HW 10

Name: Paul Holaway, NetID: paulch2

Due: 11/3/2022 11:59pm

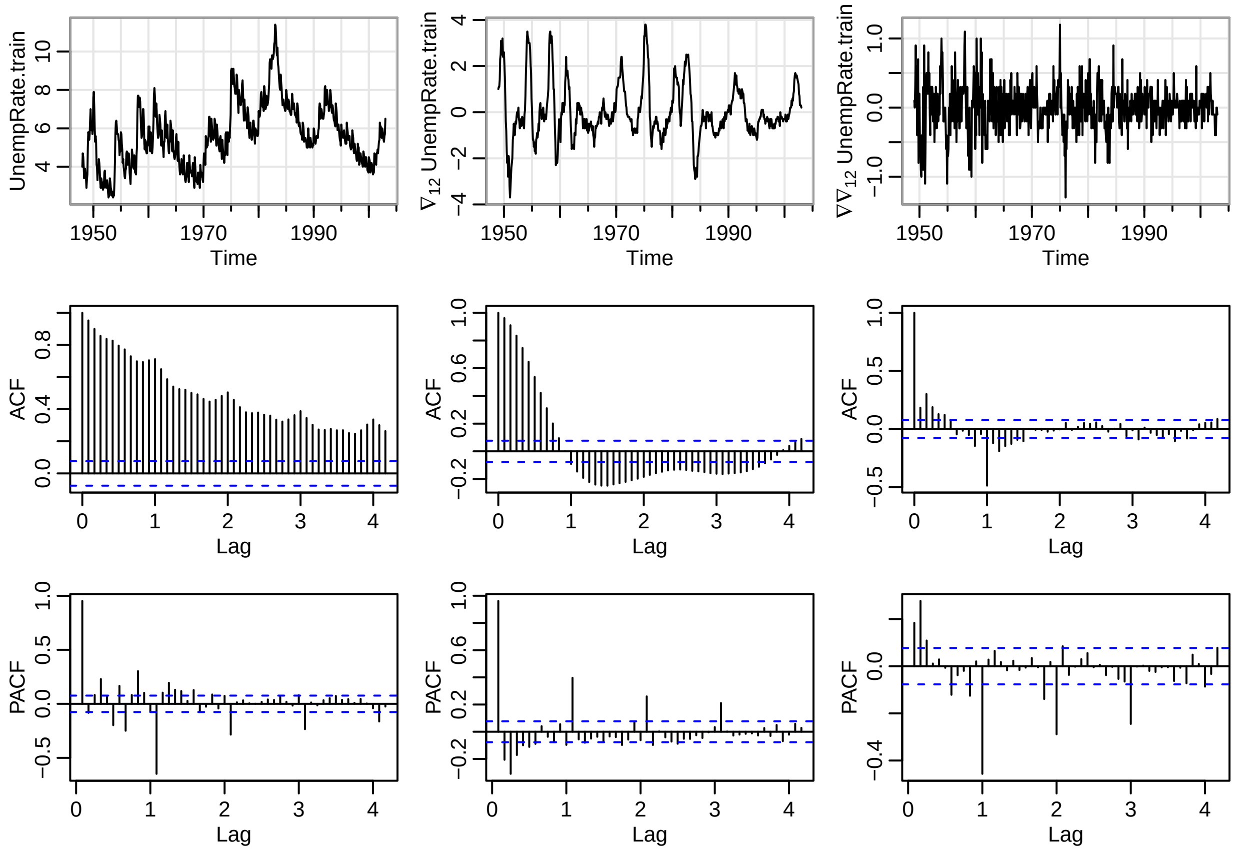
```
#Custom Function for Creating Training and Testing Data
ts.train.test = function(data, freq){
  total.length = length(data)
  test.length = round(total.length * 0.2, 0)
  train.length = total.length - test.length
  data.test = data[train.length:total.length]
  data.train = data[1:(train.length - 1)]
  data.test = ts(data.test, start = time(data)[train.length], frequency = freq)
  data.train = ts(data.train, start = time(data)[1], frequency = freq)
  x = list(data.train, data.test)
  names(x) <- c("train","test")
  return(x)
}
```

Question 1

```
###Creating Training and Testing Data for UnempRate###
UnempRate.train = ts.train.test(UnempRate, 12)$train
UnempRate.test = ts.train.test(UnempRate, 12)$test
```

Plots

```
par(mfrow = c(3,3))
tsplot(UnempRate.train)
tsplot(diff(UnempRate.train, 12), ylab = TeX(r"($\nabla_{12}$ UnempRate.train)"))
tsplot(diff(diff(UnempRate.train, 12)), ylab = TeX(r"($\nabla\nabla_{12}$ UnempRate.train)"))
acf(UnempRate.train, lag.max = 50)
acf(diff(UnempRate.train, 12), lag.max = 50)
acf(diff(diff(UnempRate.train, 12)), lag.max = 50)
pacf(UnempRate.train, ylab = "PACF", lag.max = 50)
pacf(diff(UnempRate.train, 12), ylab = "PACF", lag.max = 50)
pacf(diff(diff(UnempRate.train, 12)), ylab = "PACF", lag.max = 50)
```



Tests

```
#1
adf.test(UnempRate.train)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  UnempRate.train
## Dickey-Fuller = -3.5126, Lag order = 8, p-value = 0.04109
## alternative hypothesis: stationary
```

```
kpss.test(UnempRate.train)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  UnempRate.train
## KPSS Level = 1.8456, Truncation lag parameter = 6, p-value = 0.01
```

```
#Power Transformation?
BoxCox.lambda(UnempRate.train)
```

```
## [1] 0.8911713
```

```r
#2; D = 1
adf.test(diff(UnempRate.train, 12))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(UnempRate.train, 12)
## Dickey-Fuller = -7.5384, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(diff(UnempRate.train, 12))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(UnempRate.train, 12)
## KPSS Level = 0.069902, Truncation lag parameter = 6, p-value = 0.1
```

```r
#3; d = 1, D = 1
adf.test(diff(diff(UnempRate.train, 12)))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(UnempRate.train, 12))
## Dickey-Fuller = -8.1476, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

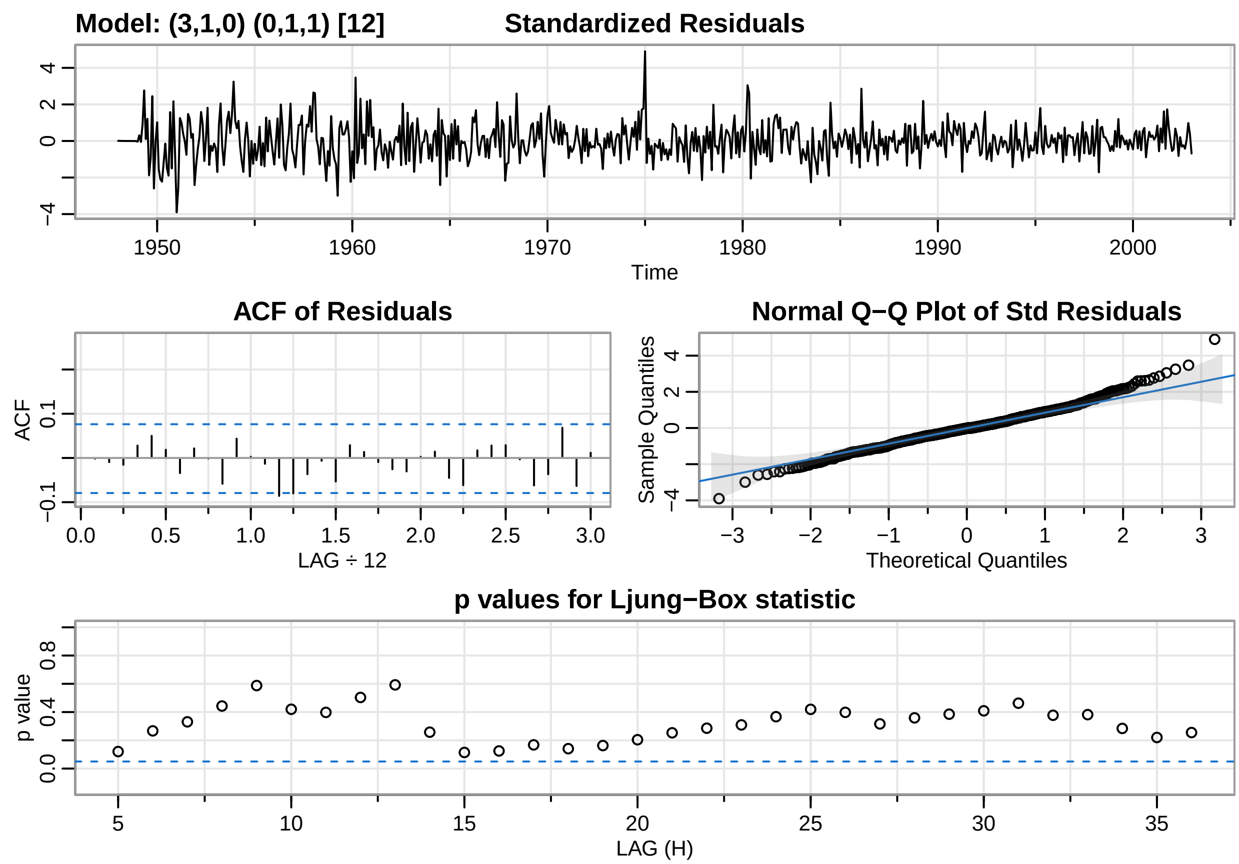
```r
kpss.test(diff(diff(UnempRate.train, 12)))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(diff(UnempRate.train, 12))
## KPSS Level = 0.011765, Truncation lag parameter = 6, p-value = 0.1
```

Modeling

```
model1.1 = sarima(UnempRate.train, p = 3, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE)
```

```
## initial  value -1.106252
## iter   2 value -1.329128
## iter   3 value -1.333262
## iter   4 value -1.363598
## iter   5 value -1.373498
## iter   6 value -1.377396
## iter   7 value -1.379961
## iter   8 value -1.380265
## iter   9 value -1.380278
## iter  10 value -1.380278
## iter  10 value -1.380278
## final  value -1.380278
## converged
## initial  value -1.387957
## iter   2 value -1.388141
## iter   3 value -1.388330
## iter   4 value -1.388335
## iter   5 value -1.388335
## iter   5 value -1.388335
## iter   5 value -1.388335
## final  value -1.388335
## converged
```



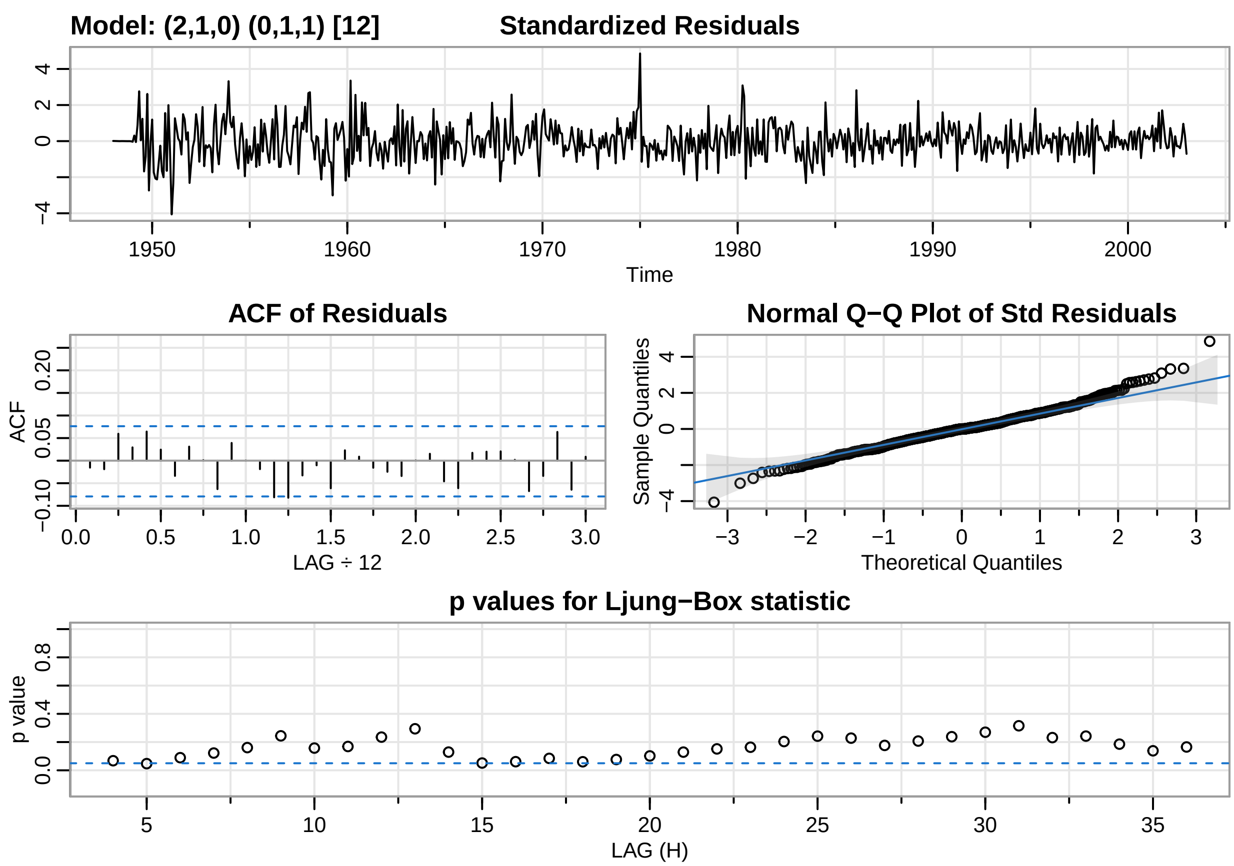

```r
model1.1$ttable
```

```
##      Estimate     SE  t.value p.value
## ar1    0.1201 0.0394   3.0516  0.0024
## ar2    0.1985 0.0388   5.1209  0.0000
## ar3    0.0802 0.0393   2.0439  0.0414
## sma1  -0.7513 0.0293 -25.6429  0.0000
```

```r
model1.2 = sarima(UnempRate.train, p = 2, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE)
```

```
## initial  value -1.106389
## iter   2 value -1.336462
## iter   3 value -1.356311
## iter   4 value -1.367782
## iter   5 value -1.374036
## iter   6 value -1.377816
## iter   7 value -1.378386
## iter   8 value -1.378422
## iter   9 value -1.378422
## iter  10 value -1.378422
## iter  10 value -1.378422
## iter  10 value -1.378422
## final  value -1.378422
## converged
```

```
## initial  value -1.384826
## iter   2 value -1.384932
## iter   3 value -1.385120
## iter   4 value -1.385122
## iter   5 value -1.385123
## iter   5 value -1.385123
## iter   5 value -1.385123
## final  value -1.385123
## converged
```




```
model1.2$ttable
```

```
##       Estimate     SE  t.value p.value
## ar1     0.1367 0.0387   3.5342   4e-04
## ar2     0.2091 0.0386   5.4224   0e+00
## sma1   -0.7540 0.0289 -26.1183   0e+00
```

```
###Testing for Autocorrelations###
#p = 3
Box.test(model1.1$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
##
```

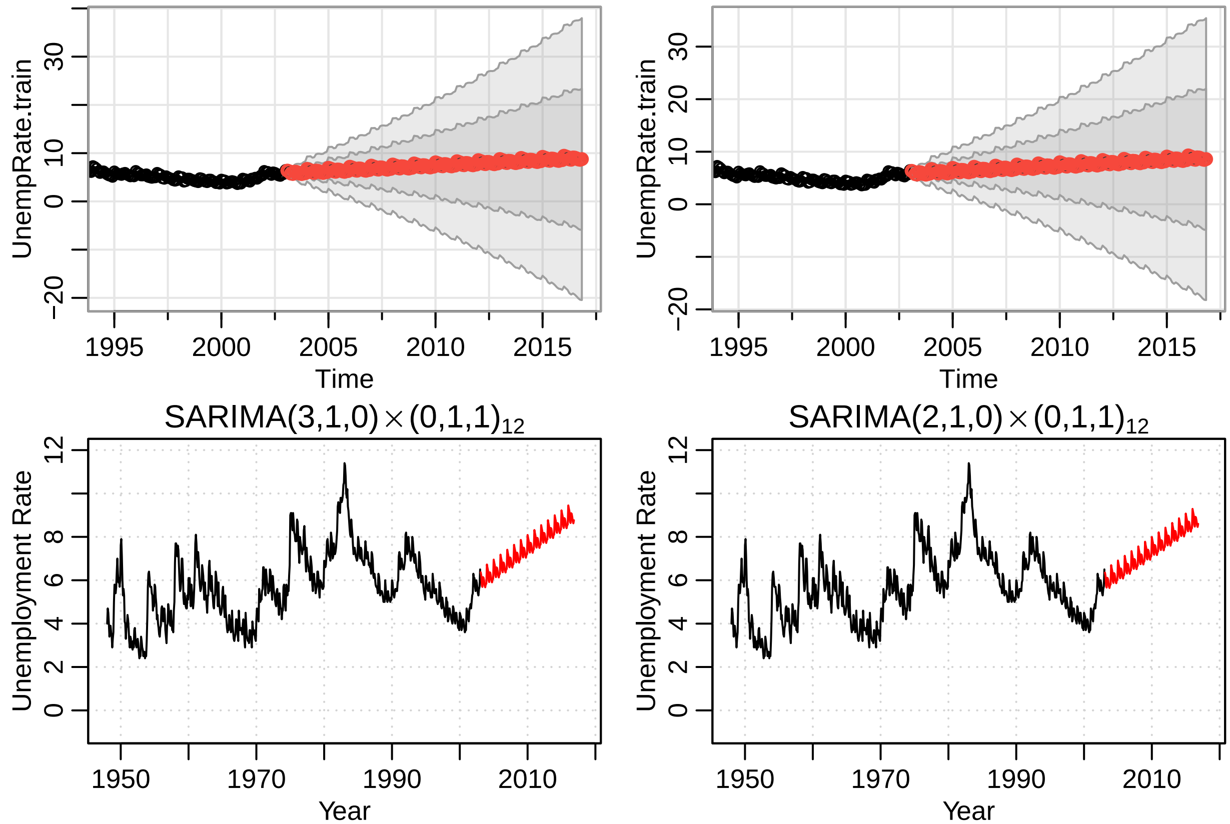
```
## data:  model1.1$fit$residuals
## X-squared = 21.163, df = 25, p-value = 0.6835
```

```
#p = 2
Box.test(model1.2$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  model1.2$fit$residuals
## X-squared = 25.683, df = 25, p-value = 0.4247
```

Predictions

```
par(mfrow = c(2,2))
model1.1F = sarima.for(UnempRate.train, p = 3, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE, n.ahead = length(UnempRate.test))
model1.2F = sarima.for(UnempRate.train, p = 2, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE, n.ahead = length(UnempRate.test))
################################################################################
plot(1948:2018,rep(0,71),col = "white", ylim = c(-1,12), xlab = "Year",
     ylab = "Unemployment Rate", main = TeX(r"(SARIMA(3,1,0)$\times$(0,1,1)$_{12}$)"))
grid()
box()
lines(UnempRate.train)
lines(model1.1F$pred, col = "red")
plot(1948:2018,rep(0,71),col = "white", ylim = c(-1,12), xlab = "Year",
     ylab = "Unemployment Rate", main = TeX(r"(SARIMA(2,1,0)$\times$(0,1,1)$_{12}$)"))
grid()
box()
lines(UnempRate.train)
lines(model1.2F$pred, col = "red")
```




```r
#Accuracy
#p = 3
accuracy(object = model1.1F$pred, x = UnempRate.test)
```

```
##                   ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.9338216 2.032468 1.703602 -21.45311 29.72381 0.9818276  6.486735
```

```r
#p = 2
accuracy(object = model1.2F$pred, x = UnempRate.test)
```

```
##                   ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.8363455 1.978557 1.664636 -19.88994 28.82251 0.9821786  6.253824
```

Final Model

$$\boxed{\text{SARIMA}(2,1,0)\times(0,1,1)_{12}}$$

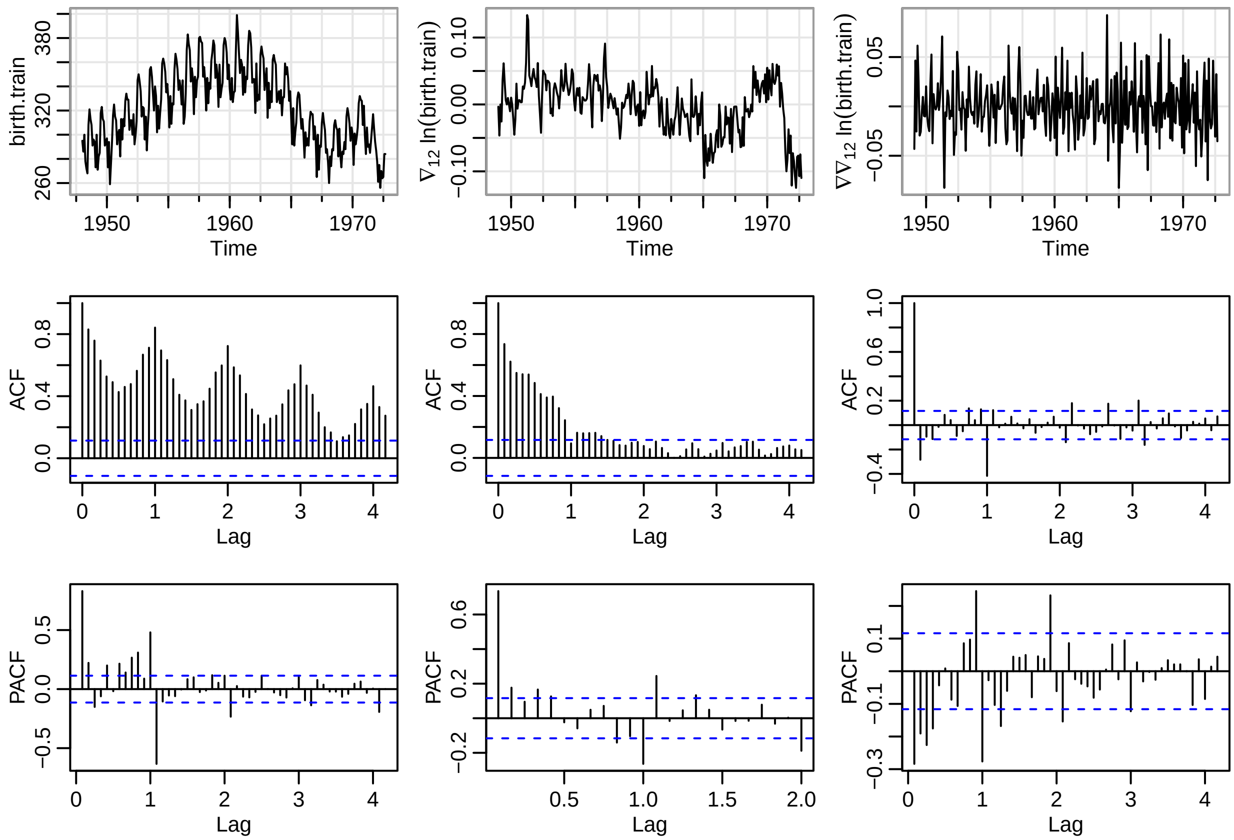
When first looking at the data, there appears to be a seasonal trend present. After running both `adf.test()` and `kpss.test()` I decided that there was indeed some kind of trend. While the p-value for the ADF-Test was below 0.05, it was 0.041 which was close and the KPSS-Test p-value was 0.01. Therefore I did a seasonal difference of the data and noticed that the data still looked like there was some sort of trend, however, both the ADF-Test and KPSS-Test did not show evidence of the data not being stationary. The Box-Cox test also output an optimal $\lambda \approx 0.8912$, which is close to 1, so I did not see any reason to perform a transformation. After doing non-seasonal difference, the data looked much closer to white noise and the ACF and PACF were both easier to distinguish orders for the parameters. We can see that the ACF cuts of after seasonal lag 1 and the PACF tails off for seasonal lags, suggesting $P = 0$ and $Q = 1$. For the non-seasonal part, we can see that the ACF tails off and the PACF cuts off at lag 3, but lag 3 is barely significant. Therefore I ran two different models. One with $p = 3$ and $q = 0$ and the other with $p = 2$ and $q = 0$. In either case, all of the parameters in the models were statistically significant, the time series plots of the residuals are white noise like, the ACF of the residuals shows that the vast majority are statistically zero, the Normal QQ-Plot looks good (except for a few outliers at the tail ends), and for the most part there does not appear to be autocorrelations between observations. We can confirm this using the Box-Pierce test as we fail to reject the null hypotheses for both that there is no autocorrelation. As both models were similar in all of the diagnostics, AIC, and BIC, I ended up choosing the one with the lower RMSE value as the best model.

Question 2.

```
#Creating Training and Testing Data for birth
birth.train = ts.train.test(birth, 12)$train
birth.test = ts.train.test(birth, 12)$test
```

Plots

```
par(mfrow = c(3,3))
tsplot(birth.train)
tsplot(diff(log(birth.train), 12), ylab = TeX(r"($\nabla_{12}$ $\ln(birth.train)$)"))
tsplot(diff(diff(log(birth.train), 12)), ylab = TeX(r"($\nabla\nabla_{12}$ $\ln(birth.train)$)"))
acf(birth.train, lag.max = 50)
acf(diff(log(birth.train), 12), lag.max = 50)
acf(diff(diff(log(birth.train), 12)), lag.max = 50)
pacf(birth.train, ylab = "PACF", lag.max = 50)
pacf(diff(log(birth.train), 12), ylab = "PACF")
pacf(diff(diff(log(birth.train), 12)), ylab = "PACF", lag.max = 50)
```



Tests

```r
#1
adf.test(birth.train)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  birth.train
## Dickey-Fuller = -2.7256, Lag order = 6, p-value = 0.2704
## alternative hypothesis: stationary
```

```r
kpss.test(birth.train)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  birth.train
## KPSS Level = 1.196, Truncation lag parameter = 5, p-value = 0.01
```

```r
#Power Transformation?
BoxCox.lambda(birth.train)
```

```
## [1] -0.1413726
```

```r
#Retesting with log-transformation
adf.test(log(birth.train))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  log(birth.train)
## Dickey-Fuller = -2.6963, Lag order = 6, p-value = 0.2828
## alternative hypothesis: stationary
```

```r
kpss.test(log(birth.train))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  log(birth.train)
## KPSS Level = 1.201, Truncation lag parameter = 5, p-value = 0.01
```

```r
#2; D = 1
adf.test(diff(log(birth.train), 12))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(log(birth.train), 12)
## Dickey-Fuller = -3.0069, Lag order = 6, p-value = 0.1519
## alternative hypothesis: stationary
```

```r
kpss.test(diff(log(birth.train), 12))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(log(birth.train), 12)
## KPSS Level = 1.4726, Truncation lag parameter = 5, p-value = 0.01
```

```r
#3; d = 1, D = 1
adf.test(diff(diff(log(birth.train), 12)))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(log(birth.train), 12))
## Dickey-Fuller = -8.5088, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

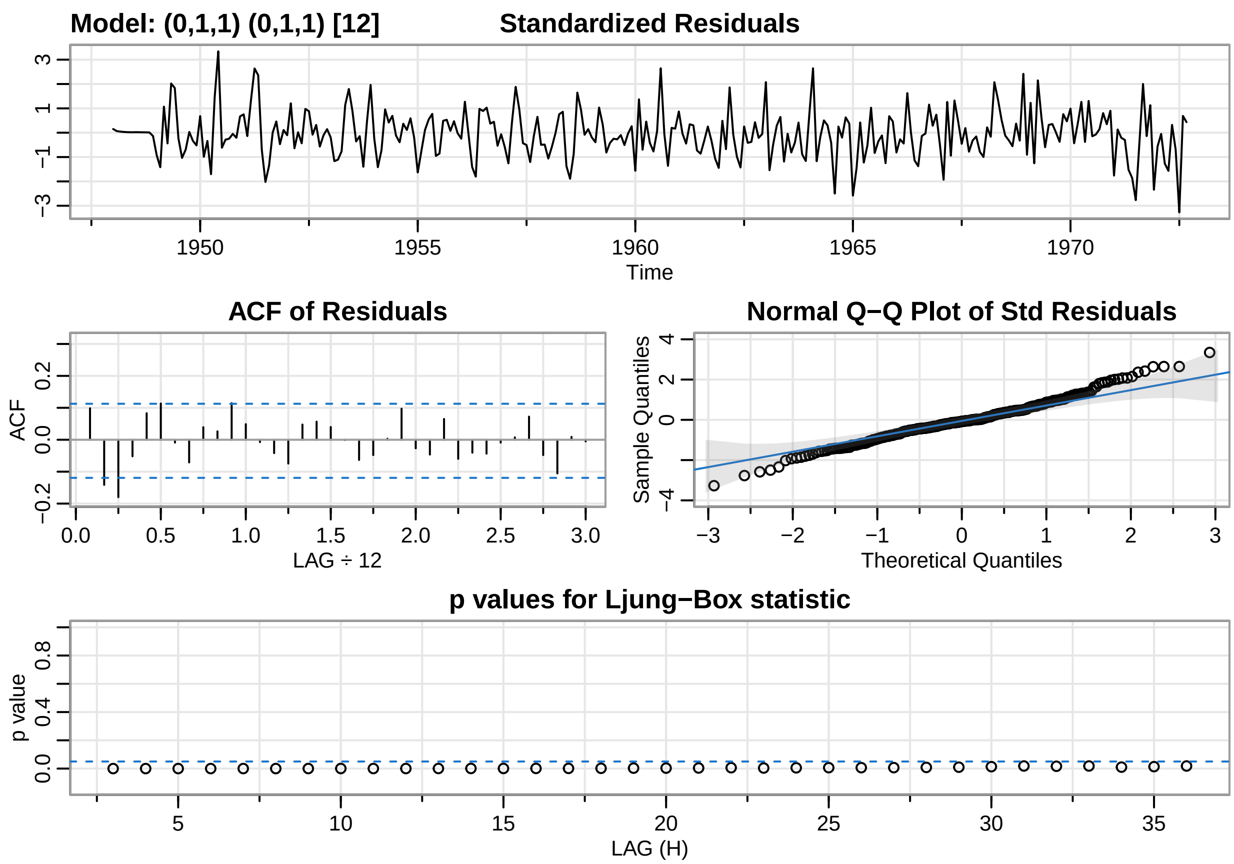
```r
kpss.test(diff(diff(log(birth.train), 12)))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(diff(log(birth.train), 12))
## KPSS Level = 0.06118, Truncation lag parameter = 5, p-value = 0.1
```

Modeling

```r
model2.1 = sarima(log(birth.train), p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE)
```

```
## initial  value -3.541003
## iter   2 value -3.731385
## iter   3 value -3.755380
## iter   4 value -3.761681
## iter   5 value -3.774530
## iter   6 value -3.776707
## iter   7 value -3.777065
## iter   8 value -3.777139
## iter   9 value -3.777140
## iter   9 value -3.777140
## iter   9 value -3.777140
## final  value -3.777140
## converged
## initial  value -3.798802
## iter   2 value -3.809963
## iter   3 value -3.810099
## iter   4 value -3.810143
## iter   5 value -3.810143
## iter   5 value -3.810143
## iter   5 value -3.810143
## final  value -3.810143
## converged
```



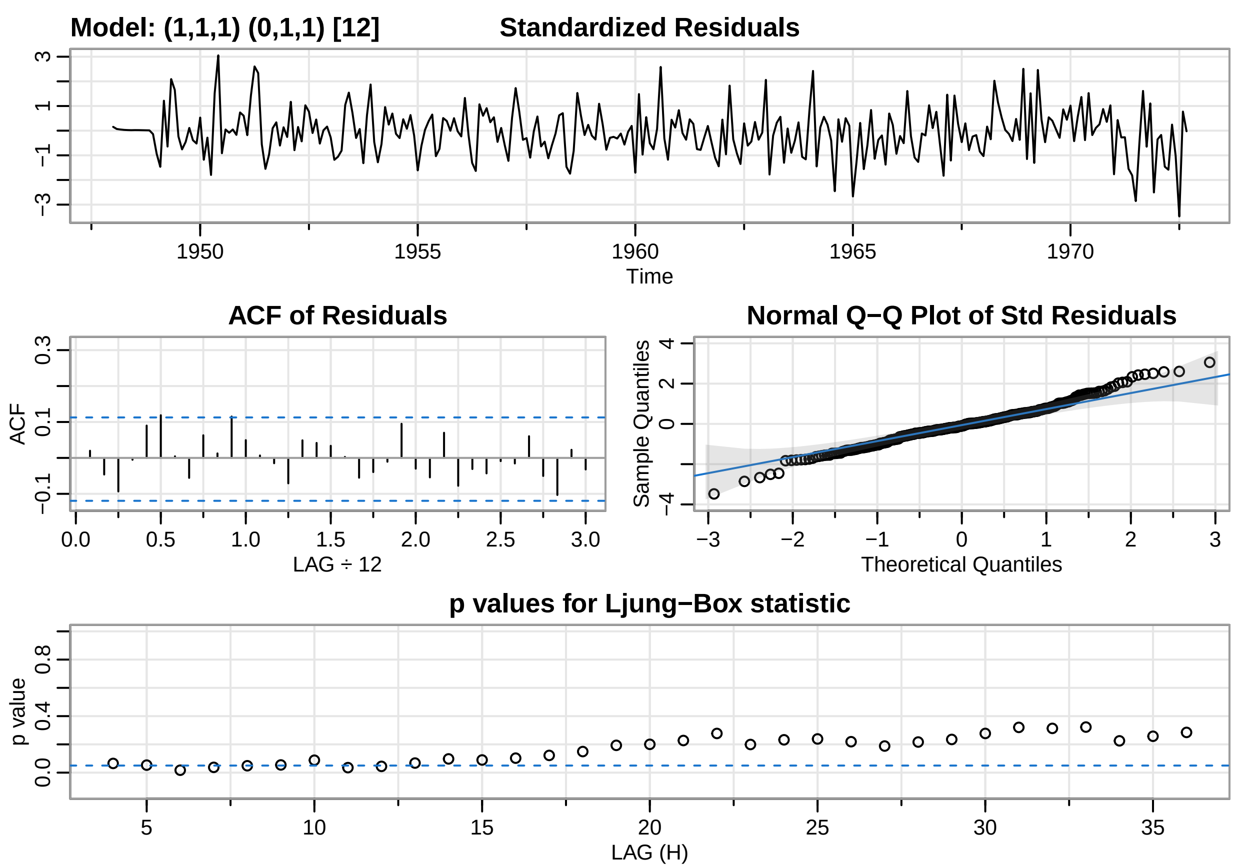

```r
model2.1$ttable
```

```
##      Estimate     SE  t.value p.value
## ma1   -0.4559 0.0695  -6.5602       0
## sma1  -0.7985 0.0432 -18.4626       0
```

```r
model2.2 = sarima(log(birth.train), p = 1, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE)
```

```
## initial  value -3.543152
## iter   2 value -3.714077
## iter   3 value -3.759085
## iter   4 value -3.766194
## iter   5 value -3.770656
## iter   6 value -3.773380
## iter   7 value -3.776197
## iter   8 value -3.777713
## iter   9 value -3.778330
## iter  10 value -3.779441
## iter  11 value -3.779830
## iter  12 value -3.780128
## iter  13 value -3.780211
## iter  14 value -3.780215
## iter  14 value -3.780215
## iter  14 value -3.780215
```

```
## final  value -3.780215
## converged
## initial  value -3.809137
## iter   2 value -3.822321
## iter   3 value -3.824671
## iter   4 value -3.826166
## iter   5 value -3.827028
## iter   6 value -3.827441
## iter   7 value -3.827472
## iter   8 value -3.827473
## iter   8 value -3.827473
## final  value -3.827473
## converged
```



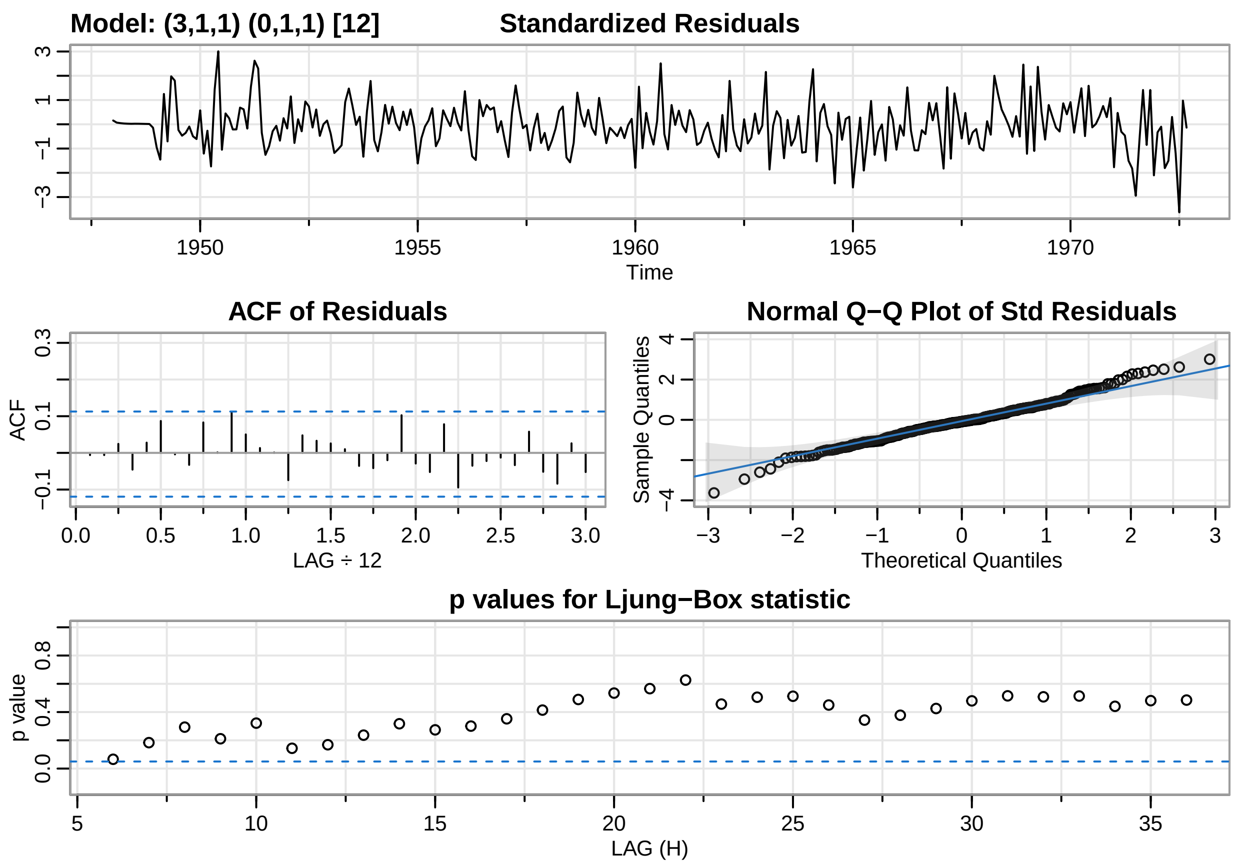

```
model2.2$ttable
```

```
##      Estimate     SE  t.value p.value
## ar1    0.3167 0.0945   3.3498   9e-04
## ma1   -0.6942 0.0659 -10.5342   0e+00
## sma1  -0.8080 0.0428 -18.8951   0e+00
```

```
model2.3 = sarima(log(birth.train), p = 3, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE)
```

```
## initial  value -3.545586
```

```
## iter   2 value -3.747946
## iter   3 value -3.787049
## iter   4 value -3.795858
## iter   5 value -3.802989
## iter   6 value -3.804141
## iter   7 value -3.804375
## iter   8 value -3.804388
## iter   9 value -3.804400
## iter  10 value -3.804423
## iter  11 value -3.804444
## iter  12 value -3.804453
## iter  13 value -3.804454
## iter  14 value -3.804455
## iter  14 value -3.804455
## iter  14 value -3.804455
## final  value -3.804455
## converged
## initial  value -3.824354
## iter   2 value -3.834083
## iter   3 value -3.836182
## iter   4 value -3.837237
## iter   5 value -3.837747
## iter   6 value -3.838087
## iter   7 value -3.839045
## iter   8 value -3.839120
## iter   9 value -3.839129
## iter  10 value -3.839129
## iter  10 value -3.839129
## iter  10 value -3.839129
## final  value -3.839129
## converged
```




```
model2.3$ttable
```

```
##       Estimate     SE  t.value p.value
## ar1     0.0712 0.1471   0.4837  0.6290
## ar2    -0.1348 0.0723  -1.8645  0.0633
## ar3    -0.1757 0.0730  -2.4076  0.0167
## ma1    -0.4335 0.1421  -3.0499  0.0025
## sma1   -0.8270 0.0426 -19.4014  0.0000
```

```
###Testing for Autocorrelations###
#p = 0
Box.test(model2.1$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  model2.1$fit$residuals
## X-squared = 41.777, df = 25, p-value = 0.01899
```

```
#p = 1
Box.test(model2.2$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
```

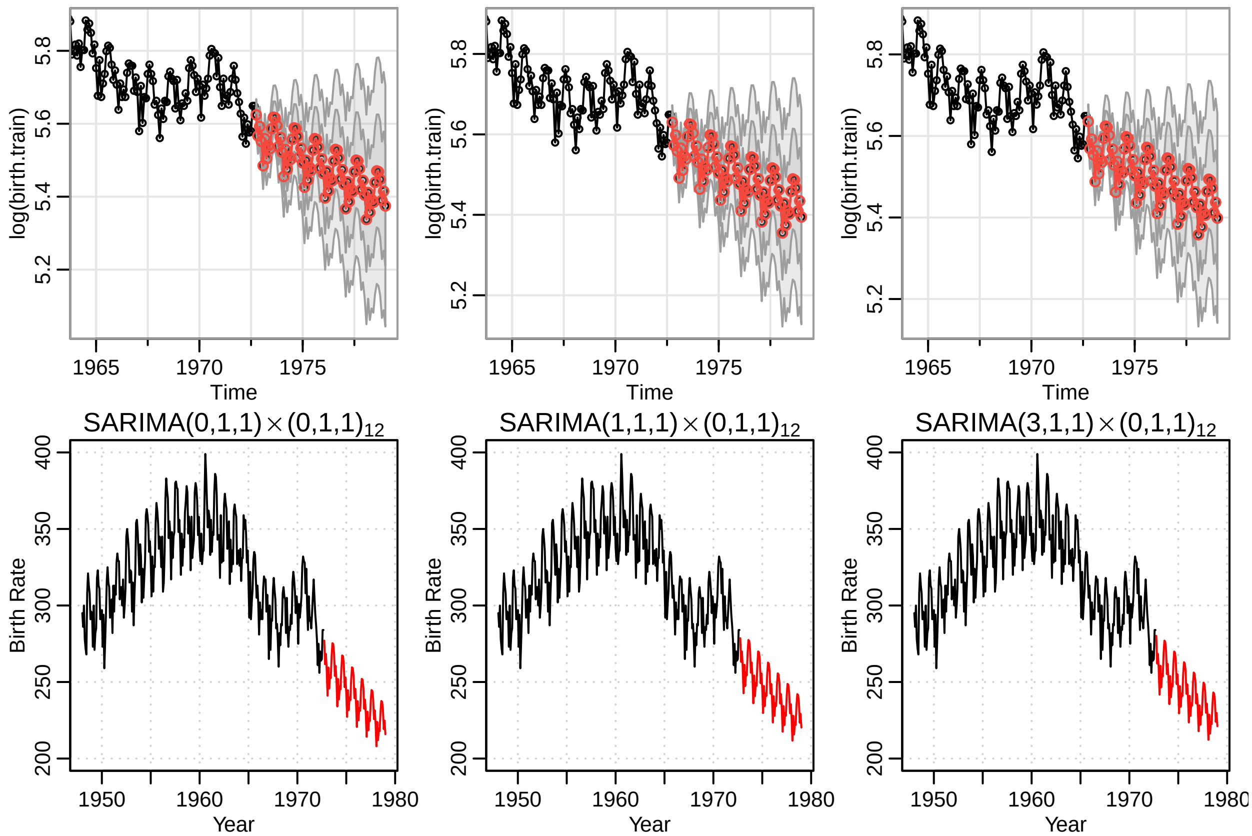
```
##
## data:  model2.2$fit$residuals
## X-squared = 25.123, df = 25, p-value = 0.4555
```

```
#p = 3
Box.test(model2.3$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  model2.3$fit$residuals
## X-squared = 18.123, df = 25, p-value = 0.8371
```

Predictions

```
par(mfrow = c(2,3))
model2.1F = sarima.for(log(birth.train), p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE, n.ahead = length(birth.test))
model2.2F = sarima.for(log(birth.train), p = 1, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE, n.ahead = length(birth.test))
model2.3F = sarima.for(log(birth.train), p = 3, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,
                  no.constant = TRUE, n.ahead = length(birth.test))
################################################################################
plot(1948:1979,rep(0,32), col = "white", ylim = c(200,400), xlab = "Year",
     ylab = "Birth Rate", main = TeX(r"(SARIMA(0,1,1)$\times$(0,1,1)$_{12}$)"))
grid()
box()
lines(birth.train)
lines(exp(model2.1F$pred), col = "red")
plot(1948:1979,rep(0,32), col = "white", ylim = c(200,400), xlab = "Year",
     ylab = "Birth Rate", main = TeX(r"(SARIMA(1,1,1)$\times$(0,1,1)$_{12}$)"))
grid()
box()
lines(birth.train)
lines(exp(model2.2F$pred), col = "red")
plot(1948:1979,rep(0,32), col = "white", ylim = c(200,400), xlab = "Year",
     ylab = "Birth Rate", main = TeX(r"(SARIMA(3,1,1)$\times$(0,1,1)$_{12}$)"))
grid()
box()
lines(birth.train)
lines(exp(model2.3F$pred), col = "red")
```




```
###Accuracy###
#p = 0
accuracy(object = model2.1F$pred, x = log(birth.test))
```

```
##                  ME      RMSE       MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.1022522 0.1284684 0.1029996 1.822503 1.835885 0.8971834  2.202145
```

```
#p = 1
accuracy(object = model2.2F$pred, x = log(birth.test))
```

```
##                   ME      RMSE        MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.08958704 0.1161866 0.09162646 1.596149 1.632762 0.8917326  1.991225
```

```
#p = 3
accuracy(object = model2.3F$pred, x = log(birth.test))
```

```
##                   ME      RMSE        MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.08928912 0.1146808 0.09106257 1.591072 1.622867 0.8867771  1.965385
```

Final Model

$$\boxed{\text{SARIMA}(3, 1, 1) \times (0, 1, 1)_{12}}$$

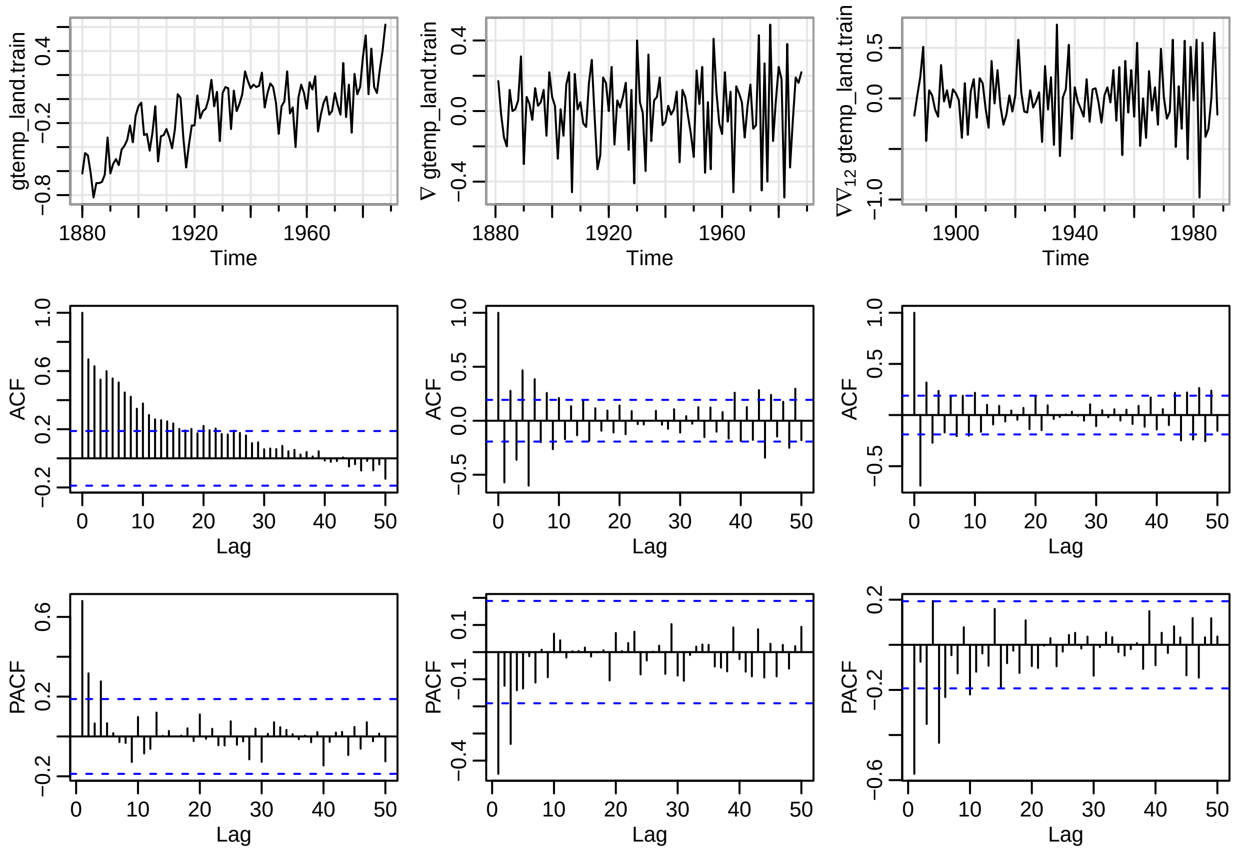
When first looking at the data, there appears to be both a seasonal and non-seasonal trend present. However, the Box-Cox test output an optimal $\lambda \approx -0.1414$, which is close to 0, so I performed a log-transformation on the data. After running both `adf.test()` and `kpss.test()` I decided that there was indeed some kind of trend. The p-value for the ADF-Test was above 0.05, but the KPSS-Test p-value was 0.01. Therefore I did a seasonal difference of the data to see what would happen. I then noticed that the data was better, but still showed some sort of trend. The ADF-Test showed signs of a trend being present while KPSS-Test did not. After doing non-seasonal difference, the data looked much closer to white noise and the ACF and PACF were both easier to distinguish orders for the parameters. We can see that the ACF cuts of after seasonal lag 1 and the PACF tails off for seasonal lags, suggesting $P = 0$ and $Q = 1$. For the non-seasonal part, we can see that the ACF cuts off at lag 1 and the PACF tails off, with lags 1-4 being significant. Therefore I ran three different models. One with $p = 0$ and $q = 1$, $p = 1$ and $q = 1$, and the other with $p = 3$ and $q = 1$. $p = 2$ resulted in almost all of the coefficients being insignificant. In all cases, the time series plot, Normal QQ-Plot, and ACF of the residuals all looked about the same. The plot of the residuals was white noise like, the ACF was almost all statistically zero, and the Normal QQ-Plots all followed the line close. The AIC and BIC were all close to each other with $p = 3$ having the lowest. The RMSE of $p = 3$ was also the smallest and it was the only one that showed no signs of autocorrelation between observations in both the Box-Pierce test and Ljung-Box test. Therefore, I will choose the model with $p = 3$ for my final model.

Question 3.

```r
#Creating Training and Testing Data for gtemp_land
gtemp_land.train = ts.train.test(gtemp_land, 1)$train
gtemp_land.test = ts.train.test(gtemp_land, 1)$test
```

Plots

```r
par(mfrow = c(3,3))
tsplot(gtemp_land.train)
tsplot(diff(gtemp_land.train), ylab = TeX(r"($\nabla$ gtemp_land.train)"))
tsplot(diff(diff(gtemp_land.train), 5), ylab = TeX(r"($\nabla\nabla_{12}$ gtemp_land.train)"))
acf(gtemp_land.train, lag.max = 50)
acf(diff(diff(gtemp_land.train), 5), lag.max = 50)
acf(diff(diff(gtemp_land.train)), lag.max = 50)
pacf(gtemp_land.train, ylab = "PACF", lag.max = 50)
pacf(diff(gtemp_land.train), ylab = "PACF", lag.max = 50)
pacf(diff(diff(gtemp_land.train), 5), ylab = "PACF", lag.max = 50)
```



Tests

```r
#1
adf.test(gtemp_land.train)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gtemp_land.train
## Dickey-Fuller = -2.7867, Lag order = 4, p-value = 0.2503
## alternative hypothesis: stationary
```

```r
kpss.test(gtemp_land.train)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  gtemp_land.train
## KPSS Level = 1.7304, Truncation lag parameter = 4, p-value = 0.01
```

```r
#2; d = 1
adf.test(diff(gtemp_land.train))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(gtemp_land.train)
## Dickey-Fuller = -7.3807, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(gtemp_land.train))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(gtemp_land.train)
## KPSS Level = 0.056877, Truncation lag parameter = 4, p-value = 0.1
```

```
#3; d = 1, D = 1
adf.test(diff(diff(gtemp_land.train), 5))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(gtemp_land.train), 5)
## Dickey-Fuller = -7.6943, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(diff(gtemp_land.train), 5))
```

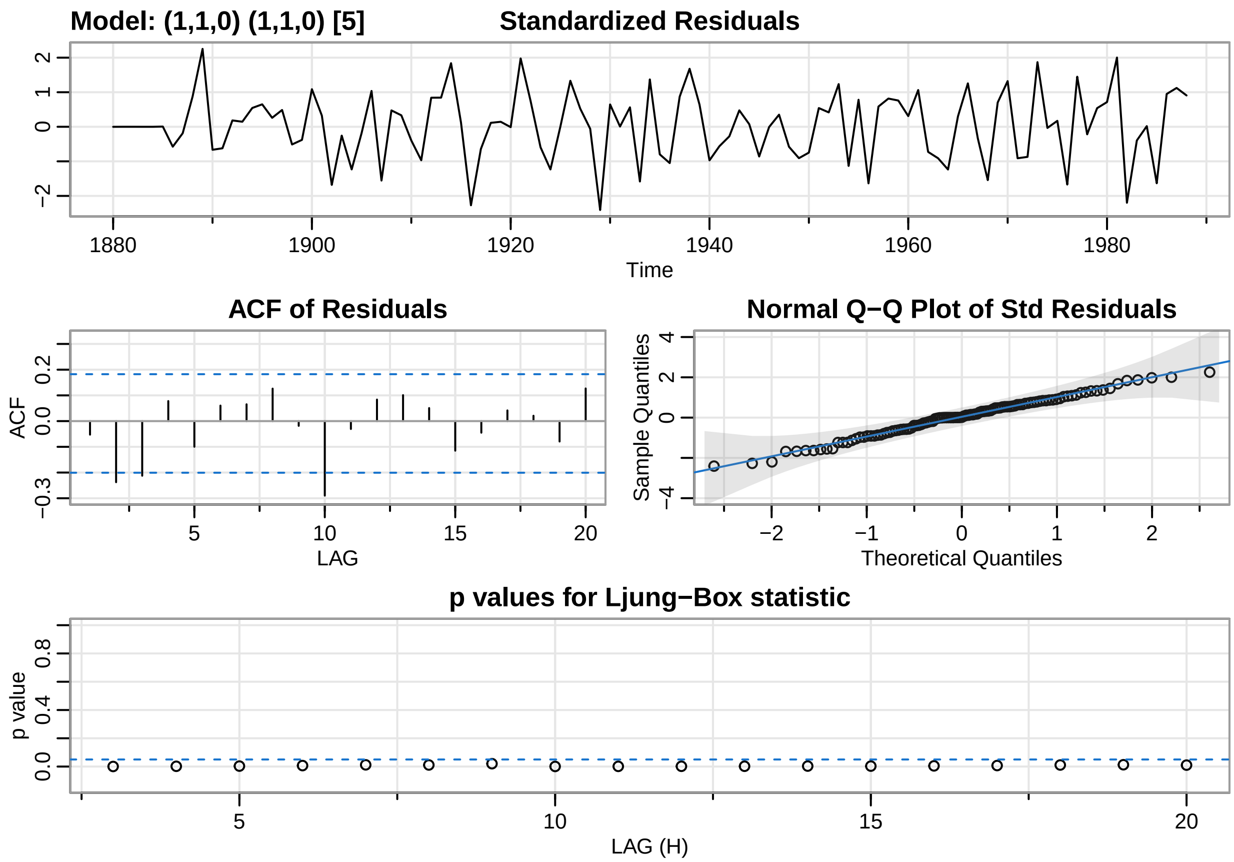
```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(diff(gtemp_land.train), 5)
## KPSS Level = 0.0194, Truncation lag parameter = 4, p-value = 0.1
```

Modeling

```
model3.1 = sarima(gtemp_land.train, p = 1, d = 1, q = 0, P = 1, D = 1, Q = 0, S = 5,
                  no.constant = TRUE)
```

```
## initial  value -1.167512
## iter   2 value -1.529010
## iter   3 value -1.544384
## iter   4 value -1.545178
## iter   5 value -1.545181
## iter   5 value -1.545181
## iter   5 value -1.545181
## final  value -1.545181
## converged
## initial  value -1.528218
## iter   2 value -1.528488
```

```
## iter   3 value -1.528552
## iter   4 value -1.528553
## iter   4 value -1.528553
## iter   4 value -1.528553
## final  value -1.528553
## converged
```



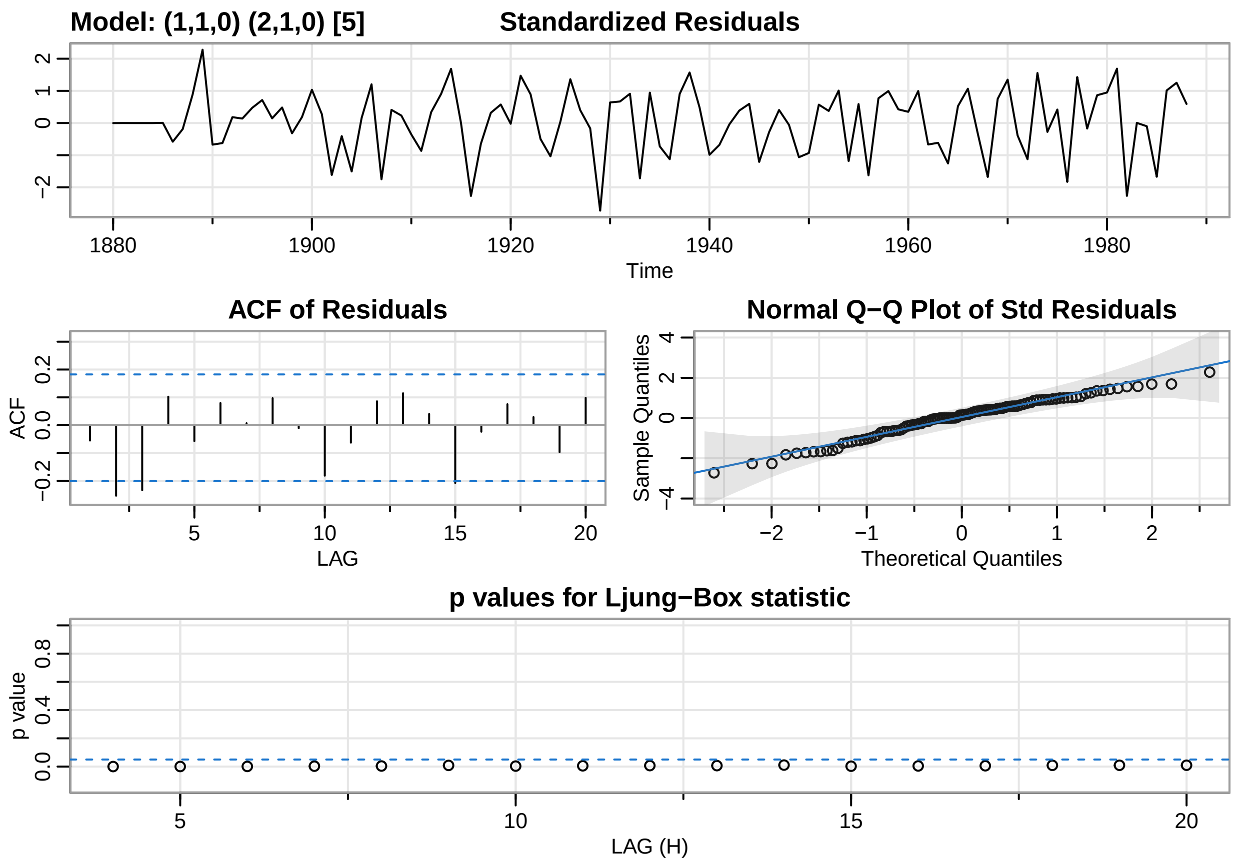

```r
model3.1$ttable
```

```
##      Estimate     SE t.value p.value
## ar1   -0.4456 0.0893 -4.9899       0
## sar1  -0.5729 0.0874 -6.5522       0
```

```r
model3.2 = sarima(gtemp_land.train, p = 1, d = 1, q = 0, P = 2, D = 1, Q = 0, S = 5,
                  no.constant = TRUE)
```

```
## initial  value -1.149383
## iter   2 value -1.428830
## iter   3 value -1.533205
## iter   4 value -1.549567
## iter   5 value -1.551103
## iter   6 value -1.551109
## iter   7 value -1.551110
## iter   7 value -1.551110
```

```
## iter   7 value -1.551110
## final  value -1.551110
## converged
## initial  value -1.553199
## iter   2 value -1.553516
## iter   3 value -1.553569
## iter   4 value -1.553569
## iter   4 value -1.553569
## iter   4 value -1.553569
## final  value -1.553569
## converged
```



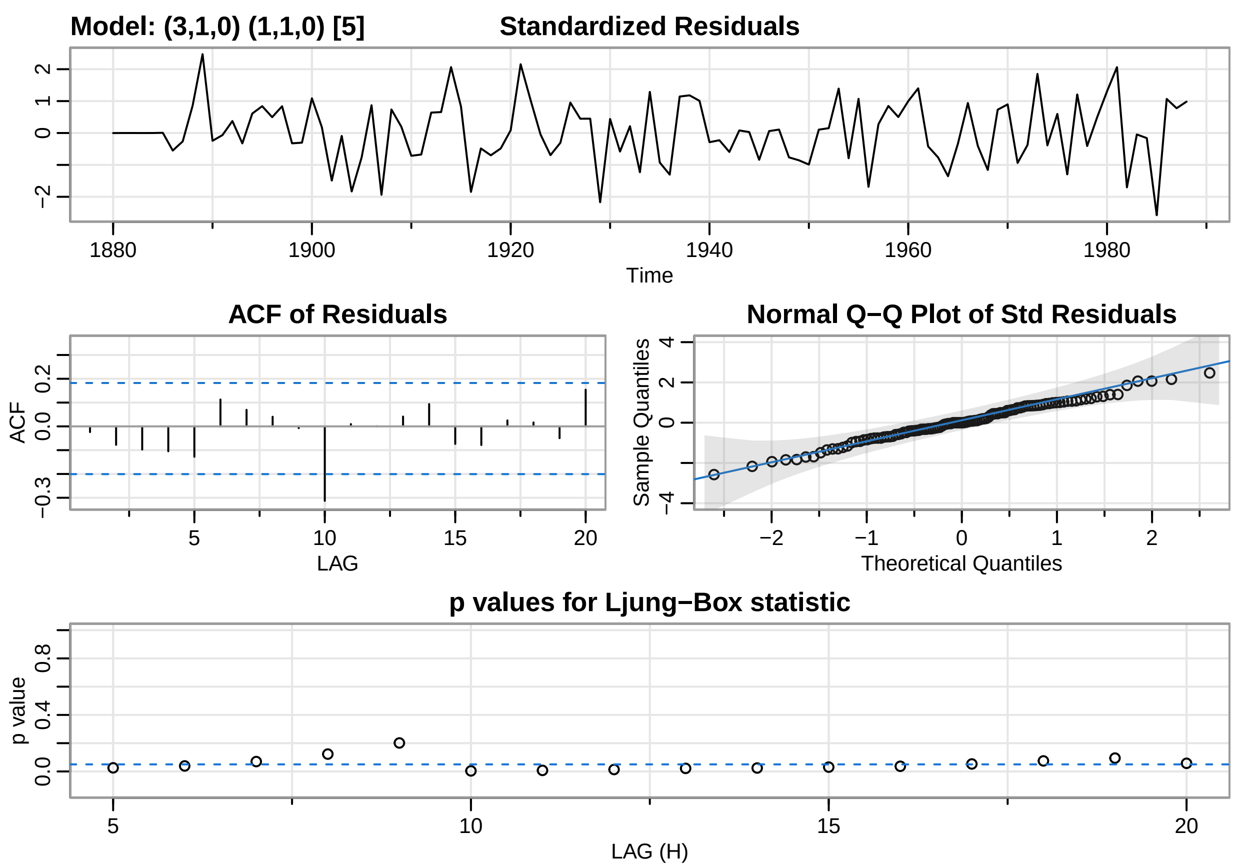

```
model3.2$ttable
```

```
##      Estimate     SE t.value p.value
## ar1   -0.4459 0.0907 -4.9143  0.0000
## sar1  -0.6922 0.1019 -6.7906  0.0000
## sar2  -0.2438 0.1054 -2.3139  0.0227
```

```
model3.3 = sarima(gtemp_land.train, p = 3, d = 1, q = 0, P = 1, D = 1, Q = 0, S = 5,
                  no.constant = TRUE)
```

```
## initial  value -1.157787
## iter   2 value -1.374284
```

```
## iter   3 value -1.570678
## iter   4 value -1.586715
## iter   5 value -1.602401
## iter   6 value -1.602928
## iter   7 value -1.602949
## iter   8 value -1.602951
## iter   8 value -1.602951
## iter   8 value -1.602951
## final  value -1.602951
## converged
## initial  value -1.591403
## iter   2 value -1.591896
## iter   3 value -1.592035
## iter   4 value -1.592073
## iter   5 value -1.592075
## iter   6 value -1.592075
## iter   6 value -1.592075
## iter   6 value -1.592075
## final  value -1.592075
## converged
```



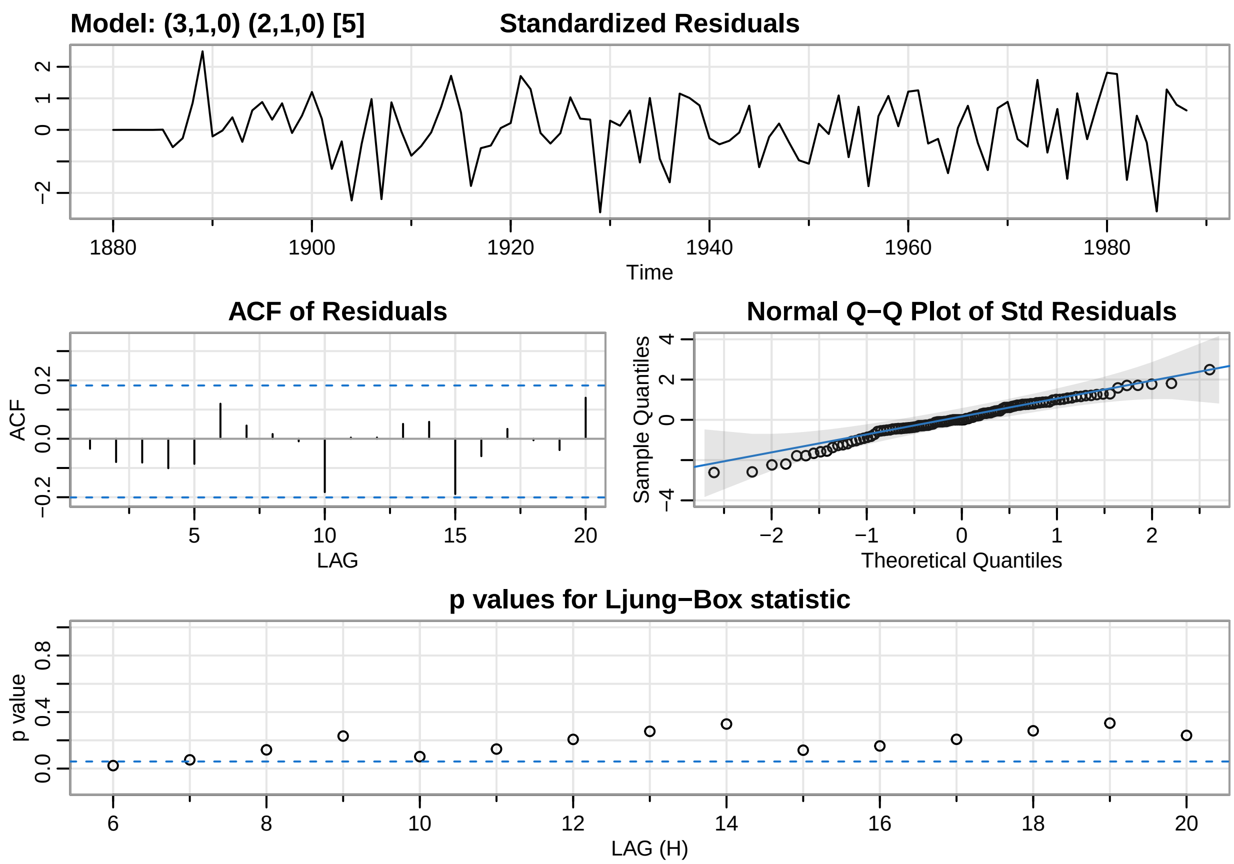

```
model3.3$ttable
```

```
##      Estimate     SE t.value p.value
## ar1   -0.5391 0.0938 -5.7498  0.0000
```

```
## ar2   -0.2909 0.1037 -2.8050  0.0061
## ar3   -0.3330 0.0942 -3.5351  0.0006
## sar1  -0.5674 0.0888 -6.3900  0.0000
```

```
model3.4 = sarima(gtemp_land.train, p = 3, d = 1, q = 0, P = 2, D = 1, Q = 0, S = 5,
                  no.constant = TRUE)
```

```
## initial  value -1.139179
## iter   2 value -1.268748
## iter   3 value -1.549786
## iter   4 value -1.569918
## iter   5 value -1.630004
## iter   6 value -1.630626
## iter   7 value -1.630814
## iter   8 value -1.630824
## iter   9 value -1.630826
## iter  10 value -1.630826
## iter  10 value -1.630826
## iter  10 value -1.630826
## final  value -1.630826
## converged
## initial  value -1.630141
## iter   2 value -1.630514
## iter   3 value -1.631011
## iter   4 value -1.631038
## iter   5 value -1.631044
## iter   6 value -1.631044
## iter   7 value -1.631044
## iter   7 value -1.631044
## iter   7 value -1.631044
## final  value -1.631044
## converged
```




```r
model3.4$ttable
```

```
##      Estimate     SE t.value p.value
## ar1   -0.5423 0.0929 -5.8378  0.0000
## ar2   -0.3142 0.1017 -3.0907  0.0026
## ar3   -0.3730 0.0940 -3.9662  0.0001
## sar1  -0.7237 0.1031 -7.0158  0.0000
## sar2  -0.3092 0.1057 -2.9247  0.0043
```

```r
###Testing for Autocorrelations###
#p = 1, P = 1
Box.test(model3.1$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  model3.1$fit$residuals
## X-squared = 33.818, df = 25, p-value = 0.1119
```

```r
#p = 1, P = 2
Box.test(model3.2$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
```

```
##
## data:  model3.2$fit$residuals
## X-squared = 33.625, df = 25, p-value = 0.1162
```

```
#p = 3, P = 1
Box.test(model3.3$fit$residuals, lag = 25, fitdf = 0)
```

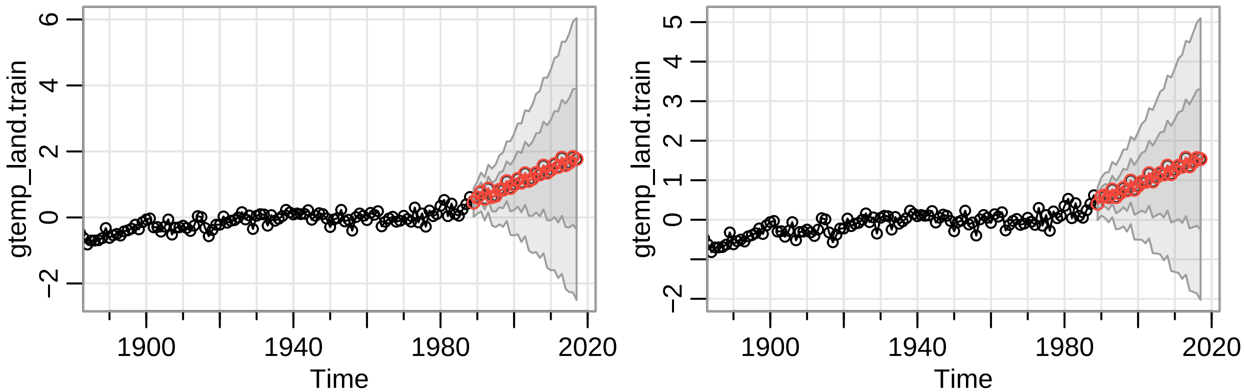
```
##
##  Box-Pierce test
##
## data:  model3.3$fit$residuals
## X-squared = 24.265, df = 25, p-value = 0.5041
```

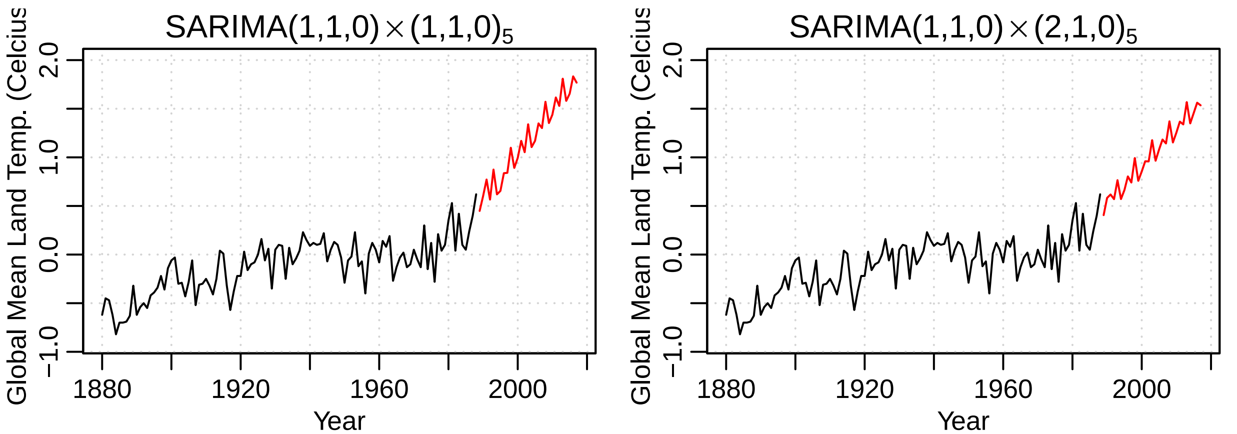
```
#p = 3, P = 2
Box.test(model3.4$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  model3.4$fit$residuals
## X-squared = 18.129, df = 25, p-value = 0.8369
```

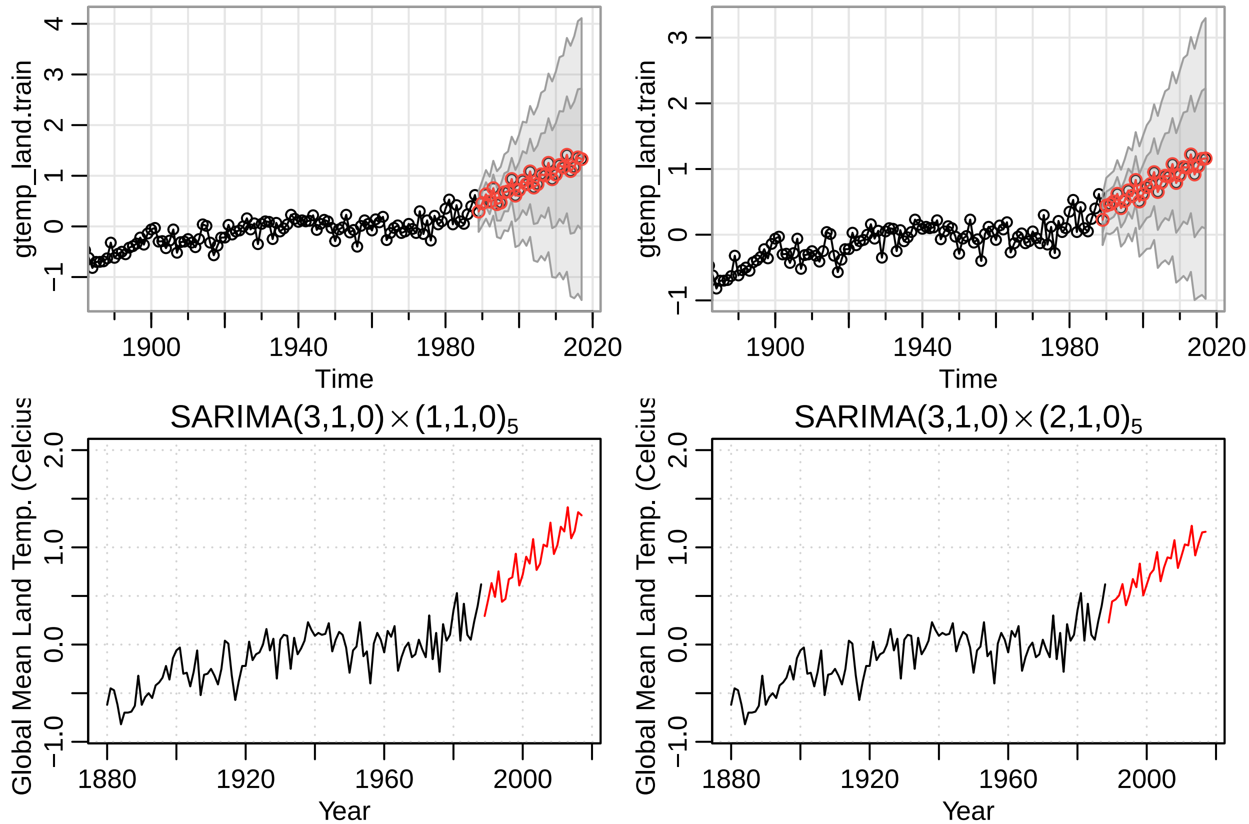
Predictions

```
par(mfrow = c(2,2))
model3.1F = sarima.for(gtemp_land.train, p = 1, d = 1, q = 0, P = 1, D = 1, Q = 0, S = 5,
                       no.constant = TRUE, n.ahead = length(gtemp_land.test))
model3.2F = sarima.for(gtemp_land.train, p = 1, d = 1, q = 0, P = 2, D = 1, Q = 0, S = 5,
                       no.constant = TRUE, n.ahead = length(gtemp_land.test))
################################################################################
plot(1880:2017,rep(0,138), col = "white", ylim = c(-0.9,2), xlab = "Year",
     ylab = "Global Mean Land Temp. (Celcius)",
     main = TeX(r"(SARIMA(1,1,0)$\times$(1,1,0)$_{5}$)"))
grid()
box()
lines(gtemp_land.train)
lines(model3.1F$pred, col = "red")
plot(1880:2017,rep(0,138), col = "white", ylim = c(-0.9,2), xlab = "Year",
     ylab = "Global Mean Land Temp. (Celcius)",
     main = TeX(r"(SARIMA(1,1,0)$\times$(2,1,0)$_{5}$)"))
grid()
box()
lines(gtemp_land.train)
lines(model3.2F$pred, col = "red")
```






```
model3.3F = sarima.for(gtemp_land.train, p = 3, d = 1, q = 0, P = 1, D = 1, Q = 0, S = 5,
                       no.constant = TRUE, n.ahead = length(gtemp_land.test))
model3.4F = sarima.for(gtemp_land.train, p = 3, d = 1, q = 0, P = 2, D = 1, Q = 0, S = 5,
                       no.constant = TRUE, n.ahead = length(gtemp_land.test))
################################################################################
plot(1880:2017,rep(0,138), col = "white", ylim = c(-0.9,2), xlab = "Year",
     ylab = "Global Mean Land Temp. (Celcius)",
     main = TeX(r"(SARIMA(3,1,0)$\times$(1,1,0)$_{5}$)"))
grid()
box()
lines(gtemp_land.train)
lines(model3.3F$pred, col = "red")
plot(1880:2017,rep(0,138), col = "white", ylim = c(-0.9,2), xlab = "Year",
     ylab = "Global Mean Land Temp. (Celcius)",
     main = TeX(r"(SARIMA(3,1,0)$\times$(2,1,0)$_{5}$)"))
grid()
box()
lines(gtemp_land.train)
lines(model3.4F$pred, col = "red")
```




```
###Accuracy###
accuracy(object = model3.1F$pred, x = gtemp_land.test)
```

```
##                   ME      RMSE       MAE       MPE    MAPE      ACF1 Theil's U
## Test set -0.3403314 0.4079919 0.3481634 -50.05216 51.1573 0.2894604  1.945276
```

```
accuracy(object = model3.2F$pred, x = gtemp_land.test)
```

```
##                   ME      RMSE       MAE       MPE     MAPE      ACF1 Theil's U
## Test set -0.1991032 0.2696612 0.2110493 -33.12065 34.68415 0.1551503  1.473968
```

```
accuracy(object = model3.3F$pred, x = gtemp_land.test)
```

```
##                    ME      RMSE       MAE       MPE     MAPE       ACF1
## Test set -0.05499039 0.2045864 0.1568246 -15.06828 27.37842 0.02852728
##          Theil's U
## Test set  1.356123
```

```
accuracy(object = model3.4F$pred, x = gtemp_land.test)
```

```
##                   ME      RMSE       MAE       MPE     MAPE      ACF1 Theil's U
## Test set 0.05453636 0.1904942 0.1668656 -1.304436 27.01191 0.1103265  1.108629
```

Final Model

$$\boxed{\text{SARIMA}(3,1,0) \times (2,1,0)_5}$$

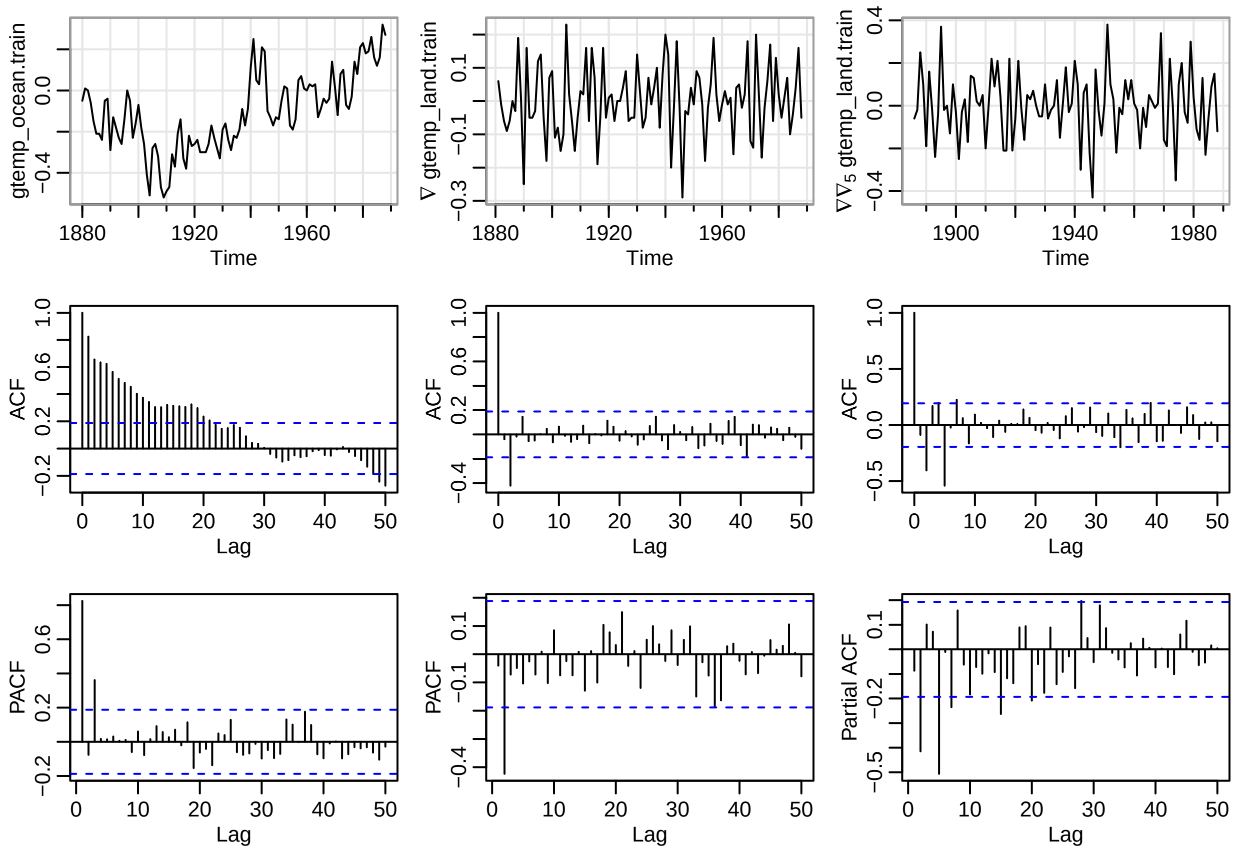
When first looking at the data, there appears to be a non-seasonal trend present. After running both `adf.test()` and `kpss.test()` I decided that there was indeed some kind of trend. The p-value for the ADF-Test was above 0.05, but the KPSS-Test p-value was 0.01. Therefore I did a difference of the data to see what would happen. I then noticed that the data was better, but when creating models, the predictions were not good. I ended up doing a seasonal difference of 5 years (10 didn't look great) and the ACF and PACF was more distinct. The ADF-Test and KPSS-Test both remained as no evidence of a trend being present. We can see that the ACF tails off and the PACF cuts off for seasonal lag 1 and is borderline for seasonal lag 2. We then see non-seasonal lags are significant for lags 1 and 3 on the PACF, suggesting $P \in [1,2]$ and $p \in [1,3]$. I first used $p = 1$ and $P = 1$, but it showed autocorrelation between observations. $P = 2$ and $p = 3$ and $P = 1$ also showed autocorrelations between observations. While the Box-Pierce test did not show up anything, the Ljung-Box test did. The final model with $p = 3$ and $P = 2$ was the one that did not and had great diagnostics. Its residual time series plot looks similar to white noise, the ACF of the residuals does not have any significant lags, the Normal QQ-Plot follows the line, and to put the icing on the cake, all coefficient estimates are significant. It also has the lowest AIC and BIC of the four models. It also has the lowest RMSE, so that is why I would choose this as the final model.

Question 4.

```
gtemp_ocean.train = ts.train.test(gtemp_ocean, 1)$train
gtemp_ocean.test = ts.train.test(gtemp_ocean, 1)$test
```

Plots

```
par(mfrow = c(3,3))
tsplot(gtemp_ocean.train)
tsplot(diff(gtemp_ocean.train), ylab = TeX(r"($\nabla$ gtemp_land.train)"))
tsplot(diff(diff(gtemp_ocean.train), 5), ylab = TeX(r"($\nabla\nabla_{5}$ gtemp_land.train)"))
acf(gtemp_ocean.train, lag.max = 50)
acf(diff(gtemp_ocean.train), lag.max = 50)
acf(diff(diff(gtemp_ocean.train), 5), lag.max = 50)
pacf(gtemp_ocean.train, ylab = "PACF", lag.max = 50)
pacf(diff(gtemp_ocean.train), ylab = "PACF", lag.max = 50)
pacf(diff(diff(gtemp_ocean.train), 5), lag.max = 50)
```



Tests

```r
#1
adf.test(gtemp_ocean.train)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gtemp_ocean.train
## Dickey-Fuller = -2.7915, Lag order = 4, p-value = 0.2483
## alternative hypothesis: stationary
```

```r
kpss.test(gtemp_ocean.train)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  gtemp_ocean.train
## KPSS Level = 1.3107, Truncation lag parameter = 4, p-value = 0.01
```

```r
#2; d = 1
adf.test(diff(gtemp_ocean.train))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(gtemp_ocean.train)
## Dickey-Fuller = -6.1955, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(gtemp_ocean.train))
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(gtemp_ocean.train)
## KPSS Level = 0.12782, Truncation lag parameter = 4, p-value = 0.1
```

```
#3; d = 1, D = 1
adf.test(diff(diff(gtemp_ocean.train), 5))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(diff(gtemp_ocean.train), 5)
## Dickey-Fuller = -7.864, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(diff(diff(gtemp_ocean.train), 5))
```

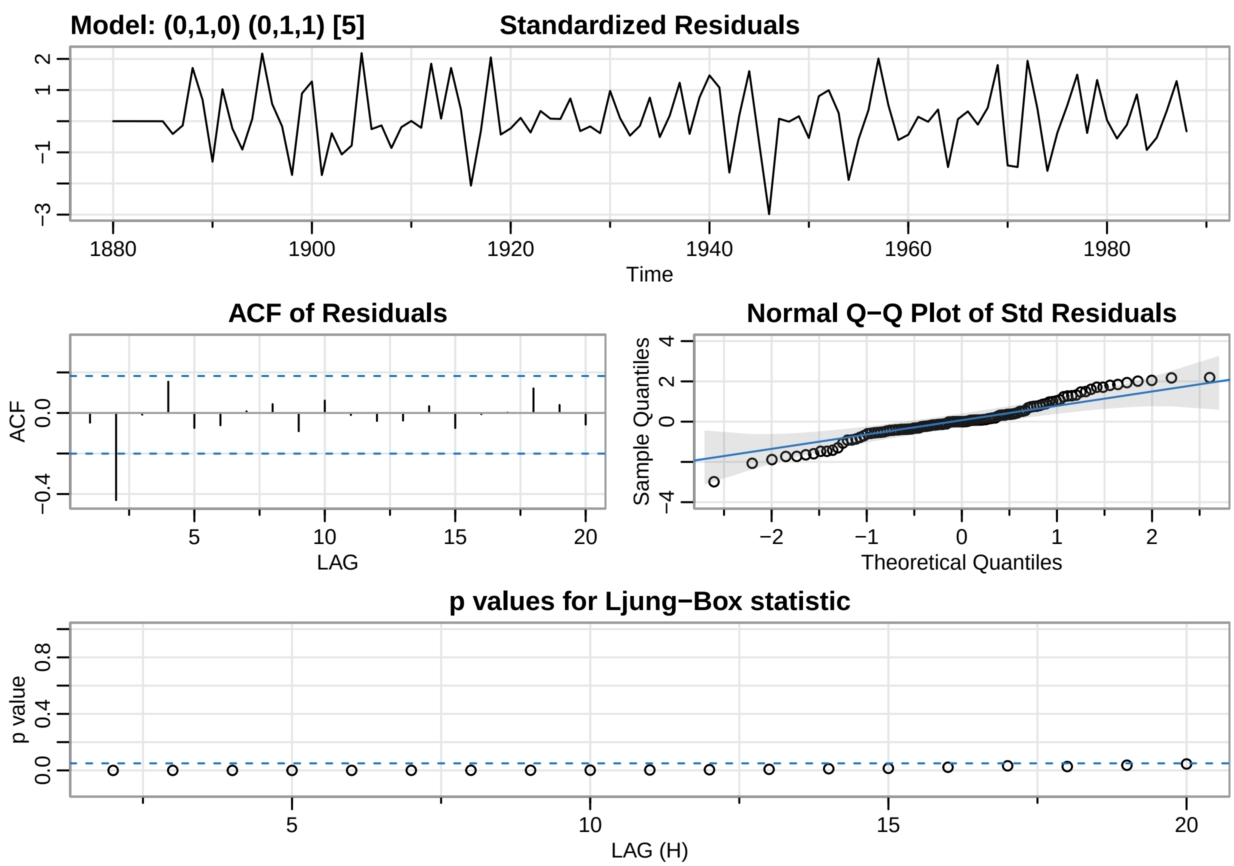
```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(diff(gtemp_ocean.train), 5)
## KPSS Level = 0.023421, Truncation lag parameter = 4, p-value = 0.1
```

Modeling

```
model4.1 = sarima(gtemp_ocean.train, p = 0, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 5,
                  no.constant = TRUE)
```

```
## initial  value -1.885603
## iter   2 value -2.114633
## iter   3 value -2.170046
## iter   4 value -2.191335
## iter   5 value -2.193260
## iter   6 value -2.193795
## iter   7 value -2.193856
## iter   8 value -2.193857
## iter   9 value -2.193857
## iter   9 value -2.193857
## iter   9 value -2.193857
```

```
## final  value -2.193857
## converged
## initial  value -2.183238
## iter   2 value -2.184676
## iter   3 value -2.184869
## iter   4 value -2.184891
## iter   5 value -2.184896
## iter   6 value -2.184896
## iter   6 value -2.184896
## iter   6 value -2.184896
## final  value -2.184896
## converged
```



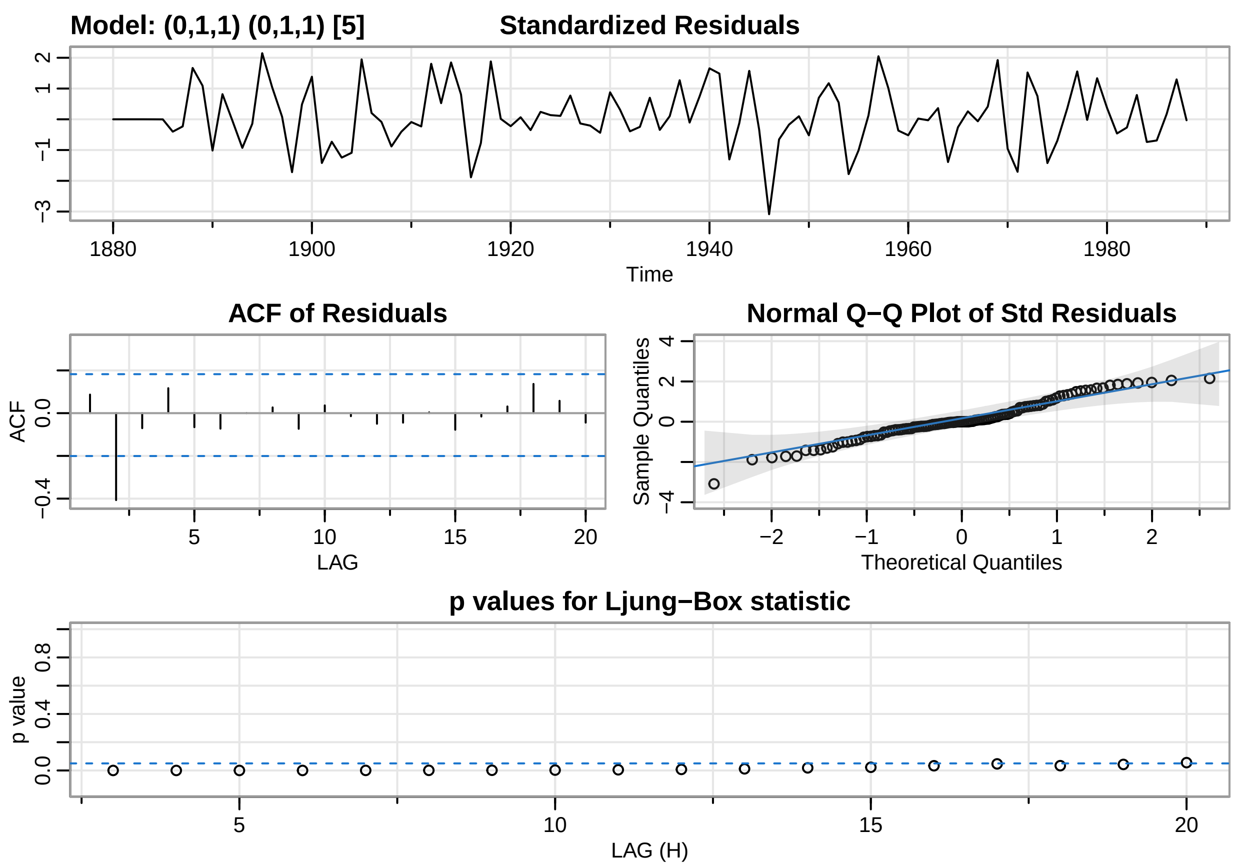

```r
model4.1$ttable
```

```
##      Estimate     SE t.value p.value
## sma1  -0.9221 0.0926 -9.9535       0
```

```r
model4.2 = sarima(gtemp_ocean.train, p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 5,
                  no.constant = TRUE)
```

```
## initial  value -1.885603
## iter   2 value -2.119125
## iter   3 value -2.173067
```

```
## iter   4 value -2.193150
## iter   5 value -2.194888
## iter   6 value -2.195923
## iter   7 value -2.196035
## iter   8 value -2.196061
## iter   9 value -2.196076
## iter  10 value -2.196084
## iter  11 value -2.196085
## iter  12 value -2.196085
## iter  12 value -2.196085
## iter  12 value -2.196085
## final  value -2.196085
## converged
## initial  value -2.187877
## iter   2 value -2.189864
## iter   3 value -2.190342
## iter   4 value -2.190593
## iter   5 value -2.190697
## iter   6 value -2.190733
## iter   6 value -2.190733
## final  value -2.190733
## converged
```



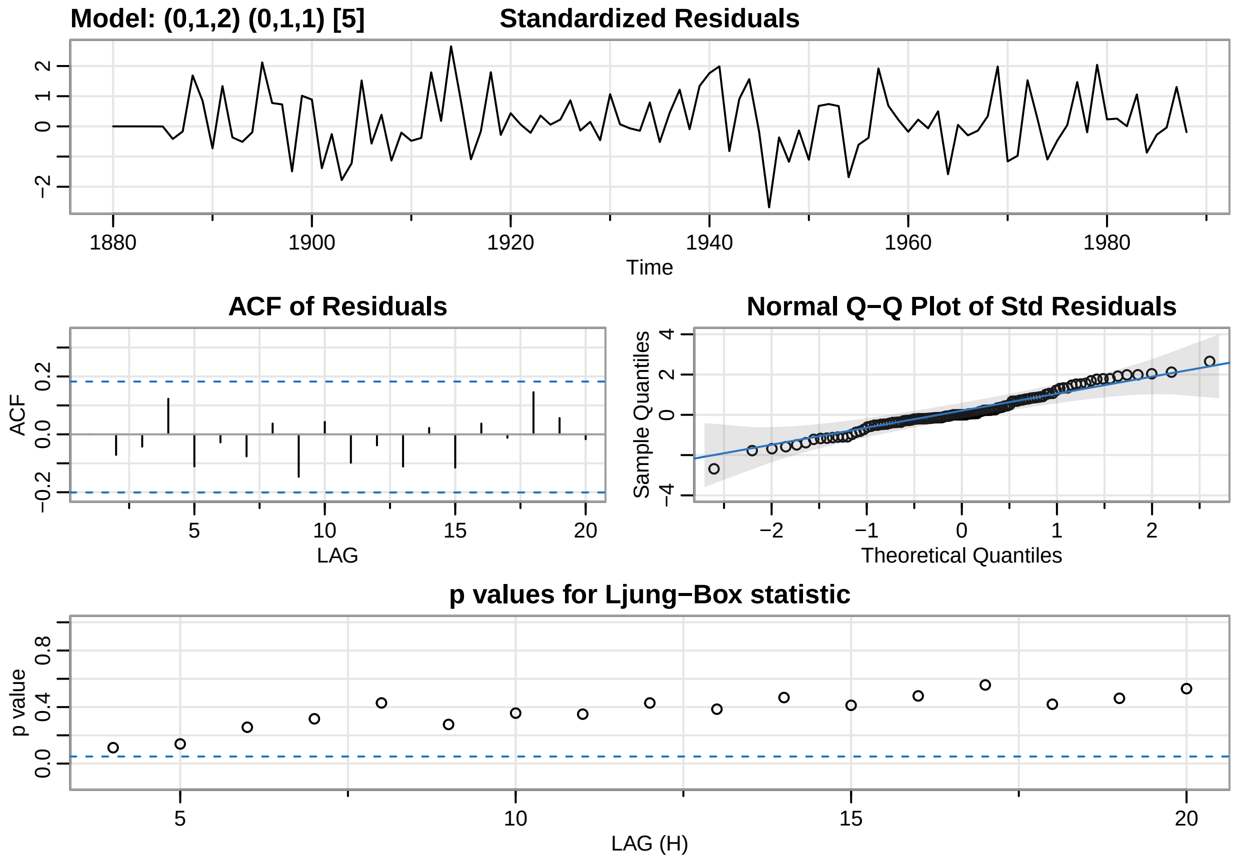

```
model4.2$ttable
```

```
##       Estimate     SE  t.value p.value
```

```
## ma1   -0.2399 0.1950  -1.2301  0.2215
## sma1  -0.9110 0.0862 -10.5665  0.0000
```

```
model4.3 = sarima(gtemp_ocean.train, p = 0, d = 1, q = 2, P = 0, D = 1, Q = 1, S = 5,
                  no.constant = TRUE)
```

```
## initial  value -1.885603
## iter   2 value -2.204088
## iter   3 value -2.251465
## iter   4 value -2.268903
## iter   5 value -2.272229
## iter   6 value -2.274753
## iter   7 value -2.274920
## iter   8 value -2.274945
## iter   9 value -2.274945
## iter   9 value -2.274945
## iter   9 value -2.274945
## final  value -2.274945
## converged
## initial  value -2.272615
## iter   2 value -2.281060
## iter   3 value -2.281616
## iter   4 value -2.281796
## iter   5 value -2.282064
## iter   6 value -2.282127
## iter   7 value -2.282136
## iter   8 value -2.282136
## iter   8 value -2.282136
## final  value -2.282136
## converged
```




```
model4.3$ttable
```

```
##       Estimate     SE  t.value p.value
## ma1    -0.1401 0.0887  -1.5791  0.1175
## ma2    -0.4120 0.0834  -4.9377  0.0000
## sma1   -0.9224 0.0918 -10.0499  0.0000
```

```
###Testing for Autocorrelations###
#q = 1
Box.test(model4.1$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  model4.1$fit$residuals
## X-squared = 29.491, df = 25, p-value = 0.2439
```

```
#q = 1
Box.test(model4.2$fit$residuals, lag = 25, fitdf = 0)
```

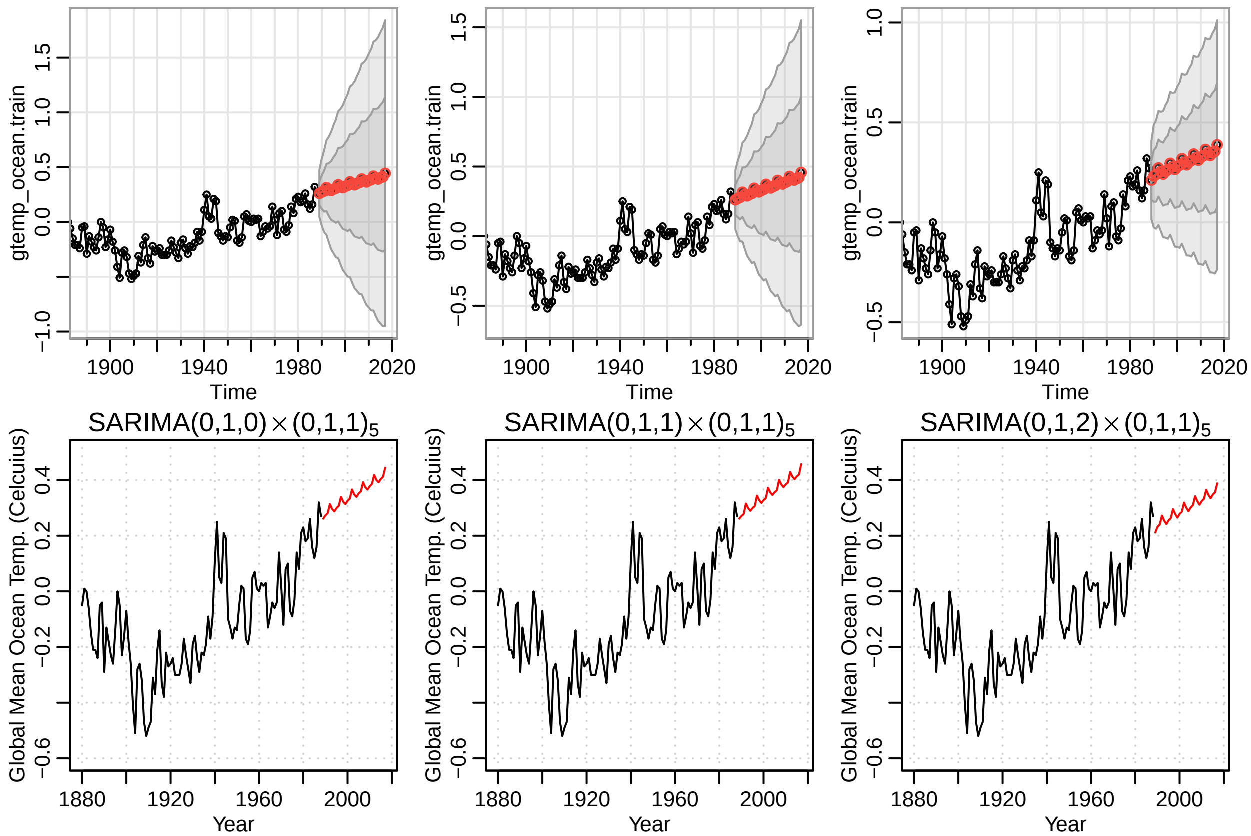
```
##
##  Box-Pierce test
##
## data:  model4.2$fit$residuals
## X-squared = 27.696, df = 25, p-value = 0.322
```

```r
#q = 3
Box.test(model4.3$fit$residuals, lag = 25, fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  model4.3$fit$residuals
## X-squared = 14.553, df = 25, p-value = 0.9512
```

Predictions

```r
par(mfrow = c(2,3))
model4.1F = sarima.for(gtemp_ocean.train, p = 0, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 5,
                       no.constant = TRUE, n.ahead = length(gtemp_ocean.test))
model4.2F = sarima.for(gtemp_ocean.train, p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 5,
                       no.constant = TRUE, n.ahead = length(gtemp_ocean.test))
model4.3F = sarima.for(gtemp_ocean.train, p = 0, d = 1, q = 2, P = 0, D = 1, Q = 1, S = 5,
                       no.constant = TRUE, n.ahead = length(gtemp_ocean.test))
################################################################################
plot(1880:2017,rep(0,138), col = "white", ylim = c(-0.6,0.5), xlab = "Year",
     ylab = "Global Mean Ocean Temp. (Celcuius)",
     main = TeX(r"(SARIMA(0,1,0)$\times$(0,1,1)$_{5}$)"))
grid()
box()
lines(gtemp_ocean.train)
lines(model4.1F$pred, col = "red")
plot(1880:2017,rep(0,138), col = "white", ylim = c(-0.6,0.5), xlab = "Year",
     ylab = "Global Mean Ocean Temp. (Celcuius)",
     main = TeX(r"(SARIMA(0,1,1)$\times$(0,1,1)$_{5}$)"))
grid()
box()
lines(gtemp_ocean.train)
lines(model4.2F$pred, col = "red")
plot(1880:2017,rep(0,138), col = "white", ylim = c(-0.6,0.5), xlab = "Year",
     ylab = "Global Mean Ocean Temp. (Celcuius)",
     main = TeX(r"(SARIMA(0,1,2)$\times$(0,1,1)$_{5}$)"))
grid()
box()
lines(gtemp_ocean.train)
lines(model4.3F$pred, col = "red")
```




```
###Accuracy###
accuracy(object = model4.1F$pred, x = gtemp_ocean.test)
```

```
##                   ME      RMSE        MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.04153824 0.1083051 0.08542298 3.502928 21.16637 0.6146684 0.9892725
```

```
accuracy(object = model4.2F$pred, x = gtemp_ocean.test)
```

```
##                   ME      RMSE        MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.03636555 0.1045261 0.08340984 2.293885 20.97669 0.6002874 0.9678524
```

```
accuracy(object = model4.3F$pred, x = gtemp_ocean.test)
```

```
##                   ME      RMSE       MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.09056293 0.1371448 0.1027468 17.28145 22.57965 0.6350994   1.22821
```

Final Model

$$\boxed{\text{SARIMA}(0,1,0) \times (0,1,1)_5}$$

When first looking at the data, there appears to be a non-seasonal trend present. After running both `adf.test()` and `kpss.test()` I decided that there was indeed some kind of trend. The p-value for the ADF-Test was above 0.05, but the KPSS-Test p-value was 0.01. Therefore I did a difference of the data to see what would happen. I then noticed that the data was better, but when creating models, the predictions were not good (just like problem 3). I ended up doing a seasonal difference of 5 years (again, 10 didn't look great) and the ACF and PACF was more distinct. The ADF-Test and KPSS-Test both remained as no evidence of a trend being present. We can see that the ACF cuts off at seasonal lag 1 and the PACF tails off. We then see non-seasonal lags are significant for lag 2 on the ACF, suggesting $Q = 1$ and $q \in [0, 2]$. I tried out three different models, one for each of the three potential model orders of q. While the Box-Pierce test did not show any autocorrelation, the Ljung-Box test did for the first two models. The other diagnostics such as the standardized residual time series plot, ACF of the residuals, and Normal QQ-Plot were approximately the same for each one and the results showed no signs of any model issues. The final model with $q = 2$ showed no signs of autocorrelation and had the lowest AIC and BIC, but the second model had the best RMSE. I ended up deciding to use the first model. It has the second best RMSE and second lowest AIC/BIC. While one test shows it may have autocorrelation between observations, it is borderline so it is close to not having autocorrelation, which is what the Box-Pierce test says. $\hat{\theta}_1$ is not statistically significant in the second model, so I did not think it would make sense to use that one over the first model. The third model also has the highest RMSE of the three, so I did not think it would make sense to use that one either.