Chapter 5 Homework

Orly Olbum

10/29/2020

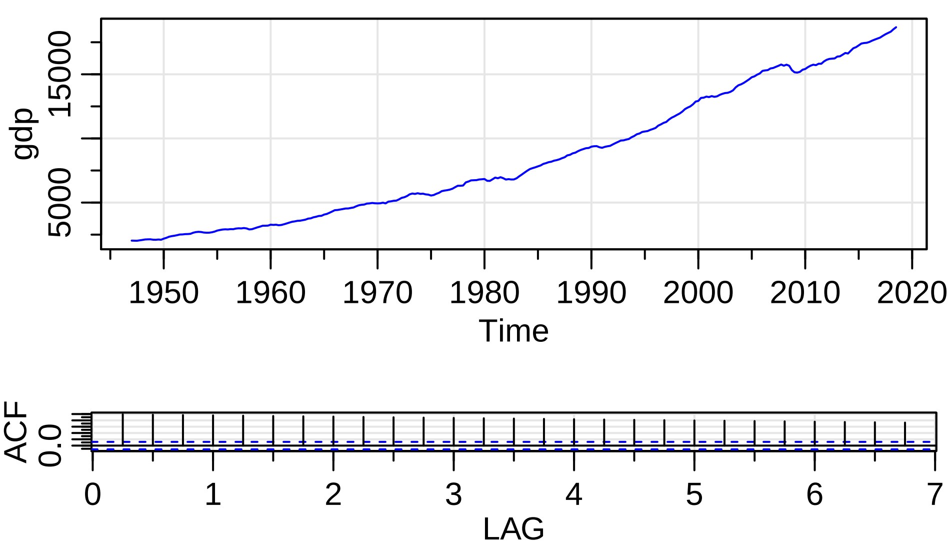
5.2

In Example 5.6, we fit an ARIMA model to the quarterly GNP series. Repeat the analysis for the US GDP series in gdp. Discuss all aspects of the fit as specified in the points at the beginning of Section 5.2 from plotting the data to diagnostics and model choice.

To fit an ARIMA model to quarterly data, we follow the following steps: - plot the data - possibly transform the data - identify dependence orders of the model - estimate the parameters - display diagnostics - choose a model

First we graph a tsplot of the gdp data and see what the ACF looks like.

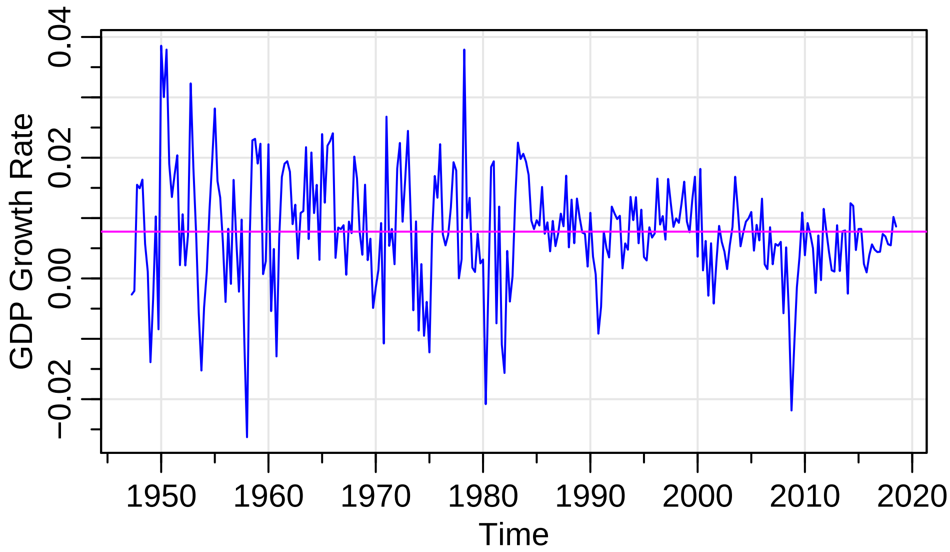
```
layout(1:2, heights = 2:1)
tsplot(gdp, col = 4)
acf1(gdp, main = "")
```




```
##  [1] 0.99 0.98 0.97 0.96 0.95 0.94 0.93 0.92 0.91 0.90 0.89 0.88 0.87 0.86 0.85
## [16] 0.84 0.83 0.82 0.81 0.80 0.79 0.78 0.77 0.76 0.75 0.74 0.73
```

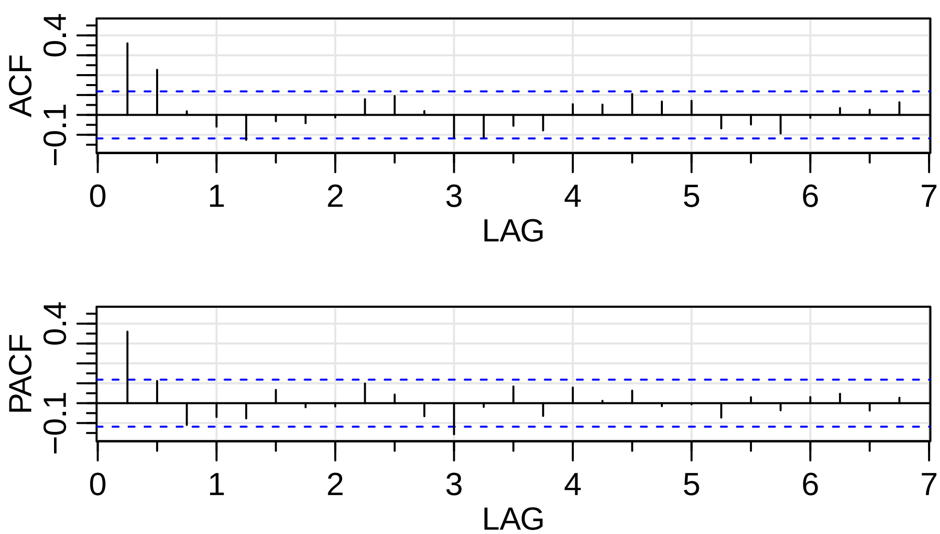
The plot shows a steady incline of gdp data over time, and the trend covers up other potential effects. We might use log to display the data in terms of growth rate rather than actual data.

```r
par(mfrow = c(1, 1))
tsplot(diff(log(gdp)), ylab = "GDP Growth Rate", col = 4)
mean_dif = mean(diff(log(gdp)))
abline(h = mean_dif, col = 6)
```



The logged/differenced data is much more stable. Now we can investigate the ACF and PACF to fit the ARIMA model.

```r
acf2(diff(log(gdp)), main = "")
```



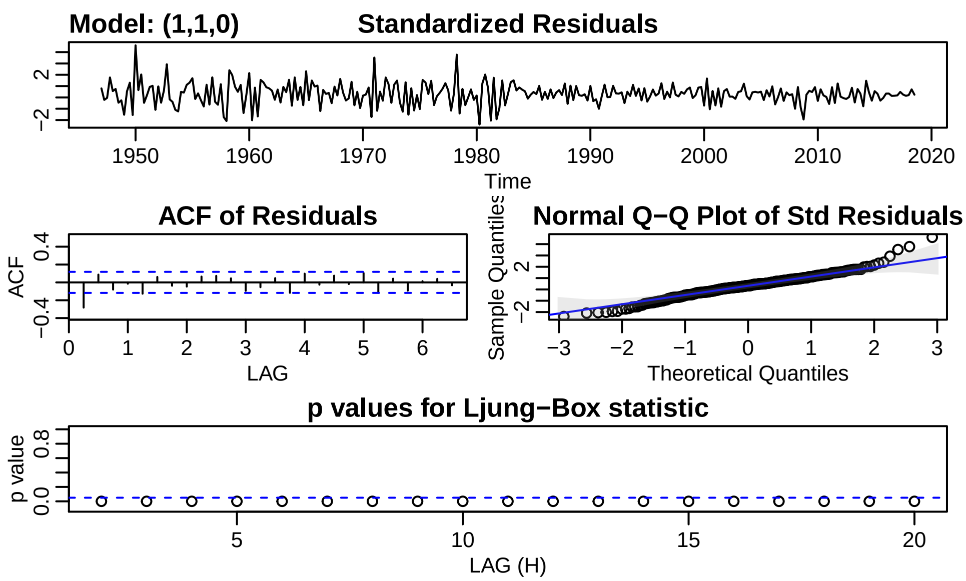

```
##       [,1] [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.36 0.23  0.02 -0.06 -0.13 -0.03 -0.04 -0.01 0.08  0.10  0.02 -0.11 -0.11
## PACF 0.36 0.11 -0.11 -0.07 -0.08  0.07 -0.02 -0.02 0.10  0.04 -0.07 -0.16 -0.02
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.05 -0.08  0.05  0.05  0.11  0.07  0.07 -0.07 -0.05 -0.09 -0.02  0.03
```

```
## PACF  0.08 -0.06  0.08  0.01  0.06 -0.02 -0.01 -0.07  0.03 -0.04  0.03  0.05
##      [,26] [,27]
## ACF   0.03  0.06
## PACF -0.04  0.03
```

We can see that the ACF cuts off around lag 2 and the PACF cuts off around 2, which suggests an ARMA(1, 2) process after differencing, or ARIMA(1, 1, 2) on the logged data log(gdp). We now can fit the model and analyze the diagnostics. First we should try AR(1) and MA(2) alone to check out the diagnostics and compare to the ARIMA model proposed.

```r
sarima(log(gdp), 1, 1, 0, no.constant = TRUE)
```

```
## initial  value -4.409274
## iter   2 value -4.653694
## iter   3 value -4.653694
## iter   4 value -4.653694
## iter   4 value -4.653694
## final  value -4.653694
## converged
## initial  value -4.654504
## iter   2 value -4.654507
## iter   3 value -4.654508
## iter   3 value -4.654508
## iter   3 value -4.654508
## final  value -4.654508
## converged
```

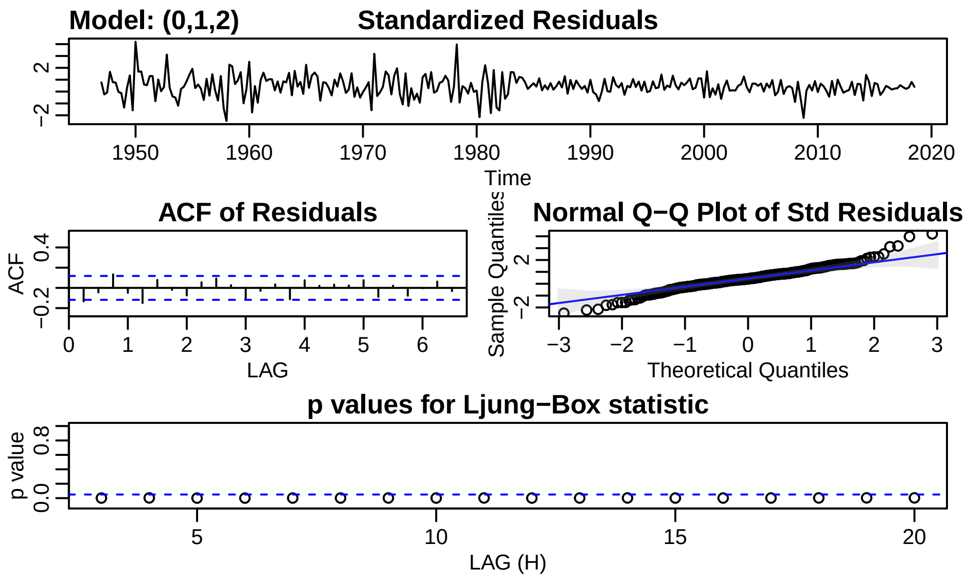



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
```

```
##
## Coefficients:
##          ar1
##       0.6203
## s.e.  0.0461
##
## sigma^2 estimated as 9.045e-05:  log likelihood = 925.37,  aic = -1846.75
##
## $degrees_of_freedom
## [1] 285
##
## $ttable
##     Estimate     SE t.value p.value
## ar1   0.6203 0.0461 13.4416       0
##
## $AIC
## [1] -6.457153
##
## $AICc
## [1] -6.457103
##
## $BIC
## [1] -6.431586
```

```r
sarima(log(gdp), 0, 1, 2, no.constant = TRUE)
```

```
## initial  value -4.410941
## iter   2 value -4.571030
## iter   3 value -4.612413
## iter   4 value -4.623939
## iter   5 value -4.625520
## iter   6 value -4.625527
## iter   7 value -4.625527
## iter   7 value -4.625527
## iter   7 value -4.625527
## final  value -4.625527
## converged
## initial  value -4.625172
## iter   2 value -4.625173
## iter   2 value -4.625173
## iter   2 value -4.625173
## final  value -4.625173
## converged
```



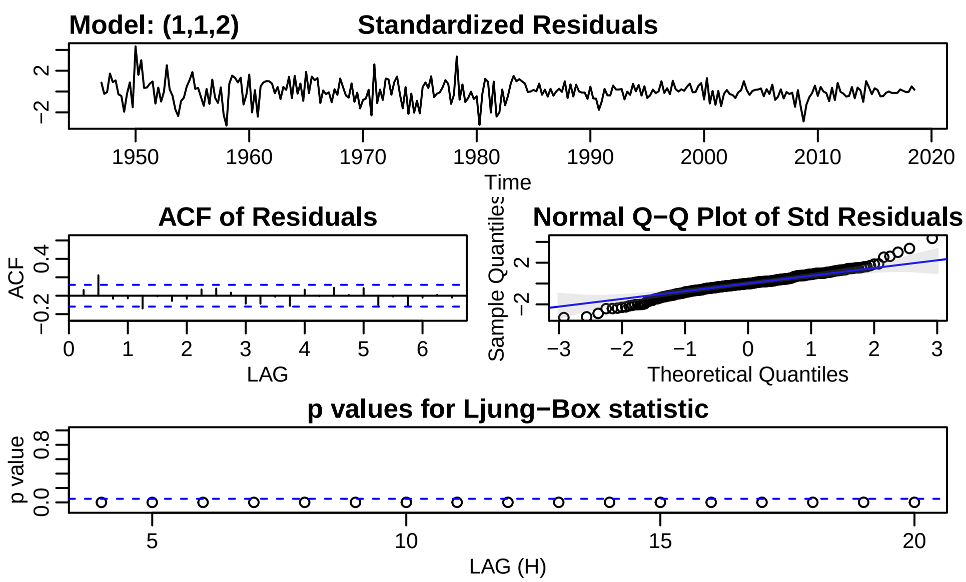

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ma1     ma2
##       0.4560  0.3341
## s.e.  0.0607  0.0473
##
## sigma^2 estimated as 9.596e-05:  log likelihood = 916.98,  aic = -1827.97
##
## $degrees_of_freedom
## [1] 284
##
## $ttable
##     Estimate     SE t.value p.value
## ma1   0.4560 0.0607  7.5096       0
## ma2   0.3341 0.0473  7.0653       0
##
## $AIC
## [1] -6.39149
##
## $AICc
## [1] -6.391342
##
## $BIC
## [1] -6.353141
```

```r
sarima(log(gdp), 1, 1, 2, no.constant = TRUE)
```

```
## initial  value -4.409274
```

```
## iter   2 value -4.575603
## iter   3 value -4.657000
## iter   4 value -4.662588
## iter   5 value -4.670962
## iter   6 value -4.680038
## iter   7 value -4.685092
## iter   8 value -4.686879
## iter   9 value -4.687059
## iter  10 value -4.687137
## iter  11 value -4.687328
## iter  12 value -4.687746
## iter  13 value -4.688211
## iter  14 value -4.688439
## iter  15 value -4.692273
## iter  16 value -4.693780
## iter  17 value -4.697493
## iter  18 value -4.697681
## iter  19 value -4.697708
## iter  20 value -4.697730
## iter  21 value -4.697732
## iter  22 value -4.697732
## iter  22 value -4.697732
## final  value -4.697732
## converged
## initial  value -4.705641
## iter   2 value -4.709167
## iter   3 value -4.709318
## iter   4 value -4.709768
## iter   5 value -4.710408
## iter   6 value -4.710814
## iter   7 value -4.710946
## iter   8 value -4.713375
## iter   9 value -4.713640
## iter  10 value -4.714262
## iter  11 value -4.714305
## iter  12 value -4.714345
## iter  13 value -4.714545
## iter  14 value -4.714875
## iter  15 value -4.715446
## iter  16 value -4.715451
## iter  17 value -4.715794
## iter  18 value -4.715800
## iter  19 value -4.715811
## iter  20 value -4.715842
## iter  21 value -4.715904
## iter  22 value -4.716015
## iter  23 value -4.716034
## iter  24 value -4.716098
## iter  25 value -4.716102
## iter  26 value -4.716103
## iter  27 value -4.716115
## iter  28 value -4.716134
## iter  29 value -4.716173
## iter  30 value -4.716175
```

```
## iter  31 value -4.716195
## iter  32 value -4.716200
## iter  33 value -4.716200
## iter  34 value -4.716210
## iter  35 value -4.716219
## iter  36 value -4.716231
## iter  37 value -4.716233
## iter  38 value -4.716234
## iter  39 value -4.716234
## iter  39 value -4.716234
## iter  39 value -4.716234
## final  value -4.716234
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1      ma2
##       0.9997  -0.7205  -0.2591
## s.e.  0.0006   0.0479   0.0472
##
## sigma^2 estimated as 7.931e-05:  log likelihood = 943.03,  aic = -1878.05
##
## $degrees_of_freedom
## [1] 283
##
## $ttable
##     Estimate     SE   t.value p.value
## ar1   0.9997 0.0006 1760.8080       0
```

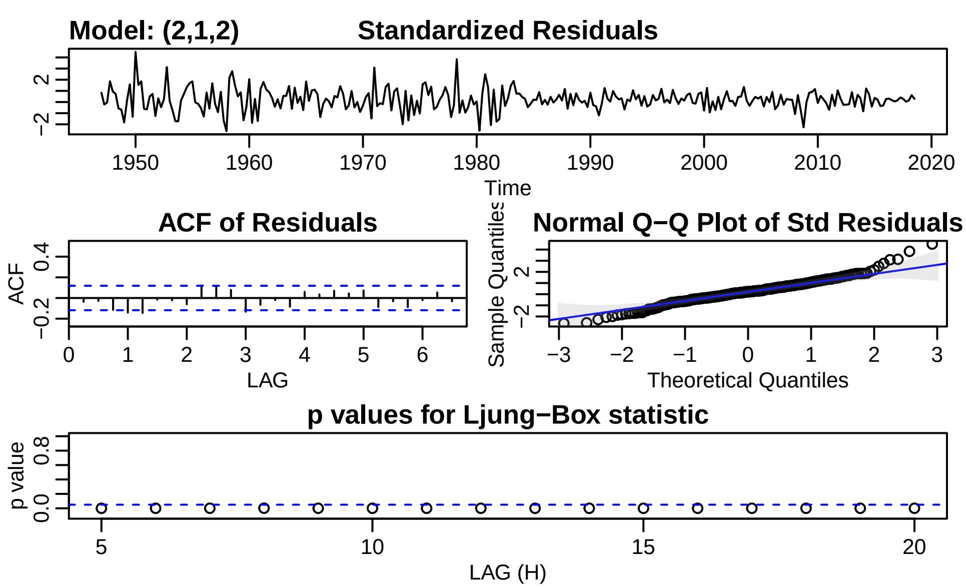
```
## ma1  -0.7205 0.0479  -15.0370       0
## ma2  -0.2591 0.0472   -5.4867       0
##
## $AIC
## [1] -6.56662
##
## $AICc
## [1] -6.566322
##
## $BIC
## [1] -6.515487
```

We see that the ACF of residuals is essentially all 0, the q-q plot shows normal diagnostics, and the p-values are where we want them. We might try a higher order to see if it is even better, and if not, use AIC or BIC to choose the best.

```r
sarima(log(gdp), 2, 1, 2, no.constant = TRUE)
```

```
## initial  value -4.407567
## iter   2 value -4.611974
## iter   3 value -4.674509
## iter   4 value -4.681211
## iter   5 value -4.683491
## iter   6 value -4.687128
## iter   7 value -4.687295
## iter   8 value -4.687315
## iter   9 value -4.687327
## iter  10 value -4.687396
## iter  11 value -4.687423
## iter  12 value -4.687441
## iter  13 value -4.687467
## iter  14 value -4.687513
## iter  15 value -4.687587
## iter  16 value -4.687692
## iter  17 value -4.687872
## iter  18 value -4.688091
## iter  19 value -4.688336
## iter  20 value -4.688425
## iter  21 value -4.688551
## iter  22 value -4.688559
## iter  23 value -4.688682
## iter  24 value -4.688710
## iter  25 value -4.688719
## iter  26 value -4.688725
## iter  27 value -4.688744
## iter  28 value -4.688791
## iter  29 value -4.688806
## iter  30 value -4.688816
## iter  31 value -4.688839
## iter  32 value -4.688862
## iter  33 value -4.688872
## iter  34 value -4.688885
## iter  35 value -4.688931
```

```
## iter  36 value -4.688974
## iter  37 value -4.689027
## iter  38 value -4.689080
## iter  39 value -4.689206
## iter  40 value -4.689322
## iter  41 value -4.689476
## iter  42 value -4.689613
## iter  43 value -4.689656
## iter  44 value -4.689698
## iter  45 value -4.689742
## iter  46 value -4.689785
## iter  47 value -4.689789
## iter  48 value -4.689791
## iter  49 value -4.689794
## iter  50 value -4.689795
## iter  51 value -4.689796
## iter  51 value -4.689796
## iter  51 value -4.689796
## final  value -4.689796
## converged
## initial  value -4.692166
## iter   2 value -4.692169
## iter   3 value -4.692183
## iter   4 value -4.692184
## iter   5 value -4.692184
## iter   6 value -4.692185
## iter   6 value -4.692185
## iter   6 value -4.692185
## final  value -4.692185
## converged
```




```
## $fit
##
## Call:
```

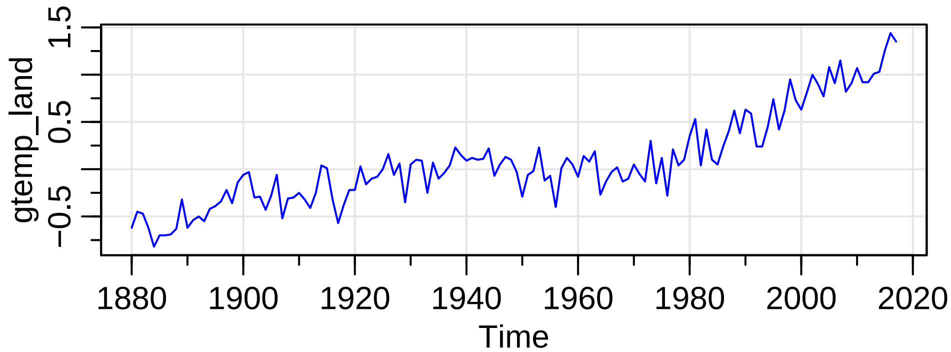
```
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     ar2     ma1      ma2
##       -0.1292  0.8286  0.5741  -0.3684
## s.e.   0.0637  0.0656  0.1048   0.1053
##
## sigma^2 estimated as 8.381e-05:  log likelihood = 936.15,  aic = -1862.3
##
## $degrees_of_freedom
## [1] 282
##
## $ttable
##     Estimate     SE t.value p.value
## ar1  -0.1292 0.0637 -2.0272  0.0436
## ar2   0.8286 0.0656 12.6402  0.0000
## ma1   0.5741 0.1048  5.4757  0.0000
## ma2  -0.3684 0.1053 -3.4969  0.0005
##
## $AIC
## [1] -6.511527
##
## $AICc
## [1] -6.51103
##
## $BIC
## [1] -6.447611
```

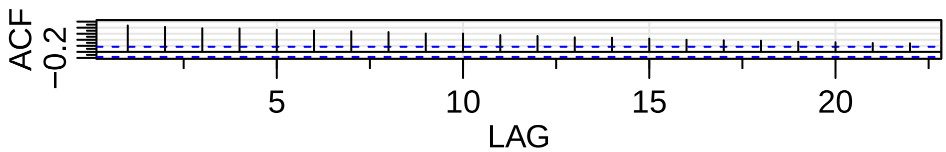
There is not significant change, and the AIC and BIC point towards using the ARIMA(1, 1, 2) model to fit the logged gdp data.

5.4

Fit an ARIMA(p, d, q) model to gtemp_land, the land-based global temperature data, performing all of the necessary diagnostics; include a model choice analysis. After deciding on an appropriate model, forecast (with limits) the next 10 years. Comment.

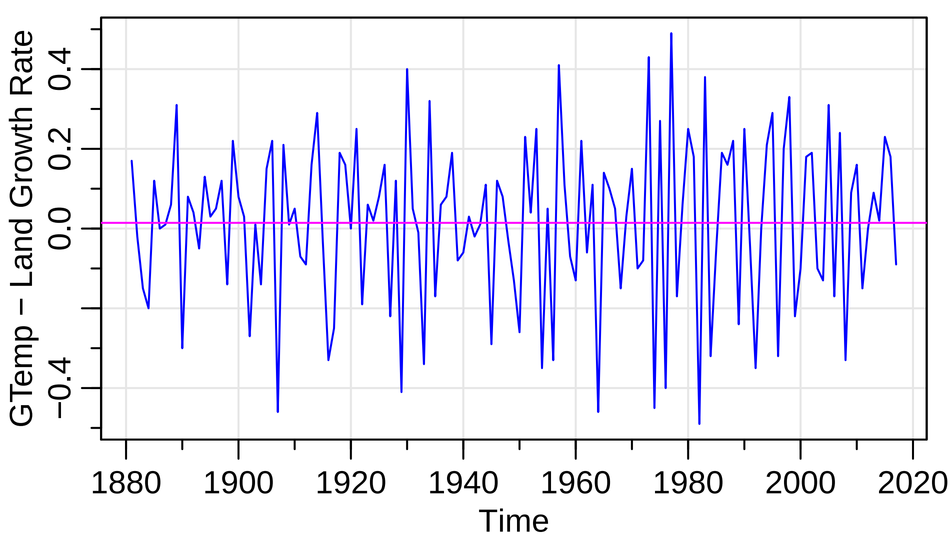
```r
layout(1:2, heights = 2:1)
tsplot(gtemp_land, col = 4)
acf1(gtemp_land, main = "")
```



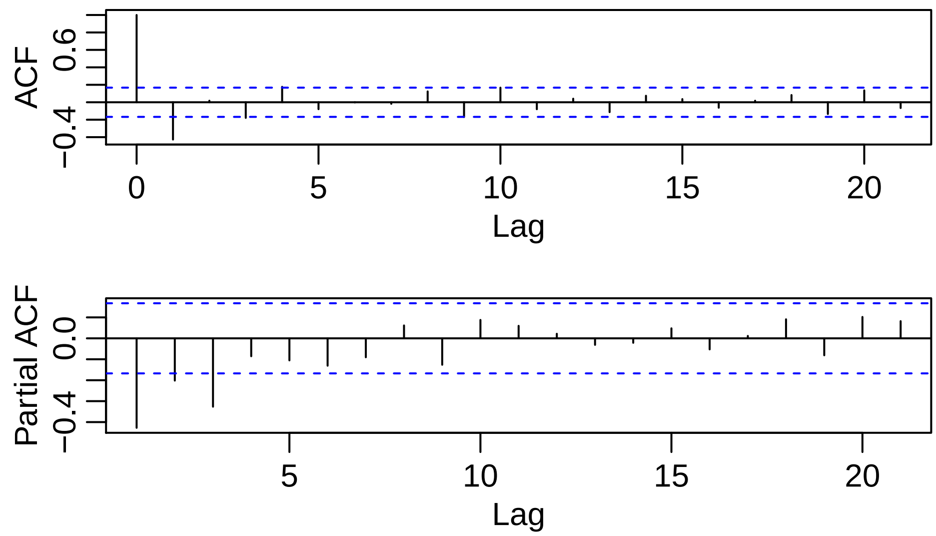



```
##  [1] 0.87 0.82 0.78 0.77 0.73 0.70 0.68 0.65 0.61 0.61 0.55 0.52 0.48 0.47 0.44
## [16] 0.40 0.38 0.37 0.33 0.32 0.29 0.28
```

```r
par(mfrow = c(1, 1))
tsplot(diff(gtemp_land), ylab = "GTemp - Land Growth Rate", col = 4)
mean_dif2 = mean(diff(gtemp_land))
abline(h = mean_dif2, col = 6)
```



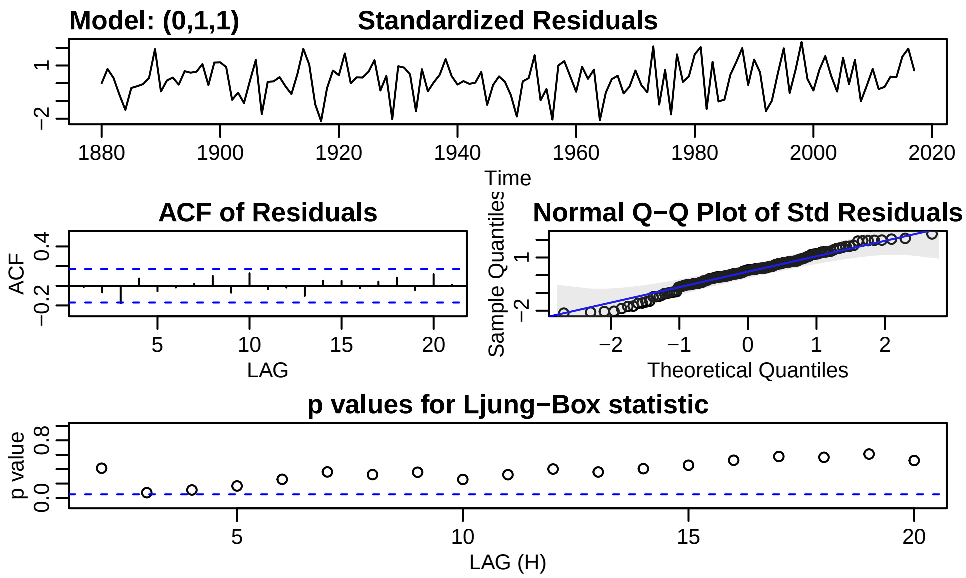

```r
par(mfrow = c(2, 1))
acf(diff(gtemp_land))
pacf(diff(gtemp_land))
```



Repeating the steps mentioned in 5.2, the graph of the gtemp_land data shows a trend prime for differencing. We cannot/need not take the log this time, and we can see from the differenced data that it is much more stable. The ACF cuts off at about lag 1, and the PACF trails off, leading us to fit an MA(1) model on the differenced data, or ARIMA(0, 1, 1).

```
sarima(gtemp_land, 0, 1, 1, no.constant = TRUE)
```

```
## initial  value -1.566843
## iter   2 value -1.694707
## iter   3 value -1.707823
## iter   4 value -1.714248
## iter   5 value -1.719347
## iter   6 value -1.719752
## iter   7 value -1.719812
## iter   8 value -1.719812
## iter   9 value -1.719812
## iter   9 value -1.719812
## iter   9 value -1.719812
## final  value -1.719812
## converged
## initial  value -1.718348
## iter   2 value -1.718358
## iter   3 value -1.718368
## iter   3 value -1.718368
## iter   3 value -1.718368
## final  value -1.718368
## converged
```

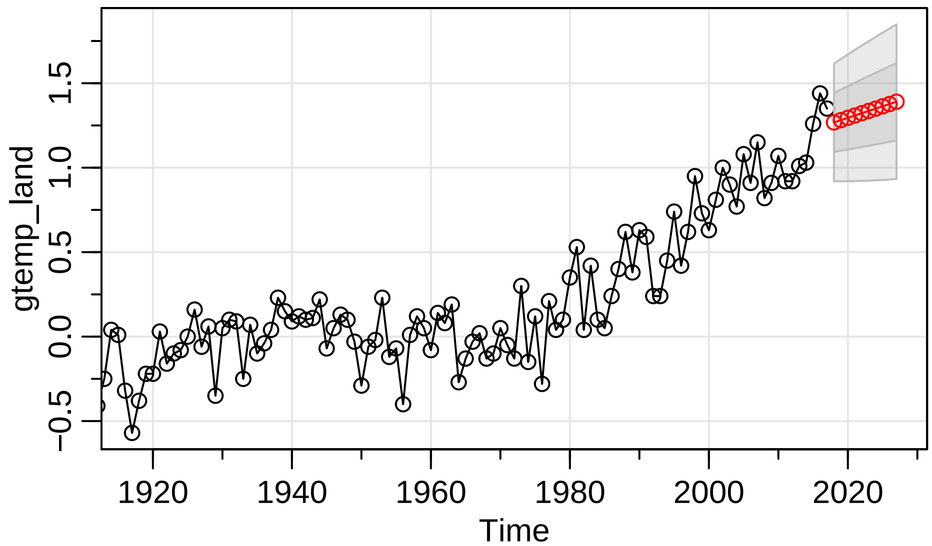



```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1
##       -0.6288
## s.e.   0.0658
##
## sigma^2 estimated as 0.03205:  log likelihood = 41.02,  aic = -78.04
##
## $degrees_of_freedom
## [1] 136
##
## $ttable
##     Estimate     SE t.value p.value
## ma1  -0.6288 0.0658 -9.5603       0
##
## $AIC
## [1] -0.569662
##
## $AICc
## [1] -0.5694457
##
## $BIC
## [1] -0.5270345
```

The model diagnostics show 0-ish residuals (white noise), standard residuals (white noise), a normal q-q plot (minimal divergence from the normal line), and acceptable p-values.

Now we can forecast to the next 10 years.

```r
par(mfrow = c(1, 1))
sarima.for(gtemp_land, 10, 0, 1, 1)
```




```
## $pred
## Time Series:
## Start = 2018
## End = 2027
## Frequency = 1
##  [1] 1.267809 1.281445 1.295081 1.308716 1.322352 1.335987 1.349623 1.363258
##  [9] 1.376894 1.390529
##
## $se
## Time Series:
## Start = 2018
## End = 2027
## Frequency = 1
##  [1] 0.1741466 0.1810522 0.1877038 0.1941277 0.2003458 0.2063765 0.2122360
##  [8] 0.2179380 0.2234945 0.2289163
```

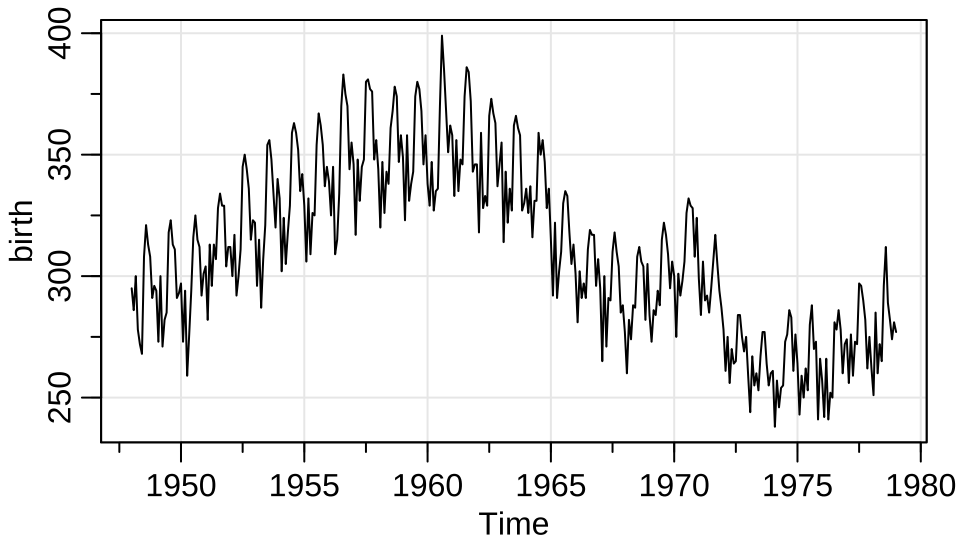
The forecast from the ARIMA(0, 1, 1) model shows a steady upward trend over the next 10 years, consistent with the trajectory of the past 30 years or so.

5.11

Fit a seasonal ARIMA model of your choice to the U.S. Live Birth Series, birth. Use the estimated model to forecast the next 12 months.

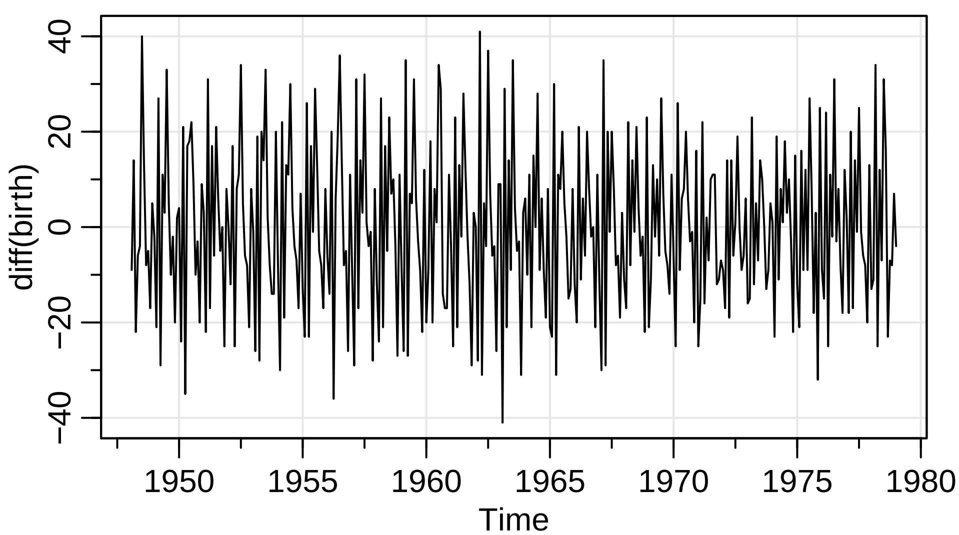
First, plot the data.

```r
tsplot(birth)
```

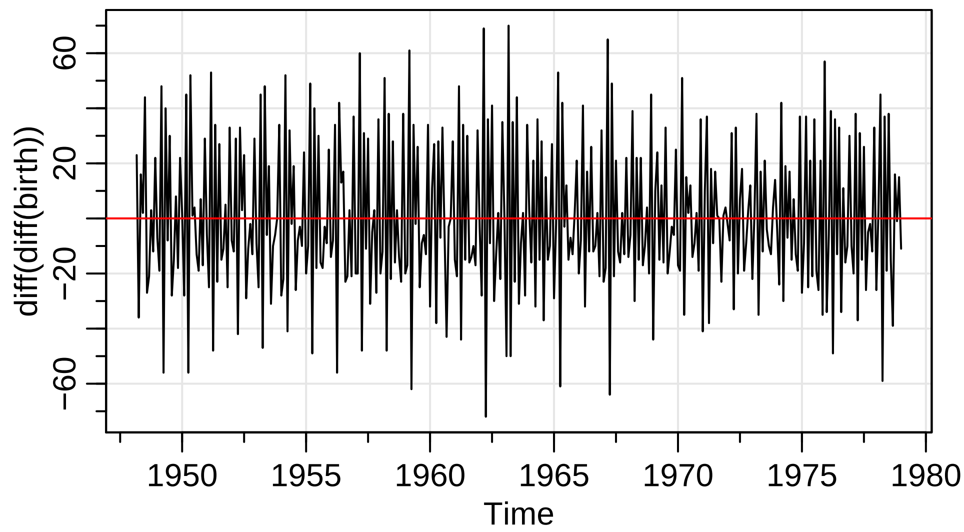


We need to transform the data before fitting a model. The first difference doesn't quite get rid of the trend, but the second one does much better.

```
tsplot(diff(birth))
```

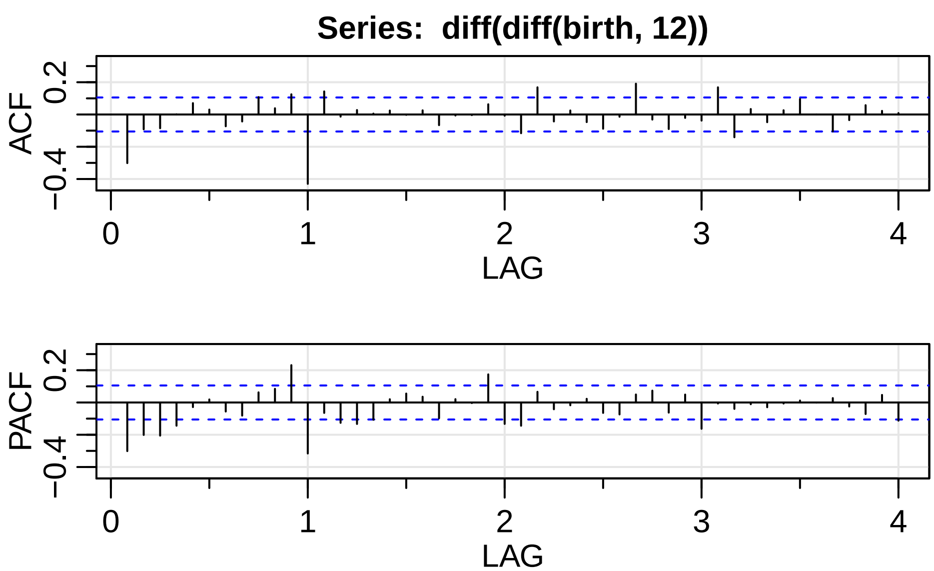



```
tsplot(diff(diff(birth)))
abline(h = mean(diff(diff(birth))), col = "red")
```



Now we can fit the model. First we investigate the ACF, PACF of the differenced data.

```
acf2(diff(diff(birth, 12)))
```

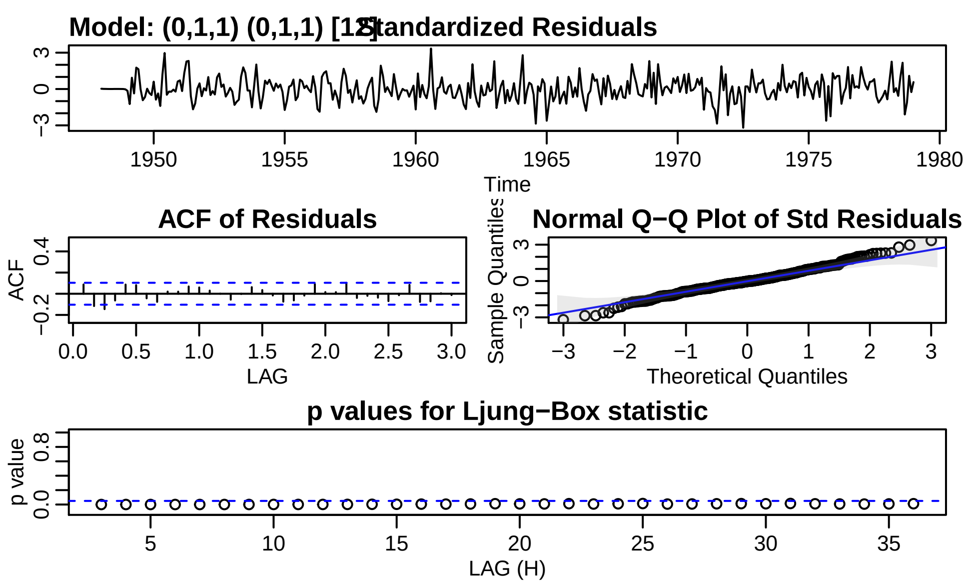



```
##       [,1]  [,2]  [,3]  [,4]  [,5] [,6]  [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  -0.3 -0.09 -0.09  0.00  0.07 0.03 -0.07 -0.04 0.11  0.04  0.13 -0.43  0.14
## PACF -0.3 -0.20 -0.21 -0.14 -0.03 0.02 -0.06 -0.08 0.06  0.08  0.23 -0.32 -0.06
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.01  0.03  0.01  0.02  0.00  0.03 -0.07 -0.01     0  0.06 -0.01 -0.12
## PACF -0.13 -0.13 -0.11  0.02  0.06  0.04 -0.10  0.02     0  0.17 -0.13 -0.14
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.17 -0.04  0.03 -0.05 -0.09 -0.01  0.19 -0.03 -0.09 -0.02 -0.04  0.17
## PACF  0.07 -0.04 -0.02  0.02 -0.06 -0.07  0.05  0.07 -0.06  0.05 -0.16 -0.01
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.14  0.03 -0.05  0.03  0.10     0 -0.10 -0.03  0.06  0.02  0.01
## PACF -0.04 -0.01 -0.03 -0.01  0.01     0  0.03 -0.02 -0.07  0.05 -0.11
```

We can see that the ACF repeats at lags 1, 2, 3, etc. and the PACF tails off, indicating s = 12 and MA(1), for seasonal component. The ACF, without the seasonal component, seems to cut off at lag 1 indicating p = 1 and PACF tails off, which is also an MA(1) for the non-seasonal component. We can model SARIMA(0, 1, 1)x(0, 1, 1) with s = 12.

```r
sarima(birth, 0, 1, 1, 0, 1, 1, 12)
```

```
## initial  value 2.219164
## iter   2 value 2.013310
## iter   3 value 1.988107
## iter   4 value 1.980026
## iter   5 value 1.967594
## iter   6 value 1.965384
## iter   7 value 1.965049
## iter   8 value 1.964993
## iter   9 value 1.964992
## iter   9 value 1.964992
## iter   9 value 1.964992
## final  value 1.964992
## converged
## initial  value 1.951264
## iter   2 value 1.945867
## iter   3 value 1.945729
## iter   4 value 1.945723
## iter   5 value 1.945723
## iter   5 value 1.945723
## iter   5 value 1.945723
## final  value 1.945723
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
```

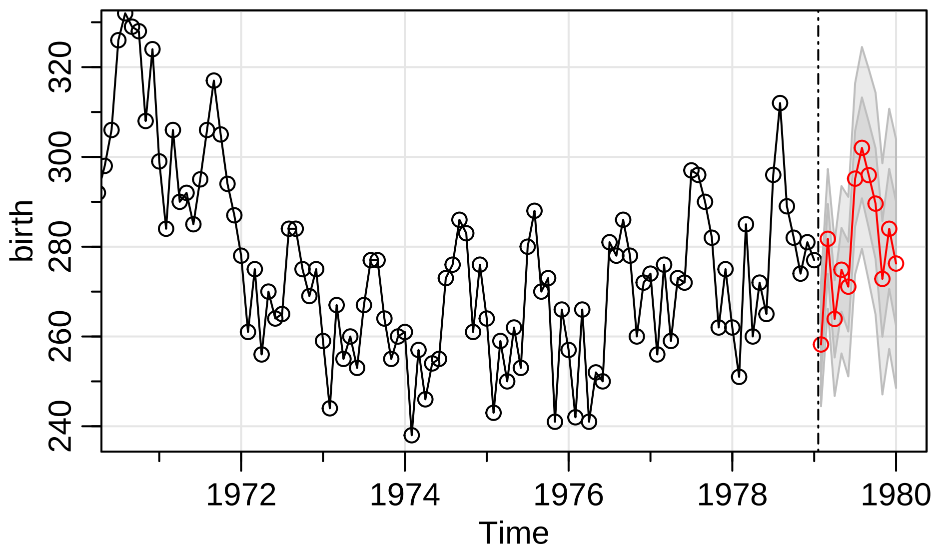
```
##     Q), period = S), include.mean = !no.constant, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.4734  -0.7861
## s.e.   0.0598   0.0451
##
## sigma^2 estimated as 47.4:  log likelihood = -1211.28,  aic = 2428.56
##
## $degrees_of_freedom
## [1] 358
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.4734 0.0598  -7.9097       0
## sma1  -0.7861 0.0451 -17.4227       0
##
## $AIC
## [1] 6.545975
##
## $AICc
## [1] 6.546062
##
## $BIC
## [1] 6.577399
```

After testing a few different SARIMA models to the birth data, I landed on ARIMA(0, 1, 1)X(0, 1, 1)s = 12, where the diagnostics most agree with our requirements for a well-fit model. The forecasted model is below:

```r
sarima.for(birth, 12, 0, 1, 1, 0, 1, 1, 12)
```

```
## $pred
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1979          258.2171 281.7558 263.9016 274.8702 271.1040 295.1588 302.0120
## 1980 276.2490
##           Sep      Oct      Nov      Dec
## 1979 295.9419 289.5935 272.8265 283.9909
## 1980
##
## $se
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 1979            6.884911  7.781346  8.584677  9.319014  9.999569 10.636668
## 1980 13.857229
##            Aug       Sep       Oct       Nov       Dec
## 1979 11.237707 11.808192 12.352358 12.873542 13.374432
## 1980
```

```r
abline(v = 1979.05, lty = 6)
```

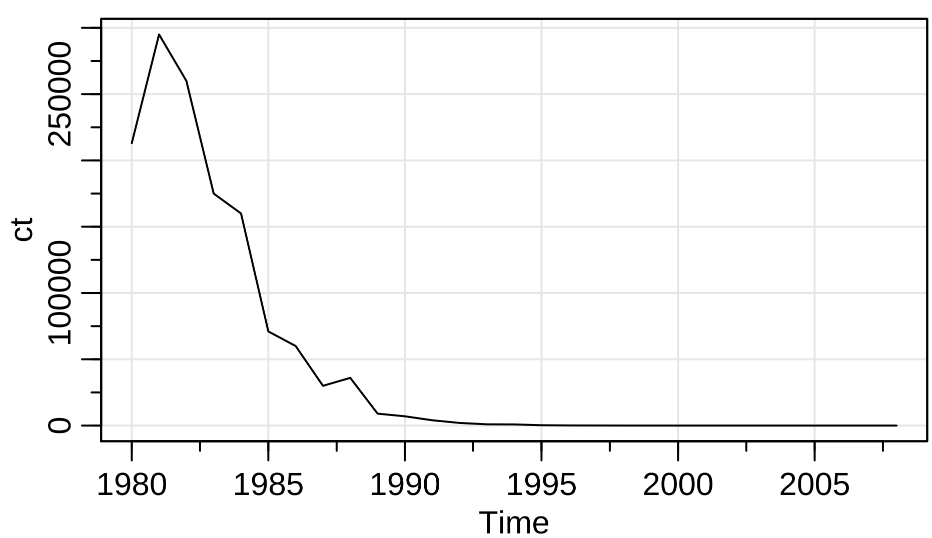


5.14

One of the remarkable technological developments in the computer industry has been the ability to store information densely on a hard drive. In addition, the cost of storage has steadily declined causing problems of too much data as opposed to big data. The data set for this assignment is cpg, which consists of the median annual retail price per GB of hard drives, say ct, taken from a sample of manufacturers from 1980 to 2008.

(a) Plot ct and describe what you see.

```
ct = cpg
tsplot(ct)
```



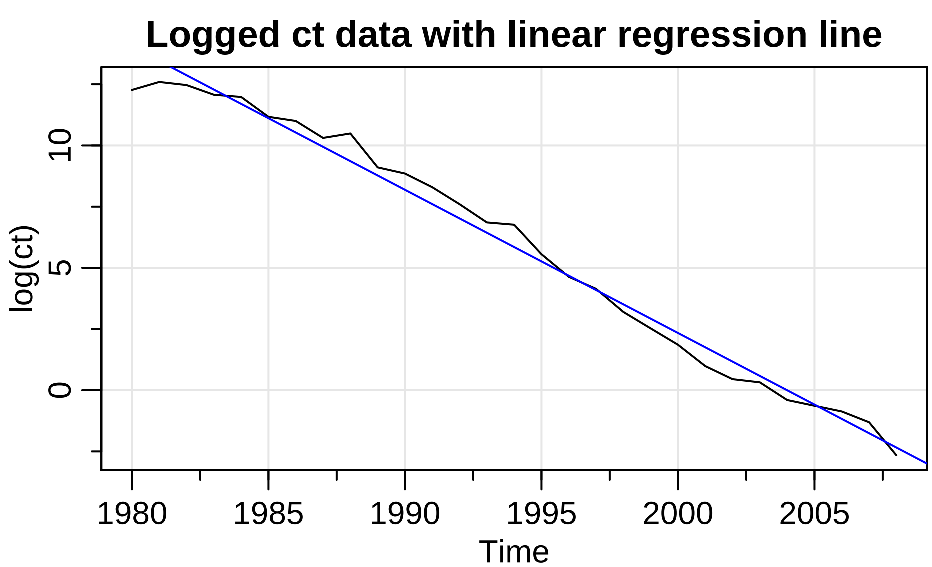
The cpg dataset is high to begin with in the 1980s and sharply decreases until 1990, where it smooths out around 0 for the rest of the time interval.

(b) Argue that the curve ct versus t behaves like ct = ae^bt by fitting a linear regression of log ct on t and then plotting the fitted line to compare it to the logged data. Comment.

```r
t = seq(1980, 2008, 1)
data = cbind(ct, t)
model = lm(log(ct) ~ t)
summary(model)
```

```
##
## Call:
## lm(formula = log(ct) ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.77156 -0.39840  0.04726  0.42186  1.13129
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1172.49431   27.57793   42.52   <2e-16 ***
## t             -0.58508    0.01383  -42.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6231 on 27 degrees of freedom
## Multiple R-squared:  0.9851, Adjusted R-squared:  0.9846
## F-statistic:  1790 on 1 and 27 DF,  p-value: < 2.2e-16
```

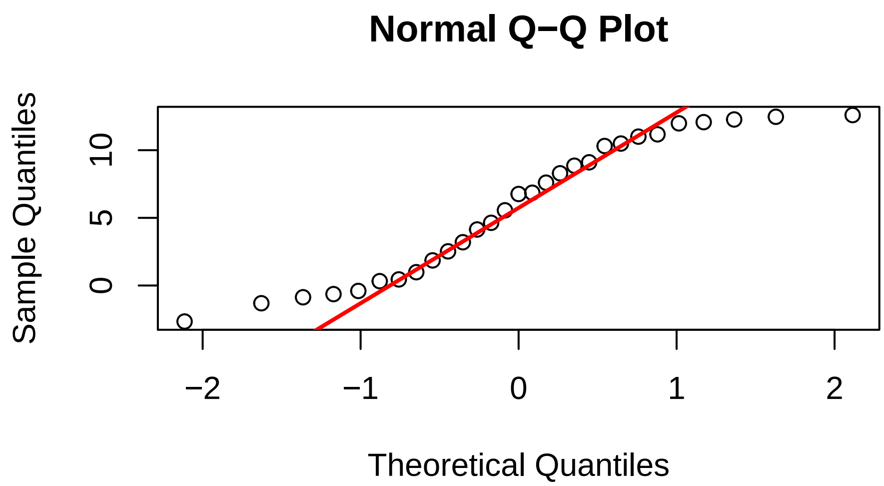
```r
tsplot(log(ct), main = "Logged ct data with linear regression line")
abline(model, col = 4)
```



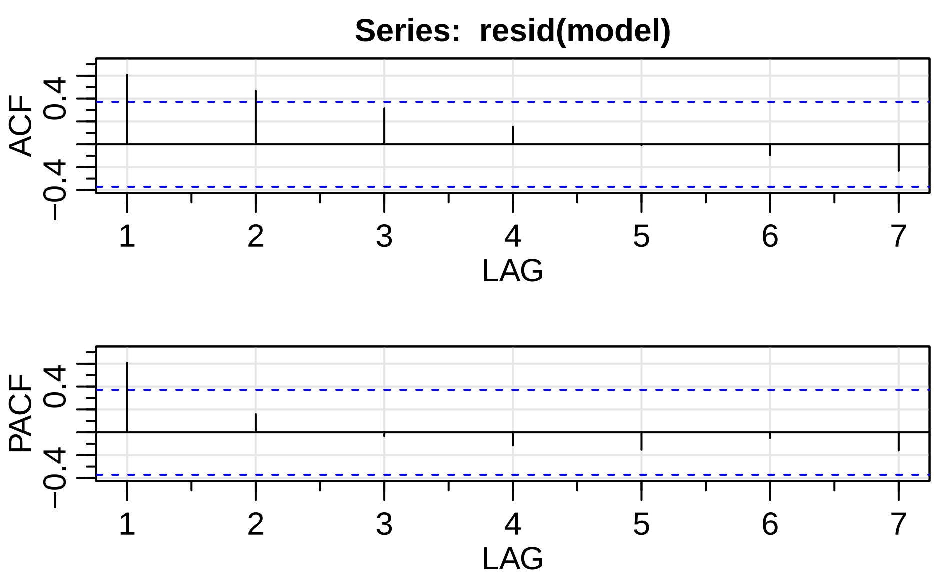
The logged data looks smoother, rather than a sharp decrease it steadily decreases over the time interval of the data. The fitted linear model of logged ct onto t (year) has promising diagnostics for modeling the logged ct data. We now have a model of log(ct) = Bt, or ct = e^Bt.

(c) Inspect the residuals of the linear regression fit and comment.

```r
qqnorm(log(ct), pch = 1)
qqline(log(ct), col = "red", lwd = 2)
```




```r
acf2(resid(model))
```




```
##      [,1] [,2]  [,3]  [,4]  [,5]  [,6]  [,7]
## ACF  0.61 0.47  0.32  0.15 -0.01 -0.10 -0.23
## PACF 0.61 0.16 -0.03 -0.11 -0.15 -0.05 -0.16
```

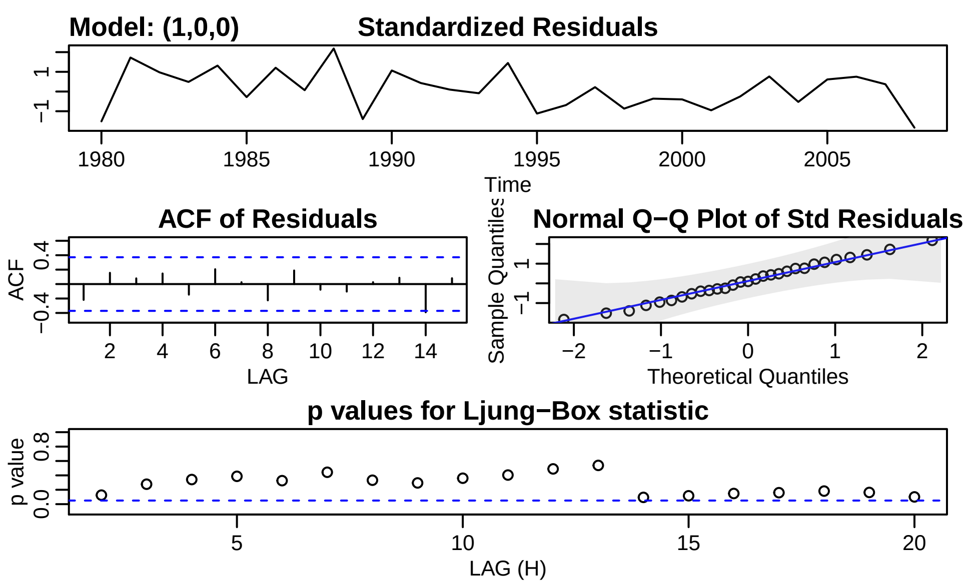
The trend present in the residuals shows that they are likely correlated. A good model would have the residuals mostly flatly on the qqline, but here we see that they trend somewhat cyclically around the line. The ACF and PACF of the residuals also show that they are not white.

(d) Fit the regression again, but now using the fact that the errors are autocorrelated. Comment.

Since the PACF lagged at 1 and the ACF tailed off, we can try an AR(1) model.

```r
sarima(log(ct), 1, 0, 0, xreg = t)
```

```
## initial  value -0.669056
## iter   2 value -0.999488
## iter   3 value -1.088763
## iter   4 value -1.102248
## iter   5 value -1.128914
## iter   6 value -1.131945
## iter   7 value -1.132479
## iter   8 value -1.132525
## iter   9 value -1.132540
## iter  10 value -1.132543
## iter  11 value -1.132545
## iter  12 value -1.132545
## iter  12 value -1.132545
## iter  12 value -1.132545
## final  value -1.132545
## converged
## initial  value -0.701381
## iter   2 value -0.882862
## iter   3 value -0.886699
## iter   4 value -0.888651
## iter   5 value -0.888966
## iter   6 value -0.889035
## iter   7 value -0.889043
## iter   8 value -0.889045
## iter   9 value -0.889045
## iter  10 value -0.889045
## iter  10 value -0.889045
## iter  10 value -0.889045
## final  value -0.889045
## converged
```




```
## $fit
```

```
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1  intercept     xreg
##       0.8297  1113.0105  -0.5554
## s.e.  0.1190    73.5665   0.0368
##
## sigma^2 estimated as 0.1623:  log likelihood = -15.37,  aic = 38.73
##
## $degrees_of_freedom
## [1] 26
##
## $ttable
##            Estimate      SE  t.value p.value
## ar1          0.8297  0.1190   6.9741       0
## intercept 1113.0105 73.5665  15.1293       0
## xreg        -0.5554  0.0368 -15.0716       0
##
## $AIC
## [1] 1.335649
##
## $AICc
## [1] 1.368752
##
## $BIC
## [1] 1.524241
```

Now, the residuals are fairly white and we are happy with the model.