STA4003_Project

Yuzhou Peng

2023-12-01

```
library(dplyr)
```

```
##
## 载入程辑包：'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(fpp3)
```

```
## Warning: 程辑包'fpp3'是用R版本4.2.3 来建造的
```

```
## ── Attaching packages ─────────────────────────────────────
──────────────── fpp3 0.5 ──
```

```
## ✓ tibble      3.1.8     ✓ tsibbledata 0.4.1
## ✓ tidyr       1.3.0     ✓ feasts      0.3.1
## ✓ lubridate   1.9.2     ✓ fable       0.3.3
## ✓ ggplot2     3.4.1     ✓ fabletools  0.3.3
## ✓ tsibble     1.1.3
```

```
## Warning: 程辑包'tsibble'是用R版本4.2.3 来建造的
```

```
## Warning: 程辑包'tsibbledata'是用R版本4.2.3 来建造的
```

```
## Warning: 程辑包'feasts'是用R版本4.2.3 来建造的
```

```
## Warning: 程辑包'fabletools'是用R版本4.2.3 来建造的
```

```
## Warning: 程辑包'fable'是用R版本4.2.3 来建造的
```

```
## ── Conflicts ──────────────────────────────────────────────────────
──────────────── fpp3_conflicts ──
## ✖ lubridate::date()    masks base::date()
## ✖ dplyr::filter()      masks stats::filter()
## ✖ tsibble::intersect() masks base::intersect()
## ✖ tsibble::interval()  masks lubridate::interval()
## ✖ dplyr::lag()         masks stats::lag()
## ✖ tsibble::setdiff()   masks base::setdiff()
## ✖ tsibble::union()     masks base::union()
```

```
library(ggplot2)
```

1. Data preprocessing

1. Combine the data from 2014 to 2017 in **Train_all**
2. Extract data of 1st, 2nd, 3rd …….11th race of the racing days

```
load("data2014.RData")
load("data2015.RData")
load("data2016.RData")
load("data2017.RData")
load("data2018.RData")

Train1 <- data2014
Train2 <- data2015
Train3 <- data2016
Train4 <- data2017
Test1 <- data2018

#data from 2014 to 2017
Train_all <- rbind(Train1, Train2, Train3, Train4)

train_all_Csum <- Train_all %>%
  mutate(Csum = WIN_POOL.y - WIN_POOL.x)

#data in 2018
test_Csum <- Test1 %>%
  mutate(Csum = WIN_POOL.y - WIN_POOL.x)

#data from 2014 to 2018
data_all <- rbind(Train_all, Test1)

data_all_Csum <- data_all %>%
  mutate(Csum = WIN_POOL.y - WIN_POOL.x)

#-------------------------------------------

#Extract the i-th race on each day

#1.
data_all_1 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 1) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_1 <- test_Csum %>%
  filter(WIN_NUMBER.y == 1) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_1 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 1) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#2.
data_all_2 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 2) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_2 <- test_Csum %>%
```

```
  filter(WIN_NUMBER.y == 2) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_2 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 2) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#3.
data_all_3 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 3) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_3 <- test_Csum %>%
  filter(WIN_NUMBER.y == 3) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_3 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 3) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#4.
data_all_4 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 4) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_4 <- test_Csum %>%
  filter(WIN_NUMBER.y == 4) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_4 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 4) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#5.
data_all_5 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 5) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_5 <- test_Csum %>%
  filter(WIN_NUMBER.y == 5) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_5 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 5) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)
```

```
#6.
data_all_6 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 6) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_6 <- test_Csum %>%
  filter(WIN_NUMBER.y == 6) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_6 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 6) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#7.
data_all_7 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 7) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_7 <- test_Csum %>%
  filter(WIN_NUMBER.y == 7) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_7 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 7) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#8.
data_all_8 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 8) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_8 <- test_Csum %>%
  filter(WIN_NUMBER.y == 8) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_8 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 8) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#9.
data_all_9 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 9) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_9 <- test_Csum %>%
```

```
  filter(WIN_NUMBER.y == 9) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_9 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 9) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#10.
data_all_10 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 10) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_10 <- test_Csum %>%
  filter(WIN_NUMBER.y == 10) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_10 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 10) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

#11.
data_all_11 <- data_all_Csum %>%
  filter(WIN_NUMBER.y == 11) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

test_11 <- test_Csum %>%
  filter(WIN_NUMBER.y == 11) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)

train_all_11 <- train_all_Csum %>%
  filter(WIN_NUMBER.y == 11) %>%
  mutate(time_index = row_number()) %>%
  as_tsibble(index = time_index)
```

2. Model fitting and prediction

The model includes 2 models: (1) a linear model with **Csum** , **WIN_POOL.x** being the response variable (y) and explanatory variable (x). (2) a vector autoregression (**VAR**) model For data of i-th race, we fit the model as follow.

Step 1: Fit a **TSLM** model to the first 37.5% days (rounded by **floor()**) of the i-th races and predict the first 30 days data in the test set. If the test set have fewer days then 30, predict all data using this model.

Step 2: Fit a vector autoregression model to the first j (j > 30) days in test set and predict data on day j+1. And thus iteratively generate all prediction of data after 30th day.

3. The result is stored in **result_i** for i-th race. **result_all** is the combined set of all **result_i**, containing **true_value**, **forecast**, **ape** (absolute percentatage error).

```
# 1st race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_1 <- train_all_1 %>%
  filter(time_index <= floor(nrow(train_all_1) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_1 <- tslm_pool_1 %>%
  forecast(new_data = test_1 %>% filter(time_index <= 30))

tslm_1_forecast <- data.frame(forecast =  fc_tslm_pool_1$.mean)

#Fit the rest data using VAR model

var_1_forecast = data.frame()

for (i in 30:(nrow(test_1) - 1)) {
  var_pool <- test_1 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_1_forecast <- rbind(var_1_forecast, fc_val)
}

names(var_1_forecast)[1] <- "forecast"

forecast_1 <- rbind(tslm_1_forecast, var_1_forecast)

result_1 <- data.frame(true_value = test_1$Csum,
                       forecast_value = forecast_1,
                       ape = abs((forecast_1$forecast - test_1$Csum)/test_1$Csum))
```

```
# 2nd race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_2 <- train_all_2 %>%
  filter(time_index <= floor(nrow(train_all_2) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_2 <- tslm_pool_2 %>%
  forecast(new_data = test_2 %>% filter(time_index <= 30))

tslm_2_forecast <- data.frame(forecast =  fc_tslm_pool_2$.mean)

#Fit the rest data using VAR model

var_2_forecast = data.frame()

for (i in 30:(nrow(test_2) - 1)) {
  var_pool <- test_2 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_2_forecast <- rbind(var_2_forecast, fc_val)
}

names(var_2_forecast)[1] <- "forecast"

forecast_2 <- rbind(tslm_2_forecast, var_2_forecast)

result_2 <- data.frame(true_value = test_2$Csum,
                       forecast_value = forecast_2,
                       ape = abs((forecast_2$forecast - test_2$Csum)/test_2$Csum))
```

```
# 3rd race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_3 <- train_all_3 %>%
  filter(time_index <= floor(nrow(train_all_3) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_3 <- tslm_pool_3 %>%
  forecast(new_data = test_3 %>% filter(time_index <= 30))

tslm_3_forecast <- data.frame(forecast =  fc_tslm_pool_3$.mean)

#Fit the rest data using VAR model

var_3_forecast = data.frame()

for (i in 30:(nrow(test_3) - 1)) {
  var_pool <- test_3 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_3_forecast <- rbind(var_3_forecast, fc_val)
}
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
```

```
names(var_3_forecast)[1] <- "forecast"

forecast_3 <- rbind(tslm_3_forecast, var_3_forecast)

result_3 <- data.frame(true_value = test_3$Csum,
                       forecast_value = forecast_3,
                       ape = abs((forecast_3$forecast - test_3$Csum)/test_3$Csum))
```

```
# 4th race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_4 <- train_all_4 %>%
  filter(time_index <= floor(nrow(train_all_4) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_4 <- tslm_pool_4 %>%
  forecast(new_data = test_4 %>% filter(time_index <= 30))

tslm_4_forecast <- data.frame(forecast =  fc_tslm_pool_4$.mean)

#Fit the rest data using VAR model

var_4_forecast = data.frame()

for (i in 30:(nrow(test_4) - 1)) {
  var_pool <- test_4 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_4_forecast <- rbind(var_4_forecast, fc_val)
}
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
```

```
names(var_4_forecast)[1] <- "forecast"

forecast_4 <- rbind(tslm_4_forecast, var_4_forecast)

result_4 <- data.frame(true_value = test_4$Csum,
                       forecast_value = forecast_4,
                       ape = abs((forecast_4$forecast - test_4$Csum)/test_4$Csum))
```

```
# 5th race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_5 <- train_all_5 %>%
  filter(time_index <= floor(nrow(train_all_5) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_5 <- tslm_pool_5 %>%
  forecast(new_data = test_5 %>% filter(time_index <= 30))

tslm_5_forecast <- data.frame(forecast =  fc_tslm_pool_5$.mean)

#Fit the rest data using VAR model

var_5_forecast = data.frame()

for (i in 30:(nrow(test_5) - 1)) {
  var_pool <- test_5 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_5_forecast <- rbind(var_5_forecast, fc_val)
}
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
```

```
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
```

```
names(var_5_forecast)[1] <- "forecast"

forecast_5 <- rbind(tslm_5_forecast, var_5_forecast)

result_5 <- data.frame(true_value = test_5$Csum,
                       forecast_value = forecast_5,
                       ape = abs((forecast_5$forecast - test_5$Csum)/test_5$Csum))
```

```
# 6th race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_6 <- train_all_6 %>%
  filter(time_index <= floor(nrow(train_all_6) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_6 <- tslm_pool_6 %>%
  forecast(new_data = test_6 %>% filter(time_index <= 30))

tslm_6_forecast <- data.frame(forecast =  fc_tslm_pool_6$.mean)

#Fit the rest data using VAR model

var_6_forecast = data.frame()

for (i in 30:(nrow(test_6) - 1)) {
  var_pool <- test_6 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_6_forecast <- rbind(var_6_forecast, fc_val)
}
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
```

```
names(var_6_forecast)[1] <- "forecast"

forecast_6 <- rbind(tslm_6_forecast, var_6_forecast)

result_6 <- data.frame(true_value = test_6$Csum,
                       forecast_value = forecast_6,
                       ape = abs((forecast_6$forecast - test_6$Csum)/test_6$Csum))
```

```
# 7th race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_7 <- train_all_7 %>%
  filter(time_index <= floor(nrow(train_all_7) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_7 <- tslm_pool_7 %>%
  forecast(new_data = test_7 %>% filter(time_index <= 30))

tslm_7_forecast <- data.frame(forecast =  fc_tslm_pool_7$.mean)

#Fit the rest data using VAR model

var_7_forecast = data.frame()

for (i in 30:(nrow(test_7) - 1)) {
  var_pool <- test_7 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_7_forecast <- rbind(var_7_forecast, fc_val)
}
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
```

```
names(var_7_forecast)[1] <- "forecast"

forecast_7 <- rbind(tslm_7_forecast, var_7_forecast)

result_7 <- data.frame(true_value = test_7$Csum,
                       forecast_value = forecast_7,
                       ape = abs((forecast_7$forecast - test_7$Csum)/test_7$Csum))
```

```
# 8th race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_8 <- train_all_8 %>%
  filter(time_index <= floor(nrow(train_all_8) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_8 <- tslm_pool_8 %>%
  forecast(new_data = test_8 %>% filter(time_index <= 30))

tslm_8_forecast <- data.frame(forecast =  fc_tslm_pool_8$.mean)

#Fit the rest data using VAR model

var_8_forecast = data.frame()

for (i in 30:(nrow(test_8) - 1)) {
  var_pool <- test_8 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_8_forecast <- rbind(var_8_forecast, fc_val)
}
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
```

```
names(var_8_forecast)[1] <- "forecast"

forecast_8 <- rbind(tslm_8_forecast, var_8_forecast)

result_8 <- data.frame(true_value = test_8$Csum,
                       forecast_value = forecast_8,
                       ape = abs((forecast_8$forecast - test_8$Csum)/test_8$Csum))
```

```
# 9th race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_9 <- train_all_9 %>%
  filter(time_index <= floor(nrow(train_all_9) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_9 <- tslm_pool_9 %>%
  forecast(new_data = test_9 %>% filter(time_index <= 30))

tslm_9_forecast <- data.frame(forecast =  fc_tslm_pool_9$.mean)

#Fit the rest data using VAR model

var_9_forecast = data.frame()

for (i in 30:(nrow(test_9) - 1)) {
  var_pool <- test_9 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_9_forecast <- rbind(var_9_forecast, fc_val)
}

names(var_9_forecast)[1] <- "forecast"

forecast_9 <- rbind(tslm_9_forecast, var_9_forecast)

result_9 <- data.frame(true_value = test_9$Csum,
                       forecast_value = forecast_9,
                       ape = abs((forecast_9$forecast - test_9$Csum)/test_9$Csum))
```

```
# 10th race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_10 <- train_all_10 %>%
  filter(time_index <= floor(nrow(train_all_10) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_10 <- tslm_pool_10 %>%
  forecast(new_data = test_10 %>% filter(time_index <= 30))

tslm_10_forecast <- data.frame(forecast =  fc_tslm_pool_10$.mean)

#Fit the rest data using VAR model

var_10_forecast = data.frame()

for (i in 30:(nrow(test_10) - 1)) {
  var_pool <- test_10 %>%
    filter(time_index <= i) %>%
    model(var = VAR(vars(Csum, WIN_POOL.x)))

  fc <- var_pool %>%
    forecast(h = 1)

  fc_val <- fc$.mean[1]

  var_10_forecast <- rbind(var_10_forecast, fc_val)
}
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
## There was 1 warning in `mutate()`.
## ℹ In argument: `var = (function (object, ...) ...`.
## Caused by warning in `FUN()`:
## ! 产生了NaNs
```

```
names(var_10_forecast)[1] <- "forecast"

forecast_10 <- rbind(tslm_10_forecast, var_10_forecast)

result_10 <- data.frame(true_value = test_10$Csum,
                        forecast_value = forecast_10,
                        ape = abs((forecast_10$forecast - test_10$Csum)/test_10$Csum))
```

```
# 11th race
#Fit a linear regression to Csum ~ WIN_POOL.x

tslm_pool_11 <- train_all_11 %>%
  filter(time_index <= floor(nrow(train_all_11) * 0.375)) %>%
  model(TSLM(Csum ~ WIN_POOL.x))

fc_tslm_pool_11 <- tslm_pool_11 %>%
  forecast(new_data = test_11)

tslm_11_forecast <- data.frame(forecast =  fc_tslm_pool_11$.mean)

result_11 <- data.frame(true_value = test_11$Csum,
                        forecast_value = tslm_11_forecast,
                        ape = abs((tslm_11_forecast$forecast - test_11$Csum)/test_11$Csum))
```

3. Prediction results

1. MAPE is calculated by the mean value of **result_all$ape**.
2. 0.95-quantile score is stored in **QS**.

```
result_all = rbind(result_1, result_2, result_3, result_4, result_5, result_6, result_7, result
_8, result_9, result_10, result_11)

MAPE <- mean(result_all$ape)

qs_0.95 = data.frame()

for (i in 1:nrow(result_all)){
  qs = quantile_score(result_all$forecast[i], result_all$true_value[i], 0.95)
  qs_0.95 = rbind(qs_0.95, qs)
}

names(qs_0.95)[1] <- "QS"

QS <- mean(qs_0.95$QS)

MAPE
```

```
## [1] 0.2996481
```

```
QS
```

```
## [1] 1327762
```