Homework-5 groupwork

Leo, Jack and Yefei Cui

2022-05-24

First include basic settings and related packages.

```
> library(tidyverse)
> library(skimr)
> library(MASS)
> library(ISLR)
> library(fpp2)
> library(forecast)
> library(tseries)
> library(Metrics)
> library(ggplot2)
```

0.Read dataset 'monthly-sunspots' in the work directory and transform it into time series set

```
> sunspots <- read.csv("monthly-sunspots.csv")
> sunspots <- ts(sunspots$Sunspots,frequency = 12,start = c(1749,1), end = c(1983,12))
> train_sunspots <- window(sunspots,start = c(1749,1), end = c(1937,12))
> test_sunspots <- window(sunspots,start = c(1938,1), end = c(1983,12))
```

1.Conduct some EDA

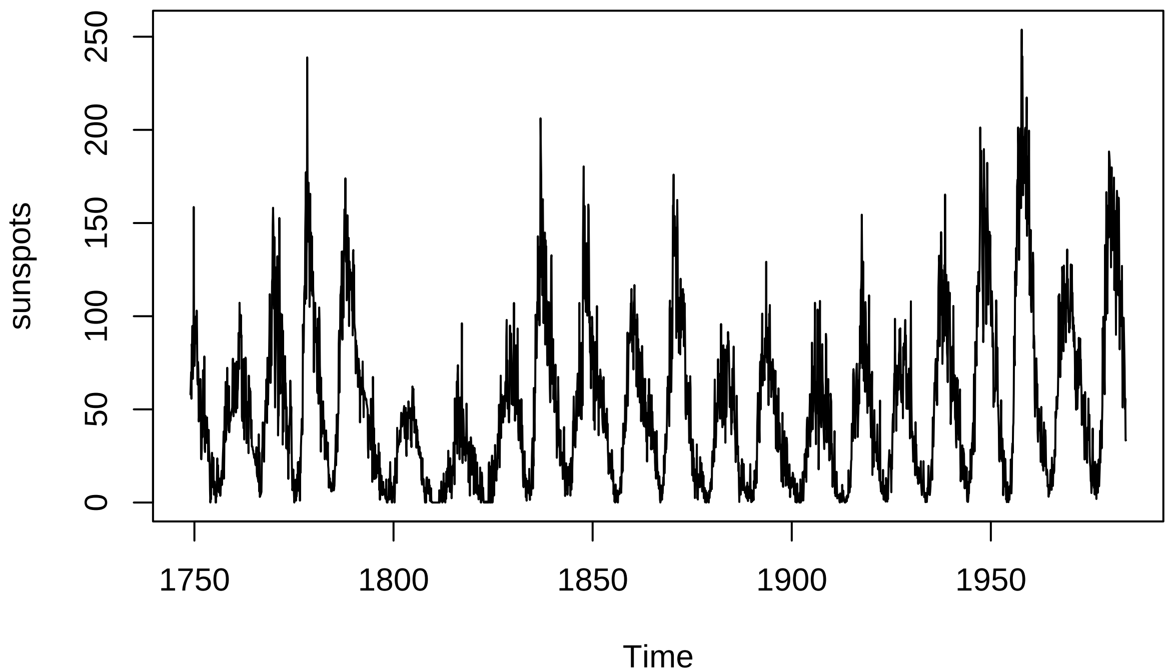
```
> # EDA with whole data
> head(sunspots)
```

```
##       Jan  Feb  Mar  Apr  May  Jun
## 1749 58.0 62.6 70.0 55.7 85.0 83.5
```

```
> summary(sunspots)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   15.70   42.00   51.27   74.92  253.80
```

```
> plot.ts(sunspots)
```




```
> # EDA with training part and testing part to predict
> predict_meanf <- meanf(train_sunspots,h = 12)
> rmse(test_sunspots,predict_meanf$mean)
```

```
## [1] 67.90384
```

```
> predict_naive <- naive(train_sunspots,h = 12)
> rmse(predict_naive$mean,test_sunspots)
```

```
## [1] 29.64999
```

```
> predict_snaive <- snaive(train_sunspots,h = 12)
> rmse(predict_snaive$mean,test_sunspots)
```

```
## [1] 23.26697
```

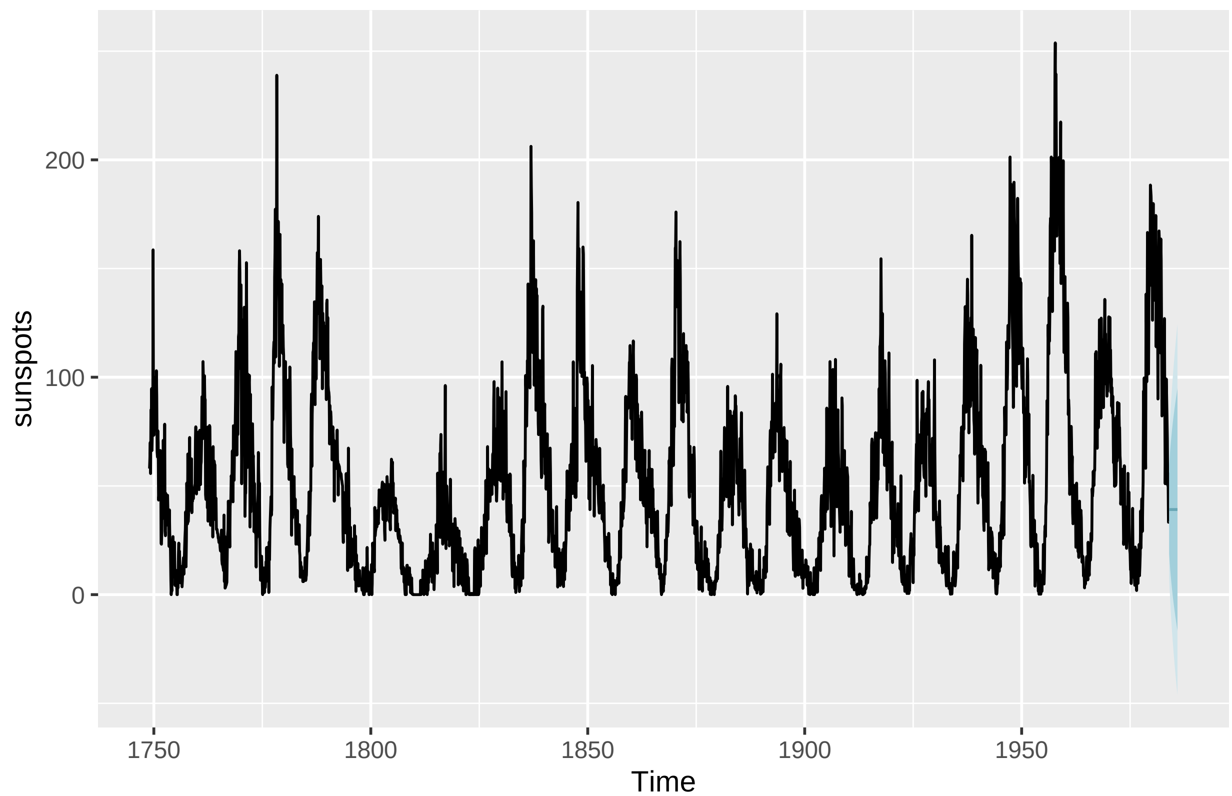
```
> predict_rwf <- rwf(train_sunspots,h = 12)
> rmse(predict_rwf$mean,test_sunspots)
```

```
## [1] 29.64999
```

```
> predict_ses <- ses(train_sunspots,h = 12)
> rmse(predict_ses$mean,test_sunspots)
```

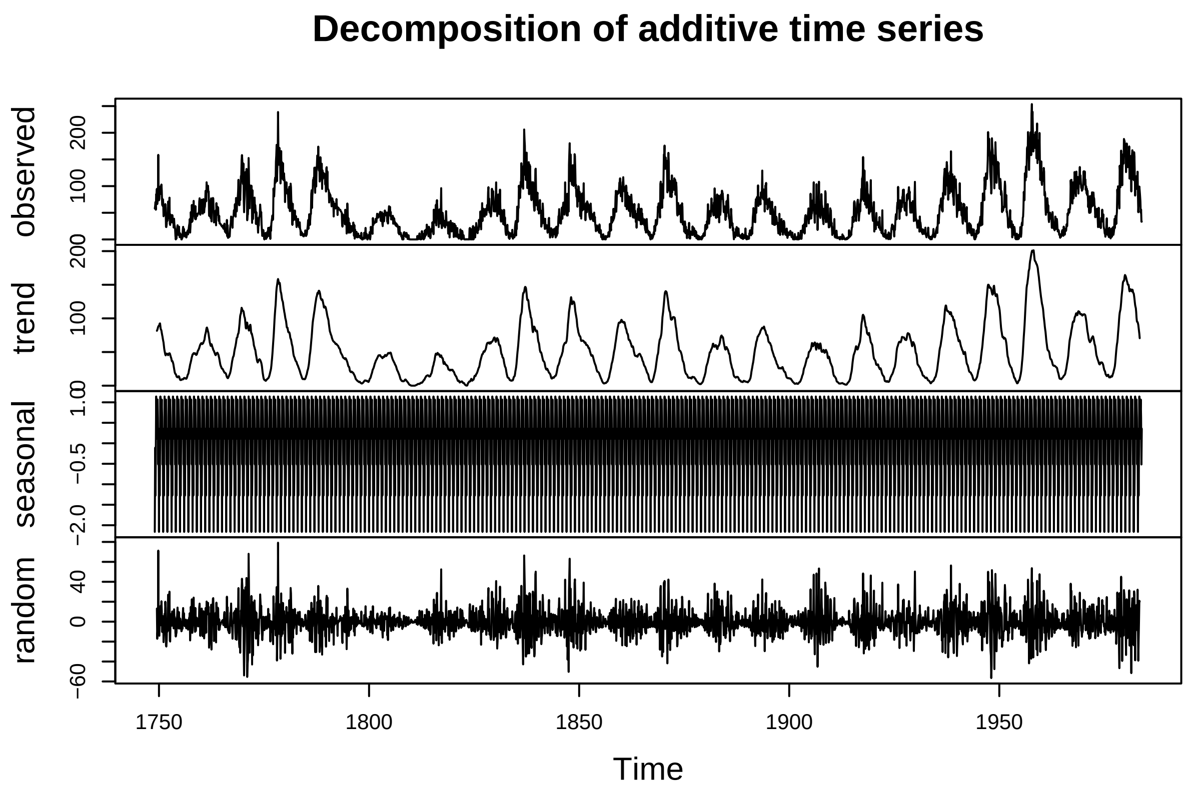
```
## [1] 26.73065
```

```
> #EDA with ggplot2
> autoplot(sunspots) + geom_forecast(colour = "lightblue")
```



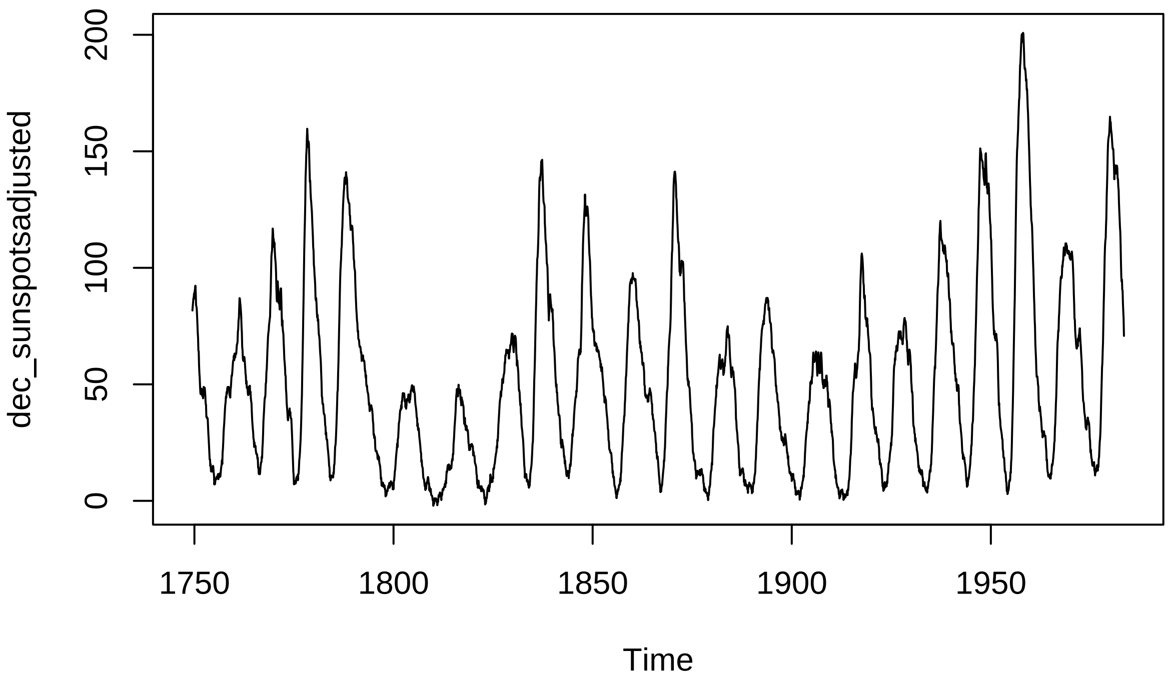
2.Decompose the data

```
> dec_sunspots <- decompose(sunspots)
> Res <- dec_sunspots$Sunspots
> plot(dec_sunspots)
```



We can observe from the plot that the data has a strong seasonal pattern. But we think the random noise is too big so we try to adjust our plot.

```
> dec_sunspotsadjusted <- sunspots - dec_sunspots$random
> plot(dec_sunspotsadjusted)
```



As is shown, the plot is much smoother without random noise. But we can still find some jumps in several periods.

3.Test data

```
> adf.test(sunspots)
```

```
## Warning in adf.test(sunspots): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  sunspots
## Dickey-Fuller = -6.494, Lag order = 14, p-value = 0.01
## alternative hypothesis: stationary
```

We can see that p value is smaller than 0.01, which means our data is very stationary.

```
> ndiffs(sunspots)
```

```
## [1] 1
```

Because k > 0 and k = 1, so we need to use first difference to conduct a stantionary data.

```
> adf.test(diff(sunspots))
```

```
## Warning in adf.test(diff(sunspots)): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(sunspots)
## Dickey-Fuller = -11.273, Lag order = 14, p-value = 0.01
## alternative hypothesis: stationary
```

Actually both initial data and first difference data are stationary enough.

4.Fit ARMA models

First fit initial data

```
> arma_model1 <- arima(sunspots, order = c(1, 0, 1))
> arma_model1
```

```
##
## Call:
## arima(x = sunspots, order = c(1, 0, 1))
##
## Coefficients:
##          ar1      ma1  intercept
##       0.9780  -0.4493    51.2641
## s.e.  0.0042   0.0205     7.3471
##
## sigma^2 estimated as 251.3:  log likelihood = -11795.07,  aic = 23598.13
```

```
> lmtest::coeftest(arma_model1)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1        0.9779630  0.0041775 234.1009 < 2.2e-16 ***
## ma1       -0.4493071  0.0205445 -21.8700 < 2.2e-16 ***
## intercept 51.2640537  7.3471496   6.9774 3.007e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> arma_model2 <- arima(sunspots, order = c(4, 0, 3))
> arma_model2
```

```
##
## Call:
## arima(x = sunspots, order = c(4, 0, 3))
##
## Coefficients:
##          ar1     ar2     ar3      ar4     ma1     ma2      ma3  intercept
##      -0.1603  0.4539  0.8071  -0.1469  0.7438  0.1082  -0.5548    51.0454
## s.e.   0.0588  0.0482  0.0438   0.0391  0.0540  0.0442   0.0404     8.1500
##
## sigma^2 estimated as 246.8:  log likelihood = -11769.75,  aic = 23557.5
```

```
> lmtest::coeftest(arma_model2)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error  z value  Pr(>|z|)
## ar1       -0.160308   0.058775  -2.7275  0.006382 **
## ar2        0.453917   0.048238   9.4099 < 2.2e-16 ***
## ar3        0.807120   0.043794  18.4300 < 2.2e-16 ***
## ar4       -0.146925   0.039092  -3.7585  0.000171 ***
## ma1        0.743814   0.053961  13.7842 < 2.2e-16 ***
## ma2        0.108191   0.044219   2.4467  0.014417 *
## ma3       -0.554777   0.040356 -13.7472 < 2.2e-16 ***
## intercept 51.045437   8.149967   6.2633  3.77e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> arma_model3 <- arima(sunspots, order = c(2, 0, 2))
> arma_model3
```

```
##
## Call:
## arima(x = sunspots, order = c(2, 0, 2))
##
## Coefficients:
```

```
##          ar1     ar2      ma1      ma2  intercept
##       0.9960  -0.013  -0.4134  -0.1074    51.2325
## s.e.  0.1214   0.119   0.1201   0.0576     8.2069
##
## sigma^2 estimated as 247.9:  log likelihood = -11775.69,  aic = 23563.38
```

```
> lmtest::coeftest(arma_model3)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1        0.996045   0.121413  8.2038 2.330e-16 ***
## ar2       -0.013023   0.118963 -0.1095 0.9128270
## ma1       -0.413434   0.120113 -3.4420 0.0005773 ***
## ma2       -0.107372   0.057614 -1.8636 0.0623711 .
## intercept 51.232491   8.206892  6.2426 4.303e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that concerning AIC(If AIC is smaller, it's better), (4,0,3) is the best type of selected models

Then try to fit first difference data

```
> sunspots %>%
+   diff() %>%
+   arima(order = c(1,1,1)) %>%
+   lmtest::coeftest()
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.3005355  0.0179684  -16.726 < 2.2e-16 ***
## ma1 -1.0000000  0.0011401 -877.096 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> sunspots %>%
+   diff() %>%
+   arima(order = c(4,1,3)) %>%
+   lmtest::coeftest()
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.063932   1.453363 -0.0440   0.9649
## ar2 -0.083797   0.227283 -0.3687   0.7124
```

```
## ar3 -0.068096   0.099566 -0.6839   0.4940
## ar4 -0.015015   0.036735 -0.4087   0.6827
## ma1 -1.344861   1.454535 -0.9246   0.3552
## ma2  0.280189   2.264534  0.1237   0.9015
## ma3  0.064673   0.811928  0.0797   0.9365
```

```
> sunspots %>%
+   diff() %>%
+   arima(order = c(2,1,2)) %>%
+   lmtest::coeftest()
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ar1  0.3642595  0.0305329  11.9301   <2e-16 ***
## ar2  0.0033003  0.0233045   0.1416   0.8874
## ma1 -1.7714705  0.0240323 -73.7120   <2e-16 ***
## ma2  0.7960034  0.0241057  33.0214   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Use auto.arima to select model

```
> sunspots %>%
+   auto.arima() %>%
+   summary()
```

```
## Series: .
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.3467  -0.3963  -1.7710  0.8103
## s.e.  0.0303   0.0287   0.0205  0.0194
##
## sigma^2 = 243.8:  log likelihood = -11745.5
## AIC=23500.99   AICc=23501.01   BIC=23530.71
##
## Training set error measures:
##                       ME     RMSE      MAE MPE MAPE      MASE         ACF1
## Training set -0.02672716 15.60055 11.02575 NaN  Inf 0.4775401 -0.01055012
```

```
> sunspots %>%
+   auto.arima() %>%
+   lmtest::coeftest()
```

```
##
## z test of coefficients:
##
```

```
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  1.346694   0.030290  44.460 < 2.2e-16 ***
## ar2 -0.396287   0.028744 -13.787 < 2.2e-16 ***
## ma1 -1.771007   0.020494 -86.415 < 2.2e-16 ***
## ma2  0.810314   0.019443  41.676 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that (2,1,2) is the best of all model types to fit in

---

5.Roles of our group

Leo: Extra EDA part and some decomposition parts.

Jack: ts.plot, EDA.

Yefei Cui: EDA and ARIMA model.

We mainly discussed online through wechat, and exchanged our thoughts about coding and some problems, and leo helped explain some definitions.