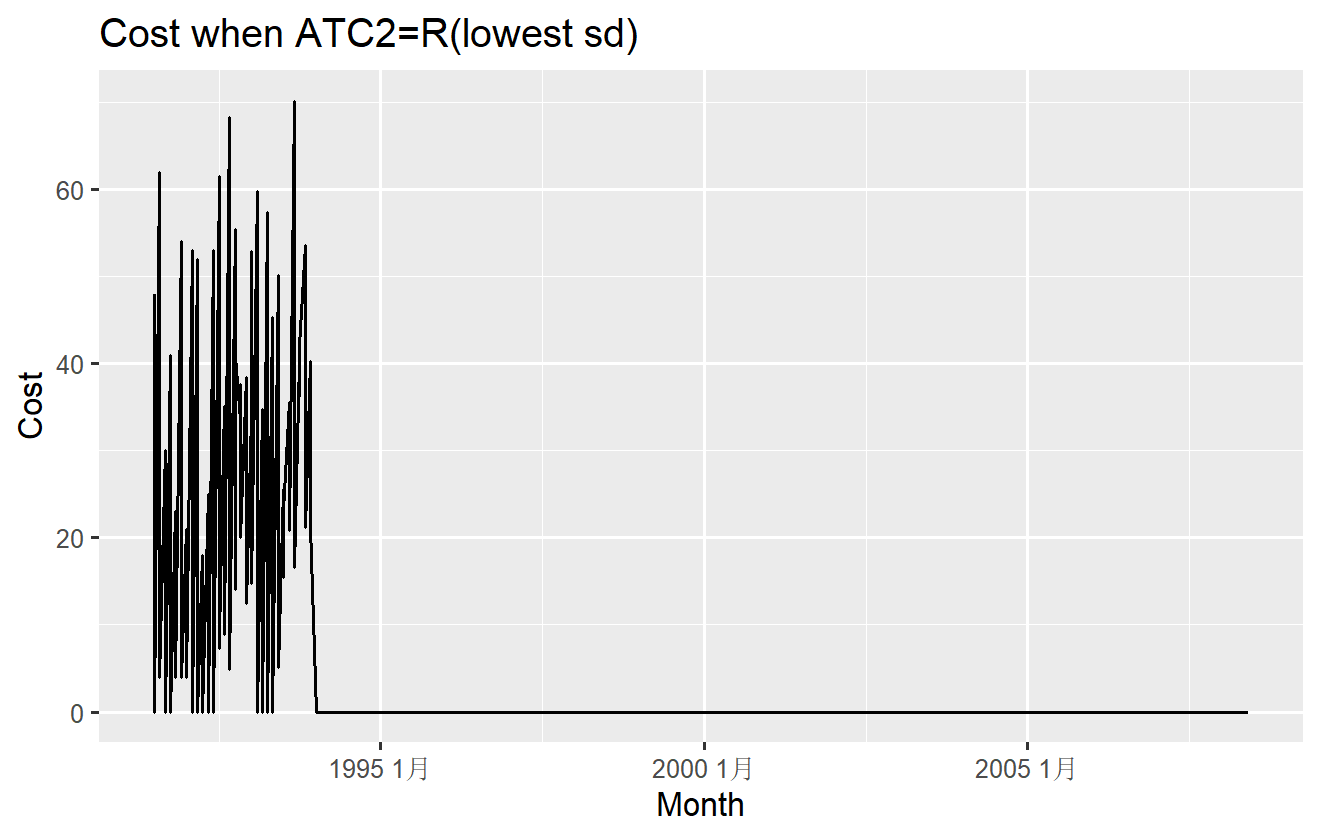
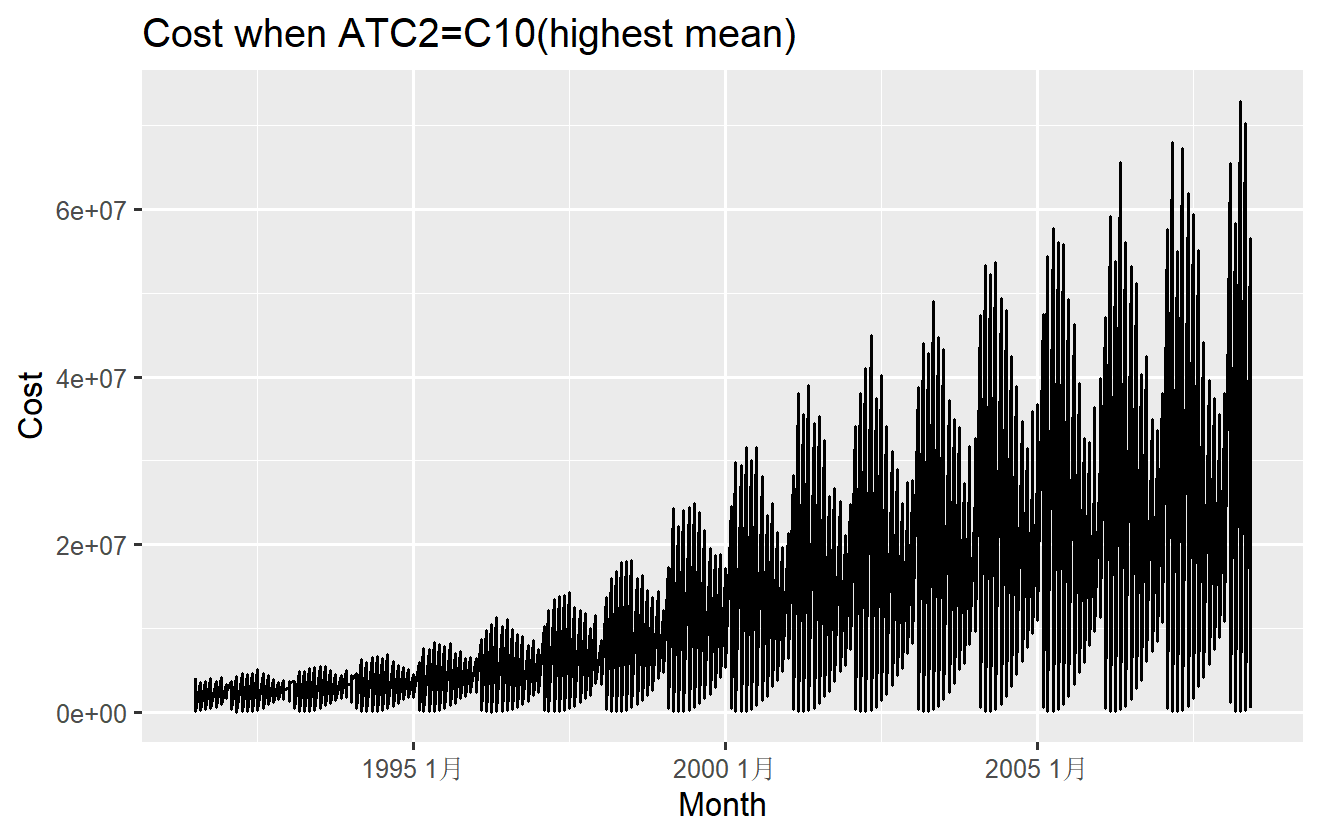
MDS5130 Assignment 2

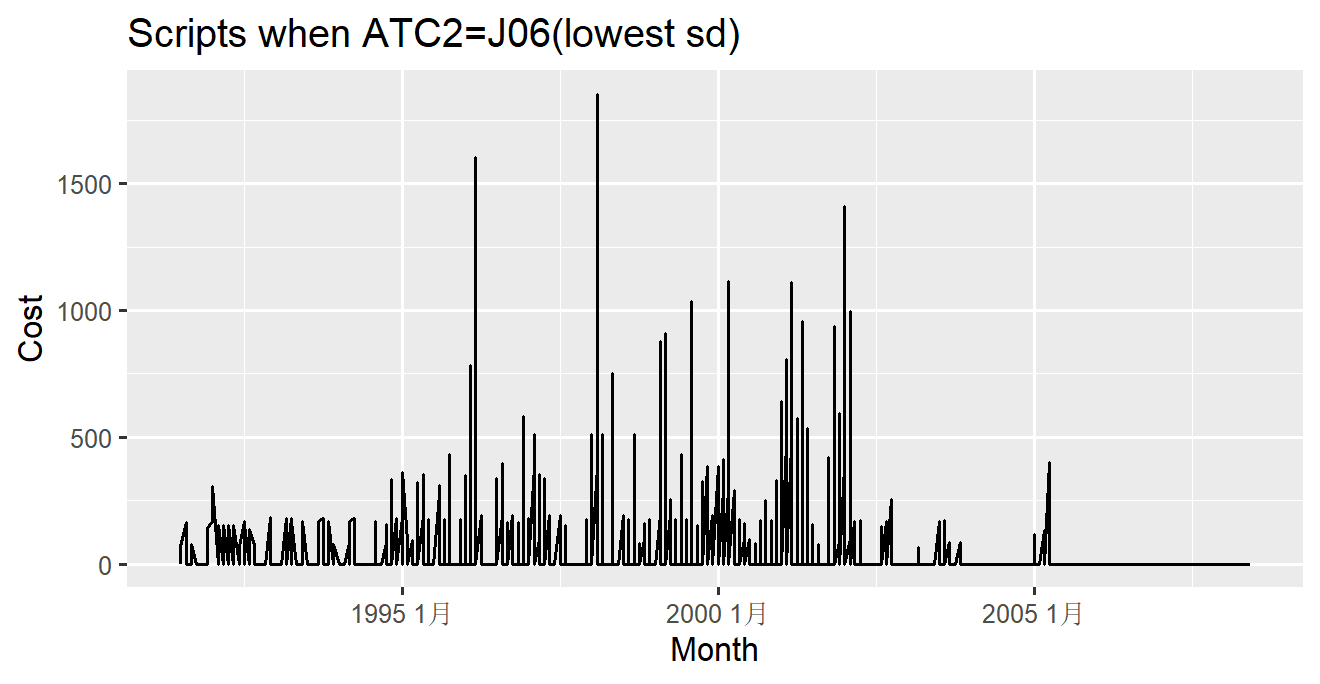
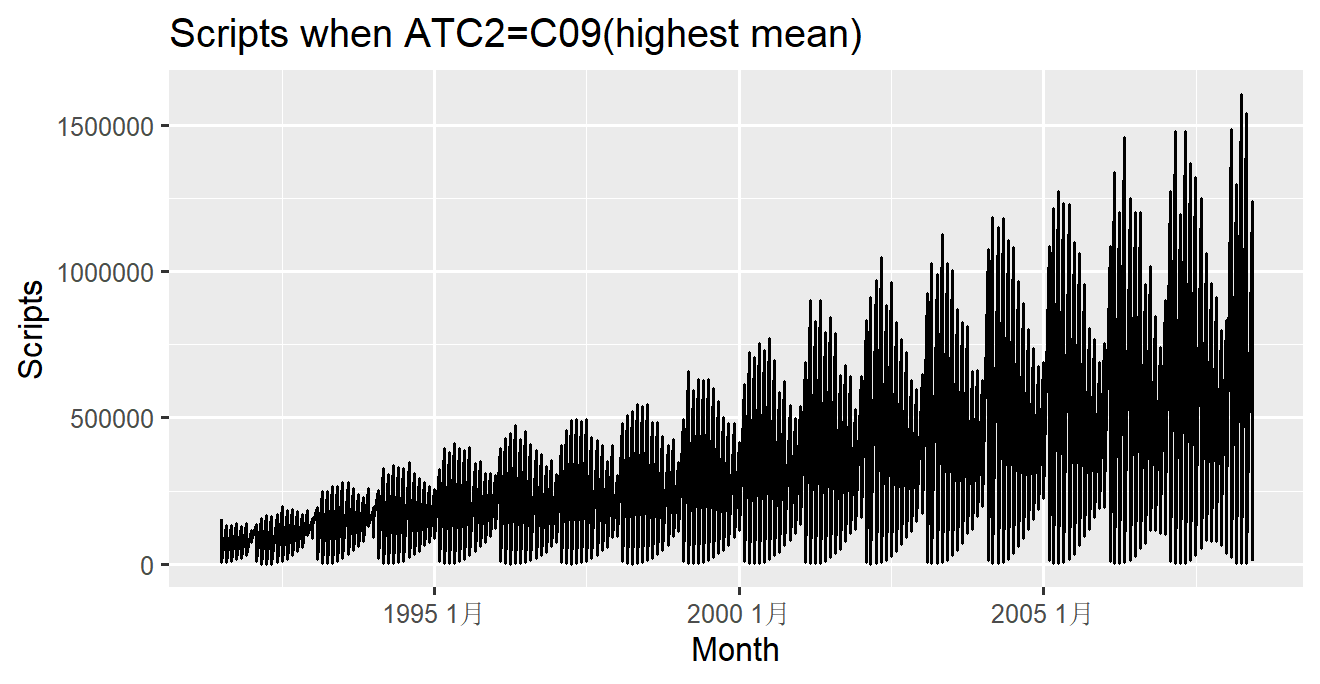
222041037 李子依

Section 4.6

Exercise 1

Each ATC2 has a complete timechain.Calculate the mean and standard deviation for each category of ACT2Cost and Scripts. The following results are obtained:





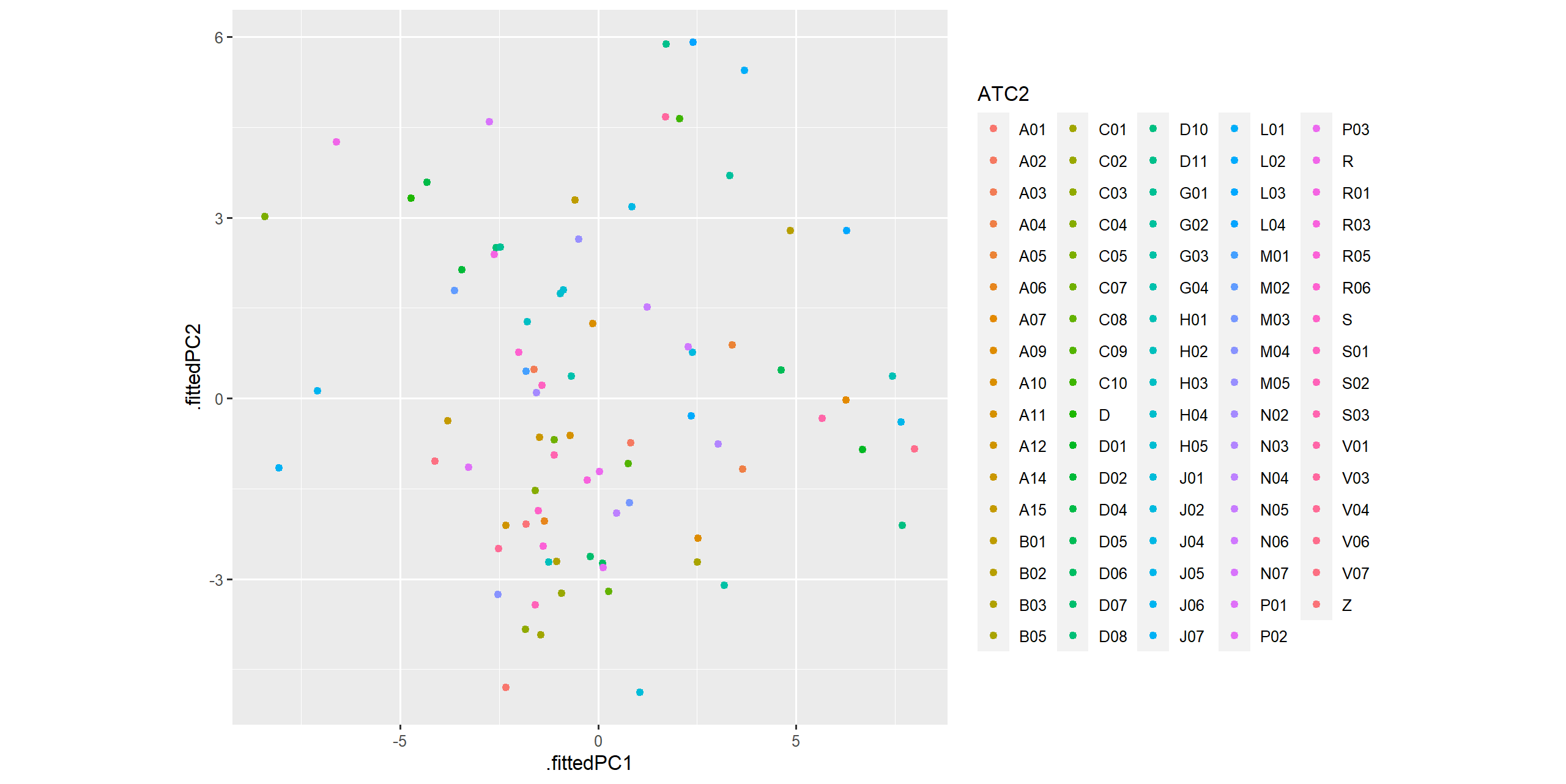
Exercise 3

First, calculate the feature of each time series

Code

PBS\_features <- PBS\_1 |>features(Scripts, feature\_set(pkgs = "feasts"))

After normalizing the feature matrix, the principal component analysis is done. Visualize principal component analysis results.



Based on the results, outliers are picked and an anomaly time series image is plotted

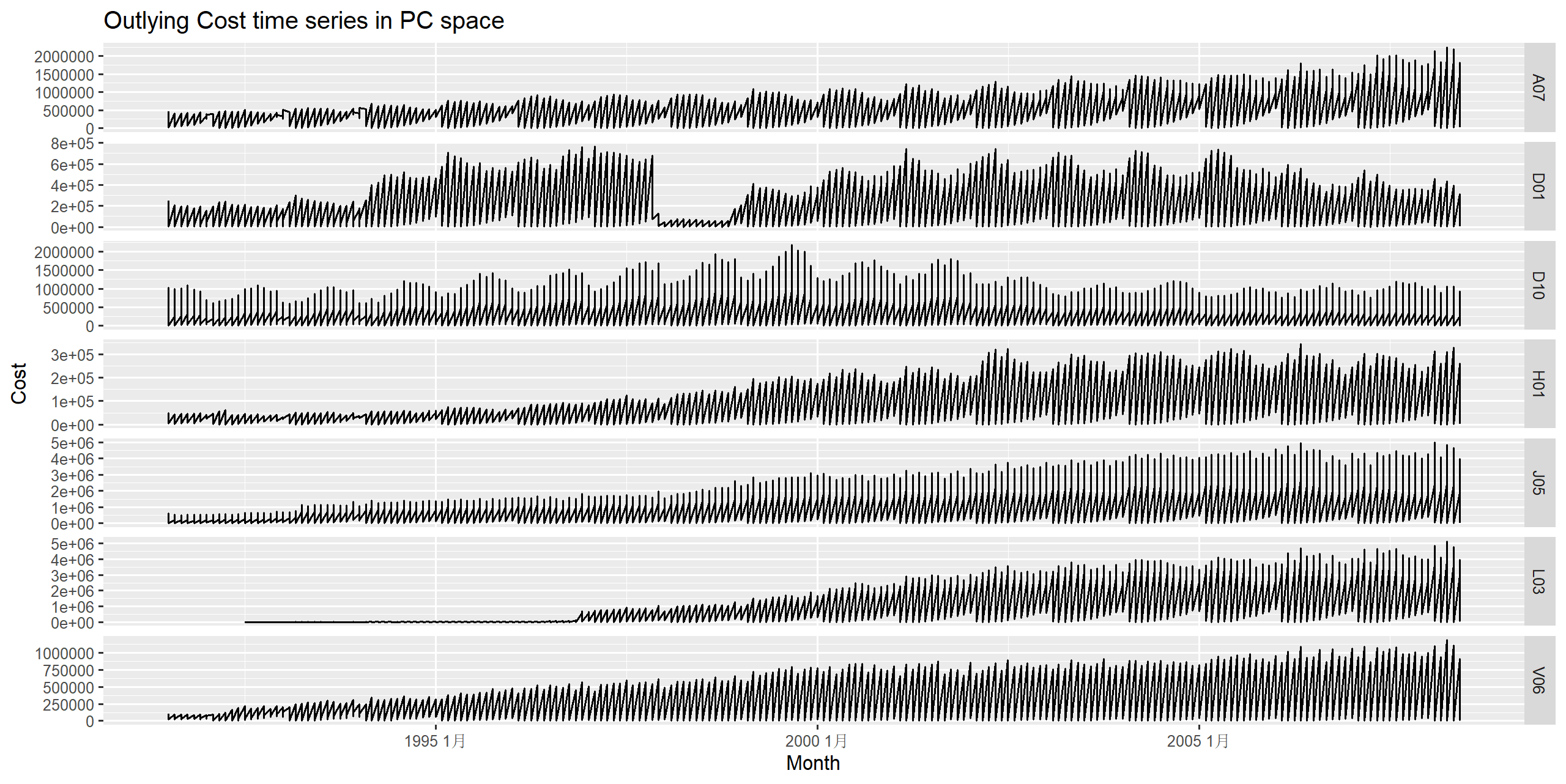
Code

outliers <- pcs |>

filter(.fittedPC1 > 6) |>

select(ATC2, .fittedPC1, .fittedPC2)

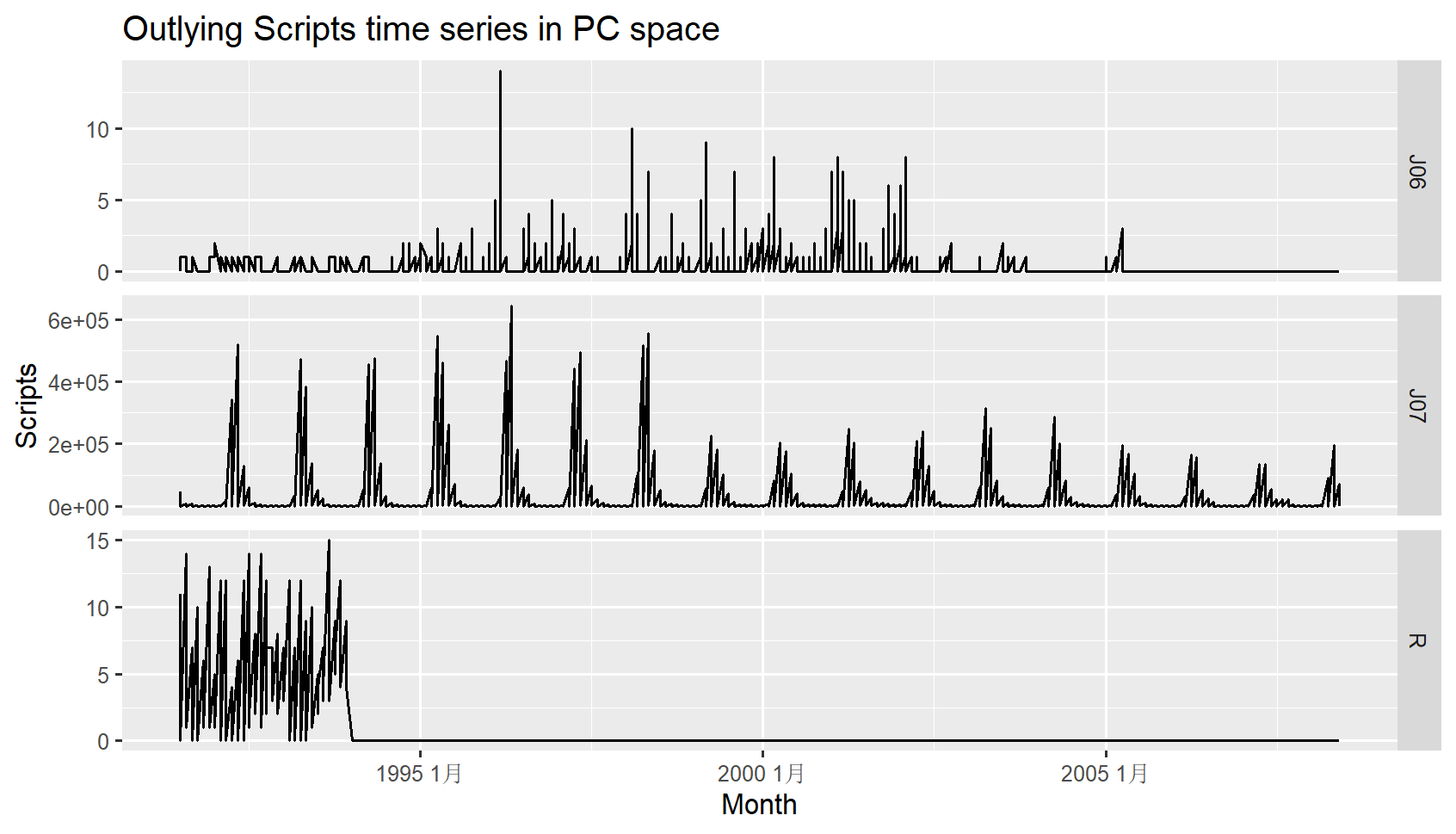
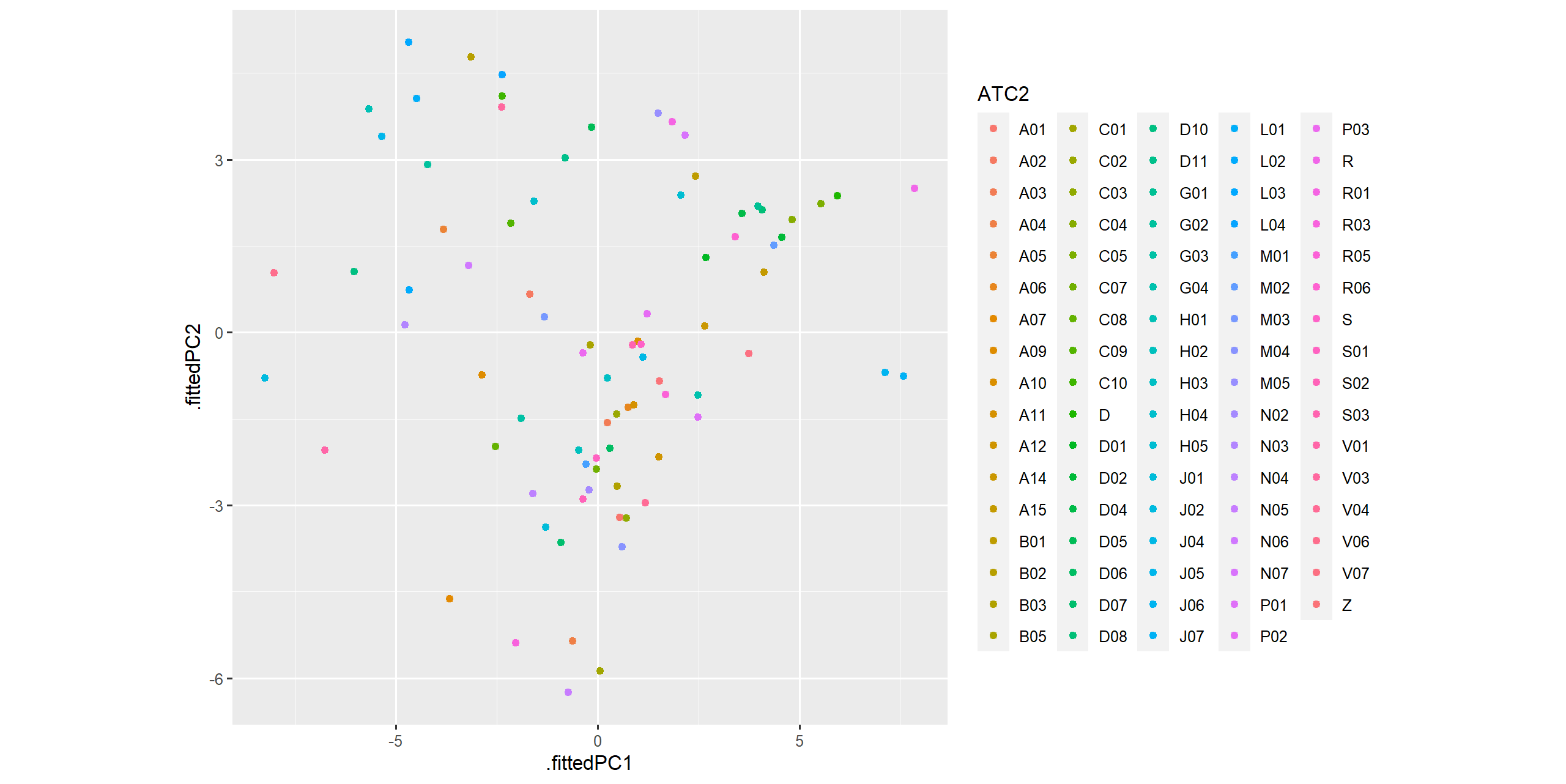
outliers



Outliers：A07/D01/D10/H01/J05/L03/V06

Comment: The time series is unstable, and there is a phenomenon that the sequence value changes at a certain point in time.

Scripts



Outliers：J06/J07/R

Comment: The time series is unstable, and there is a phenomenon that the sequence value changes at a certain point in time.

Section 5.11

Exercise 6

1. False. In some cases, the residuals may have a non-normal distribution due to the presence of outliers or a skewed data distribution. In such situations, alternative measures such as median absolute error (MAE) or mean absolute percentage error (MAPE) may be more appropriate to evaluate the model's performance.
2. True.small residuals can be an indication of a good model fit.
3. False. it has some limitationsIt can be sensitive to extreme values or outliers, which can skew the results.
4. False.It may be Overfitted
5. False. The test set may not be representative of the population, which means that the model's performance on the test set may not reflect its performance on new data. This is especially true when the sample size is small.

Exercise 9

1. To Create a training set for household wealthby withholding the last four years as a test set.We write R codes as follows:

data(hh\_budget)

#Find out the maximum year

max(hh\_budget$Year)

#Add and integrate data from different cities

mydf <- aggregate(hh\_budget[-c(1,2)], by=list(hh\_budget$Year), FUN=sum)

#Rename the column name

names(mydf)[1]="Year"

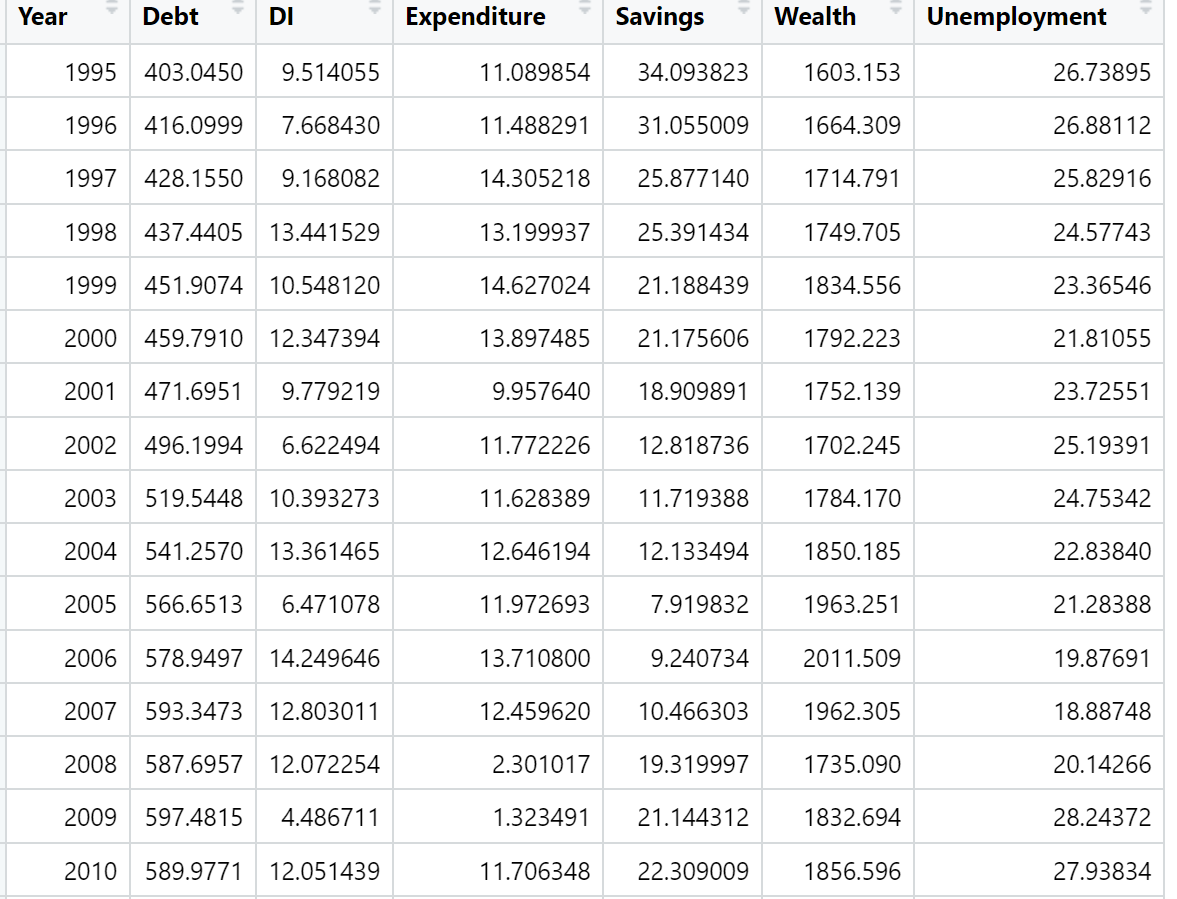
#Filter the training dataset

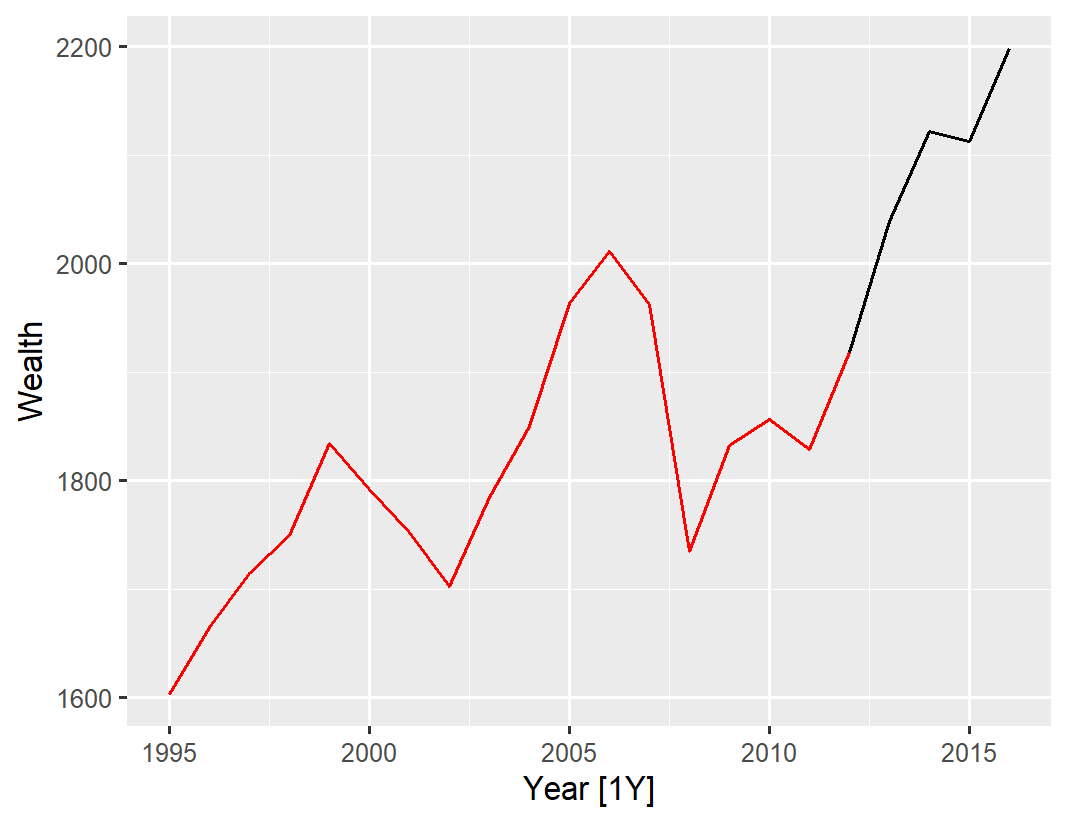
myseries = as\_tsibble(mydf,index =seq(1,22,1))

myseries\_train <-myseries |>

filter(Year < 2013)

The final data frame shows like





1. We use four benchmark methods:mean method,naïve method,snaive mthod and drift method. The effects of the above four methods are described in turn.

First,Mean method.

Code:

fit <- myseries\_train |> model(MEAN(Wealth))

fit |> gg\_tsresiduals()

fc <- fit |>

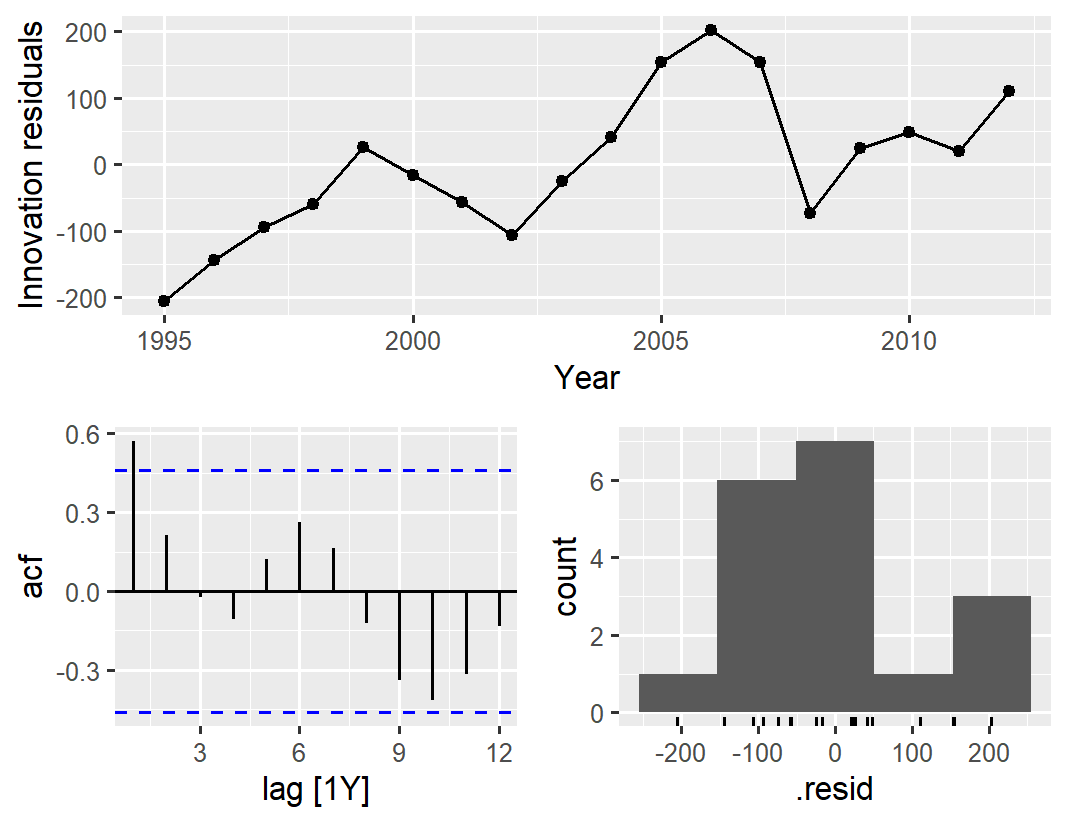
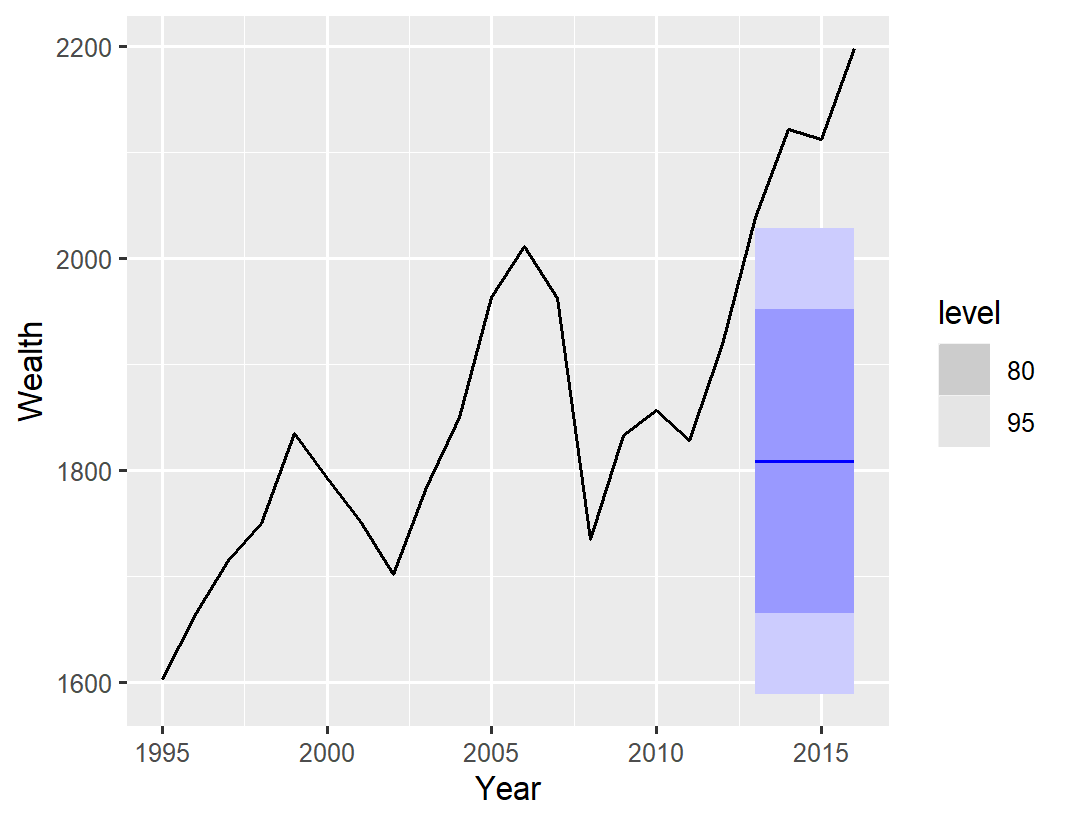
forecast(new\_data = anti\_join(myseries, myseries\_train))

fc |> autoplot(myseries)

fit |> accuracy()

fc |> accuracy(myseries)

The results are as follows:

.model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1

*<chr>* *<chr>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>*

1 MEAN(Wealth) Test 309. 314. 309. 14.5 14.5 4.42 3.75 -0.0642

Second, Naïve method.

Code:

#Naïve method

fit <- myseries\_train |> model(NAIVE(Wealth))

fit |> gg\_tsresiduals()

fc <- fit |>

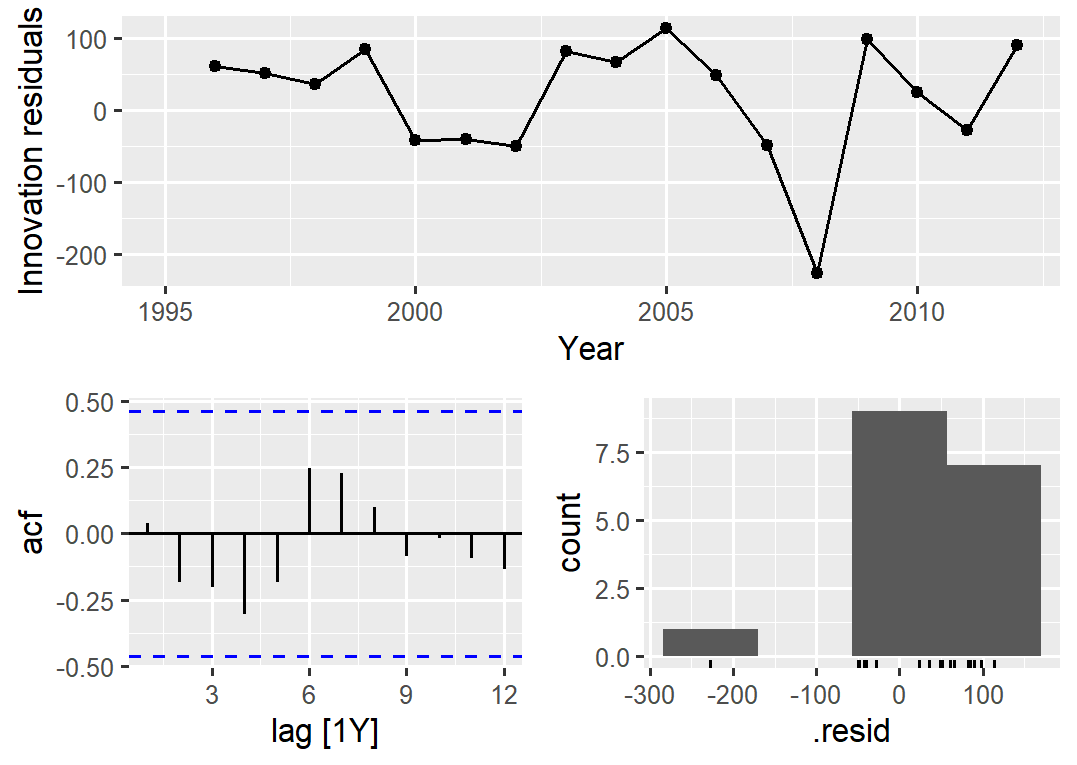
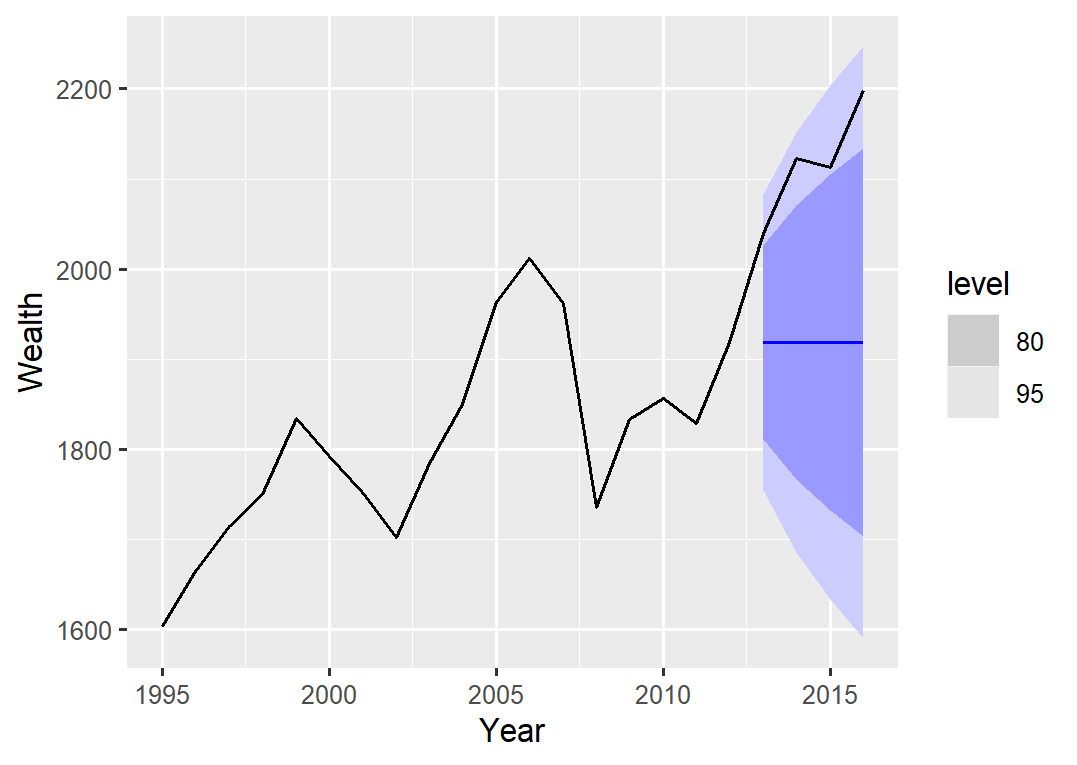
forecast(new\_data = anti\_join(myseries, myseries\_train))

fc |> autoplot(myseries)

fit |> accuracy()

fc |> accuracy(myseries)

The results are as follows:

.model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1

*<chr>* *<chr>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>*

1 NAIVE(Wealth) Test 199. 207. 199. 9.35 9.35 2.85 2.47 -0.0642

Third, SNaïve method. When I try to model, R shows that this model is not suitable for the series.

The last one:drift method.

code

#Drift method

fit <- myseries\_train |> model(RW(Wealth ~ drift()))

fit |> gg\_tsresiduals()

fc <- fit |>

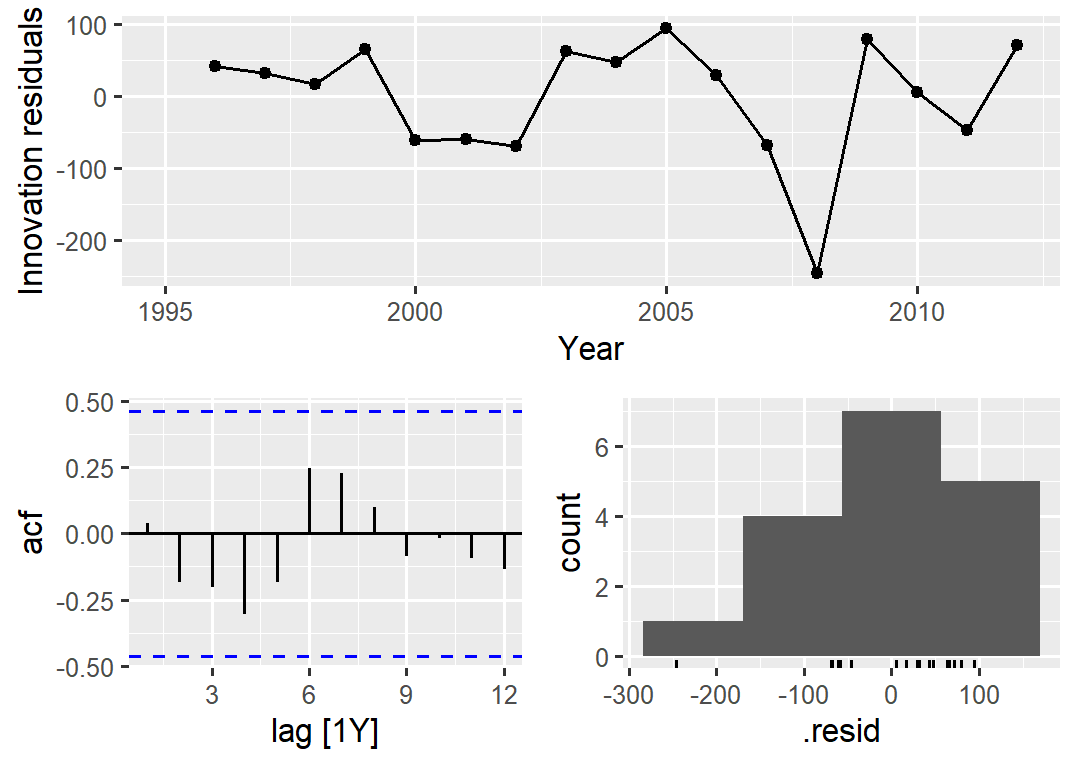
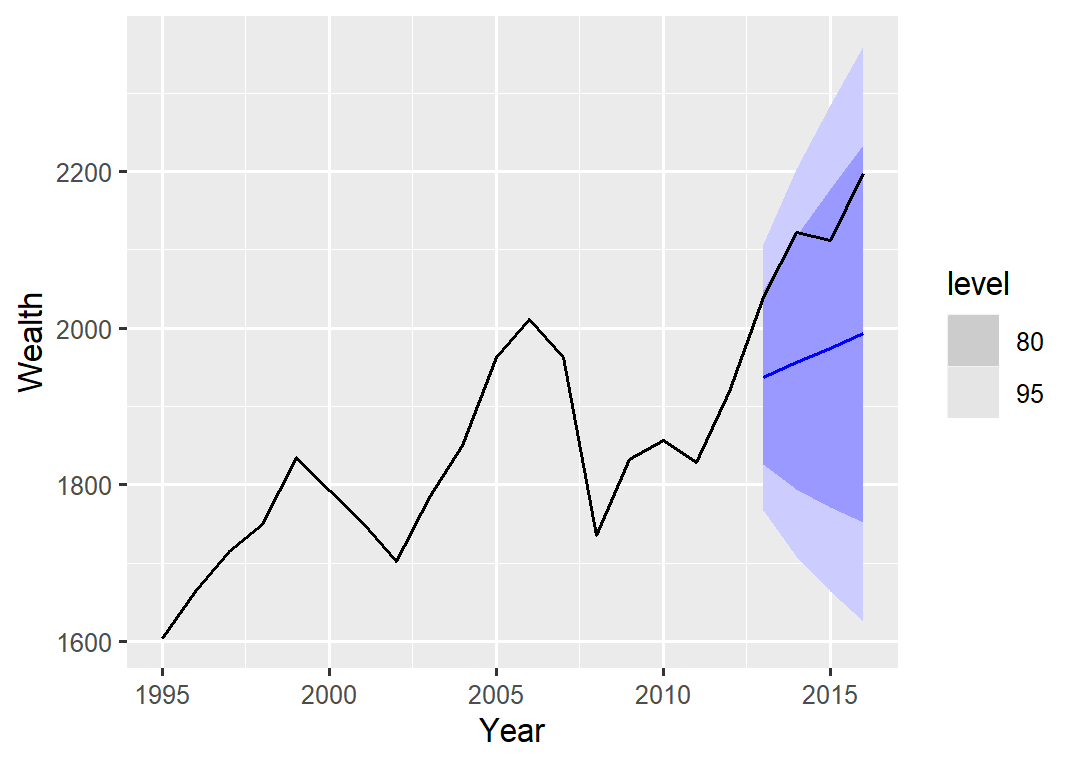
forecast(new\_data = anti\_join(myseries, myseries\_train))

fc |> autoplot(myseries)

fit |> accuracy()

fc |> accuracy(myseries)

The results are as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1  <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  1 RW(Wealth ~ drift()) Test 153. 158. 153. 7.18 7.18 2.19 1.88 -0.292   1. We use RMSE to measure accuracy. According to b:  |  |  | | --- | --- | | METHOD | RMSE(Test) | | Mean | 314. | | Naive | 207. | | SNaive | nan | | Drift | 158. |   Method Drift has the smallest RMSE.   1. Applying the Ljung-Box test:   Code:  augment(fit) |> features(.innov, ljung\_box, lag=5)  we obtain the following result.  .model lb\_stat lb\_pvalue  <chr> <dbl> <dbl>  1 RW(Wealth ~ drift()) 9.05 0.527  The residuals from the drift method are indistinguishable from a white noise series.  Exercise 12   1. To get the new dataset   Code:  data(tourism)  #Extract data from the Gold Coast region  gc\_tourism <-tourism |>  filter( Region=="Gold Coast")  #aggregate total overnight trips  gc\_tourism = summarise(index\_by(gc\_tourism,Quarter),sum(Trips))  gc\_tourism = as\_tsibble(gc\_tourism)  names(gc\_tourism)[2]<-"Trips"  The new dataset is like:     1. We use slice()to great train datesets:   Code:  #create three training sets for this data excluding the last 1, 2 and 3 years  gc\_train\_1 <- gc\_tourism |> slice(1:(n()-4))  gc\_train\_2 <- gc\_tourism |> slice(1:(n()-4\*2))  gc\_train\_3 <- gc\_tourism |> slice(1:(n()-4\*3))   1. Compute one year of forecasts for each training set using the seasonal naïve method   Code  #gc\_train\_1  fit\_1 <- gc\_train\_1 |> model(SNAIVE(Trips))  test\_data = gc\_tourism |> slice((n()-3):n())  fc\_1 <- fit\_1 |>  forecast(new\_data = test\_data)  fc\_1 |> autoplot(gc\_tourism)  fc\_1 |> accuracy(gc\_tourism)  #gc\_train\_2  fit\_2 <- gc\_train\_2|> model(SNAIVE(Trips))  test\_data = gc\_tourism |> slice((n()-7):(n()-4))  fc\_2 <- fit\_2 |>  forecast(new\_data = test\_data)  fc\_2 |> autoplot(gc\_tourism)  fc\_2 |> accuracy(gc\_tourism)  #gc\_train\_3  fit\_3 <- gc\_train\_3|> model(SNAIVE(Trips))  test\_data = gc\_tourism |> slice((n()-11):(n()-8))  fc\_3 <- fit\_3 |>  forecast(new\_data = test\_data)  fc\_3 |> autoplot(gc\_tourism)  fc\_3|> accuracy(gc\_tourism)   1. The results of above codes are:  |  |  | | --- | --- | | Train\_set | Accuracy(MAPE) | | Gc\_train\_1 | 15.1 | | Gc\_train\_2 | 4.32 | | Gc\_train\_3 | 9.07 |     MAPE is inversely proportional to the model effect, therefore gc\_train\_3 performs best and  Gc\_train\_1 performs worst.  Comment: As the number of training datasets increases, the model effect first becomes better  and then deteriorates, and overfitting occurs. |
|  |
| |  | | --- | |  | |