Time Series Analysis Project

Rene Lizarraga

April 21, 2019

Load libraries

```
library(astsa)

## Warning: package 'astsa' was built under R version 3.4.4

library(fpp2)

## Warning: package 'fpp2' was built under R version 3.4.4

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.4

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.4.4

##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas

## Loading required package: fma

## Warning: package 'fma' was built under R version 3.4.4

##
## Attaching package: 'fma'

## The following objects are masked from 'package:astsa':
##
##     chicken, sales

## Loading required package: expsmooth

## Warning: package 'expsmooth' was built under R version 3.4.4

##
## Attaching package: 'fpp2'

## The following object is masked from 'package:astsa':
##
##     oil
```

```r
library(ggplot2)
```

I selected “flu” dataset for analysis. Run “?flu”" in R console for data description.

flu {astsa} R Documentation Monthly pneumonia and influenza deaths in the U.S., 1968 to 1978.

Description

Monthly pneumonia and influenza deaths per 10,000 people in the United States for 11 years, 1968 to 1978.

Usage

data(flu) Format

The format is: Time-Series [1:132] from 1968 to 1979: 0.811 0.446 0.342 0.277 0.248 …

References

http://www.stat.pitt.edu/stoffer/tsa4/

Print head of flu

```r
data(flu)
head(flu)

##            Jan       Feb       Mar       Apr       May       Jun
## 1968 0.8113721 0.4458291 0.3415985 0.2774243 0.2484958 0.2525427
```

Verify object class is time-series. It is class ts.

```r
class(flu)

## [1] "ts"
```

View summary statistics.

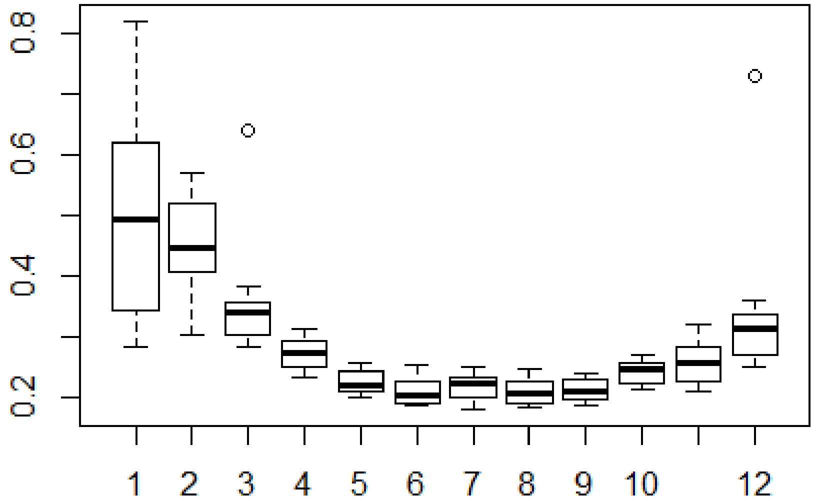
```r
cat(" mean:",mean(flu))

##  mean: 0.2918623

cat(" std dev:",sd(flu))

##  std dev: 0.1256254

cat(" min:",min(flu))

##  min: 0.1798065

cat(" max:",max(flu))

##  max: 0.8192756
```

View start, end, and frequecy of data. Data strats January 1968 and ends December 1978. The frequency is monthly.

```
cat("start:",start(flu))

## start: 1968 1

cat(" start:",end(flu))

##  start: 1978 12

cat(" frequency:",frequency(flu))

##  frequency: 12
```

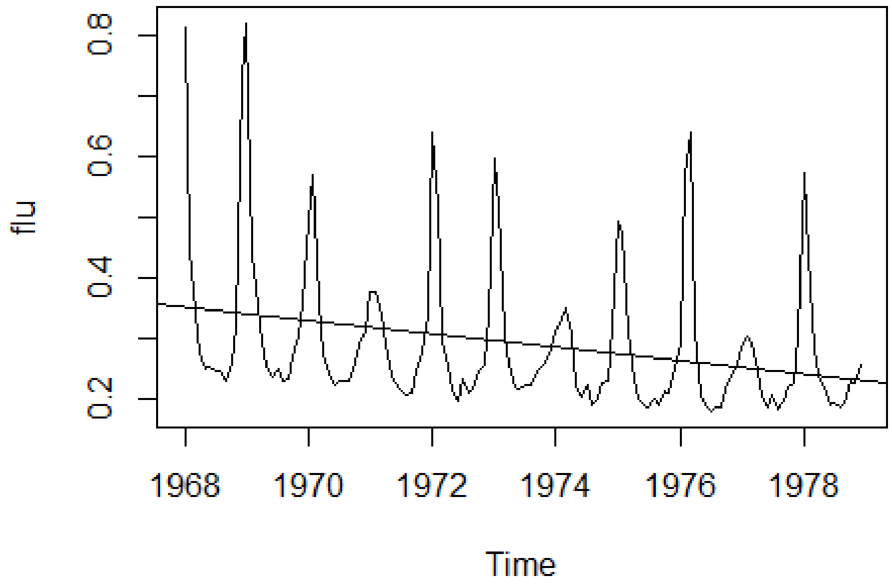
Generate a boxplot, There is a declining trend with an annual cycle. Flu deaths generally peak in January and bottom out in June.

```
boxplot(flu~cycle(flu))
```



Plot ab line for flu data.

```
plot(flu)
abline(reg=lm(flu~time(flu)))
```



The data show a decrease of variation over time. Note the min/max spread in 1968 and 1969 versus those in subsequent years.

Use a Box Cox lambda estimate function to estimate the degree to which the data should be transformed.

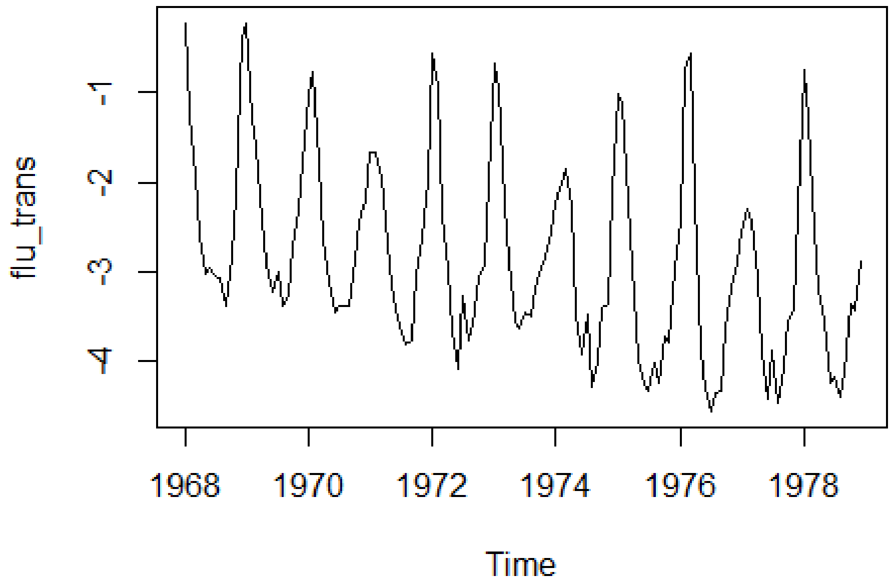
```
BoxCox.lambda(flu)

## [1] -0.9999242
```

Take this value for lambda and perform a Box Cox transformation.

```
flu_trans <- BoxCox(flu,lambda = -1)
```

Plot the transformed results. This seems to reduce the variations.

```
plot(flu_trans)
```

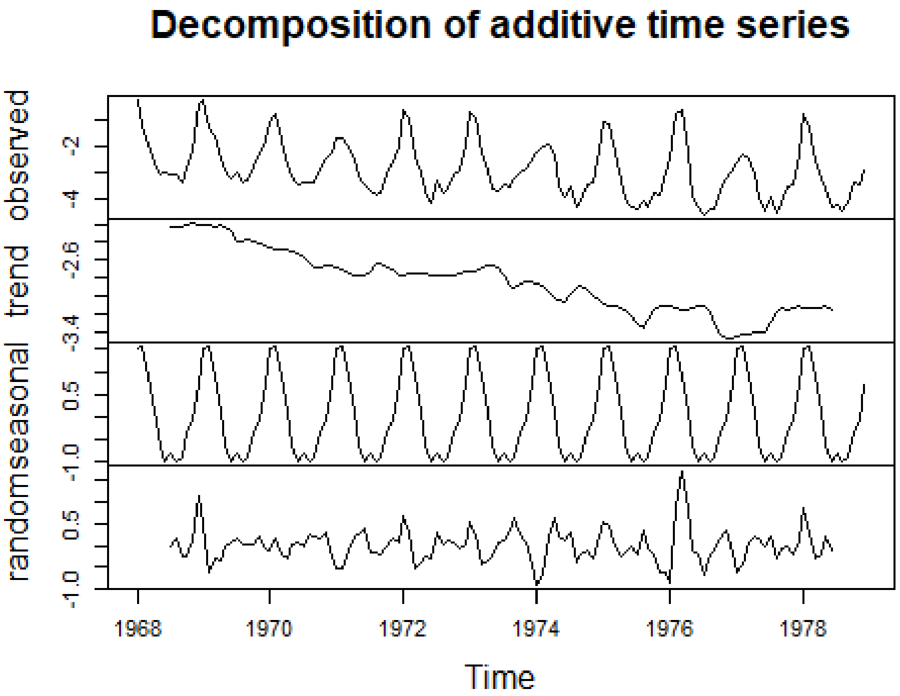


We decompose the flu data.

“Observed” is our transformed time series “Trend” show the general direction up or down of a time series. In this case there is a clear downward trend. “Seasonal” is a repeating pattern “Random is the noise in the time series not explained by trend or seasonality.

This plot separates the time series into the trend, seasonal, and random components.

```r
plot(decompose(flu_trans))
```

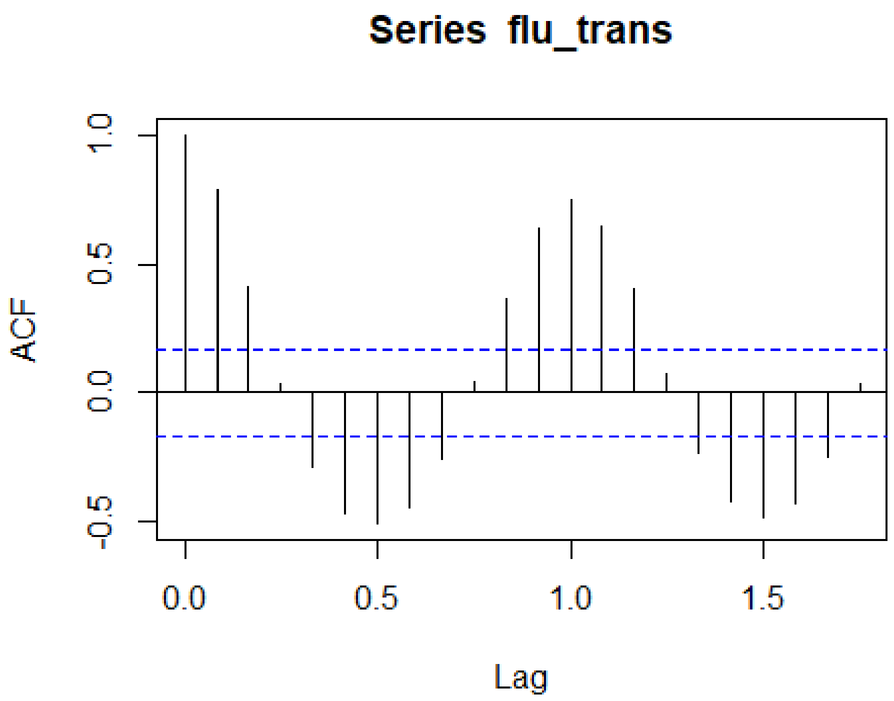


Fit a regression model. The r-square of 0.12 indicates 12% of the variation is explained by the model. This result is not good. A linear regression model does not do a good job estimating the data.

```
fit_lm <- lm(flu_trans~time(flu_trans), na.action=NULL)
summary(fit_lm)

##
## Call:
## lm(formula = flu_trans ~ time(flu_trans), na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3725 -0.8012 -0.2449  0.6788  2.6309
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     231.3145    53.8842   4.293 3.42e-05 ***
## time(flu_trans)  -0.1187     0.0273  -4.346 2.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9961 on 130 degrees of freedom
## Multiple R-squared:  0.1268, Adjusted R-squared:  0.1201
## F-statistic: 18.88 on 1 and 130 DF,  p-value: 2.776e-05
```

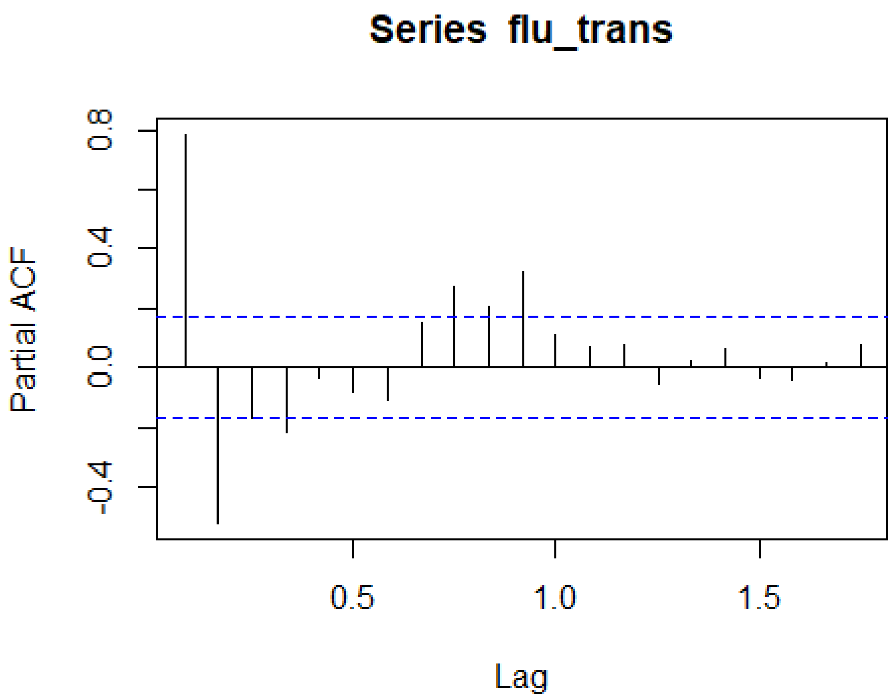
Plot autocorrelation (acf). ACF exhibits a sinosoidal pattern that does not tail off and does not cut off.

```
acf(flu_trans)
```



Plot partial-autocorrelation (pacf). The PACF tails off and does not cut off lag q.

```
pacf(flu_trans)
```



Lets fit six ARIMA models

Model 1

```
fit_1 <- arima(flu_trans,order = c(0,0,1))
summary(fit_1)
```

```
##
## Call:
## arima(x = flu_trans, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.8313    -2.8308
## s.e.  0.0389     0.1073
##
## sigma^2 estimated as 0.4559:  log likelihood = -136.05,  aic = 278.1
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.006499418 0.6752152 0.5082772 -25.26174 38.86777 0.9767744
##                   ACF1
## Training set 0.4442232
```

Model 2

```
fit_2 <- arima(flu_trans,order = c(1,0,0))
summary(fit_2)
```

```
##
## Call:
## arima(x = flu_trans, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.8152    -2.7623
## s.e.  0.0520     0.2888
##
## sigma^2 estimated as 0.398:  log likelihood = -127.04,  aic = 260.08
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.02368845 0.6308658 0.4949679 -17.60237 34.92692 0.9511974
##                  ACF1
## Training set 0.441091
```

Model 3

```
fit_3 <- arima(flu_trans,order = c(1,0,1))
summary(fit_3)
```

```
##
## Call:
## arima(x = flu_trans, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.6924  0.5772    -2.7772
## s.e.  0.0698  0.0677     0.2357
##
## sigma^2 estimated as 0.2884:  log likelihood = -106.11,  aic = 220.21
##
## Training set error measures:
##                       ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.01578081 0.5370485 0.4100738 -11.74388 27.79516 0.7880535
##                    ACF1
## Training set 0.05462355
```

Model 4

```
fit_4 <- arima(flu_trans,order = c(0,0,2))
summary(fit_4)
```

```
##
## Call:
## arima(x = flu_trans, order = c(0, 0, 2))
##
## Coefficients:
##          ma1     ma2  intercept
##       1.2185  0.5563    -2.8142
## s.e.  0.0689  0.0685     0.1351
##
## sigma^2 estimated as 0.3166:  log likelihood = -112.24,  aic = 232.48
##
## Training set error measures:
##                        ME      RMSE       MAE      MPE    MAPE      MASE
## Training set -0.009388603 0.5626927 0.4284391 -16.3543 29.1239 0.8233467
##                   ACF1
## Training set 0.1048284
```

Model 5

```
fit_5 <- arima(flu_trans,order = c(1,0,2))
summary(fit_5)
```

```
##
## Call:
## arima(x = flu_trans, order = c(1, 0, 2))
##
## Coefficients:
##          ar1     ma1     ma2  intercept
##       0.5987  0.7216  0.1873    -2.7851
## s.e.  0.0991  0.1141  0.0908     0.2159
##
## sigma^2 estimated as 0.2793:  log likelihood = -104.01,  aic = 218.01
##
## Training set error measures:
##                       ME      RMSE       MAE      MPE     MAPE     MASE
## Training set -0.01378233 0.5284641 0.4028238 -11.16897 27.23359 0.774121
##                     ACF1
## Training set -0.01965572
```

Model 6

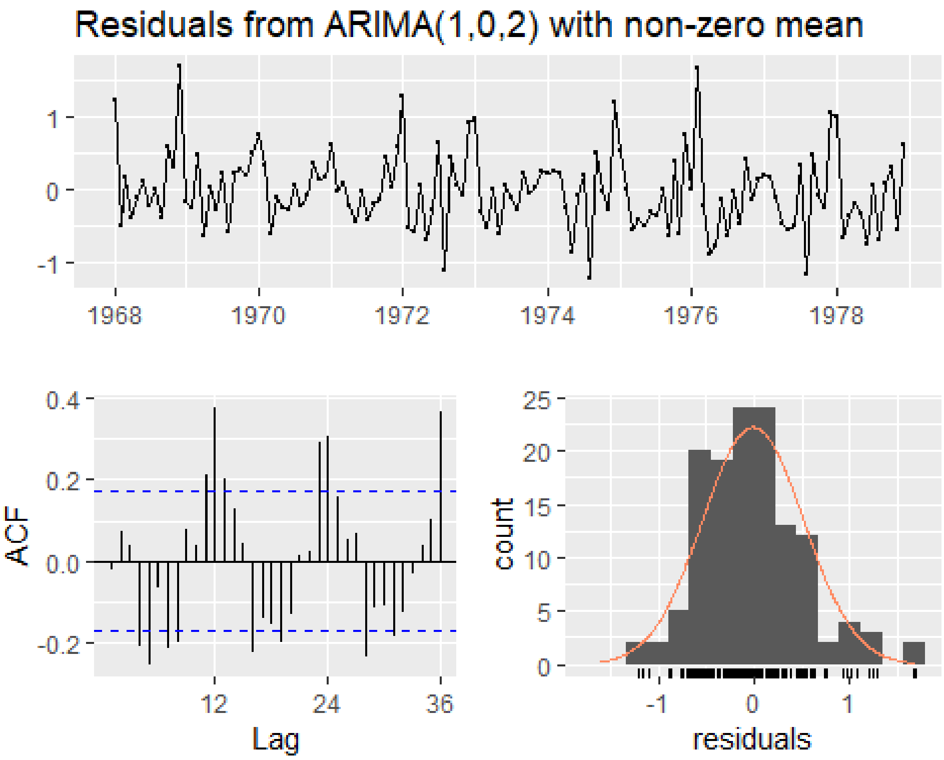
```
fit_6 <- arima(flu_trans,order = c(1,1,1))
summary(fit_6)
```

```
##
## Call:
## arima(x = flu_trans, order = c(1, 1, 1))
##
## Coefficients:
##          ar1     ma1
##       0.1341  0.4163
## s.e.  0.1713  0.1590
##
## sigma^2 estimated as 0.3335:  log likelihood = -114.12,  aic = 234.24
##
## Training set error measures:
##                        ME      RMSE      MAE       MPE     MAPE      MASE
## Training set -0.009858947 0.5753296 0.446408 -1.586504 27.61737 0.8578782
##                           ACF1
## Training set -0.004330189
```

Model 5 performed best with lowest AIC of 218.01 and RMSE of 0.5285.

We will check residuals for Model 3 to see if the residuals look like white noise.

```
checkresiduals(fit_5)
```



```
## 
##  Ljung-Box test
## 
## data:  Residuals from ARIMA(1,0,2) with non-zero mean
## Q* = 117.97, df = 20, p-value = 6.661e-16
## 
## Model df: 4.   Total lags used: 24
```

There is still signifcant autocorrelation in the residuals and the Ljung-Box test p<0.05 show the residuals do NOT appear to be white noise. The model needs to be improved.

I like to use the auto arima function from the R fpp2 library.

```
library(fpp2)
```

Lets fit an ARIMA model using this library.

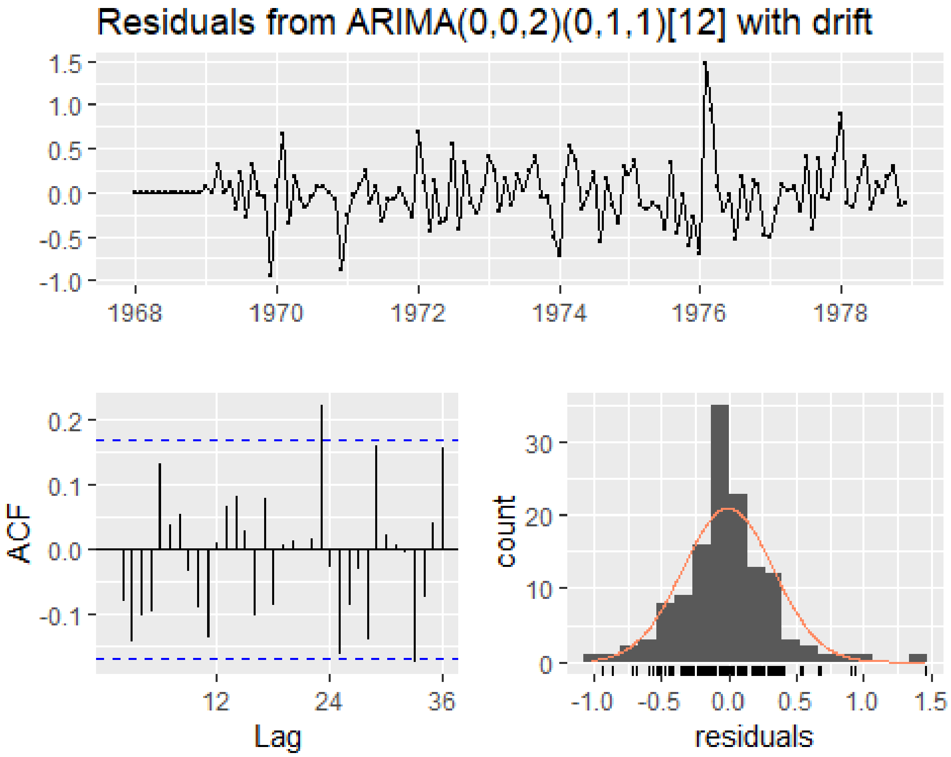
```
fit_auto <- auto.arima(flu_trans)
summary(fit_auto)
```

```
## Series: flu_trans
## ARIMA(0,0,2)(0,1,1)[12] with drift
##
## Coefficients:
##          ma1     ma2     sma1    drift
##       0.7054  0.1507  -0.8832  -0.0090
## s.e.  0.0986  0.0958   0.1449   0.0015
##
## sigma^2 estimated as 0.1271:  log likelihood=-53.43
## AIC=116.87   AICc=117.4   BIC=130.81
##
## Training set error measures:
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set -0.008778798 0.3342714 0.2343301 -4.651256 13.22999 0.5702182
##                     ACF1
## Training set 0.001013807
```

The model is significantly improved with an AIC of 116.87 and RMSE of 0.3342.

Lets check the residuals.

```
checkresiduals(fit_auto)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,2)(0,1,1)[12] with drift
## Q* = 27.648, df = 20, p-value = 0.118
##
## Model df: 4.   Total lags used: 24
```

The residuals appear to pass the white noise test. The residuals do not have significant autocorrelations (except at lag-23) and the Ljung-Box test p-value is 0.118; above the alpha level of 0.05.