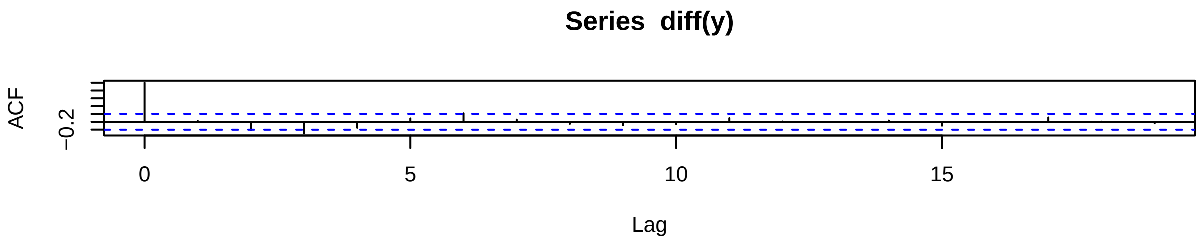


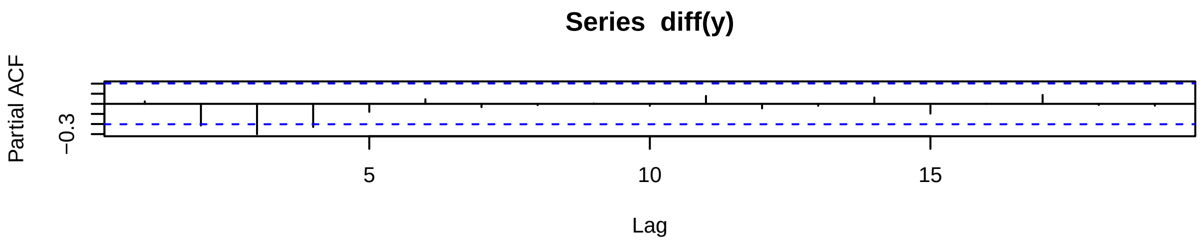

```
ndiffs(y)
```

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

```
#look at diff(y)
#Step 1 Tentative Identification
# diffs appear stationary bc no linear decay in the ACF
# cuts after lag 1 in PACF
plot(ts(diff(y)))
acf(diff(y))
pacf(diff(y))
```








```r
#Step 2 Estimation (the fit)
fit <- arima(y, order=c(0,1,1))    # MA(1) model
fit1<-arima(y,order=c(1,0,0))    # AR(1) model
fit3 <- auto.arima(y)

fit
```

```
##
## Call:
## arima(x = y, order = c(0, 1, 1))
##
## Coefficients:
##          ma1
##       0.0413
## s.e.  0.1276
##
## sigma^2 estimated as 6.765:  log likelihood = -223.24,  aic = 450.47
```

```r
fit1
```

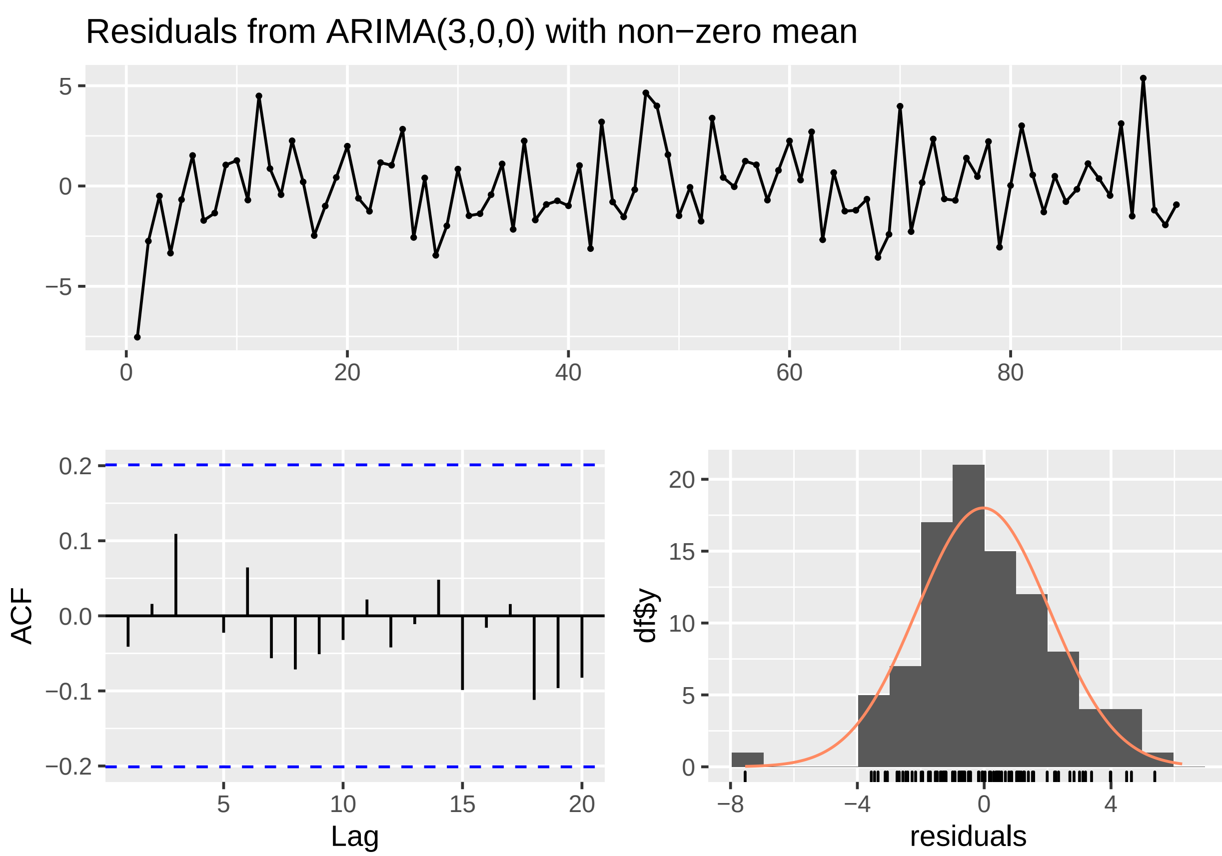
```
##
## Call:
## arima(x = y, order = c(1, 0, 0))
##
## Coefficients:
```

```
##          ar1  intercept
##       0.5213    34.7778
## s.e.  0.0976     0.4954
##
## sigma^2 estimated as 5.389:  log likelihood = -214.96,  aic = 435.92
```

```
fit3
```

```
## Series: y
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3     mean
##       0.5961  -0.3042  -0.2309  34.9858
## s.e.  0.1039   0.1184   0.1098   0.2306
##
## sigma^2 estimated as 4.522:  log likelihood=-204.89
## AIC=419.78   AICc=420.46   BIC=432.55
```

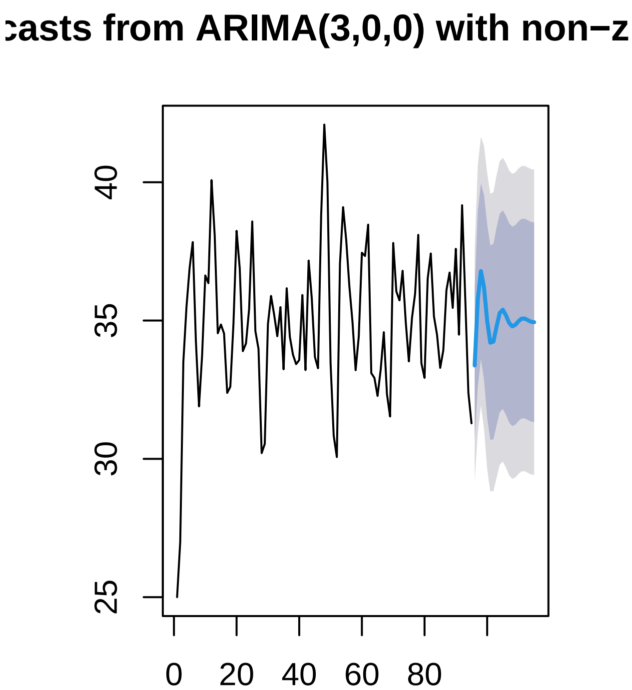
```
#Step 3 Check Residuals
# there is no evidence of significant autocorrelation at any lag of the residuals
checkresiduals(fit3)
```




```
##
##  Ljung-Box test
```

```
##
## data:  Residuals from ARIMA(3,0,0) with non-zero mean
## Q* = 3.1318, df = 6, p-value = 0.7921
##
## Model df: 4.   Total lags used: 10
```

```
#Step 4 Forecast
par(mfrow=c(1,2))
#plot(forecast(fit2,h=20))
plot(forecast(fit3,h=20))
```



Problem 2

```
library(forecast)
library(lmtest)
```

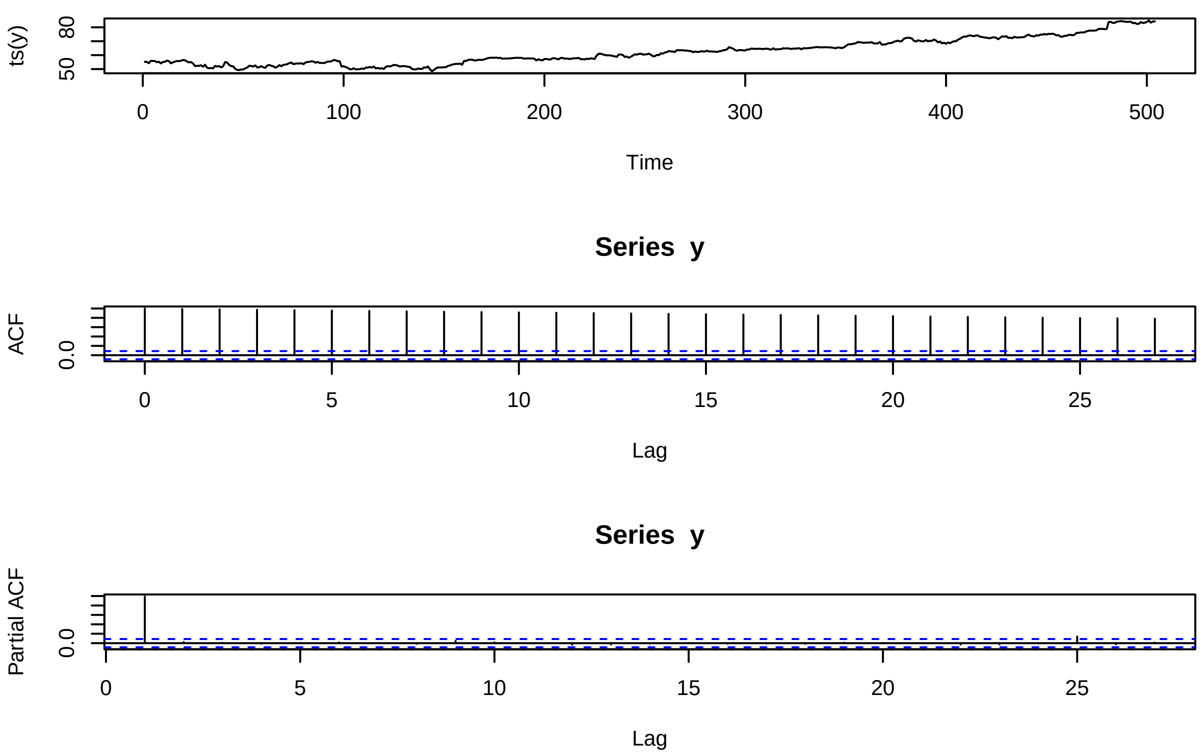
```
## Warning: package 'lmtest' was built under R version 4.1.2
```

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

```
## Warning: package 'zoo' was built under R version 4.1.2
```

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

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

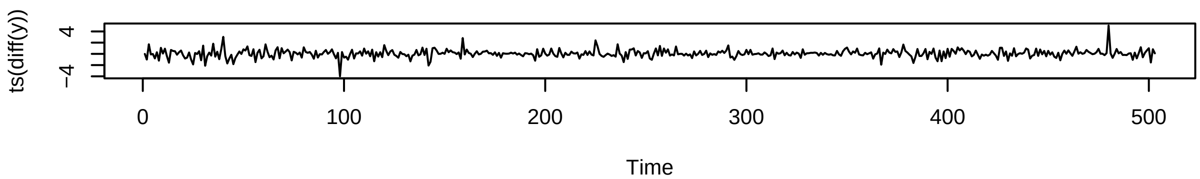
```r
tech = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/exam3/TechStocks.csv")
attach(tech)

y = MSFT

#Step 0 Graph
# data appears non-stationary bc linear decay in the ACF
par(mfrow=c(3,1))
plot(ts(y))
acf(y)
pacf(y)
```

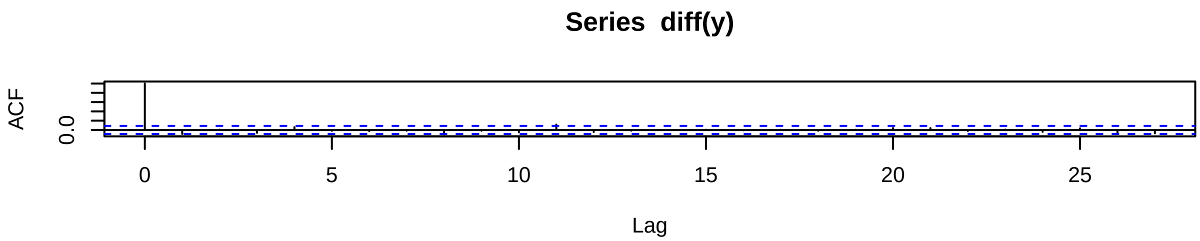


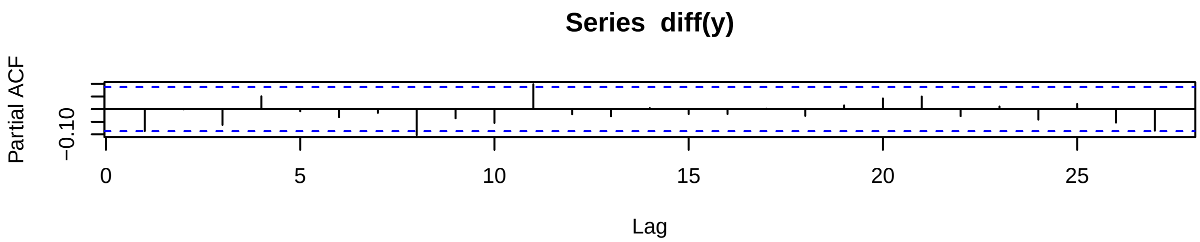

```r
ndiffs(y)
```

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

```r
#look at diff(y)
#Step 1 Tentative Identification
# diffs appear stationary bc no linear decay in the ACF
# dies on PACF
plot(ts(diff(y)))
acf(diff(y))
pacf(diff(y))
```








```r
ydiff = diff(y)

#Step 2 Estimation (the fit)
#fit <- arima(ydiff, order=c(0,1,1))   # MA(1) model
fit1.0<-arima(ydiff,order=c(1,0,0))   # AR(1) model; 2nd term not important, so we stop at AR(1)
fit2 <- auto.arima(ydiff)

fit1.0
```

```
##
## Call:
## arima(x = ydiff, order = c(1, 0, 0))
##
## Coefficients:
##           ar1  intercept
##       -0.0861     0.0578
## s.e.   0.0444     0.0295
```

```
##
## sigma^2 estimated as 0.516:  log likelihood = -547.35,  aic = 1100.7
```

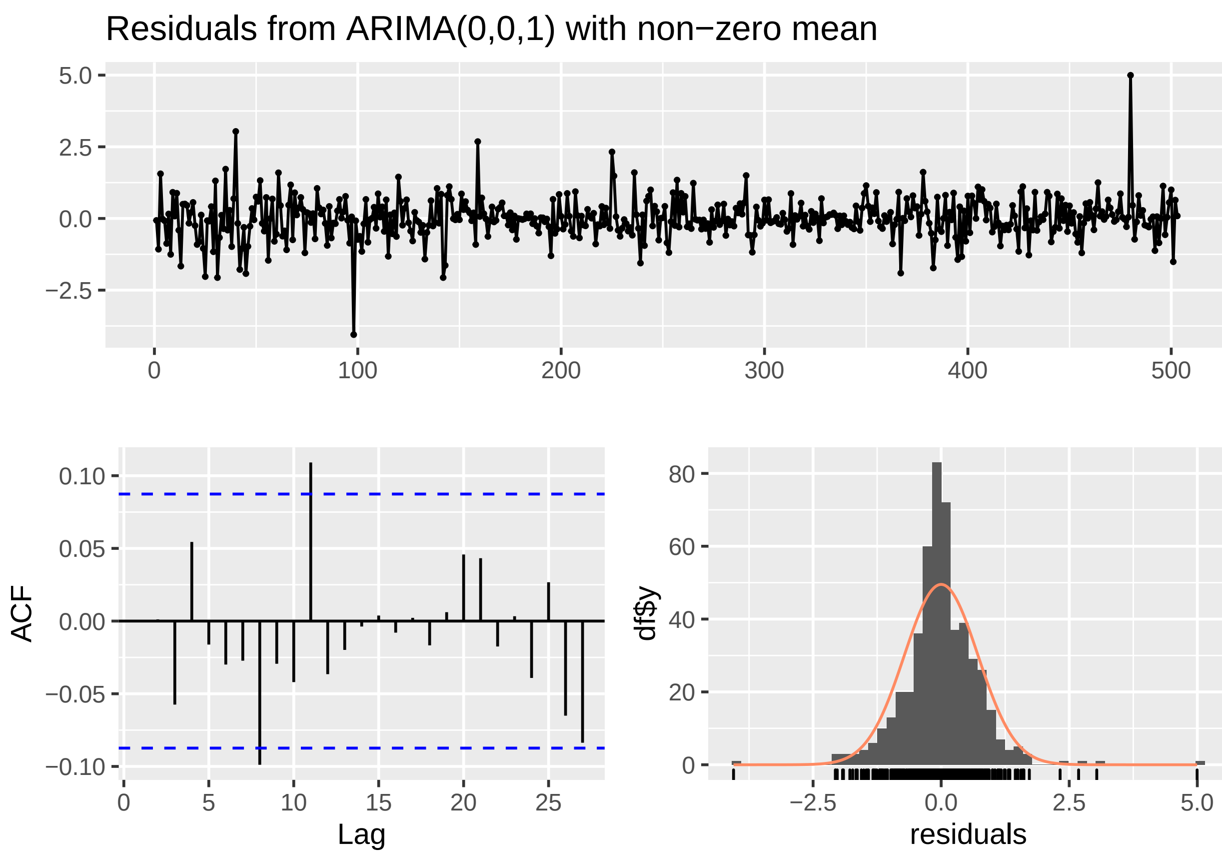
```r
fit2
```

```
## Series: ydiff
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##           ma1    mean
##       -0.0870  0.0577
## s.e.   0.0447  0.0292
##
## sigma^2 estimated as 0.5181:  log likelihood=-547.34
## AIC=1100.68   AICc=1100.73   BIC=1113.34
```

```r
coeftest(fit2)    # all tests are significant
```

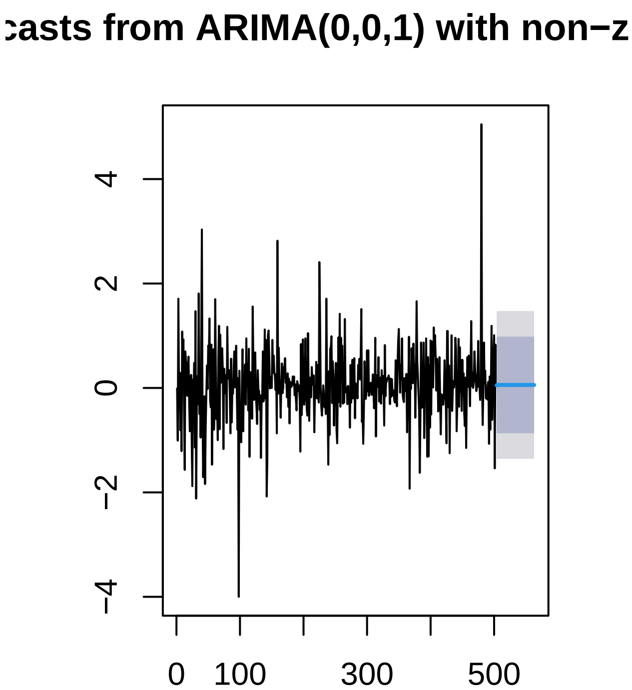
```
##
## z test of coefficients:
##
##            Estimate Std. Error z value Pr(>|z|)
## ma1       -0.086977   0.044723 -1.9448  0.05180 .
## intercept  0.057742   0.029249  1.9741  0.04837 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Step 3 Check Residuals
# there is supposed significant autocorrelation at lags 8 and 11 of the residuals but they are not seas
# so we can proceed to forecasting
checkresiduals(fit2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,1) with non-zero mean
## Q* = 10.519, df = 8, p-value = 0.2304
##
## Model df: 2.   Total lags used: 10
```

```
#Step 4 Forecast
par(mfrow=c(1,2))
#plot(forecast(fit2,h=20))
plot(forecast(fit2,h=60))
```



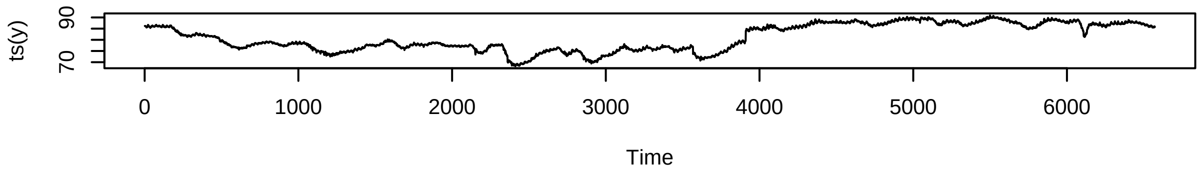
Problem 3

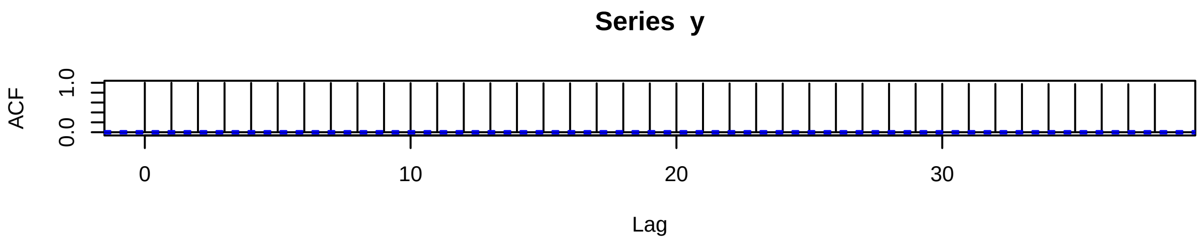
Data is seasonal so we need to do seasonal ARIMA.

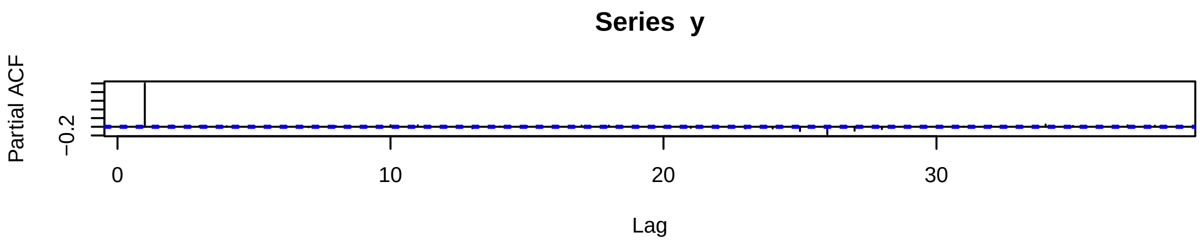
```
library(forecast)
library(lmtest)

west = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/exam3/KeyWest.csv")
attach(west)
```

```
## The following object is masked from tech:
##
##     t
```

```
y = WaterTemp

#Step 0 Graph
# data appears non-stationary bc linear decay in the ACF
# cuts on lag 1 for PACF
par(mfrow=c(3,1))
plot(ts(y))
acf(y)
pacf(y)
```



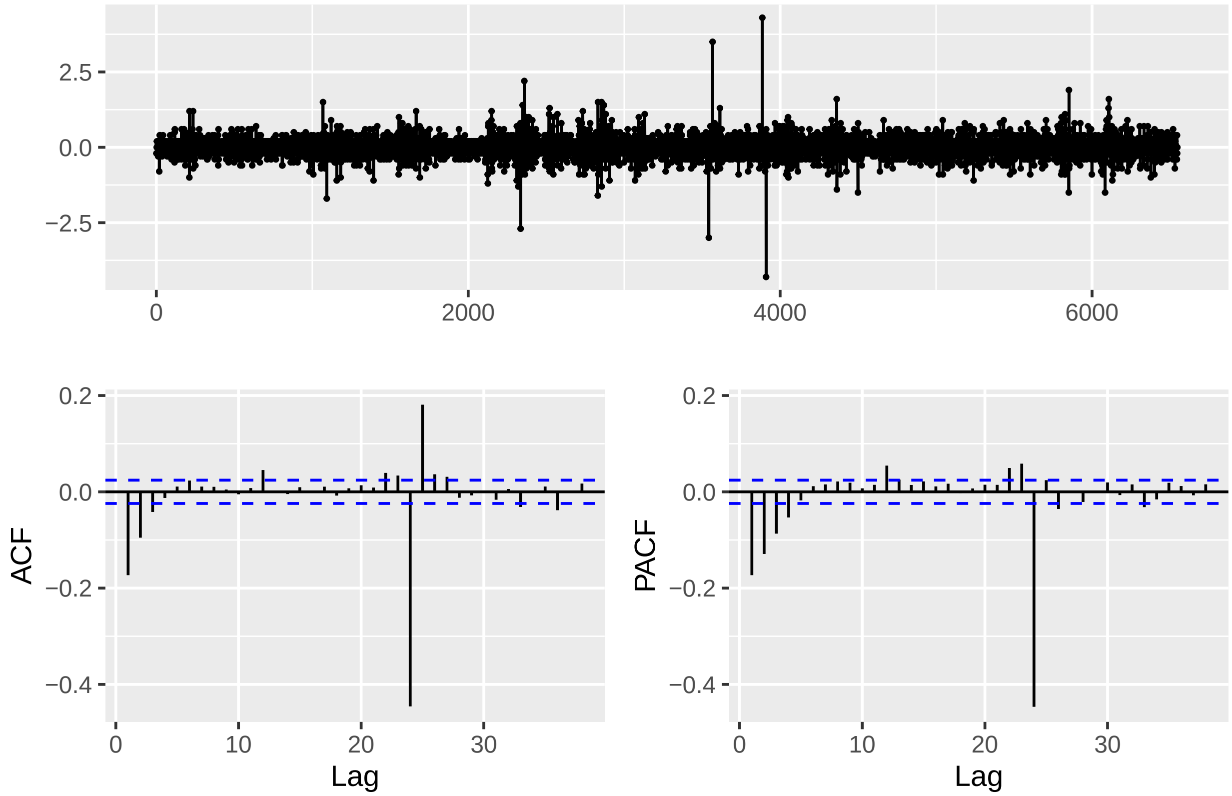





```
ndiffs(y)
```

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

```
#look at diff(y)
#Step 1 Tentative Identification
# there are significant autocorrelations scattered on the ACF and PACF

# The data is incomplete bc water temperature is seasonal on a yearly basis, but the data does not
# go beyond a year. This means seasonal models will not work well on the data despite the fact that
# the data is seasonal. We'd need more data to create an effective model.

y %>% diff() %>% diff(lag=24) %>% ggtsdisplay()
```




```
#Step 2 Estimation (the fit)
#fit <- arima(ydiff, order=c(0,1,1))   # MA(1) model
fit1.0<-arima(y,order=c(0,0,1), seasonal = c(0,1,1))   # AR(1) model; 2nd term not important, so we sto
fit2 <- auto.arima(y)   # MA(2) model

fit1.0
```

```
##
## Call:
## arima(x = y, order = c(0, 0, 1), seasonal = c(0, 1, 1))
##
## Coefficients:
```

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

```
##           ma1     sma1
##       -0.0195  -0.0195
## s.e.      NaN      NaN
##
## sigma^2 estimated as 0.04896:  log likelihood = 587.56,  aic = -1169.12
```

```
fit2
```

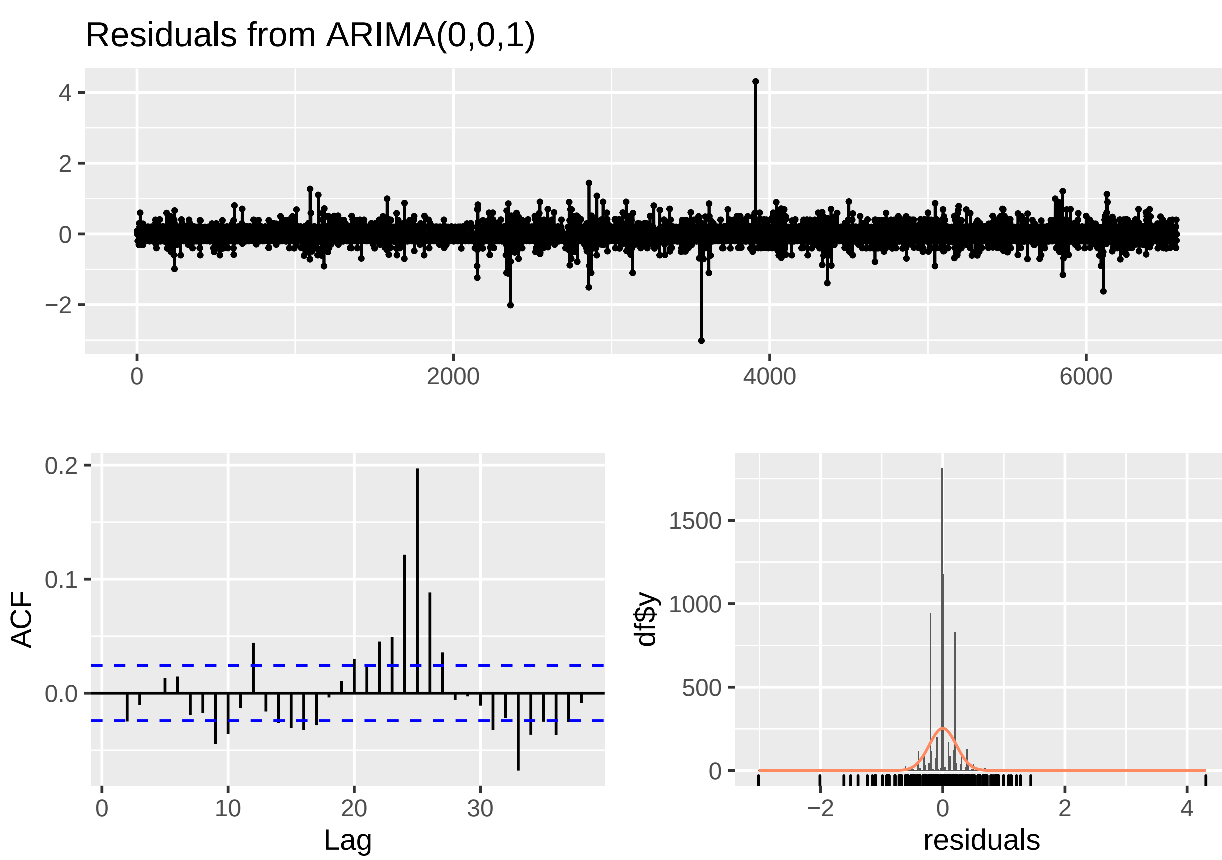
```
## Series: y
```

```
## ARIMA(0,1,2)
##
## Coefficients:
##           ma1      ma2
##       -0.0390  -0.0248
## s.e.   0.0123   0.0123
##
## sigma^2 estimated as 0.04895:  log likelihood=589.64
## AIC=-1173.29   AICc=-1173.29   BIC=-1152.92
```

```
coeftest(fit2)     # all tests are significant
```

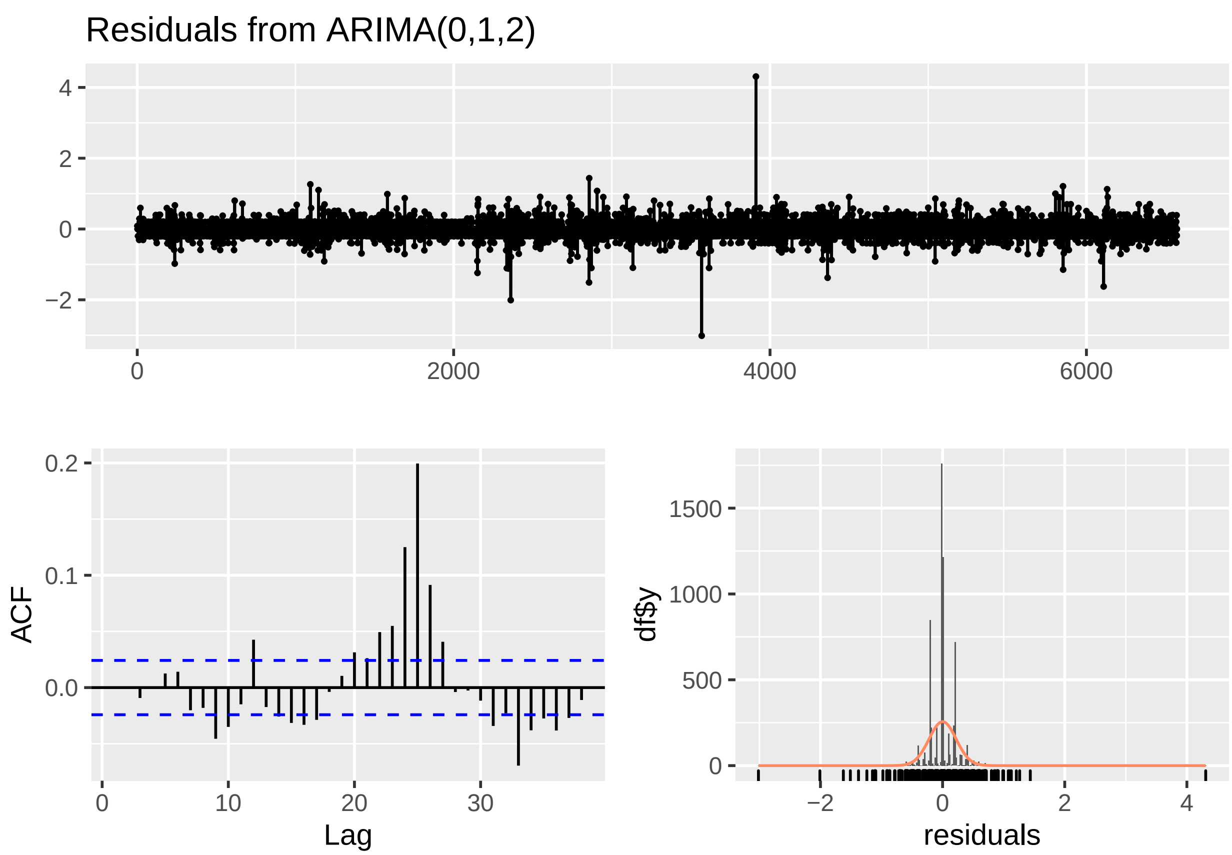
```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.038962   0.012329 -3.1603 0.001576 **
## ma2 -0.024779   0.012308 -2.0133 0.044088 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Step 3 Check Residuals
# there is supposed significant autocorrelation at lags 8 and 11 of the residuals but they are not seas
# so we can proceed to forecasting
checkresiduals(fit1.0)
```



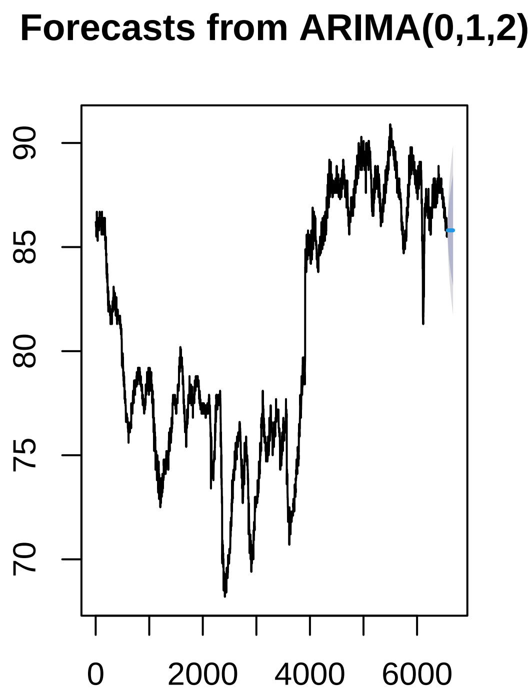
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,1)
## Q* = 33.362, df = 8, p-value = 5.299e-05
##
## Model df: 2.   Total lags used: 10
```

```
checkresiduals(fit2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)
## Q* = 29.474, df = 8, p-value = 0.0002617
##
## Model df: 2.   Total lags used: 10
```

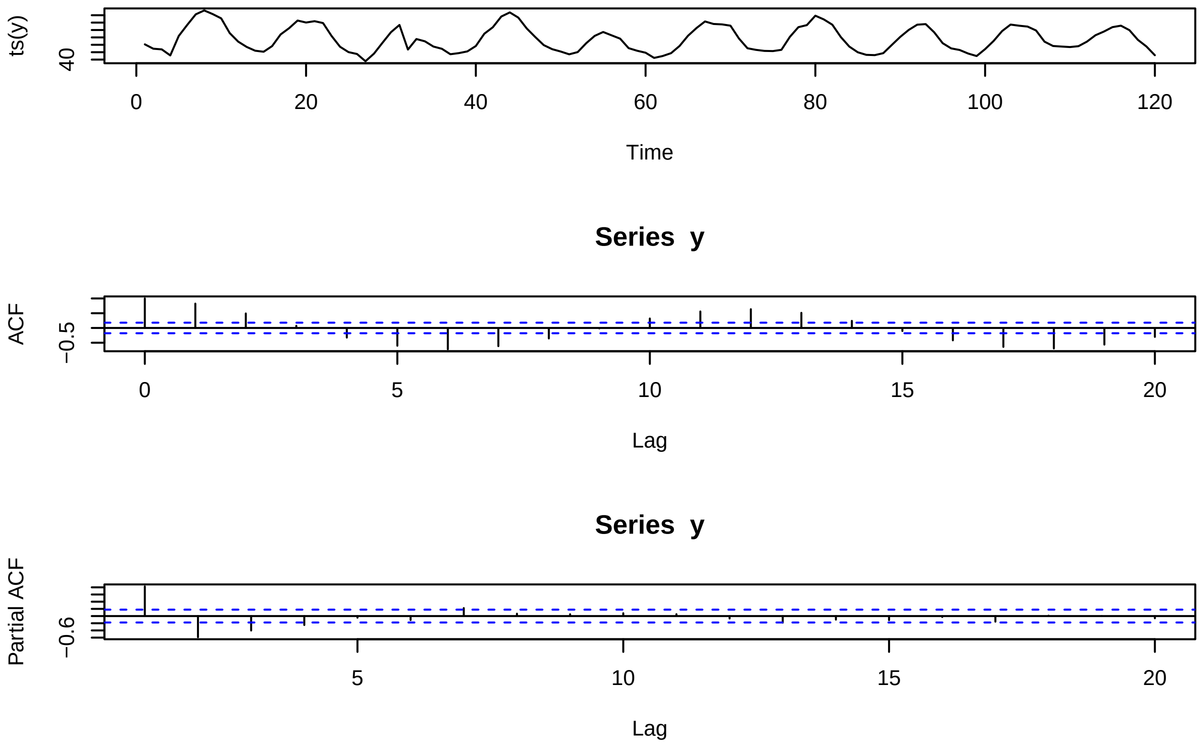
```
#Step 4 Forecast
par(mfrow=c(1,2))
#plot(forecast(fit2,h=20))
plot(forecast(fit2,h=100))
```



Problem 4

Data is seasonal so we need to do seasonal ARIMA.

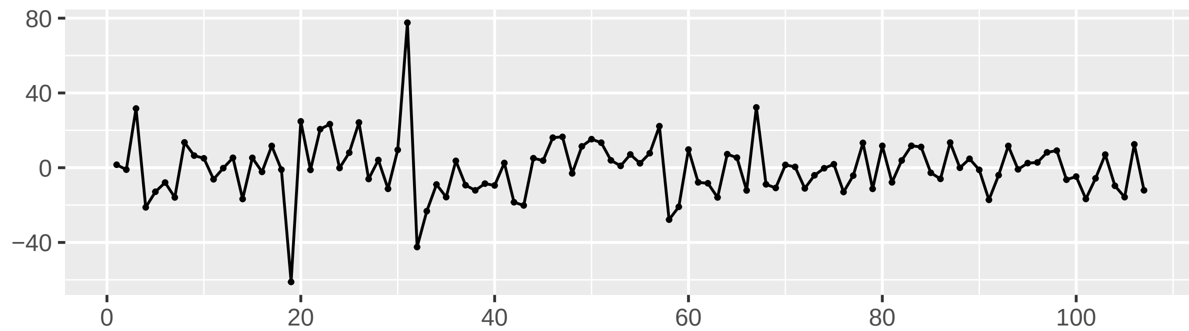
```
library(forecast)
library(lmtest)

beans = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/exam3/StockBeans.csv")
attach(beans)

y = stockbeans

#Step 0 Graph
# data appear stationary bc no linear decay in the ACF
# cuts on lag 1 for PACF
par(mfrow=c(3,1))
plot(ts(y))
acf(y)
pacf(y)
```

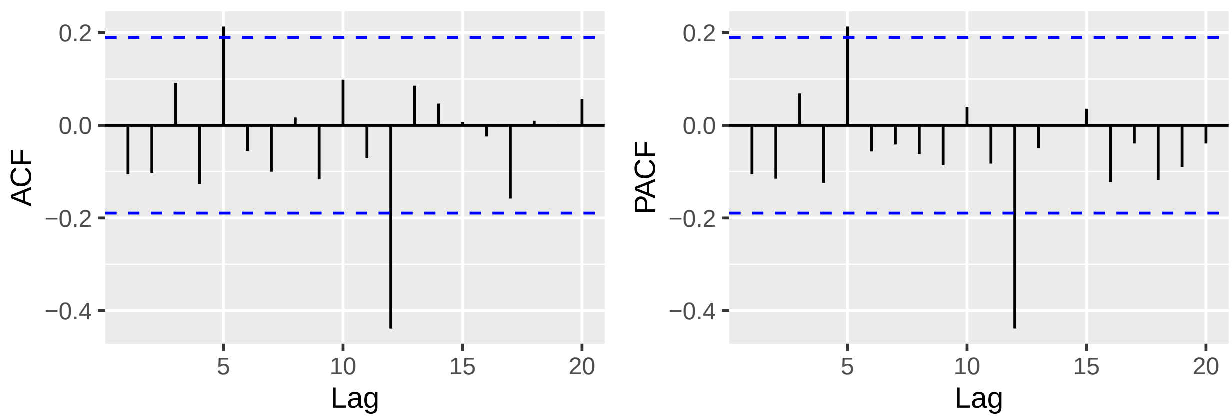



```
ndiffs(y)
```

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

```
#Step 1 Tentative Identification
# ACF appears stationary now
# cuts after lag 12 for ACF and PACF
y %>% diff() %>% diff(lag=12) %>% ggtsdisplay()
```





```
#Step 2 Estimation (the fit)
fit1.0<-arima(y,order=c(0,1,1), seasonal = list(order = c(0,1,1), period = 12))
fit1.1<-arima(y,order=c(1,1,0), seasonal = list(order = c(0,1,1), period = 12))
#fit2 <- auto.arima(y)   # AR(2) MA(2) model

fit1.0
```

```
##
## Call:
## arima(x = y, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12))
##
## Coefficients:
##           ma1     sma1
##       -0.1661  -0.9311
## s.e.   0.1005   0.2887
##
## sigma^2 estimated as 130.1:  log likelihood = -422.74,  aic = 851.48
```

```
fit1.1
```

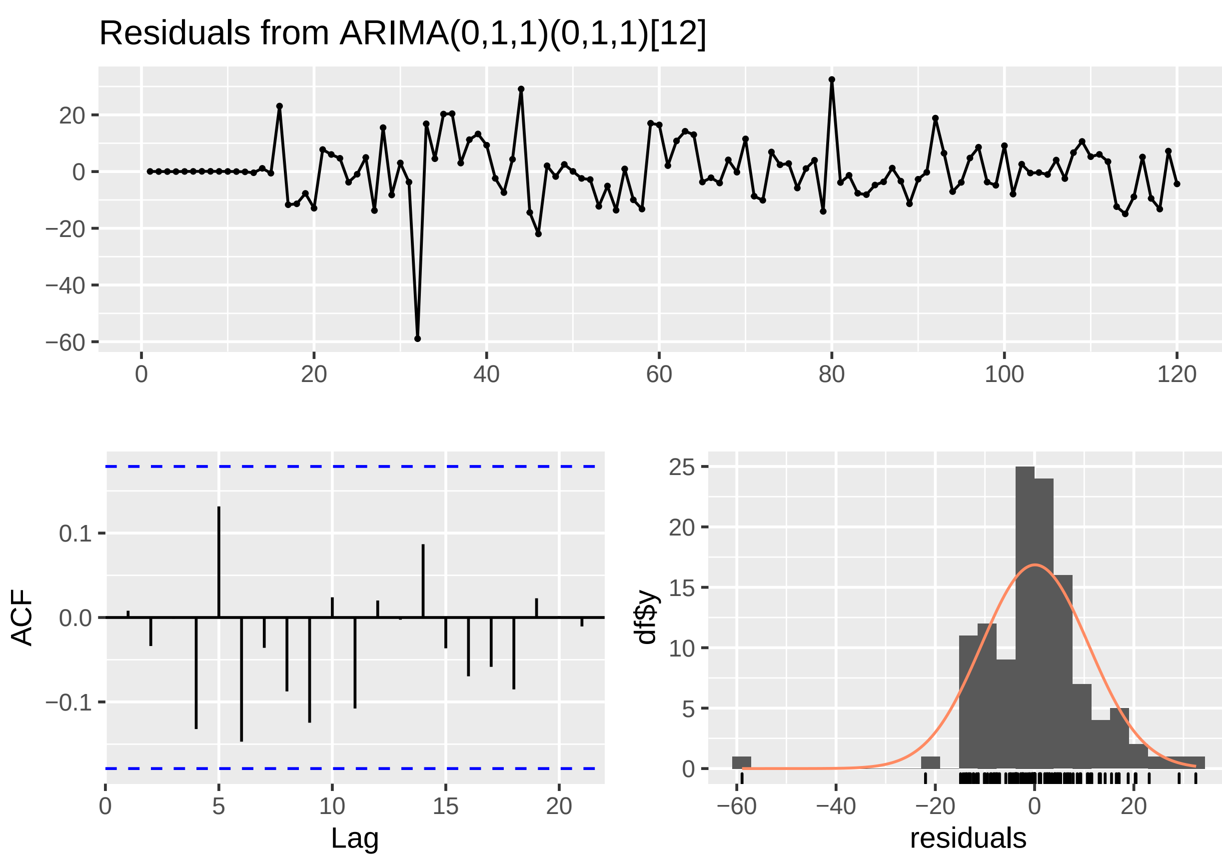
```
##
## Call:
## arima(x = y, order = c(1, 1, 0), seasonal = list(order = c(0, 1, 1), period = 12))
##
## Coefficients:
```

```
##           ar1     sma1
##       -0.1490  -0.9266
## s.e.   0.0956   0.2700
##
## sigma^2 estimated as 130.9:  log likelihood = -422.87,  aic = 851.74
```

```
#fit2
coeftest(fit2)    # all tests are significant
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 -0.038962   0.012329 -3.1603 0.001576 **
## ma2 -0.024779   0.012308 -2.0133 0.044088 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

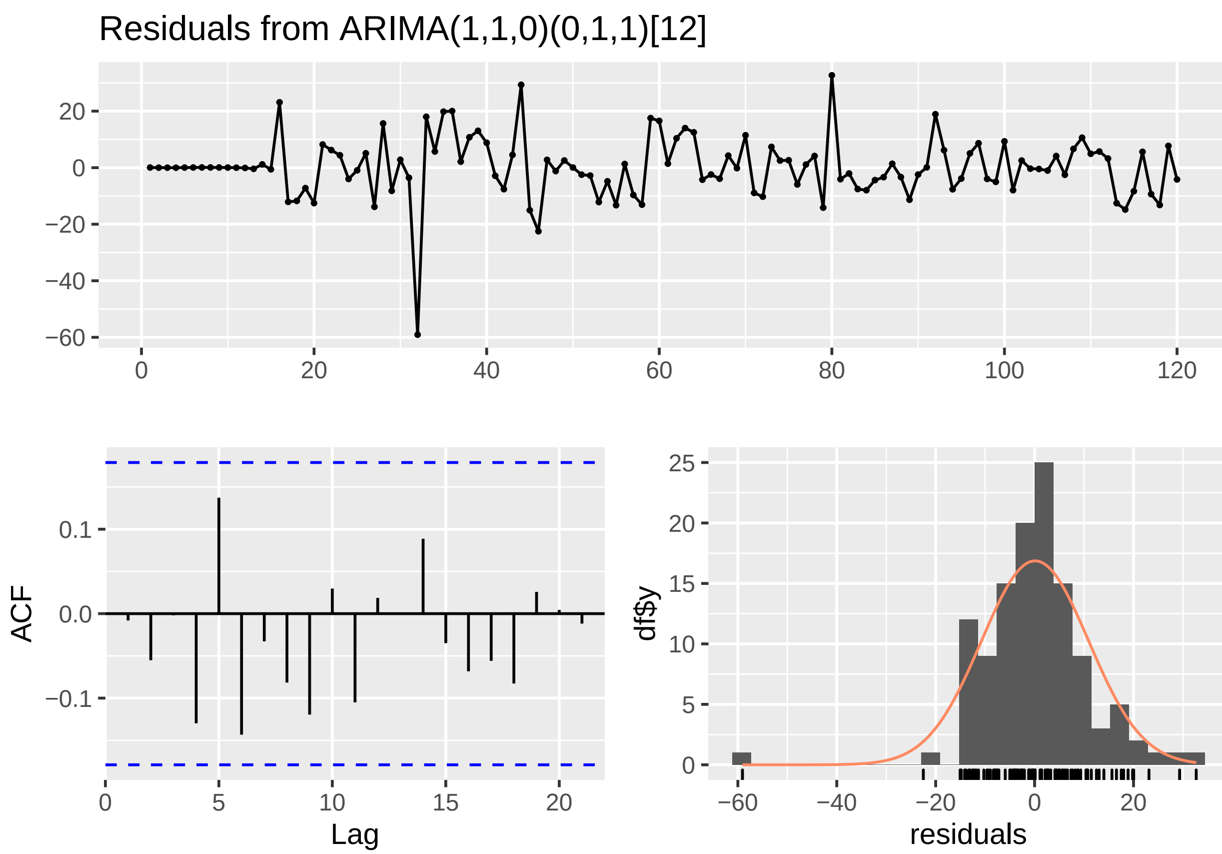
```
#Step 3 Check Residuals
# there is supposed significant autocorrelation at lags 8 and 11 of the residuals but they are not seas
# so we can proceed to forecasting
checkresiduals(fit1.0)
```




```
##
##  Ljung-Box test
```

```
##
## data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
## Q* = 10.627, df = 8, p-value = 0.2237
##
## Model df: 2.   Total lags used: 10
```

```
checkresiduals(fit1.1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(0,1,1)[12]
## Q* = 10.555, df = 8, p-value = 0.2282
##
## Model df: 2.   Total lags used: 10
```

```
#checkresiduals(fit2)


#Step 4 Forecast
forecast(fit1.0,h=10)
```

```
##     Point Forecast    Lo 80    Hi 80      Lo 95    Hi 95
## 121       42.42900 27.44426 57.41373 19.5118139 65.34618
```

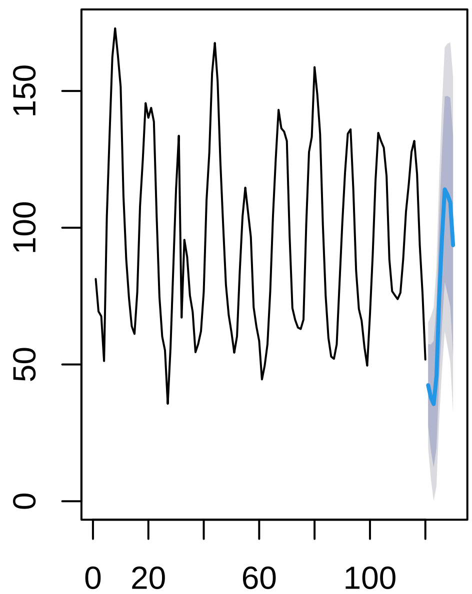
```
## 122        37.86205 18.38372  57.34038  8.0725151  67.65159
## 123        35.49648 12.38224  58.61073  0.1462944  70.84667
## 124        45.57799 19.32668  71.82930  5.4300780  85.72590
## 125        74.59643 45.54486 103.64801 30.1658794 119.02699
## 126        96.29131 64.68661 127.89601 47.9560974 144.62653
## 127       114.03078 80.06433 147.99723 62.0835707 165.97799
## 128       112.01473 75.84039 148.18907 56.6908543 167.33861
## 129       109.27271 71.01770 147.52772 50.7667236 167.77870
## 130        93.60832 53.38011 133.83652 32.0845863 155.13205
```

```
par(mfrow=c(1,2))
plot(forecast(fit1.0,h=10))

forecast(fit1.1,h=10)
```

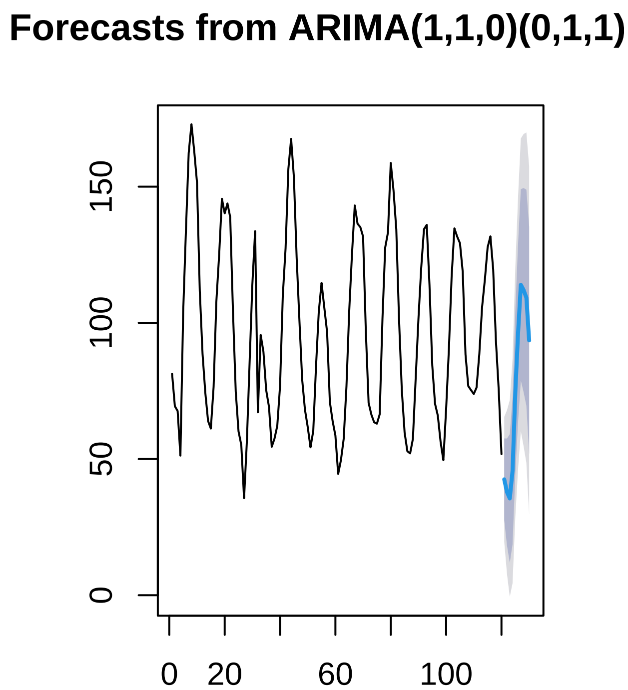
```
##     Point Forecast    Lo 80     Hi 80      Lo 95     Hi 95
## 121       42.54401 27.53345  57.55456 19.5873407  65.50067
## 122       37.85802 18.18124  57.53480  7.7649837  67.95106
## 123       35.52657 11.91330  59.13984 -0.5868166  71.63995
## 124       45.62594 18.66839  72.58350  4.3979195  86.85396
## 125       74.58365 44.64999 104.51732 28.8040612 120.36325
## 126       96.25859 63.61950 128.89769 46.3414059 146.17578
## 127      113.96493 78.82804 149.10181 60.2276962 167.70216
## 128      112.01701 74.54849 149.48554 54.7138485 169.32018
## 129      109.24408 69.58074 148.90742 48.5842396 169.90392
## 130       93.55155 51.80866 135.29444 29.7113064 157.39179
```

```
par(mfrow=c(1,2))
```

Forecasts from ARIMA(0,1,1)(0,1,1)




```
plot(forecast(fit1.1,h=10))
```



a) The time series plot appears to have a cyclical trend and is potentially seasonal. It continues this pattern quite consistently.

b) The ACF shows strong seasonal pattern and a declining trend that indicates a lack of stationarity. The PACF cuts after lag 4. From the ACF and PACF we can tell we will need to conduct differencing analysis and will need to build a seasonal ARIMA model.

c) When we use the difference values, the ACF becomes stationary and cuts after lag 12 and the PACF cuts after lag 12 as well.

d) The residuals of the two models are very similar. The AIC of the two models are very similar. The L'Jung Box test was also similar for the two models with similar p-values, df's, and lags used. The standard error values for the two models were also very similar.

e) $Y_t = \delta + \Phi_1 L Y_t + \epsilon_t - \theta_1 \epsilon_{t-s}$; $Y_t = \delta + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-s}$

f) Forecast is in the code above.