# MATH 4760 Exam 2

*Liam Fruzyna*

---

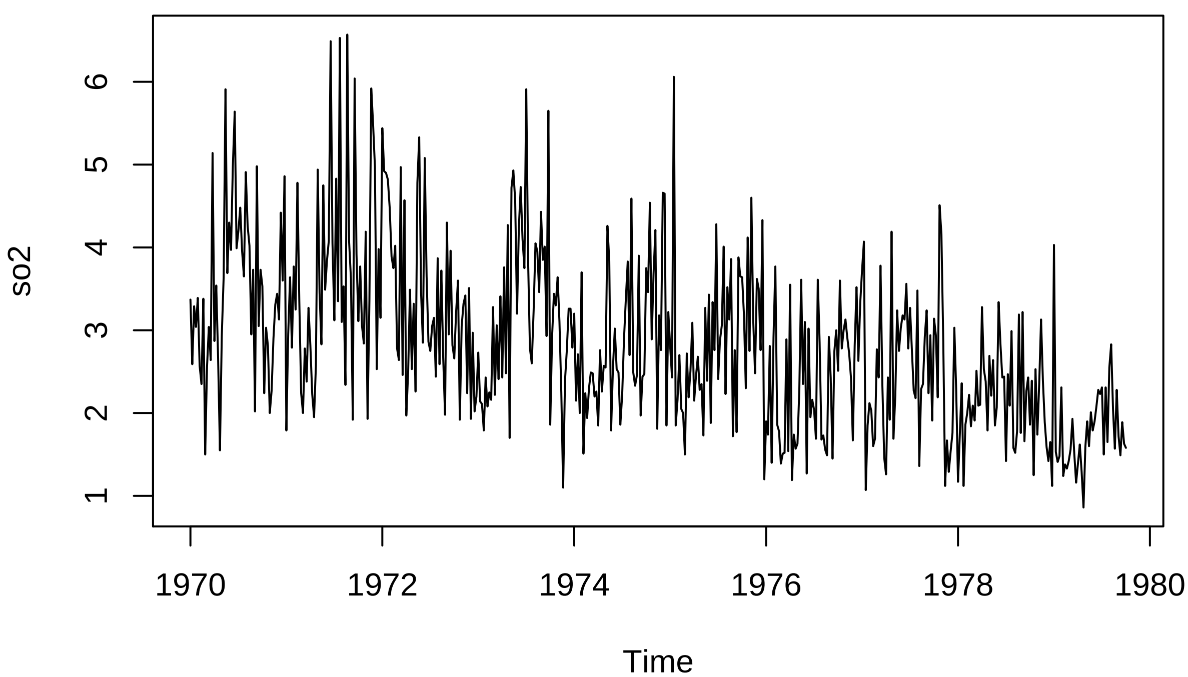
**5) For the following four data sets, your objective is to come up with an appropriate ARIMA model (seasonal or non-seasonal).**

- 1 Sulfer dioxide series, so2
- 2 Crude oil prices, oil
- 3 Global temperature data, gtemp
- 4 Johnson and Johnson earnings, jj

**1) Sulfer dioxide series, so2**

**a) Plot of the data**

```
library(astsa)
plot.ts(so2)
```



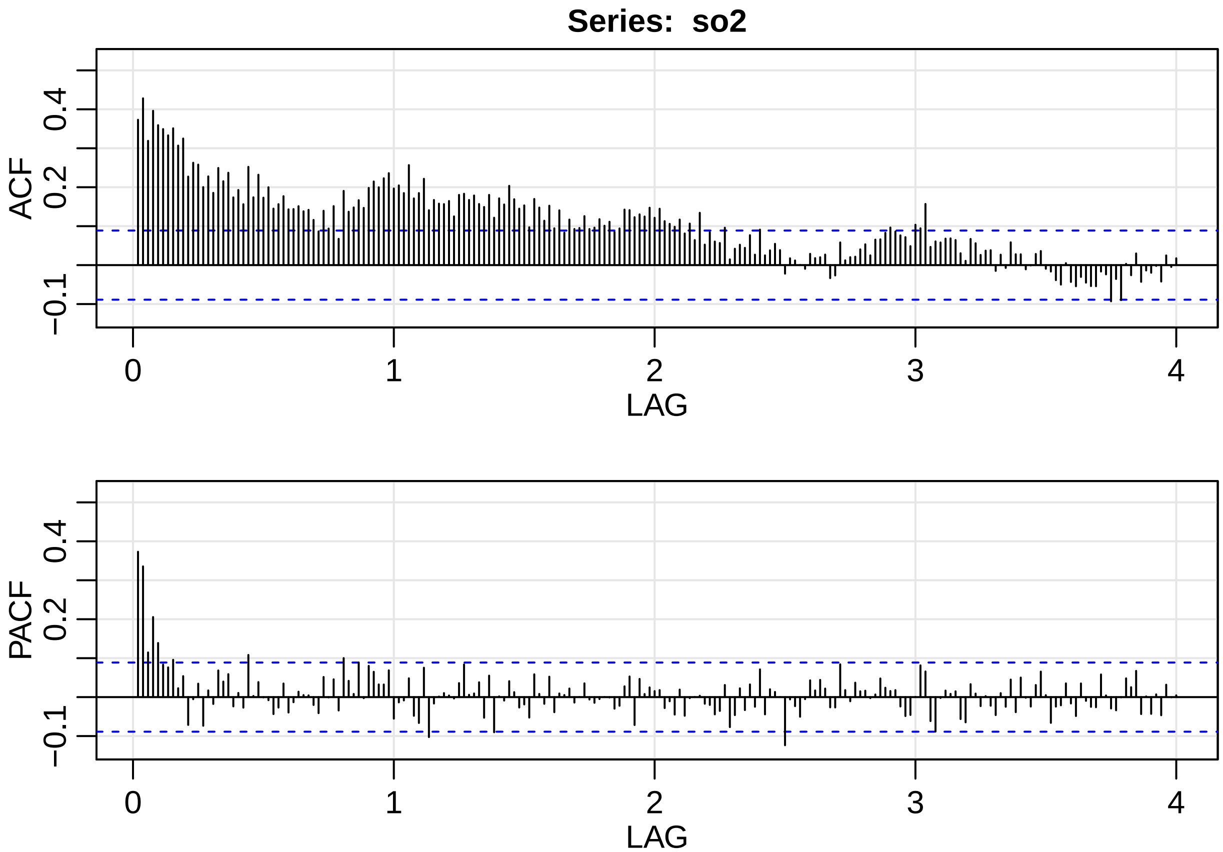

**b) Box-Cox transformation if necessary, and the plot of the transformed data. Note that if a transformation is necessary, the transformed data must be used throughout.**

Transformation is not necessary

**c) Use appropriate techniques (if necessary) to remove trend and seasonal variations. Explain clearly what method(s) was used. Also submit the plot.**

**d) Plot of ACF and PACF. Explain clearly how you use them to determine a range of ARIMA model. Make sure to use differencing if necessary.**

```
acf2(so2)
```




```
##        ACF  PACF
##  [1,] 0.37  0.37
##  [2,] 0.43  0.34
##  [3,] 0.32  0.11
##  [4,] 0.40  0.21
##  [5,] 0.36  0.14
##  [6,] 0.35  0.08
##  [7,] 0.33  0.08
##  [8,] 0.35  0.10
##  [9,] 0.31  0.02
## [10,] 0.33  0.05
## [11,] 0.23 -0.07
## [12,] 0.26 -0.01
## [13,] 0.26  0.03
## [14,] 0.20 -0.07
## [15,] 0.23  0.02
```

```
## [16,] 0.19 -0.02
## [17,] 0.25  0.07
## [18,] 0.22  0.04
## [19,] 0.24  0.06
## [20,] 0.17 -0.02
## [21,] 0.19  0.01
## [22,] 0.16 -0.03
## [23,] 0.25  0.11
## [24,] 0.17  0.00
## [25,] 0.23  0.04
## [26,] 0.17  0.00
## [27,] 0.20 -0.01
## [28,] 0.15 -0.04
## [29,] 0.16 -0.03
## [30,] 0.18  0.03
## [31,] 0.14 -0.04
## [32,] 0.14 -0.01
## [33,] 0.15  0.01
## [34,] 0.14  0.01
## [35,] 0.14  0.00
## [36,] 0.12 -0.02
## [37,] 0.09 -0.04
## [38,] 0.14  0.05
## [39,] 0.09  0.00
## [40,] 0.15  0.05
## [41,] 0.07 -0.03
## [42,] 0.19  0.10
## [43,] 0.14  0.04
## [44,] 0.15  0.01
## [45,] 0.17  0.09
## [46,] 0.15  0.00
## [47,] 0.20  0.08
## [48,] 0.21  0.07
## [49,] 0.20  0.03
## [50,] 0.22  0.03
## [51,] 0.24  0.07
## [52,] 0.20 -0.06
## [53,] 0.20 -0.01
## [54,] 0.18 -0.01
## [55,] 0.26  0.05
## [56,] 0.17 -0.05
## [57,] 0.19 -0.07
## [58,] 0.22  0.08
## [59,] 0.14 -0.10
## [60,] 0.17 -0.02
## [61,] 0.16  0.00
## [62,] 0.16  0.01
## [63,] 0.16  0.00
## [64,] 0.13  0.00
## [65,] 0.18  0.04
## [66,] 0.18  0.08
## [67,] 0.17  0.01
## [68,] 0.18  0.01
## [69,] 0.16  0.04
```

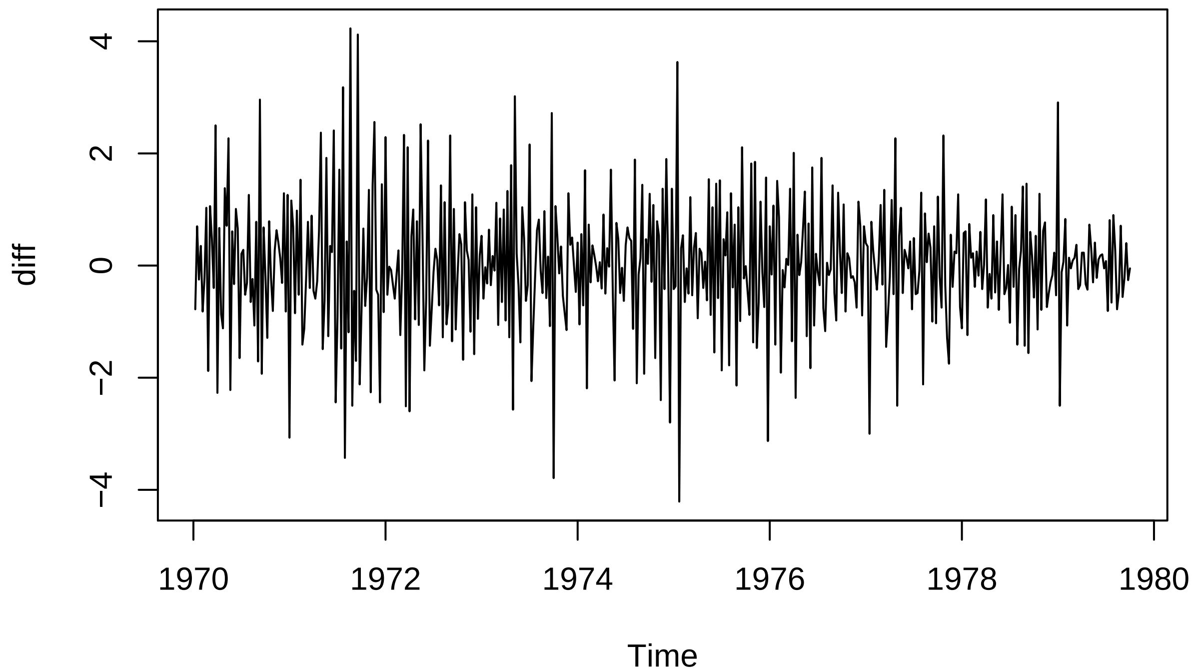
```
##  [70,]  0.15 -0.05
##  [71,]  0.18  0.06
##  [72,]  0.12 -0.09
##  [73,]  0.17  0.00
##  [74,]  0.16 -0.01
##  [75,]  0.20  0.04
##  [76,]  0.17  0.01
##  [77,]  0.14 -0.03
##  [78,]  0.15 -0.02
##  [79,]  0.10 -0.05
##  [80,]  0.17  0.06
##  [81,]  0.15  0.01
##  [82,]  0.11 -0.02
##  [83,]  0.15  0.05
##  [84,]  0.09 -0.04
##  [85,]  0.14  0.01
##  [86,]  0.08  0.01
##  [87,]  0.12  0.02
##  [88,]  0.09 -0.01
##  [89,]  0.10  0.00
##  [90,]  0.13  0.04
##  [91,]  0.09 -0.01
##  [92,]  0.10 -0.01
##  [93,]  0.12  0.00
##  [94,]  0.10  0.00
##  [95,]  0.11  0.00
##  [96,]  0.09 -0.03
##  [97,]  0.09 -0.02
##  [98,]  0.14  0.03
##  [99,]  0.14  0.05
## [100,]  0.12 -0.07
## [101,]  0.13  0.05
## [102,]  0.12  0.01
## [103,]  0.15  0.03
## [104,]  0.12  0.02
## [105,]  0.14  0.02
## [106,]  0.11 -0.03
## [107,]  0.11 -0.01
## [108,]  0.10 -0.04
## [109,]  0.12  0.02
## [110,]  0.08 -0.05
## [111,]  0.11  0.00
## [112,]  0.06  0.00
## [113,]  0.13  0.00
## [114,]  0.05 -0.02
## [115,]  0.08 -0.02
## [116,]  0.06 -0.04
## [117,]  0.06 -0.04
## [118,]  0.10  0.03
## [119,]  0.01 -0.08
## [120,]  0.04 -0.05
## [121,]  0.05  0.02
## [122,]  0.04 -0.03
## [123,]  0.08  0.03
```

```
## [124,]  0.03 -0.02
## [125,]  0.09  0.07
## [126,]  0.03 -0.04
## [127,]  0.04  0.02
## [128,]  0.05  0.01
## [129,]  0.04  0.00
## [130,] -0.02 -0.12
## [131,]  0.02 -0.01
## [132,]  0.01 -0.02
## [133,]  0.00 -0.05
## [134,] -0.01 -0.01
## [135,]  0.03  0.04
## [136,]  0.02  0.02
## [137,]  0.02  0.04
## [138,]  0.03  0.02
## [139,] -0.03 -0.03
## [140,] -0.03 -0.03
## [141,]  0.06  0.08
## [142,]  0.01  0.02
## [143,]  0.02 -0.01
## [144,]  0.02  0.04
## [145,]  0.04  0.01
## [146,]  0.05  0.02
## [147,]  0.03  0.00
## [148,]  0.07  0.01
## [149,]  0.07  0.05
## [150,]  0.08  0.02
## [151,]  0.10  0.02
## [152,]  0.09  0.02
## [153,]  0.08 -0.02
## [154,]  0.07 -0.05
## [155,]  0.05 -0.05
## [156,]  0.10  0.00
## [157,]  0.09  0.08
## [158,]  0.16  0.07
## [159,]  0.05 -0.06
## [160,]  0.06 -0.09
## [161,]  0.06  0.00
## [162,]  0.07  0.02
## [163,]  0.07  0.01
## [164,]  0.06  0.01
## [165,]  0.03 -0.06
## [166,]  0.01 -0.06
## [167,]  0.07  0.03
## [168,]  0.06  0.01
## [169,]  0.03 -0.02
## [170,]  0.04  0.00
## [171,]  0.04 -0.02
## [172,] -0.01 -0.05
## [173,]  0.03  0.01
## [174,] -0.01 -0.02
## [175,]  0.06  0.05
## [176,]  0.03 -0.04
## [177,]  0.03  0.05
```

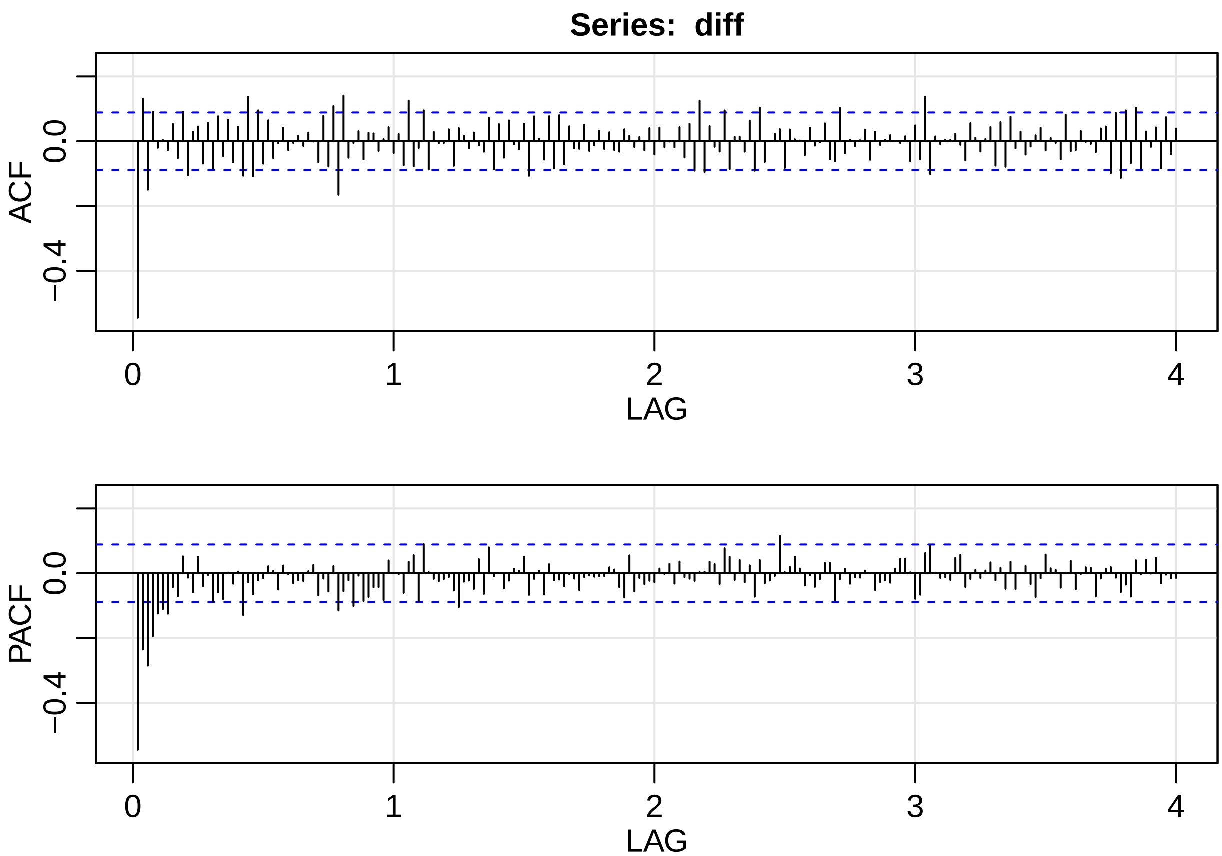
```
## [178,] -0.01  0.00
## [179,]  0.00 -0.02
## [180,]  0.03  0.03
## [181,]  0.04  0.07
## [182,] -0.01  0.01
## [183,] -0.02 -0.07
## [184,] -0.04 -0.02
## [185,] -0.05 -0.02
## [186,]  0.01  0.04
## [187,] -0.04 -0.02
## [188,] -0.05 -0.05
## [189,] -0.03  0.04
## [190,] -0.04 -0.01
## [191,] -0.05 -0.03
## [192,] -0.05 -0.03
## [193,] -0.02  0.06
## [194,] -0.02  0.00
## [195,] -0.09 -0.03
## [196,] -0.04 -0.03
## [197,] -0.09  0.00
## [198,]  0.00  0.05
## [199,] -0.03  0.03
## [200,]  0.03  0.07
## [201,] -0.04 -0.04
## [202,] -0.01  0.00
## [203,] -0.02 -0.04
## [204,]  0.00  0.01
## [205,] -0.04 -0.05
## [206,]  0.02  0.03
## [207,]  0.00  0.00
## [208,]  0.02  0.00
```

Data set is not stationary so differencing is required.

```
diff = diff(so2)
plot.ts(diff)
```




```
acf2(diff)
```




```
##          ACF  PACF
##   [1,] -0.54 -0.54
##   [2,]  0.13 -0.24
##   [3,] -0.15 -0.28
##   [4,]  0.09 -0.19
##   [5,] -0.02 -0.12
##   [6,]  0.00 -0.11
##   [7,] -0.03 -0.12
##   [8,]  0.05 -0.04
##   [9,] -0.05 -0.07
##  [10,]  0.09  0.05
##  [11,] -0.11 -0.01
##  [12,]  0.03 -0.06
##  [13,]  0.05  0.05
##  [14,] -0.07 -0.04
##  [15,]  0.06 -0.01
##  [16,] -0.08 -0.09
##  [17,]  0.08 -0.06
##  [18,] -0.05 -0.08
##  [19,]  0.07  0.00
##  [20,] -0.06 -0.03
##  [21,]  0.04  0.01
##  [22,] -0.11 -0.13
##  [23,]  0.14 -0.03
##  [24,] -0.11 -0.06
##  [25,]  0.10 -0.02
```

```
##  [26,] -0.07 -0.01
##  [27,]  0.06  0.02
##  [28,] -0.05  0.01
##  [29,] -0.01 -0.05
##  [30,]  0.04  0.02
##  [31,] -0.03  0.00
##  [32,] -0.01 -0.03
##  [33,]  0.02 -0.02
##  [34,] -0.01 -0.02
##  [35,]  0.03  0.01
##  [36,]  0.00  0.03
##  [37,] -0.06 -0.07
##  [38,]  0.08 -0.02
##  [39,] -0.08 -0.06
##  [40,]  0.11  0.02
##  [41,] -0.17 -0.11
##  [42,]  0.14 -0.06
##  [43,] -0.05 -0.02
##  [44,] -0.01 -0.10
##  [45,]  0.03 -0.01
##  [46,] -0.06 -0.09
##  [47,]  0.03 -0.07
##  [48,]  0.02 -0.04
##  [49,] -0.03 -0.04
##  [50,]  0.01 -0.08
##  [51,]  0.04  0.04
##  [52,] -0.04  0.00
##  [53,]  0.02  0.00
##  [54,] -0.07 -0.06
##  [55,]  0.13  0.04
##  [56,] -0.08  0.06
##  [57,] -0.02 -0.09
##  [58,]  0.10  0.09
##  [59,] -0.09  0.00
##  [60,]  0.03 -0.02
##  [61,] -0.01 -0.02
##  [62,] -0.01 -0.02
##  [63,]  0.04 -0.01
##  [64,] -0.08 -0.05
##  [65,]  0.04 -0.10
##  [66,]  0.02 -0.03
##  [67,] -0.02 -0.02
##  [68,]  0.03 -0.05
##  [69,] -0.01  0.04
##  [70,] -0.03 -0.06
##  [71,]  0.07  0.08
##  [72,] -0.09 -0.01
##  [73,]  0.05  0.00
##  [74,] -0.05 -0.05
##  [75,]  0.06 -0.02
##  [76,] -0.01  0.01
##  [77,] -0.02  0.01
##  [78,]  0.05  0.05
##  [79,] -0.11 -0.07
```

```
##  [80,]  0.08 -0.02
##  [81,]  0.01  0.01
##  [82,] -0.06 -0.07
##  [83,]  0.08  0.03
##  [84,] -0.08 -0.02
##  [85,]  0.08 -0.02
##  [86,] -0.07 -0.04
##  [87,]  0.05  0.00
##  [88,] -0.02 -0.02
##  [89,] -0.02 -0.05
##  [90,]  0.05 -0.01
##  [91,] -0.03 -0.01
##  [92,] -0.01 -0.01
##  [93,]  0.03 -0.01
##  [94,] -0.02 -0.01
##  [95,]  0.03  0.02
##  [96,] -0.03  0.01
##  [97,] -0.03 -0.04
##  [98,]  0.04 -0.07
##  [99,]  0.02  0.06
## [100,] -0.02 -0.06
## [101,]  0.01 -0.01
## [102,] -0.03 -0.03
## [103,]  0.04 -0.02
## [104,] -0.04 -0.03
## [105,]  0.04  0.01
## [106,] -0.02  0.00
## [107,]  0.00  0.03
## [108,] -0.02 -0.03
## [109,]  0.04  0.04
## [110,] -0.05 -0.01
## [111,]  0.05 -0.02
## [112,] -0.09 -0.02
## [113,]  0.13  0.00
## [114,] -0.09  0.00
## [115,]  0.05  0.04
## [116,] -0.02  0.03
## [117,] -0.03 -0.03
## [118,]  0.10  0.08
## [119,] -0.09  0.05
## [120,]  0.01 -0.02
## [121,]  0.01  0.04
## [122,] -0.03 -0.03
## [123,]  0.06  0.02
## [124,] -0.09 -0.07
## [125,]  0.10  0.04
## [126,] -0.06 -0.03
## [127,]  0.00 -0.02
## [128,]  0.02 -0.01
## [129,]  0.04  0.12
## [130,] -0.08  0.00
## [131,]  0.04  0.02
## [132,]  0.01  0.05
## [133,]  0.00  0.01
```

```
## [134,] -0.04 -0.04
## [135,]  0.04 -0.01
## [136,] -0.01 -0.04
## [137,]  0.00 -0.02
## [138,]  0.06  0.03
## [139,] -0.06  0.03
## [140,] -0.06 -0.08
## [141,]  0.10 -0.02
## [142,] -0.04  0.01
## [143,]  0.01 -0.03
## [144,] -0.02 -0.01
## [145,]  0.00 -0.01
## [146,]  0.04  0.01
## [147,] -0.06  0.00
## [148,]  0.03 -0.05
## [149,] -0.01 -0.03
## [150,]  0.00 -0.02
## [151,]  0.02 -0.03
## [152,]  0.00  0.01
## [153,]  0.00  0.04
## [154,]  0.02  0.05
## [155,] -0.06  0.00
## [156,]  0.05 -0.08
## [157,] -0.06 -0.07
## [158,]  0.14  0.06
## [159,] -0.10  0.09
## [160,]  0.01  0.00
## [161,] -0.01 -0.01
## [162,]  0.00 -0.01
## [163,]  0.00 -0.02
## [164,]  0.02  0.05
## [165,] -0.01  0.06
## [166,] -0.06 -0.04
## [167,]  0.06 -0.02
## [168,]  0.01  0.01
## [169,] -0.03 -0.01
## [170,]  0.01  0.01
## [171,]  0.04  0.03
## [172,] -0.08 -0.02
## [173,]  0.06  0.02
## [174,] -0.08 -0.05
## [175,]  0.08  0.04
## [176,] -0.02 -0.05
## [177,]  0.03  0.00
## [178,] -0.04  0.02
## [179,] -0.02 -0.03
## [180,]  0.02 -0.07
## [181,]  0.04 -0.02
## [182,] -0.03  0.06
## [183,]  0.01  0.02
## [184,] -0.01  0.01
## [185,] -0.06 -0.04
## [186,]  0.08  0.00
## [187,] -0.03  0.04
```

```
## [188,] -0.03 -0.05
## [189,]  0.03  0.00
## [190,]  0.00  0.02
## [191,] -0.01  0.02
## [192,] -0.03 -0.07
## [193,]  0.04 -0.02
## [194,]  0.05  0.01
## [195,] -0.10  0.02
## [196,]  0.09 -0.01
## [197,] -0.11 -0.06
## [198,]  0.10 -0.03
## [199,] -0.07 -0.07
## [200,]  0.10  0.04
## [201,] -0.08  0.00
## [202,]  0.03  0.04
## [203,] -0.02  0.00
## [204,]  0.04  0.05
## [205,] -0.08 -0.03
## [206,]  0.07  0.00
## [207,] -0.04 -0.02
## [208,]  0.04 -0.01
```

The ACF is used to determine q which would be 4 in this case and the PACF is used to determine p which would be 7 in this case.

### e) Using certain criterion, determine an optimal ARMA(p, q) model.

```r
sarima(so2, 7, 1, 4)
```

```
## initial  value 0.165810
## iter   2 value 0.026484
## iter   3 value -0.018376
## iter   4 value -0.067862
## iter   5 value -0.099750
## iter   6 value -0.112857
## iter   7 value -0.119496
## iter   8 value -0.122155
## iter   9 value -0.123672
## iter  10 value -0.124475
## iter  11 value -0.124744
## iter  12 value -0.126163
## iter  13 value -0.126456
## iter  14 value -0.126632
## iter  15 value -0.126978
## iter  16 value -0.127762
## iter  17 value -0.129886
## iter  18 value -0.130833
## iter  19 value -0.132037
## iter  20 value -0.134387
## iter  21 value -0.135563
## iter  22 value -0.136393
## iter  23 value -0.136870
## iter  24 value -0.137664
## iter  25 value -0.140189
## iter  26 value -0.141845
```

```
## iter  27 value -0.142181
## iter  28 value -0.143320
## iter  29 value -0.144449
## iter  30 value -0.147233
## iter  31 value -0.147854
## iter  32 value -0.148878
## iter  33 value -0.149186
## iter  34 value -0.149867
## iter  35 value -0.150160
## iter  36 value -0.150958
## iter  37 value -0.151351
## iter  38 value -0.151783
## iter  39 value -0.151865
## iter  40 value -0.151992
## iter  41 value -0.152187
## iter  42 value -0.152425
## iter  43 value -0.152556
## iter  44 value -0.152837
## iter  45 value -0.153512
## iter  46 value -0.153784
## iter  47 value -0.153867
## iter  48 value -0.154173
## iter  48 value -0.154173
## iter  48 value -0.154173
## final  value -0.154173
## converged
## initial  value -0.143900
## iter   2 value -0.144455
## iter   3 value -0.145422
## iter   4 value -0.145677
## iter   5 value -0.145804
## iter   6 value -0.145811
## iter   7 value -0.145819
## iter   8 value -0.145822
## iter   9 value -0.145830
## iter  10 value -0.145861
## iter  11 value -0.145892
## iter  12 value -0.145916
## iter  13 value -0.145922
## iter  14 value -0.145925
## iter  15 value -0.145928
## iter  16 value -0.145943
## iter  17 value -0.145984
## iter  18 value -0.146075
## iter  19 value -0.146185
## iter  20 value -0.146266
## iter  21 value -0.146318
## iter  22 value -0.146411
## iter  23 value -0.146448
## iter  24 value -0.146540
## iter  25 value -0.146600
## iter  26 value -0.146615
## iter  27 value -0.146617
## iter  28 value -0.146618
```

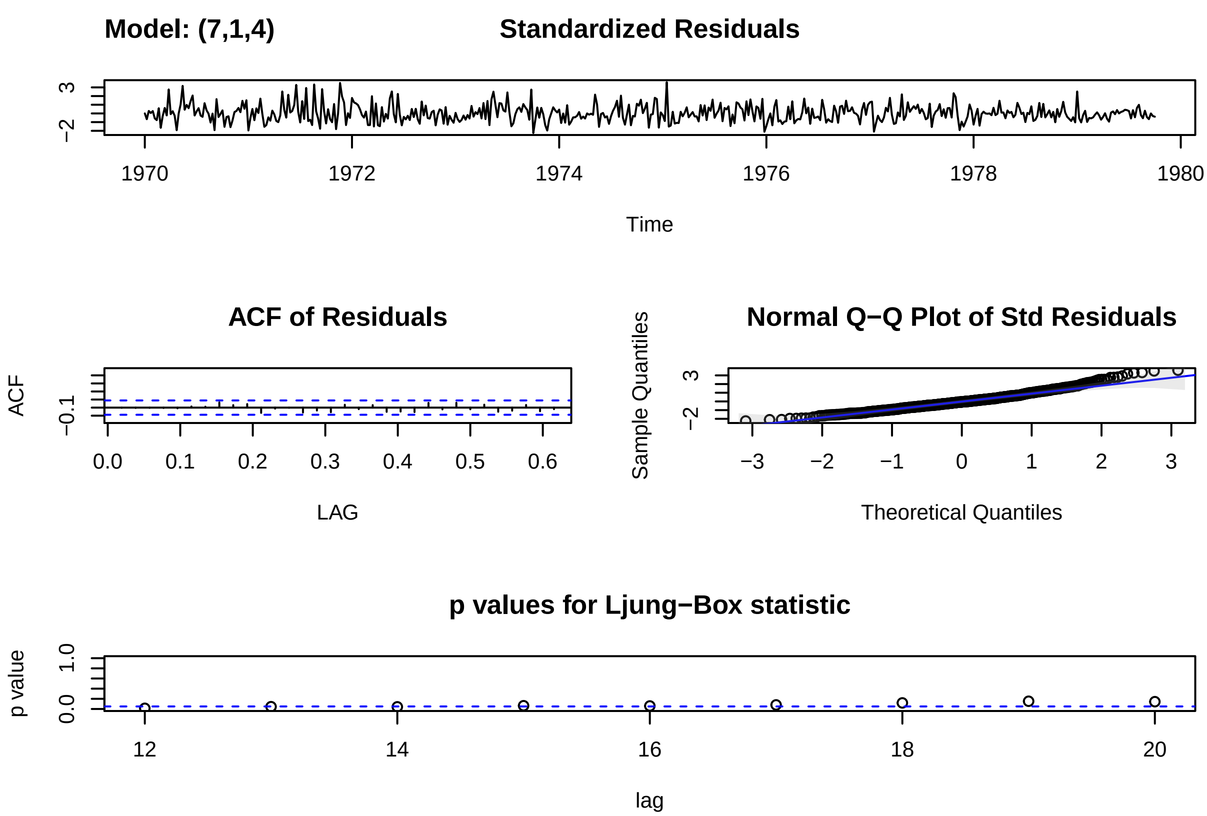
```
## iter  29 value -0.146626
## iter  30 value -0.146631
## iter  31 value -0.146650
## iter  32 value -0.146707
## iter  33 value -0.146920
## iter  34 value -0.147018
## iter  35 value -0.147088
## iter  36 value -0.147103
## iter  37 value -0.147133
## iter  38 value -0.147141
## iter  39 value -0.147181
## iter  40 value -0.147229
## iter  41 value -0.147239
## iter  42 value -0.147447
## iter  43 value -0.147594
```

```
## Warning in log(s2): NaNs produced
```

```
## iter  44 value -0.147779
## iter  45 value -0.147795
## iter  46 value -0.147826
## iter  47 value -0.147844
## iter  48 value -0.147872
## iter  49 value -0.147910
## iter  50 value -0.147957
## iter  51 value -0.148005
## iter  52 value -0.148006
## iter  53 value -0.148006
## iter  54 value -0.148008
## iter  55 value -0.148008
## iter  56 value -0.148026
## iter  57 value -0.148044
## iter  58 value -0.148133
## iter  59 value -0.148159
## iter  60 value -0.148209
## iter  61 value -0.148264
## iter  62 value -0.148288
## iter  63 value -0.148341
## iter  64 value -0.148485
## iter  65 value -0.148529
## iter  66 value -0.148625
## iter  67 value -0.148791
## iter  68 value -0.149253
## iter  69 value -0.149683
## iter  70 value -0.149890
## iter  71 value -0.150086
## iter  72 value -0.150442
## iter  73 value -0.151008
## iter  74 value -0.151277
## iter  75 value -0.151519
## iter  76 value -0.151663
## iter  77 value -0.151668
## iter  78 value -0.151675
## iter  79 value -0.151679
## iter  80 value -0.151680
```

```
## iter  81 value -0.151692
## iter  82 value -0.151711
## iter  83 value -0.151745
## iter  84 value -0.151839
## iter  85 value -0.152010
## iter  86 value -0.152177
## iter  87 value -0.152495
## iter  88 value -0.152588
## iter  89 value -0.152620
## iter  90 value -0.152696
## iter  91 value -0.152738
## iter  92 value -0.152757
## iter  93 value -0.152764
## iter  94 value -0.152765
## iter  95 value -0.152767
## iter  96 value -0.152769
## iter  97 value -0.152772
## iter  98 value -0.152773
## iter  99 value -0.152774
## iter 100 value -0.152774
## final  value -0.152774
## stopped after 100 iterations
```

```
## Warning in stats::arima(xdata, order = c(p, d, q), seasonal = list(order =
## c(P, : possible convergence problem: optim gave code = 1
```




```
## $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     ar2     ar3    ar4     ar5     ar6     ar7      ma1
##       -0.7072  0.0814  0.7159  0.089  0.0937  0.1270  0.0084  -0.1776
## s.e.   0.1724  0.2188  0.1200  0.086  0.0846  0.0771  0.0580   0.1670
##           ma2      ma3     ma4  constant
##       -0.6348  -0.7671  0.5795   -0.0035
## s.e.   0.0678   0.0682  0.1667    0.0006
##
## sigma^2 estimated as 0.7244:  log likelihood = -641.95,  aic = 1309.89
##
## $degrees_of_freedom
## [1] 495
##
## $ttable
##          Estimate     SE  t.value p.value
## ar1       -0.7072 0.1724  -4.1026  0.0000
## ar2        0.0814 0.2188   0.3722  0.7099
## ar3        0.7159 0.1200   5.9637  0.0000
## ar4        0.0890 0.0860   1.0347  0.3013
## ar5        0.0937 0.0846   1.1079  0.2685
## ar6        0.1270 0.0771   1.6460  0.1004
## ar7        0.0084 0.0580   0.1450  0.8847
## ma1       -0.1776 0.1670  -1.0637  0.2880
## ma2       -0.6348 0.0678  -9.3581  0.0000
## ma3       -0.7671 0.0682 -11.2442  0.0000
## ma4        0.5795 0.1667   3.4766  0.0006
## constant  -0.0035 0.0006  -5.8140  0.0000
##
## $AIC
## [1] 0.7248298
##
## $AICc
## [1] 0.7302172
##
## $BIC
## [1] -0.1752376
```

**f) Using hypothesis testing methods, check if certain parameters of the ARMA model can be removed.**

**g) Performed diagnostic check for the model you obtained. Submit the appropriate plots. Make sure to use Box-Ljung statistics to test for white noise. If diagnostic check failed, adjust your model and start all over. Compare all possible models you considered with AIC values and p-values of the Box-Ljung statistics. Determine the final model.**

See e

**h) Write the equation of the final model with clearly indicating the AR and MA coefficients. What is the estimate of the white noise variance? What does it tell you?**

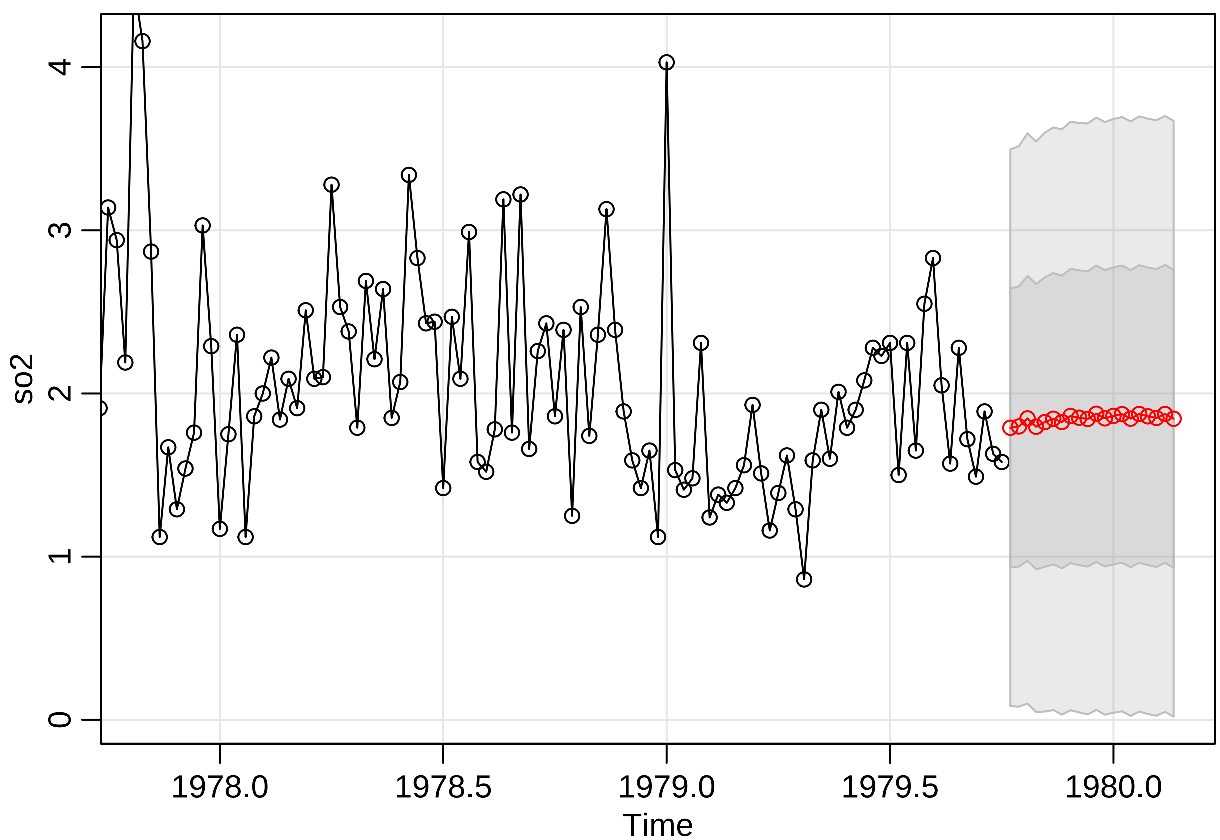
$ X_t+0.707X_{t-1}-0.081X_{t-2}-0.716X_{t-3}-0.089X_{t-4}-0.0937X_{t-5}-0.127X_{t-6}-0.008X_{t-7}=Z_t-0.178Z_{t-1}-0.635Z_{t-2}-0.767Z_{t-3}+0.580X_{t-4} $

**i) Forecast the next 20 values, and submit the plot showing the data with forecast values together with their prediction intervals. State the forecasting values with their standard errors.**

```
sarima.for(so2, 20, 7, 1, 4)
```

```
## Warning in log(s2): NaNs produced
```

```
## Warning in stats::arima(xdata, order = c(p, d, q), seasonal = list(order =
## c(P, : possible convergence problem: optim gave code = 1
```



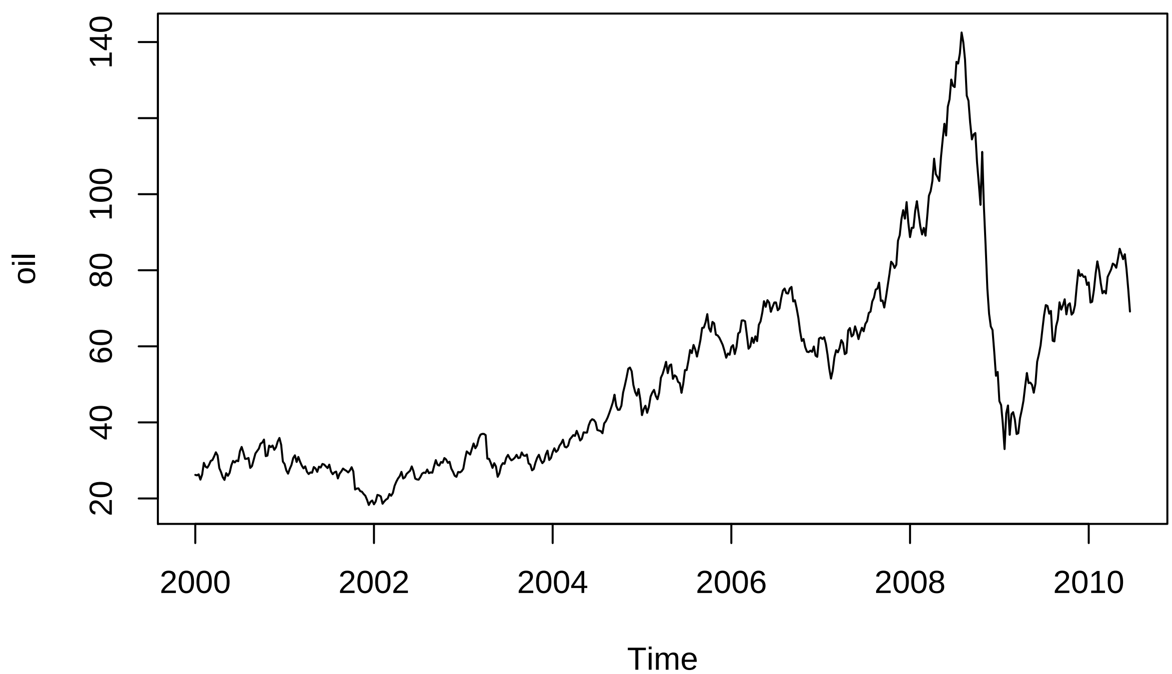

```
## $pred
## Time Series:
## Start = c(1979, 41)
## End = c(1980, 8)
## Frequency = 52
##  [1] 1.789945 1.797850 1.847186 1.796077 1.824172 1.845213 1.825311
##  [8] 1.861968 1.851443 1.843966 1.875890 1.847388 1.862850 1.873197
## [15] 1.845759 1.874403 1.859928 1.849785 1.874449 1.845153
##
## $se
## Time Series:
```

```
## Start = c(1979, 41)
## End = c(1980, 8)
## Frequency = 52
##  [1] 0.8533545 0.8589532 0.8740232 0.8741468 0.8870853 0.8928275 0.8969337
##  [8] 0.9018564 0.9033068 0.9053947 0.9078511 0.9082557 0.9100757 0.9107214
## [15] 0.9109628 0.9122414 0.9122964 0.9126614 0.9132209 0.9132107
```

### 2) Crude oil prices, oil

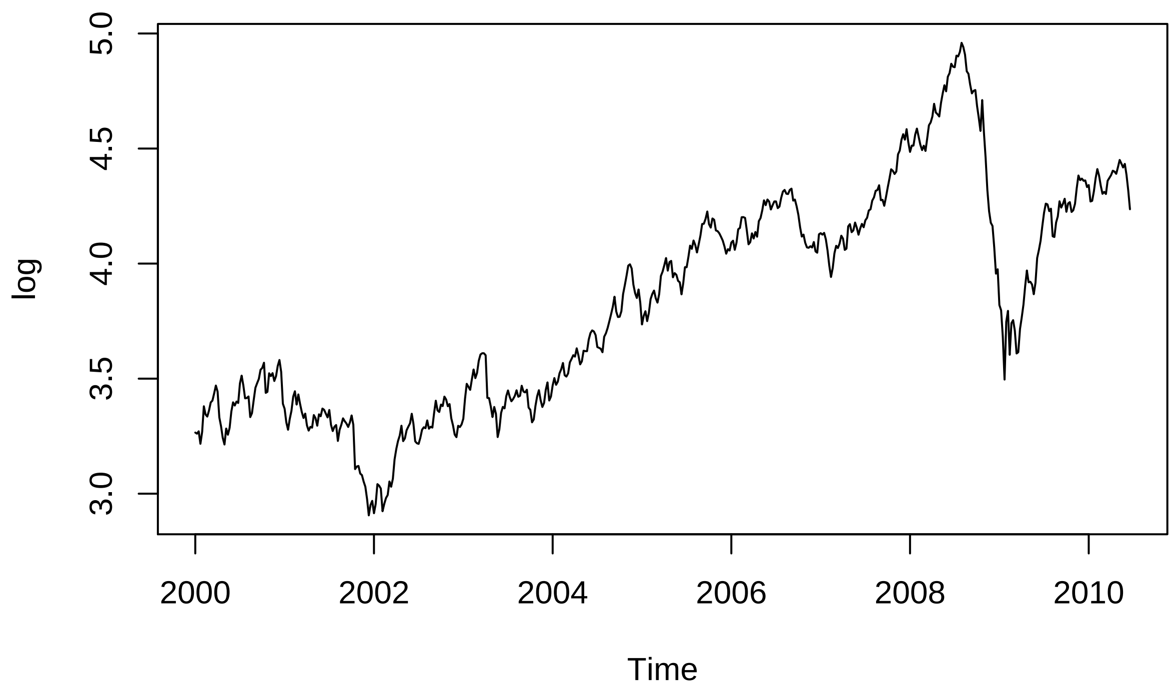
#### a) Plot of the data

```r
plot.ts(oil)
```




#### b) Box-Cox transformation if necessary, and the plot of the transformed data. Note that if a transformation is necessary, the transformed data must be used throughout.

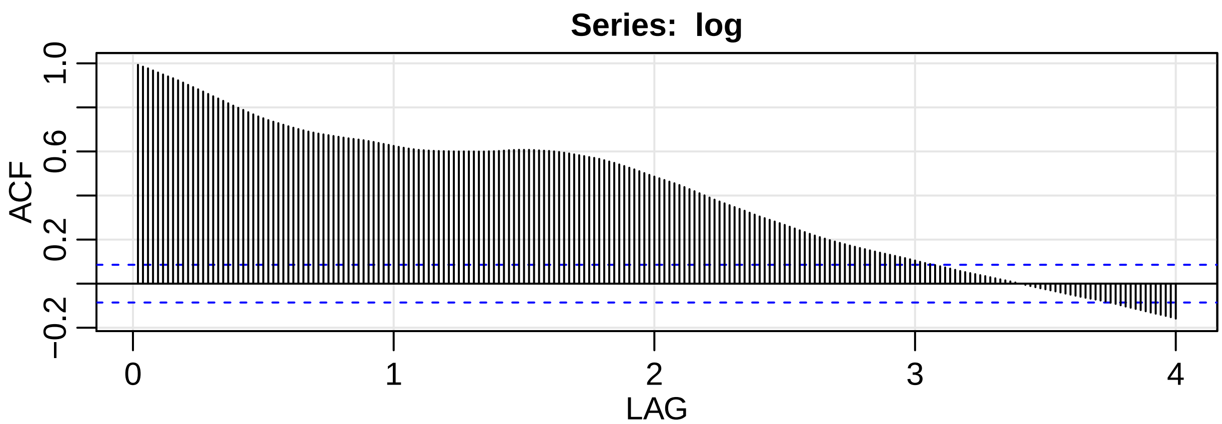
```r
log = log(oil)
plot.ts(log)
```

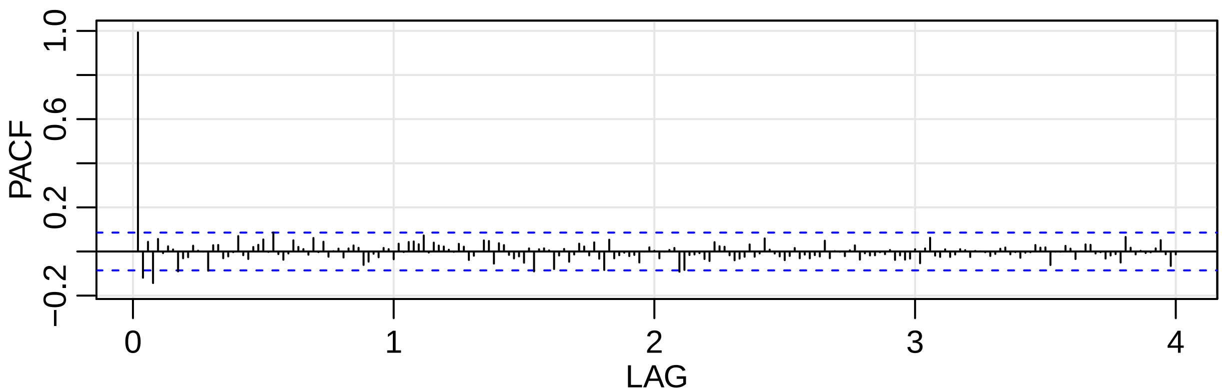



**c) Use appropriate techniques (if necessary) to remove trend and seasonal variations. Explain clearly what method(s) was used. Also submit the plot.**

**d) Plot of ACF and PACF. Explain clearly how you use them to determine a range of ARIMA model. Make sure to use differencing if necessary.**

```
acf2(log)
```




```
##         ACF  PACF
##  [1,]  0.99  0.99
##  [2,]  0.98 -0.12
##  [3,]  0.98  0.04
##  [4,]  0.97 -0.14
##  [5,]  0.96  0.06
##  [6,]  0.95 -0.01
##  [7,]  0.94  0.02
##  [8,]  0.93  0.01
##  [9,]  0.92 -0.09
## [10,]  0.91 -0.03
## [11,]  0.90 -0.03
## [12,]  0.89  0.03
## [13,]  0.88  0.00
## [14,]  0.87  0.00
## [15,]  0.86 -0.09
## [16,]  0.85  0.03
## [17,]  0.84  0.03
## [18,]  0.83 -0.03
## [19,]  0.82 -0.02
## [20,]  0.81  0.00
## [21,]  0.80  0.07
## [22,]  0.79 -0.02
## [23,]  0.78 -0.03
## [24,]  0.77  0.02
## [25,]  0.76  0.03
```

```
## [26,] 0.75  0.05
## [27,] 0.74  0.00
## [28,] 0.74  0.09
## [29,] 0.73 -0.01
## [30,] 0.72 -0.04
## [31,] 0.71 -0.01
## [32,] 0.71  0.05
## [33,] 0.70  0.02
## [34,] 0.70  0.01
## [35,] 0.69 -0.01
## [36,] 0.69  0.06
## [37,] 0.68  0.00
## [38,] 0.68  0.04
## [39,] 0.67 -0.02
## [40,] 0.67  0.00
## [41,] 0.67  0.01
## [42,] 0.66 -0.03
## [43,] 0.66  0.01
## [44,] 0.66  0.03
## [45,] 0.65  0.02
## [46,] 0.65 -0.06
## [47,] 0.65 -0.05
## [48,] 0.64 -0.01
## [49,] 0.64 -0.03
## [50,] 0.63  0.02
## [51,] 0.63  0.01
## [52,] 0.63 -0.04
## [53,] 0.62  0.04
## [54,] 0.62  0.00
## [55,] 0.61  0.04
## [56,] 0.61  0.05
## [57,] 0.61  0.03
## [58,] 0.61  0.07
## [59,] 0.60  0.00
## [60,] 0.60  0.04
## [61,] 0.60  0.03
## [62,] 0.60  0.02
## [63,] 0.60  0.01
## [64,] 0.60  0.00
## [65,] 0.60  0.03
## [66,] 0.60  0.02
## [67,] 0.60 -0.04
## [68,] 0.60 -0.02
## [69,] 0.60  0.00
## [70,] 0.60  0.05
## [71,] 0.60  0.05
## [72,] 0.60 -0.05
## [73,] 0.60  0.04
## [74,] 0.60  0.03
## [75,] 0.61 -0.02
## [76,] 0.61 -0.03
## [77,] 0.61 -0.02
## [78,] 0.61 -0.05
## [79,] 0.61  0.01
```

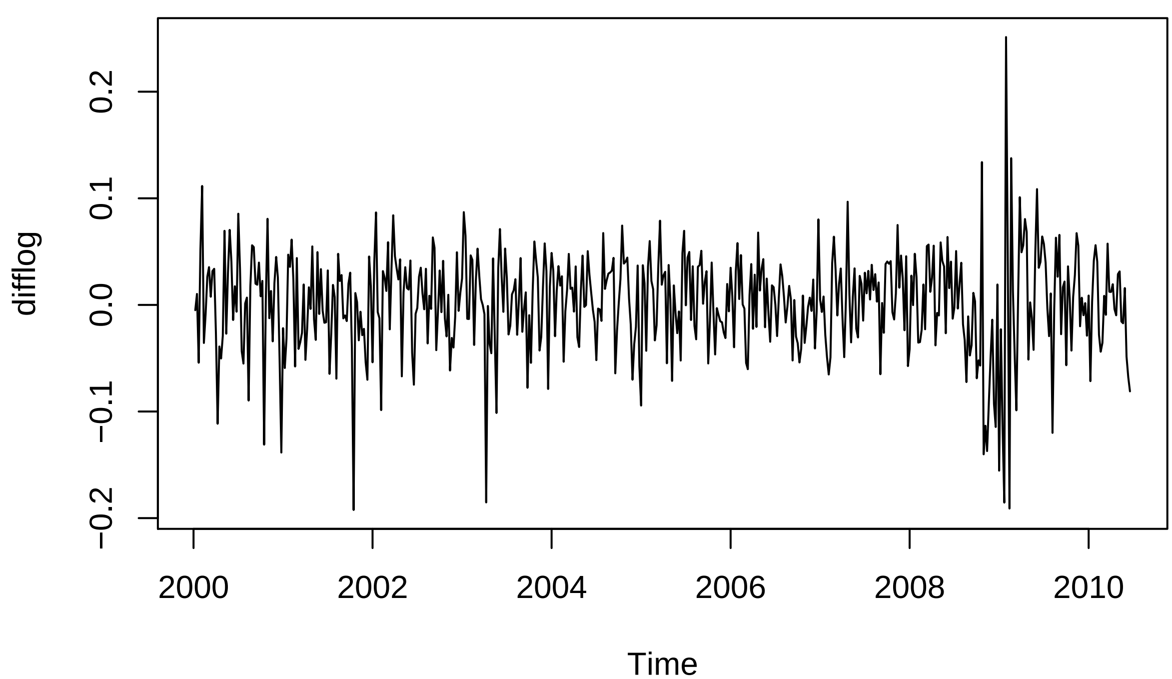
```
##  [80,]  0.61 -0.09
##  [81,]  0.60  0.01
##  [82,]  0.60  0.01
##  [83,]  0.60  0.01
##  [84,]  0.60 -0.08
##  [85,]  0.60 -0.02
##  [86,]  0.59  0.01
##  [87,]  0.59 -0.05
##  [88,]  0.59 -0.01
##  [89,]  0.58  0.04
##  [90,]  0.58  0.02
##  [91,]  0.57 -0.02
##  [92,]  0.57  0.04
##  [93,]  0.57 -0.03
##  [94,]  0.56 -0.08
##  [95,]  0.55  0.05
##  [96,]  0.55 -0.03
##  [97,]  0.54 -0.02
##  [98,]  0.53 -0.01
##  [99,]  0.53 -0.02
## [100,]  0.52 -0.02
## [101,]  0.51 -0.05
## [102,]  0.50  0.00
## [103,]  0.49  0.02
## [104,]  0.49  0.00
## [105,]  0.48 -0.03
## [106,]  0.47  0.00
## [107,]  0.46  0.01
## [108,]  0.46  0.02
## [109,]  0.45 -0.09
## [110,]  0.44 -0.08
## [111,]  0.43 -0.02
## [112,]  0.42 -0.01
## [113,]  0.41 -0.01
## [114,]  0.40 -0.03
## [115,]  0.39 -0.04
## [116,]  0.38  0.04
## [117,]  0.37  0.02
## [118,]  0.36  0.02
## [119,]  0.36 -0.02
## [120,]  0.35 -0.04
## [121,]  0.34 -0.03
## [122,]  0.33 -0.02
## [123,]  0.32  0.03
## [124,]  0.31 -0.02
## [125,]  0.30 -0.01
## [126,]  0.30  0.06
## [127,]  0.29  0.01
## [128,]  0.28 -0.01
## [129,]  0.27 -0.02
## [130,]  0.27 -0.04
## [131,]  0.26 -0.02
## [132,]  0.25  0.02
## [133,]  0.24 -0.03
```

```
## [134,]  0.23 -0.01
## [135,]  0.23 -0.03
## [136,]  0.22 -0.02
## [137,]  0.21 -0.02
## [138,]  0.20  0.05
## [139,]  0.20 -0.03
## [140,]  0.19  0.00
## [141,]  0.19  0.00
## [142,]  0.18 -0.02
## [143,]  0.17  0.01
## [144,]  0.17  0.03
## [145,]  0.16 -0.04
## [146,]  0.16 -0.01
## [147,]  0.15 -0.02
## [148,]  0.15 -0.02
## [149,]  0.14  0.00
## [150,]  0.14 -0.01
## [151,]  0.13  0.01
## [152,]  0.13 -0.04
## [153,]  0.12 -0.02
## [154,]  0.12 -0.04
## [155,]  0.11 -0.03
## [156,]  0.10  0.01
## [157,]  0.10 -0.05
## [158,]  0.09  0.01
## [159,]  0.09  0.06
## [160,]  0.08 -0.02
## [161,]  0.08 -0.02
## [162,]  0.07  0.01
## [163,]  0.07 -0.02
## [164,]  0.06 -0.01
## [165,]  0.06  0.01
## [166,]  0.05  0.01
## [167,]  0.05 -0.02
## [168,]  0.04  0.00
## [169,]  0.04  0.00
## [170,]  0.03  0.00
## [171,]  0.03 -0.02
## [172,]  0.02 -0.01
## [173,]  0.02  0.01
## [174,]  0.01  0.02
## [175,]  0.01 -0.01
## [176,]  0.01  0.00
## [177,]  0.00 -0.03
## [178,] -0.01 -0.01
## [179,] -0.01  0.00
## [180,] -0.02  0.03
## [181,] -0.02  0.02
## [182,] -0.03  0.02
## [183,] -0.03 -0.06
## [184,] -0.04  0.00
## [185,] -0.04  0.00
## [186,] -0.05  0.03
## [187,] -0.05  0.01
```

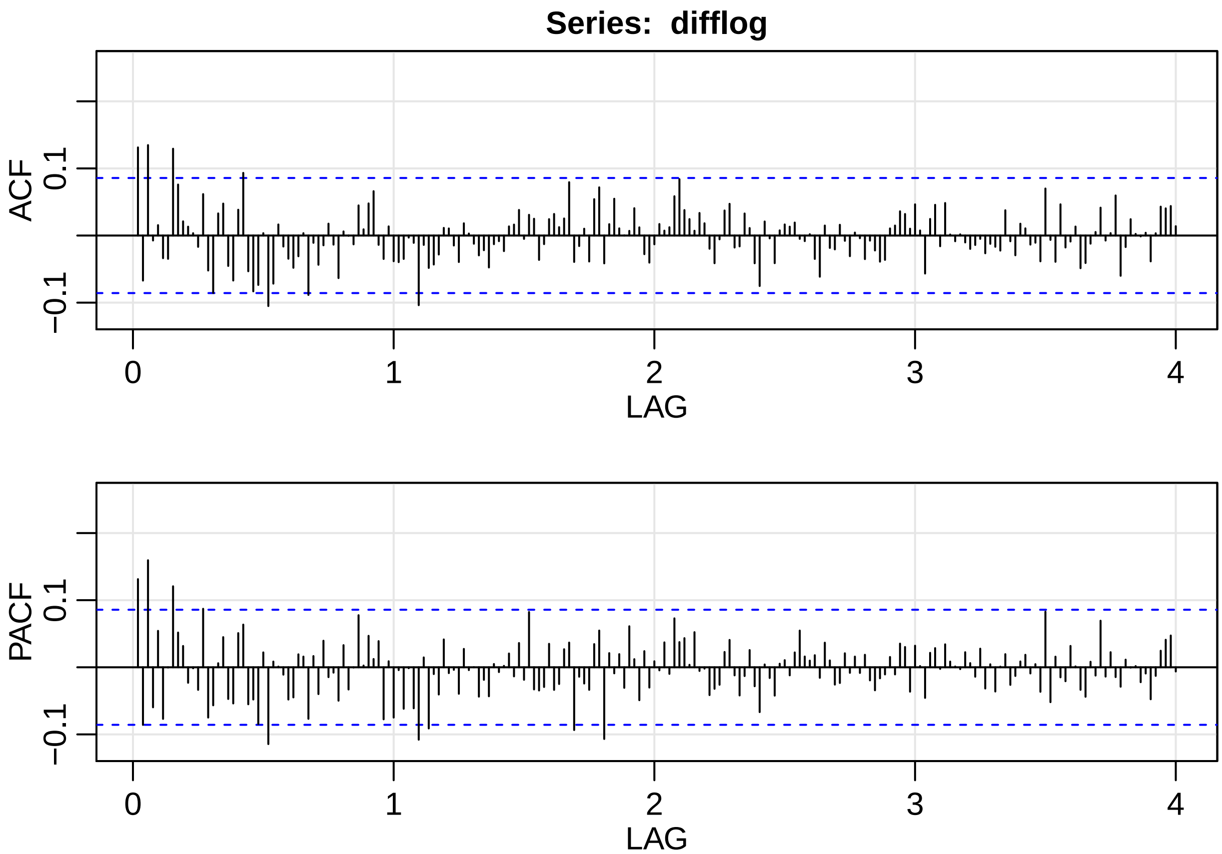
```
## [188,] -0.05 -0.03
## [189,] -0.06  0.00
## [190,] -0.06  0.03
## [191,] -0.07  0.03
## [192,] -0.07 -0.01
## [193,] -0.08  0.00
## [194,] -0.08 -0.03
## [195,] -0.09 -0.02
## [196,] -0.09 -0.01
## [197,] -0.10 -0.05
## [198,] -0.10  0.07
## [199,] -0.11  0.02
## [200,] -0.11 -0.01
## [201,] -0.12  0.00
## [202,] -0.13 -0.01
## [203,] -0.13  0.00
## [204,] -0.14  0.01
## [205,] -0.14  0.05
## [206,] -0.15 -0.01
## [207,] -0.15 -0.07
## [208,] -0.16 -0.01
```

Data set is not stationary so differencing is required.

```r
difflog = diff(log)
plot.ts(difflog)
```



```r
acf2(difflog)
```




```
##          ACF  PACF
##   [1,]  0.13  0.13
##   [2,] -0.07 -0.09
##   [3,]  0.13  0.16
##   [4,] -0.01 -0.06
##   [5,]  0.02  0.05
##   [6,] -0.03 -0.08
##   [7,] -0.03  0.00
##   [8,]  0.13  0.12
##   [9,]  0.08  0.05
##  [10,]  0.02  0.03
##  [11,]  0.01 -0.02
##  [12,]  0.00  0.00
##  [13,] -0.02 -0.03
##  [14,]  0.06  0.09
##  [15,] -0.05 -0.07
##  [16,] -0.09 -0.06
##  [17,]  0.03  0.01
##  [18,]  0.05  0.04
##  [19,] -0.05 -0.05
##  [20,] -0.07 -0.05
##  [21,]  0.04  0.05
##  [22,]  0.09  0.06
##  [23,] -0.05 -0.06
```

```
## [24,] -0.08 -0.05
## [25,] -0.07 -0.08
## [26,]  0.00  0.02
## [27,] -0.11 -0.11
## [28,] -0.07  0.01
## [29,]  0.02  0.00
## [30,] -0.02 -0.01
## [31,] -0.03 -0.05
## [32,] -0.05 -0.04
## [33,] -0.03  0.02
## [34,]  0.00  0.02
## [35,] -0.09 -0.08
## [36,] -0.01  0.02
## [37,] -0.04 -0.04
## [38,] -0.01  0.04
## [39,]  0.02 -0.01
## [40,] -0.01 -0.01
## [41,] -0.06 -0.05
## [42,]  0.01  0.03
## [43,]  0.00 -0.03
## [44,] -0.01  0.00
## [45,]  0.04  0.08
## [46,]  0.01  0.00
## [47,]  0.05  0.05
## [48,]  0.07  0.01
## [49,] -0.01  0.04
## [50,] -0.03 -0.08
## [51,]  0.01  0.01
## [52,] -0.04 -0.07
## [53,] -0.04  0.00
## [54,] -0.03 -0.06
## [55,]  0.00  0.00
## [56,] -0.01 -0.06
## [57,] -0.10 -0.11
## [58,] -0.01  0.01
## [59,] -0.05 -0.09
## [60,] -0.04 -0.01
## [61,] -0.03 -0.04
## [62,]  0.01  0.04
## [63,]  0.01 -0.01
## [64,] -0.01  0.00
## [65,] -0.04 -0.04
## [66,]  0.02  0.03
## [67,]  0.00  0.00
## [68,] -0.01  0.00
## [69,] -0.03 -0.04
## [70,] -0.02 -0.02
## [71,] -0.05 -0.04
## [72,] -0.01  0.00
## [73,] -0.01 -0.01
## [74,] -0.02  0.00
## [75,]  0.01  0.02
## [76,]  0.02 -0.01
## [77,]  0.04  0.04
```

```
##  [78,] -0.01 -0.02
##  [79,]  0.03  0.08
##  [80,]  0.02 -0.03
##  [81,] -0.04 -0.03
##  [82,] -0.01 -0.03
##  [83,]  0.02  0.03
##  [84,]  0.03 -0.03
##  [85,]  0.01 -0.02
##  [86,]  0.03  0.03
##  [87,]  0.08  0.04
##  [88,] -0.04 -0.09
##  [89,] -0.02 -0.01
##  [90,]  0.01 -0.02
##  [91,] -0.04 -0.03
##  [92,]  0.05  0.03
##  [93,]  0.07  0.05
##  [94,] -0.04 -0.11
##  [95,]  0.02  0.02
##  [96,]  0.05 -0.01
##  [97,]  0.01  0.02
##  [98,]  0.00 -0.03
##  [99,]  0.01  0.06
## [100,]  0.04  0.01
## [101,]  0.01 -0.05
## [102,] -0.03  0.02
## [103,] -0.04 -0.03
## [104,] -0.01  0.01
## [105,]  0.02  0.00
## [106,]  0.01  0.04
## [107,]  0.01 -0.01
## [108,]  0.06  0.07
## [109,]  0.08  0.04
## [110,]  0.04  0.04
## [111,]  0.02  0.00
## [112,]  0.01  0.05
## [113,]  0.03 -0.01
## [114,]  0.02  0.00
## [115,] -0.02 -0.04
## [116,] -0.04 -0.03
## [117,] -0.01 -0.03
## [118,]  0.04  0.02
## [119,]  0.05  0.04
## [120,] -0.02 -0.01
## [121,] -0.02 -0.04
## [122,]  0.03 -0.01
## [123,]  0.01  0.03
## [124,] -0.04 -0.03
## [125,] -0.08 -0.07
## [126,]  0.02  0.00
## [127,]  0.00 -0.02
## [128,] -0.04 -0.04
## [129,]  0.01  0.01
## [130,]  0.02  0.01
## [131,]  0.01 -0.01
```

```
## [132,]  0.02  0.02
## [133,]  0.00  0.05
## [134,] -0.01  0.02
## [135,]  0.00  0.01
## [136,] -0.03  0.02
## [137,] -0.06 -0.02
## [138,]  0.01  0.04
## [139,] -0.02  0.01
## [140,] -0.02 -0.03
## [141,]  0.02 -0.02
## [142,] -0.01  0.02
## [143,] -0.03 -0.01
## [144,]  0.00  0.02
## [145,]  0.00 -0.01
## [146,] -0.04  0.02
## [147,] -0.01 -0.02
## [148,] -0.02 -0.03
## [149,] -0.04 -0.02
## [150,] -0.04 -0.01
## [151,]  0.01  0.02
## [152,]  0.01 -0.01
## [153,]  0.04  0.04
## [154,]  0.03  0.03
## [155,]  0.01 -0.04
## [156,]  0.05  0.03
## [157,]  0.01  0.00
## [158,] -0.06 -0.05
## [159,]  0.02  0.02
## [160,]  0.05  0.03
## [161,] -0.02  0.00
## [162,]  0.05  0.03
## [163,]  0.00  0.01
## [164,] -0.01  0.00
## [165,]  0.00  0.00
## [166,] -0.01  0.02
## [167,] -0.02  0.01
## [168,] -0.01 -0.01
## [169,]  0.00  0.03
## [170,] -0.03 -0.03
## [171,] -0.01  0.00
## [172,] -0.02 -0.04
## [173,] -0.02  0.00
## [174,]  0.04  0.02
## [175,] -0.01 -0.03
## [176,] -0.03 -0.01
## [177,]  0.02  0.01
## [178,]  0.01  0.02
## [179,] -0.01 -0.01
## [180,] -0.01  0.00
## [181,] -0.04 -0.04
## [182,]  0.07  0.08
## [183,] -0.01 -0.05
## [184,] -0.04  0.02
## [185,]  0.05 -0.01
```

```
## [186,] -0.02 -0.02
## [187,] -0.01  0.03
## [188,]  0.01  0.00
## [189,] -0.05 -0.03
## [190,] -0.04 -0.04
## [191,] -0.01  0.01
## [192,]  0.01 -0.01
## [193,]  0.04  0.07
## [194,] -0.01 -0.01
## [195,]  0.00  0.02
## [196,]  0.06 -0.01
## [197,] -0.06 -0.03
## [198,] -0.02  0.01
## [199,]  0.02  0.00
## [200,]  0.00  0.00
## [201,]  0.00 -0.02
## [202,]  0.00 -0.01
## [203,] -0.04 -0.05
## [204,]  0.00 -0.01
## [205,]  0.04  0.02
## [206,]  0.04  0.04
## [207,]  0.04  0.05
## [208,]  0.01 -0.01
```

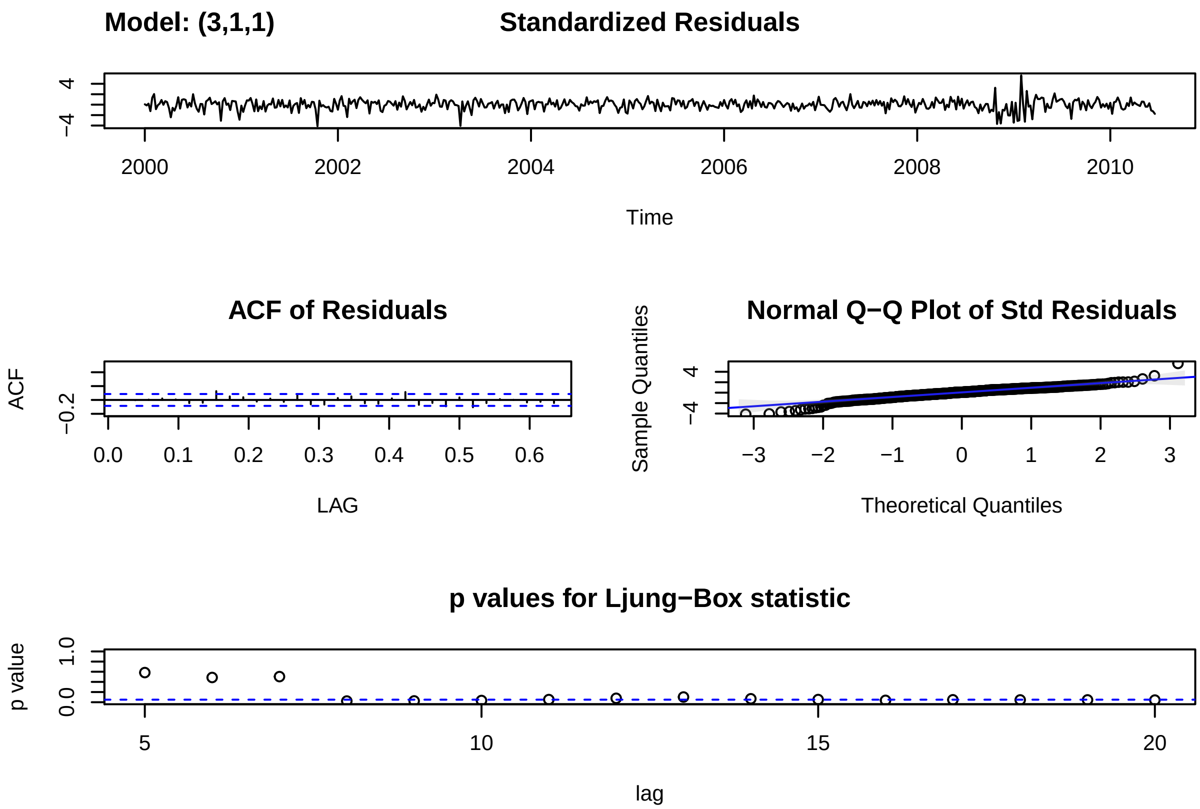
The ACF is used to determine q which would be 1 in this case and the PACF is used to determine p which would be 3 in this case.

### e) Using certain criterion, determine an optimal ARMA(p, q) model.

```r
sarima(log, 3, 1, 1)
```

```
## initial  value -3.057088
## iter   2 value -3.078020
## iter   3 value -3.083582
## iter   4 value -3.083857
## iter   5 value -3.083933
## iter   6 value -3.084908
## iter   7 value -3.085721
## iter   8 value -3.086118
## iter   9 value -3.086275
## iter  10 value -3.086280
## iter  11 value -3.086281
## iter  12 value -3.086282
## iter  13 value -3.086283
## iter  14 value -3.086283
## iter  15 value -3.086283
## iter  16 value -3.086283
## iter  16 value -3.086283
## iter  16 value -3.086283
## final  value -3.086283
## converged
## initial  value -3.087475
## iter   2 value -3.087482
## iter   3 value -3.087517
## iter   4 value -3.087582
```

```
## iter   5 value -3.087597
## iter   6 value -3.087600
## iter   7 value -3.087602
## iter   8 value -3.087604
## iter   9 value -3.087604
## iter  10 value -3.087605
## iter  10 value -3.087605
## iter  10 value -3.087605
## final  value -3.087605
## converged
```




```
## $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      ar2     ar3     ma1  constant
##       -0.3219  -0.0390  0.1180  0.4954    0.0017
## s.e.   0.1552   0.0506  0.0511  0.1529    0.0024
##
## sigma^2 estimated as 0.00208:  log likelihood = 907.75,  aic = -1803.51
##
## $degrees_of_freedom
```

```
## [1] 539
##
## $ttable
##          Estimate     SE t.value p.value
## ar1       -0.3219 0.1552 -2.0746  0.0385
## ar2       -0.0390 0.0506 -0.7720  0.4404
## ar3        0.1180 0.0511  2.3082  0.0214
## ma1        0.4954 0.1529  3.2406  0.0013
## constant   0.0017 0.0024  0.7327  0.4641
##
## $AIC
## [1] -5.157125
##
## $AICc
## [1] -5.153168
##
## $BIC
## [1] -6.117668
```

**f) Using hypothesis testing methods, check if certain parameters of the ARMA model can be removed.**

**g) Performed diagnostic check for the model you obtained. Submit the appropriate plots. Make sure to use Box-Ljung statistics to test for white noise. If diagnostic check failed, adjust your model and start all over. Compare all possible models you considered with AIC values and p-values of the Box-Ljung statistics. Determine the final model.**

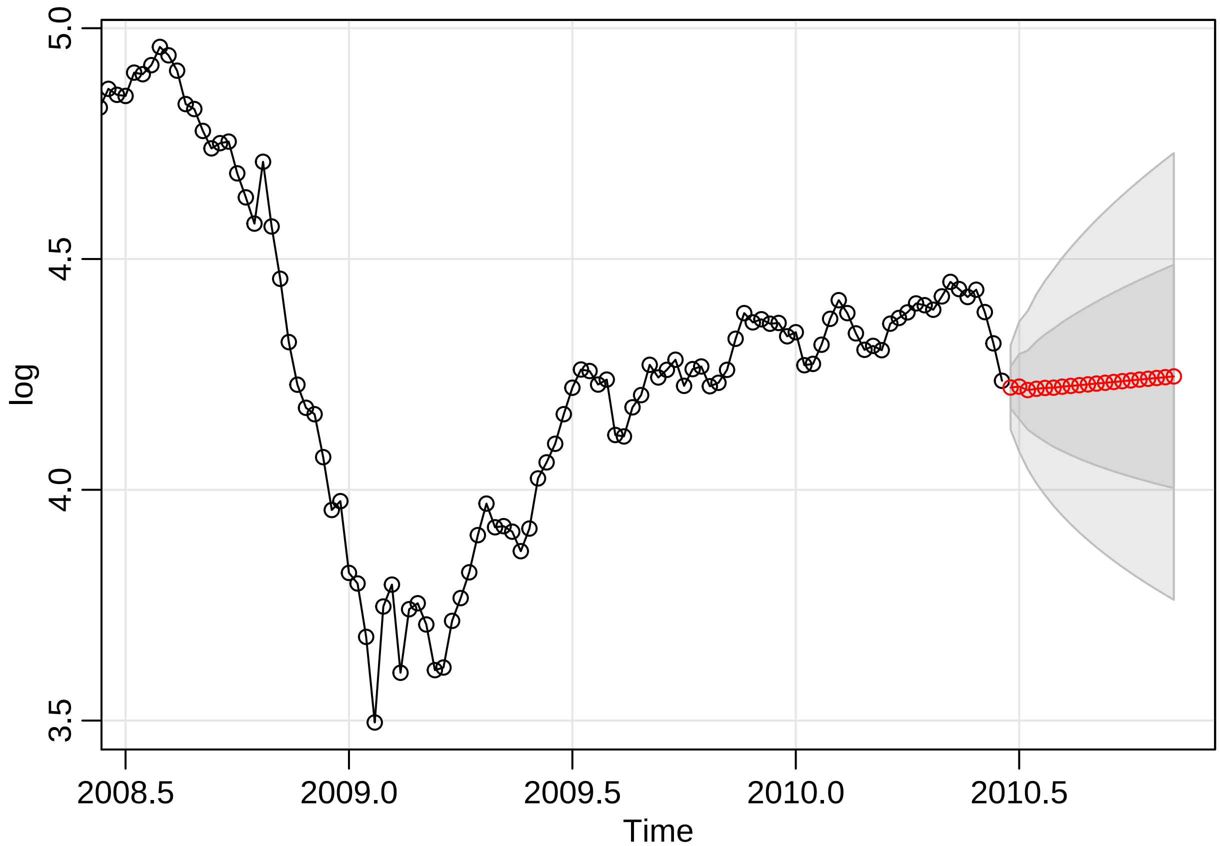
See e

**h) Write the equation of the final model with clearly indicating the AR and MA coefficients. What is the estimate of the white noise variance? What does it tell you?**

$ X_t+0.322X_{t-1}+0.039X_{t-2}-0.118X_{t-3}=Z_t+0.495Z_{t-1} $

**i) Forecast the next 20 values, and submit the plot showing the data with forecast values together with their prediction intervals. State the forecasting values with their standard errors.**

```
sarima.for(log, 20, 3, 1, 1)
```



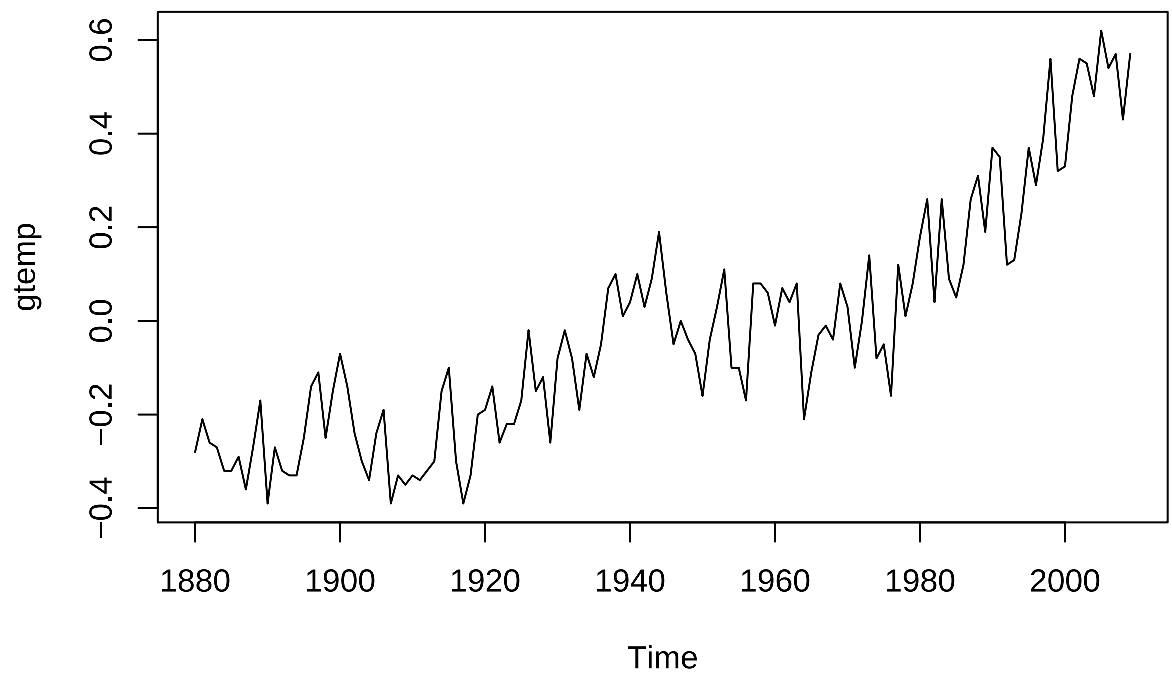

```
## $pred
## Time Series:
## Start = c(2010, 26)
## End = c(2010, 45)
## Frequency = 52
##  [1] 4.221606 4.223591 4.216095 4.218859 4.220638 4.221215 4.223427
##  [8] 4.225044 4.226647 4.228471 4.230153 4.231871 4.233609 4.235323
## [15] 4.237048 4.238773 4.240494 4.242218 4.243941 4.245663
##
## $se
## Time Series:
## Start = c(2010, 26)
## End = c(2010, 45)
## Frequency = 52
##  [1] 0.04560506 0.07031232 0.08581053 0.10227787 0.11597452 0.12802608
##  [7] 0.13940812 0.14978103 0.15949436 0.16868636 0.17737619 0.18566795
## [13] 0.19360716 0.20122963 0.20857567 0.21567140 0.22254061 0.22920438
## [19] 0.23567969 0.24198177
```

---

**3) Global temperature data, gtemp**

**a) Plot of the data**

```r
plot.ts(gtemp)
```



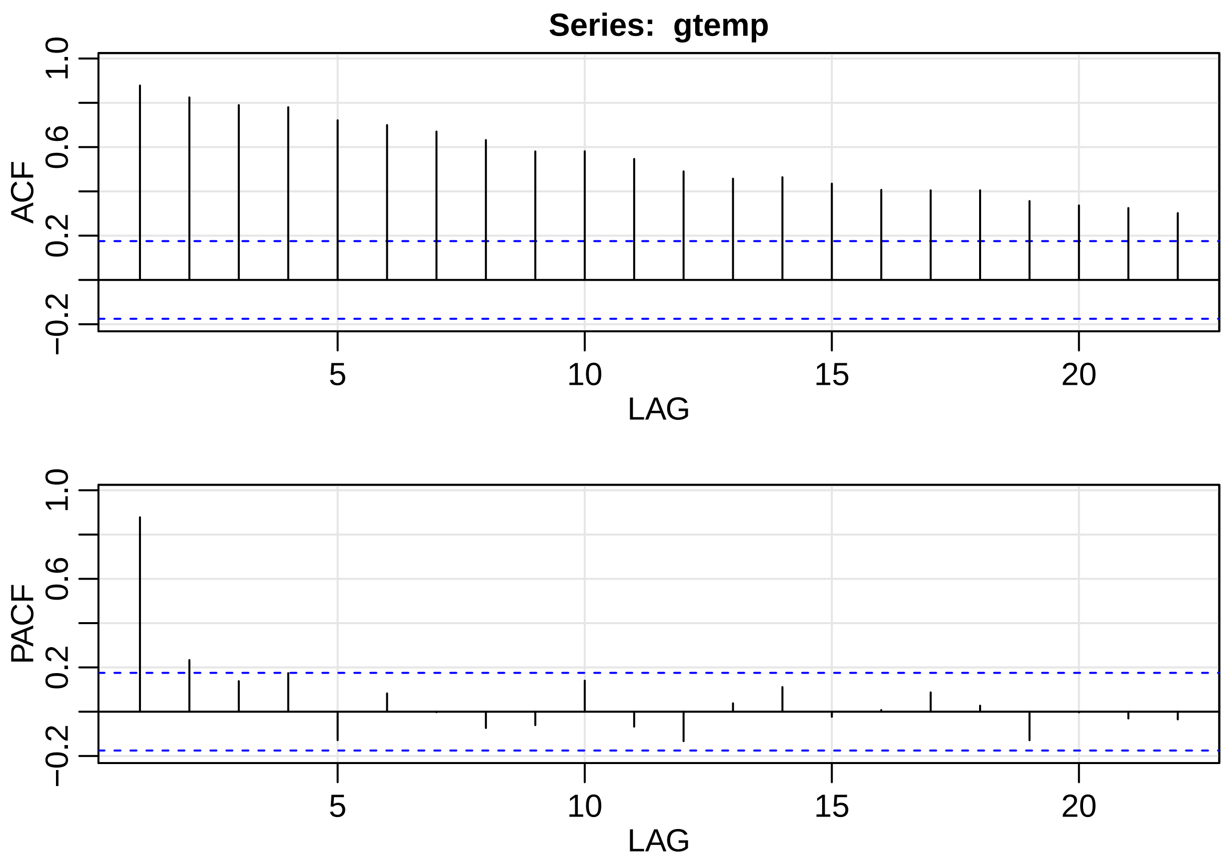

**b) Box-Cox transformation if necessary, and the plot of the transformed data. Note that if a transformation is necessary, the transformed data must be used throughout.**

Transformation was not necessary.

**c) Use appropriate techniques (if necessary) to remove trend and seasonal variations. Explain clearly what method(s) was used. Also submit the plot.**

**d) Plot of ACF and PACF. Explain clearly how you use them to determine a range of ARIMA model. Make sure to use differencing if necessary.**

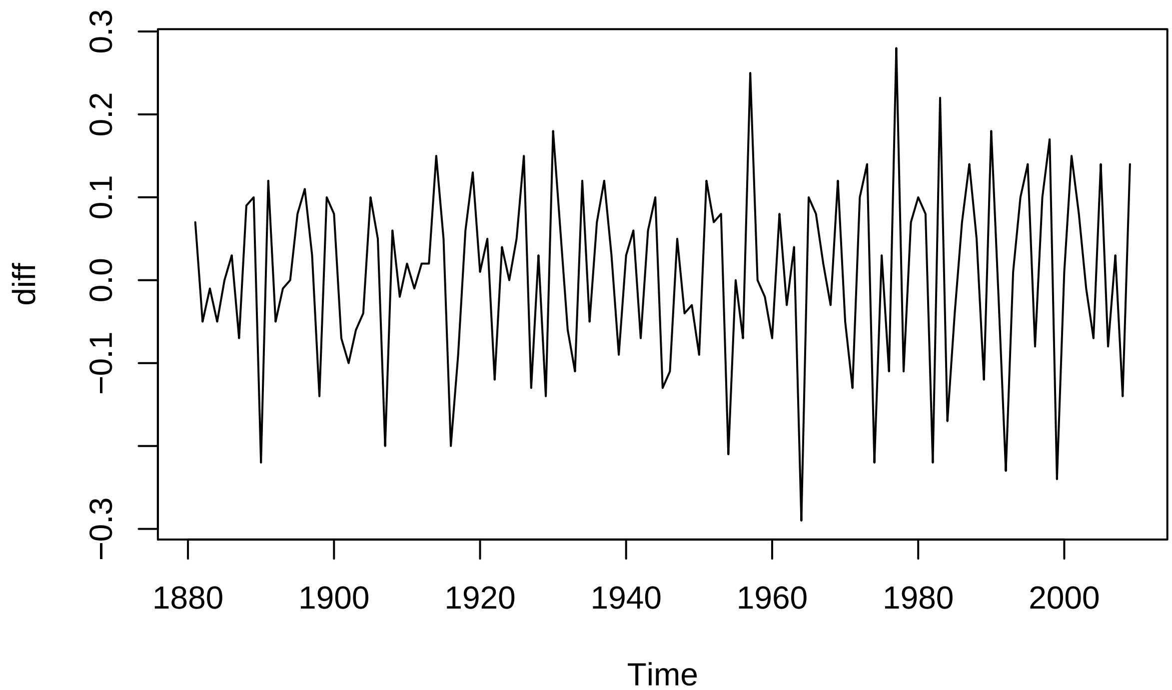
```r
acf2(gtemp)
```



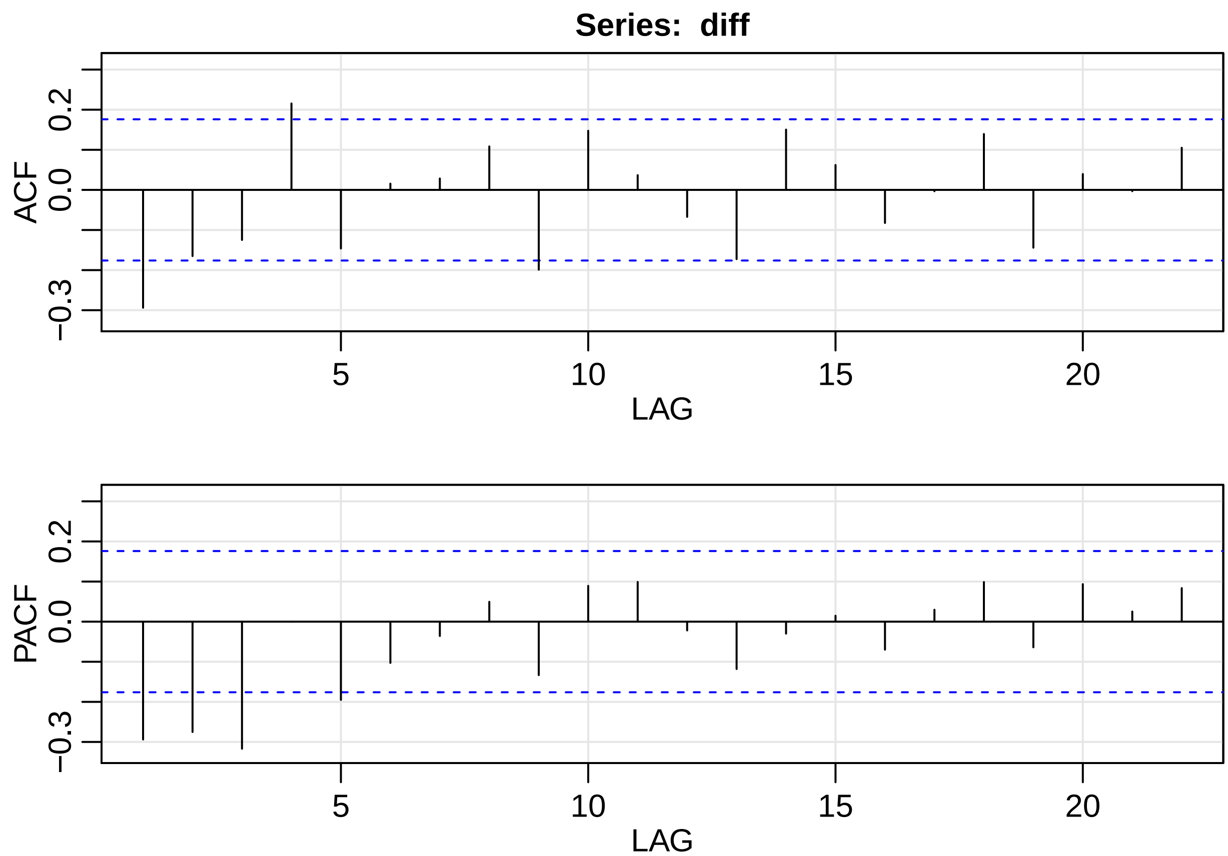

```
##        ACF  PACF
##  [1,] 0.88  0.88
##  [2,] 0.82  0.23
##  [3,] 0.79  0.14
##  [4,] 0.78  0.17
##  [5,] 0.72 -0.13
##  [6,] 0.70  0.08
##  [7,] 0.67  0.00
##  [8,] 0.63 -0.07
##  [9,] 0.58 -0.06
## [10,] 0.58  0.14
## [11,] 0.55 -0.07
## [12,] 0.49 -0.13
## [13,] 0.46  0.04
## [14,] 0.46  0.11
## [15,] 0.44 -0.02
## [16,] 0.41  0.01
## [17,] 0.40  0.09
## [18,] 0.40  0.03
## [19,] 0.36 -0.13
## [20,] 0.34  0.00
## [21,] 0.32 -0.03
## [22,] 0.30 -0.03
```

Data set is not stationary so differencing is required.

```
diff = diff(gtemp)
plot.ts(diff)
```




```
acf2(diff)
```



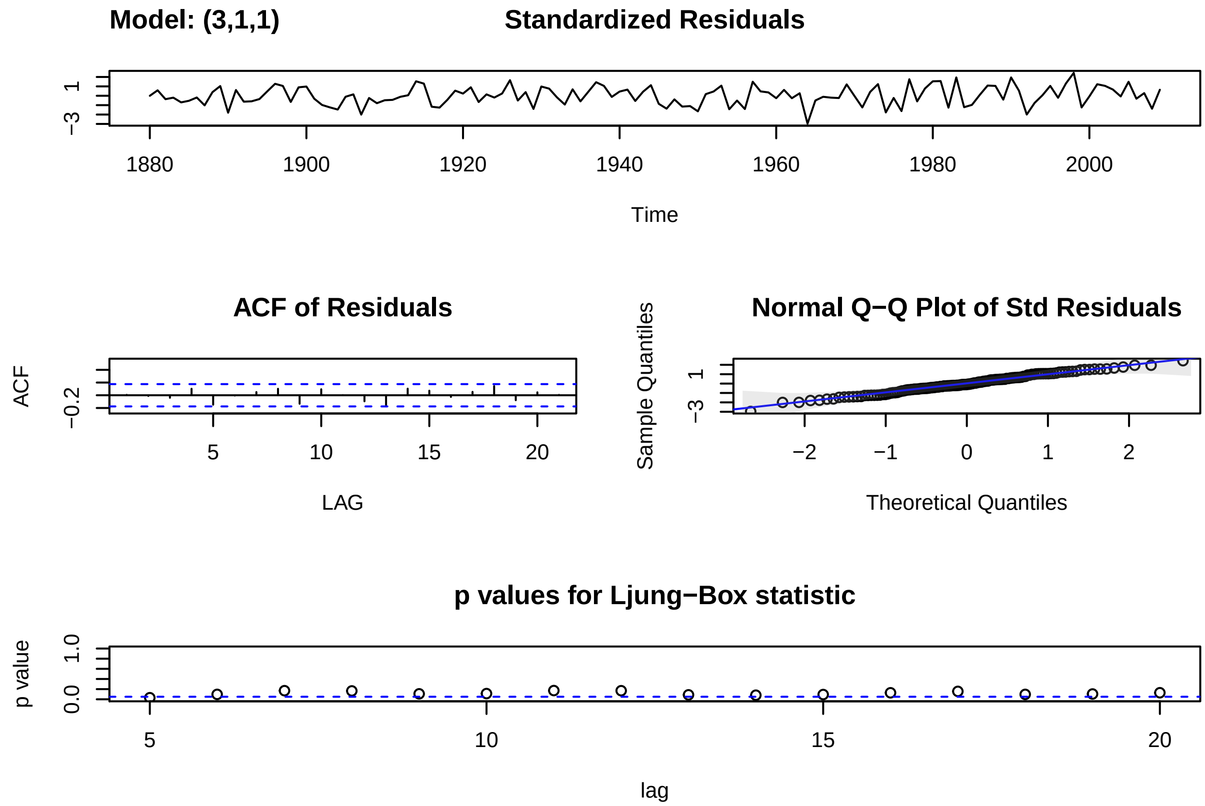

```
##         ACF  PACF
##  [1,] -0.29 -0.29
##  [2,] -0.16 -0.28
##  [3,] -0.12 -0.32
##  [4,]  0.22  0.00
##  [5,] -0.15 -0.20
##  [6,]  0.02 -0.10
##  [7,]  0.03 -0.04
##  [8,]  0.11  0.05
##  [9,] -0.20 -0.13
## [10,]  0.15  0.09
## [11,]  0.04  0.10
## [12,] -0.07 -0.02
## [13,] -0.17 -0.12
## [14,]  0.15 -0.03
## [15,]  0.06  0.01
## [16,] -0.08 -0.07
## [17,]  0.00  0.03
## [18,]  0.14  0.10
## [19,] -0.14 -0.06
## [20,]  0.04  0.09
## [21,]  0.00  0.03
## [22,]  0.11  0.08
```

The ACF is used to determine q which would be 1 in this case and the PACF is used to determine p which would be 3 in this case.

**e) Using certain criterion, determine an optimal ARMA(p, q) model.**

```
sarima(gtemp, 3, 1, 1)
```

```
## initial  value -2.204527
## iter   2 value -2.320466
## iter   3 value -2.340026
## iter   4 value -2.342073
## iter   5 value -2.343602
## iter   6 value -2.344256
## iter   7 value -2.344340
## iter   8 value -2.344683
## iter   9 value -2.345686
## iter  10 value -2.346309
## iter  11 value -2.347424
## iter  12 value -2.347474
## iter  13 value -2.347475
## iter  14 value -2.347475
## iter  15 value -2.347476
## iter  16 value -2.347476
## iter  16 value -2.347476
## iter  16 value -2.347476
## final  value -2.347476
## converged
## initial  value -2.353882
## iter   2 value -2.353904
## iter   3 value -2.353915
## iter   4 value -2.353922
## iter   5 value -2.353925
## iter   6 value -2.353934
## iter   7 value -2.353935
## iter   8 value -2.353935
## iter   8 value -2.353935
## iter   8 value -2.353935
## final  value -2.353935
## converged
```




```
## $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      ar2      ar3      ma1  constant
##       0.0697  -0.1816  -0.1300  -0.5806    0.0064
## s.e.  0.2297   0.1257   0.1327   0.2245    0.0029
##
## sigma^2 estimated as 0.00898:  log likelihood = 120.61,  aic = -229.23
##
## $degrees_of_freedom
## [1] 124
##
## $ttable
##          Estimate     SE t.value p.value
## ar1        0.0697 0.2297  0.3035  0.7620
## ar2       -0.1816 0.1257 -1.4444  0.1512
## ar3       -0.1300 0.1327 -0.9797  0.3291
## ma1       -0.5806 0.2245 -2.5861  0.0109
## constant   0.0064 0.0029  2.2514  0.0261
##
## $AIC
```

```
## [1] -3.635873
##
## $AICc
## [1] -3.615235
##
## $BIC
## [1] -4.525583
```

**f) Using hypothesis testing methods, check if certain parameters of the ARMA model can be removed.**

**g) Performed diagnostic check for the model you obtained. Submit the appropriate plots. Make sure to use Box-Ljung statistics to test for white noise. If diagnostic check failed, adjust your model and start all over. Compare all possible models you considered with AIC values and p-values of the Box-Ljung statistics. Determine the final model.**

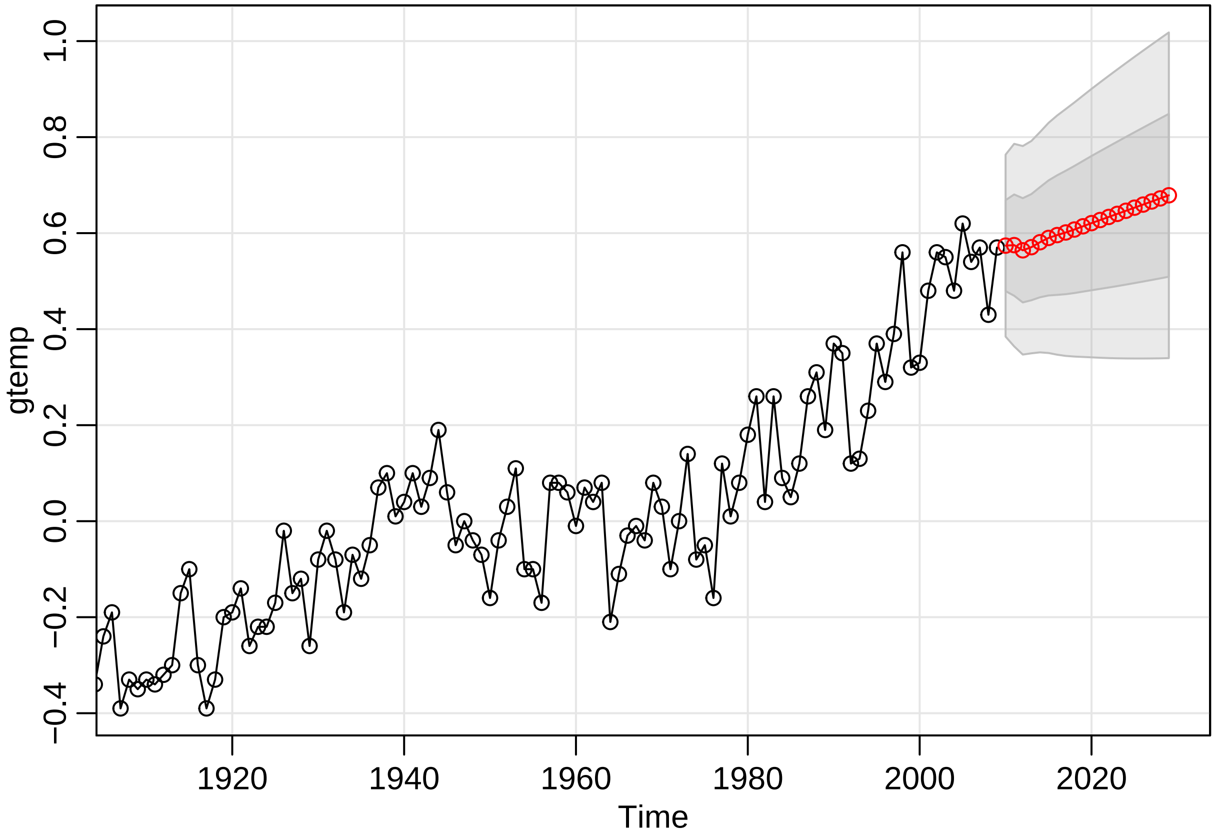
See e

**h) Write the equation of the final model with clearly indicating the AR and MA coefficients. What is the estimate of the white noise variance? What does it tell you?**

$ X_t-0.070X_{t-1}+0.182X_{t-2}+0.130X_{t-3}=Z_t-0.581Z_{t-1} $

**i) Forecast the next 20 values, and submit the plot showing the data with forecast values together with their prediction intervals. State the forecasting values with their standard errors.**

```
sarima.for(gtemp, 20, 3, 1, 1)
```



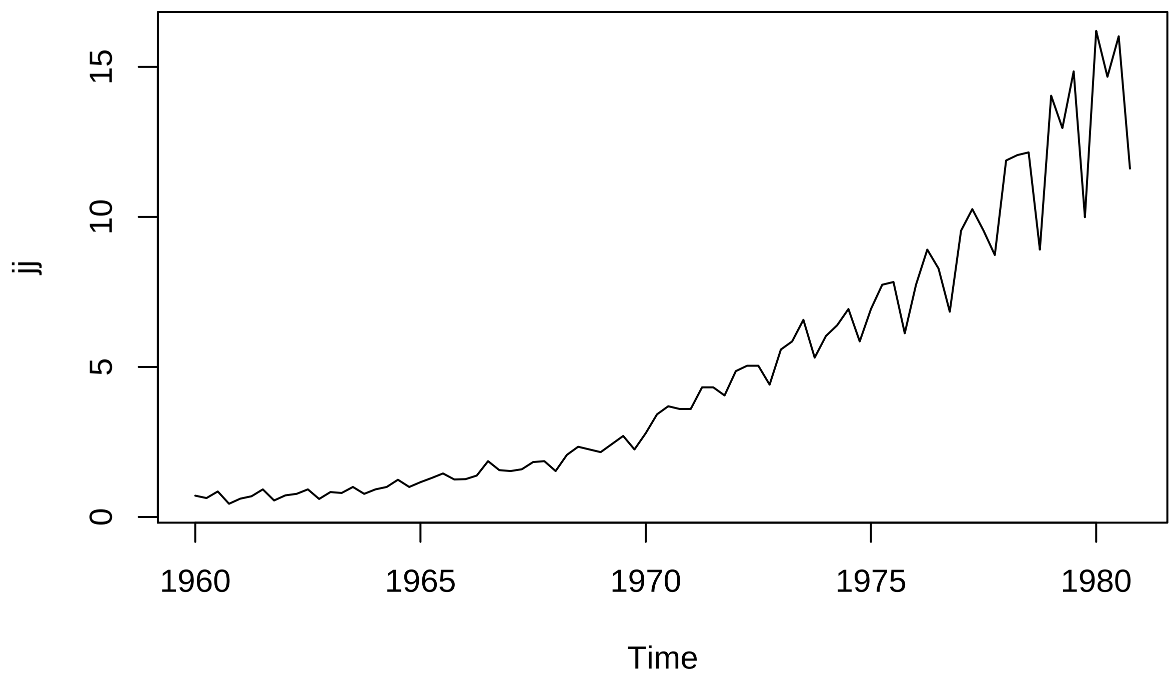

```
## $pred
## Time Series:
## Start = 2010
## End = 2029
## Frequency = 1
##  [1] 0.5739982 0.5750626 0.5642230 0.5707629 0.5810569 0.5900044 0.5959175
##  [8] 0.6013756 0.6075280 0.6142058 0.6208532 0.6273130 0.6336968 0.6401134
## [15] 0.6465705 0.6530343 0.6594869 0.6659323 0.6723784 0.6788272
##
## $se
## Time Series:
## Start = 2010
## End = 2029
## Frequency = 1
##  [1] 0.09476095 0.10548852 0.10859026 0.11056579 0.11469293 0.11983914
##  [7] 0.12454826 0.12855295 0.13230554 0.13605812 0.13980185 0.14345203
## [13] 0.14698326 0.15041786 0.15377925 0.15707477 0.16030405 0.16346817
## [19] 0.16657110 0.16961725
```

---

**4) Johnson and Johnson earnings, jj**

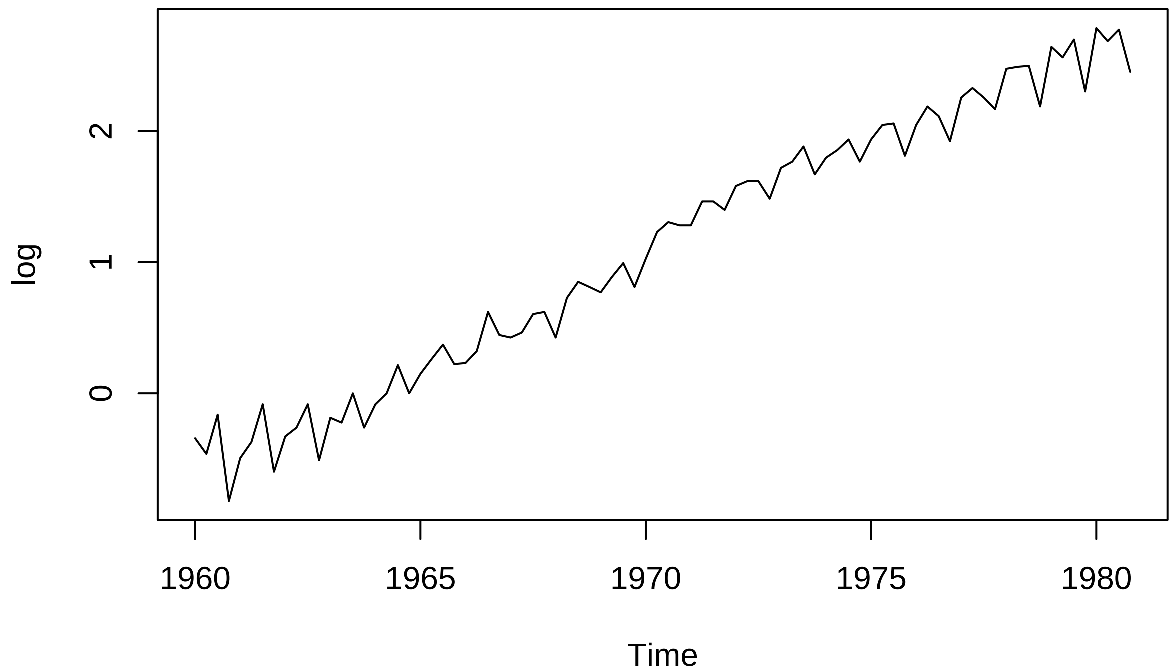
**a) Plot of the data**

```
plot.ts(jj)
```




**b) Box-Cox transformation if necessary, and the plot of the transformed data. Note that if a transformation is necessary, the transformed data must be used throughout.**

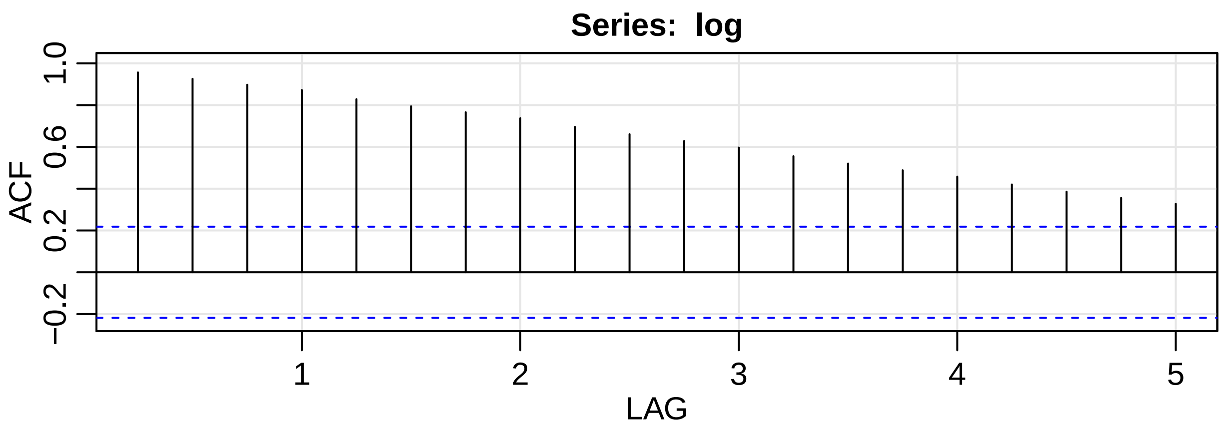
```
log = log(jj)
plot.ts(log)
```

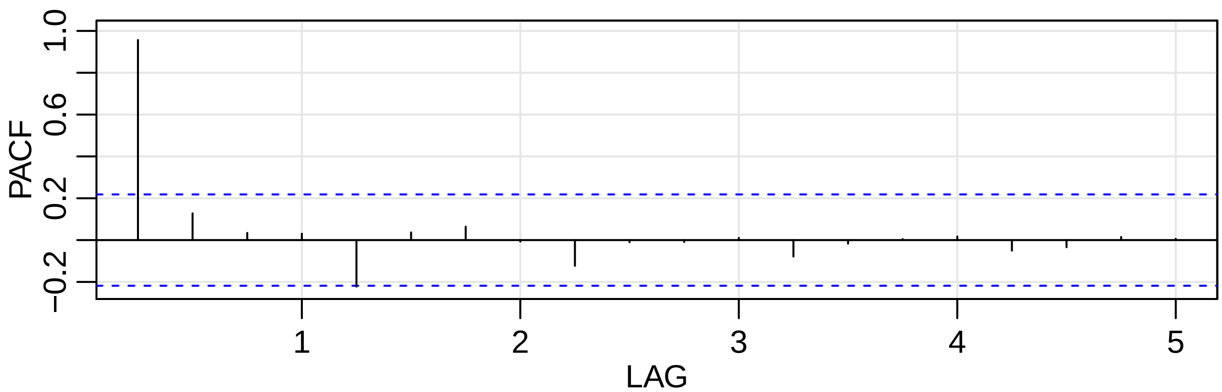



**c) Use appropriate techniques (if necessary) to remove trend and seasonal variations. Explain clearly what method(s) was used. Also submit the plot.**

**d) Plot of ACF and PACF. Explain clearly how you use them to determine a range of ARIMA model. Make sure to use differencing if necessary.**

```
acf2(log)
```

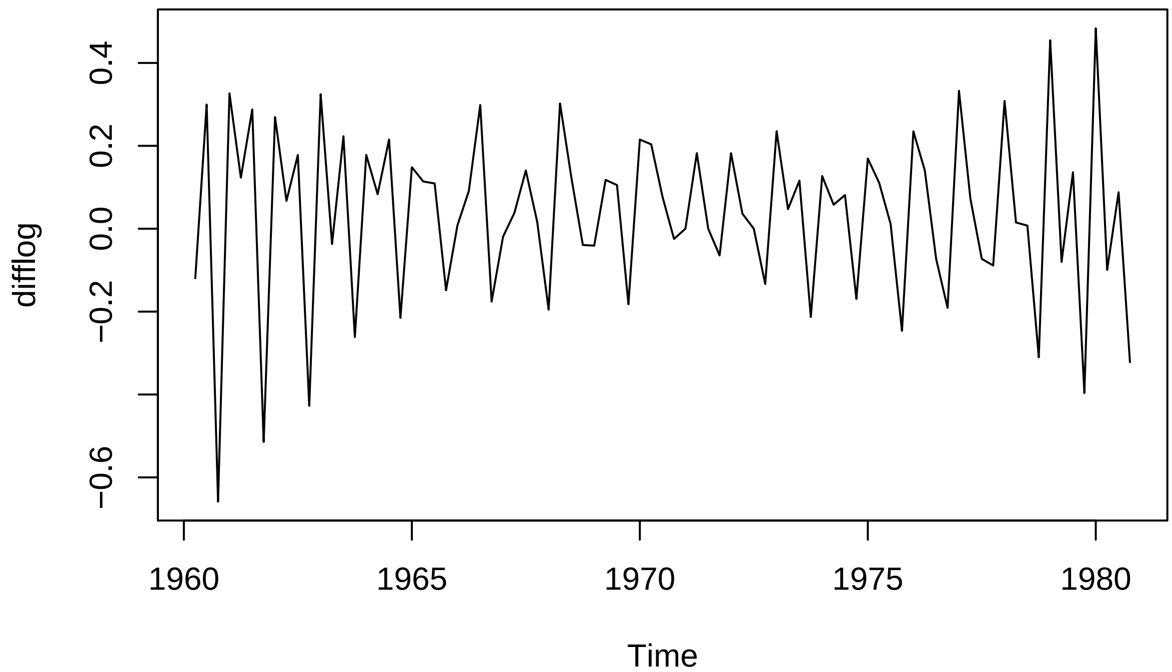


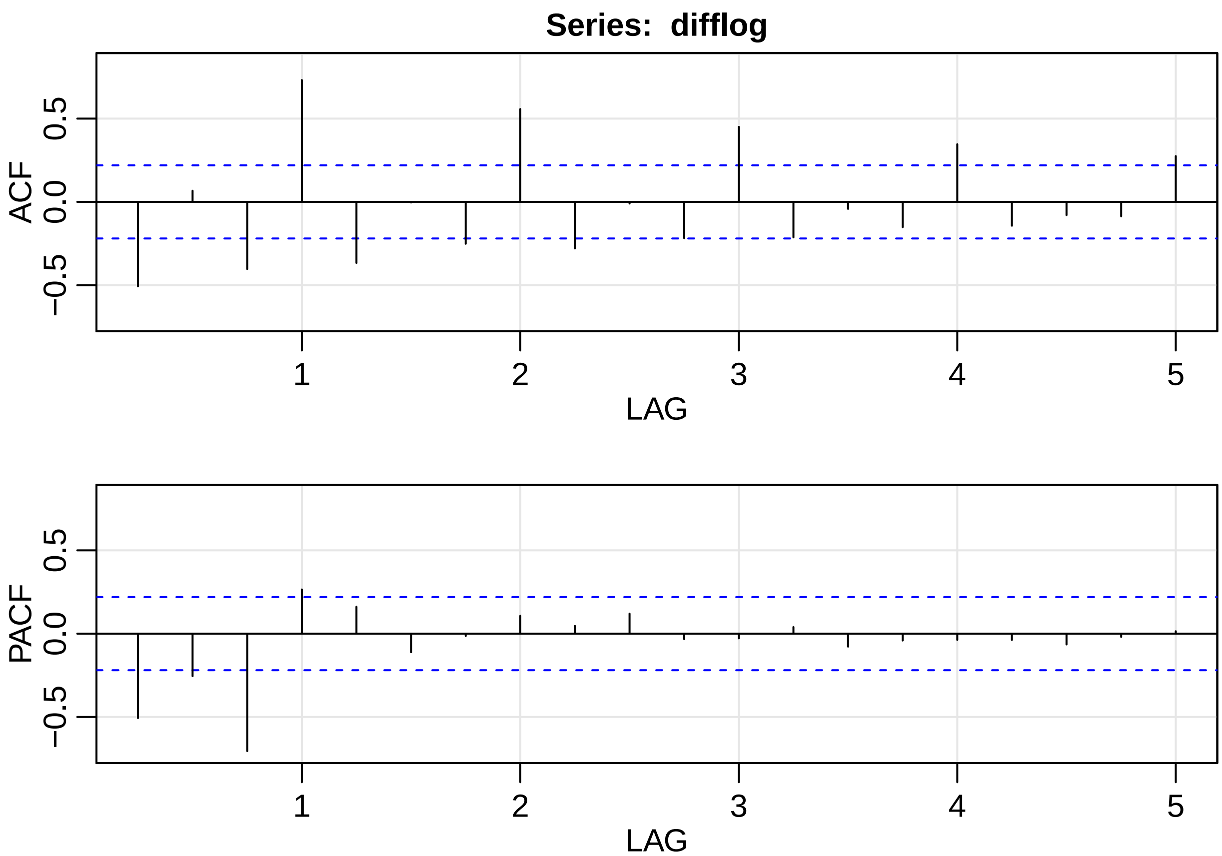

```
##        ACF  PACF
##  [1,] 0.96  0.96
##  [2,] 0.93  0.13
##  [3,] 0.90  0.03
##  [4,] 0.87  0.03
##  [5,] 0.83 -0.22
##  [6,] 0.79  0.04
##  [7,] 0.77  0.06
##  [8,] 0.74 -0.01
##  [9,] 0.70 -0.12
## [10,] 0.66 -0.01
## [11,] 0.63 -0.01
## [12,] 0.60  0.01
## [13,] 0.56 -0.08
## [14,] 0.52 -0.02
## [15,] 0.49  0.00
## [16,] 0.46  0.02
## [17,] 0.42 -0.05
## [18,] 0.39 -0.03
## [19,] 0.36  0.02
## [20,] 0.33  0.01
```

Data set is not stationary so differencing is required.

```
difflog = diff(log)
plot.ts(difflog)
```




```
acf2(difflog)
```




```
##        ACF  PACF
##  [1,] -0.51 -0.51
##  [2,]  0.07 -0.26
##  [3,] -0.40 -0.70
##  [4,]  0.73  0.27
##  [5,] -0.37  0.16
##  [6,]  0.00 -0.11
##  [7,] -0.25 -0.01
##  [8,]  0.56  0.11
##  [9,] -0.28  0.05
## [10,] -0.01  0.12
## [11,] -0.22 -0.03
## [12,]  0.45 -0.03
## [13,] -0.21  0.04
## [14,] -0.04 -0.08
## [15,] -0.15 -0.04
## [16,]  0.35 -0.04
## [17,] -0.14 -0.04
## [18,] -0.08 -0.06
## [19,] -0.09 -0.02
## [20,]  0.27  0.01
```

The ACF is used to determine q which would be 5 in this case and the PACF is used to determine p which would be 1 in this case.

**e) Using certain criterion, determine an optimal ARMA(p, q) model.**

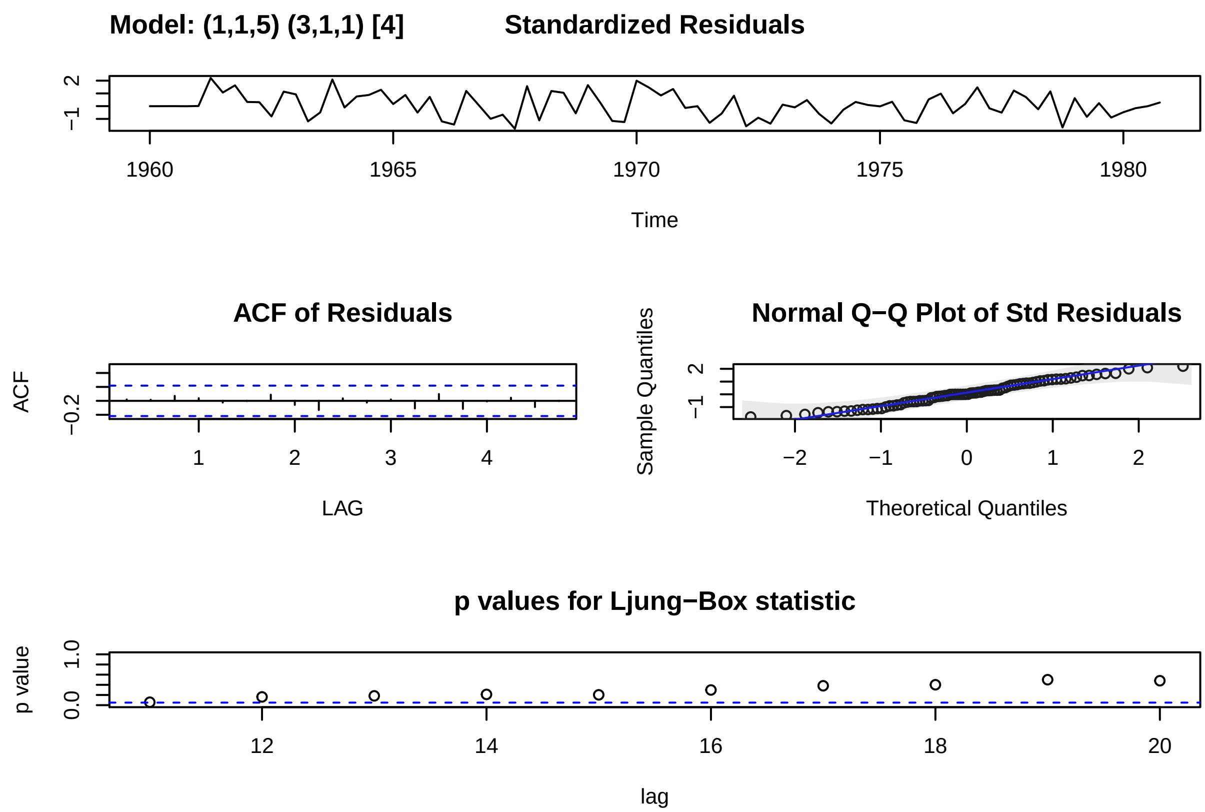
```r
sarima(log, 1,1,5, 3,1,1, 4)
```

```
## initial  value -2.236976
## iter   2 value -2.297512
## iter   3 value -2.402648
## iter   4 value -2.455297
## iter   5 value -2.475545
## iter   6 value -2.479297
## iter   7 value -2.500791
## iter   8 value -2.505027
## iter   9 value -2.508980
## iter  10 value -2.510167
## iter  11 value -2.511142
## iter  12 value -2.511385
## iter  13 value -2.511851
## iter  14 value -2.517313
## iter  15 value -2.530261
## iter  16 value -2.533044
## iter  17 value -2.547628
## iter  18 value -2.550606
## iter  19 value -2.552807
## iter  20 value -2.559340
## iter  21 value -2.565085
## iter  22 value -2.572611
## iter  23 value -2.583633
## iter  23 value -2.583633
## iter  24 value -2.583972
## iter  24 value -2.583972
## iter  25 value -2.583981
## iter  25 value -2.583981
## iter  25 value -2.583981
## final  value -2.583981
## converged
## initial  value -2.413537
## iter   2 value -2.421644
## iter   3 value -2.431981
## iter   4 value -2.433359
## iter   5 value -2.434767
## iter   6 value -2.436164
## iter   7 value -2.441811
## iter   8 value -2.446449
## iter   9 value -2.455595
## iter  10 value -2.465806
```

```
## Warning in log(s2): NaNs produced
```

```
## iter  11 value -2.467688
## iter  12 value -2.471041
## iter  13 value -2.473191
## iter  14 value -2.474208
## iter  15 value -2.474773
## iter  16 value -2.475835
## iter  17 value -2.476478
## iter  18 value -2.476833
## iter  19 value -2.476915
```

```
## iter  20 value -2.476944
## iter  21 value -2.476985
## iter  22 value -2.477022
## iter  23 value -2.477031
## iter  24 value -2.477036
## iter  25 value -2.477043
## iter  26 value -2.477046
## iter  27 value -2.477055
## iter  28 value -2.477088
## iter  29 value -2.477158
## iter  30 value -2.477288
## iter  31 value -2.477464
## iter  32 value -2.478492
## iter  33 value -2.478641
## iter  34 value -2.479328
## iter  35 value -2.480239
## iter  36 value -2.481030
## iter  37 value -2.481990
## iter  38 value -2.483216
## iter  39 value -2.483908
## iter  40 value -2.487112
## iter  41 value -2.492090
## iter  42 value -2.492817
## iter  43 value -2.493657
## iter  44 value -2.494718
## iter  45 value -2.495799
## iter  46 value -2.496088
## iter  47 value -2.496250
## iter  48 value -2.496335
## iter  49 value -2.496372
## iter  50 value -2.496382
## iter  51 value -2.496383
## iter  52 value -2.496384
## iter  53 value -2.496384
## iter  54 value -2.496385
## iter  55 value -2.496385
## iter  56 value -2.496385
## iter  56 value -2.496385
## iter  56 value -2.496385
## final  value -2.496385
## 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, optim.control = list(trace = trc,
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1      ma1      ma2      ma3     ma4      ma5     sar1
##       -0.2434  -0.3600  -0.0612  -0.3030  0.8314  -0.3343  -1.8467
## s.e.   0.2334   0.2343   0.1110   0.1045  0.1308   0.2060   0.1527
##          sar2     sar3    sma1
##       -1.0858  -0.1652  0.9998
## s.e.   0.2498   0.1337  0.1840
##
## sigma^2 estimated as 0.005777:  log likelihood = 85.12,  aic = -148.24
##
## $degrees_of_freedom
## [1] 69
##
## $ttable
##     Estimate     SE  t.value p.value
## ar1  -0.2434 0.2334  -1.0431  0.3005
## ma1  -0.3600 0.2343  -1.5363  0.1290
## ma2  -0.0612 0.1110  -0.5508  0.5836
## ma3  -0.3030 0.1045  -2.9001  0.0050
```

```
## ma4    0.8314 0.1308   6.3552  0.0000
## ma5   -0.3343 0.2060  -1.6224  0.1093
## sar1  -1.8467 0.1527 -12.0962  0.0000
## sar2  -1.0858 0.2498  -4.3460  0.0000
## sar3  -0.1652 0.1337  -1.2358  0.2207
## sma1   0.9998 0.1840   5.4353  0.0000
##
## $AIC
## [1] -3.915857
##
## $AICc
## [1] -3.848397
##
## $BIC
## [1] -4.626474
```

**f) Using hypothesis testing methods, check if certain parameters of the ARMA model can be removed.**

**g) Performed diagnostic check for the model you obtained. Submit the appropriate plots. Make sure to use Box-Ljung statistics to test for white noise. If diagnostic check failed, adjust your model and start all over. Compare all possible models you considered with AIC values and p-values of the Box-Ljung statistics. Determine the final model.**

See e

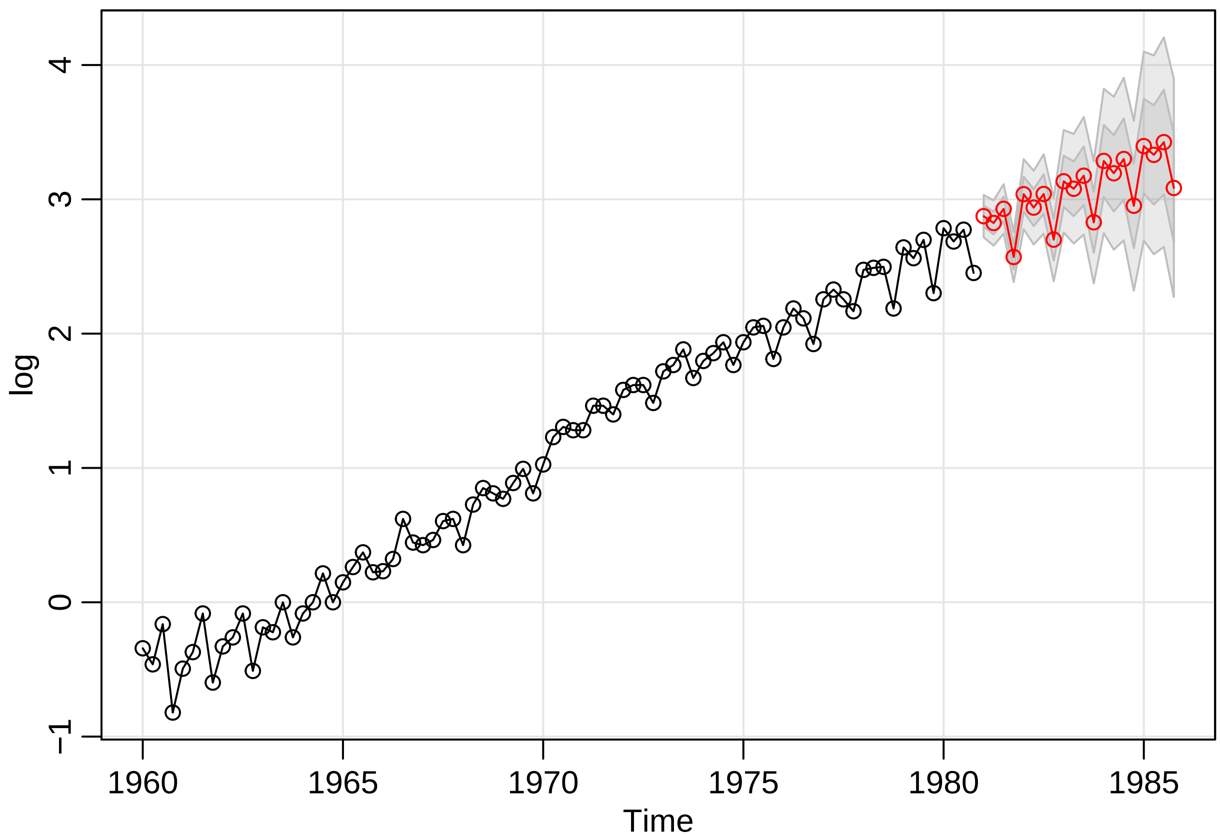
**h) Write the equation of the final model with clearly indicating the AR and MA coefficients. What is the estimate of the white noise variance? What does it tell you?**

$ X_t+0.243X_{t-1}=Z_t-0.360Z_{t-1}-0.061Z_{t-2}-0.303Z_{t-3}+0.831Z_{t-4}-0.334Z_{t-5} $

**i) Forecast the next 20 values, and submit the plot showing the data with forecast values together with their prediction intervals. State the forecasting values with their standard errors.**

```r
sarima.for(log, 20, 1,1,5, 3,1,1, 4)
```

```
## Warning in log(s2): NaNs produced
```




```
## $pred
##          Qtr1     Qtr2     Qtr3     Qtr4
## 1981 2.874475 2.823692 2.928948 2.570597
## 1982 3.038258 2.938318 3.039623 2.699907
## 1983 3.134291 3.078486 3.175658 2.828897
## 1984 3.285821 3.193898 3.300145 2.952171
## 1985 3.396159 3.331135 3.425763 3.084600
##
## $se
##            Qtr1       Qtr2       Qtr3       Qtr4
## 1981 0.07894928 0.08421382 0.09203012 0.09302182
## 1982 0.13013733 0.13716118 0.14754335 0.15556817
## 1983 0.19062120 0.20385443 0.21850745 0.22728934
## 1984 0.26867147 0.28443223 0.30266968 0.31620215
## 1985 0.35163935 0.37002601 0.38991586 0.40575462
```