# Group Coursework Submission Form

## Specialist Masters Programme

| | |
|---|---|
| **Please list all names of group members:**<br>(Surname, first name)<br>1. Bian, Yu<br>2. Huang, Ruiqi<br>3. Liu, Ching-Wei | 4. Liu, Mengjie<br>5. Wang, Zhaofei<br>6.<br>7.<br>**GROUP NUMBER:** 5 |
| **MSc in:**<br>Business analytics | |
| **Module Code:**<br>SMM634 | |
| **Module Title:**<br>Analytics Methods for Business | |
| **Lecturer:**<br>Rosalba Radice | **Submission Date:**<br>15/4/2019 |

**Declaration:**

By submitting this work, we declare that this work is entirely our own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the coursework instructions and any other relevant programme and module documentation. In submitting this work we acknowledge that we have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. We also acknowledge that this work will be subject to a variety of checks for academic misconduct.

We acknowledge that work submitted late without a granted extension will be subject to penalties, as outlined in the Programme Handbook. Penalties will be applied for a maximum of five days lateness, after which a mark of zero will be awarded.

**Marker's Comments (if not being marked on-line):**

**Deduction for Late Submission:**

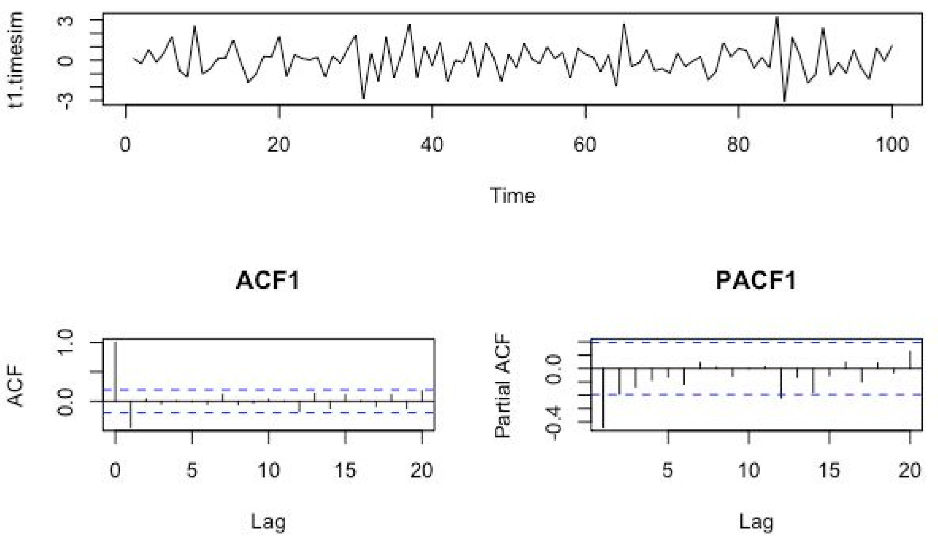
**Final Mark:** %

**Question 1**

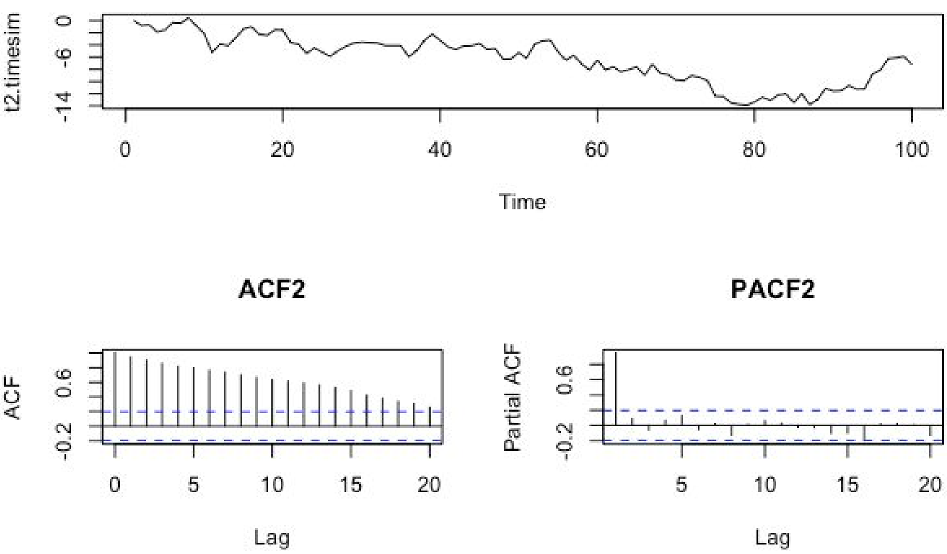
The first and the third time series are relatively stationary, because their time series plots show no obvious trend or seasonality, and their spreads are fairly even. Moreover, there are no slow decay in the ACF plots for the first and the third time series, supporting the judgment that these two time series are stationary.

On the other hand, the second and the fourth time series are obviously non-stationary. For the second time series, the plot exhibits a general decreasing trend with a sudden increase at the end. For the fourth time series, there is an obvious decrease followed by an increase, then a decrease again. Their ACF plots display slow gradual decay, confirming the non-stationarity of these two time series. A simple method to transform these two from non-stationary to stationary time series is to use the diff() function. By taking differences, the deterministic trends would be removed from the time series. The plots for the transformed second and the fourth time series show stationary characteristics, i.e. no trend or seasonality and no slow decay in ACF plots.

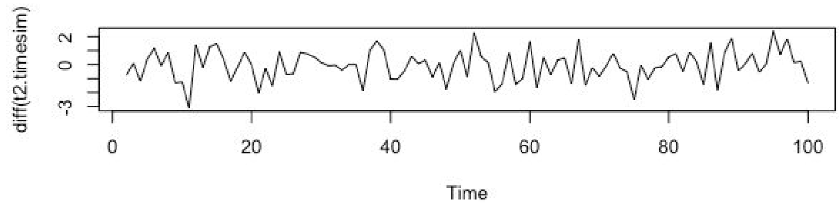
1.

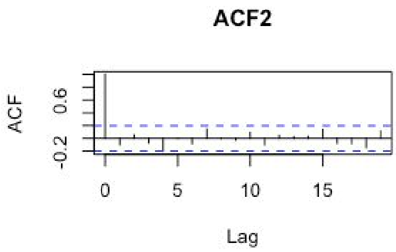


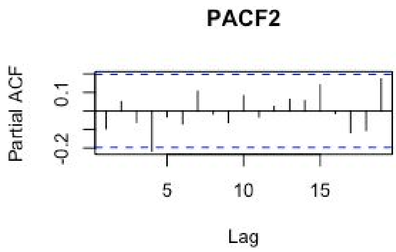

2:



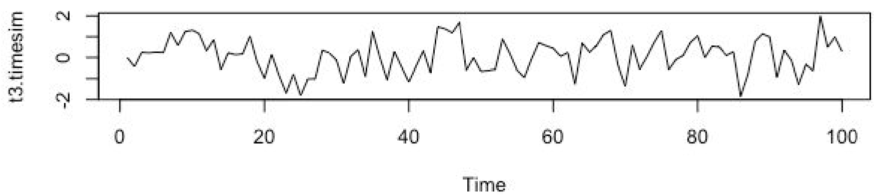
2 (differenced):

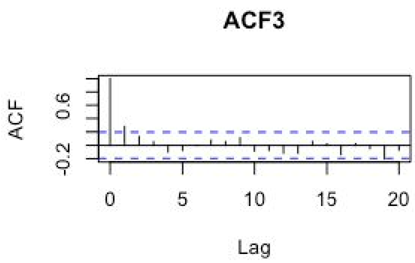


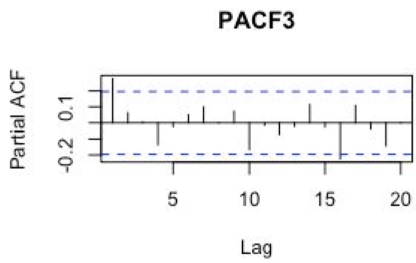



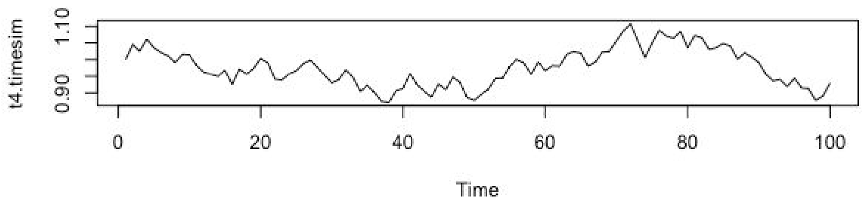

3:

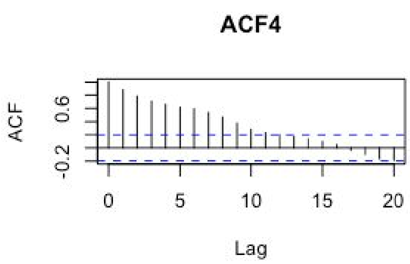


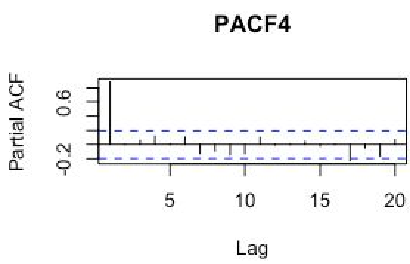



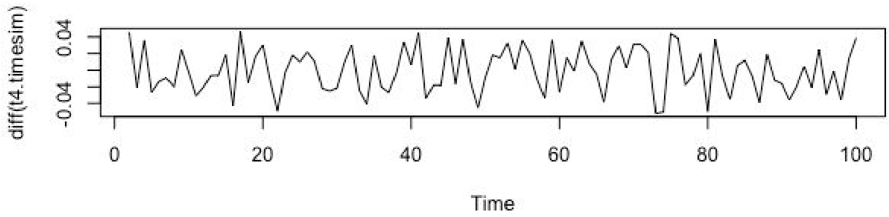

4:

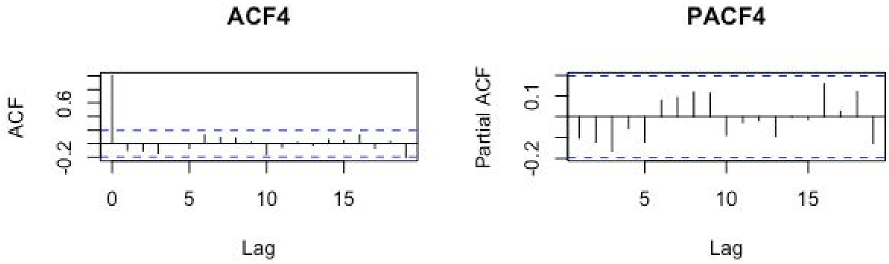


4 (differenced):

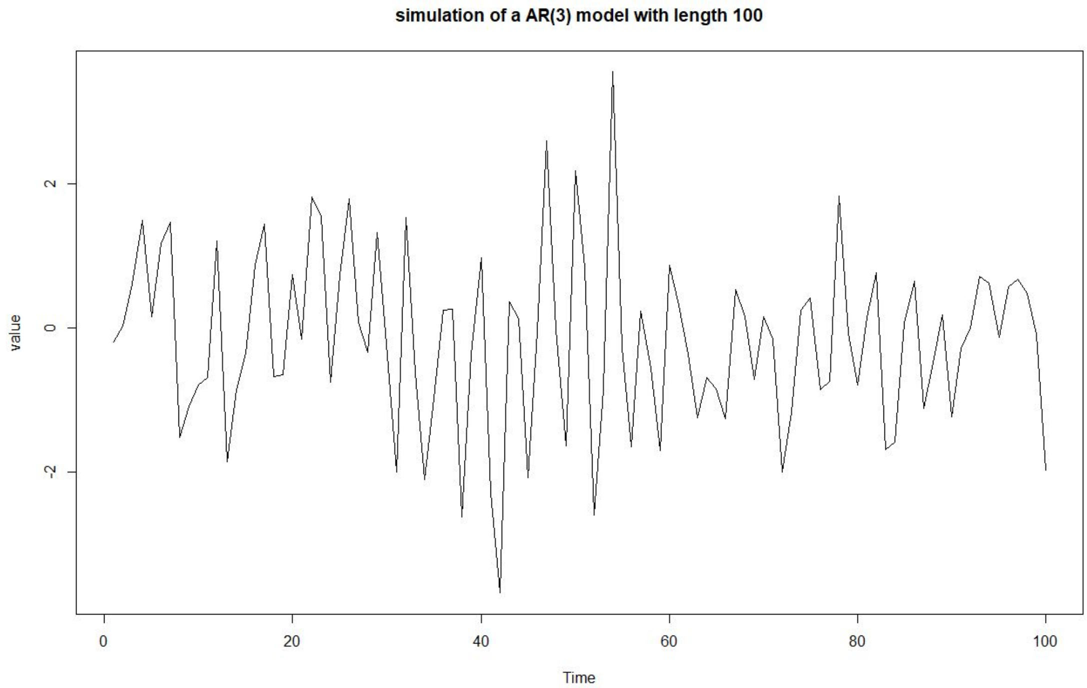



## Question 2

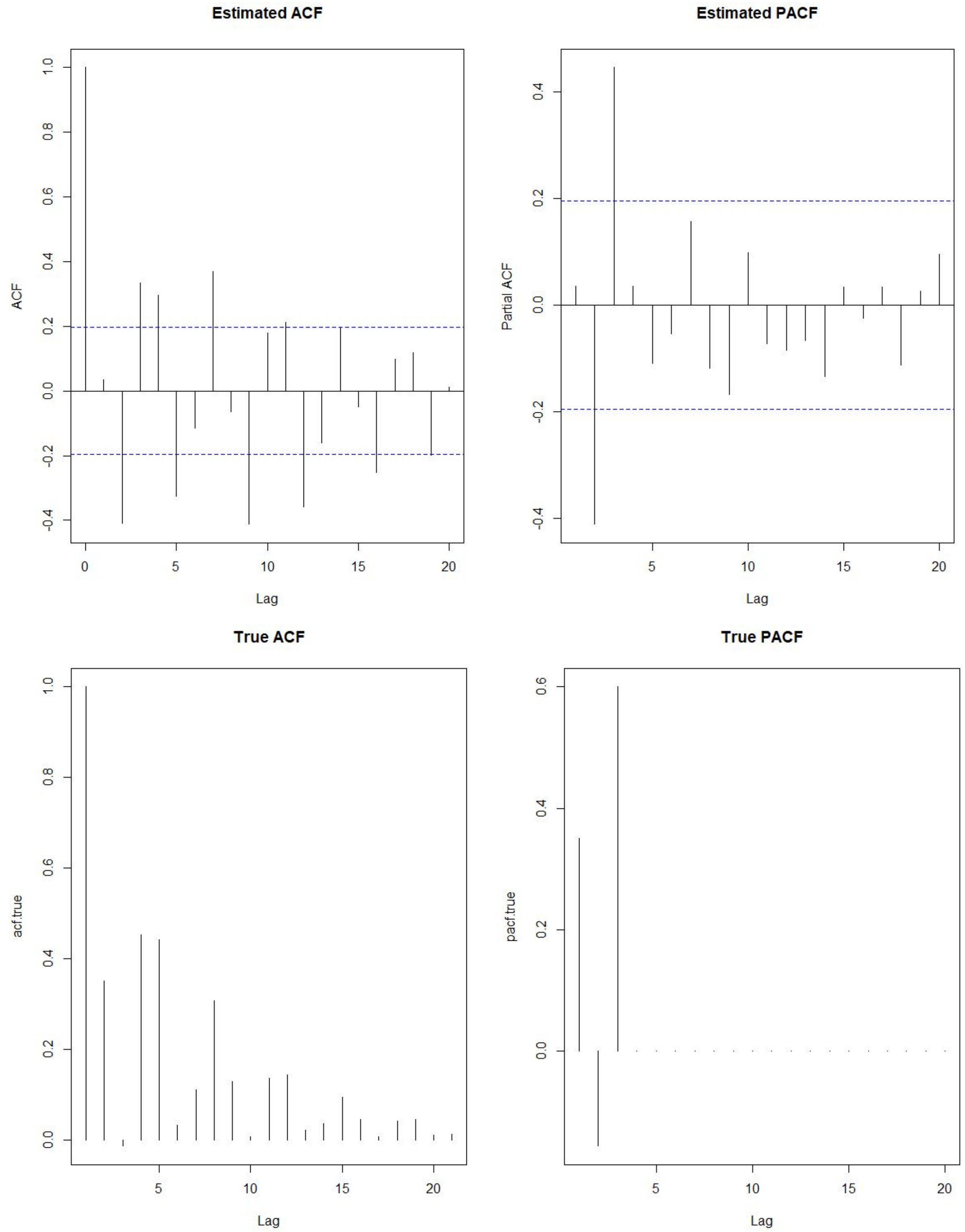
**(a)**

To simulate the AR model, we set seed at 100 and apply the arima.sim() function with coefficients ar = c(0.5, -0.4, 0.6) and length of 100. The time series plot of this simulation is shown below. As we can see in this figure, these 100 random variables are identically distributed with mean zero but the variances are not quite constant. However, there is no obvious linear trend and seasonal variation. As a result, this simulation can be said looked a little bit stationary but may need more transform such as differencing.



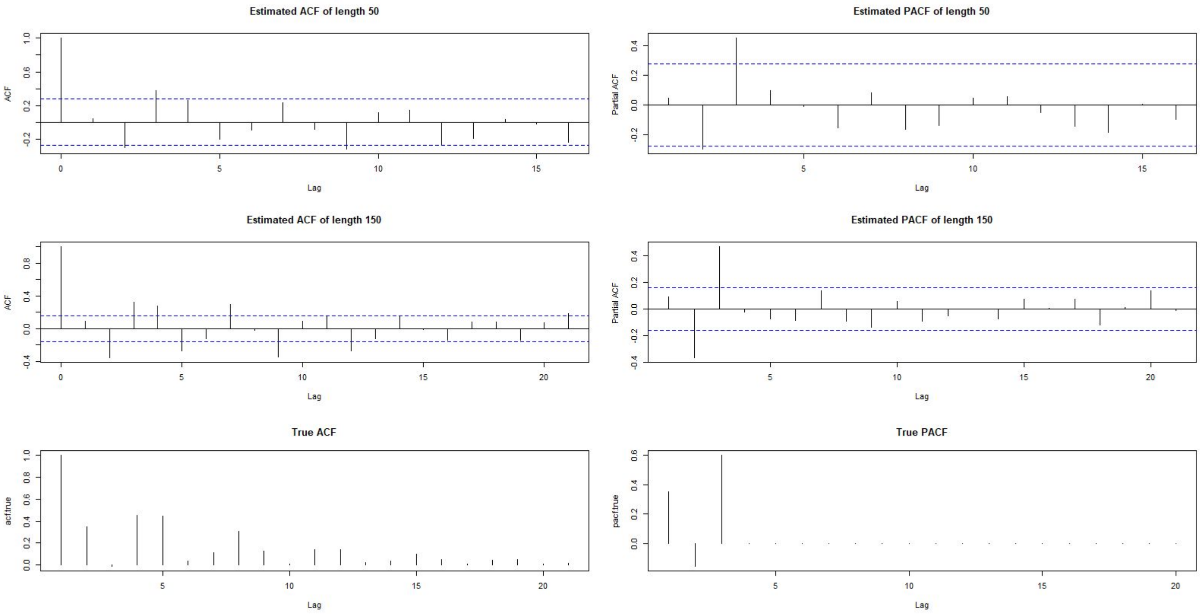

**(b)**

We observe that the estimates are not pretty accurate: the theoretical correlogram does not show a clear exponentially decaying envelope for the magnitude of the autocorrelations, and the PACF does not seem to feature some alternating behavior with a clear cut-off in absolute value as well, which indicate that this simulation is not quite stationary. Nevertheless, when looking at the true ACF and PACF of this AR(3) model, the behaviour of the plots is typical: the ACF has an exponentially decaying, whereas the PACF has a recognizable cut-off.



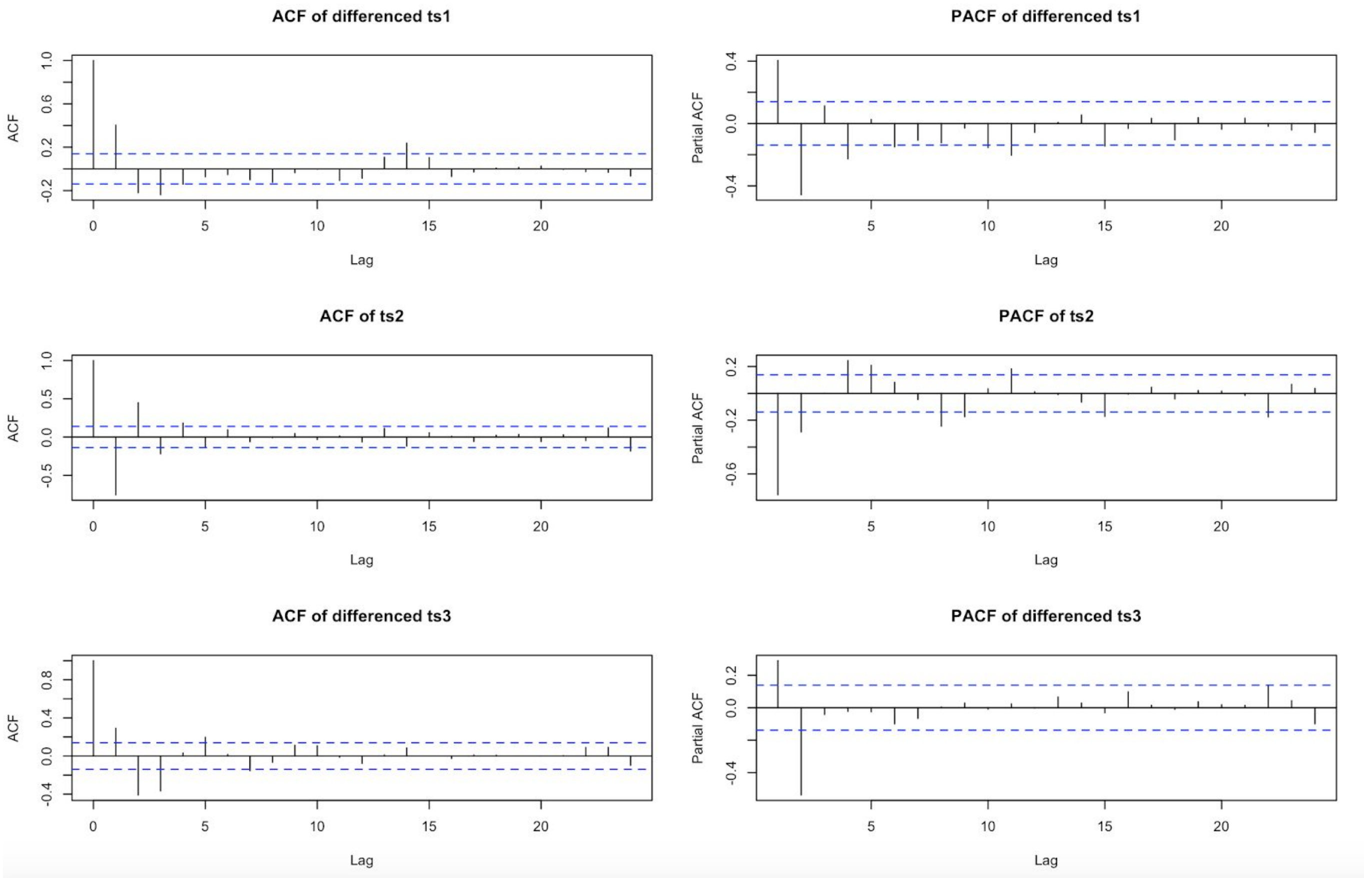

**(c)**

In simulations of this time series data of different lengths, we choose to fit an AR model of length 50 for the shorter realisation and an AR model of length 150 for the longer one. In comparison with the real one, we can observe that the structures of estimated ACF and PACF of the AR model of length 150 are more similar to those of the real model: the ACF exponentially decays, whereas the PACF has a clear cut-off. In addition, it is also noticeable that the range of the confidence interval of the longer simulation is narrower than that of the shorter one. This is because there are more observations and may contribute to more precise performances.




**Question 3**

By plotting the time series data we can find that ts1 and ts3 are not constant with obvious trend while ts2 looks stationary, so firstly we remove the trend of ts1 and ts3 by differencing them by order 1 while keeping the original ts2. Then, ACF and PACF are created for each time series data, as is shown below.

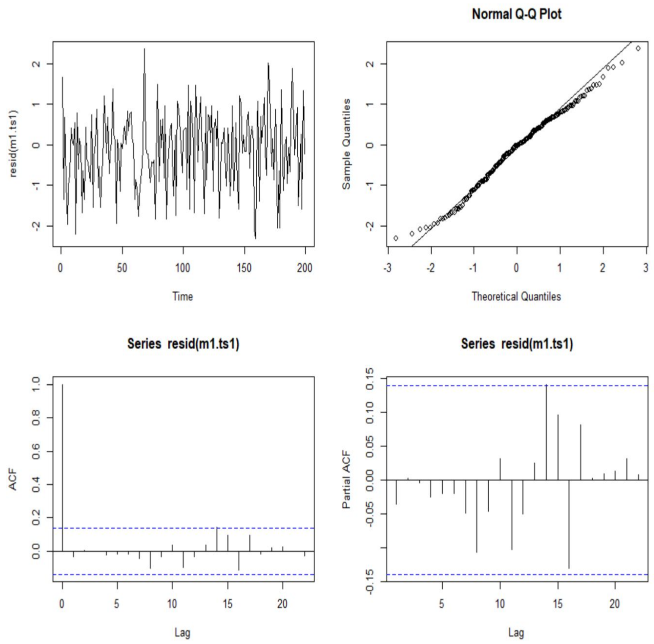



**For time series 1:**

There is a drop-off in ACF after lag 1, and in the PACF at lag 2 or 3, suggesting an ARIMA(1, 1, 3), ARIMA(3, 1, 1) or perhaps ARIMA(2, 1, 1) for ts1. For decisions on the correct order choice, we first examine the differenced ts1, which is referred to as dARIMAsim2.ts1 in the code, fit them with ARMA model and choose the one with lowest AIC. The lowest model for differenced ts1 turns out to be the order combination of ARIMA(1,0,3), which corresponds to ARIMA(1,1,3) for the original time series.

```
> m1.ts1 = arima(dARIMAsim2.ts1, order = c(1,0,3), include.mean = FALSE)
> m2.ts1 = arima(dARIMAsim2.ts1, order = c(3,0,1), include.mean = FALSE)
> m3.ts1 = arima(dARIMAsim2.ts1, order = c(2,0,1), include.mean = FALSE)
> # choose the best model with the lowest AIC--m1.ts1
> AIC(m1.ts1, m2.ts1, m3.ts1)
       df      AIC
m1.ts1  5 554.4006
m2.ts1  5 579.5689
m3.ts1  4 579.0175
```

The next step is to perform residual analysis for the chosen model, from the residual plot below shows the straight line is generally well fitted by dots in Q-Q plot, and there are no (partial) autocorrelations that exceed the confidence bounds. Therefore, ARIMA(1, 1, 3) is a suitable model for ts1. Alternatively, we can use a Ljung-Box test to formally check the serial correlation of residuals. From the result below, P-values are far higher than 0.05, which indicates no serial correlation of residuals.



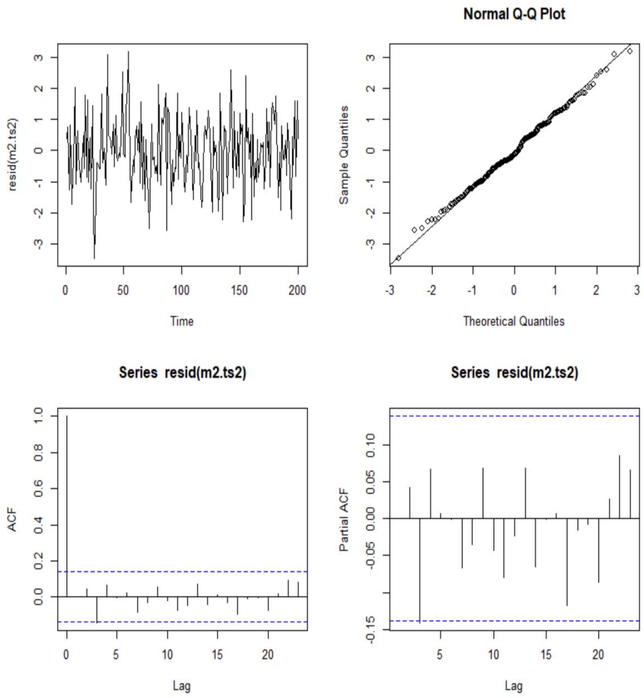

```
> Box.test(resid(m1.ts1), type = "Ljung-Box", lag = 1)

        Box-Ljung test

data:  resid(m1.ts1)
X-squared = 0.25418, df = 1, p-value = 0.6141

> Box.test(resid(m1.ts1), type = "Ljung-Box", lag = 10)

        Box-Ljung test

data:  resid(m1.ts1)
X-squared = 3.6585, df = 10, p-value = 0.9614
```

**For time series 2:**

There is a cut-off in ACF after lag 2, and in the PACF at either lag 1 or 2, suggesting an ARIMA(2, 0, 1), ARIMA(1, 0, 2) or perhaps ARIMA(2, 0, 2) for ts2. For decisions on the correct order, we choose the model with the lowest AIC, which is ARIMA(1, 0, 2) in this case. (Please see the result below.)

```
> m1.ts2 = arima(ARIMAsim2.ts2, order = c(2,0,1), include.mean = FALSE)
> m2.ts2 = arima(ARIMAsim2.ts2, order = c(1,0,2), include.mean = FALSE)
> m3.ts2 = arima(ARIMAsim2.ts2, order = c(2,0,2), include.mean = FALSE)
> ## choose the best model with the lowest AIC--m2.ts2
> AIC(m1.ts2, m2.ts2, m3.ts2)
       df      AIC
m1.ts2  4 693.1966
m2.ts2  4 637.8860
m3.ts2  5 639.3904
```

Similarly, the next step is to perform residual analysis for the chosen model, from the residual plot below shows the straight line is generally well fitted by dots in Q-Q plot, and there are no (partial) autocorrelations that exceed the confidence bounds. Therefore, ARIMA(1, 0, 2) is a suitable model for ts2, which is actually ARMA(1, 2). Alternatively, we can use a Box-Pierce test to formally check the serial correlation of residuals. From the result below, P-values are also far larger than 0.05, which indicates no serial correlation of residuals.



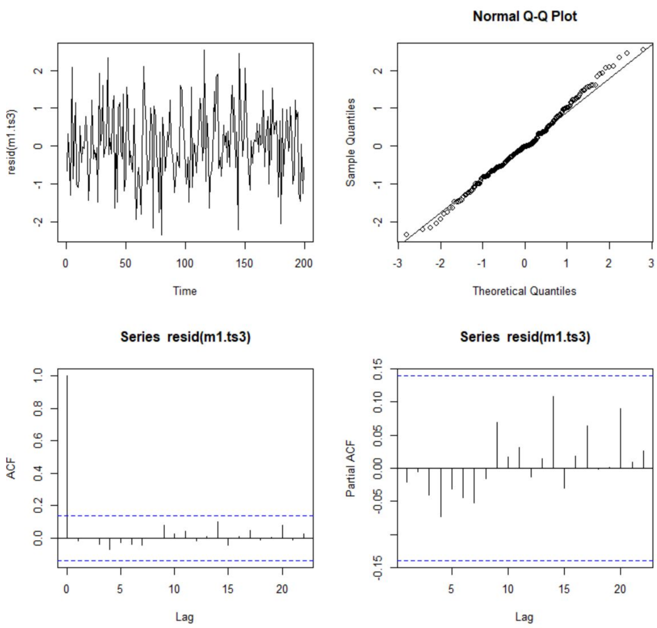

```
> Box.test(resid(m2.ts2), type = "Ljung-Box", lag = 1)

        Box-Ljung test

data:  resid(m2.ts2)
X-squared = 2.7471e-05, df = 1, p-value = 0.9958

> Box.test(resid(m2.ts2), type = "Ljung-Box", lag = 10)

        Box-Ljung test

data:  resid(m2.ts2)
X-squared = 7.8348, df = 10, p-value = 0.645
```

**For time series 3:**

There is a clear cut-off in ACF after lag 0, and in the PACF after lag 2, suggesting an ARIMA(2, 1, 0) or ARIMA(0, 1, 2) for differenced ts3. For decisions on the correct order, we choose the model with the lowest AIC, which is ARIMA(2, 0, 0) in this case, corresponding to ARIMA(2,1,0) for the original ts3. (Please see the result below.)

```
> m1.ts3 = arima(dARIMAsim2.ts3, order = c(2,0,0), include.mean = FALSE)
> m2.ts3 = arima(dARIMAsim2.ts3, order = c(0,0,2), include.mean = FALSE)
> # choose the best model with the lowest AIC--m1.ts3
> AIC(m1.ts3, m2.ts3)
       df      AIC
m1.ts3  3 555.1346
m2.ts3  3 585.2144
```

Again, the next step is to perform residual analysis for the chosen model, from the residual plot below shows the straight line is generally well fitted by dots in Q-Q plot, and there are no (partial) autocorrelations that exceed the confidence bounds. Therefore, ARIMA(2, 1, 0) is a suitable model for ts3. Alternatively, we can use a Ljung-Box test to formally check the serial correlation of residuals. From the result below,  P-values are also far larger than 0.05, which indicates no serial correlation of residuals.




```
> Box.test(resid(m1.ts3), type = "Box-Pierce", lag = 1)

        Box-Pierce test

data:  resid(m1.ts3)
X-squared = 0.081246, df = 1, p-value = 0.7756

> Box.test(resid(m1.ts3), type = "Box-Pierce", lag = 10)

        Box-Pierce test

data:  resid(m1.ts3)
X-squared = 3.5853, df = 10, p-value = 0.9641
```

**Question 4**

**(a)**

The following figures show the first six and the last six X and Y of the simulation data:

```
> head(simulation)
         X         Y
1 1.289793  1.205967
2 2.186013  4.255942
3 1.888394  2.077916
4 3.254103 17.610208
5 2.165422  4.077750
6 2.450611  5.917045
```

```
> tail(simulation)
              X          Y
995   0.5030978  0.3971961
996   1.0374104 -0.3442664
997   0.5650301  0.1700884
998   5.3044730 98.7953033
999   0.3433195  0.2654134
1000 -1.0284360 -5.5152096
```

**(b)**

The LOOCV errors of the two models, which are 199.116 for model i and 0.968 for model ii:

```
> cv.err1$delta
[1] 199.1155 199.1143
```

```
> cv.err2$delta
[1] 0.9682271 0.9682228
```

**(c)**

Our defined R function to calculate the LOOCV error:

```
CV.fun <- function(n, data, model) {
  CV <- rep(0, n)
  for (i in 1:n) {
    y_i <- data[i, 2]
    y_i.head <- model$fitted.values[i]
    h_ii <- hatvalues(model)[i]
    CV[i] <- ((y_i - y_i.head) / (1 - h_ii))^2
  }
  CV.err <- sum(CV) / n
}
```

**(d)**

The results of the two models obtained from our defined function:

```
> cv.err1.fun
[1] 199.1155
```

```
> cv.err2.fun
[1] 0.9682271
```

Since the R function writing in (c) is just a more efficient formula to calculate the LOOCV errors, we expect that the results should be the same as that using cv.glm() function. Therefore, as our expectation, the two results from the two different functions are the same.