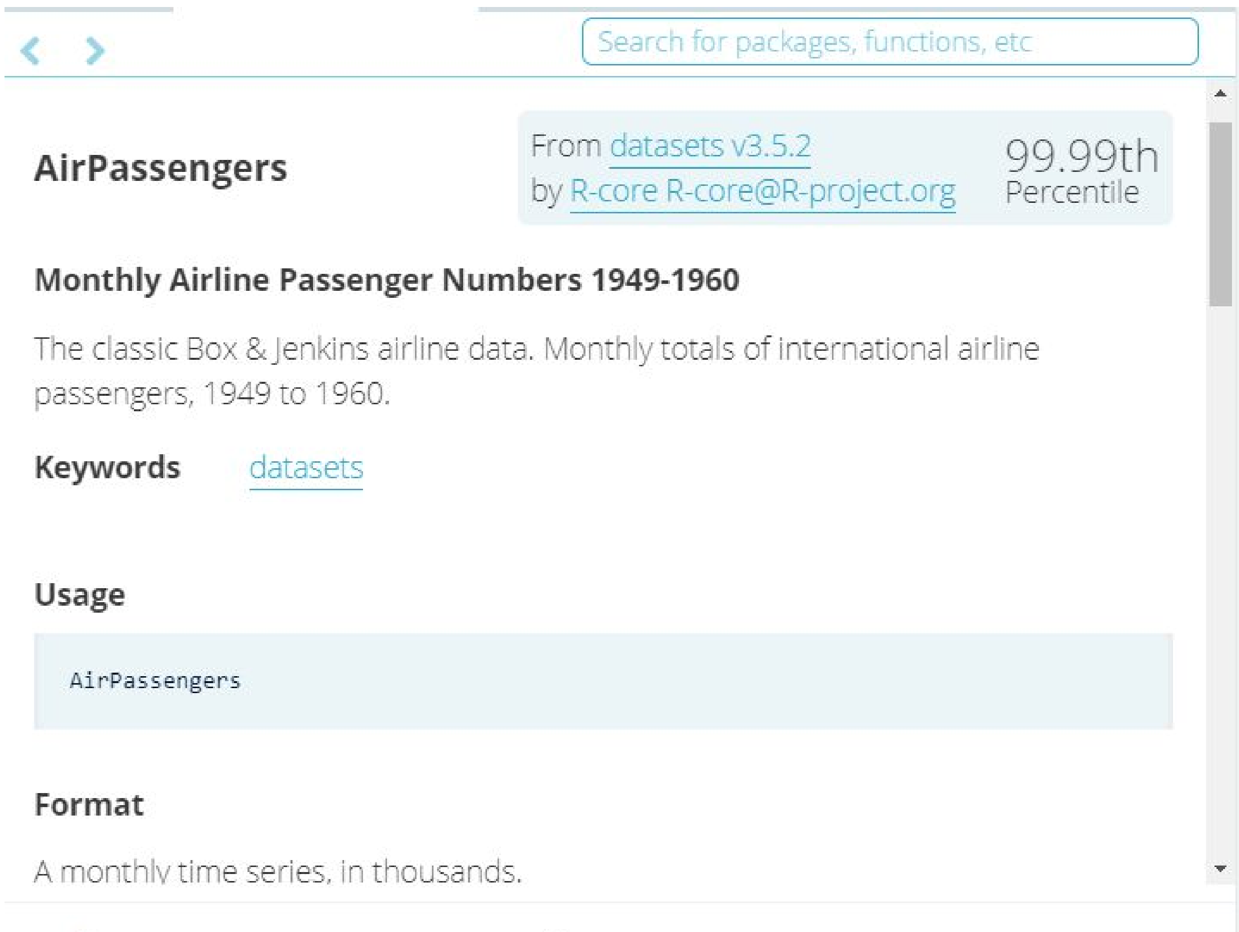
Data Play

In the video, you saw various types of data. In this exercise, you will plot additional time series data and compare them to what you saw in the video. It is useful to think about how these time series compare to the series in the video. In particular, concentrate on the type of trend, seasonality or periodicity, and homoscedasticity.
Before you use a data set for the first time, you should use the help system to see the details of the data. For example, use `help(AirPassengers)` or `?AirPassengers` to see the details of the series.

```
# View a detailed description of AirPassengers
help(AirPassengers)
```



Search for packages, functions, etc

AirPassengers

From datasets v3.5.2
by R-core R-core@R-project.org

99.99th
Percentile

Monthly Airline Passenger Numbers 1949-1960

The classic Box & Jenkins airline data. Monthly totals of international airline passengers, 1949 to 1960.

Keywords datasets

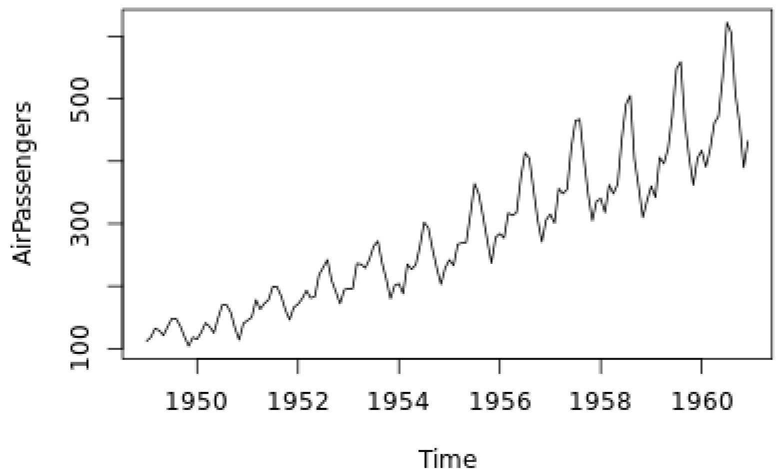
Usage

```
AirPassengers
```

Format

A monthly time series, in thousands.

```
# Plot AirPassengers
plot(AirPassengers)
```

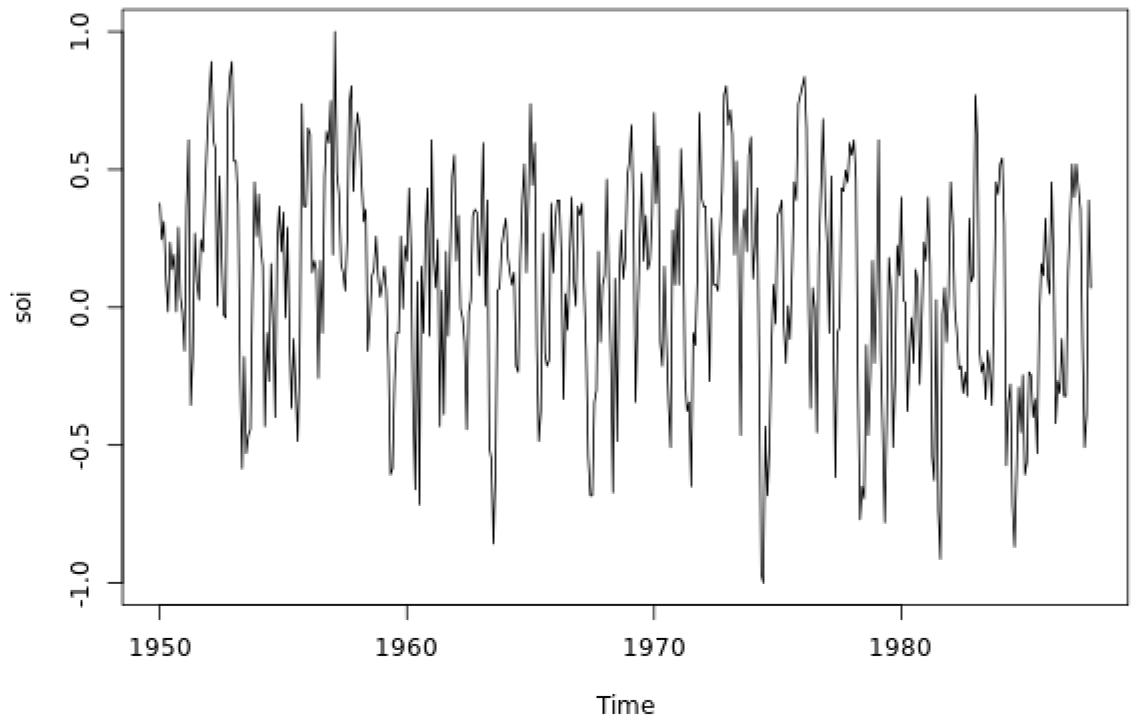


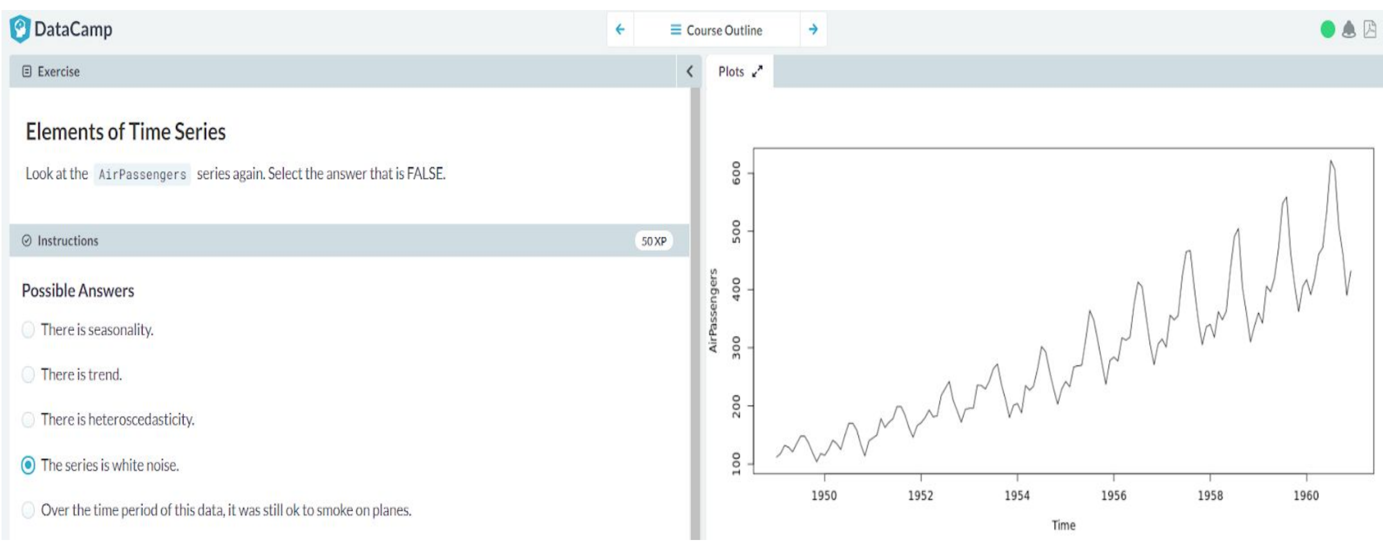
# Plot the DJIA daily closings
head(djia)

```
> head(djia)
               Open     High      Low    Close    Volume
2006-04-20 11278.53 11384.11 11275.05 11342.89 336420000
2006-04-21 11343.45 11405.88 11316.79 11347.45 325090000
2006-04-24 11346.81 11359.70 11305.83 11336.32 232000000
2006-04-25 11336.56 11355.37 11260.84 11283.25 289230000
2006-04-26 11283.25 11379.87 11282.77 11354.49 270270000
2006-04-27 11349.53 11416.93 11275.30 11382.51 361740000
>
```

plot(djia$Close)


```
# Plot the Southern Oscillation Index
plot(soi)
```


soi
1.0
0.5
0.0
-0.5
-1.0
1950
1960
1970
1980
Time


DataCamp
Course Outline
Exercise
Plots
Elements of Time Series
Look at the AirPassengers series again. Select the answer that is FALSE.
Instructions
50 XP
Possible Answers
There is seasonality.
There is trend.
There is heteroscedasticity.
The series is white noise.
Over the time period of this data, it was still ok to smoke on planes.
AirPassengers
600
500
400
300
200
100
1950
1952
1954
1956
1958
1960
Time

Exercise

Differencing

As seen in the video, when a time series is *trend stationary*, it will have stationary behavior around a trend. A simple example is $Y_t = \alpha + \beta t + X_t$ where X_t is stationary.

A different type of model for trend is *random walk*, which has the form $X_t = X_{t-1} + W_t$, where W_t is white noise. It is called a random walk because at time t the process is where it was at time $t - 1$ plus a completely random movement. For a *random walk with drift*, a constant is added to the model and will cause the random walk to drift in the direction (positive or negative) of the drift.

We simulated and plotted data from these models. Note the difference in the behavior of the two models.

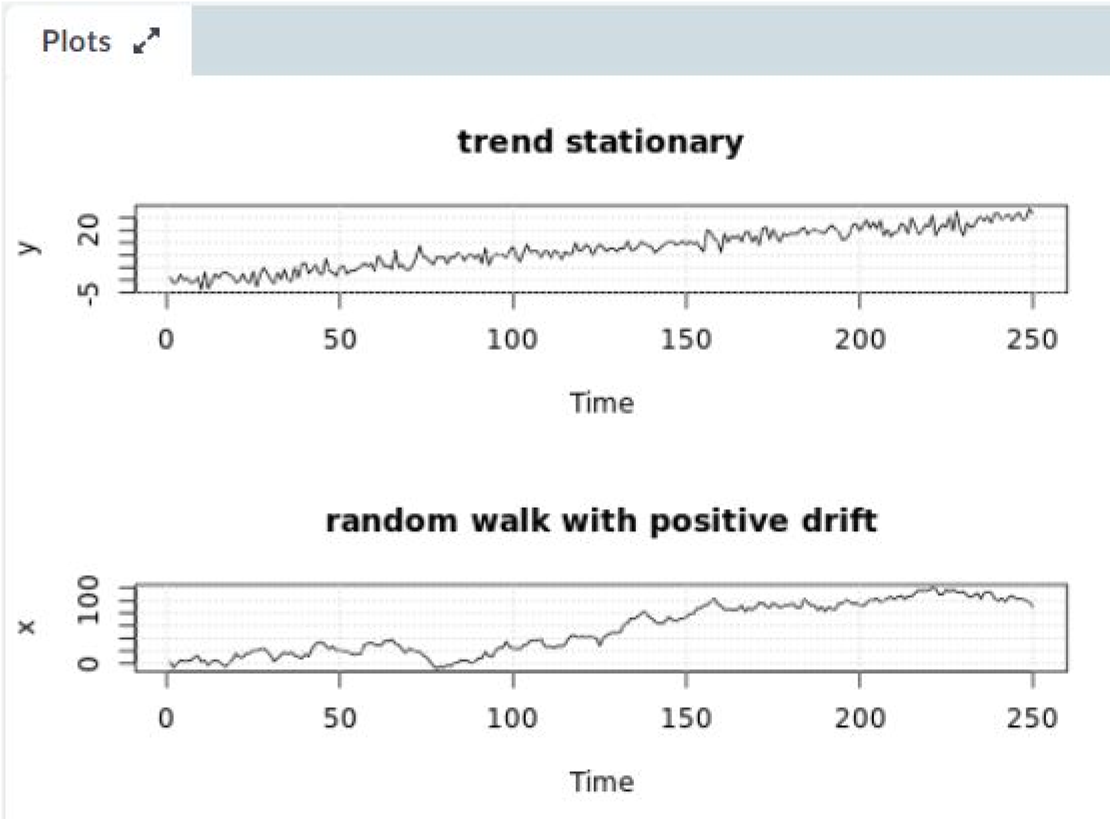
In both cases, simple *differencing* can remove the trend and coerce the data to stationarity. Differencing looks at the difference between the value of a time series at a certain point in time and its preceding value. That is, $X_t - X_{t-1}$ is computed.

To check that it works, you will difference each generated time series and plot the detrended series. If a time series is in `x`, then `diff(x)` will have the detrended series obtained by differencing the data. To plot the detrended series, simply use `plot(diff(x))`.

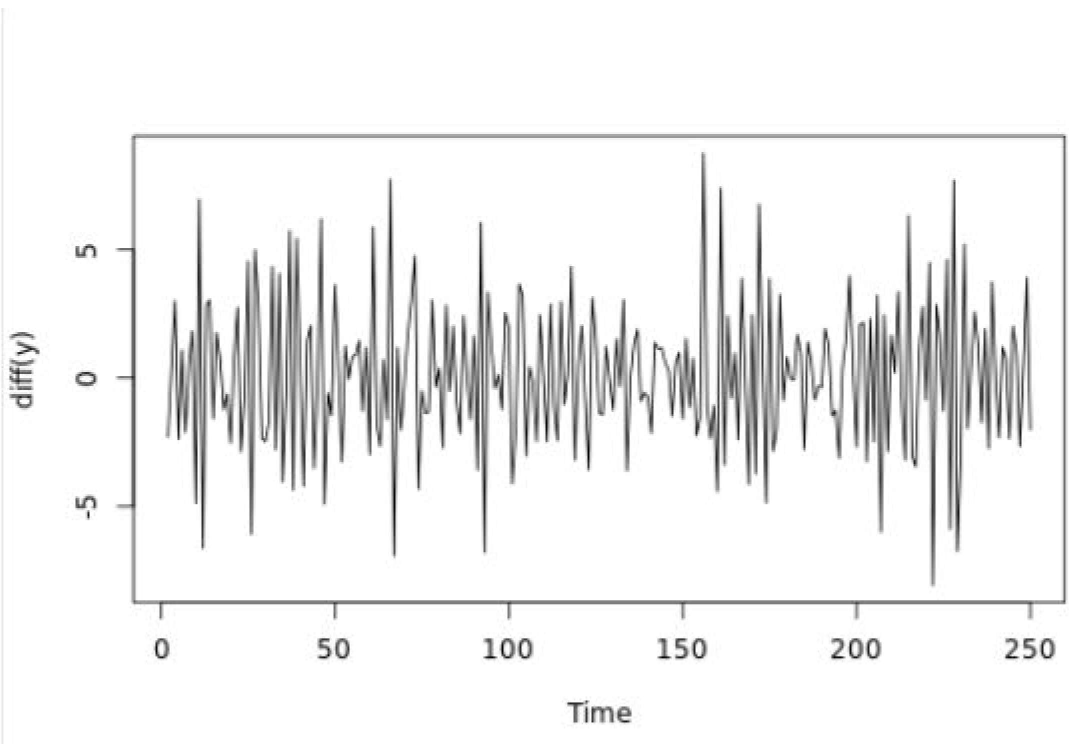
Instructions 100 XP

- In one line, difference and plot the detrended trend stationary data in `y` by nesting a call to `diff()` within a call to `plot()`. Does the result look stationary?
- Do the same for `x`. Does the result look stationary?

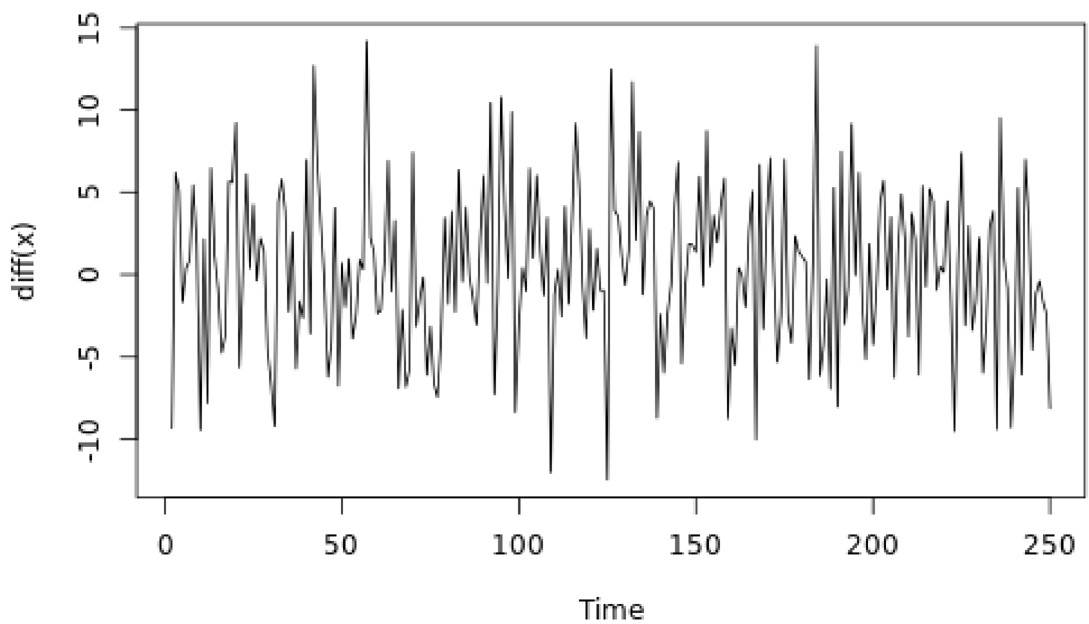
GIVEN:



# Plot detrended y (trend stationary)
plot(diff(y))



```
# Plot detrended x (random walk)
plot(diff(x))
```

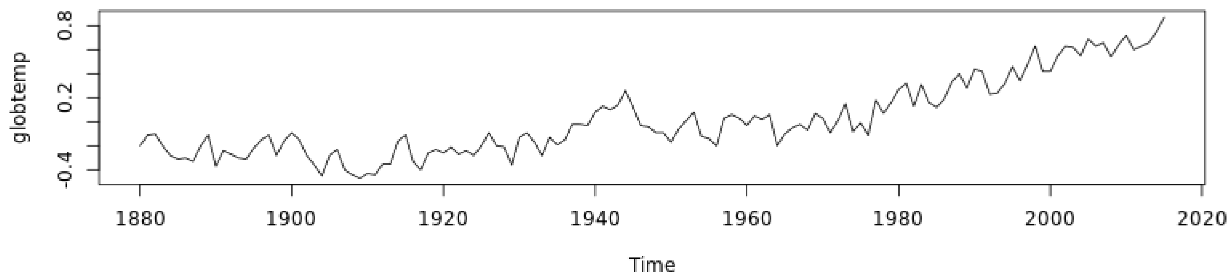


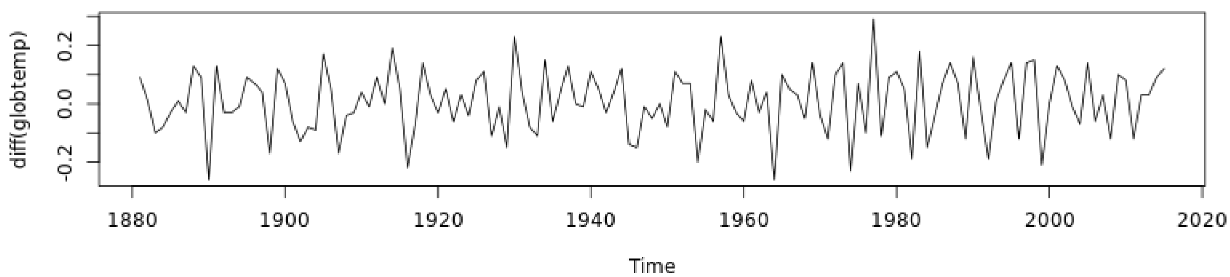
Detrending Data

As you have seen in the previous exercise, differencing is generally good for removing trend from time series data. Recall that differencing looks at the difference between the value of a time series at a certain point in time and its preceding value.
In this exercise, you will use differencing `diff()` to detrend and plot real time series data.

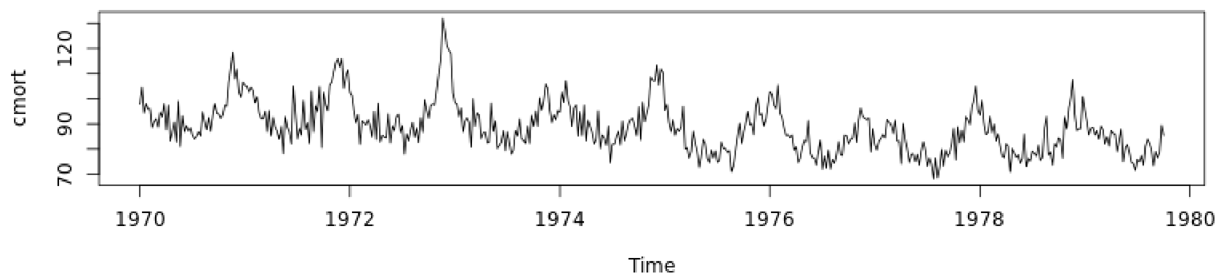
- The package **astsa** is preloaded.
- Generate a multifigure plot comparing the global temperature data (`globtemp`) with the detrended series. You can create a multifigure plot by running the pre-written `par()` command followed by two separate calls to `plot()`.
- Generate another multifigure plot comparing the weekly cardiovascular mortality in Los Angeles County (`cmort`) with the detrended series.

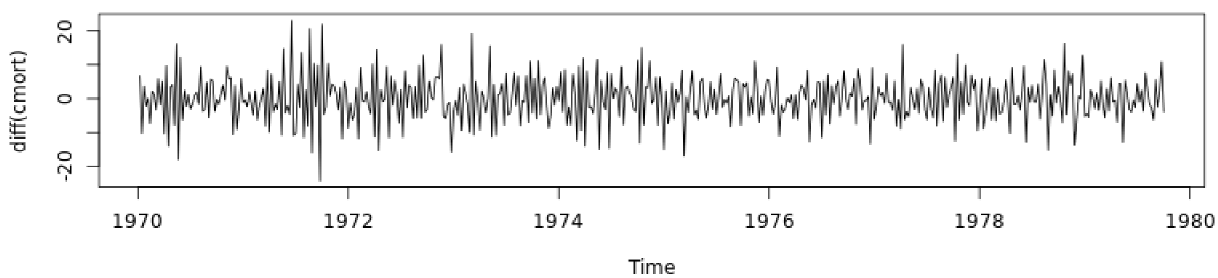
```
# Plot globtemp and detrended globtemp
par(mfrow = c(2,1))
plot(globtemp)
plot(diff(globtemp))
```






```
# Plot cmort and detrended cmort
par(mfrow = c(2,1))
plot(cmort)
plot(diff(cmort))
```


cmort
70
90
120
1970
1972
1974
1976
1978
1980
Time


diff(cmort)
-20
0
20
1970
1972
1974
1976
1978
1980
Time

Exercise

Dealing with Trend and Heteroscedasticity

Here, we will coerce nonstationary data to stationarity by calculating the return or growth rate as follows.

Often time series are generated as

$$X_t = (1 + p_t)X_{t-1}$$

meaning that the value of the time series observed at time t equals the value observed at time $t-1$ and a small percent change p_t at time t.

A simple deterministic example is putting money into a bank with a fixed interest p. In this case, X_t is the value of the account at time period t with an initial deposit of X_0.

Typically, p_t is referred to as the *return* or *growth rate* of a time series, and this process is often stable.

For reasons that are outside the scope of this course, it can be shown that the growth rate p_t can be approximated by

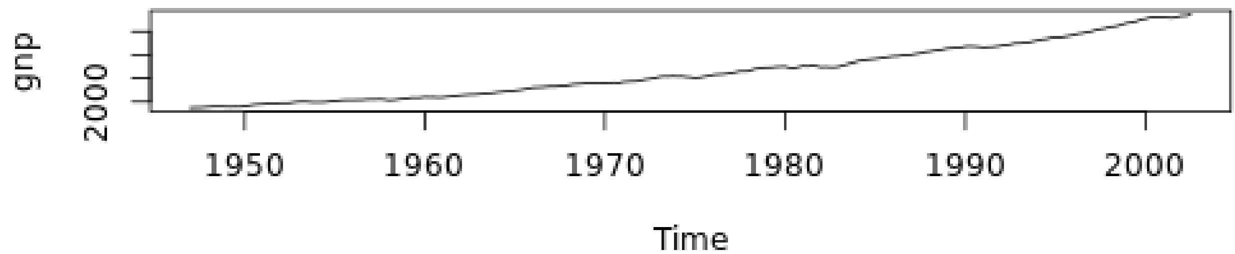
$$Y_t = \log X_t - \log X_{t-1} \approx p_t.$$

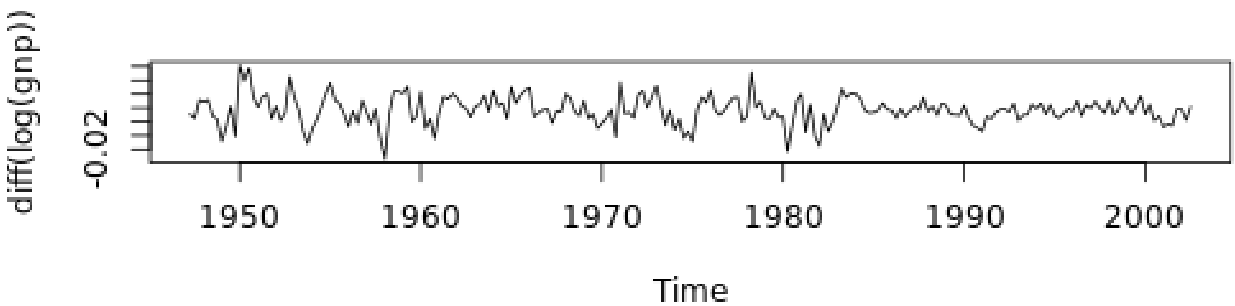
In R, p_t is often calculated as `diff(log(x))` and plotting it can be done in one line `plot(diff(log(x)))`.

-
- Generate a multifigure plot to (1) plot the quarterly US GNP (`gnp`) data and notice it is not stationary, and (2) plot the approximate growth rate of the US GNP using `diff()`and `log()`.

As before, the packages [astsa]() and [xts]() are preloaded.

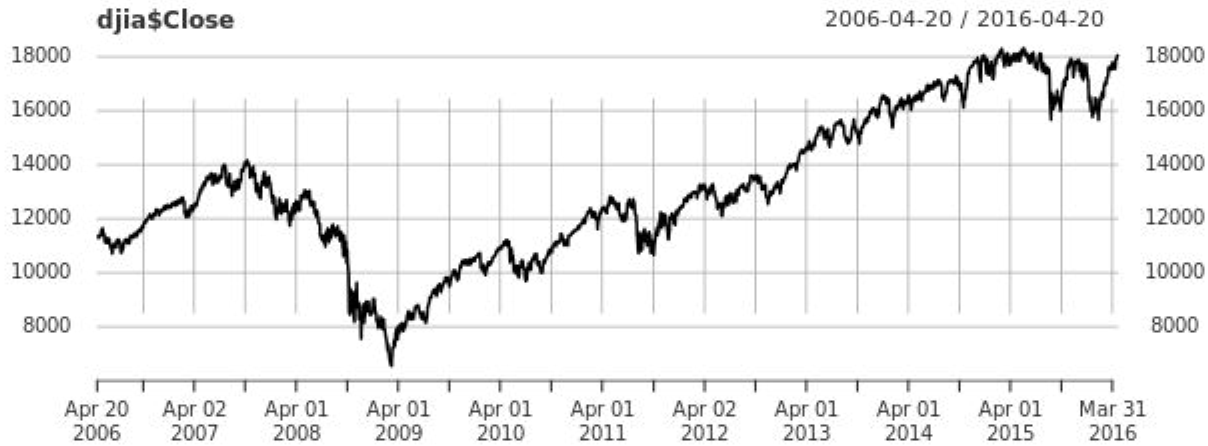
```
# astsa and xts are preloaded
# Plot GNP series (gnp) and its growth rate
par(mfrow = c(2,1))
plot(gnp)
plot(diff(log(gnp)))
```

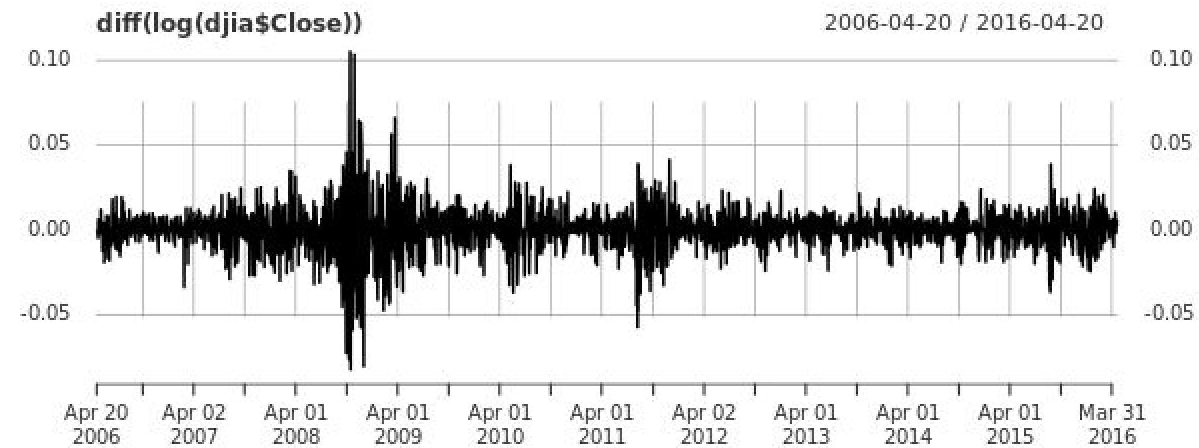




- Use a multifigure plot to (1) plot the daily DJIA closings (`djia$Close`) and notice that it is not stationary. The data are an `xts` object. Then (2) plot the approximate DJIA returns using `diff()` and `log()`. How does this compare to the growth rate of the GNP?

```
# Plot DJIA closings (djia$Close) and its returns
par(mfrow = c(2,1))
plot(diff(log(djia$Close)))
```





Simulating ARMA Models

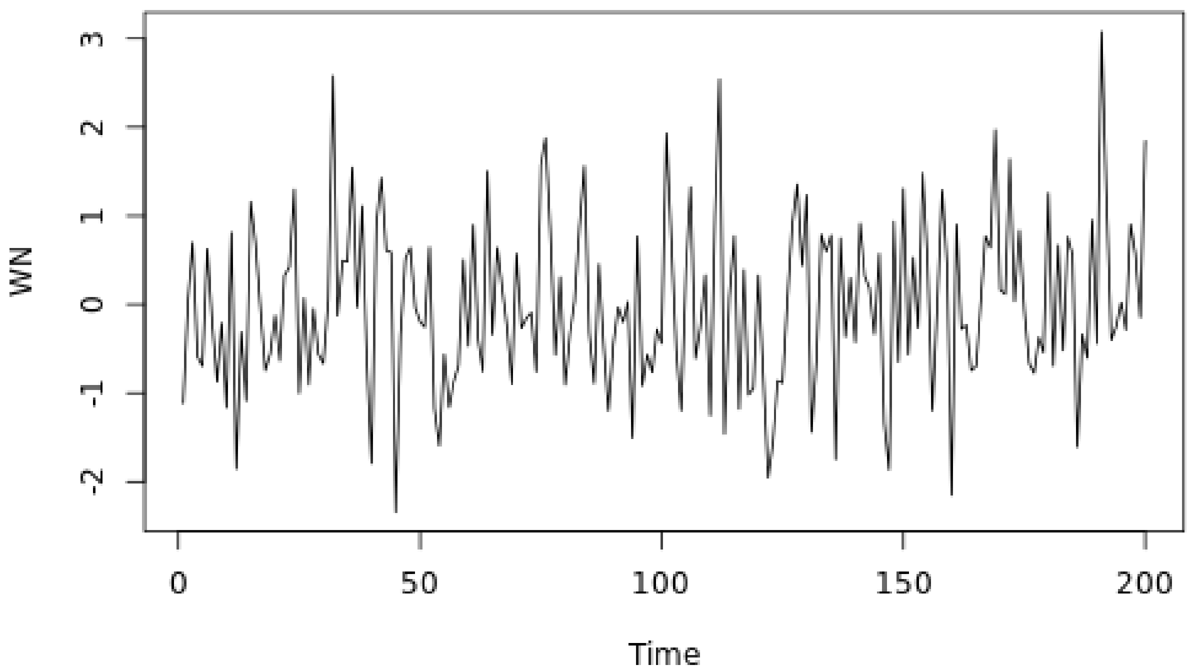
As we saw in the video, any stationary time series can be written as a linear combination of white noise. In addition, any ARMA model has this form, so it is a good choice for modeling stationary time series.

R provides a simple function called `arima.sim()` to generate data from an ARMA model. For example, the syntax for generating 100 observations from an MA(1) with parameter 0.9 is `arima.sim(model = list(order = c(0, 0, 1), ma = .9 ), n = 100)`. You can also use `order = c(0, 0, 0)` to generate white noise.

In this exercise, you will generate data from various ARMA models. For each command, generate **200** observations and plot the result.

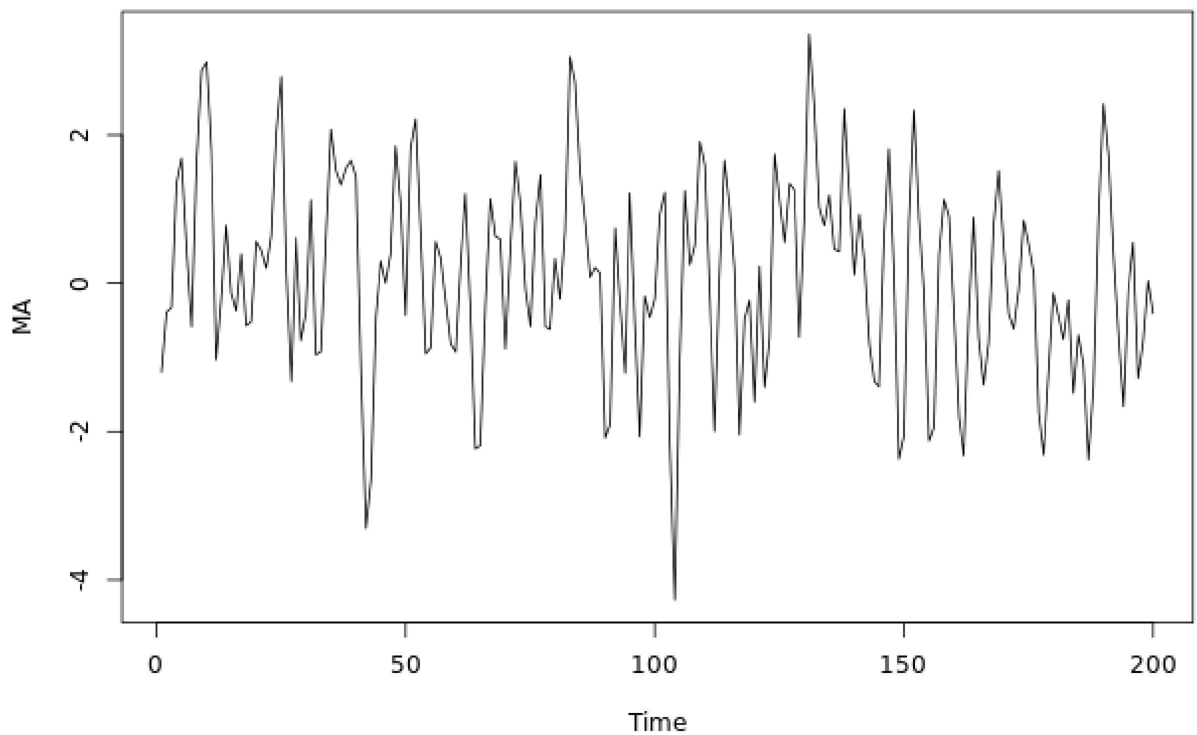
\# Generate and plot white noise

```
WN <- arima.sim(model = list(order = c(0, 0, 0)), n = 200)
plot(WN)
```

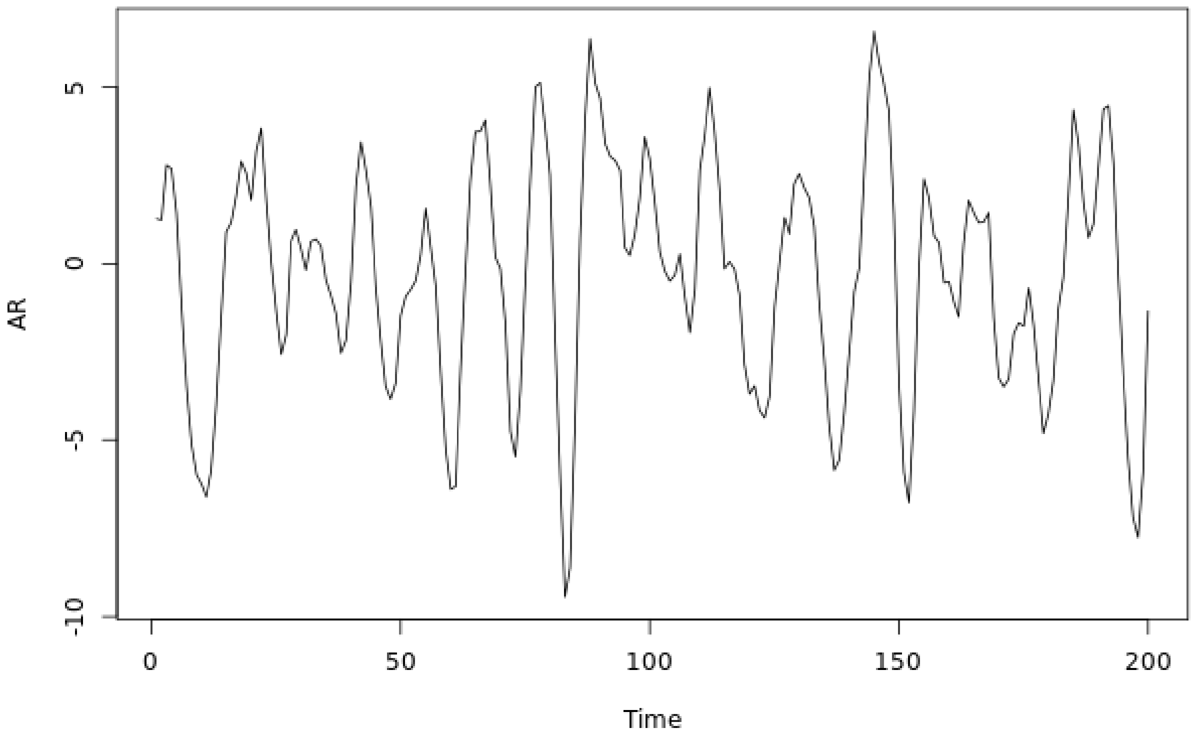


\# Generate and plot an MA(1) with parameter .9 by filtering the noise

```
MA <- arima.sim(model = list(order = c(0, 0, 1), ma = .9), n = 200)
plot(MA)
```




```
# Generate and plot an AR(1) with parameters 1.5 and -.75
AR <- arima.sim(model = list(order = c(2, 0, 0), ar = c(1.5, -.75)), n = 200)
plot(AR)
```



AR
5
0
-5
-10
0
50
100
150
200
Time

Fitting an AR(1) Model

Recall that you use the ACF and PACF pair to help identify the orders p and q of an ARMA model. The following table is a summary of the results:

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

In this exercise, you will generate data from the AR(1) model,

$$X_t = .9X_{t-1} + W_t,$$

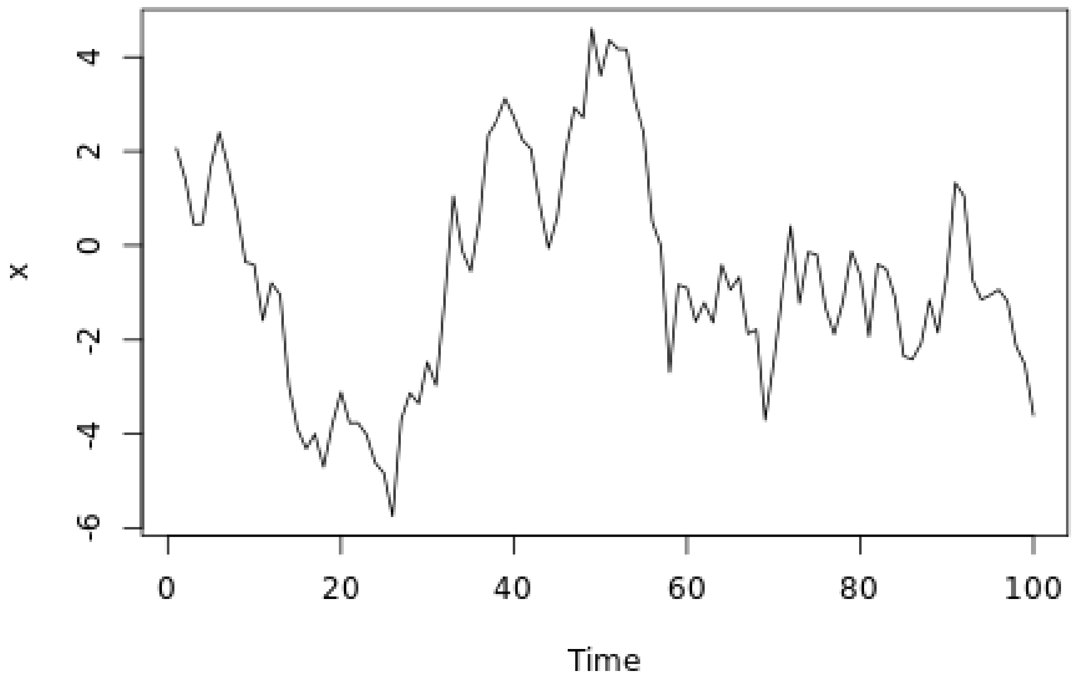
look at the simulated data and the sample ACF and PACF pair to determine the order. Then, you will fit the model and compare the estimated parameters to the true parameters.

Throughout this course, you will be using `sarima()` from the `astsa` package to easily fit models to data. The command produces a residual diagnostic graphic that can be ignored until diagnostics is discussed later in the chapter.

- Use the prewritten `arima.sim()` command to generate 100 observations from an AR(1) model with AR parameter .9. Save this to `x`.
- Plot the generated data using `plot()`.
- Use `sarima()` from `astsa` to fit an AR(1) to the previously generated data. Examine the t-table and compare the estimates to the true values. For example, if the time series is in `x`, to fit an AR(1) to the data, use `sarima(x, p = 1, d = 0, q = 0)` or simply `sarima(x, 1, 0, 0)`.

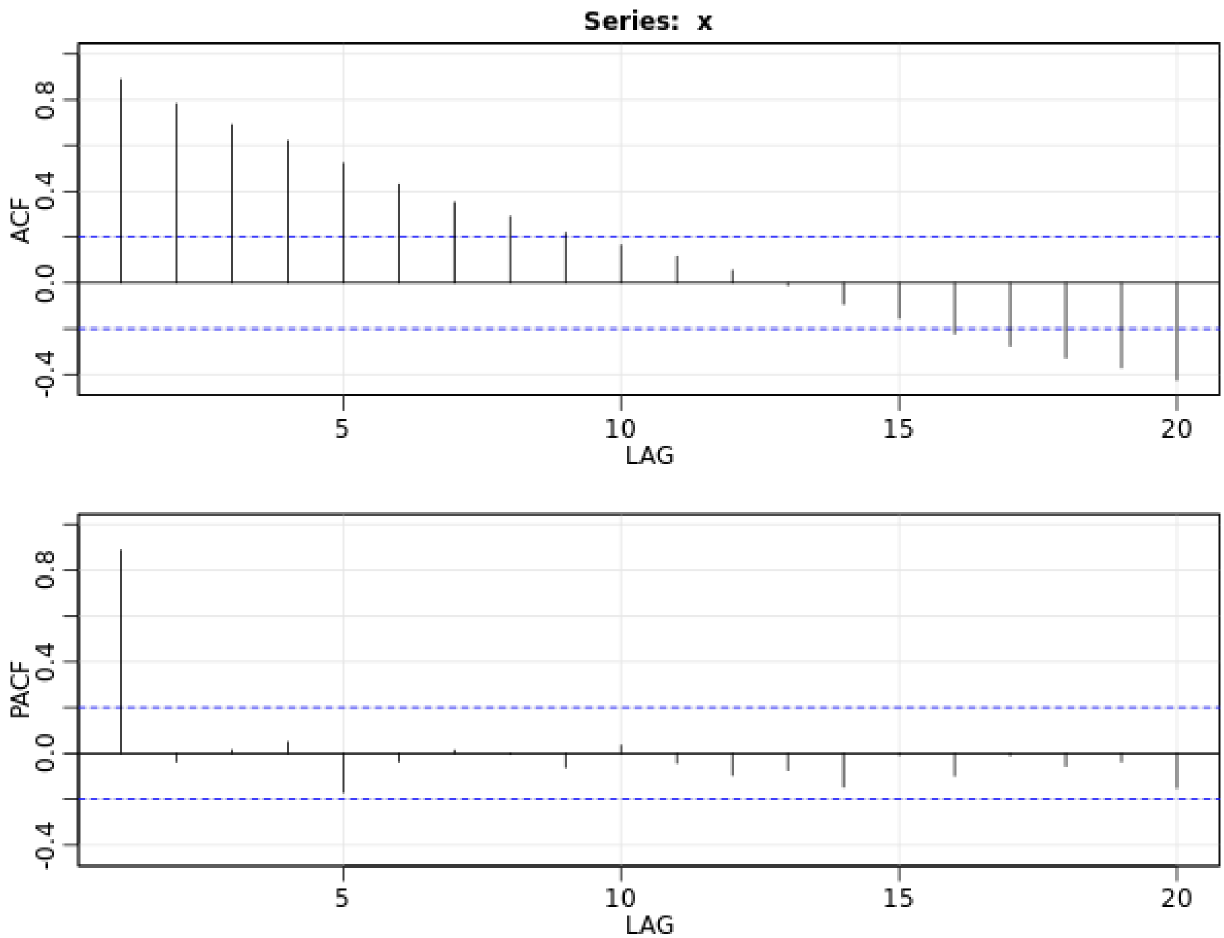
```
#The package astsa is preloaded.
library(astsa)
#Use the prewritten arima.sim() command to generate 100 observations from an AR(1) model with AR parameter .9. Save this to x.
x <- arima.sim(model = list(order = c(1, 0, 0), ar = .9), n = 100)
# Plot the generated data
```

plot(x)



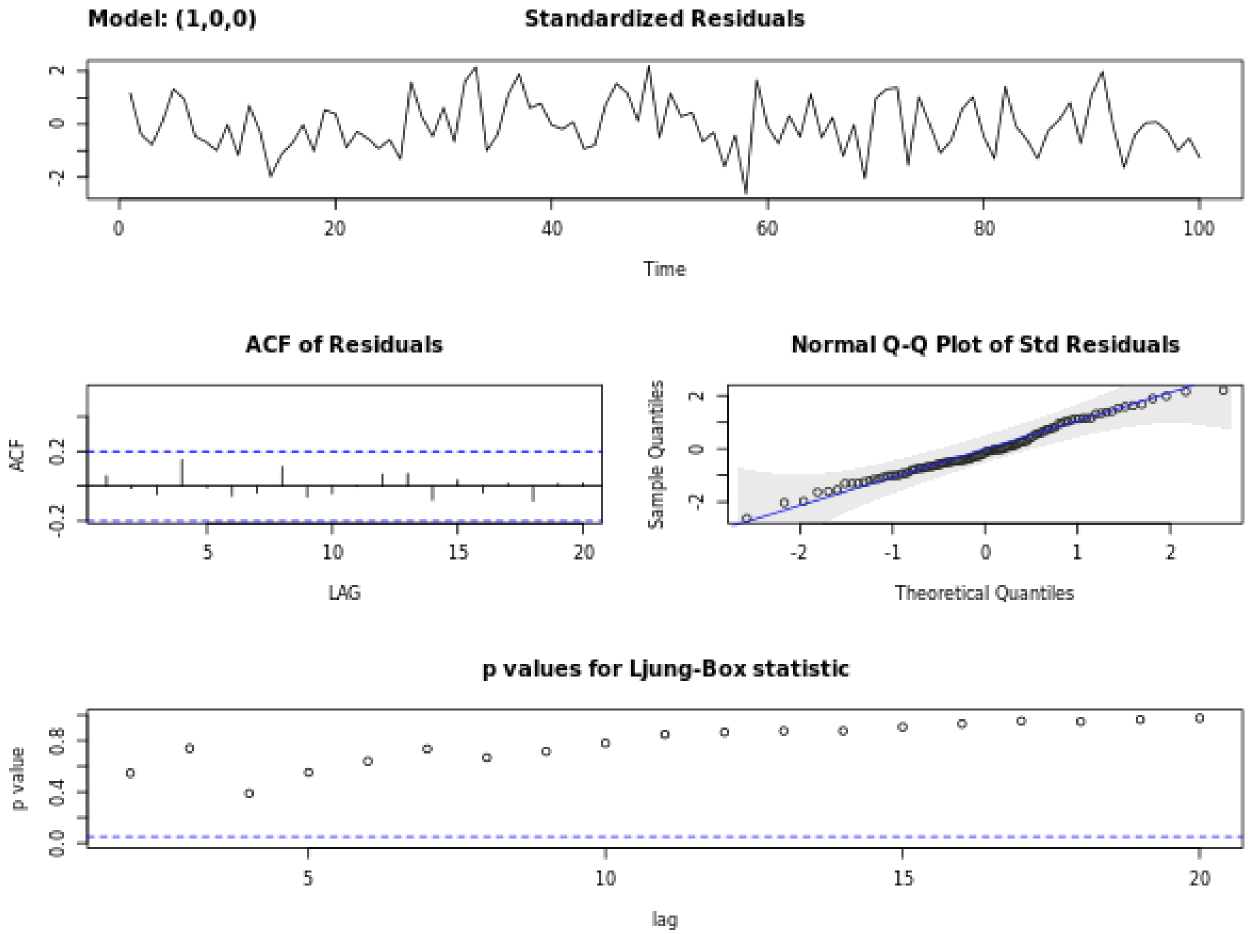
#Plot the sample ACF and PACF pairs using the `acf2()` command from the `astsa`package.

acf2(x)




```
# Fit an AR(1) to the data and examine the t-table
sarima(x, p = 1, d = 0, q = 0)
```

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, 
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc, 
    REPORT = 1, reltol = tol))

Coefficients:
         ar1    xmean
      0.8621  -2.0871
s.e.  0.0480   0.6215

sigma^2 estimated as 0.8258:  log likelihood = -133,  aic = 272.01

$degrees_of_freedom
[1] 98

$ttable
      Estimate     SE t.value p.value
ar1     0.8621 0.0480 17.9568  0.0000
xmean  -2.0871 0.6215 -3.3581  0.0011

$AIC
[1] 0.8486141

$AICc
[1] 0.8711141

$BIC
[1] -0.09928246
```


Model: (1,0,0)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

Fitting an AR(2) Model

For this exercise, we generated data from the AR(2) model,

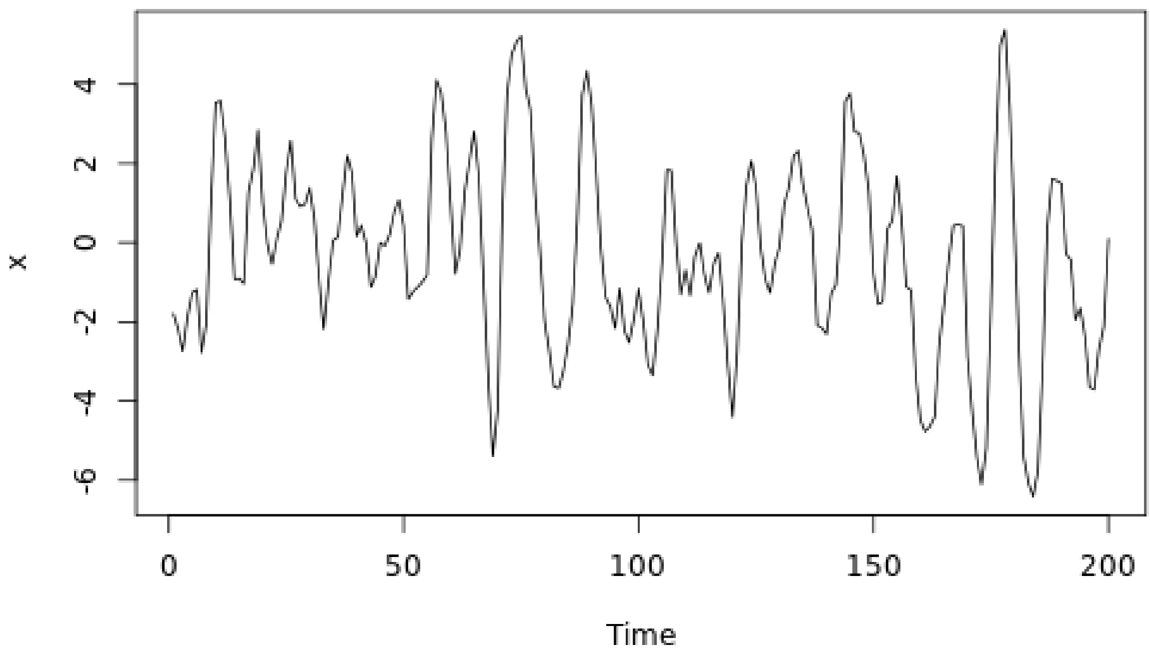
$$X_t = 1.5X_{t-1} - .75X_{t-2} + W_t,$$

using `x <- arima.sim(model = list(order = c(2, 0, 0), ar = c(1.5, -.75)), n = 200)`. Look at the simulated data and the sample ACF and PACF pair to determine the model order. Then fit the model and compare the estimated parameters to the true parameters.

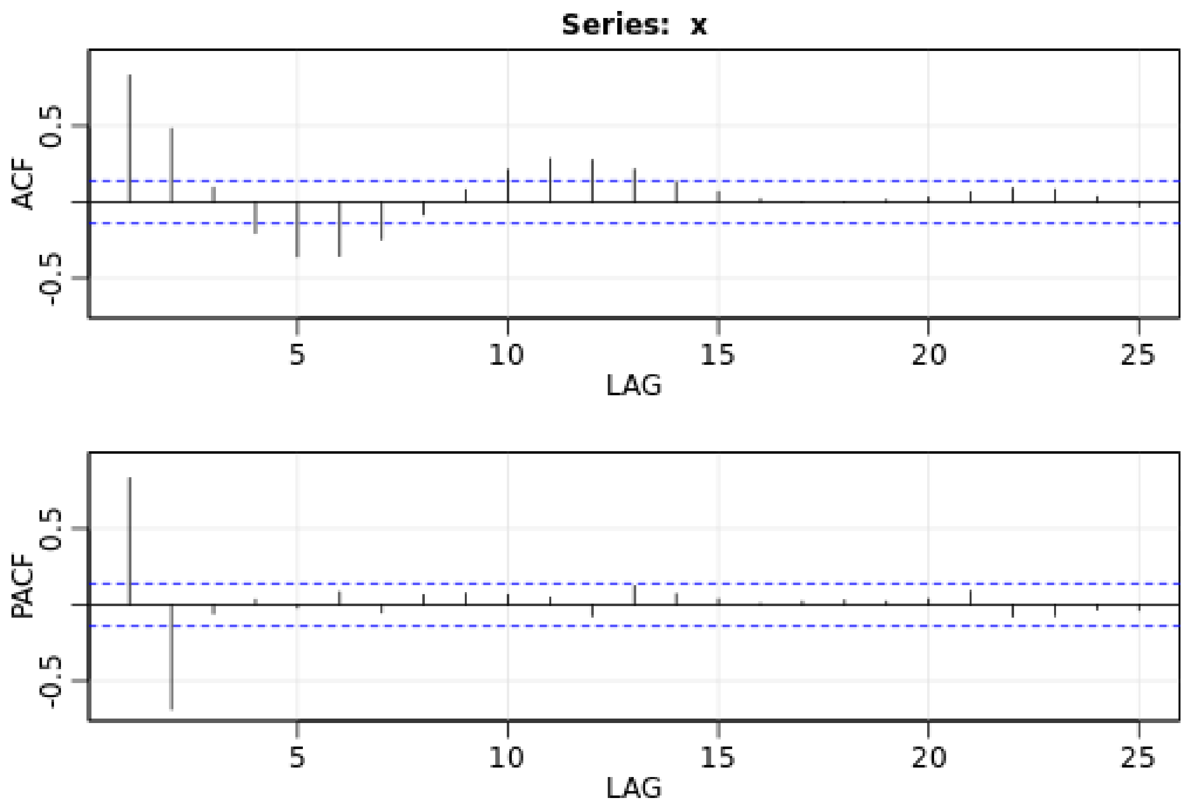
Instructions 0 XP

- The package astsa is preloaded. `x` contains the 200 AR(2) observations.
- Use `plot()` to plot the generated data in `x`.
- Plot the sample ACF and PACF pair using `acf2()` from the `astsa` package.
- Use `sarima()` to fit an AR(2) to the previously generated data in `x`. Examine the t-table and compare the estimates to the true values.

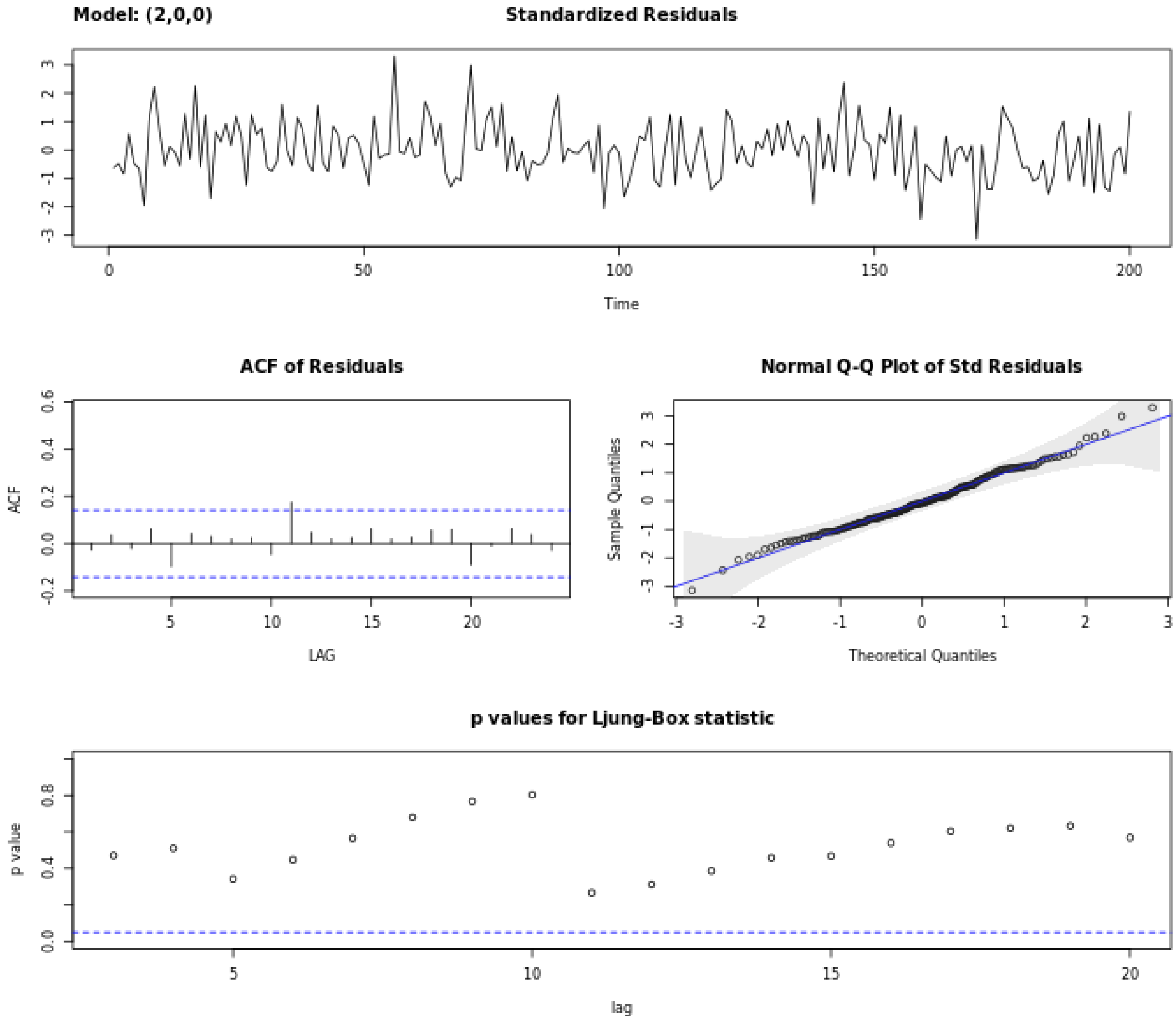
```
# astsa is preloaded
library(astsa)
plot(x)
```




```
# Plot the sample P/ACF of x
acf2(x)
```



# Fit an AR(2) to the data and examine the t-table
sarima(x, p = 2, d = 0, q = 0)



Fitting an MA(1) Model

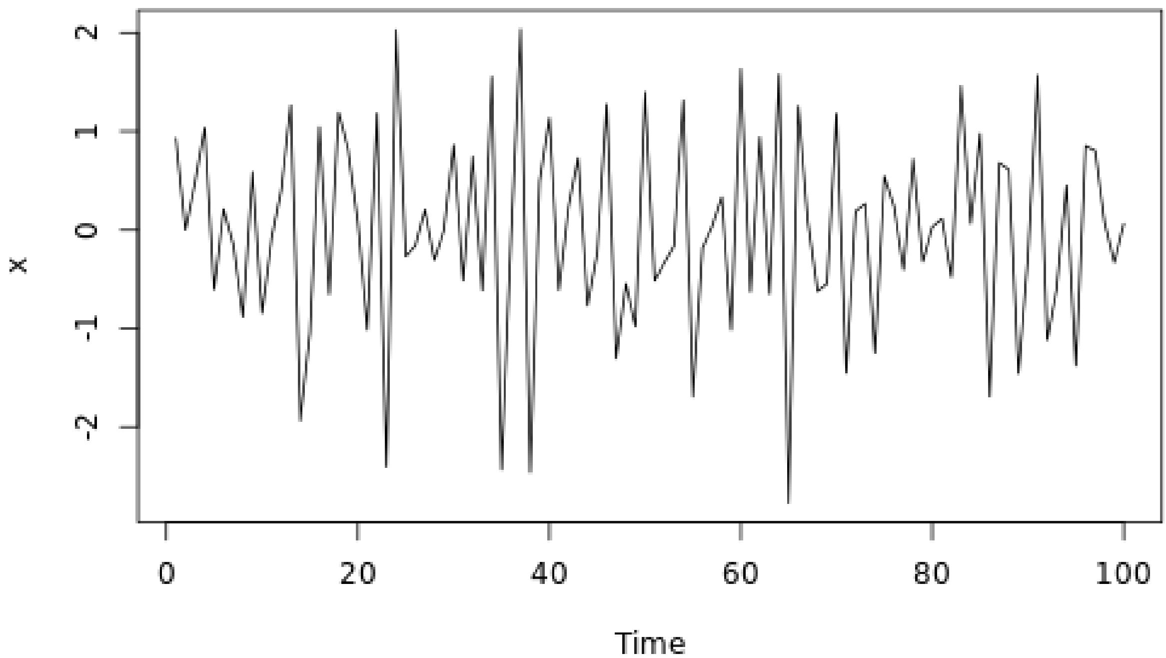
In this exercise, we generated data from an MA(1) model,

$$X_t = W_t - .8W_{t-1},$$

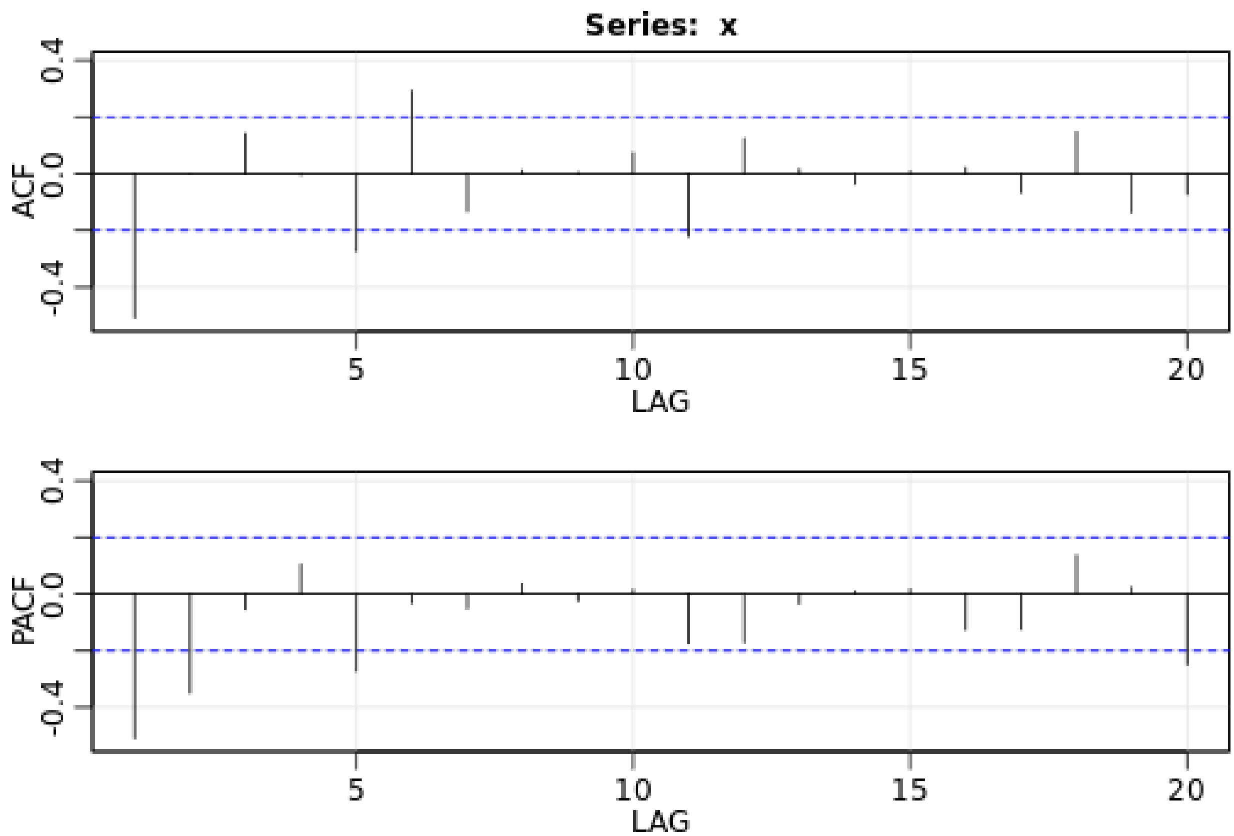
`x <- arima.sim(model = list(order = c(0, 0, 1), ma = -.8), n = 100)` . Look at the simulated data and the sample ACF and PACF to determine the order based on the table given in the first exercise. Then fit the model.

Recall that for pure MA(q) models, the theoretical ACF will cut off at lag q while the PACF will tail off.

```r
# astsa is preloaded. 100 MA(1) observations are available in your workspace as x.
library(astsa)
plot(x)
```

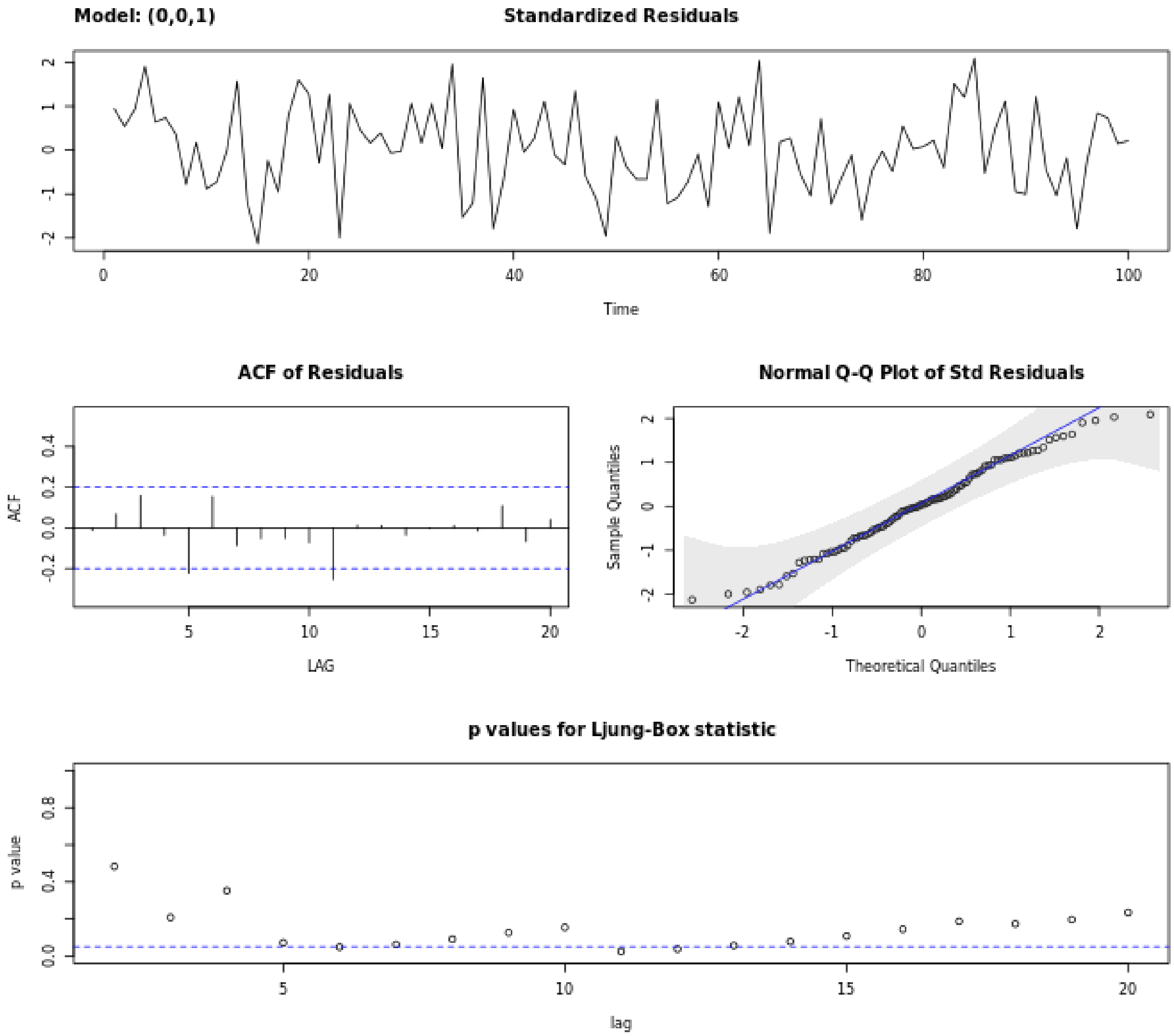



```r
# Plot the sample P/ACF of x
acf2(x)
```



Fit an MA(1) to the data and examine the t-table.Use `sarima()` from `astsa` to fit an MA(1) to the previously generated data. Examine the t-table and compare the estimates to the true values.

sarima(x, p = 0, d = 0, q = 1)


Model: (0,0,1)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

Fitting an ARMA model

You are now ready to merge the AR model and the MA model into the ARMA model. We generated data from the ARMA(2,1) model,

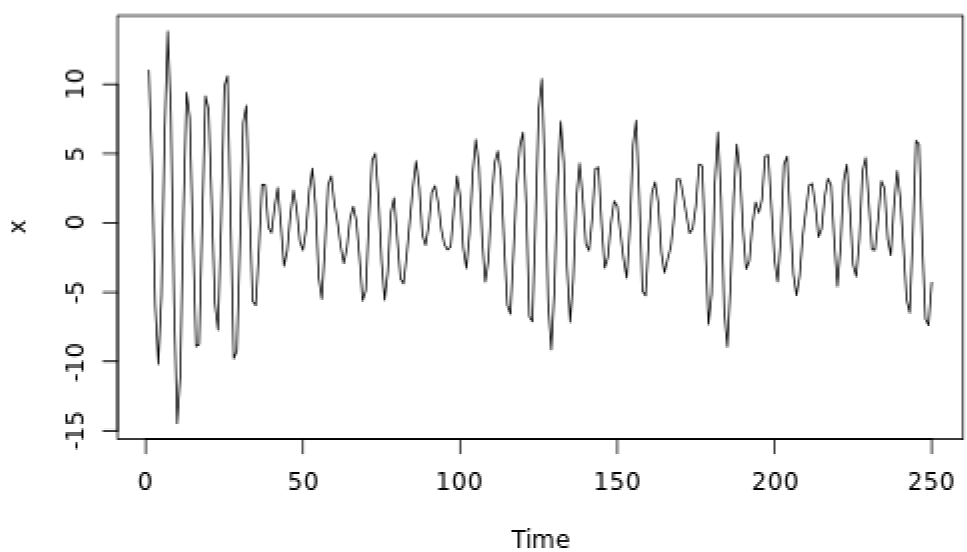
$$X_t = X_{t-1} - .9X_{t-2} + W_t + .8W_{t-1},$$

```
x <- arima.sim(model = list(order = c(2, 0, 1), ar = c(1, -.9), ma =
.8), n = 250)
```

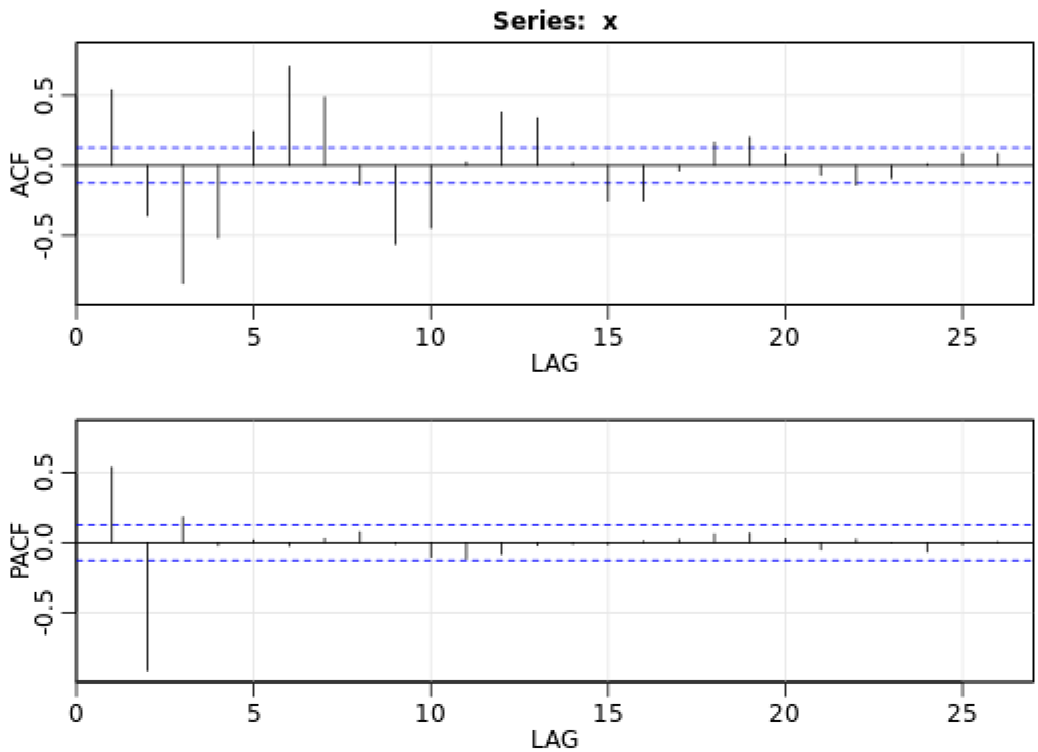
. Look at the simulated data and the sample ACF and PACF pair to determine a possible model.

Recall that for ARMA(p, q) models, both the theoretical ACF and PACF tail off. In this case, the orders are difficult to discern from data and it may not be clear if either the sample ACF or sample PACF is cutting off or tailing off. In this case, you know the actual model orders, so fit an ARMA(2,1) to the generated data. General modeling strategies will be discussed further in the course.

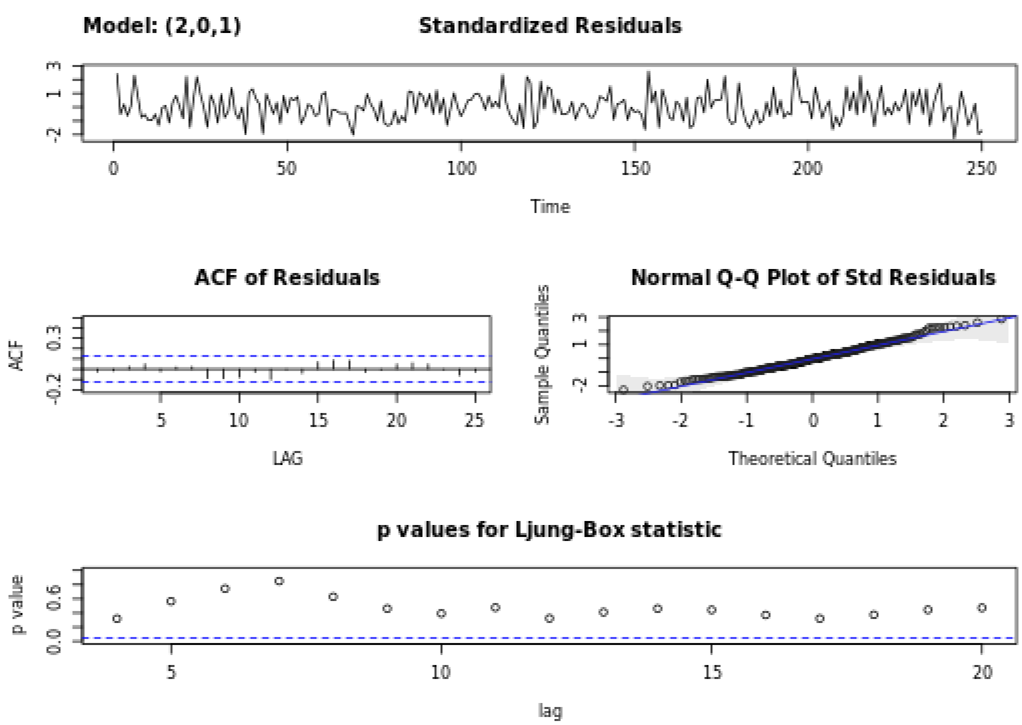
```
# astsa is preloaded

# Plot x
plot(x)
```

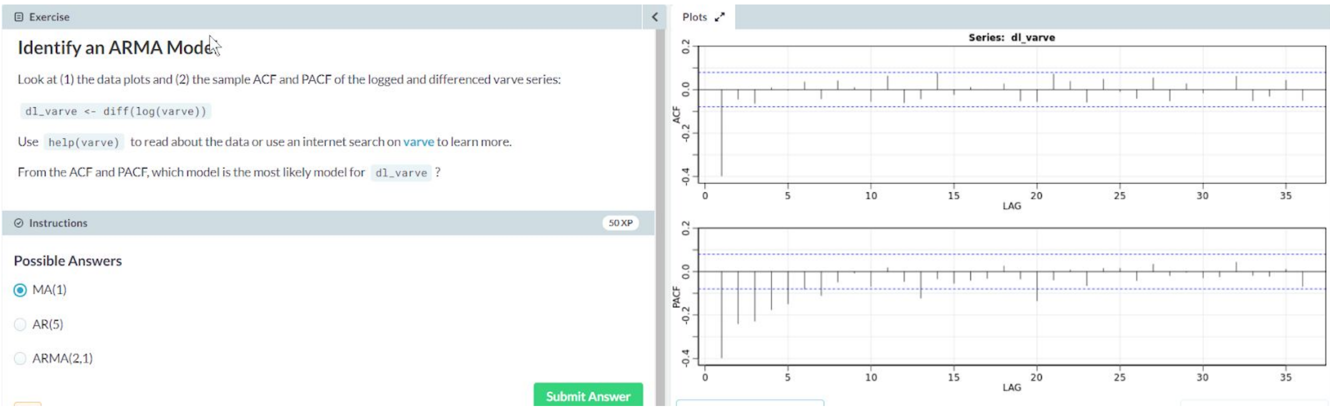



```
# Plot the sample P/ACF of x
acf2(x)
```



# Fit an ARMA(2,1) to the data and examine the t-table
sarima(x, p = 2, d = 0, q = 1)





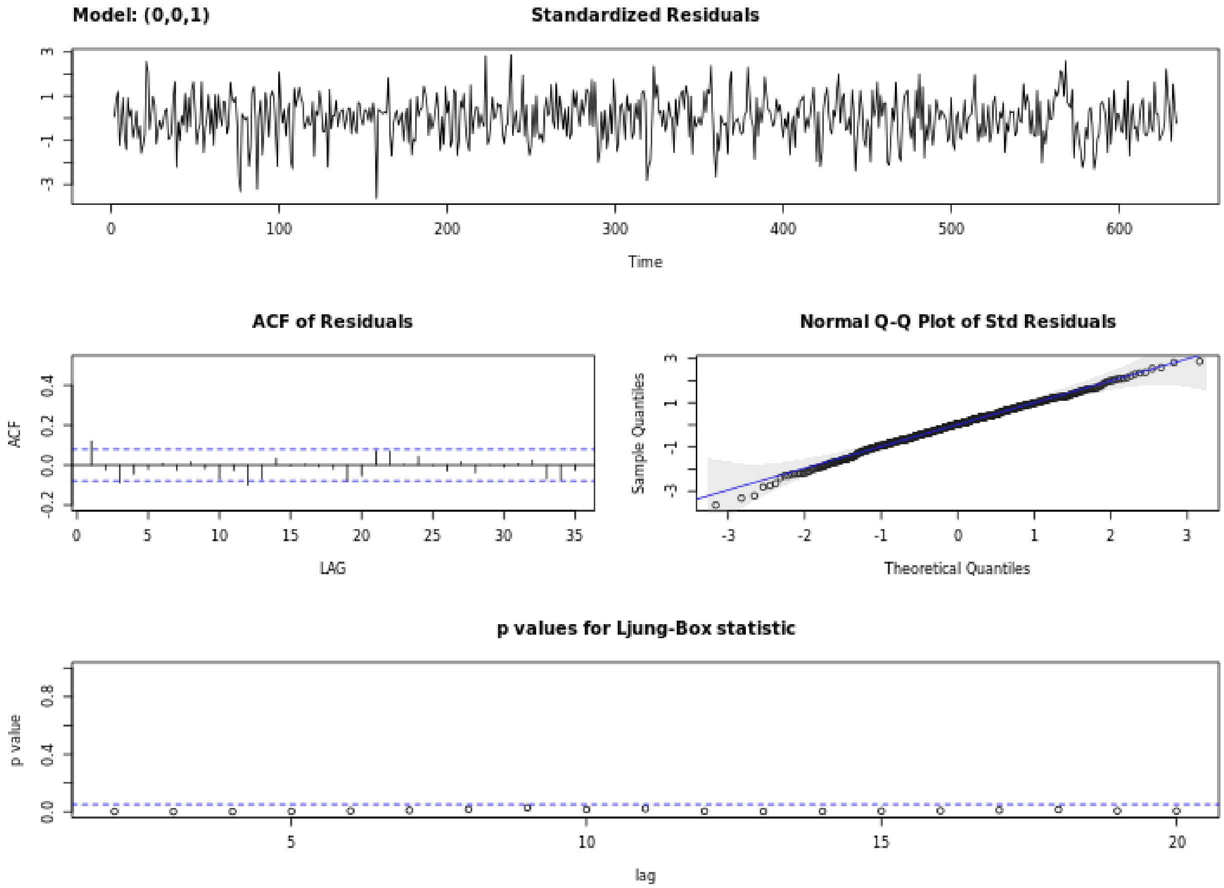
Model Choice - I

Based on the sample P/ACF pair of the logged and differenced varve data (`dl_varve`), an MA(1) was indicated. The best approach to fitting ARMA is to start with a low order model, and then try to add a parameter at a time to see if the results change.
In this exercise, you will fit various models to the `dl_varve` data and note the AIC and BIC for each model. In the next exercise, you will use these AICs and BICs to choose a model. Remember that you want to retain the model with the smallest AIC and/or BIC value.
A note before you start:
`sarima(x, p = 0, d = 0, q = 1)` and `sarima(x, 0, 0, 1)`
are the same.

- The package **astsa** is preloaded. The `varve` series has been logged and differenced as `dl_varve <- diff(log(varve))`.
- Use `sarima()` to fit an MA(1) to `dl_varve`. Take a close look at the output of your `sarima()` command to see the AIC and BIC for this model.

Fit an MA(1) to dl_varve.

sarima(dl_varve, p = 0, d = 0, q = 1)



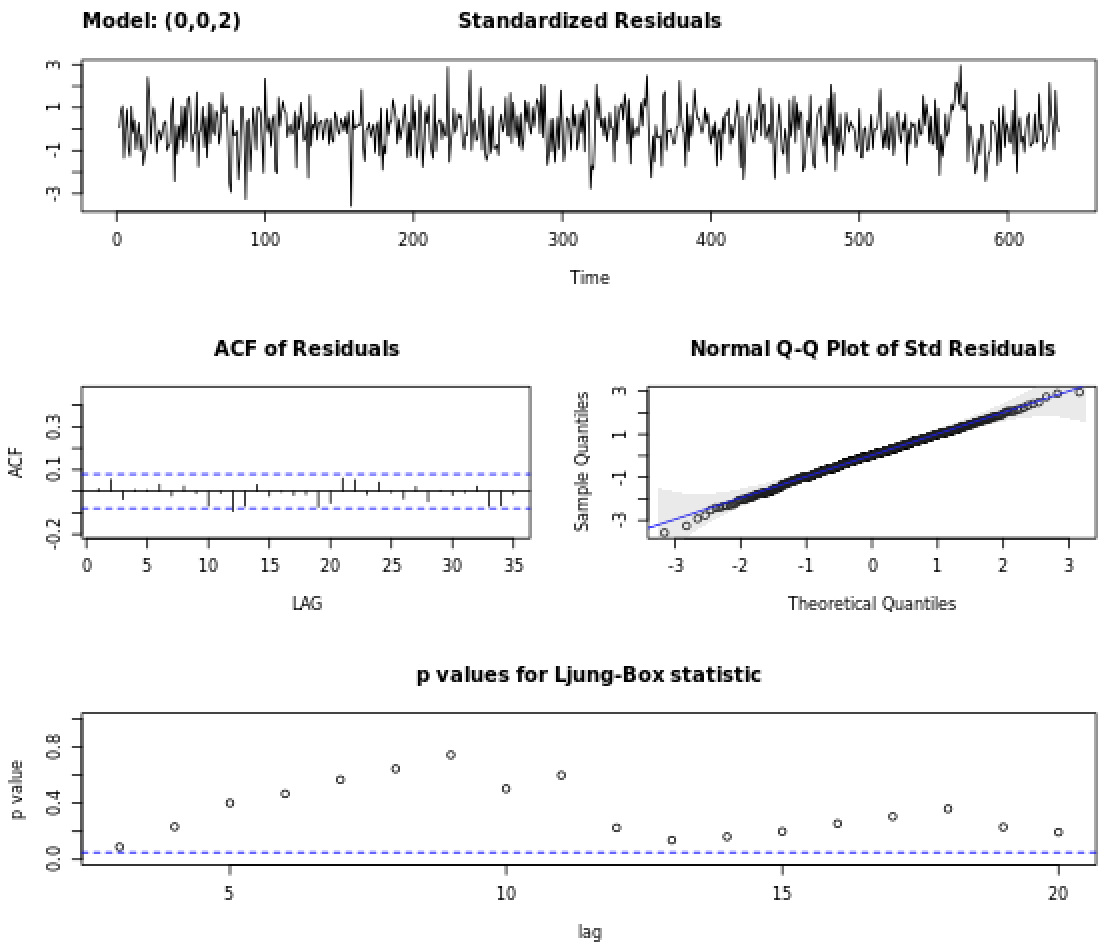
Model: (0,0,1)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
    REPORT = 1, reltol = tol))

Coefficients:
          ma1    xmean
      -0.7710  -0.0013
s.e.   0.0341   0.0044

sigma^2 estimated as 0.2353:  log likelihood = -440.68,  aic = 887.36

$degrees_of_freedom
[1] 631

$ttable
      Estimate     SE  t.value p.value
ma1    -0.7710 0.0341 -22.6002  0.0000
xmean  -0.0013 0.0044  -0.2818  0.7782

$AIC
[1] -0.4406366

$AICc
[1] -0.4374168

$BIC
[1] -1.426575
```

- Repeat the previous exercise, but add an MA parameter by fitting an MA(2) model. Based on AIC and BIC, is this an improvement over the previous model?

Fit an MA(2) to dl_varve. Improvement?

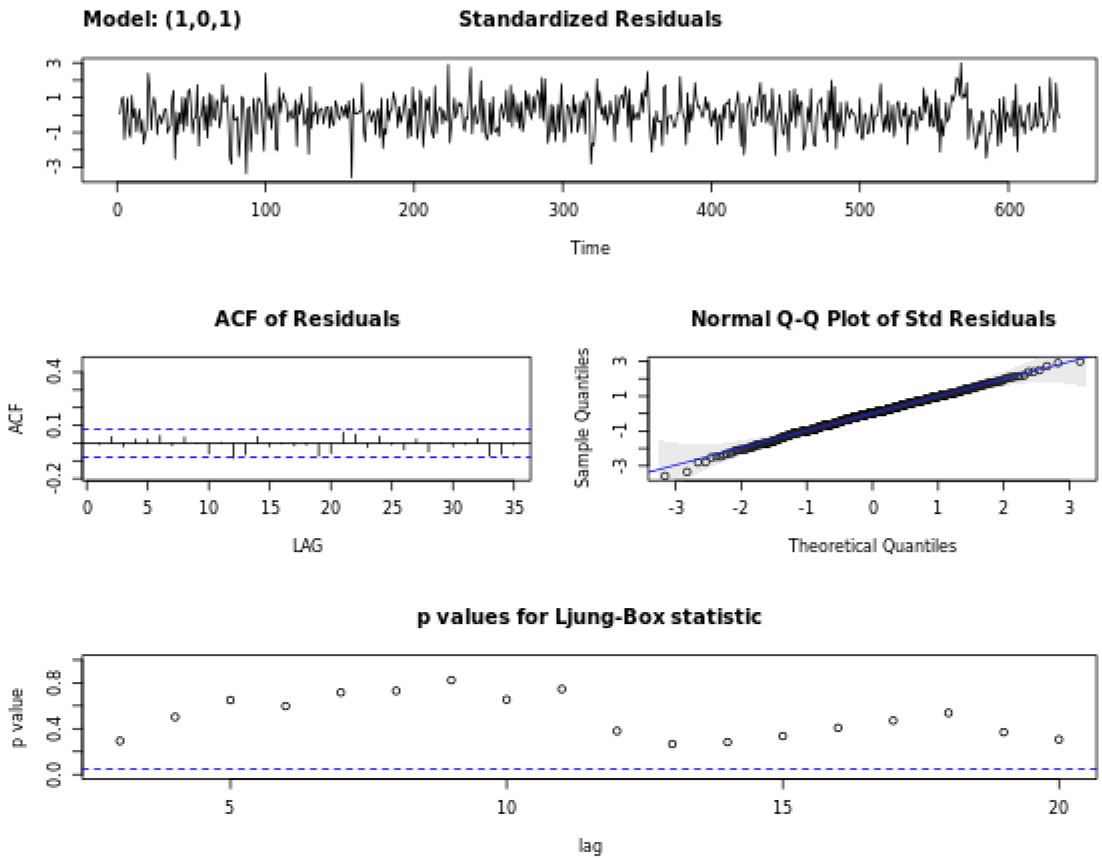
sarima(dl_varve, p = 0, d = 0, q = 2)


Model: (0,0,2)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
    REPORT = 1, reltol = tol))

Coefficients:
          ma1      ma2    xmean
      -0.6710  -0.1595  -0.0013
s.e.   0.0375   0.0392   0.0033

sigma^2 estimated as 0.2294:  log likelihood = -432.69,  aic = 873.39

$degrees_of_freedom
[1] 630

$ttable
      Estimate     SE  t.value p.value
ma1    -0.6710 0.0375 -17.9057  0.0000
ma2    -0.1595 0.0392  -4.0667  0.0001
xmean  -0.0013 0.0033  -0.4007  0.6888

$AIC
[1] -0.4629629

$AICc
[1] -0.4597027

$BIC
[1] -1.441871
```

- Instead of adding an MA parameter, add an AR parameter to the original MA(1) fit. That is, fit an ARMA(1,1) to `dl_varve`. Based on AIC and BIC, is this an improvement over the previous models?

# Fit an ARMA(1,1) to dl_varve. Improvement?
sarima(dl_varve, p = 1, d = 0, q = 1)



Model: (1,0,1)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
    REPORT = 1, reltol = tol))

Coefficients:
         ar1      ma1    xmean
      0.2341  -0.8871  -0.0013
s.e.  0.0518   0.0292   0.0028

sigma^2 estimated as 0.2284:  log likelihood = -431.33,  aic = 870.66

$degrees_of_freedom
[1] 630

$ttable
      Estimate     SE  t.value p.value
ar1     0.2341 0.0518   4.5184  0.0000
ma1    -0.8871 0.0292 -30.4107  0.0000
xmean  -0.0013 0.0028  -0.4618  0.6444

$AIC
[1] -0.467376

$AICc
[1] -0.4641159

$BIC
[1] -1.446284
```

Model Choice - II

In the previous exercise, you fit three different models to the logged and differenced varve series (`dl_varve`). The data are displayed to the right. The extracted AIC and BIC from each run are tabled below.

Model	AIC	BIC
MA(1)	-0.4437	-1.4366
MA(2)	-0.4659	-1.4518
ARMA(1,1)	-0.4702	-1.4561

Using the table, indicate which statement below is FALSE.

Instructions 50 XP

Possible Answers

- ○ AIC and BIC both prefer the ARMA(1,1) model over the other fitted models.
- ◉ AIC prefers the MA(1) model.
- ○ BIC prefers the MA(2) over the MA(1)
- ○ Because they use different penalties, the AIC and BIC can prefer different models.

Exactly! The lowest AIC value of the three models is the ARMA(1,1) model, meaning AIC prefers that model over the MA(1) model.

Residual Analysis - I

As you saw in the video, an `sarima()` run includes a residual analysis graphic. Specifically, the output shows (1) the standardized residuals, (2) the sample ACF of the residuals, (3) a normal Q-Q plot, and (4) the p-values corresponding to the Box-Ljung-Pierce Q-statistic.
In each run, check the four residual plots as follows:

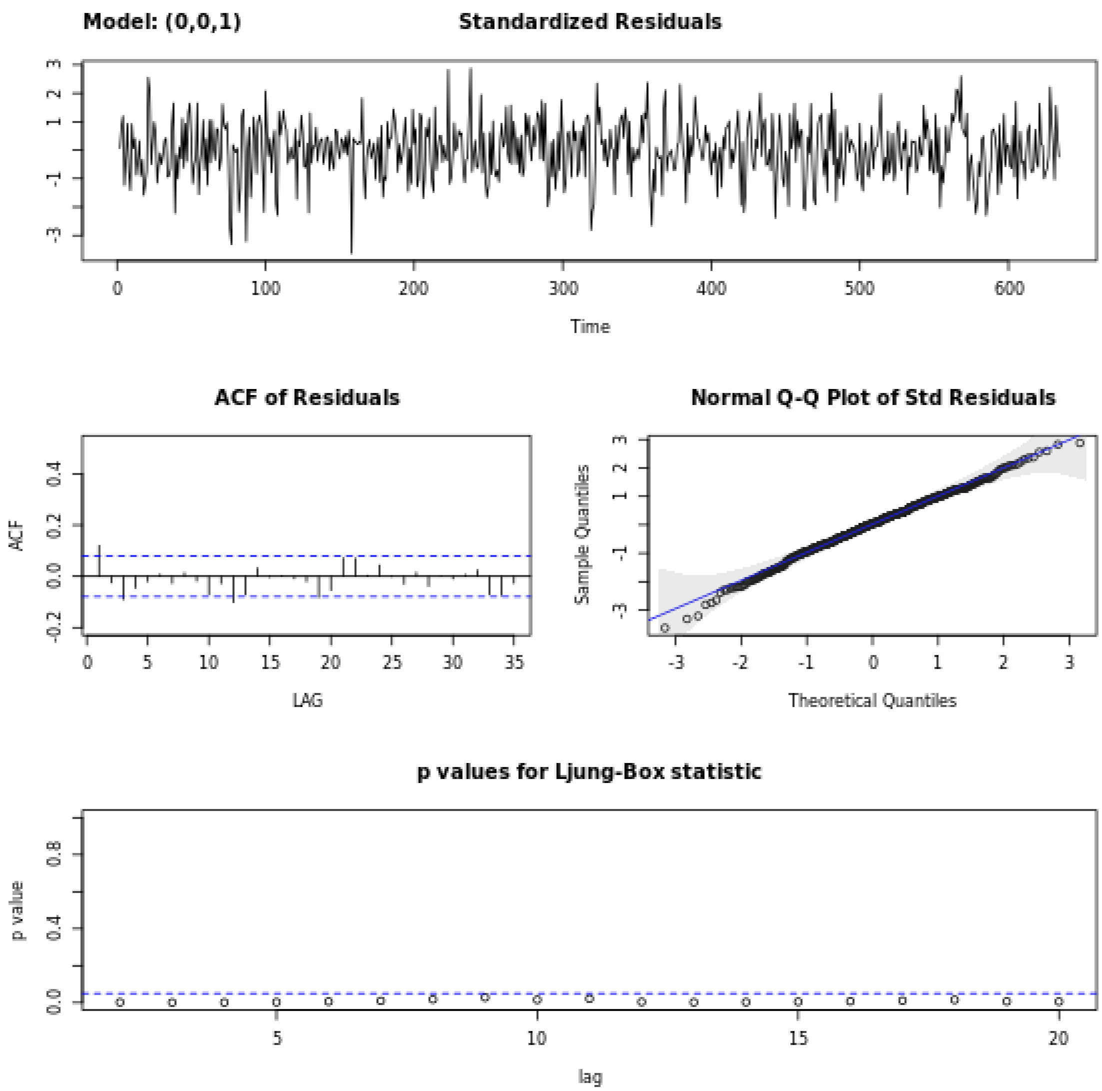
1. The standardized residuals should behave as a white noise sequence with mean zero and variance one. Examine the residual plot for departures from this behavior.
2. The sample ACF of the residuals should look like that of white noise. Examine the ACF for departures from this behavior.

3. Normality is an essential assumption when fitting ARMA models. Examine the Q-Q plot for departures from normality and to identify outliers.
4. Use the Q-statistic plot to help test for departures from whiteness of the residuals.

As in the previous exercise, `dl_varve <- diff(log(varve))`, which is plotted below a plot of `varve`. The **astsa** package is preloaded.

- Use `sarima()` to fit an MA(1) to `dl_varve` and do a complete residual analysis as prescribed above. Make a note of what you see for the next exercise.

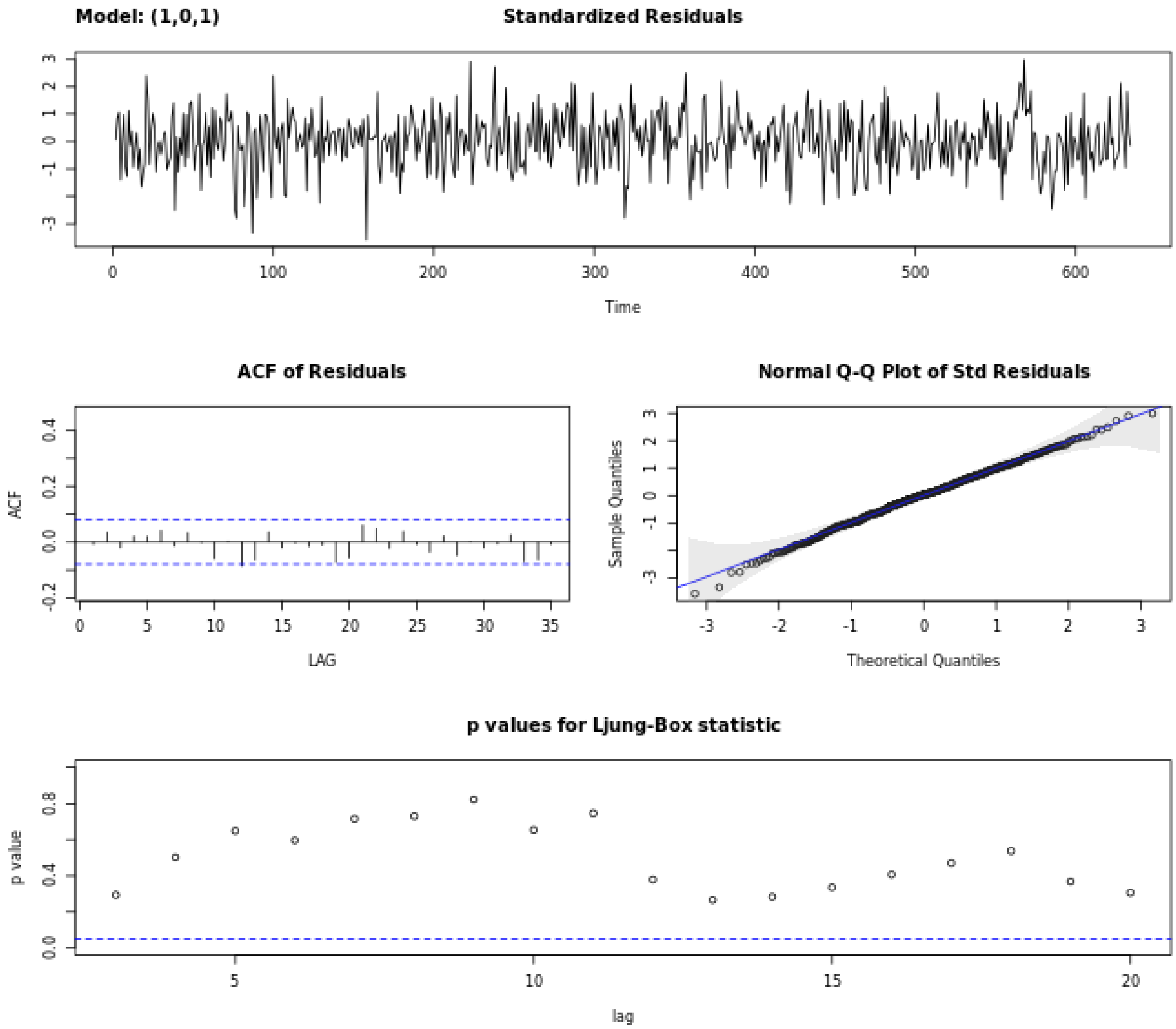
```
# Fit an MA(1) to dl_varve. Examine the residuals
sarima(dl_varve, p = 0, d = 0, q = 1)
```



```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control =
        list(trace = trc,
    REPORT = 1, reltol = tol))

Coefficients:
          ma1    xmean
      -0.7710  -0.0013
s.e.   0.0341   0.0044

sigma^2 estimated as 0.2353:  log likelihood = -440.68,  aic = 887.36

$degrees_of_freedom
[1] 631

$ttable
      Estimate     SE  t.value p.value
ma1    -0.7710 0.0341 -22.6002  0.0000
xmean  -0.0013 0.0044  -0.2818  0.7782

$AIC
[1] -0.4406366

$AICc
[1] -0.4374168

$BIC
[1] -1.426575
```

- Use another call to `sarima()` to fit an ARMA(1,1) to `dl_varve` and do a complete residual analysis as prescribed above. Again, make a note of what you see for the next exercise.

```
# Fit an ARMA(1,1) to dl_varve. Examine the residuals
sarima(dl_varve, p = 1, d = 0, q = 1)
```


Model: (1,0,1)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
    REPORT = 1, reltol = tol))

Coefficients:
         ar1      ma1    xmean
      0.2341  -0.8871  -0.0013
s.e.  0.0518   0.0292   0.0028

sigma^2 estimated as 0.2284:  log likelihood = -431.33,  aic = 870.66

$degrees_of_freedom
[1] 630

$ttable
      Estimate     SE  t.value p.value
ar1     0.2341 0.0518   4.5184  0.0000
ma1    -0.8871 0.0292 -30.4107  0.0000
xmean  -0.0013 0.0028  -0.4618  0.6444

$AIC
[1] -0.467376

$AICc
[1] -0.4641159

$BIC
[1] -1.446284
```

Residual Analysis - II

In the previous exercise, you fit two different ARMA models to the logged and differenced varve series: an MA(1) and an ARMA(1,1) model. The residual analysis graphics are displayed in order of the run:

1. MA(1)
2. ARMA(1, 1)

Which of the following statements is FALSE (partially truthful statements are false - data analysis is not politics)?

Instructions 50 XP

Possible Answers

- The residuals for the MA(1) model are not white noise.
- The residuals for the ARMA(1, 1) model appear to be Gaussian white noise.
- It is not a good idea to look at the residual analysis because it might tell you if your model is incorrect and you might have to stay late at work to figure out the correct model.

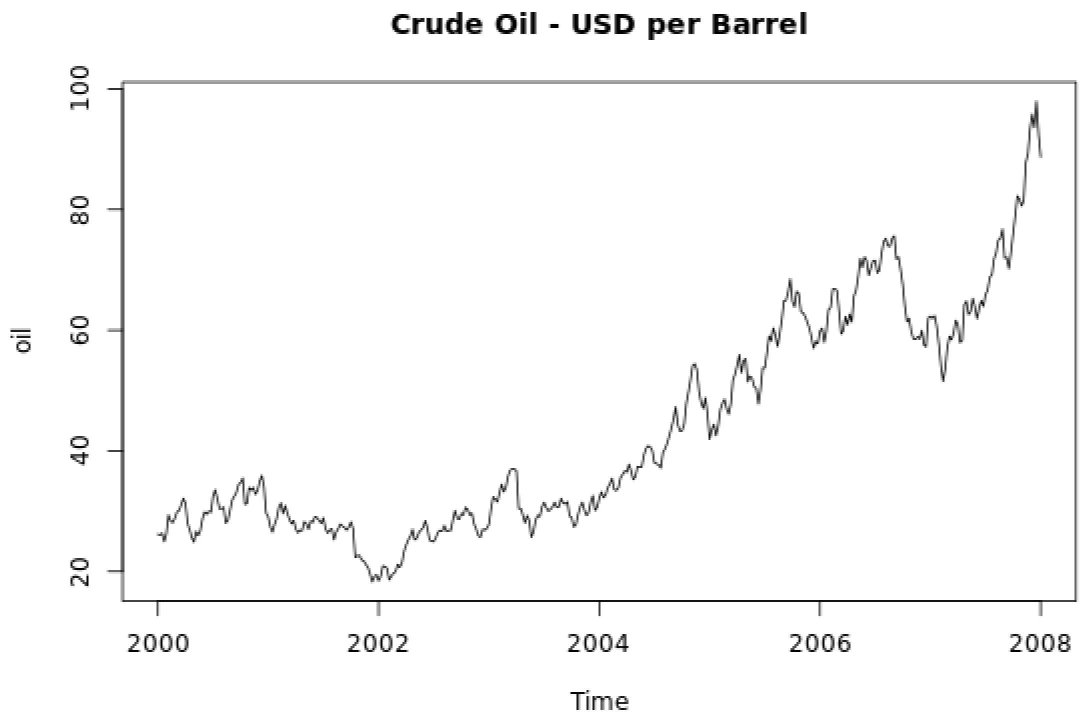
That's right! You should always examine the residuals because the model assumes the errors are Gaussian white noise.

ARMA get in

By now you have gained considerable experience fitting ARMA models to data, but before you start celebrating, try one more exercise (sort of) on your own.
The data in `oil` are crude oil, WTI spot price FOB (in dollars per barrel), weekly data from 2000 to 2008. Use your skills to fit an ARMA model to the returns. The weekly

crude oil prices (`oil`) are plotted for you. Throughout the exercise, work with the returns, which you will calculate.
As before, the [astsa](#) package is preloaded for you. The data are preloaded as `oil` and plotted on the right.

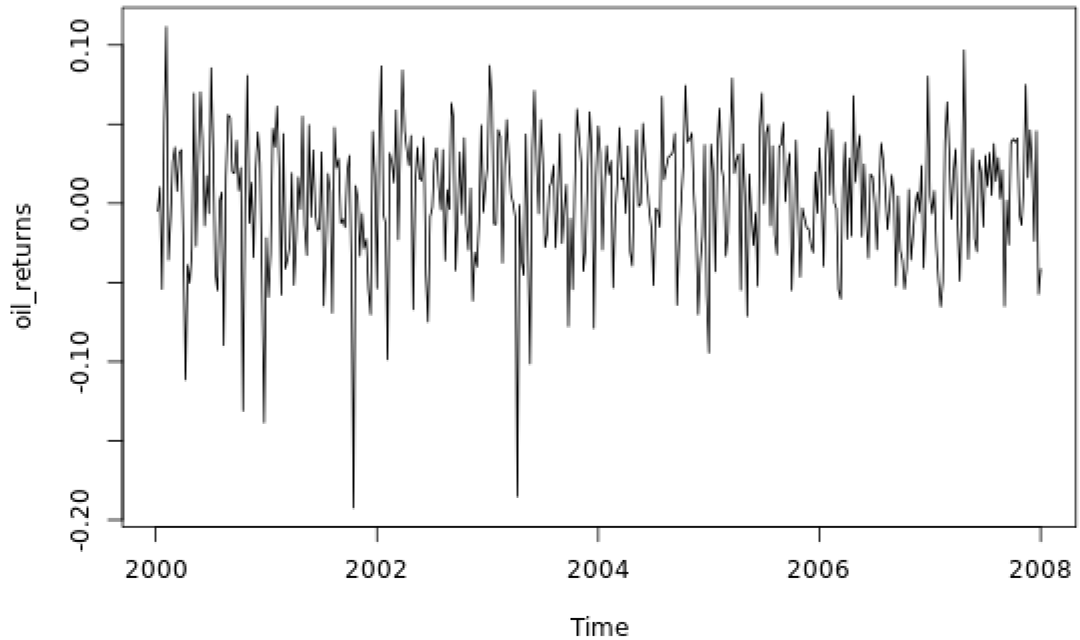
# Calculate approximate oil returns
plot(oil)



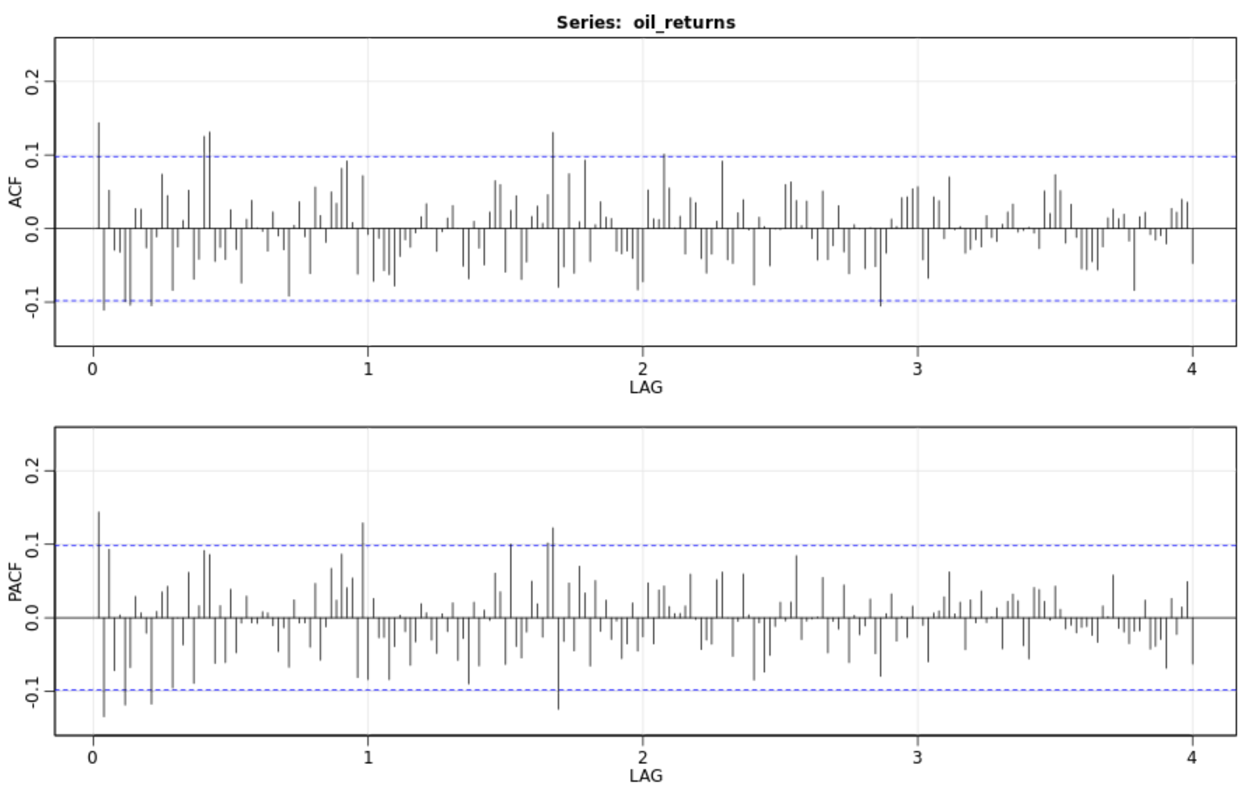
oil_returns <- diff(log(oil))

Plot oil_returns. Notice the outliers.Plot `oil_returns` and notice that there are a couple of outliers prior to 2004. Convince yourself that the returns are stationary.

plot(oil_returns)



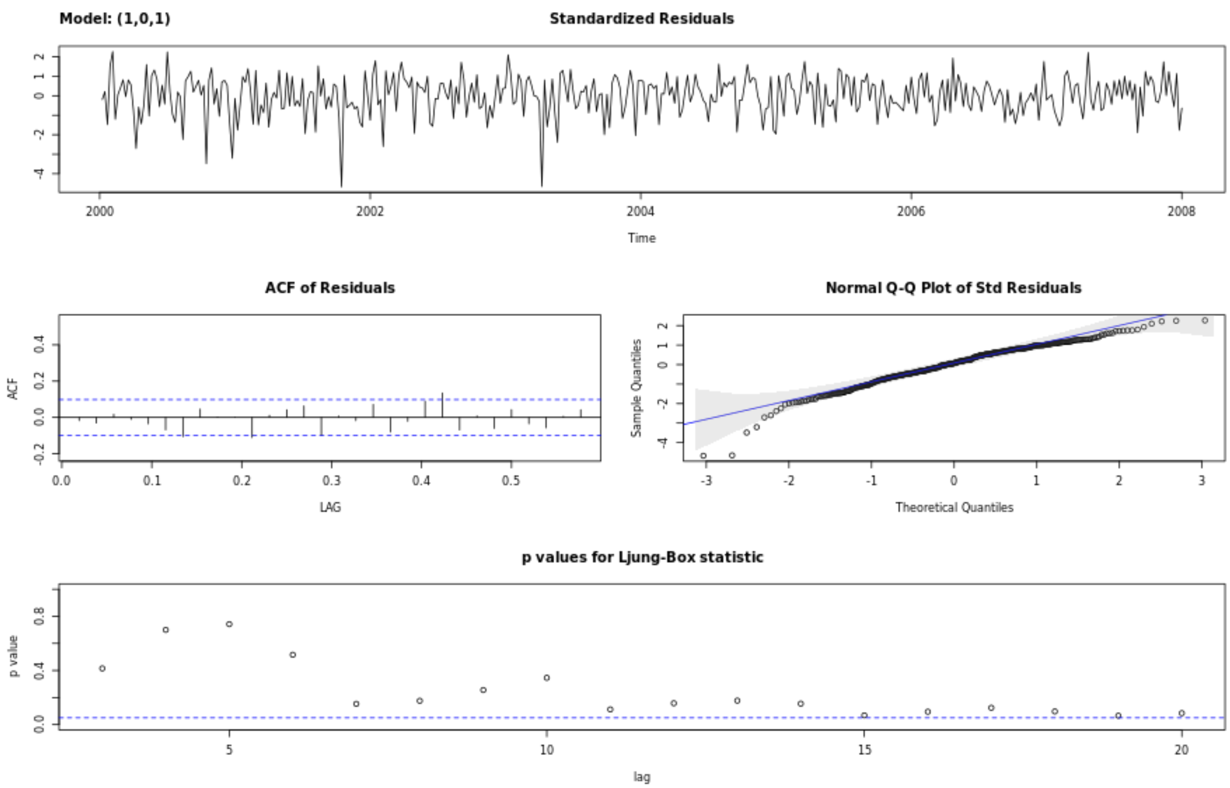
# Plot the P/ACF pair for oil_returns
acf2(oil_returns)



From the P/ACF pair, it is apparent that the correlations are small and the returns are nearly noise. But it could be that both the ACF and PACF are tailing off. If this is the case, then an ARMA(1,1) is suggested. Fit this model to the oil returns using `sarima()`. Does the model fit well? Can you see the outliers in the residual plot?

Assuming both P/ACF are tailing, fit a model

sarima(oil_returns, p = 1, d = 0, q = 1)



```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
    REPORT = 1, reltol = tol))

Coefficients:
          ar1     ma1   xmean
      -0.4752  0.6768  0.0029
s.e.   0.1084  0.0871  0.0022

sigma^2 estimated as 0.001589:  log likelihood = 750.22,  aic = -1492.43

$degrees_of_freedom
[1] 413

$ttable
      Estimate     SE t.value p.value
ar1    -0.4752 0.1084 -4.3846  0.0000
ma1     0.6768 0.0871  7.7674  0.0000
xmean   0.0029 0.0022  1.3184  0.1881

$AIC
[1] -5.430483

$AICc
[1] -5.425441

$BIC
[1] -6.401415
```

ARIMA - Integrated ARMA

ARIMA - Plug and Play

As you saw in the video, a time series is called ARIMA(p, d, q) if the differenced series (of order d) is ARMA(p, q).

To get a sense of how the model works, you will analyze simulated data from the integrated model

$$Y_t = .9Y_{t-1} + W_t$$

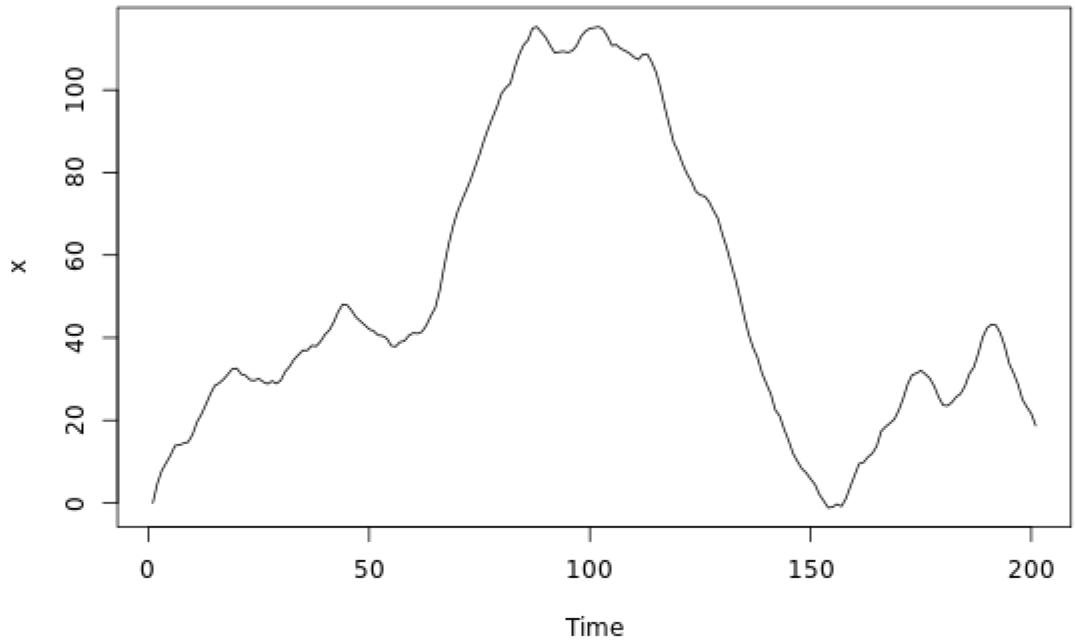
where $Y_t = \nabla X_t = X_t - X_{t-1}$. In this case, the model is an ARIMA(1,1,0) because the differenced data are an autoregression of order one.

The simulated time series is in `x` and it was generated in R as `x <- arima.sim(model = list(order = c(1, 1, 0), ar = .9), n = 200)`.

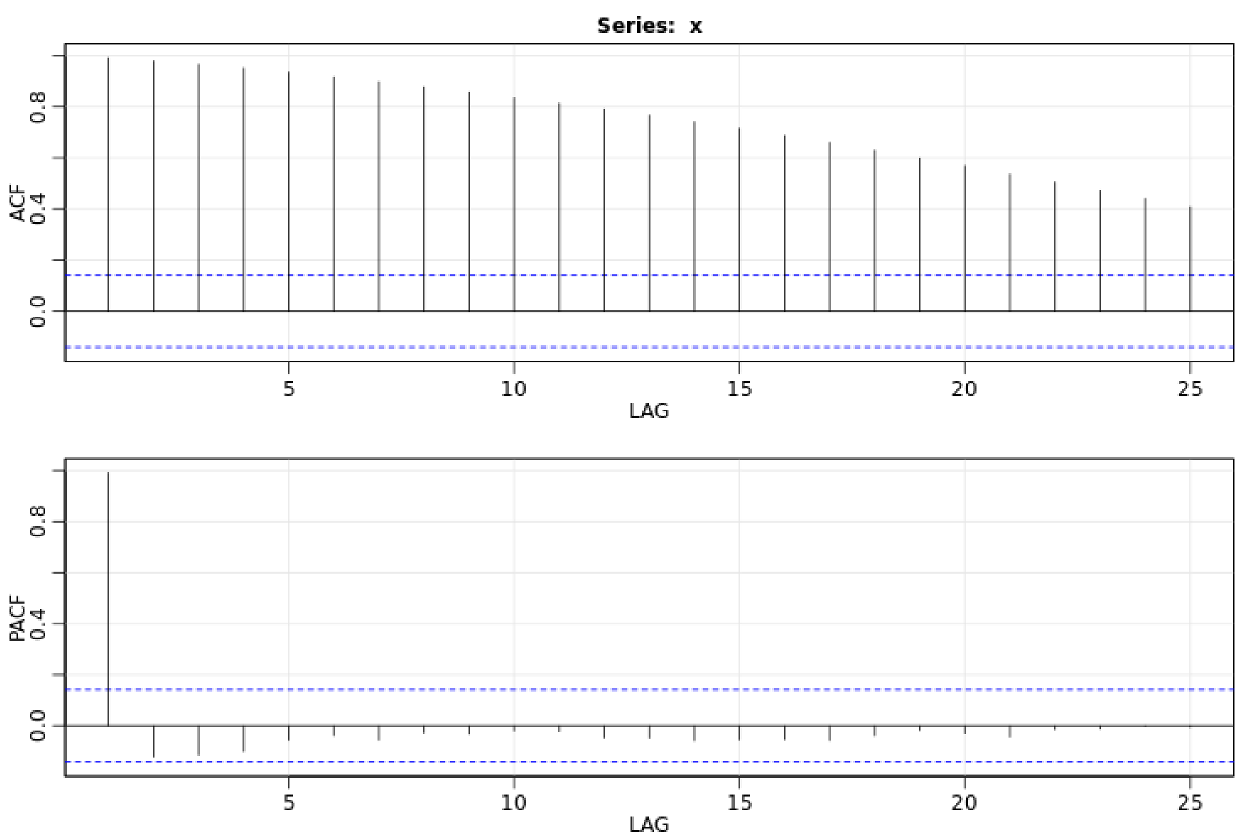
You will plot the generated data and the sample ACF and PACF of the generated data to see how integrated data behave. Then, you will difference the data to make it stationary. You will plot the differenced data and the corresponding sample ACF and PACF to see how differencing makes a difference.

As before, the **astsa** package is preloaded in your workspace. Data from an ARIMA(1,1,0) with AR parameter .9 is saved in object `x`.

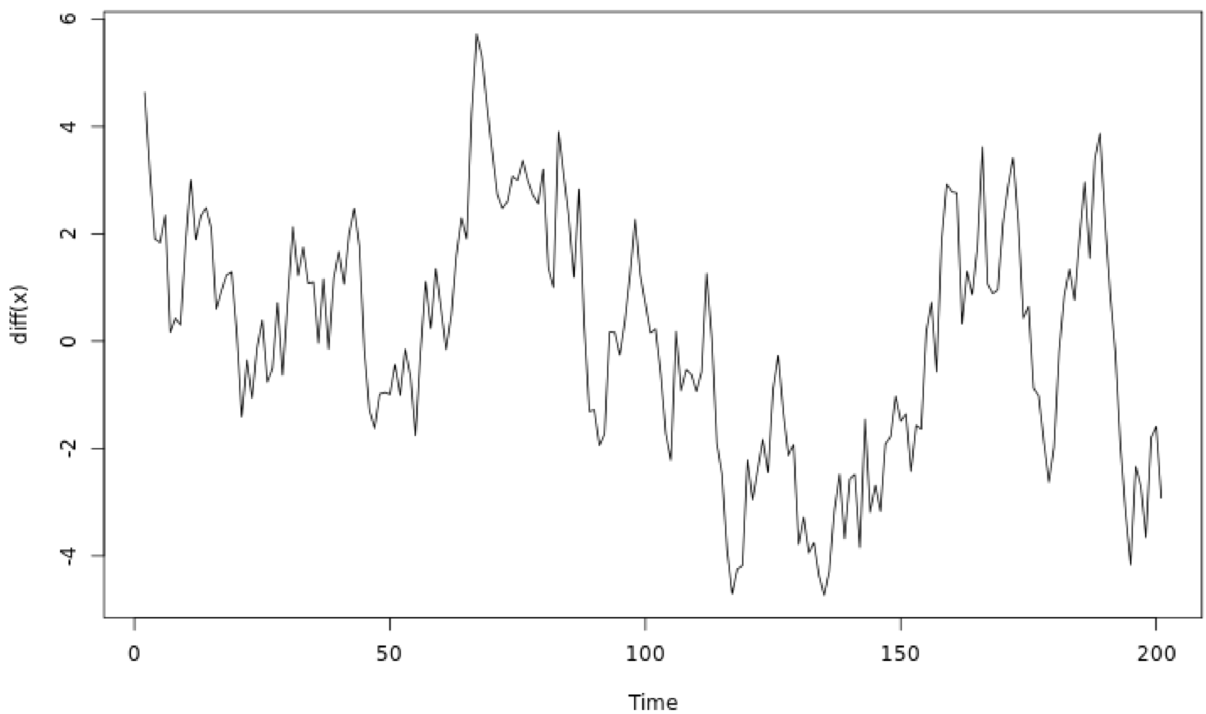
```
x <- arima.sim(model = list(order = c(1, 1, 0), ar = .9), n = 200).
plot(x)
```



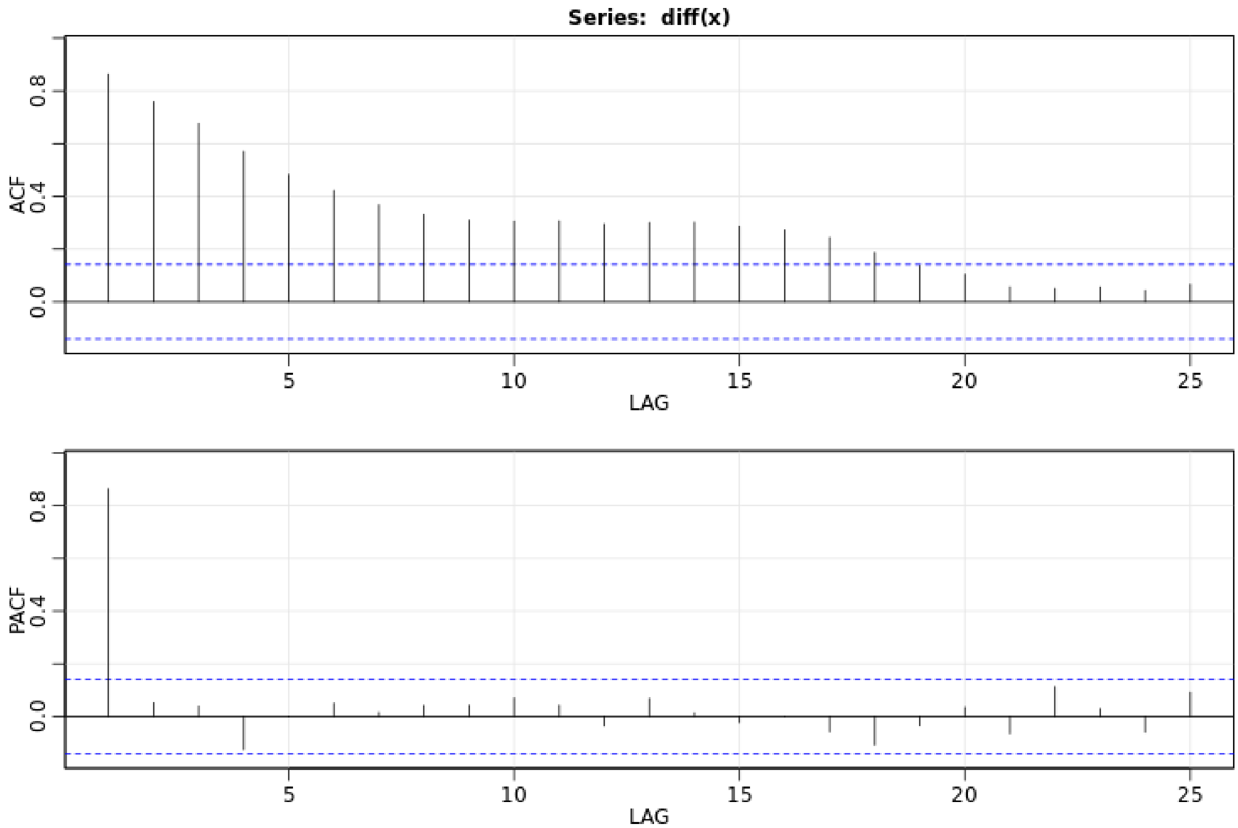
# Plot the P/ACF pair of x
acf2(x)



```
# Plot the differenced data
plot(diff(x))
```




```
# Plot the P/ACF pair of the differenced data
acf2(diff(x))
```



Simulated ARIMA

Before analyzing actual time series data, you should try working with a slightly more complicated model.

Here, we generated 250 observations from the ARIMA(2,1,0) model with drift given by

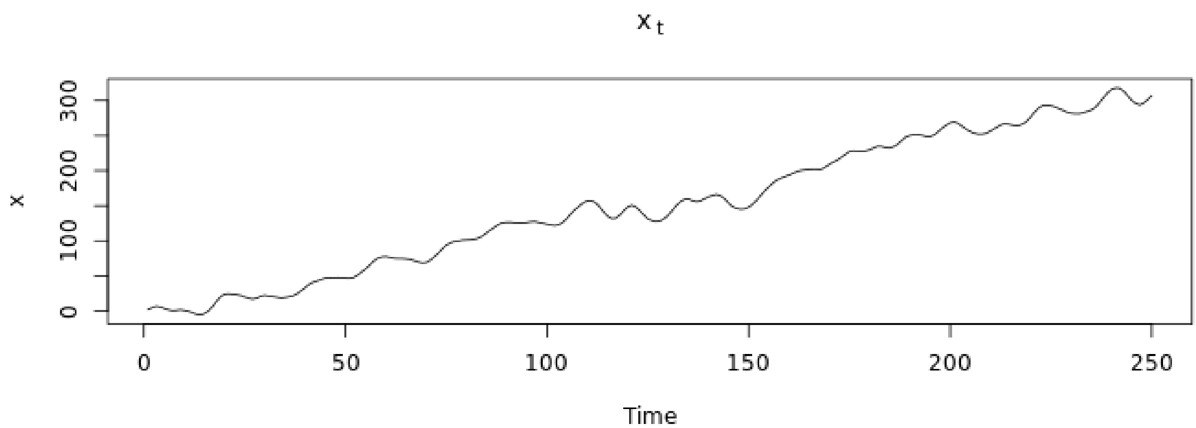
$$Y_t = 1 + 1.5Y_{t-1} - .75Y_{t-2} + W_t$$

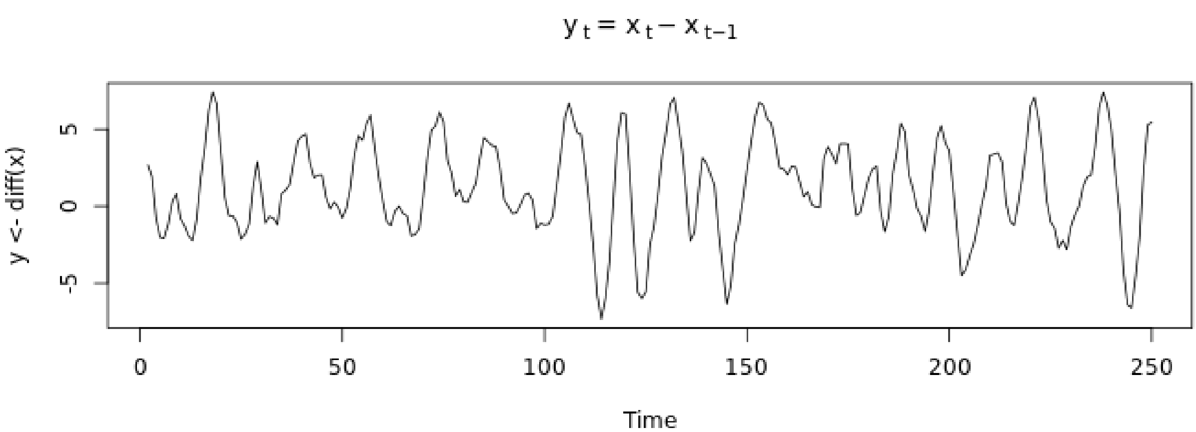
where $Y_t = \nabla X_t = X_t - X_{t-1}$.

You will use the established techniques to fit a model to the data.

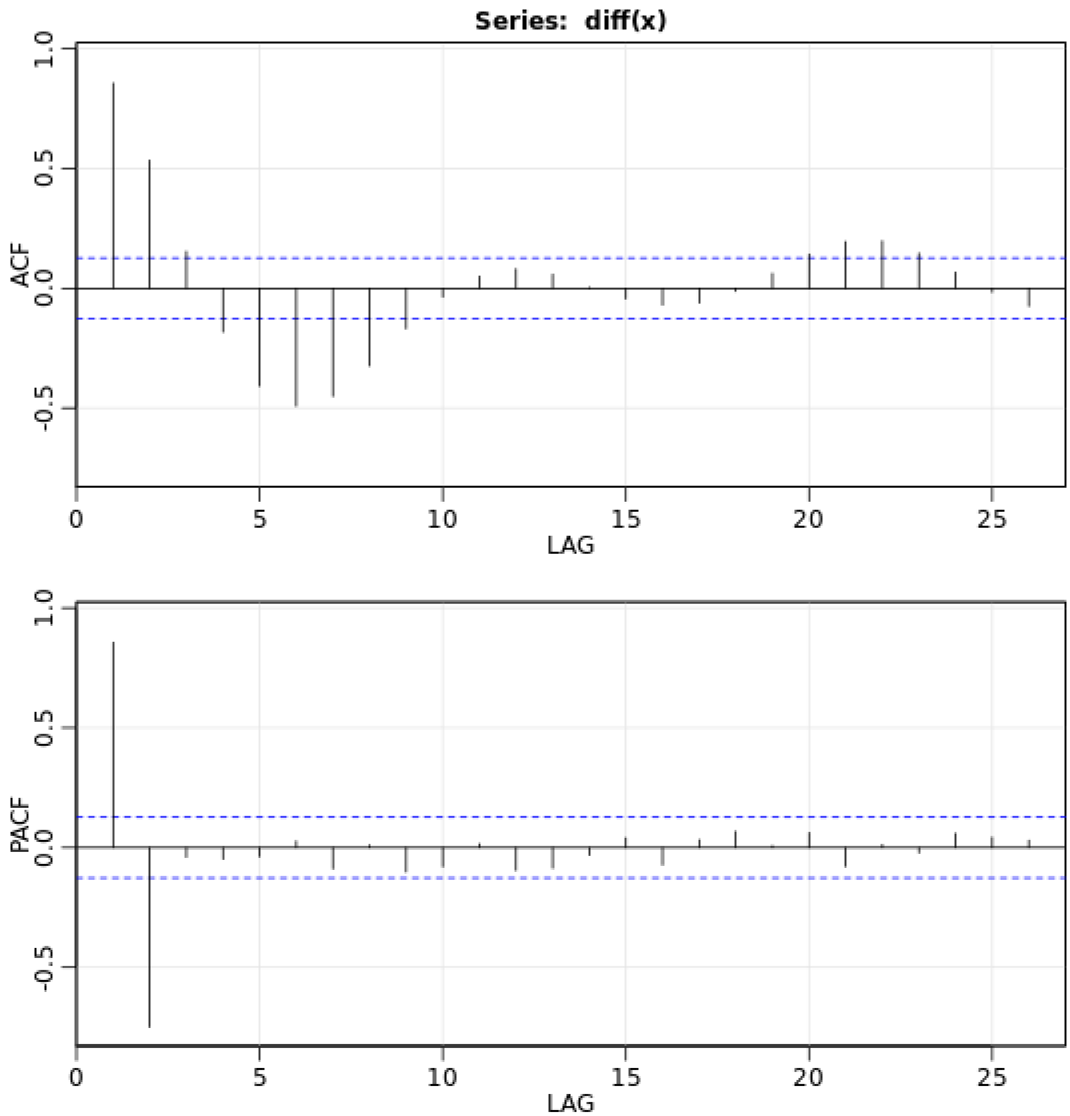
The astsa package is preloaded and the generated data are in `x`. The series `x` and the detrended series `y <- diff(x)` have been plotted.

GIVEN THAT:

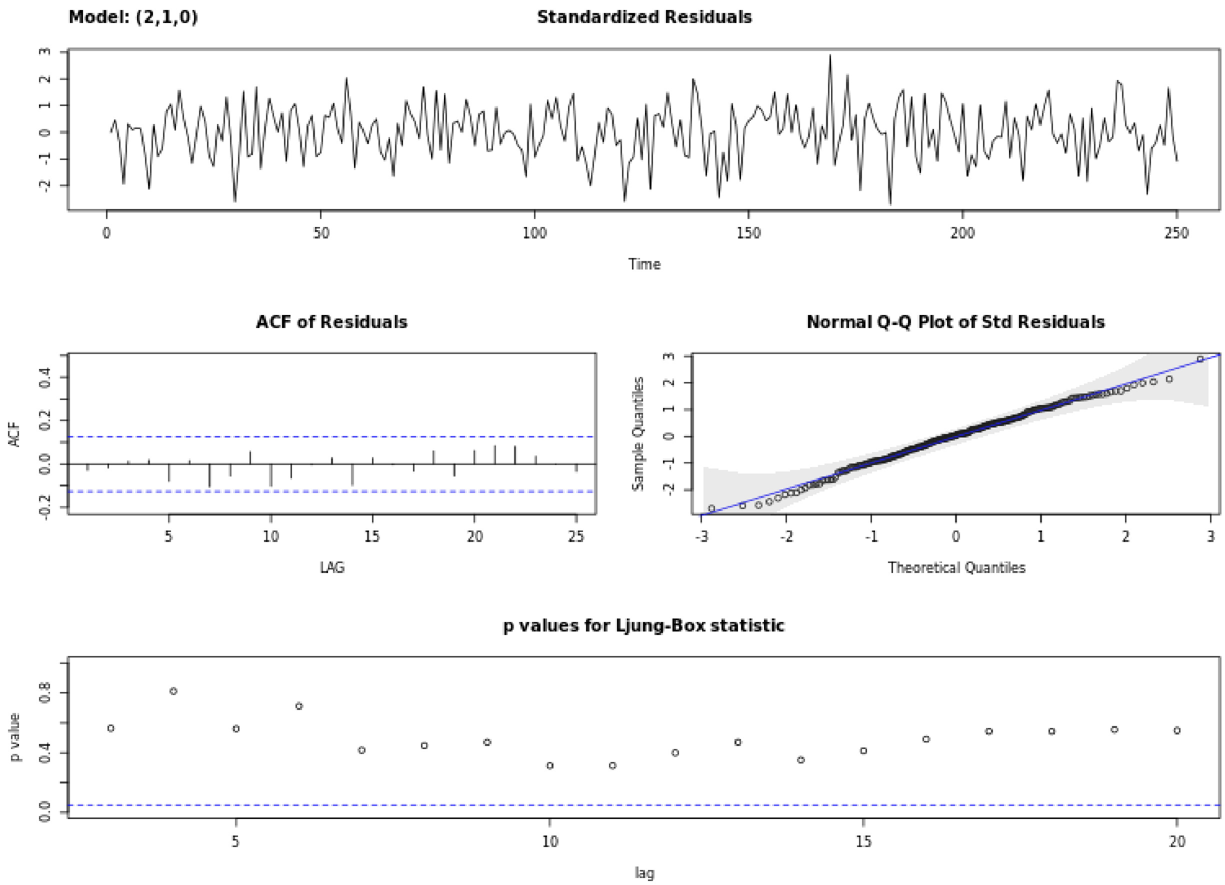





```
# Plot sample P/ACF of differenced data and determine model
acf2(diff(x))
```




```
# Estimate parameters and examine output
sarima(x, p = 2, d = 1, q = 0)
```


Model: (2,1,0)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = constant, optim.control = list(trace = trc, REPORT = 1,
    reltol = tol))

Coefficients:
         ar1      ar2  constant
      1.5197  -0.7669    1.2335
s.e.  0.0401   0.0401    0.2570

sigma^2 estimated as 1.004:  log likelihood = -355.41,  aic = 718.82

$degrees_of_freedom
[1] 246

$ttable
         Estimate     SE  t.value p.value
ar1        1.5197 0.0401  37.9191       0
ar2       -0.7669 0.0401 -19.1321       0
constant   1.2335 0.2570   4.7996       0

$AIC
[1] 1.02828

$AICc
[1] 1.036933

$BIC
[1] 0.07053753
```

Global Warming

Now that you have some experience fitting an ARIMA model to simulated data, your next task is to apply your skills to some real world data.
The data in `globtemp` (from `astsa`) are the **annual global temperature deviations** to 2015. In this exercise, you will use established techniques to fit an ARIMA model to the data. A plot of the data shows random walk behavior, which suggests you should work with the *differenced* data. The differenced data `diff(globtemp)` are also plotted. After plotting the sample ACF and PACF of the differenced data `diff(globtemp)`, you can say that either

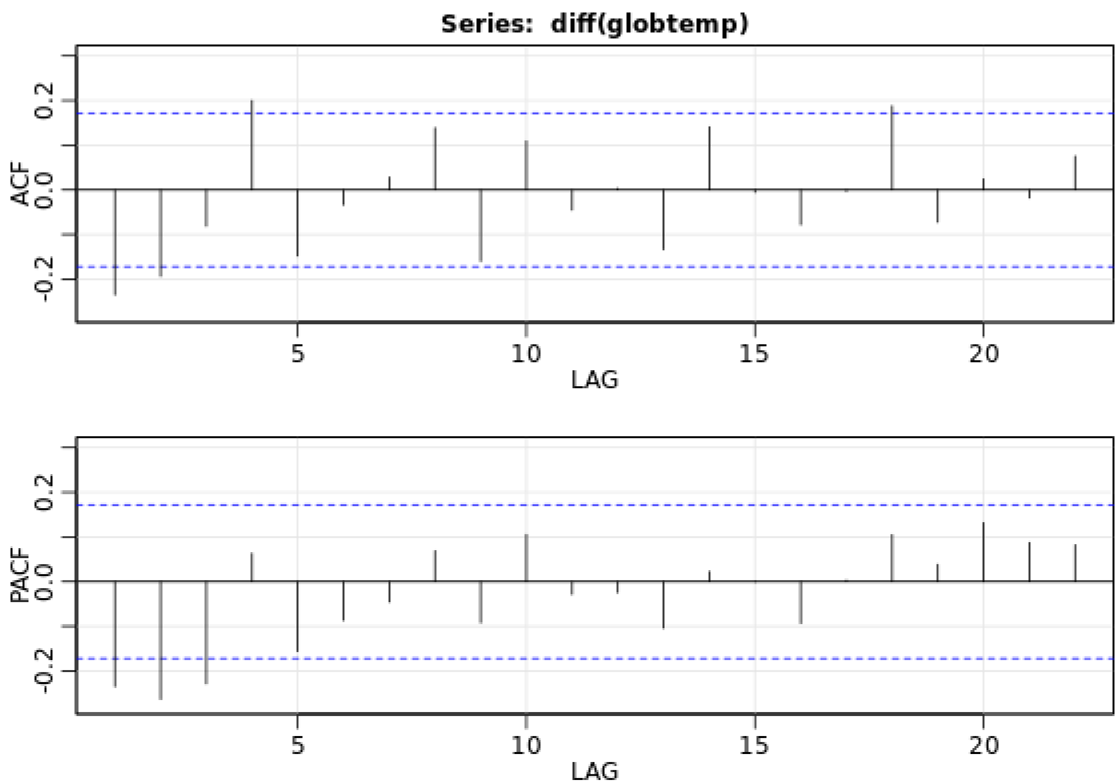
1. The ACF and the PACF are both tailing off, implying an ARIMA(1,1,1) model.
2. The ACF cuts off at lag 2, and the PACF is tailing off, implying an ARIMA(0,1,2) model.

3. The ACF is tailing off and the PACF cuts off at lag 3, implying an ARIMA(3,1,0) model. Although this model fits reasonably well, it is the worst of the three (you can check it) because it uses too many parameters for such small autocorrelations.

After fitting the first two models, check the AIC and BIC to choose the preferred model.

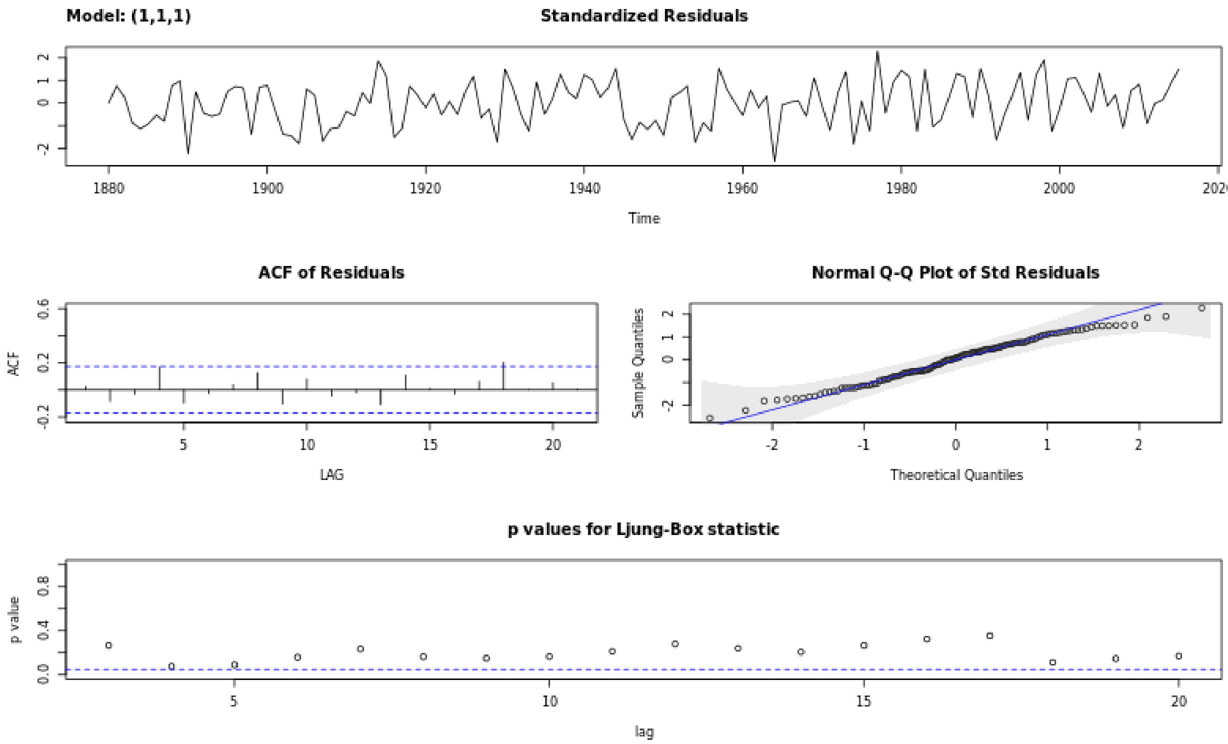
Plot the sample P/ACF pair of the differenced data, `diff(globtemp)`, to discover that 2 models seem reasonable, an ARIMA(1,1,1) and an ARIMA(0,1,2).

acf2(diff(globtemp))



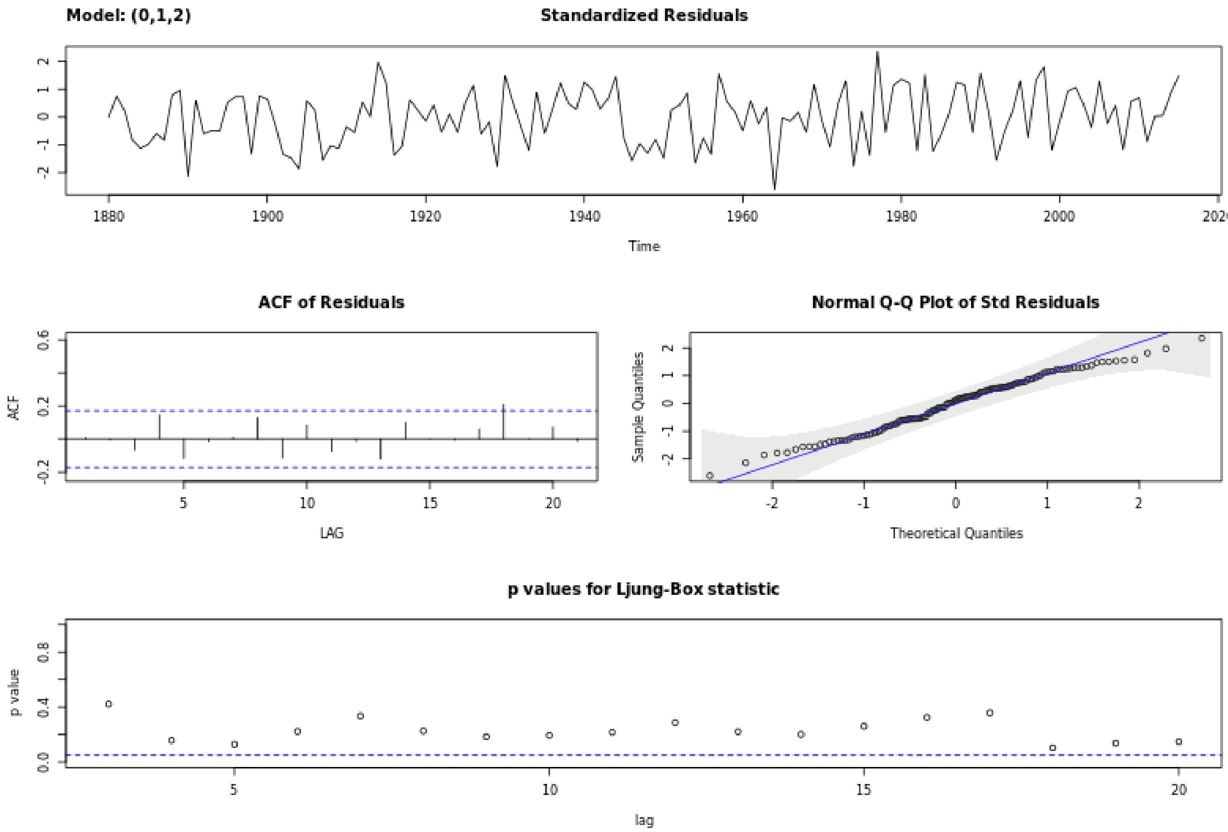
Fit an ARIMA(1,1,1) model to globtemp. Are all the parameters significant?

sarima(globtemp, p = 1, d = 1, q = 1)


Model: (1,1,1)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = constant, optim.control = list(trace = trc, REPORT = 1,
    reltol = tol))

Coefficients:
         ar1      ma1  constant
      0.3549  -0.7663    0.0072
s.e.  0.1314   0.0874    0.0032

sigma^2 estimated as 0.009885:  log likelihood = 119.88,  aic = -231.76

$degrees_of_freedom
[1] 132

$ttable
         Estimate     SE t.value p.value
ar1        0.3549 0.1314  2.7008  0.0078
ma1       -0.7663 0.0874 -8.7701  0.0000
constant   0.0072 0.0032  2.2738  0.0246

$AIC
[1] -3.572642

$AICc
[1] -3.555691

$BIC
[1] -4.508392
```

```
# Fit an ARIMA(0,1,2) model to globtemp. Which model is better?
sarima(globtemp, p = 0, d = 1, q = 2)
```


Model: (0,1,2)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

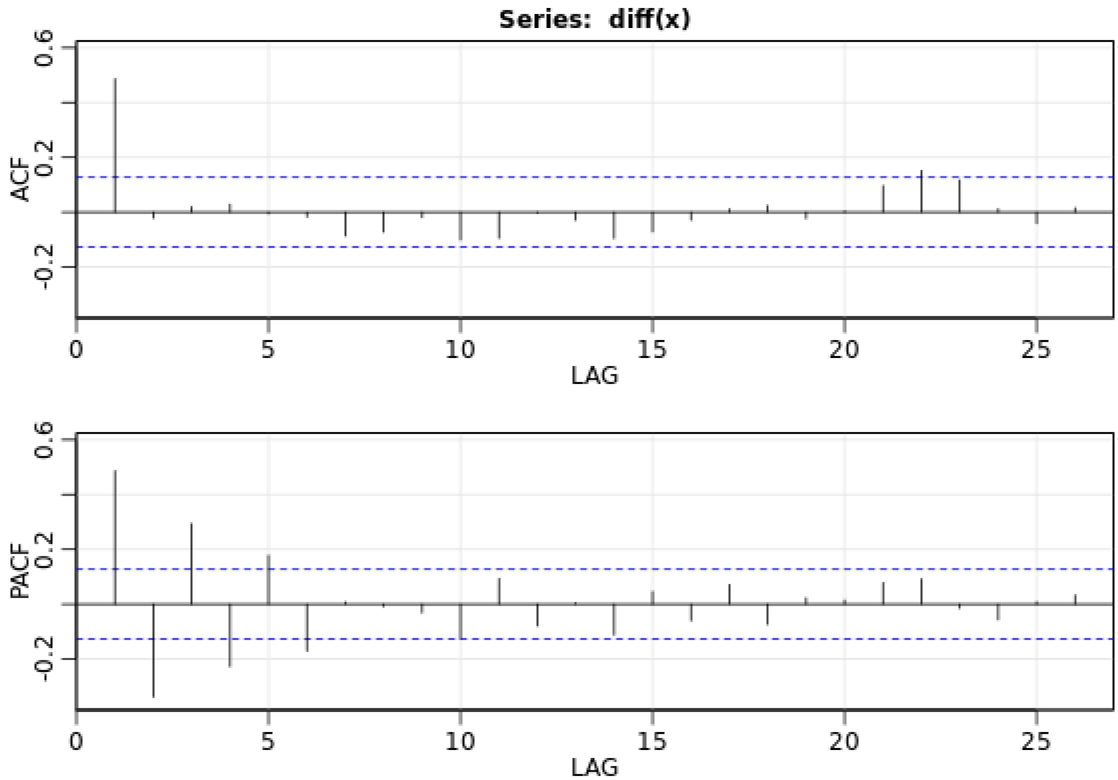
```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = constant, optim.control = list(trace = trc, REPORT = 1,
    reltol = tol))

Coefficients:
          ma1      ma2  constant
      -0.3984  -0.2173    0.0072
s.e.   0.0808   0.0768    0.0033

sigma^2 estimated as 0.00982:  log likelihood = 120.32,  aic = -232.64

$degrees_of_freedom
[1] 132

$ttable
         Estimate     SE t.value p.value
ma1       -0.3984 0.0808 -4.9313  0.0000
ma2       -0.2173 0.0768 -2.8303  0.0054
constant   0.0072 0.0033  2.1463  0.0337

$AIC
[1] -3.579224

$AICc
[1] -3.562273

$BIC
[1] -4.514974
```

Diagnostics - Simulated Overfitting

One way to check an analysis is to overfit the model by adding an extra parameter to see if it makes a difference in the results. If adding parameters changes the results drastically, then you should rethink your model. If, however, the results do not change by much, you can be confident that your fit is correct.
We generated 250 observations from an ARIMA(0,1,1) model with MA parameter 0.9. First, you will fit the model to the data using established techniques.
Then, you can check a model by overfitting (adding a parameter) to see if it makes a difference. In this case, you will add an additional MA parameter to see that it is not needed.

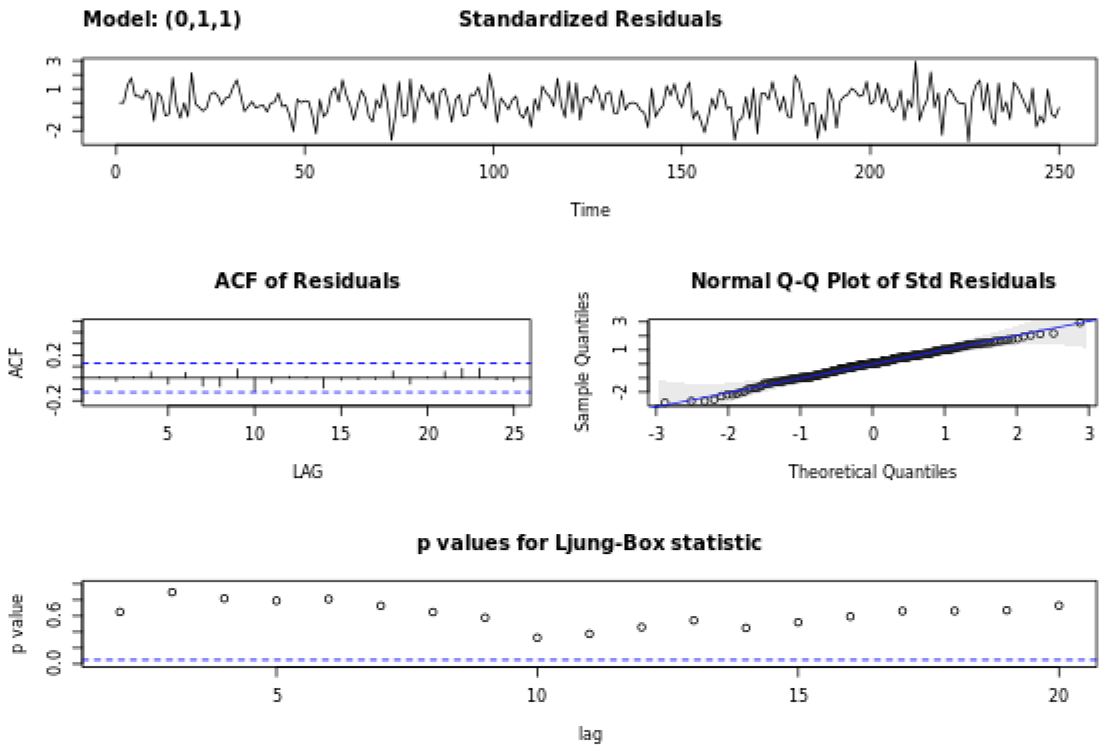
As usual, the **astsa** package is preloaded and the generated data in `x` are plotted in your workspace. The differenced data `diff(x)` are also plotted. Note that it looks stationary.

```
# Plot sample P/ACF pair of the differenced data
acf2(diff(x))
```

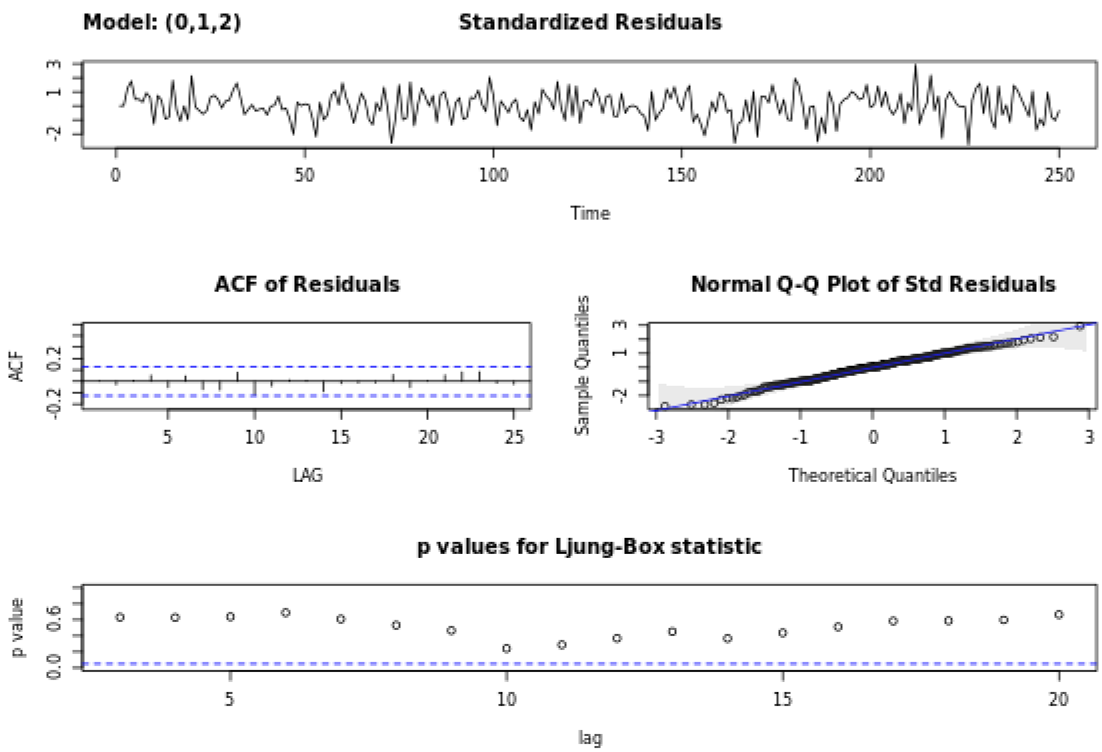


Fit an ARIMA(0,1,1) model to the simulated data using `sarima()`. Compare the MA parameter estimate to the actual value of .9, and examine the residual plots.

```
# Fit the first model, compare parameters, check diagnostics
sarima(x, p = 0, d = 1, q = 1)
```



# Fit the second model and compare fit
sarima(x, p = 0, d = 1, q = 2)



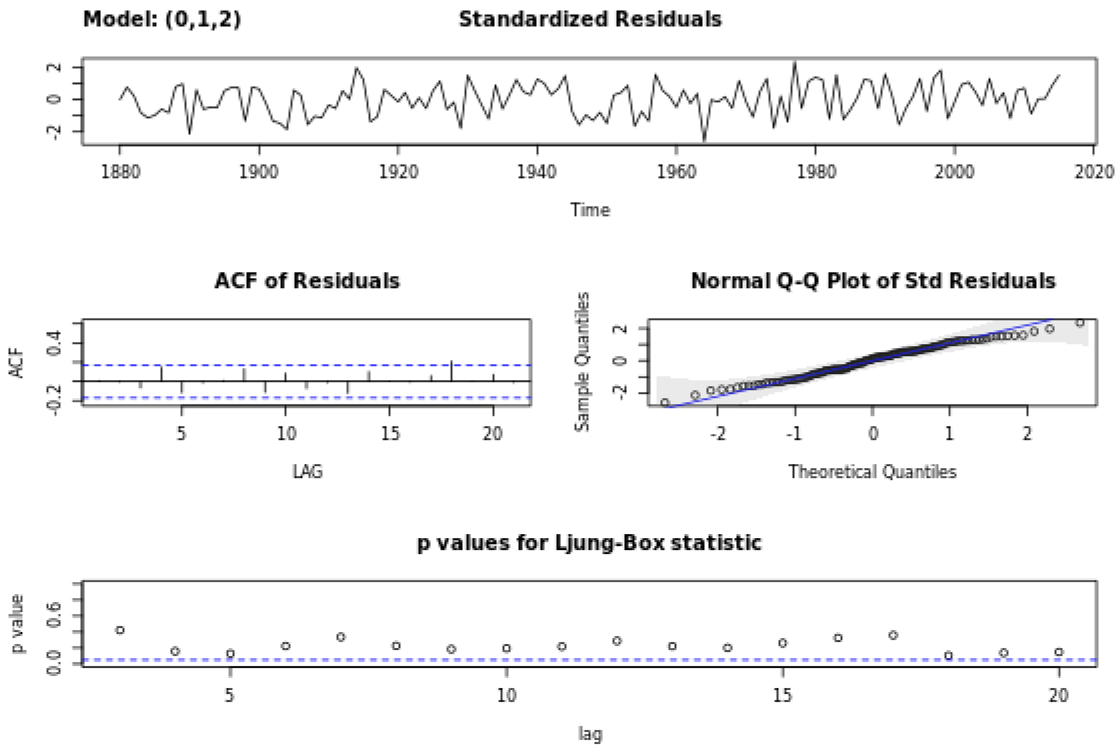
Diagnostics - Global Temperatures

You can now finish your analysis of global temperatures. Recall that you previously fit two models to the data in `globtemp`, an ARIMA(1,1,1) and an ARIMA(0,1,2). In the final analysis, check the residual diagnostics and use AIC and BIC for model choice. The data are plotted for you.

```
> globtemp
Time Series:
Start = 1880
End = 2015
Frequency = 1
  [1] -0.20 -0.11 -0.10 -0.20 -0.28 -0.31 -0.30 -0.33 -0.20 -0.11 -0.37 -0.24
 [13] -0.27 -0.30 -0.31 -0.22 -0.15 -0.11 -0.28 -0.16 -0.09 -0.15 -0.28 -0.36
 [25] -0.45 -0.28 -0.23 -0.40 -0.44 -0.47 -0.43 -0.44 -0.35 -0.35 -0.16 -0.11
 [37] -0.33 -0.40 -0.26 -0.23 -0.26 -0.21 -0.27 -0.24 -0.28 -0.20 -0.09 -0.20
 [49] -0.21 -0.36 -0.13 -0.09 -0.17 -0.28 -0.13 -0.19 -0.15 -0.02 -0.02 -0.03
 [61]  0.08  0.13  0.10  0.14  0.26  0.12 -0.03 -0.04 -0.09 -0.09 -0.17 -0.06
 [73]  0.01  0.08 -0.12 -0.14 -0.20  0.03  0.06  0.03 -0.03  0.05  0.02  0.06
 [85] -0.20 -0.10 -0.05 -0.02 -0.07  0.07  0.03 -0.09  0.01  0.15 -0.08 -0.01
 [97] -0.11  0.18  0.07  0.16  0.27  0.32  0.13  0.31  0.16  0.12  0.19  0.33
[109]  0.40  0.28  0.44  0.42  0.23  0.24  0.32  0.46  0.34  0.48  0.63  0.42
[121]  0.42  0.55  0.63  0.62  0.55  0.69  0.63  0.66  0.54  0.64  0.72  0.60
[133]  0.63  0.66  0.75  0.87
```

- Fit an ARIMA(0,1,2) model to `globtemp` and check the diagnositcs. What does the output tell you about the model?

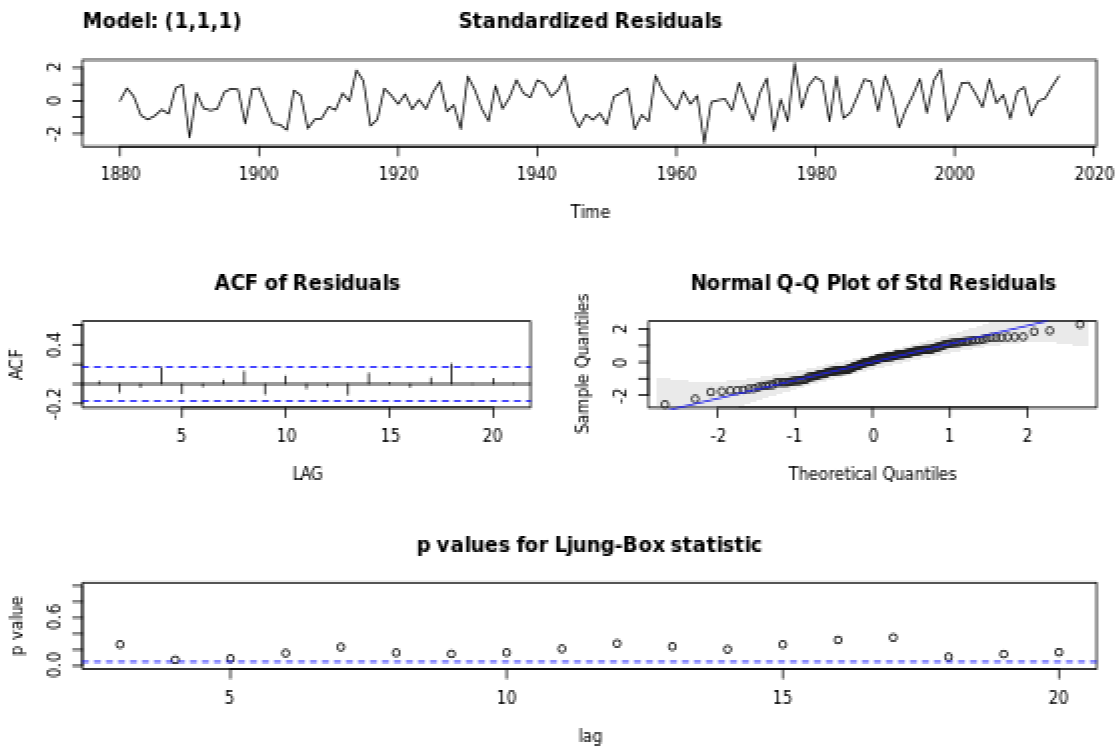
sarima(globtemp, p = 0, d = 1, q = 2)


Model: (0,1,2)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
$fit

Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = constant, optim.control = list(trace = trc, REPORT = 1,
    reltol = tol))

Coefficients:
          ma1      ma2  constant
      -0.3984  -0.2173    0.0072
s.e.   0.0808   0.0768    0.0033

sigma^2 estimated as 0.00982:  log likelihood = 120.32,  aic = -232.64

$degrees_of_freedom
[1] 132

$ttable
         Estimate     SE t.value p.value
ma1       -0.3984 0.0808 -4.9313  0.0000
ma2       -0.2173 0.0768 -2.8303  0.0054
constant   0.0072 0.0033  2.1463  0.0337

$AIC
[1] -3.579224

$AICc
[1] -3.562273

$BIC
[1] -4.514974
```

- Fit an ARIMA(1,1,1) model to `globtemp` and check the diagnostics.

sarima(globtemp, p = 1, d = 1, q = 1)


Model: (1,1,1)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
$fit

Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = constant, optim.control = list(trace = trc, REPORT = 1,
    reltol = tol))

Coefficients:
         ar1      ma1  constant
      0.3549  -0.7663    0.0072
s.e.  0.1314   0.0874    0.0032

sigma^2 estimated as 0.009885:  log likelihood = 119.88,  aic = -231.76

$degrees_of_freedom
[1] 132

$ttable
         Estimate     SE t.value p.value
ar1        0.3549 0.1314  2.7008  0.0078
ma1       -0.7663 0.0874 -8.7701  0.0000
constant   0.0072 0.0032  2.2738  0.0246

$AIC
[1] -3.572642

$AICc
[1] -3.555691

$BIC
[1] -4.508392
```

- Which is the better model? Type your answer into the blanks in your R workspace (ex. either `ARIMA(0,1,2)` or `ARIMA(1,1,1)`).

"ARIMA(0,1,2)"

Your model diagnostics suggest that both the ARIMA(0,1,2) and the ARIMA(1,1,1) are reasonable models. However, the AIC and BIC suggest that the ARIMA(0,1,2) performs slightly better, so this should be your preferred model. Although you were not asked to do so, you can use overfitting to assess the final model. For example, try fitting an ARIMA(1,1,2) or an ARIMA(0,1,3) to the data.

Forecasting Simulated ARIMA

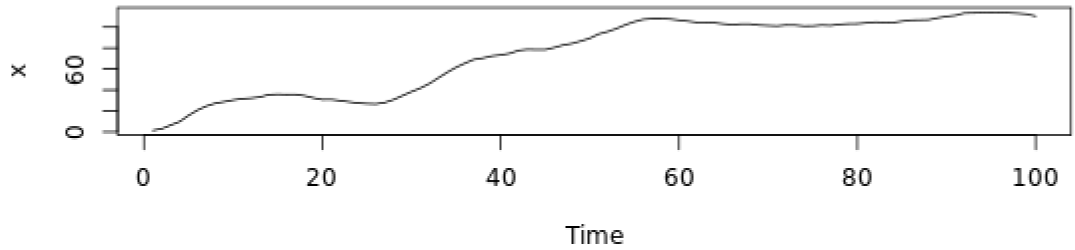
Now that you are an expert at fitting ARIMA models, you can use your skills for forecasting. First, you will work with simulated data.

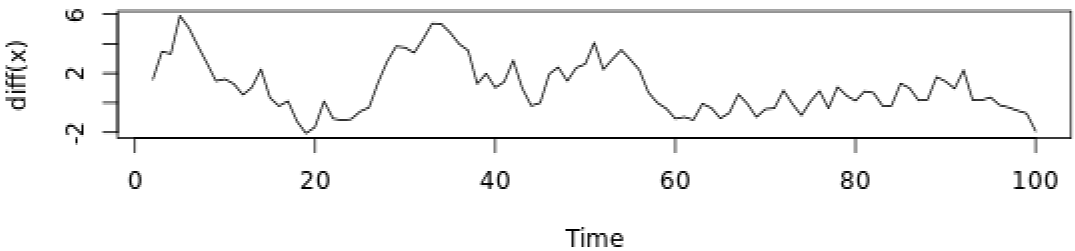
We generated 120 observations from an ARIMA(1,1,0) model with AR parameter .9. The data are in `y` and the first 100 observations are in `x`. These observations are plotted for you. You will fit an ARIMA(1,1,0) model to the data in `x` and verify that the model fits well. Then use **sarima.for()** from `astsa` to forecast the data 20 time periods ahead. You will then compare the forecasts to the actual data in `y`.

The basic syntax for forecasting is `sarima.for(data, n.ahead, p, d, q)` where `n.ahead` is a positive integer that specifies the forecast horizon. The predicted values and their standard errors are printed, the data are plotted in black, and the forecasts are in red along with 2 mean square prediction error bounds as blue dashed lines.

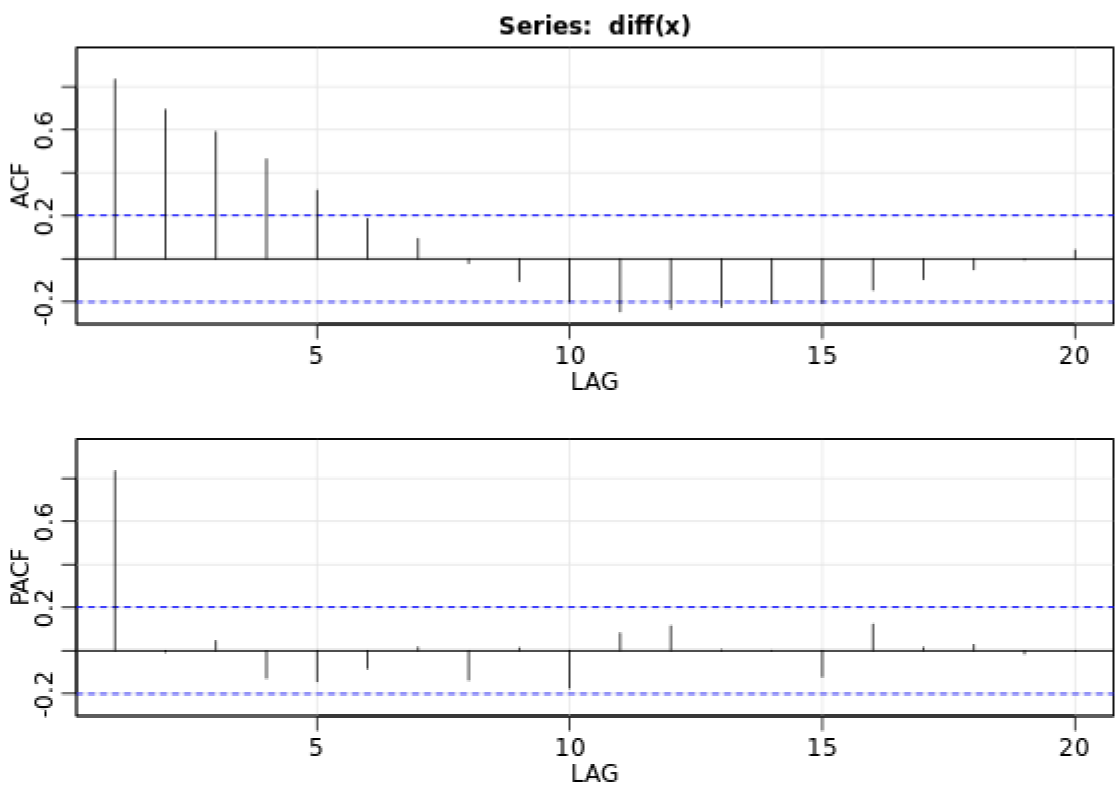
The **astsa** package is preloaded and the data (`x`) and differenced data (`diff(x)`) are plotted.

```
> x
Time Series:
Start = 1
End = 100
Frequency = 1
  [1]  1.475311  3.060689  6.530352  9.844334 15.734869 20.797598
  [7] 24.634701 27.322191 28.793491 30.399593 31.672226 32.209409
 [13] 33.254600 35.529516 35.870378 35.649949 35.766433 34.509227
 [19] 32.438305 30.803689 30.912875 29.845304 28.667229 27.554773
 [25] 26.962435 26.649197 28.018219 30.804234 34.624526 38.363188
 [31] 41.745341 46.059446 51.431467 56.778215 61.528616 65.510082
```



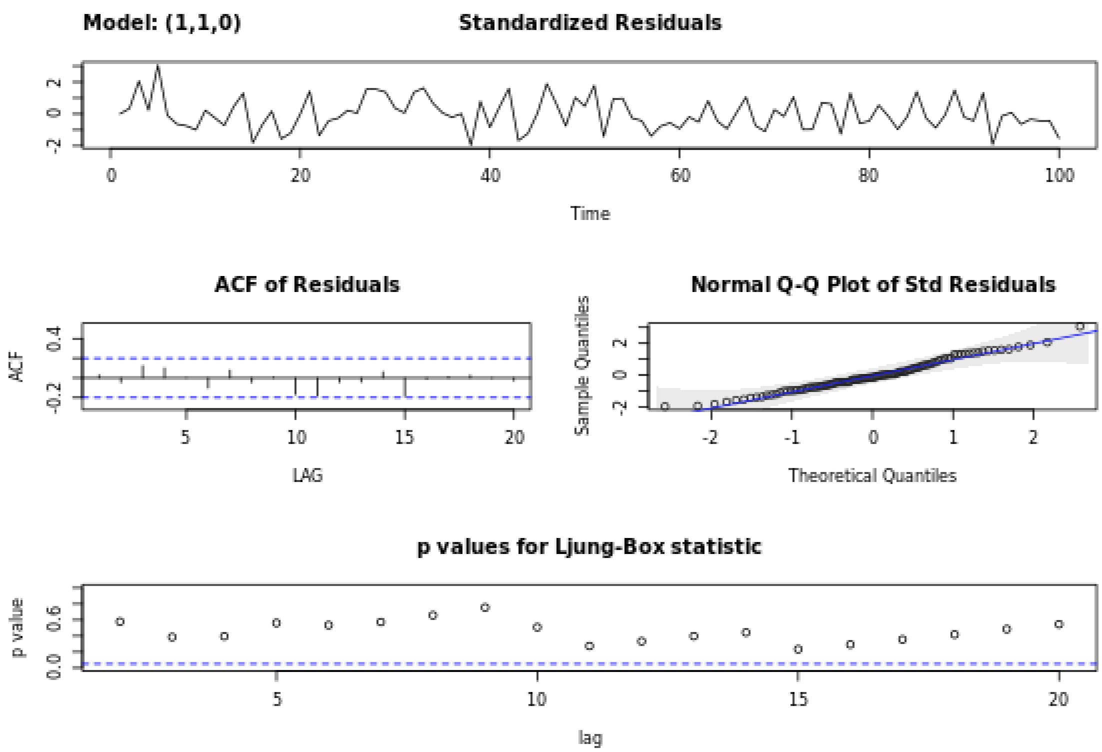



```
# Plot P/ACF pair of differenced data
acf2(diff(x))
```

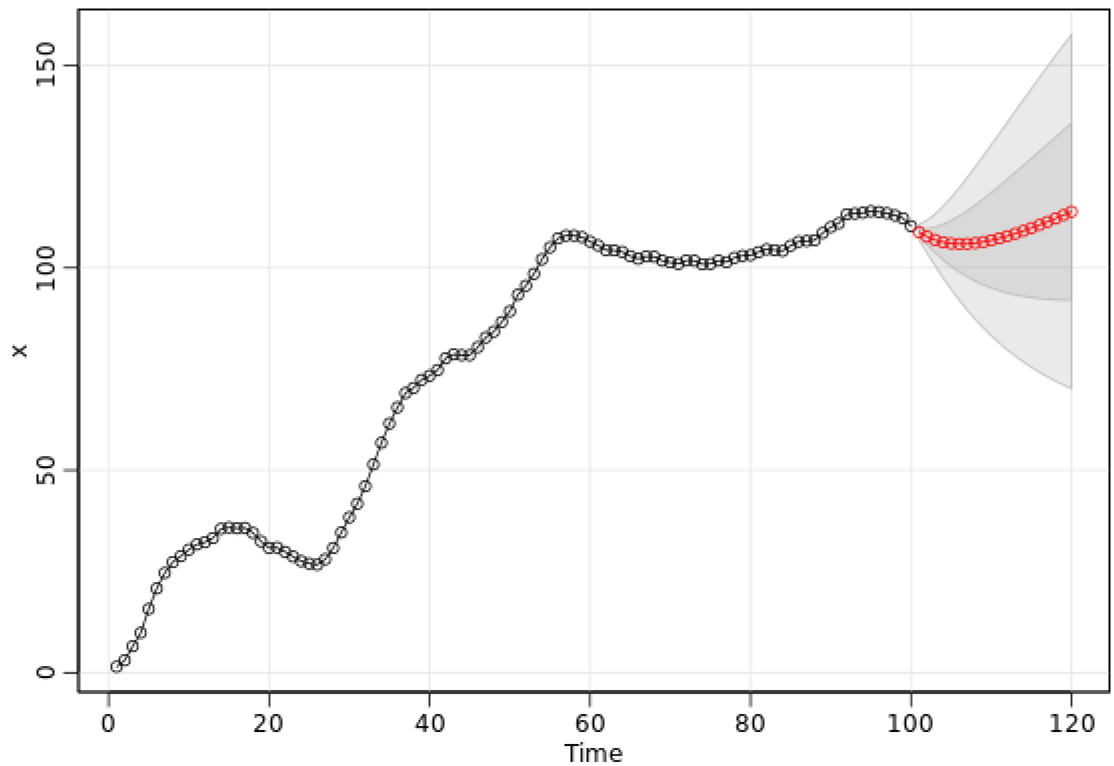


Fit model - check t-table and diagnostics

sarima(x, p = 1, d = 1, q = 0)



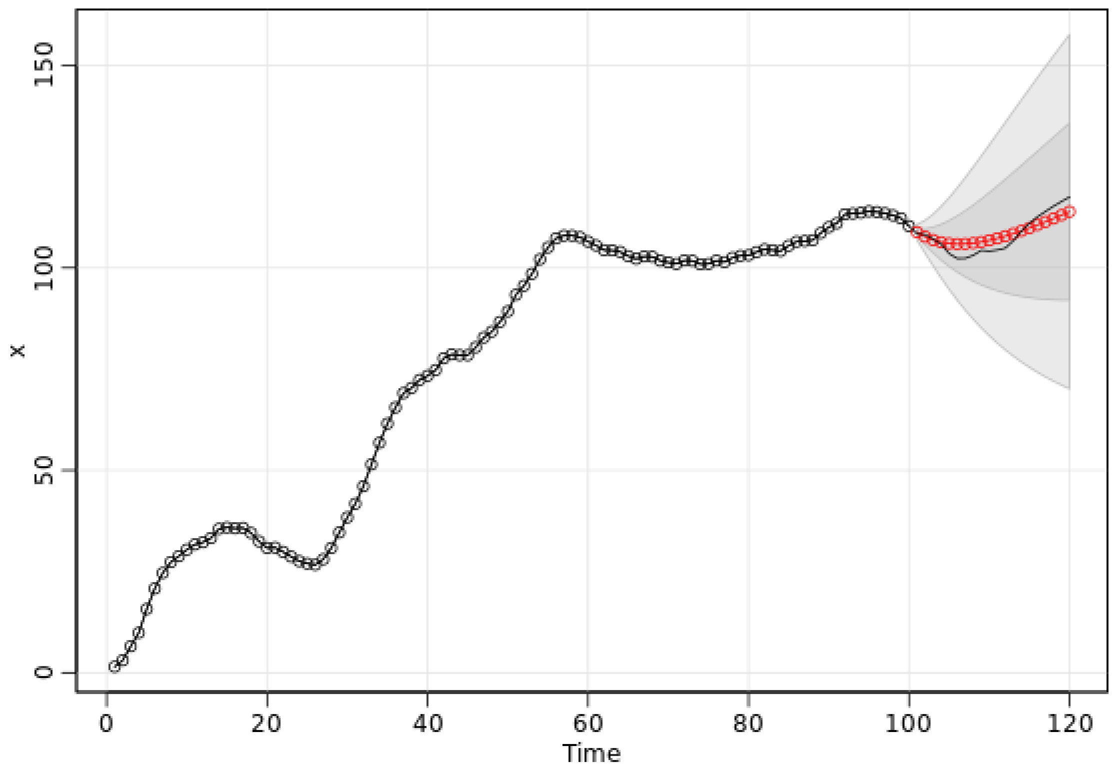
```
# Forecast the data 20 time periods ahead
sarima.for(x, n.ahead = 20, p = 1, d = 1, q = 0)
```



Y is actual value

```
> y
Time Series:
Start = 1
End = 120
Frequency = 1
  [1]  1.475311   3.060689   6.530352   9.844334  15.734869  20.797598
  [7] 24.634701  27.322191  28.793491  30.399593  31.672226  32.209409
 [13] 33.254600  35.529516  35.870378  35.649949  35.766433  34.509227
 [19] 32.438305  30.803689  30.912875  29.845304  28.667229  27.554773
 [25] 26.962435  26.649197  28.018219  30.804234  34.624526  38.363188
 [31] 41.745341  46.059446  51.431467  56.778215  61.528616  65.510082
 [37] 69.053555  70.331715  72.317919  73.340685  74.755766  77.632179
 [43] 78.617924  78.418521  78.412263  80.361745  82.771127  84.240479
```

```
lines(y)
```



the `sarima.for()` command provides a simple method for forecasting. Although the blue error bands are relatively wide, the prediction remains quite valuable.
Your model diagnostics suggest that both the ARIMA(0,1,2) and the ARIMA(1,1,1) are reasonable models. However, the AIC and BIC suggest that the ARIMA(0,1,2) performs slightly better, so this should be your preferred model. Although you were not asked to do so, you can use overfitting to assess the final model. For example, try fitting an ARIMA(1,1,2) or an ARIMA(0,1,3) to the data.

Forecasting Global Temperatures

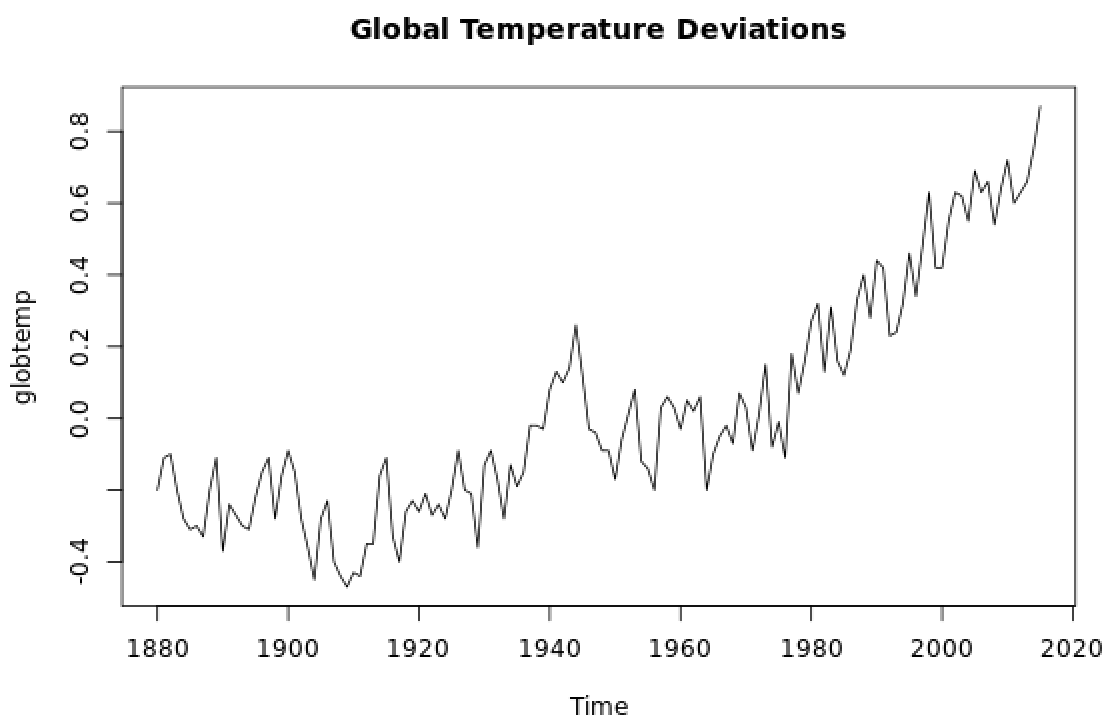
Now you can try forecasting real data.

Here, you will forecast the annual global temperature deviations `globtemp` to 2050. Recall that in previous exercises, you fit an ARIMA(0,1,2) model to the data. You will refit the model to confirm it, and then forecast the series 35 years into the future.

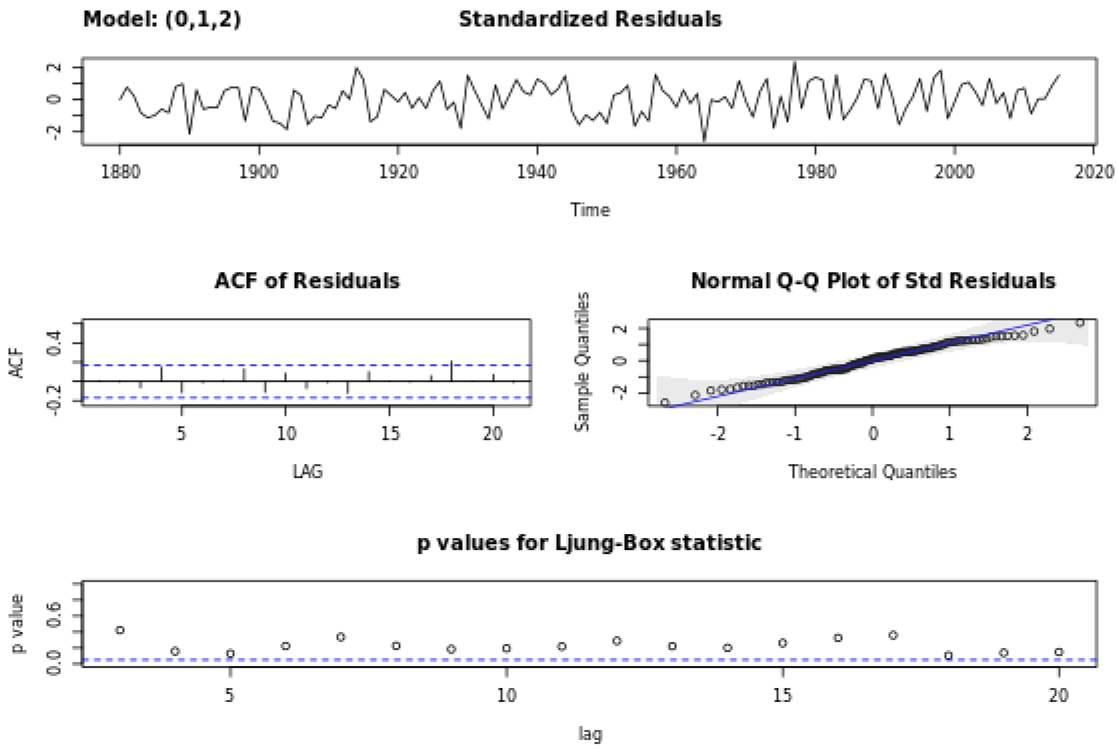
The astsa package is preloaded and the data are plotted.

Instructions 0XP

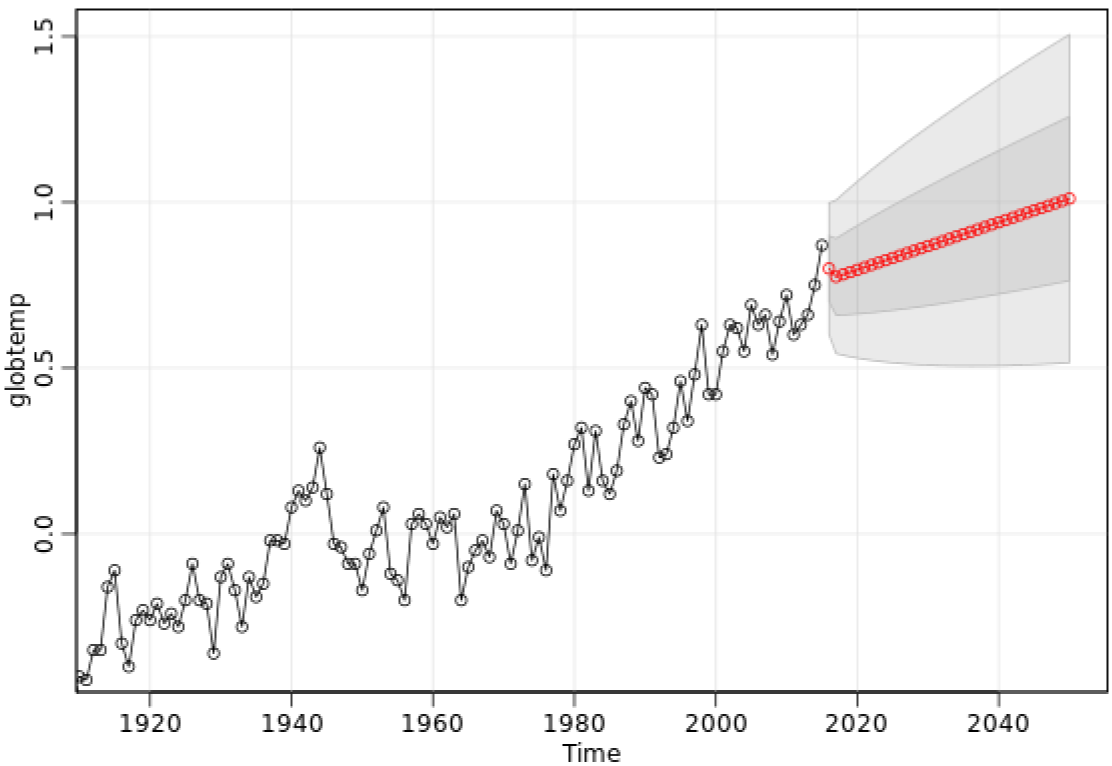
- Fit an ARIMA(0,1,2) model to the data using `sarima()`. Based on your previous analysis this was the best model for the `globtemp` data. Recheck the parameter significance in the t-table output and check the residuals for any departures from the model assumptions.
- Use `sarima.for()` to forceast your global temperature data 35 years ahead to 2050 using the ARIMA(0,1,2) fit.




```
# Fit an ARIMA(0,1,2) to globtemp and check the fit
sarima(globtemp, p = 0, d = 1, q = 2)
```




```
# Forecast data 35 years into the future
sarima.for(globtemp, n.ahead = 35, p = 0, d = 1, q = 2)
```



globtemp
Time
1.5
1.0
0.5
0.0
1920
1940
1960
1980
2000
2020
2040

P/ACF of Pure Seasonal Models

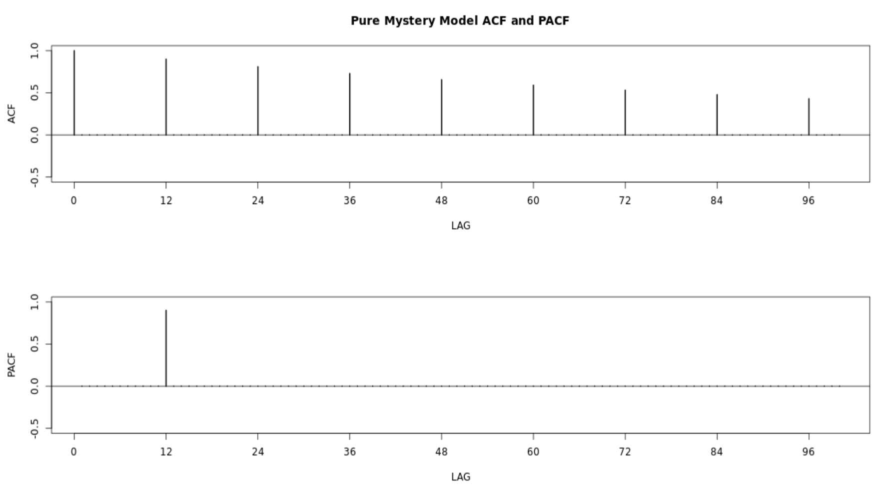
In the video, you saw that a pure seasonal ARMA time series is correlated at the seasonal lags only. Consequently, the ACF and PACF behave as the nonseasonal counterparts, but at the seasonal lags, 1S, 2S, ..., where S is the seasonal period (S = 12 for monthly data). As in the nonseasonal case, you have the pure seasonal table:

Behavior of the ACF and PACF for Pure SARMA Models

	$AR(P)_S$	$MA(Q)_S$	$ARMA(P,Q)_S$
ACF*	Tails off at seasonal lags	Cuts off after lag QS	Tails off at seasonal lags
PACF*	Cuts off after lag PS	Tails off at seasonal lags	Tails off at seasonal lags

*The values at nonseasonal lags are zero.

We have plotted the true ACF and PACF of a pure seasonal model. Identify the model with the following abbreviations $SAR(P)_S$, $SMA(Q)_S$, or $SARMA(P,Q)_S$ for the pure seasonal AR, MA or ARMA with seasonal period S, respectively.



Instructions

Possible Answers

- ( ) $SARMA(1,1)$_{12}$
- (x) $SAR(1)$_{12}$
- ( ) $SMA(1)$_{12}$
- ( ) $SMA(96)$_{12}$

The ACF tails off at seasonal lags, while the PACF cuts off after lag 12. This fits with a $SAR(1)_{12}$ model.

Fit a Pure Seasonal Model

As with other models, you can fit seasonal models in R using the `sarima()` command in the astsapackage.

To get a feeling of how pure seasonal models work, it is best to consider simulated data. We generated 250 observations from a pure seasonal model given by

$$X_t = .9X_{t-12} + W_t + .5W_{t-12} \,,$$

which we would denote as a SARMA(P = 1, Q = 1)$_{S=12}$. Three years of data and the model ACF and PACF are plotted for you.

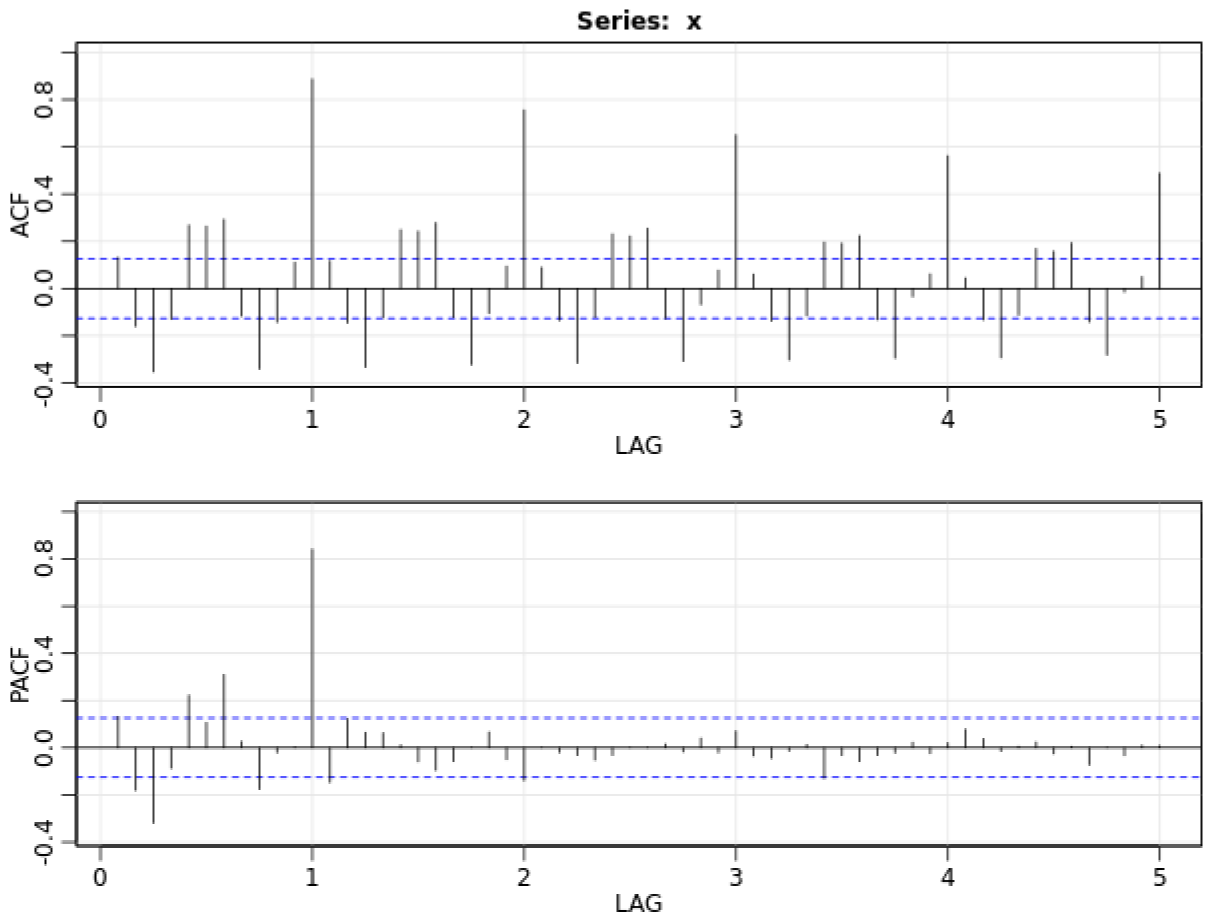
You will compare the sample ACF and PACF values from the generated data to the true values displayed to the right.

The `astsa` package is preloaded for you and the generated data are in `x`.

```
> x
          Jan         Feb         Mar         Apr         May
1  -3.06314411 -1.99732239 -3.92496516  5.37037490  7.47036620
2  -1.94517424 -1.88141894 -4.78283163  4.36126203  7.15892395
3  -1.30804812 -0.57269109 -5.37028066  3.05287812  7.74906411
4  -0.93810668 -0.75326739 -5.05940526  3.34599676  7.31869879
5  -0.79463200 -0.31968591 -4.60688949  2.94661301  6.47853409
6  -0.56086720 -0.34570051 -2.93266020  2.52539869  5.87612461
7  -0.09721245  0.89284772 -2.44650750  2.86850932  4.52193530
8  -0.45354683  2.78581592 -4.90832317  2.90947878  3.65013089
9  -0.62655684  2.90431094 -6.02331361  1.97552385  3.74475417
10 -0.80121799  2.66898321 -3.86649212  3.52588565  3.60995411
11 -0.78374543  2.85817994 -3.76443694  3.88465659  2.72543099
12 -2.74152365  1.98987065 -2.82825691  4.16920100  0.75300449
13 -3.01142452  0.56854952 -3.25466215  2.01180399 -0.39628110
14 -3.10174913  1.35999795 -2.61130287 -0.10901335  1.38760723
15 -5.05410924  1.69803266 -2.62143873 -1.53876681  1.80156657
16 -6.07191415  2.37434930 -2.65873047 -2.09769900  0.72198305
17 -6.85442291  1.90994500 -3.20517897 -1.13947532  0.58096728
18 -7.43420378  0.60211727 -4.28772788 -1.82473745 -0.24184375
19 -6.40423903  0.58476832 -3.07473338 -3.81205212 -2.48378311
20 -5.93375792  1.55061161 -1.28834140 -3.31239369 -3.32106881
21 -6.45289830  0.99855020 -0.47251819 -2.44178033 -3.73976633
          Jun         Jul         Aug         Sep         Oct
1   0.50232826  2.47700477 -10.09300993 -3.46214142  1.83489844
2   2.69850985  0.23682717 -9.93283704 -3.40560211  0.71845070
3   3.92561266 -0.35442075 -10.32638735 -1.30215365  1.79631778
4   2.80153053  0.23648769 -9.54099616 -1.46635141  3.82921216
5   0.40265389  0.41278955 -8.06925619 -2.51177993  4.10457260
6  -1.37413176 -0.83308197 -8.19281706 -1.46459077  5.50166257
```

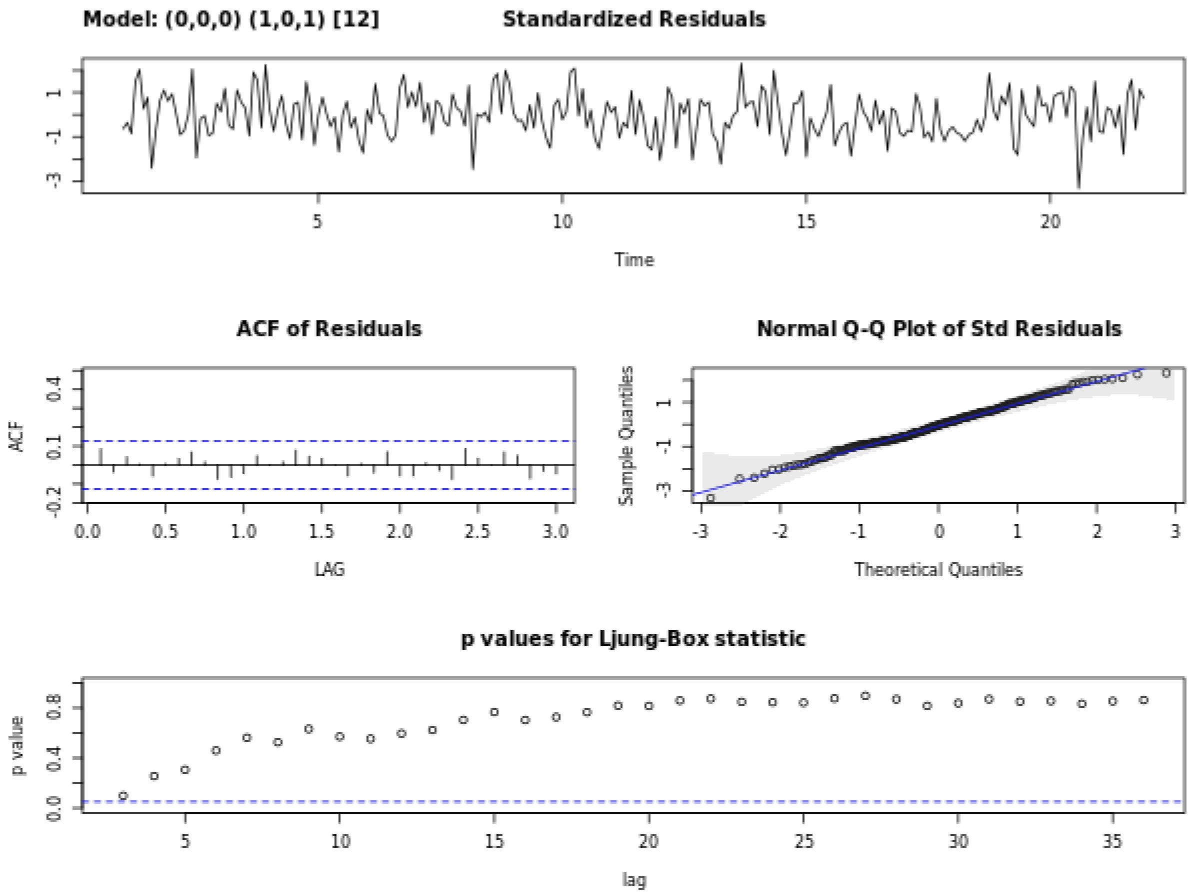
- Use `acf2()` to plot the sample ACF and PACF of the generated data to lag 60 and compare to actual values. To estimate to lag 60, set the `max.lag` argument equal to `60`.

```
# Plot sample P/ACF to lag 60 and compare to the true values
acf2(x, max.lag = 60)
```



- In addition to the `p`, `d`, and `q` arguments in your `sarima()` command, specify `P`, `D`, `Q`, and `S` (note that R is case sensitive).

# Fit the seasonal model to x # For example, sarima(x,2,1,0) will fit an ARIMA(2,1,0) model to the series in x, and sarima(x,2,1,0,0,1,1,12) will fit a seasonal ARIMA$(2,1,0)*(0,1,1)_{12}$ model to the series in x.
sarima(x, p = 0, d = 0, q = 0, P = 1, D = 0, Q = 1, S = 12)


Model: (0,0,0) (1,0,1) [12]
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = list(trace = trc,
    REPORT = 1, reltol = tol))

Coefficients:
        sar1    sma1    xmean
      0.9311  0.4824  -0.5766
s.e.  0.0204  0.0633   0.8797

sigma^2 estimated as 0.9767:  log likelihood = -372.74,  aic = 753.48

$degrees_of_freedom
[1] 249

$ttable
      Estimate     SE t.value p.value
sar1    0.9311 0.0204 45.6189  0.0000
sma1    0.4824 0.0633  7.6158  0.0000
xmean  -0.5766 0.8797 -0.6554  0.5128

$AIC
[1] 1.000217

$AICc
[1] 1.008796

$BIC
[1] 0.04223375
```

Fit a Mixed Seasonal Model

Pure seasonal dependence such as that explored earlier in this chapter is relatively rare. Most seasonal time series have *mixed* dependence, meaning only some of the variation is explained by seasonal trends.

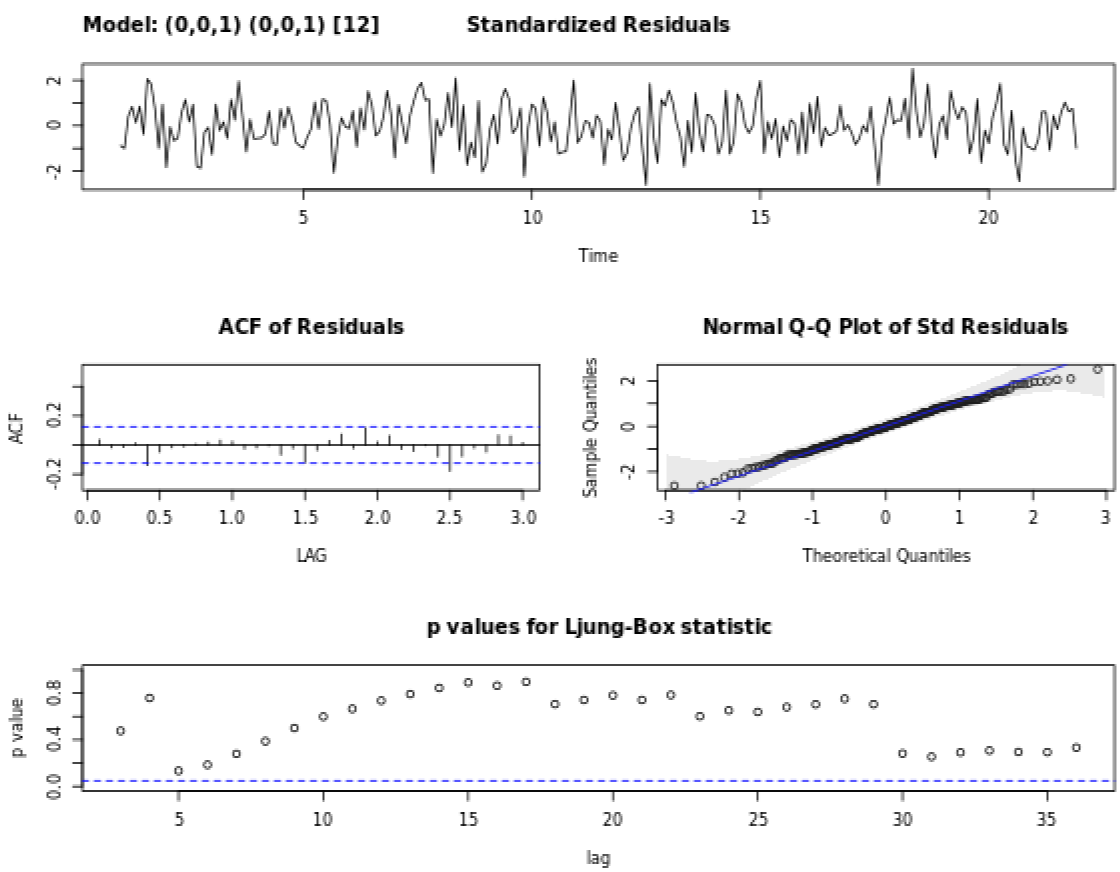
Recall that the full seasonal model is denoted by SARIMA(p,d,q)x$(P,D,Q)_S$ where capital letters denote the seasonal orders.

As before, this exercise asks you to compare the sample P/ACF pair to the true values for some simulated seasonal data and fit a model to the data using `sarima()`. This time, the simulated data come from a mixed seasonal model, SARIMA(0,0,1)x$(0,0,1)_{12}$. The plots on the right show three years of data, as well as the model ACF and PACF. Notice that, as opposed to the pure seasonal model, there are correlations at the nonseasonal lags as well as the seasonal lags.

As always, the astsa package is preloaded. The generated data are in `x`.

- Fit the model to generated data (`x`) using `sarima()`. As in the previous exercise, be sure to specify the additional seasonal arguments in your `sarima()` command.

```
# Fit the seasonal model to x
sarima(x, p = 0, d = 0, q = 1, P = 0, D = 0, Q = 1, S = 12)
```



Data Analysis - Unemployment I

In the video, we fit a seasonal ARIMA model to the log of the monthly `AirPassengers` data set. You will now start to fit a seasonal ARIMA model to the monthly US unemployment data, `unemp`, from the `astsa` package.

The first thing to do is to plot the data, notice the trend and the seasonal persistence. Then look at the detrended data and remove the seasonal persistence. After that, the fully differenced data should look stationary.

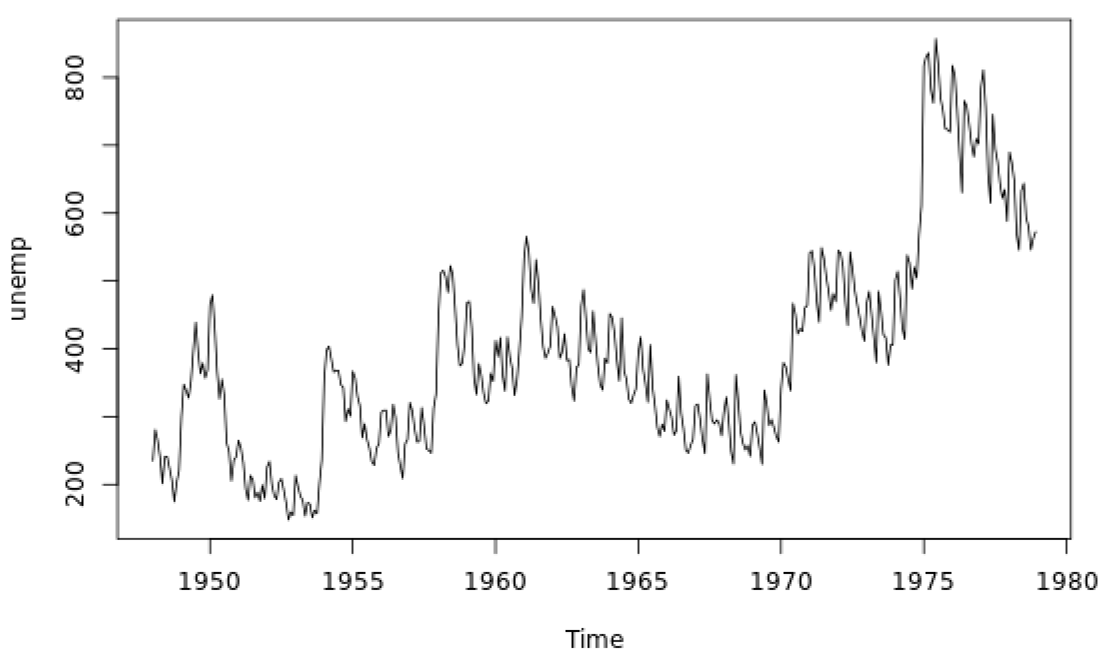
The [astsa](#) package is preloaded in your workspace.

```
> unemp
       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
1948 235.1 280.7 264.6 240.7 201.4 240.8 241.1 223.8 206.1 174.7 203.3 220.5
1949 299.5 347.4 338.3 327.7 351.6 396.6 438.8 395.6 363.5 378.8 357.0 369.0
1950 464.8 479.1 431.3 366.5 326.3 355.1 331.6 261.3 249.0 205.5 235.6 240.9
1951 264.9 253.8 232.3 193.8 177.0 213.2 207.2 180.6 188.6 175.4 199.0 179.6
1952 225.8 234.0 200.2 183.6 178.2 203.2 208.5 191.8 172.8 148.0 159.4 154.5
1953 213.2 196.4 182.8 176.4 153.6 173.2 171.0 151.2 161.9 157.2 201.7 236.4
1954 356.1 398.3 403.7 384.6 365.8 368.1 367.9 347.0 343.3 292.9 311.5 300.9
1955 366.9 356.9 329.7 316.2 269.0 289.3 266.2 253.6 233.8 228.4 253.6 260.1
1956 306.6 309.2 309.5 271.0 279.9 317.9 298.4 246.7 227.3 209.1 259.9 266.0
1957 320.6 308.5 282.2 262.7 263.5 313.1 284.3 252.6 250.3 246.5 312.7 333.2
1958 446.4 511.6 515.5 506.4 483.2 522.3 509.8 460.7 405.8 375.0 378.5 406.8
1959 467.8 469.8 429.8 355.8 332.7 378.0 360.5 334.7 319.5 323.1 363.6 352.1
1960 411.9 388.6 416.4 360.7 338.0 417.2 388.4 371.1 331.5 353.7 396.7 447.0
1961 533.5 565.4 542.3 488.7 467.1 531.3 496.1 444.0 403.4 386.3 394.1 404.1
```

```
> class(unemp)
[1] "ts"
```

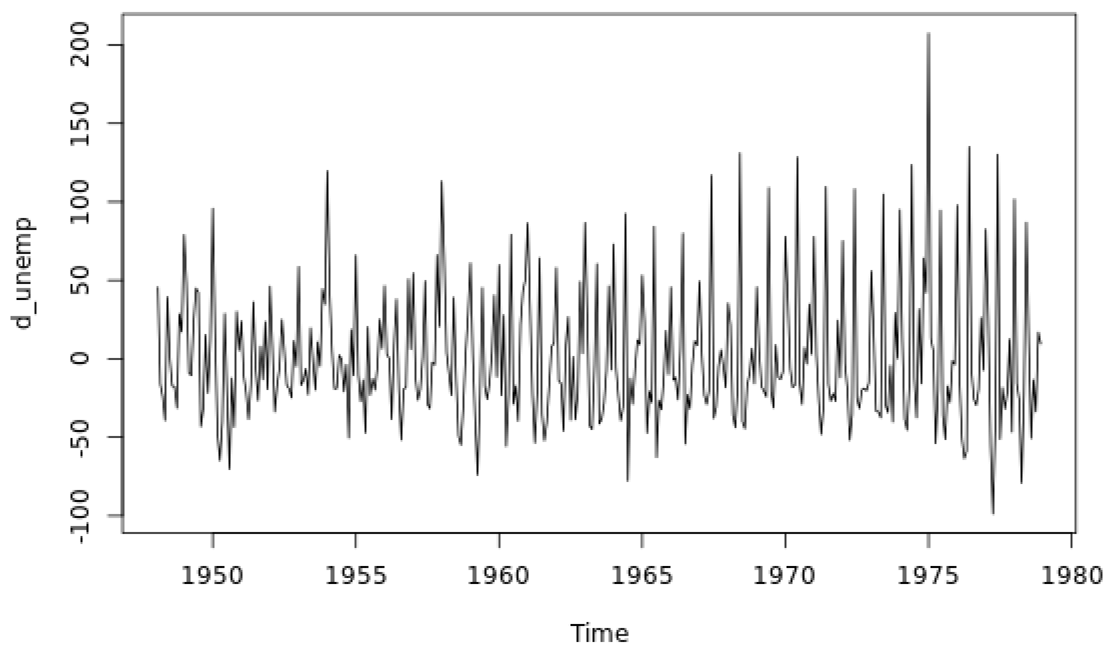
- Plot the monthly US unemployment (`unemp`) time series from `astsa`. Note trend and seasonality.

plot(unemp)



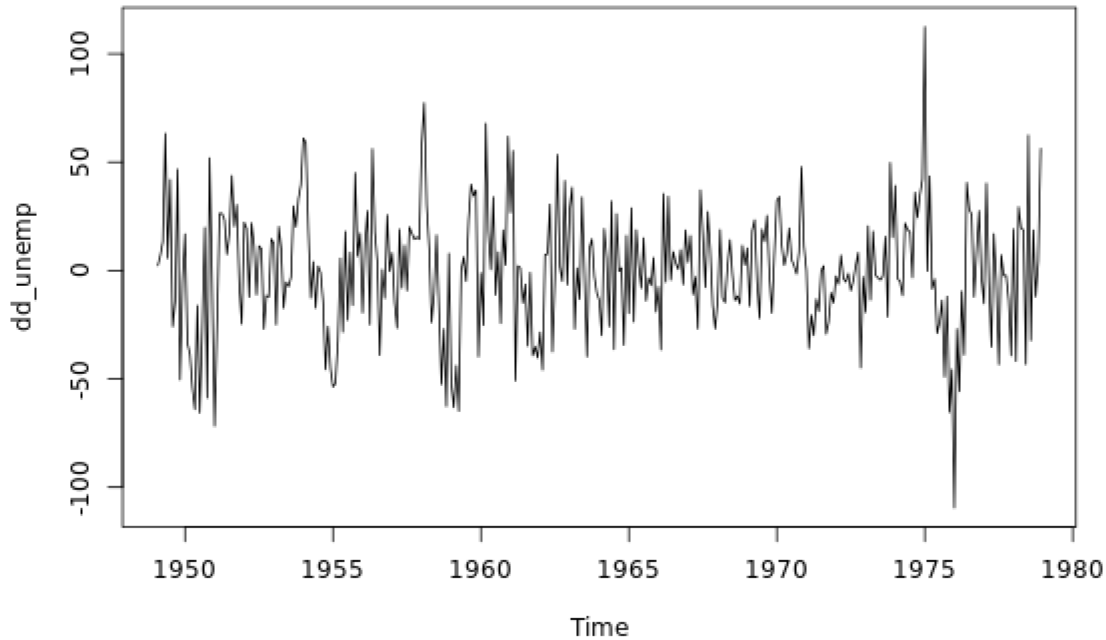
- Detrend and plot the data. Save this as `d_unemp`. Notice the seasonal persistence.

```
d_unemp <- diff(unemp)

plot(d_unemp)
```



- Seasonally difference the detrended series and save this as `dd_unemp`. Plot this new data and notice that it looks stationary now.

```
# Plot seasonal differenced diff_unemp
dd_unemp <- diff(d_unemp, lag = 12)
plot(dd_unemp)
```



Data Analysis - Unemployment II

Now, you will continue fitting an SARIMA model to the monthly US unemployment `unemp` time series by looking at the sample ACF and PACF of the fully differenced series.

Note that the lag axis in the sample P/ACF plot is in terms of years. Thus, lags 1, 2, 3, ... represent 1 year (12 months), 2 years (24 months), 3 years (36 months), ...

Once again, the **astsa** package is preloaded in your workspace.

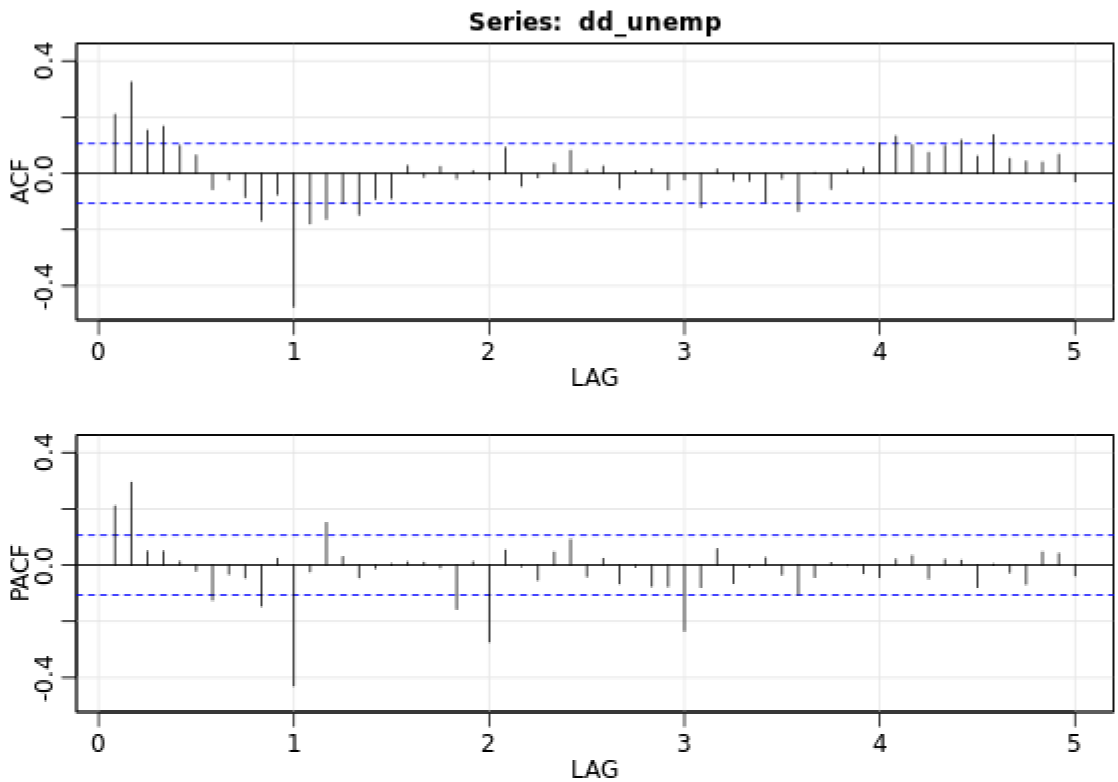
- Difference the data fully (as in the previous exercise) and plot the sample ACF and PACF of the transformed data to lag 60 months (5 years). Consider that, for

- the nonseasonal component: the PACF cuts off at lag 2 and the ACF tails off.
- the seasonal component: the ACF cuts off at lag 12 and the PACF tails off at lags 12, 24, 36, …

Plot P/ACF pair of the fully differenced data to lag 60

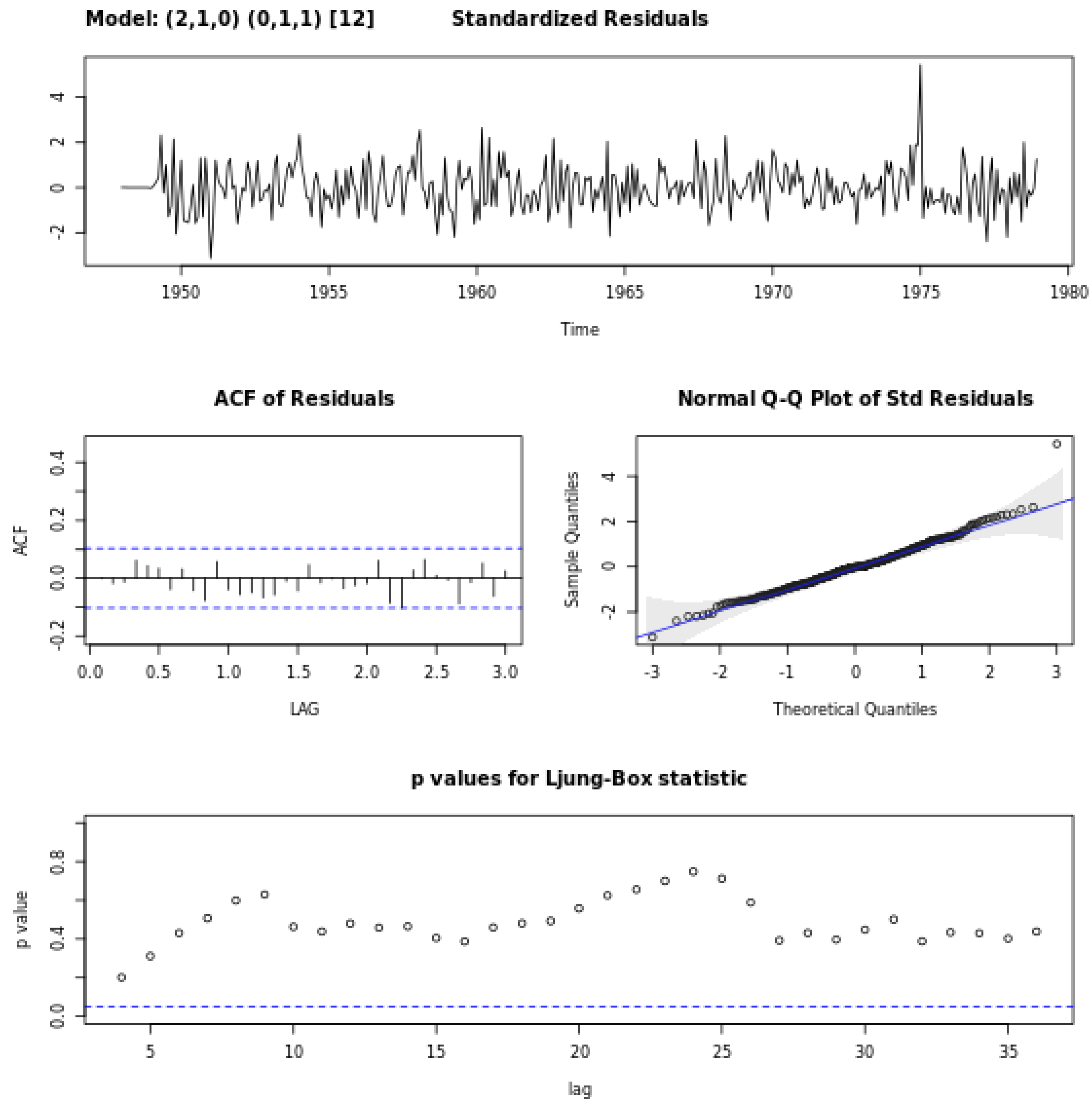
dd_unemp <- diff(diff(unemp), lag = 12)

acf2(dd_unemp, max.lag = 60)



- Suggest and fit a model using `sarima()`. Check the residuals to ensure appropriate model fit.

sarima(unemp, p = 2, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12)



Model: (2,1,0) (0,1,1) [12]
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q-Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung-Box statistic
p value
lag

```
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
    REPORT = 1, reltol = tol))

Coefficients:
         ar1     ar2     sma1
      0.1351  0.2464  -0.6953
s.e.  0.0513  0.0515   0.0381

sigma^2 estimated as 449.6:  log likelihood = -1609.91,  aic = 3227.81

$degrees_of_freedom
[1] 356

$ttable
     Estimate     SE  t.value p.value
ar1    0.1351 0.0513   2.6326  0.0088
ar2    0.2464 0.0515   4.7795  0.0000
sma1  -0.6953 0.0381 -18.2362  0.0000

$AIC
[1] 7.12457

$AICc
[1] 7.130239

$BIC
[1] 6.156174
```

Practice 2 **Data Analysis - Commodity Prices**

Making money in commodities is not easy. Most commodities traders lose money rather than make it. The package `astsa` includes the data set `chicken`, which is the monthly whole bird spot price, Georgia docks, US cents per pound, from August, 2001 to July, 2016.

The **astsa** package is preloaded in your R console and the data are plotted for you, note the trend and seasonal components.

```
> chicken
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
2001                                                 65.58  66.48  65.70
2002  62.94  62.92  62.73  62.50  63.35  63.80  64.21  64.11  64.04  63.00
2003  62.27  63.13  63.86  63.53  64.60  65.99  67.50  68.50  69.23  68.57
2004  69.58  71.59  73.09  74.75  76.59  79.63  80.94  80.10  78.16  76.00
2005  73.44  73.75  73.88  74.00  74.29  74.48  74.75  74.77  75.19  74.38
2006  69.86  69.18  68.29  67.52  67.87  68.98  69.90  70.42  70.69  69.65
2007  71.33  73.77  76.37  78.10  79.52  80.75  81.17  81.27  81.55  79.75
2008  77.25  79.15  81.23  82.04  83.46  85.71  88.25  88.42  88.40  87.54
2009  87.25  86.70  85.73  85.38  86.96  88.17  88.56  86.77  84.88  82.85
2010  83.04  83.30  84.00  85.28  86.45  87.17  87.84  87.79  87.75  86.73
2011  85.00  85.07  86.08  86.40  86.54  86.94  87.34  88.13  88.98  89.00
2012  90.35  91.17  92.79  93.25  94.06  94.50  94.73  95.02  95.65  95.84
2013  99.12 100.17 101.46 102.56 104.10 105.54 106.41 106.50 106.19 105.07
2014 104.40 104.50 105.25 107.27 108.79 110.88 112.60 112.79 113.52 113.89
2015 114.10 113.77 114.27 114.88 115.96 116.00 115.90 115.46 115.00 114.27
2016 112.52 112.10 111.56 111.55 111.98 111.84 111.46
        Nov    Dec
2001  64.33  63.23
2002  61.90  61.49
2003  68.36  68.98
2004  74.71  73.60
2005  72.69  71.21
```

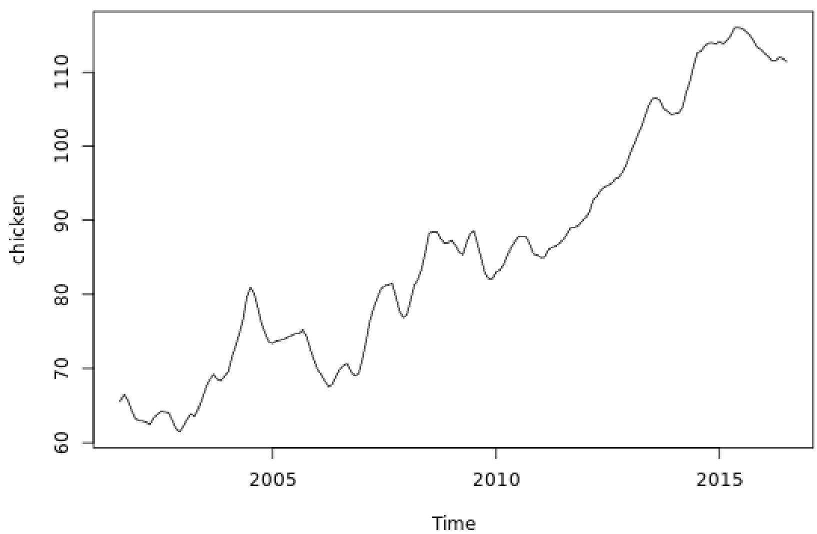
First, you will use your skills to carefully fit an SARIMA model to the commodity. Later, you will use the fitted model to try and forecast the whole bird spot price.

After removing the trend, the sample ACF and PACF suggest an AR(2) model because the PACF cuts off after lag 2 and the ACF tails off. However, the ACF has a small seasonal component remaining. This can be taken care of by fitting an addition SAR(1) component.

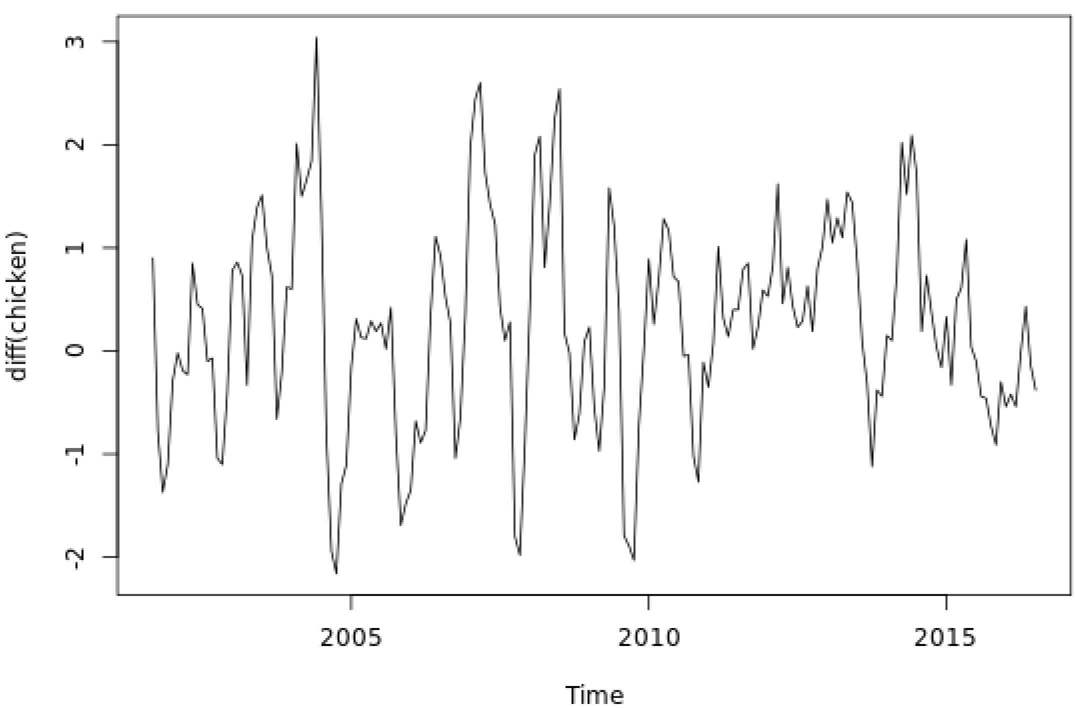
By the way, if you are interested in analyzing other commodities from various regions, you can find many different time series at index mundi.

- Plot the differenced (d = 1) data `diff(chicken)`. Note that the trend is removed and note the seasonal behavior.

plot(chicken)

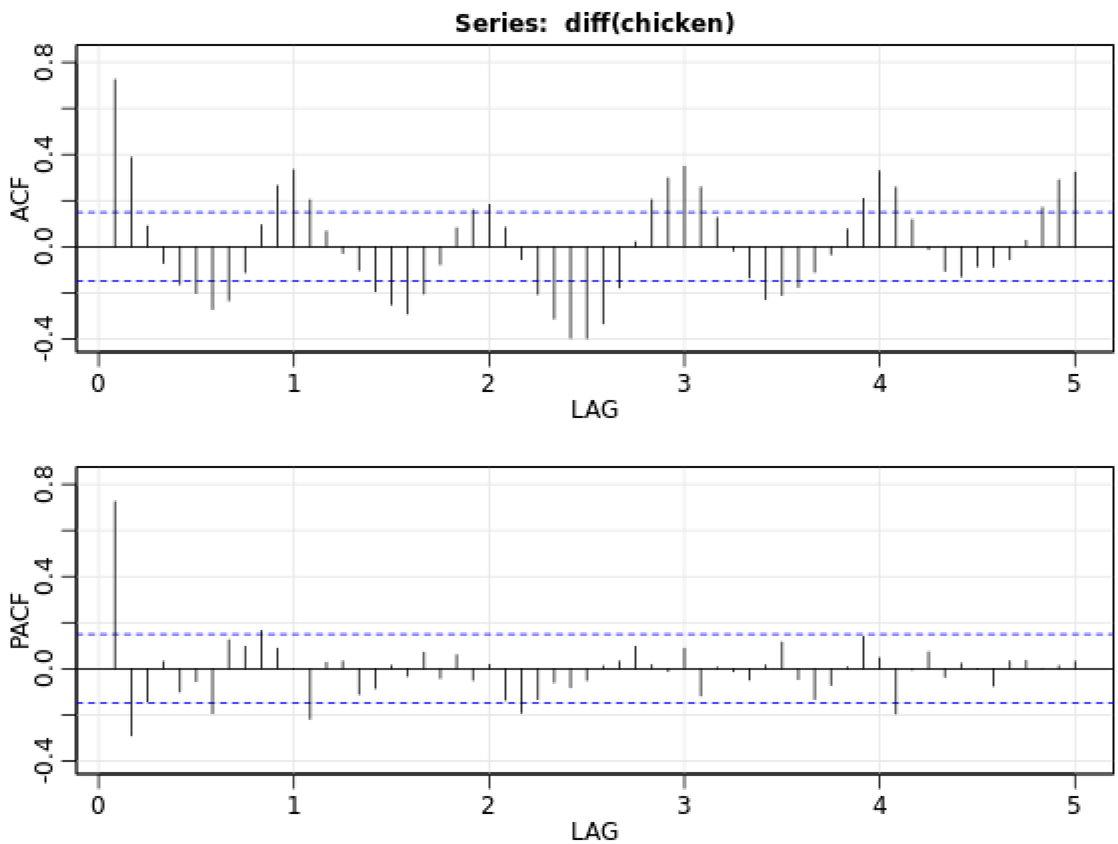


plot(diff(chicken))



- Plot the sample ACF and PACF of the differenced data to lag 60 (5 years). Notice that an AR(2) seems appropriate but there is a small but significant seasonal component remaining in the detrended data.

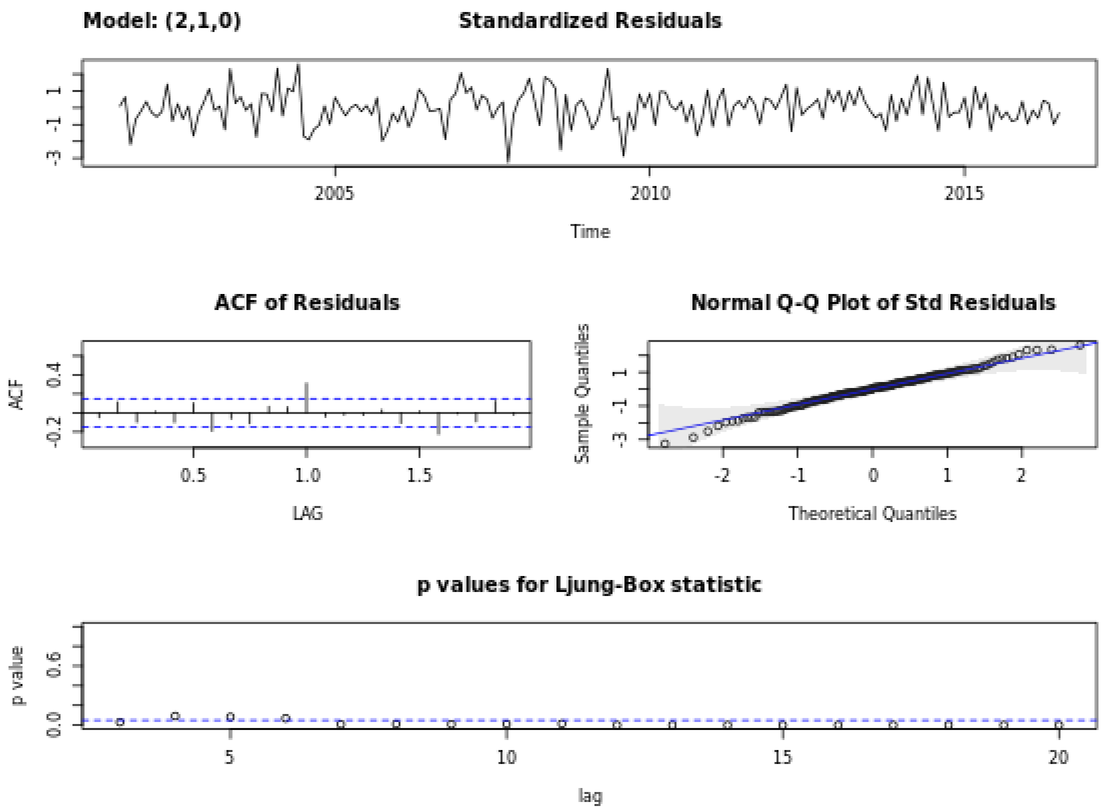
acf2(diff(chicken), max.lag = 60)



- Fit an ARIMA(2,1,0) to the `chicken` data to see that there is correlation remaining in the residuals.

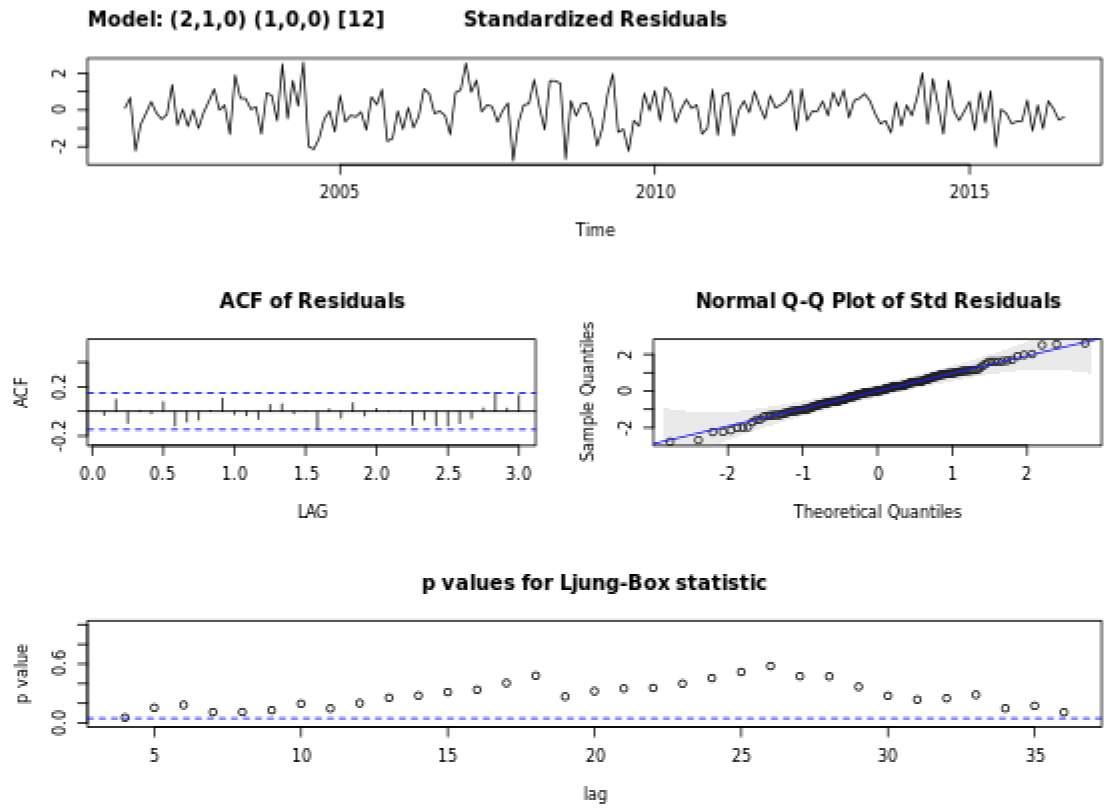
Fit ARIMA(2,1,0) to chicken - not so good

sarima(chicken, p = 2, d = 1, q = 0)



- Fit an SARIMA(2,1,0)x(1,0,0)12 and notice the model fits well.

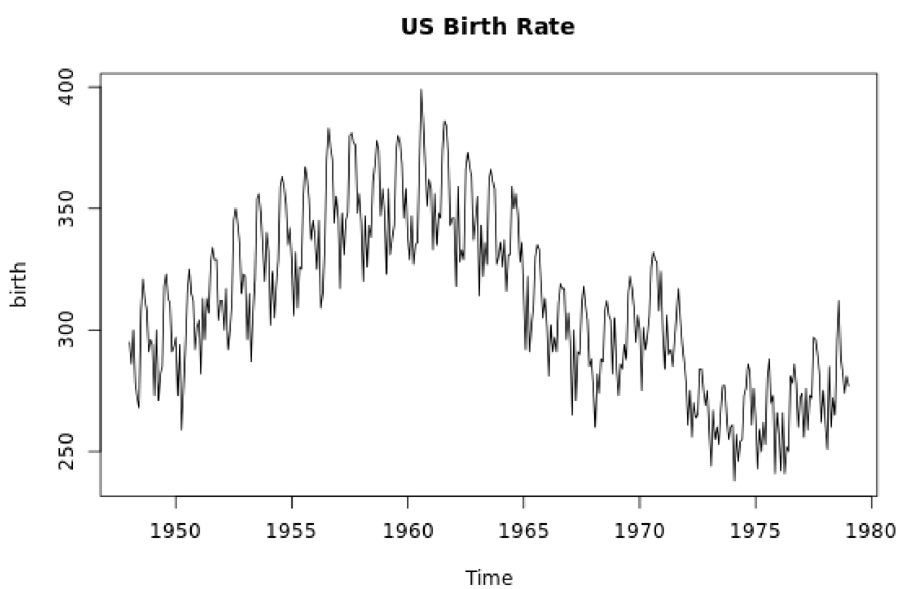
sarima(chicken, p = 2, d = 1, q = 0, P = 1, D = 0, Q = 0, S = 12)



Practice 3 Data Analysis - Birth Rate

```
> birth
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1948 295 286 300 278 272 268 308 321 313 308 291 296
1949 294 273 300 271 282 285 318 323 313 311 291 293
1950 297 273 294 259 276 294 316 325 315 312 292 301
1951 304 282 313 296 313 307 328 334 329 329 304 312
1952 312 300 317 292 300 311 345 350 344 336 315 323
1953 322 296 315 287 307 321 354 356 348 334 320 340
1954 332 302 324 305 318 329 359 363 359 352 335 342
1955 329 306 332 309 326 325 354 367 362 354 337 345
1956 339 325 345 309 315 334 370 383 375 370 344 355
1957 346 317 348 331 345 348 380 381 377 376 348 356
1958 344 320 347 326 343 338 361 368 378 374 347 358
1959 349 323 358 331 338 343 374 380 377 368 346 358
1960 338 329 347 327 335 336 370 399 385 368 351 362
```

plot(birth)



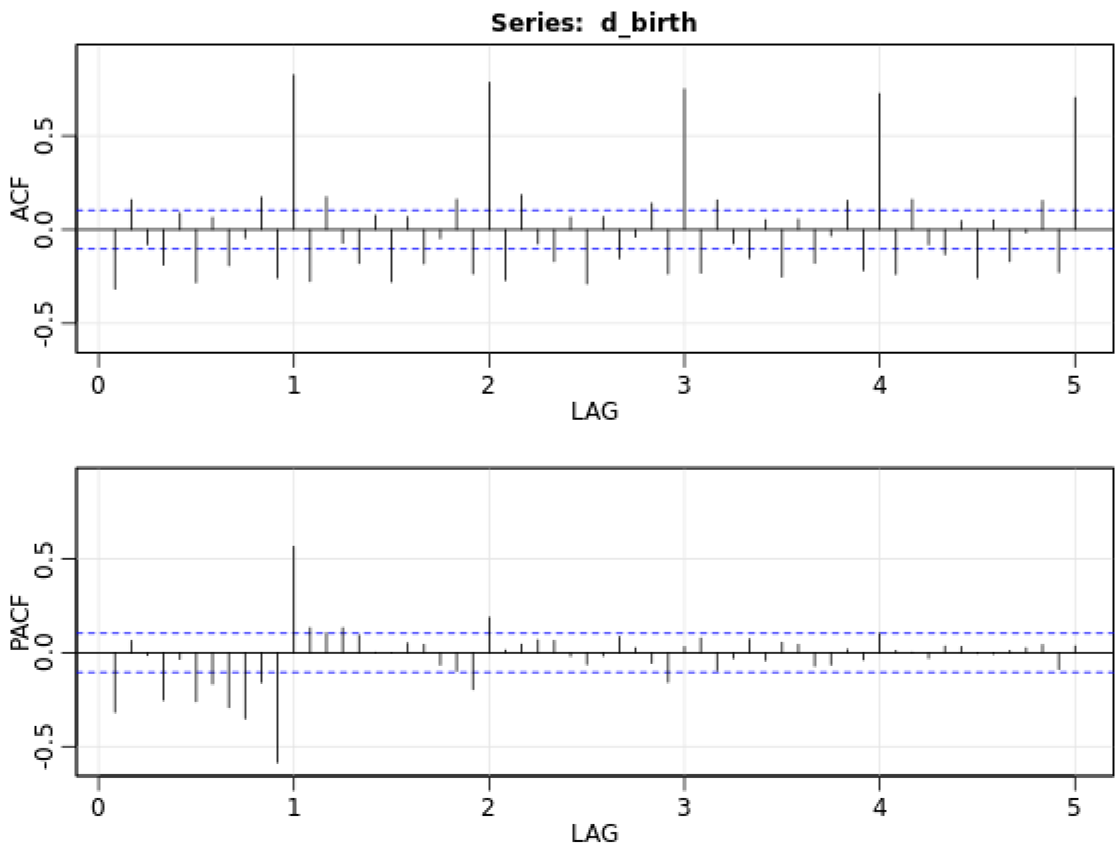
Now you will use your new skills to carefully fit an SARIMA model to the `birth` time series from `astsa`. The data are monthly live births (adjusted) in thousands for the United States, 1948-1979, and includes the baby boom after WWII.

The `birth` data are plotted in your R console. Note the long-term trend (random walk) and the seasonal component of the data.

Now you will use your new skills to carefully fit an SARIMA model to the `birth` time series from `astsa`. The data are monthly live births (adjusted) in thousands for the United States, 1948-1979, and includes the baby boom after WWII.
The `birth` data are plotted in your R console. Note the long-term trend (random walk) and the seasonal component of the data.

- Use `diff()` to difference the data (`d_birth`). Use `acf2()` to view the sample ACF and PACF of this data to lag 60. Notice the seasonal persistence.

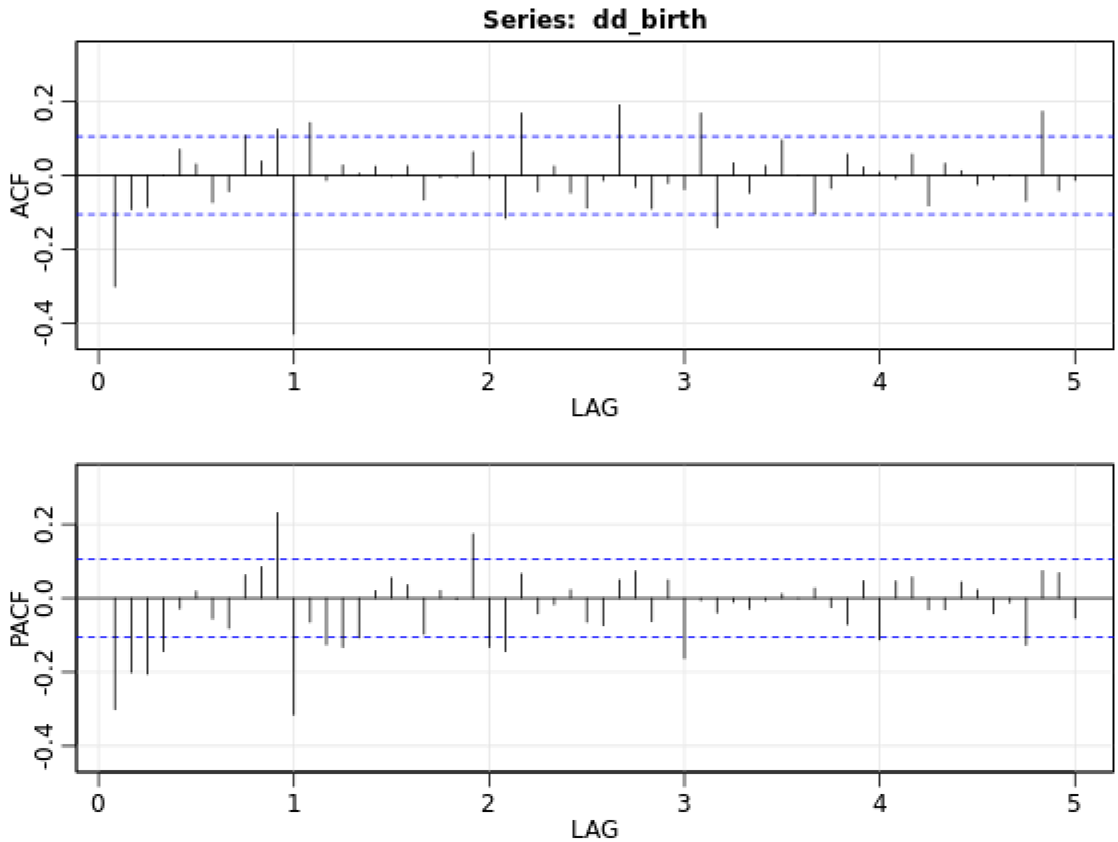
# Plot P/ACF to lag 60 of differenced data
d_birth <- diff(birth)
acf2(d_birth, max.lag = 60)



- Use another call to `diff()` to take the *seasonal* difference of the data. Save this to `dd_birth`. Use another call to `acf2()` to view the ACF and PACF of this data, again to lag 60. Conclude that an SARIMA(0,1,1)x$(0,1,1)_{12}$ model seems reasonable.

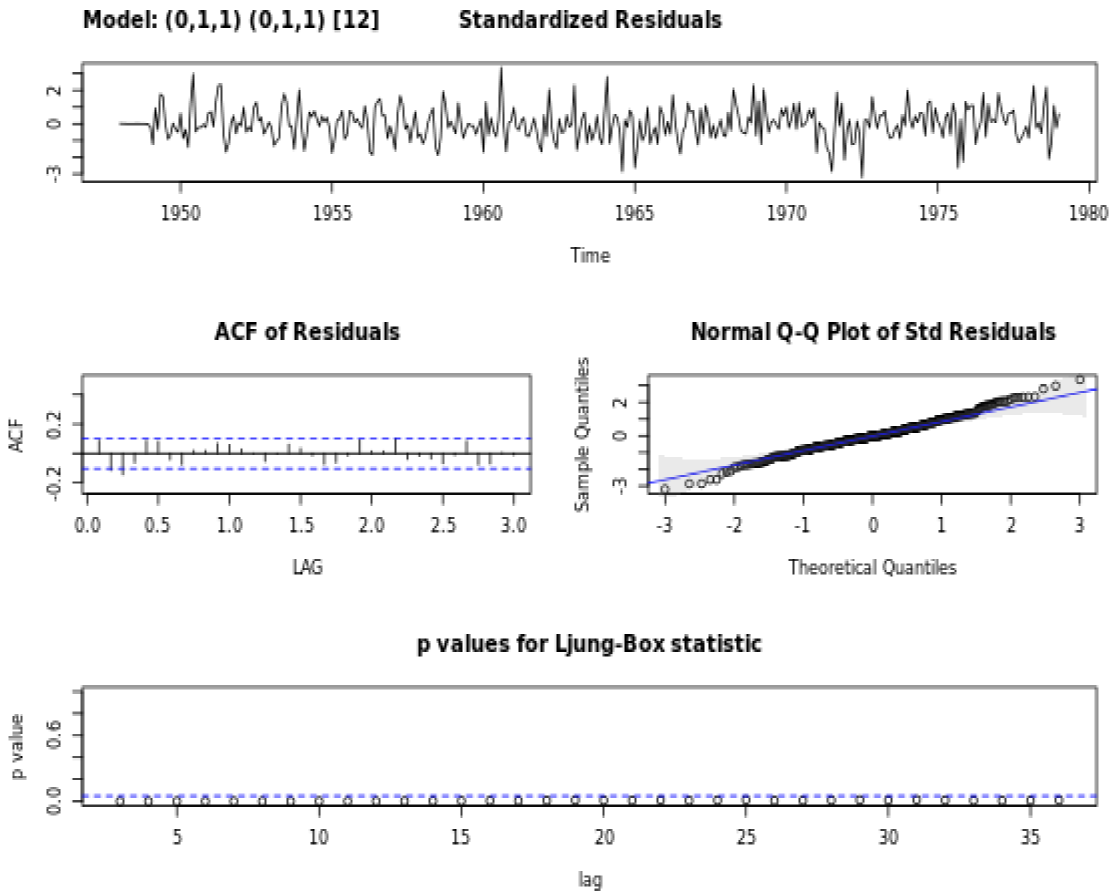
dd_birth <- diff(d_birth, lag = 12)

acf2(dd_birth, max.lag = 60)



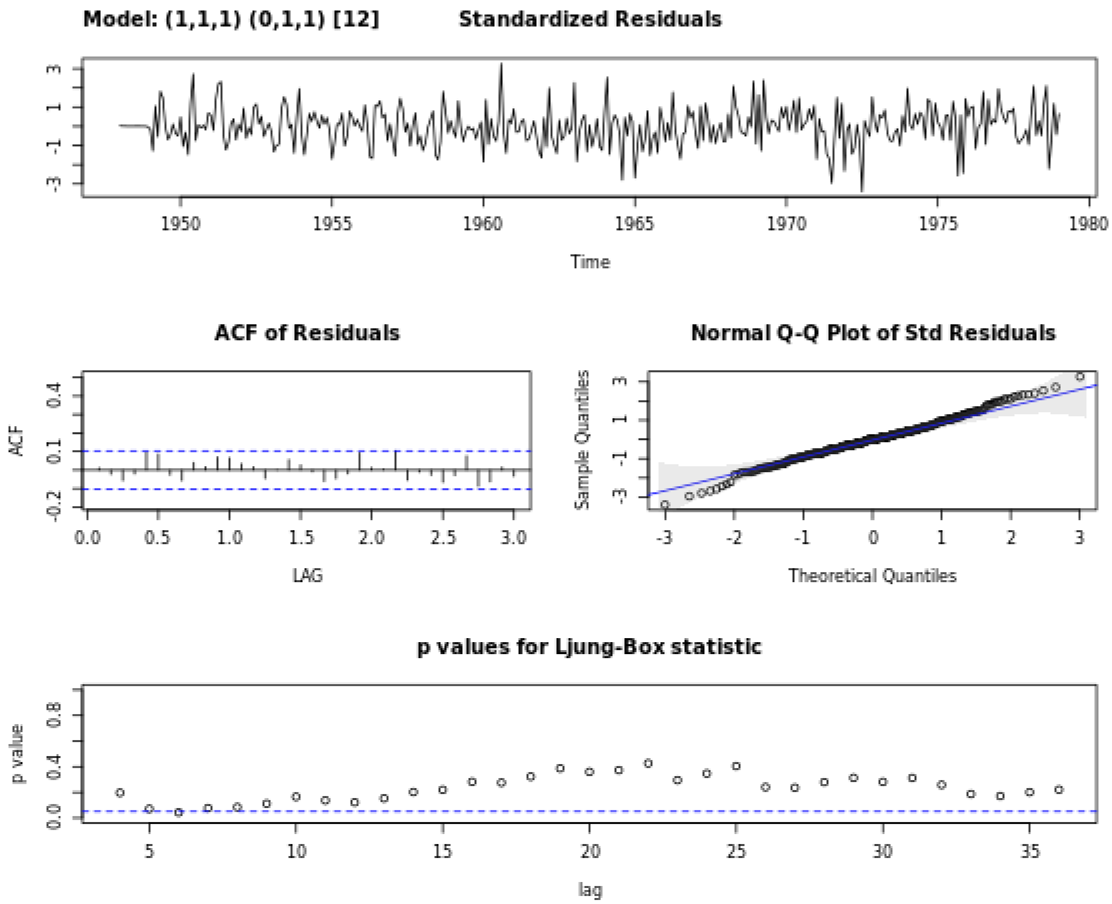
- Fit the SARIMA(0,1,1)x(0,1,1)12 model. What happens?

sarima(birth, p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12)



- Add an additional AR (nonseasonal, `p = 1`) parameter to account for additional correlation. Does the model fit well?

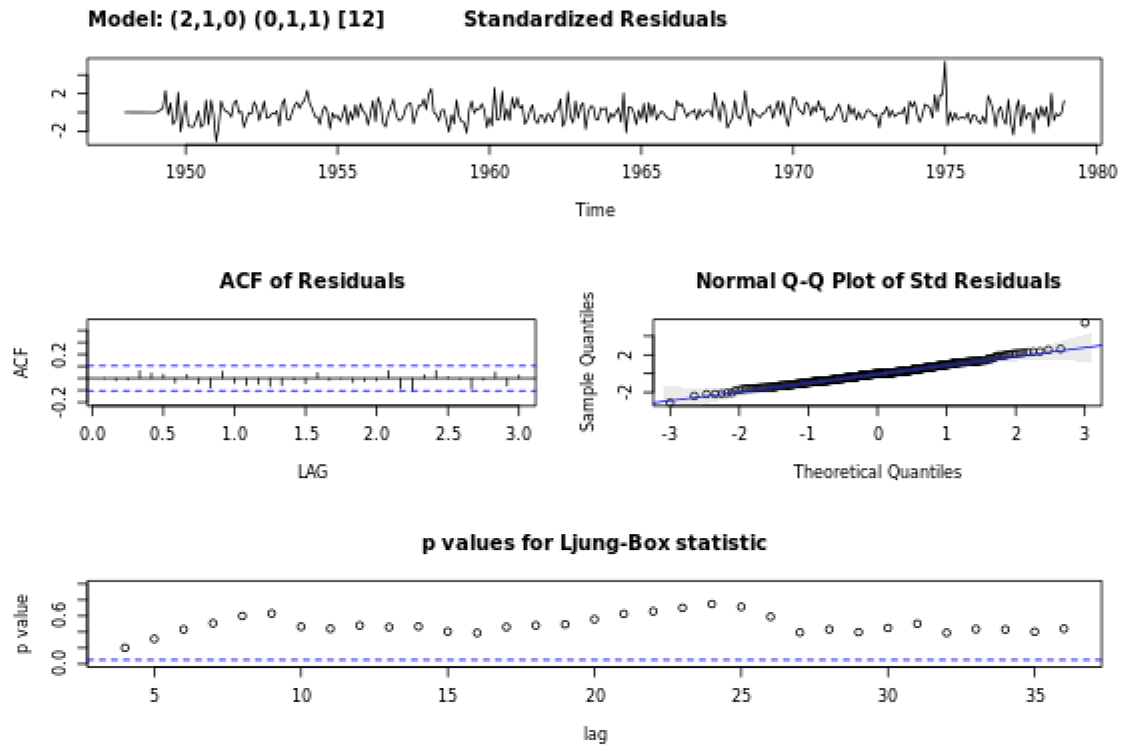
sarima(birth, p = 1, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12)



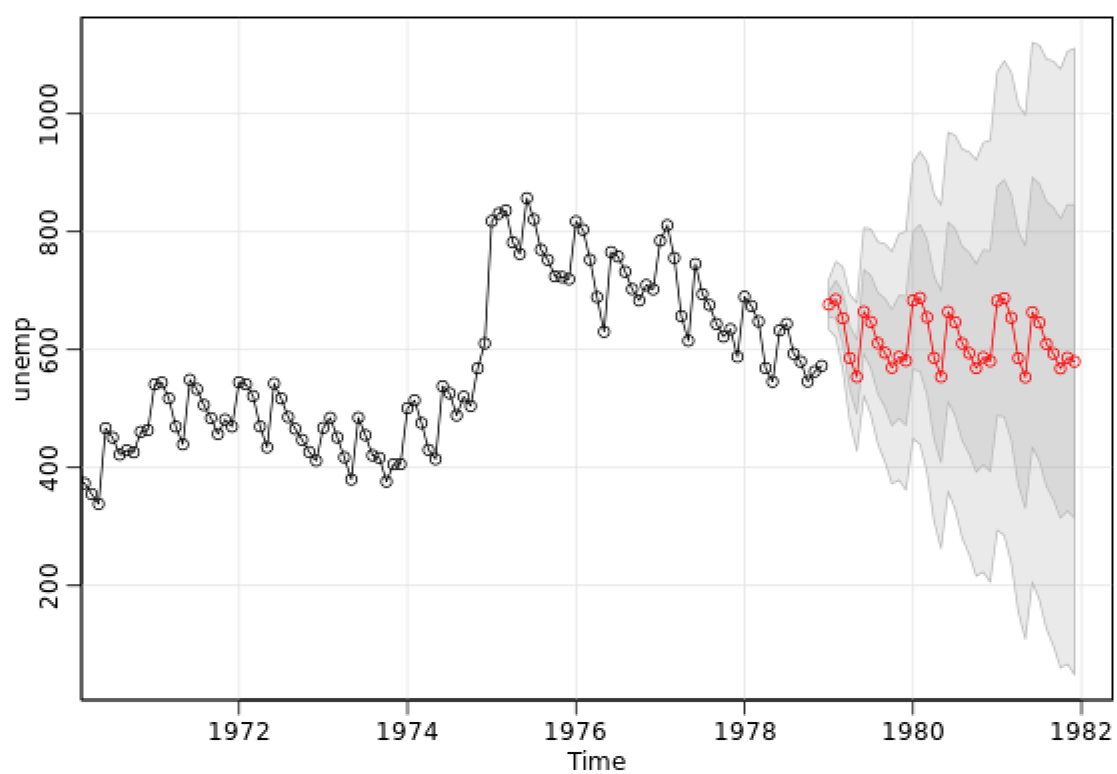
Forecasting Monthly Unemployment

Previously, you fit an SARIMA(2,1,0, 0,1,1)12 model to the monthly US unemployment time series `unemp`. You will now use that model to forecast the data 3 years. The `unemp` data are preloaded into your R workspace and plotted on the right.

```
# Fit your previous model to unemp and check the diagnostics
sarima(unemp, p = 2, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12)
```




```
# Forecast the data 3 years into the future
sarima.for(unemp, n.ahead = 36, p = 2, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12)
```



How Hard is it to Forecast Commodity Prices?

As previously mentioned, making money in commodities is not easy. To see a difficulty in predicting a commodity, you will forecast the price of chicken to five years in the future. When you complete your forecasts, you will note that even just a few years out, the acceptable range of prices is very large. This is because commodities are subject to many sources of variation.
Recall that you previously fit an SARIMA(2,1,0, 1,0,0)$_{12}$ model to the monthly US chicken price series `chicken`. You will use this model to calculate your forecasts.
The **astsa** package is preloaded for you and the monthly price of chicken data (`chicken`) are plotted on the right.

```
# Forecast the chicken data 5 years into the future
sarima.for(chicken, n.ahead = 60, p = 2, d = 1, q = 0, P = 1, D = 0, Q = 0, S = 12)
```