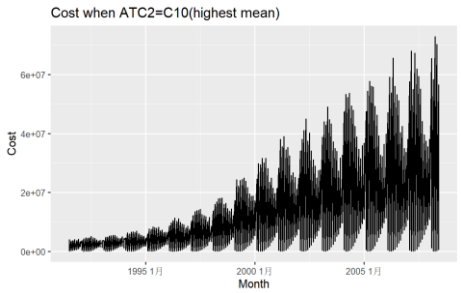
MDS5130 Assignment 2

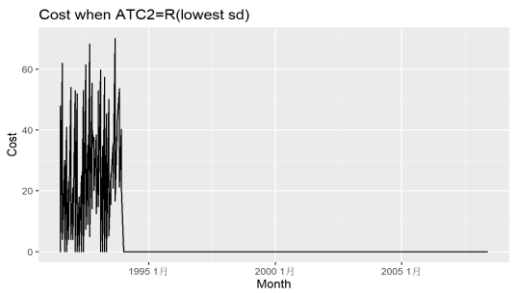
222041037 李子依

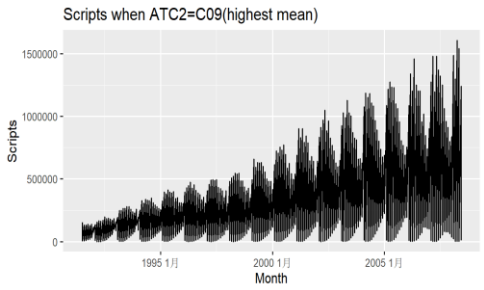
Section 4.6

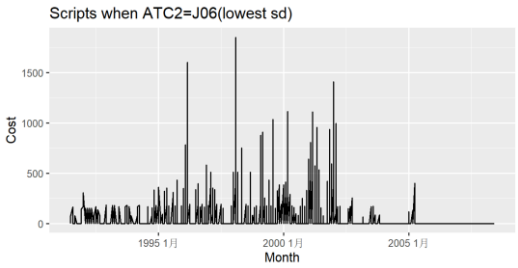
Exercise 1

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









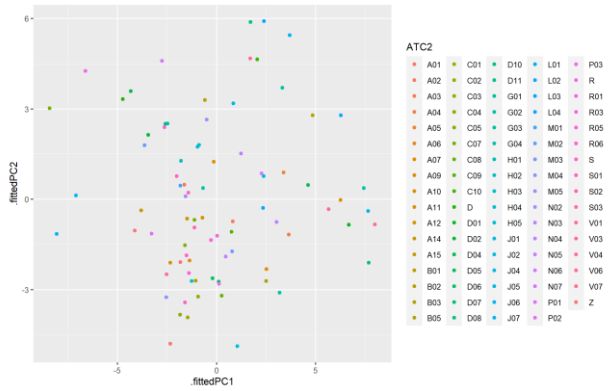
Exercise 3

First, calculate the feature of each time series

Code

```
PBS_features <- PBS_1 |>features(Scripts, feature_set(pkgs = "feasts"))
```

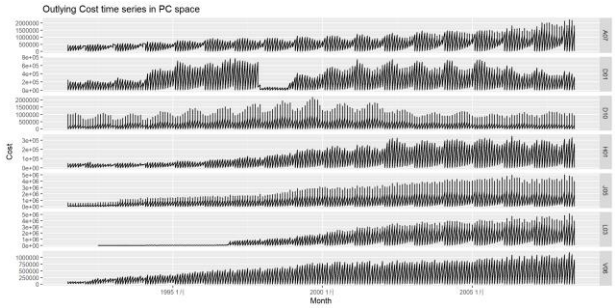
After normalizing the feature matrix, the principal component analysis is done. Visualize principal component analysis results.



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

Code

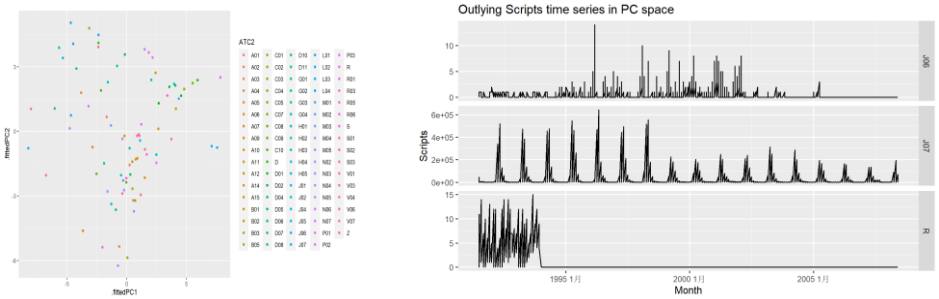
```
outliers <- pcs |>
  filter(.fittedPC1 > 6) |>
  select(ATC2, .fittedPC1, .fittedPC2)
outliers
```



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

a. 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.
b. True.small residuals can be an indication of a good model fit.
c. False. it has some limitationsIt can be sensitive to extreme values or outliers, which can skew the results.
d. False.It may be Overfitted
e. 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

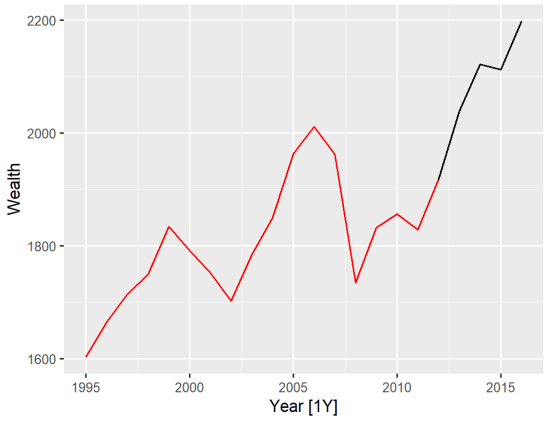
a. To Create a training set for household wealthby withholding the last four years as a test set.We write R codes as follows:

```
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

Year	Debt	DI	Expenditure	Savings	Wealth	Unemployment
1995	403.0450	9.514055	11.089854	34.093823	1603.153	26.73895
1996	416.0999	7.668430	11.488291	31.055009	1664.309	26.88112
1997	428.1550	9.168082	14.305218	25.877140	1714.791	25.82916
1998	437.4405	13.441529	13.199937	25.391434	1749.705	24.57743
1999	451.9074	10.548120	14.627024	21.188439	1834.556	23.36546
2000	459.7910	12.347394	13.897485	21.175606	1792.223	21.81055
2001	471.6951	9.779219	9.957640	18.909891	1752.139	23.72551
2002	496.1994	6.622494	11.772226	12.818736	1702.245	25.19391
2003	519.5448	10.393273	11.628389	11.719388	1784.170	24.75342
2004	541.2570	13.361465	12.646194	12.133494	1850.185	22.83840
2005	566.6513	6.471078	11.972693	7.919832	1963.251	21.28388
2006	578.9497	14.249646	13.710800	9.240734	2011.509	19.87691
2007	593.3473	12.803011	12.459620	10.466303	1962.305	18.88748
2008	587.6957	12.072254	2.301017	19.319997	1735.090	20.14266
2009	597.4815	4.486711	1.323491	21.144312	1832.694	28.24372
2010	589.9771	12.051439	11.706348	22.309009	1856.596	27.93834



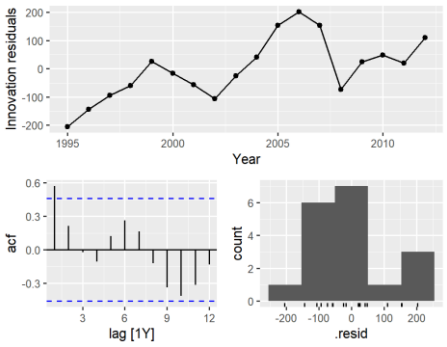
b. We use four benchmark methods:mean method,naïve method,snaive mthod and drift method. The effects of the above four methods are described in turn.

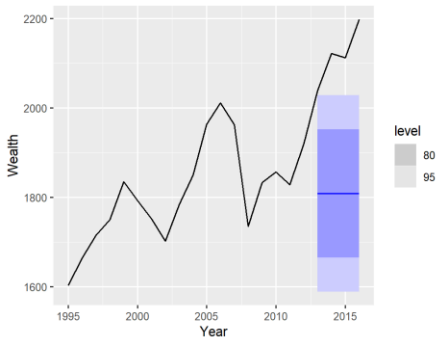
First,Mean method.

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:





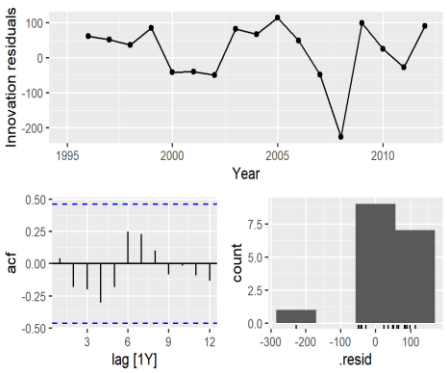
.model	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
<chr>	*<chr>*	*<dbl>*	*<dbl>*	*<dbl>*	*<dbl>*	*<dbl>*	*<dbl>*	*<dbl>*	*<dbl>*
1 MEAN(Wealth)	Test	309.	314.	309.	14.5	14.5	4.42	3.75	-0.0642

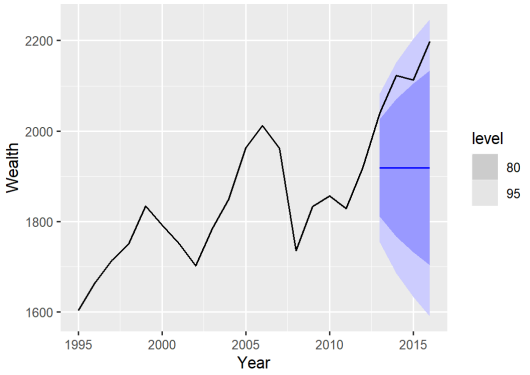
Second, Naïve method.

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:





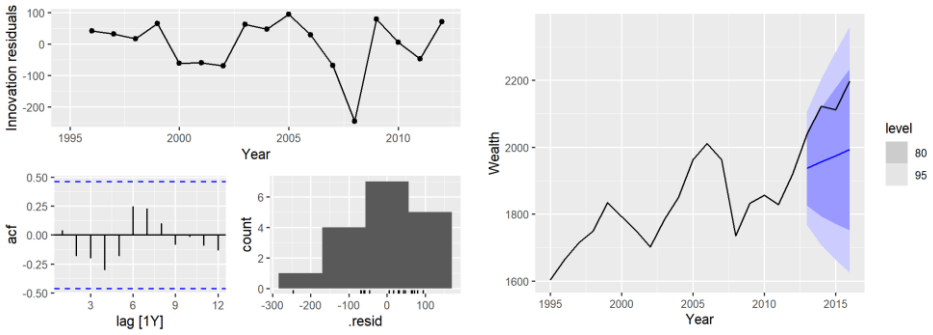
```
.model       .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE    ACF1
  <chr>      <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
1 NAIVE(Wealth) Test   199.  207.  199.  9.35  9.35  2.85  2.47 -0.0642
```

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
```

c. 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.

d. 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

a. 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:

Quarter	Trips
1998 Q1	827.0458
1998 Q2	680.7745
1998 Q3	839.0158
1998 Q4	819.8598
1999 Q1	986.8922
1999 Q2	751.0419
1999 Q3	822.1849
1999 Q4	913.5417
2000 Q1	871.3420
2000 Q2	780.2635

b. 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))
```

c. 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)
```

d. 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.