lab week 8

Mason Wong

11th April 2022

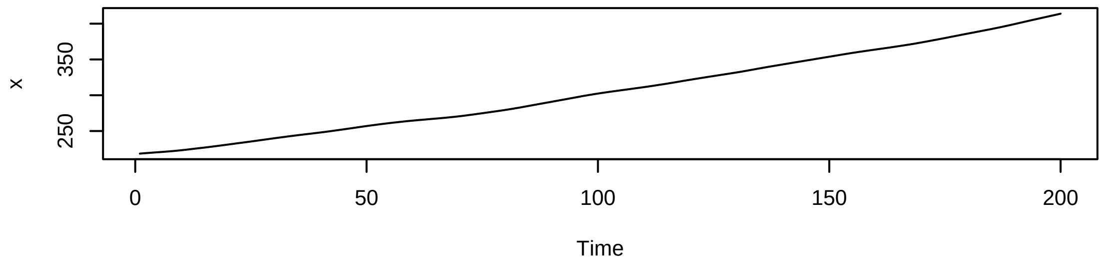
R version

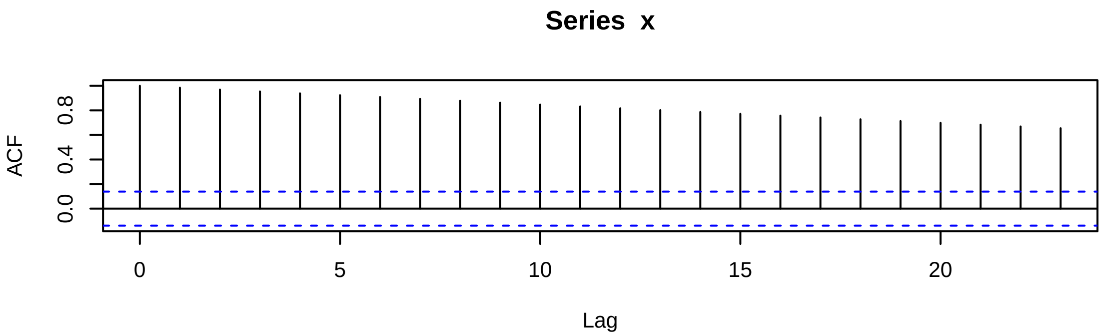
The R version used is `4.1.01`

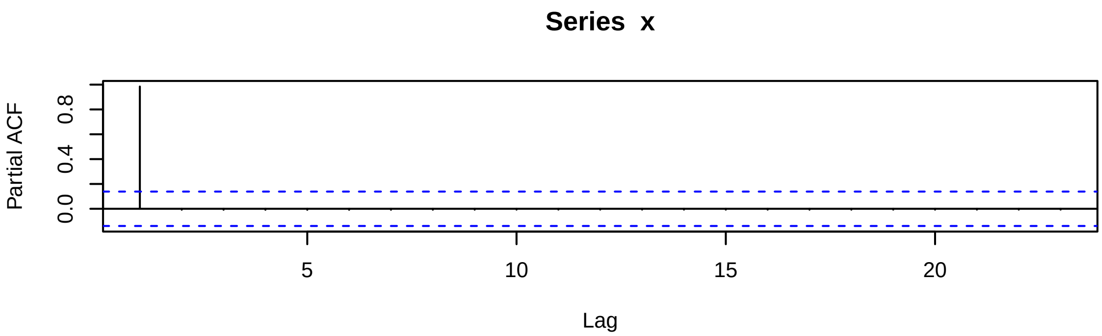
Question 1

(i)

```
x = c(218.45, 218.98, 219.48, 219.96, 220.42, 220.9, 221.4, 221.94, 222.54, 223.21,
    223.93, 224.67, 225.44, 226.19, 226.95, 227.73, 228.53, 229.36, 230.21, 231.08,
    231.96, 232.82, 233.66, 234.52, 235.38, 236.26, 237.16, 238.06, 238.96, 239.86,
    240.74, 241.59, 242.42, 243.25, 244.06, 244.86, 245.63, 246.4, 247.16, 247.96,
    248.78, 249.65, 250.53, 251.42, 252.32, 253.24, 254.18, 255.12, 256.06, 256.97,
    257.86, 258.72, 259.55, 260.35, 261.13, 261.88, 262.6, 263.29, 263.95, 264.57,
    265.17, 265.74, 266.29, 266.85, 267.41, 267.98, 268.57, 269.2, 269.87, 270.59,
    271.37, 272.21, 273.09, 273.98, 274.88, 275.78, 276.69, 277.61, 278.54, 279.51,
    280.52, 281.58, 282.7, 283.86, 285.04, 286.23, 287.41, 288.58, 289.74, 290.89,
    292.06, 293.23, 294.4, 295.59, 296.79, 297.98, 299.15, 300.26, 301.33, 302.35,
    303.33, 304.29, 305.2, 306.09, 306.95, 307.82, 308.69, 309.57, 310.46, 311.38,
    312.32, 313.29, 314.27, 315.27, 316.29, 317.35, 318.44, 319.54, 320.63, 321.71,
    322.76, 323.8, 324.82, 325.82, 326.82, 327.82, 328.83, 329.83, 330.84, 331.88,
    332.95, 334.08, 335.24, 336.41, 337.57, 338.73, 339.85, 340.94, 342.03, 343.11,
    344.19, 345.27, 346.34, 347.39, 348.46, 349.53, 350.59, 351.67, 352.75, 353.83,
    354.91, 355.99, 357.07, 358.12, 359.15, 360.13, 361.1, 362.04, 362.97, 363.88,
    364.78, 365.69, 366.63, 367.57, 368.54, 369.51, 370.53, 371.59, 372.69, 373.84,
    375.03, 376.23, 377.45, 378.68, 379.92, 381.17, 382.44, 383.7, 384.97, 386.22,
    387.46, 388.69, 389.93, 391.18, 392.46, 393.77, 395.11, 396.49, 397.91, 399.36,
    400.84, 402.32, 403.8, 405.27, 406.72, 408.15, 409.58, 411.02, 412.47, 413.94)
par(mfrow = c(3, 1))
ts.plot(x)
acf(x)
pacf(x)
```



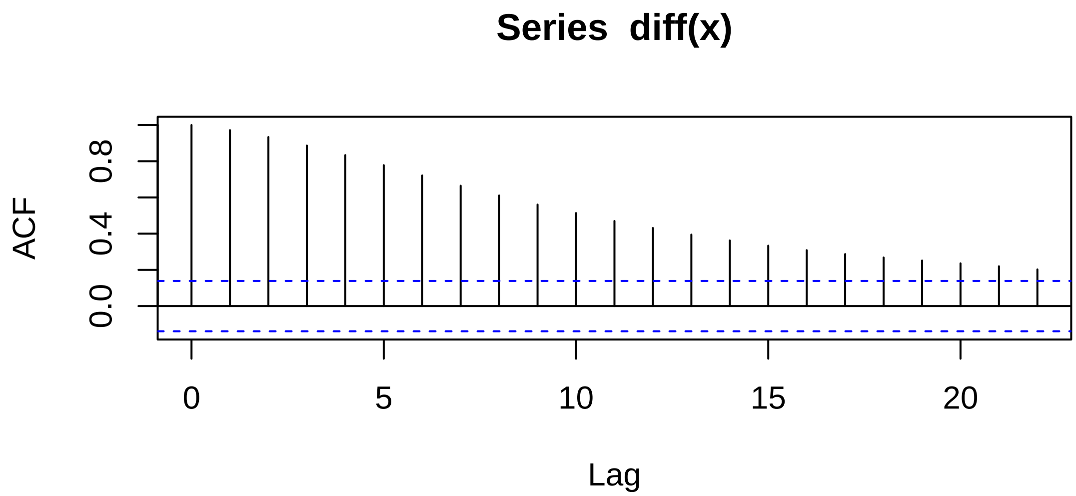


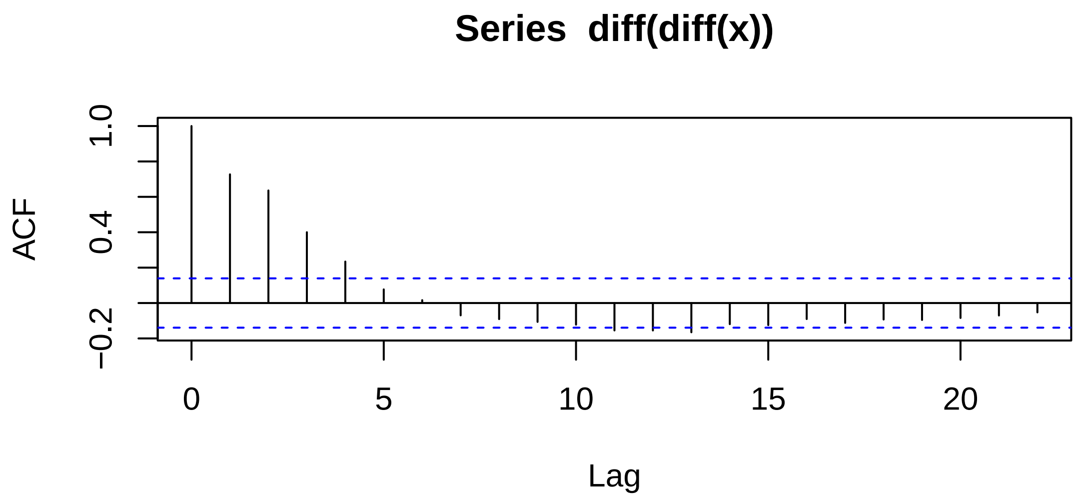


(ii) we clearly see that this plot is not stationary as the acf is not decaying quickly

(iii) The plot can be seen below:

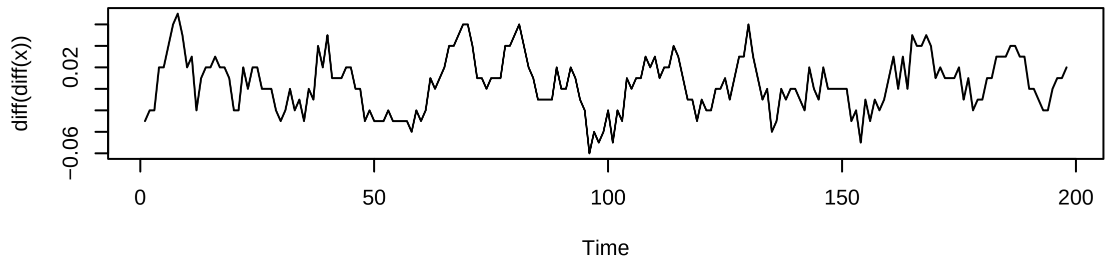
```
par(mfrow = c(2, 1))
acf(diff(x))
acf(diff(diff(x)))
```

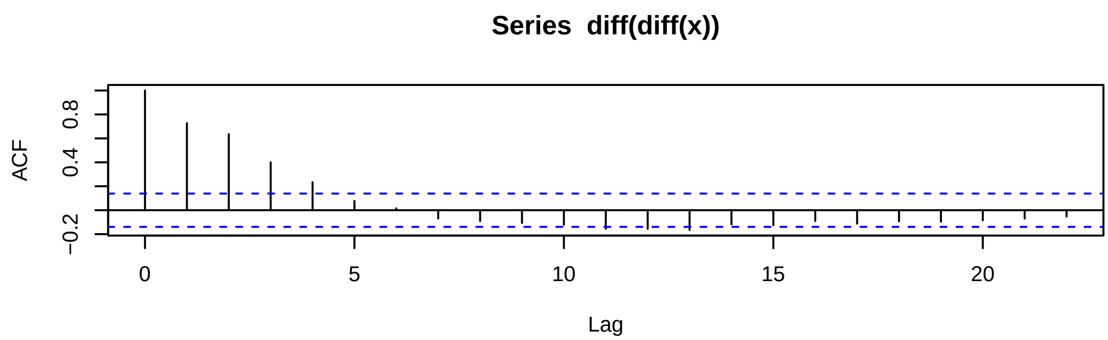


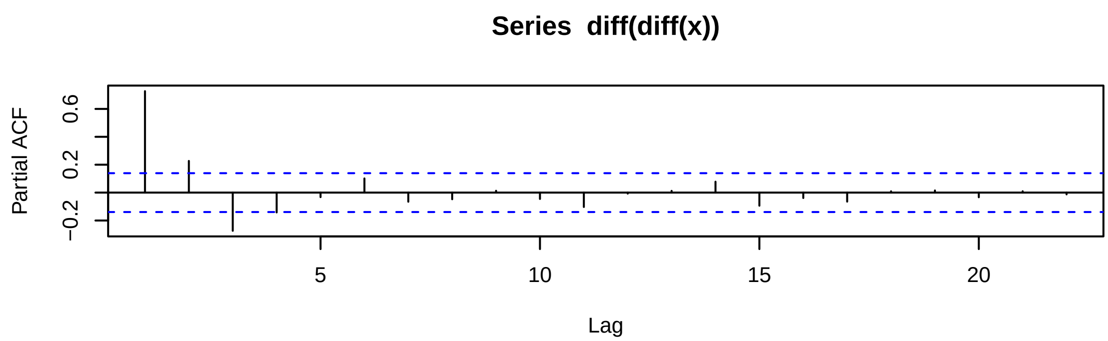


(iv) We see that $d = 2$ is suitable here.

```
par(mfrow = c(3, 1))
ts.plot(diff(diff(x)))
acf(diff(diff(x)))
pacf(diff(diff(x)))
```







Question 3

```r
data = diff(diff(x))
```

(i)

```r
f_ar1 = arima(data, order = c(1, 0, 0))
f_ar1
```

```
##
## Call:
## arima(x = data, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.7318     0.0045
## s.e.  0.0483     0.0045
##
## sigma^2 estimated as 0.0002977:  log likelihood = 522.5,  aic = -1038.99
```

```r
f_ar2 = arima(data, order = c(2, 0, 0))
f_ar2
```

```
##
## Call:
## arima(x = data, order = c(2, 0, 0))
##
## Coefficients:
##         ar1     ar2  intercept
##       0.564  0.2293     0.0043
## s.e.  0.069  0.0692     0.0056
##
## sigma^2 estimated as 0.0002819:  log likelihood = 527.84,  aic = -1047.67
```

```r
f_ar3 = arima(data, order = c(3, 0, 0))
f_ar3
```

```
##
## Call:
## arima(x = data, order = c(3, 0, 0))
##
## Coefficients:
##          ar1     ar2      ar3  intercept
##       0.6275  0.3843  -0.2754     0.0045
## s.e.  0.0680  0.0766   0.0681     0.0043
##
## sigma^2 estimated as 0.0002601:  log likelihood = 535.67,  aic = -1061.35
```

```r
f_ma3 = arima(data, order = c(0, 0, 3))
f_ma3
```

```
##
## Call:
## arima(x = data, order = c(0, 0, 3))
##
## Coefficients:
##          ma1     ma2     ma3  intercept
##       0.5341  0.6813  0.3719     0.0045
## s.e.  0.0608  0.0564  0.0580     0.0030
##
## sigma^2 estimated as 0.0002709:  log likelihood = 531.58,  aic = -1053.15
```

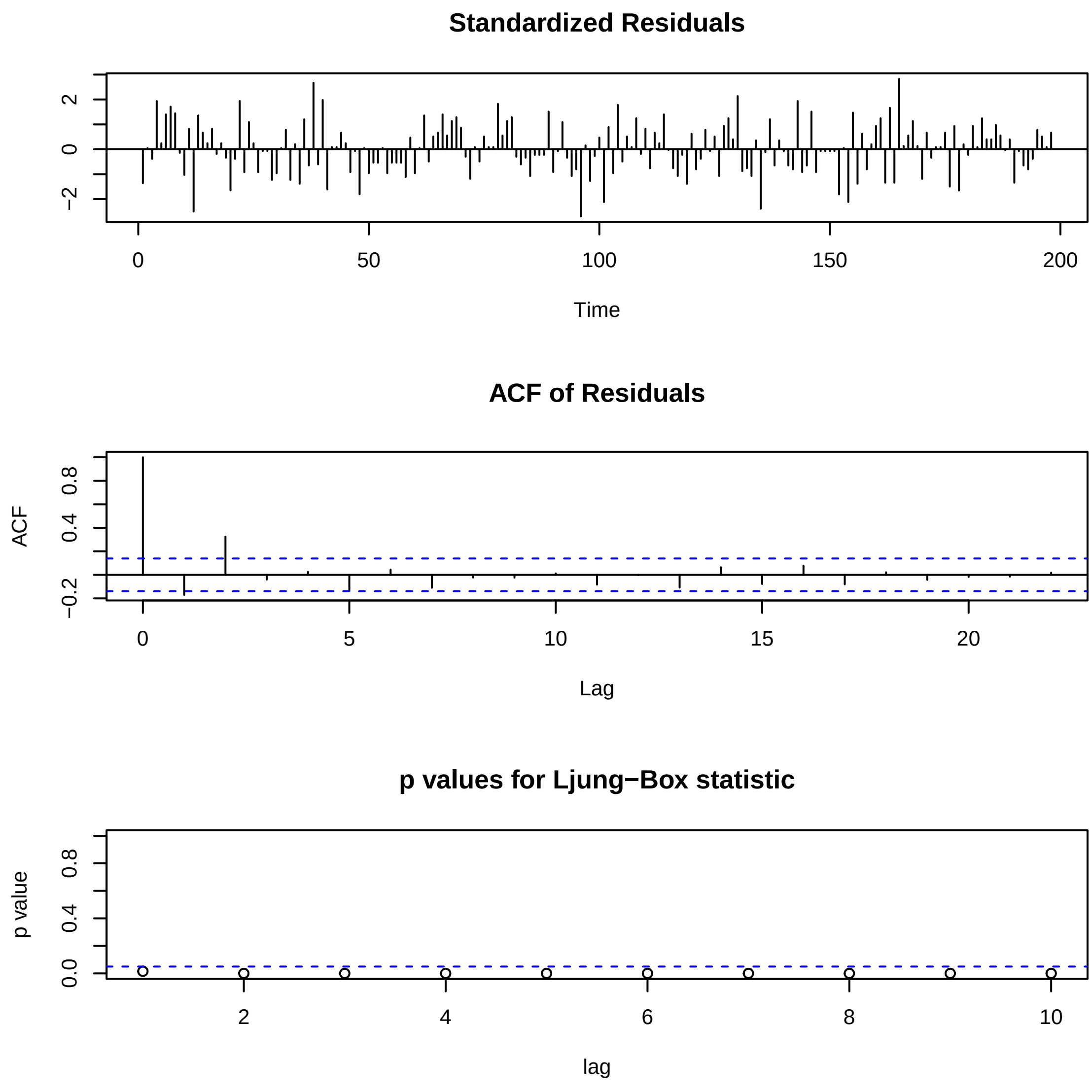
```r
f_arma12 = arima(data, order = c(1, 0, 2))
f_arma12
```

```
##
## Call:
## arima(x = data, order = c(1, 0, 2))
##
## Coefficients:
##          ar1      ma1     ma2  intercept
##       0.6810  -0.1112  0.4202     0.0044
## s.e.  0.0716   0.0794  0.0757     0.0046
##
## sigma^2 estimated as 0.0002539:  log likelihood = 537.99,  aic = -1065.98
```

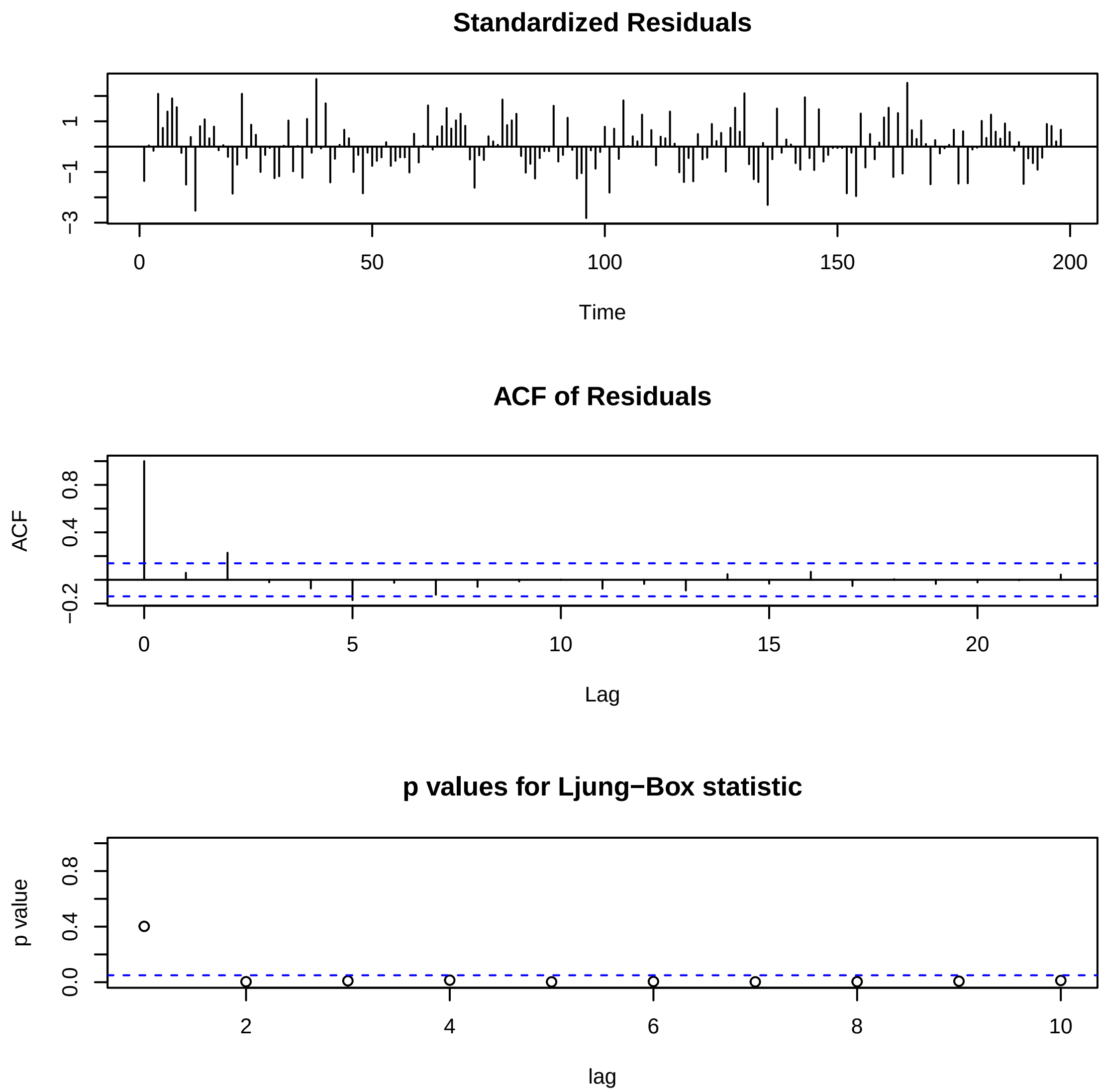
(ii) The best possible model based on the AIC value is the ARMA(1, 2) model as the AIC value is the most negative.

- ar1 = 0.68 with se 0.07
- ma1 = -0.11 with se 0.08
- ma2 = 0.42 with se 0.07
- intercept = 0.0044 with se 0.0046
- sigma^2 = 0.0002539
- aic value = -1065.98

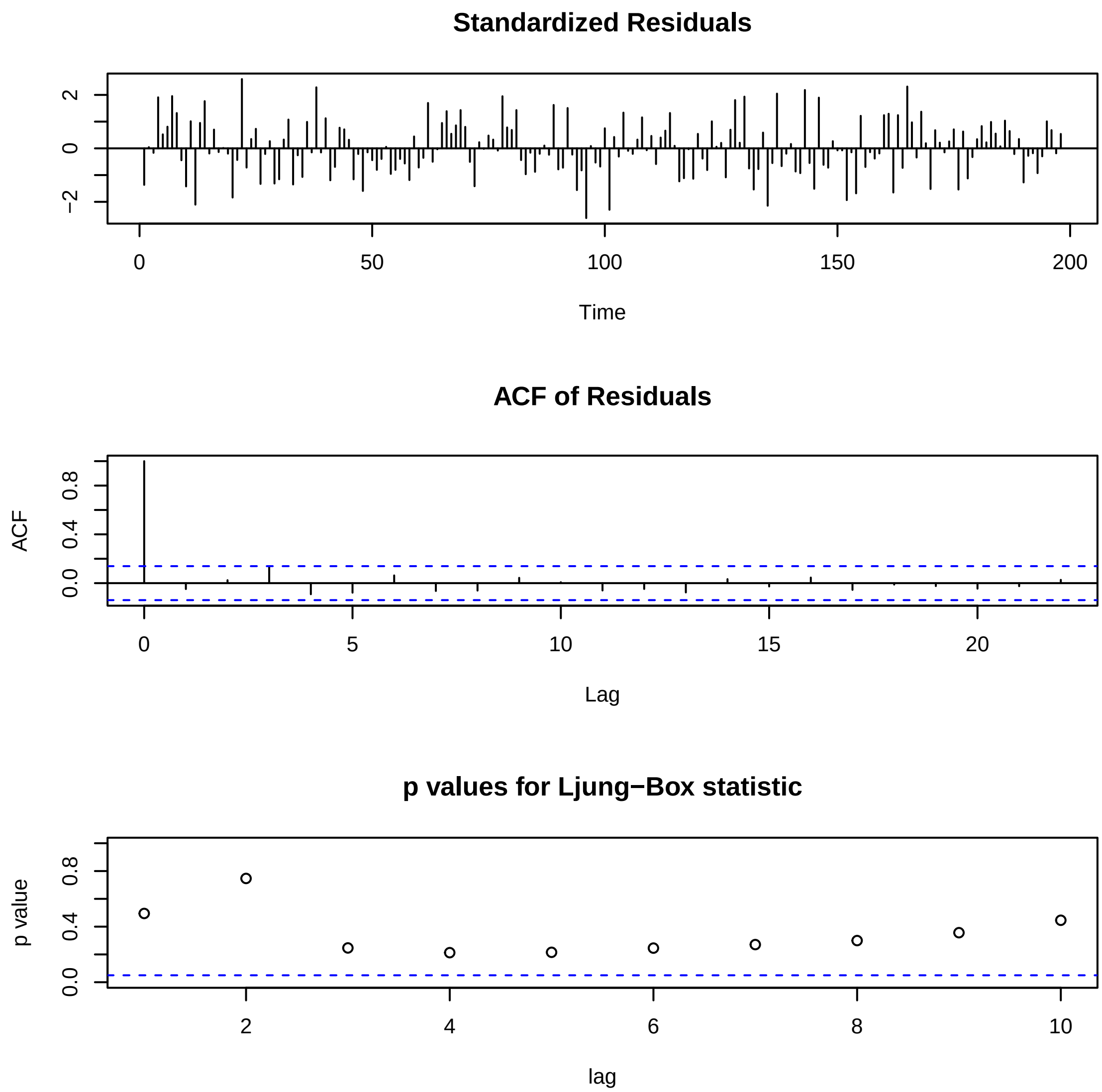
```
tsdiag(f_ar1)
```



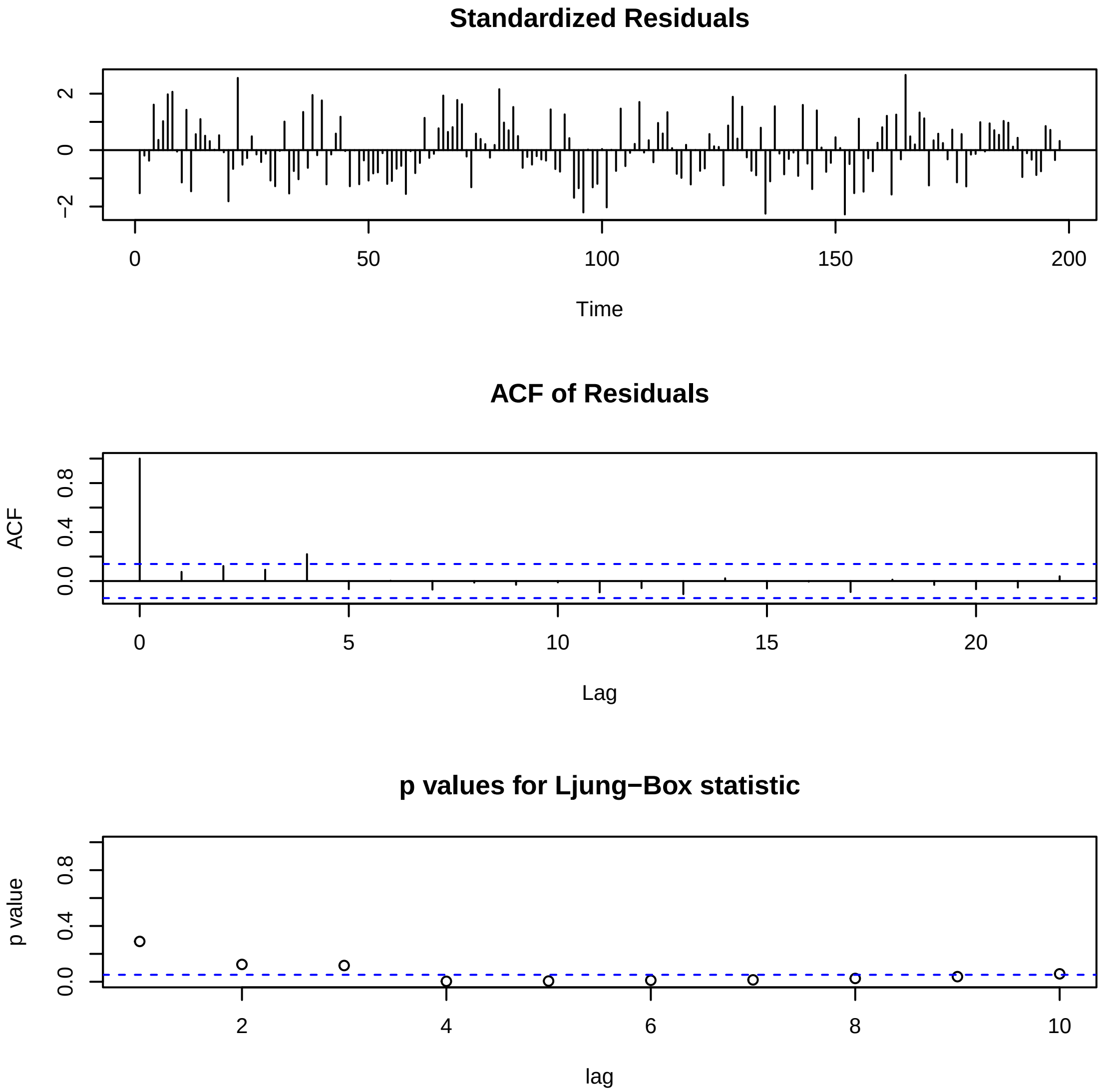

```
tsdiag(f_ar2)
```


Standardized Residuals
−3
−1
1
0
50
100
150
200
Time
ACF of Residuals
ACF
−0.2
0.4
0.8
0
5
10
15
20
Lag
p values for Ljung−Box statistic
p value
0.0
0.4
0.8
2
4
6
8
10
lag

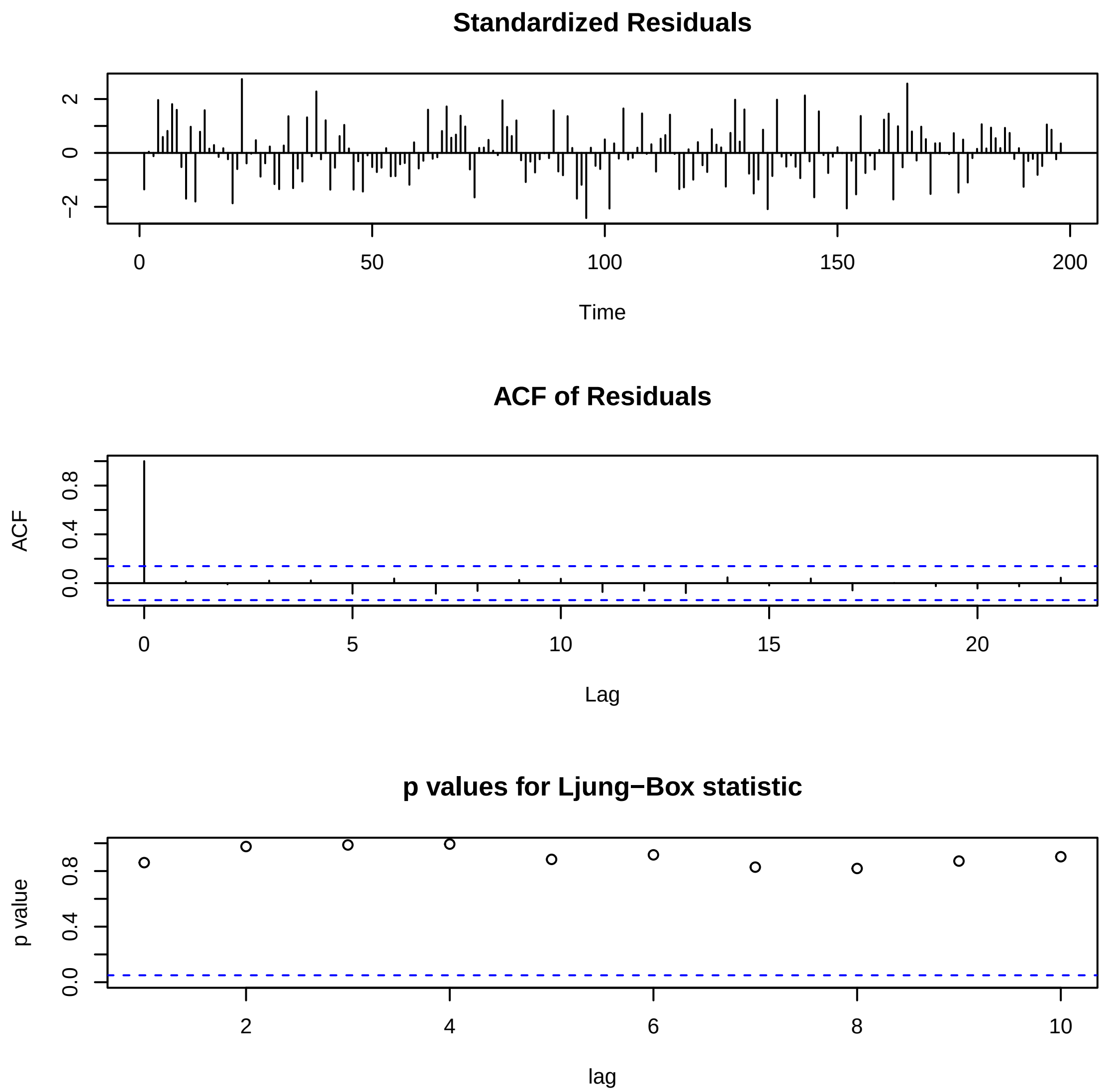
```
tsdiag(f_ar3)
```


Standardized Residuals
2
0
−2
0
50
100
150
200
Time
ACF of Residuals
ACF
0.0
0.4
0.8
0
5
10
15
20
Lag
p values for Ljung−Box statistic
p value
0.0
0.4
0.8
2
4
6
8
10
lag

```
tsdiag(f_ma3)
```




```
tsdiag(f_arma12)
```



(iii) We see that the p values remain consistently the highest (above 5% line) for the ARMA(1, 2) model and so our conclusion remains the same.

(iv)

```
my_prediction = predict(f_arma12, n.ahead = 5, se.fit = TRUE)
my_prediction
```

```
## $pred
## Time Series:
## Start = 199
## End = 203
## Frequency = 1
## [1] 0.012828029 0.012496967 0.009917557 0.008160866 0.006964485
##
## $se
## Time Series:
## Start = 199
## End = 203
## Frequency = 1
```

```
## [1] 0.01593570 0.01834138 0.02241245 0.02406806 0.02479846
```

```r
plot(199:203, my_prediction$pred, ylim = c(-0.06, 0.06), xlab = "time", ylab = "lag-2 of data")

arrows(x0 = 199:203, y0 = my_prediction$pred - my_prediction$se, x1 = 199:203, y1 = my_prediction$pred +
    my_prediction$se, code = 3, angle = 90, length = 0.05, col = "red", lwd = 3)

arrows(x0 = 199:203, y0 = my_prediction$pred - 1.96 * my_prediction$se, x1 = 199:203,
    y1 = my_prediction$pred + 1.96 * my_prediction$se, code = 3, angle = 90, length = 0.05,
    col = "blue")

legend(x = 199, y = 0.06, c("error bars", "forecast intervals"), cex = 0.8, col = c("red",
    "blue"), pch = c(1, 1))
```