Assign3

guilinXue

2020/6/13

1 The first question

Looking at the bond information, we can get the following information(Because the bond information doesn't have the price to maturity information, so I'm assuming the maturity price is 120):

Price	times	Coupon rate	par value	maturity price
100	20	2.68%	100	120

1.1 Yield to maturity

```
> Y_ma <- function(r, p, Cs, Cp) {
+     n <- length(Cs)
+     tt <- 1:n
+     Yi <- p - sum(Cs/((1 + r)^tt)) - Cp/(1 + r)^n
+     Yi
+ }
> Cs <- rep(1.34, times = 20)
> Cp <- 120
> p <- 100
> uniroot(Y_ma, c(0, 1), p = p, Cs = Cs, Cp = Cp)
  $root
  [1] 0.02151764

  $f.root
  [1] 0.01604816

  $iter
  [1] 7

  $init.it
  [1] NA

  $estim.prec
  [1] 6.103516e-05
```

We can get the yield to the maturity is **0.02151764**.

1.2 Macauley duration and modified duration

```
> du <- function(y, cupon, period, p0) {
+     c2 <- 0
```

```
+     tc2 <- 0
+     for (n in 1:period) {
+         t = n
+         if (n < period)
+             c <- cupon else if (n == period)
+             c <- cupon + p0
+         c1 = c/(1 + y)^n
+         tc1 <- t * c1
+         c2 <- c2 + c1
+         tc2 <- tc2 + tc1
+         if (n == period) {
+             d <- tc2/(c2 * 2)
+             md <- d/(1 + y)
+         }
+     }
+     list(d = d, md = md)
+ }
> (re <- du(0.0134, 1.34, 20, 100))
  $d
  [1] 8.838122

  $md
  [1] 8.721257
```

The Macauley duration is **8.838122**.The Modified duration is **8.721257**.

1.3 Convexity

```
> td <- function(y, cupon, period, p0) {
+     c2 <- 0
+     tc2 <- 0
+     for (n in 1:period) {
+         t = n
+         if (n < period)
+             c <- cupon else if (n == period)
+             c <- cupon + p0
+         c1 = c/(1 + y)^n
+         tc1 <- t * (t + 1) * c1
+         c2 <- c2 + c1
+         tc2 <- tc2 + tc1
+         if (n == period) {
+             covex <- tc2/(c2 * ((1 + y)^2))
+             yearlycovex <- covex/4
+         }
+     }
+     list(covex = covex, yearlycovex = yearlycovex)
+ }
> (re <- td(0.0134, 1.34, 20, 100))
  $covex
  [1] 346.2317

  $yearlycovex
  [1] 86.55793

> # The market value of the debt change
```

```
> delta.r <- -0.005
> (effects <- 0.5 * re$yearlycovex * delta.r^2)
  [1] 0.001081974
```

The convexity is **346.2317**.The market value of the debt change is **0.001081974**.

2 The second question

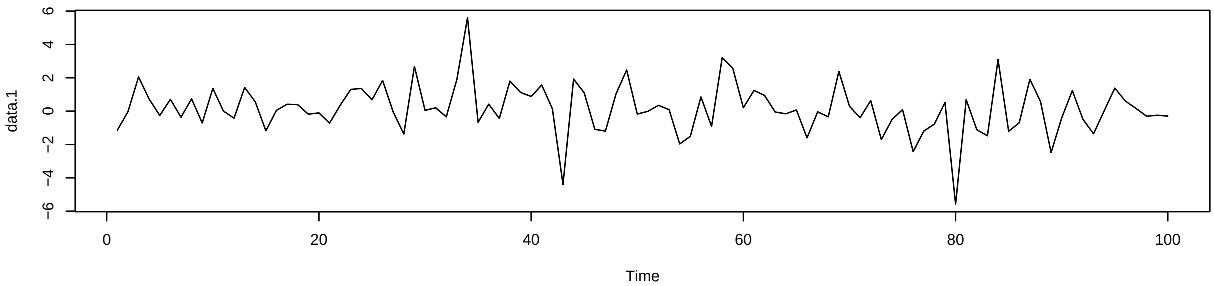
2.1 Import data and convert it into a sequence object

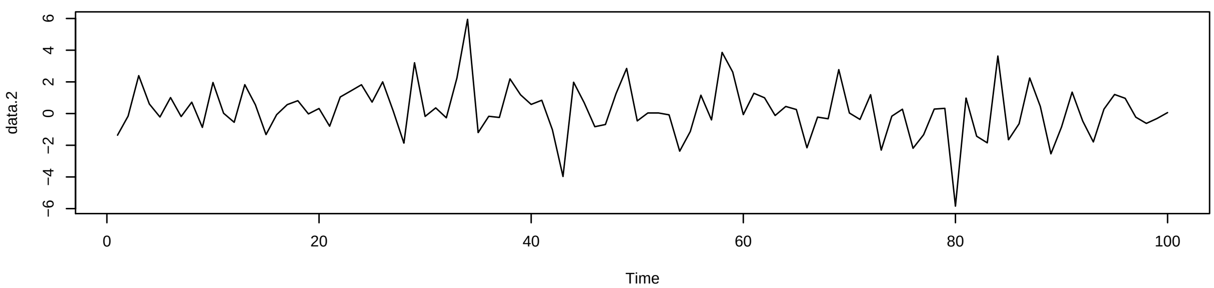
I just took the first 100 days of the index.

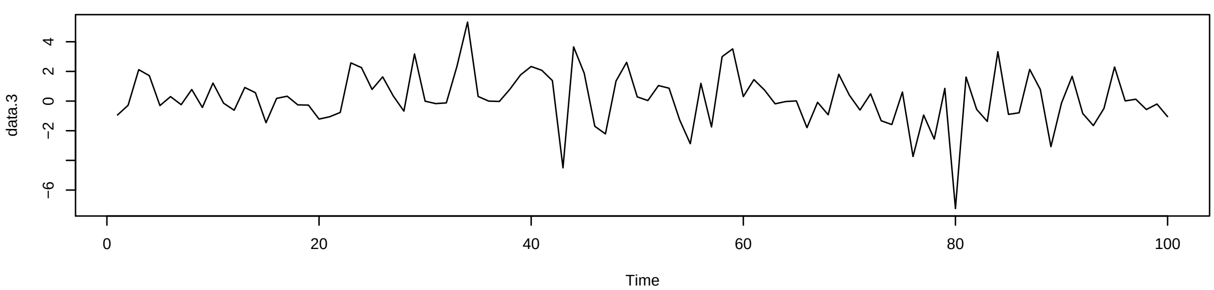
```
> library(readxl)
> library(dplyr)
> library(knitr)
> library(xts)
> library(forecast)
> library(tseries)
> library(sarima)
>
> dat <- read_excel("D:/R/R-exercise/markdown-template/IDX_Idxtrd1.xlsx") %>% as.data.frame()
>
> data <- dat[-c(1, 2), -c(3:8)]
>
> data$Idxtrd01 <- as.Date(data$Idxtrd01)
>
> data$Idxtrd08 <- as.numeric(data$Idxtrd08)
>
> data1 <- subset(data, Indexcd == "000001")
> data1 <- data1[1:100, ]
>
> data2 <- subset(data, Indexcd == "000300")
> data2 <- data2[1:100, ]
>
> data3 <- subset(data, Indexcd == "000802")
> data3 <- data3[1:100, ]
>
> data.1 <- xts(data1[, 3], as.Date(data1$Idxtrd01, format = "%Y/%m/%d")) %>% as.ts()
> data.2 <- xts(data2[, 3], as.Date(data2$Idxtrd01, format = "%Y/%m/%d")) %>% as.ts()
> data.3 <- xts(data3[, 3], as.Date(data1$Idxtrd01, format = "%Y/%m/%d")) %>% as.ts()
```

2.2 Model identification

```
> par(mfrow = c(3, 1))
> plot(data.1)
> plot(data.2)
> plot(data.3)
```

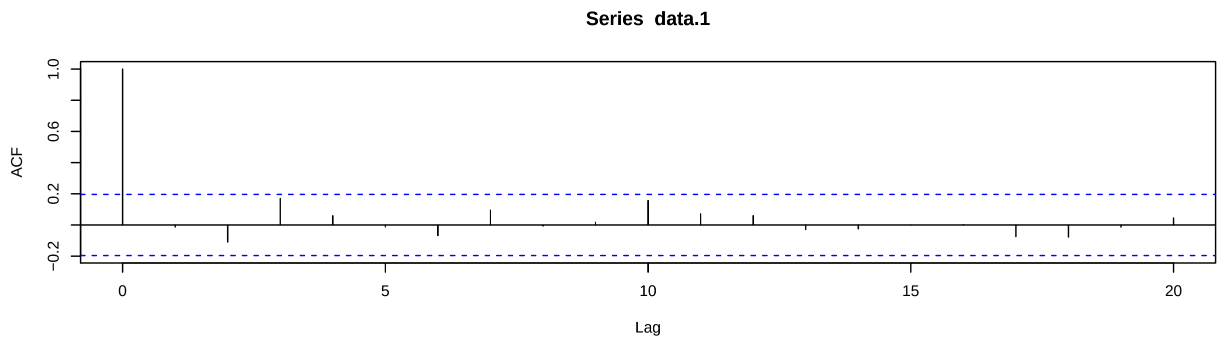


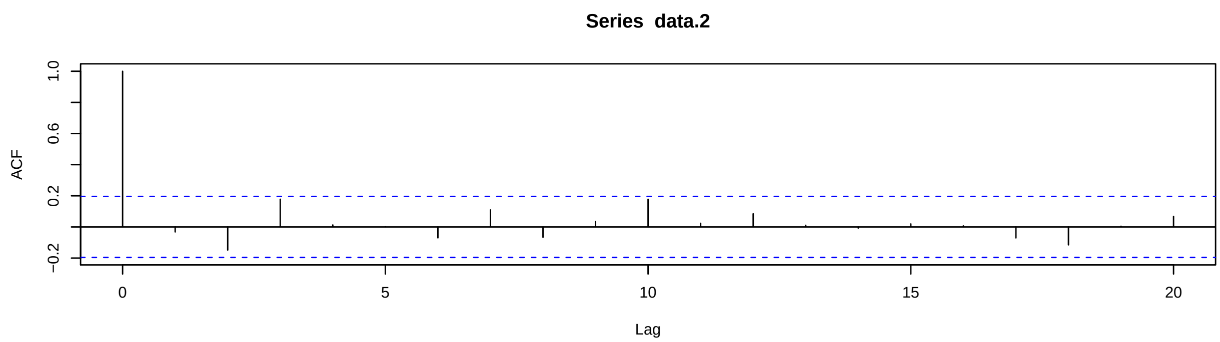


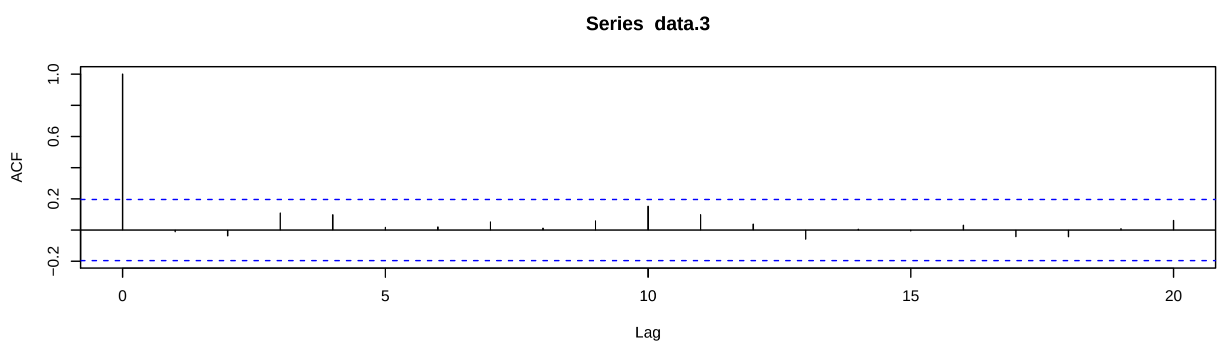



```
>
> adf.test(data.1)

	Augmented Dickey-Fuller Test

data:  data.1
Dickey-Fuller = -4.0491, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

> adf.test(data.2)

	Augmented Dickey-Fuller Test

data:  data.2
Dickey-Fuller = -4.2893, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary

> adf.test(data.3)
```

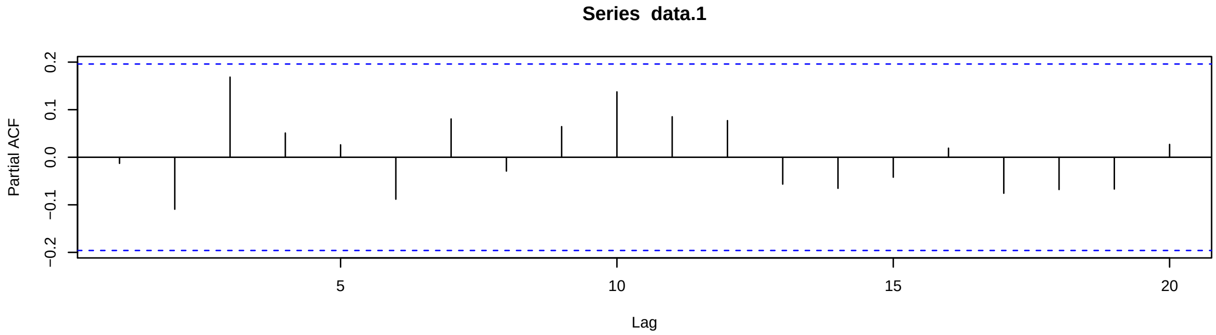
```
    Augmented Dickey-Fuller Test

  data:  data.3
  Dickey-Fuller = -3.8781, Lag order = 4, p-value = 0.01792
  alternative hypothesis: stationary

> ndiffs(data.1)
  [1] 0

>
> acf(data.1)
> acf(data.2)
> acf(data.3)
```

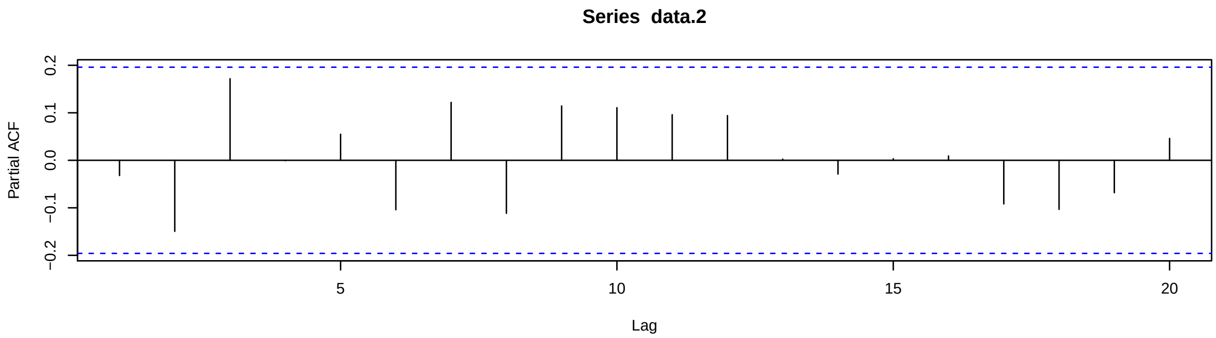


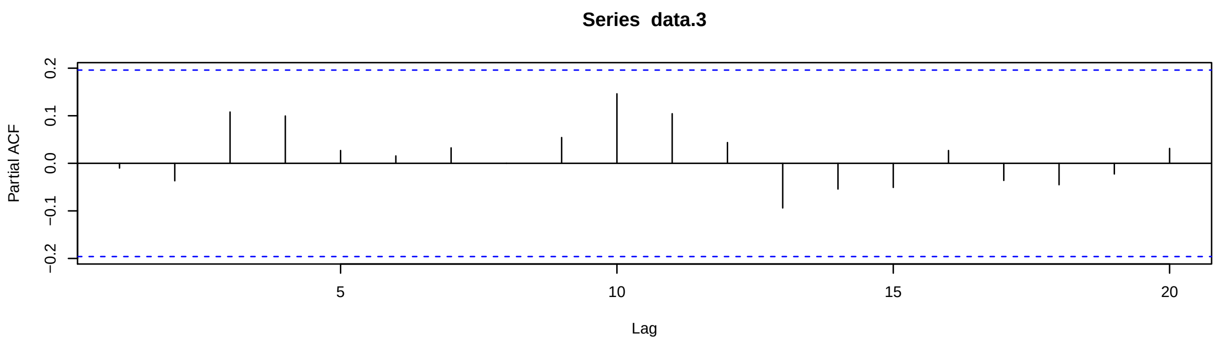





```
> pacf(data.1)
> pacf(data.2)
> pacf(data.3)
```







From the above data, the trends of the three indices are almost the same.All three indexes passed the **ADF-test**, showing that the sequence is stationary.According to their ACF index and PACF index, white noise is featured.I am difficult to judge q value and p value.Since the p+q value of ARMA model is as small as possible, I set p=q=0,and then gradually started from the low-order model to find the model with the lowest AIC value.Because the returns of the above three indexes tend to be the same, I will only use one of the indexes to build the model.

```
> am1 <- arima(data.1, order = c(0, 0, 0))
> am1

  Call:
  arima(x = data.1, order = c(0, 0, 0))

  Coefficients:
        intercept
           0.1584
  s.e.     0.1473

  sigma^2 estimated as 2.17:  log likelihood = -180.64,  aic = 365.27
```

```
> am2 <- arima(data.1, order = c(0, 0, 1))
> am2

  Call:
  arima(x = data.1, order = c(0, 0, 1))

  Coefficients:
            ma1  intercept
        -0.0163     0.1587
  s.e.   0.1118     0.1449

  sigma^2 estimated as 2.17:  log likelihood = -180.63,  aic = 367.25
> am3 <- arima(data.1, order = c(1, 0, 0))
> am3

  Call:
  arima(x = data.1, order = c(1, 0, 0))

  Coefficients:
            ar1  intercept
        -0.0129     0.1586
  s.e.   0.0999     0.1455

  sigma^2 estimated as 2.17:  log likelihood = -180.63,  aic = 367.25
> am4 <- arima(data.1, order = c(1, 0, 1))
> am4

  Call:
  arima(x = data.1, order = c(1, 0, 1))

  Coefficients:
           ar1      ma1  intercept
        0.2395  -0.2668     0.1591
  s.e.  0.9086   0.8975     0.1421

  sigma^2 estimated as 2.169:  log likelihood = -180.6,  aic = 369.19
> am5 <- arima(data.1, order = c(2, 0, 1))
> am5

  Call:
  arima(x = data.1, order = c(2, 0, 1))

  Coefficients:
            ar1      ar2     ma1  intercept
        -0.6929  -0.1335  0.6997     0.1602
  s.e.   0.2677   0.1061  0.2583     0.1351

  sigma^2 estimated as 2.103:  log likelihood = -179.1,  aic = 368.19
> am6 <- arima(data.1, order = c(1, 0, 2))
> am6

  Call:
```

```
arima(x = data.1, order = c(1, 0, 2))

Coefficients:
          ar1     ma1      ma2  intercept
      -0.5756  0.5930  -0.1215     0.1599
s.e.   0.2812  0.2795   0.0994     0.1358

sigma^2 estimated as 2.107:  log likelihood = -179.2,  aic = 368.4
```

We can get the message:

AIC(0,0,0)	AIC(0,0,1)	AIC(1,0,0)	AIC(1,0,1)	AIC(2,0,1)	AIC(1,0,2)
365.27	367.25	367.25	369.19	368.19	368.4

The smallest AIC value is 365.27.

Use **auto-arima()** to help me select the model.

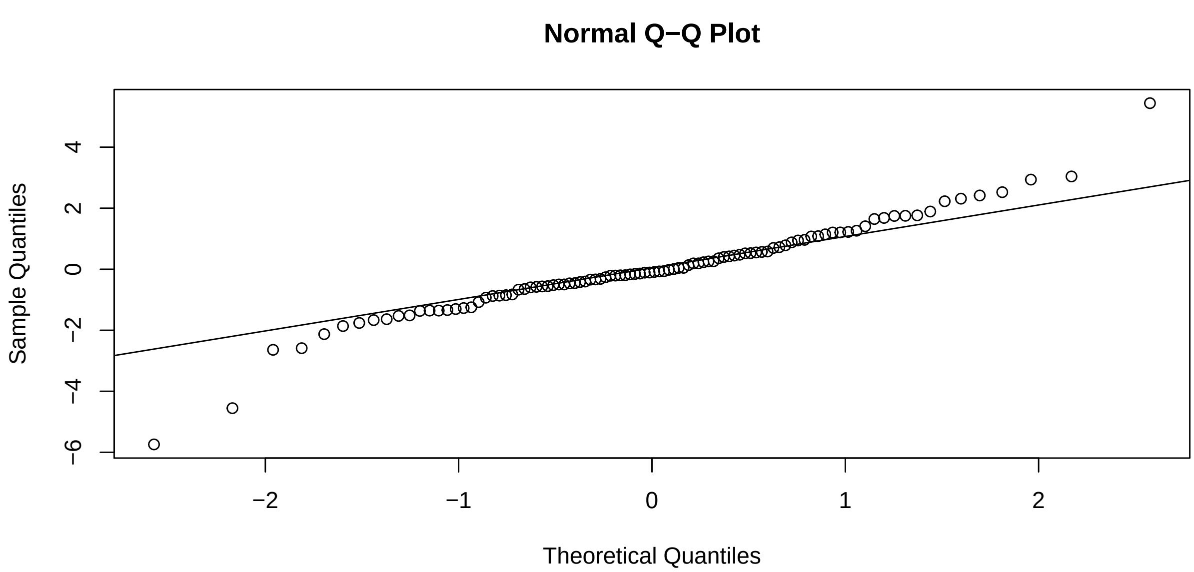
```
> am7 <- auto.arima(data.1)
> am7
  Series: data.1
  ARIMA(0,0,0) with zero mean

  sigma^2 estimated as 2.195:  log likelihood=-181.21
  AIC=364.42   AICc=364.46   BIC=367.03
```

so,I choose ARIMA(0,0,0) as the best model.

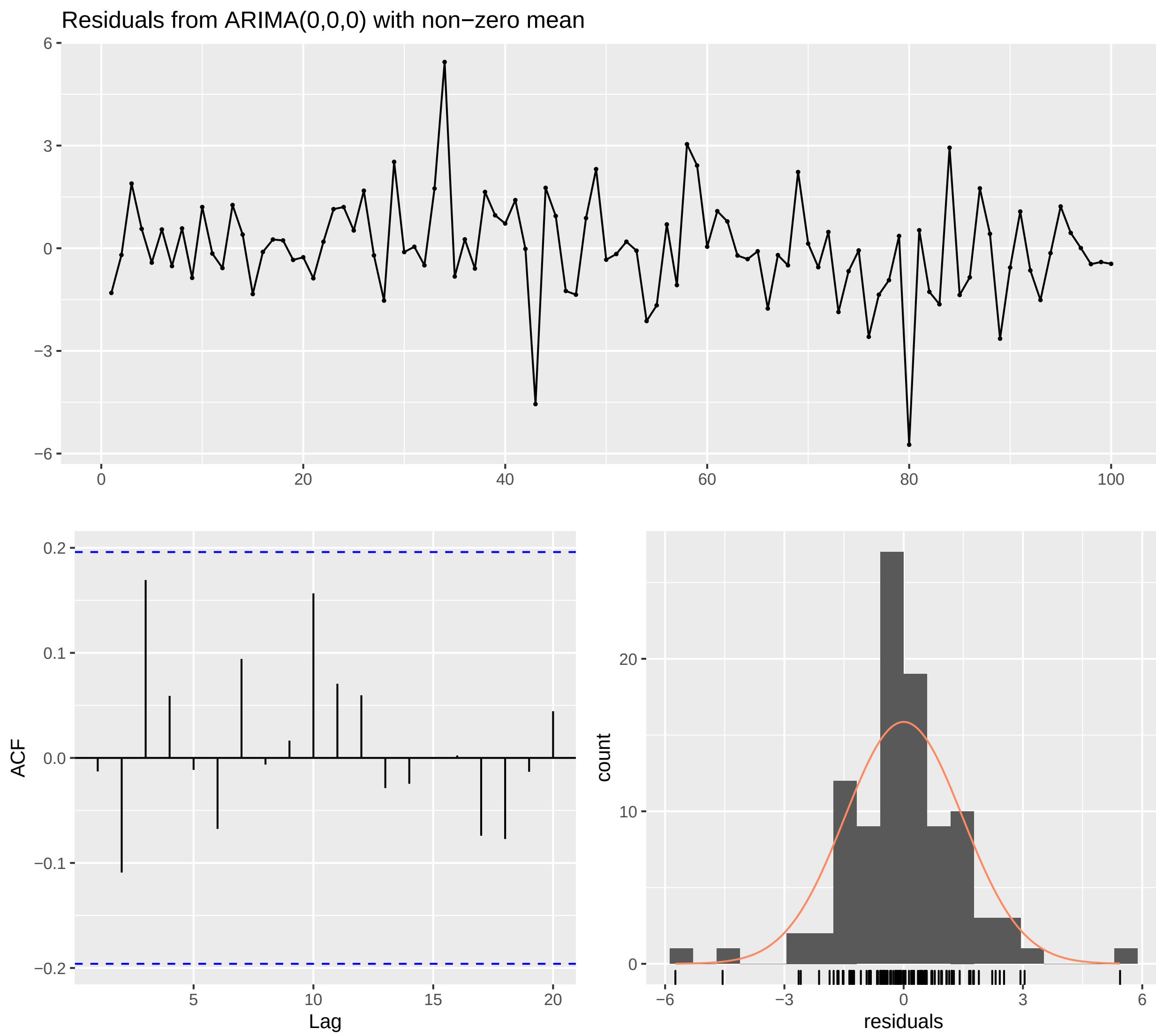
2.3 Model diagnosis

White noise test and normal distribution test are performed for residuals.

```
> par(mfrow = c(1, 1))
> qqnorm(am1$residuals)
> qqline(am1$residuals)
```



```
> shapiro.test(am1$residuals)

	Shapiro-Wilk normality test

data:  am1$residuals
W = 0.94776, p-value = 0.0005923

> checkresiduals(am1)
```




```
	Ljung-Box test

data:  Residuals from ARIMA(0,0,0) with non-zero mean
Q* = 8.9425, df = 9, p-value = 0.4426

Model df: 1.   Total lags used: 10
```

The results show that the residual is white noise, but the residual is not normally distributed.

2.4 Model forecast

```
> autoplot(forecast(am1, 5))
```

This time series is white noise. Because ARIMA(0,0,0) is a constant, the predicted value will also be a constant.