Data 624: Project 1

Group 2: Alice Friedman, Kayleah Griffen, Josh Iden, Michael Ippolito

6/11/2023

Table of Contents

## Introduction

This report is intended for colleagues from a variety of backgrounds and contains both technical and non-technical explanations of the work conducted. The objective of this project was to perform the appropriate analysis in order to forecast two variables (of five provided) each from six different time series sets. We were provided a spreadsheet that contains 1622 periods of every variable in every set and were expected to forecast 140 periods. The sets are labeled S01, S02, S03, S04, S05 and S06 and each contains variables labeled V01, V02, V03, V05, and V07. Different variables are required to be forecast depending on the set, specified below:

S01 – Forecast Var01, Var02 S02 – Forecast Var02, Var03 S03 – Forecast Var05, Var07 S04 – Forecast Var01, Var02 S05 – Forecast Var02, Var03 S06 – Forecast Var05, Var07

The goal of this report is to forecast 140 future time periods for each of the above variables, minimizing mean absolute percent error (MAPE). Results will be attached in an Excel file with each category in a separate tab.

## Data Preparation

The data was provided as an excel (.xls) file. The columns provided were the series index, the category, and then a column for each variable. To conduct the data analysis and forecasting the open source software R was used. In order to begin processing the data, the data was read into R from github (where the provided data file was stored) and stored in a format in R called a dataframe. Below is a preview of the data to get an idea of the format.

Series Provided

SeriesInd

category

Var01

Var02

Var03

Var05

Var07

40669

S03

30.64286

123432400

30.34000

30.49000

30.57286

40669

S02

10.28000

60855800

10.05000

10.17000

10.28000

40669

S01

26.61000

10369300

25.89000

26.20000

26.01000

40669

S06

27.48000

39335700

26.82000

27.02000

27.32000

40669

S05

69.26000

27809100

68.19000

68.72000

69.15000

40669

S04

17.20000

16587400

16.88000

16.94000

17.10000

40670

S03

30.79857

150476200

30.46428

30.65714

30.62571

40670

S02

11.24000

215620200

10.40000

10.45000

10.96000

40670

S01

26.30000

10943800

25.70000

25.95000

25.86000

40670

S06

28.24000

55416000

27.24000

27.27000

28.07000

S01

SeriesInd

category

Var01

Var02

Var03

Var05

Var07

40669

S01

26.61

10369300

25.89

26.20

26.01

40670

S01

26.30

10943800

25.70

25.95

25.86

40671

S01

26.03

8933800

25.56

25.90

25.67

40672

S01

25.84

10775400

25.36

25.60

25.75

40673

S01

26.34

12875600

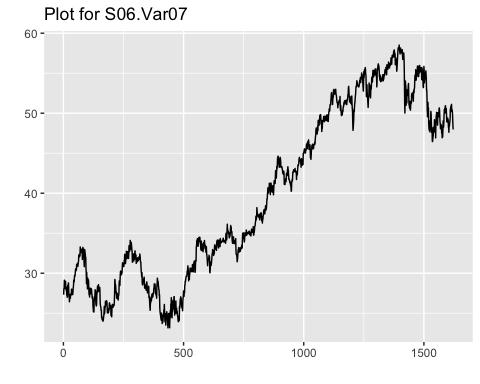
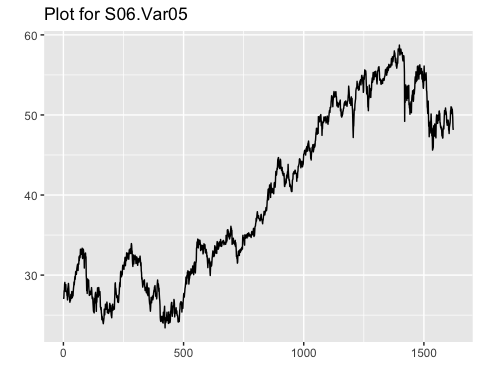
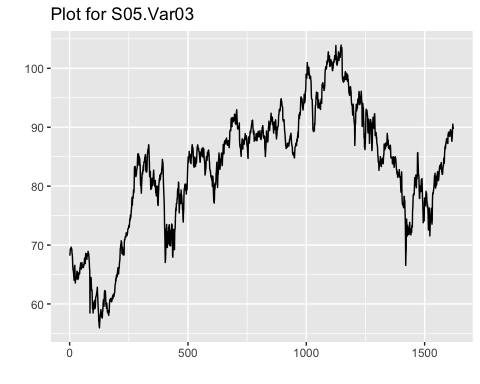
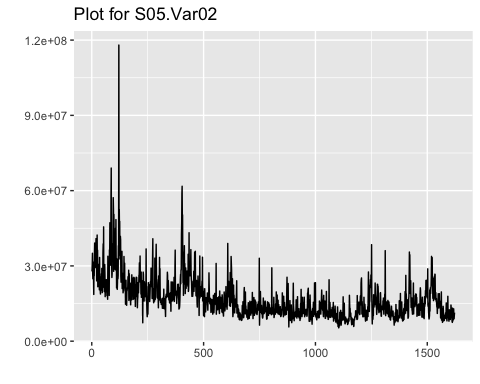
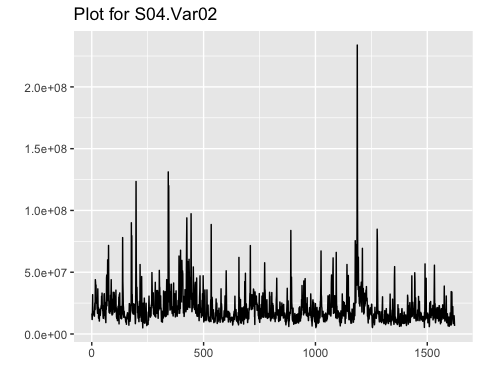
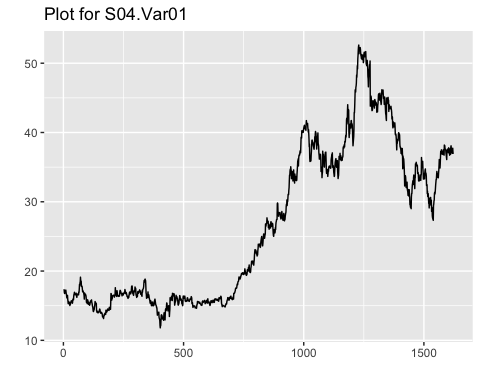
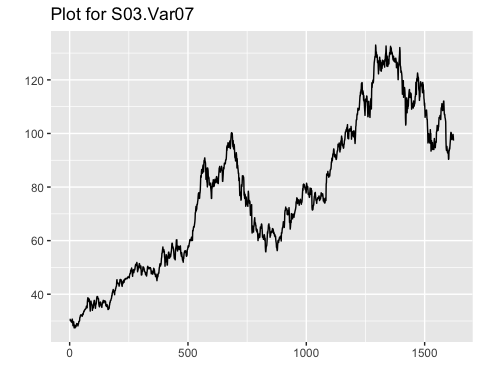
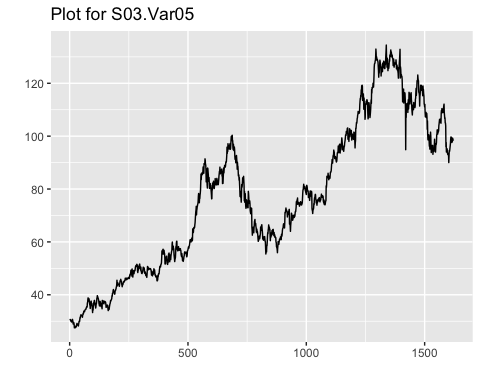
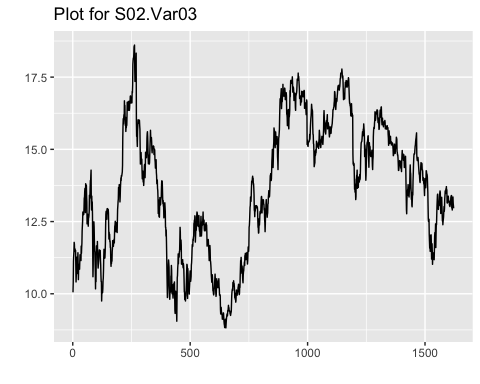
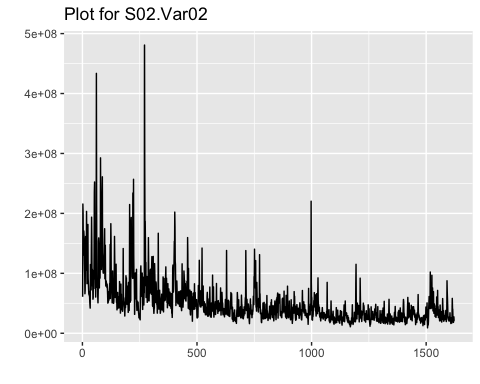
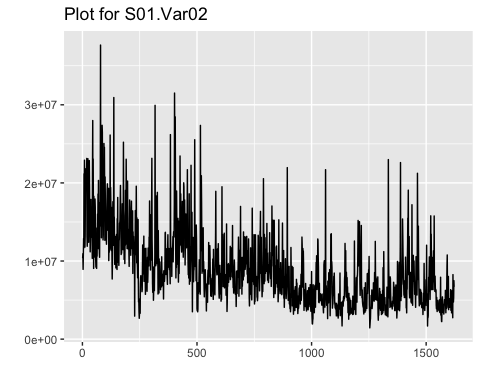
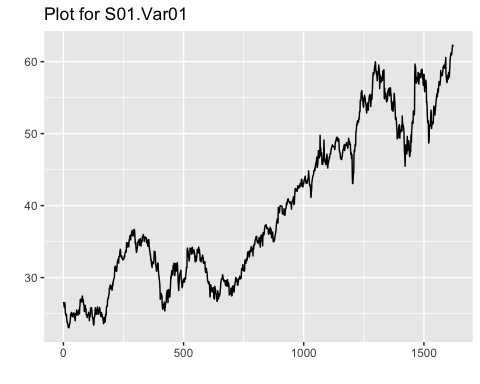
25.52

25.60

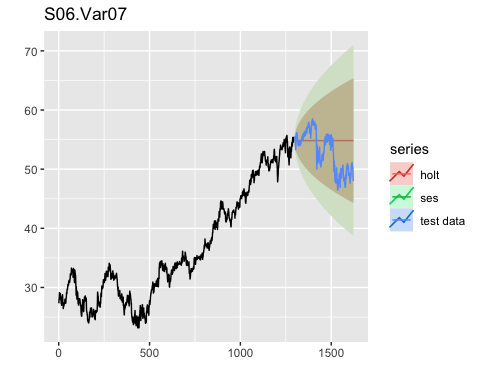
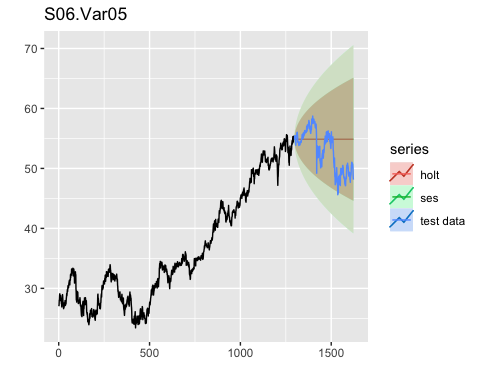
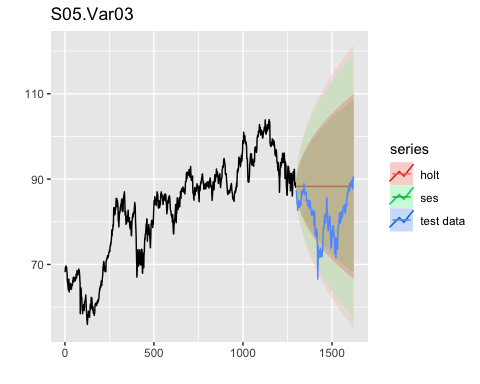
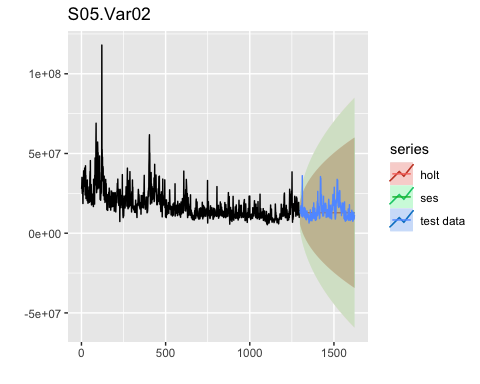
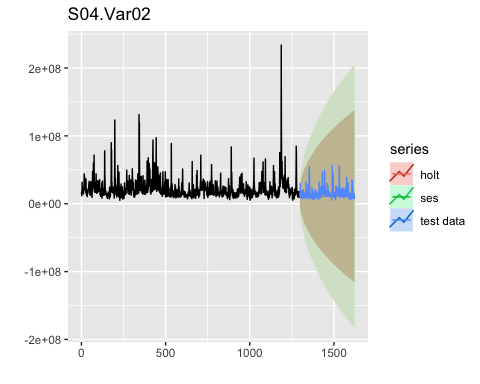
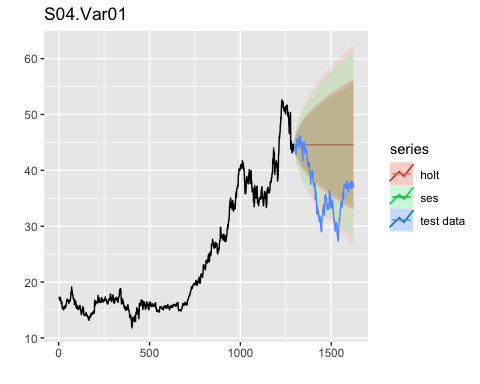
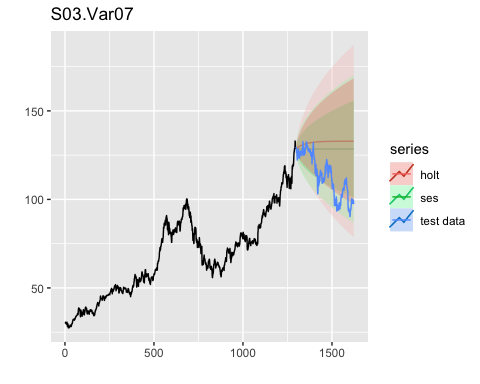
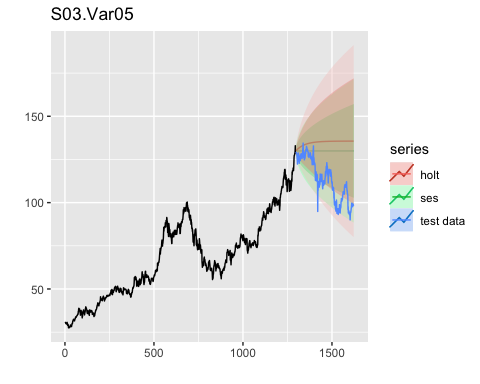
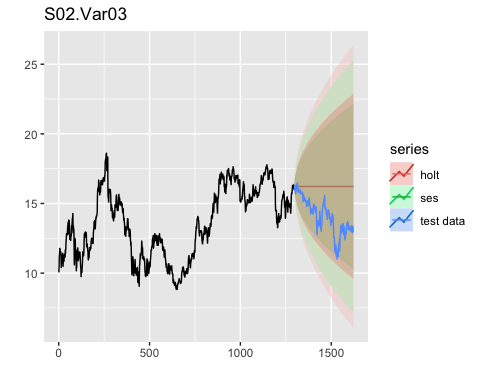
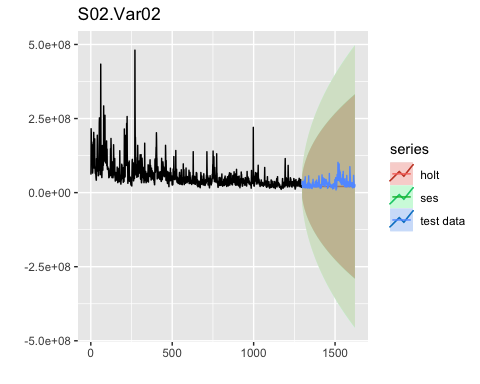
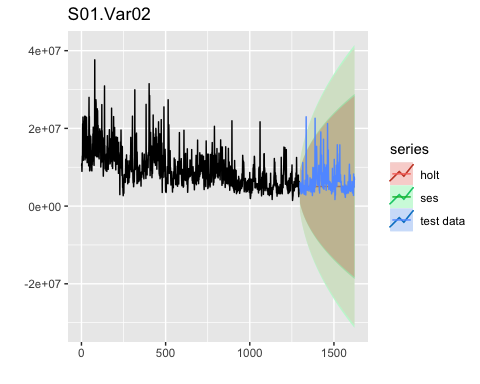
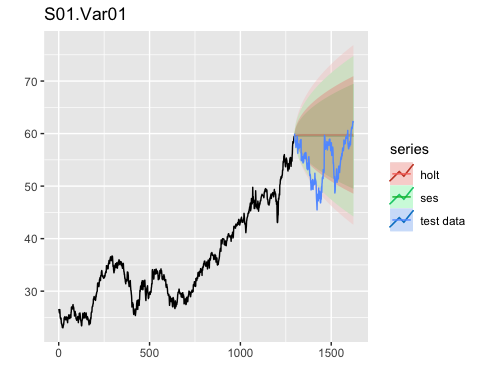
26.34

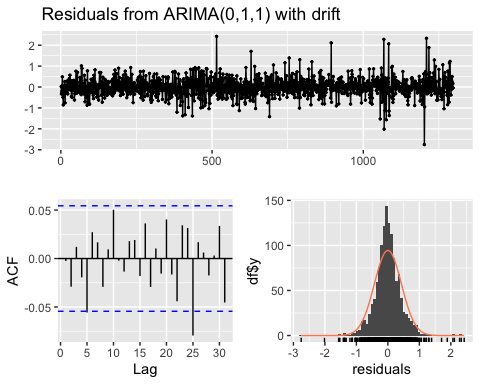
## $S01.Var01  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S01 Var01 26.6  
## 2 40670 S01 Var01 26.3  
## 3 40671 S01 Var01 26.0  
## 4 40672 S01 Var01 25.8  
## 5 40673 S01 Var01 26.3  
## 6 40676 S01 Var01 26.5  
## 7 40677 S01 Var01 26.0  
## 8 40678 S01 Var01 25.2  
## 9 40679 S01 Var01 25   
## 10 40680 S01 Var01 24.8  
## # ℹ 1,752 more rows  
##   
## $S02.Var01  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S02 Var01 10.3  
## 2 40670 S02 Var01 11.2  
## 3 40671 S02 Var01 11.5  
## 4 40672 S02 Var01 11.7  
## 5 40673 S02 Var01 11.7  
## 6 40676 S02 Var01 12.1  
## 7 40677 S02 Var01 12.0  
## 8 40678 S02 Var01 11.9  
## 9 40679 S02 Var01 11.9  
## 10 40680 S02 Var01 11.8  
## # ℹ 1,752 more rows  
##   
## $S03.Var01  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S03 Var01 30.6  
## 2 40670 S03 Var01 30.8  
## 3 40671 S03 Var01 30.7  
## 4 40672 S03 Var01 30.3  
## 5 40673 S03 Var01 30.3  
## 6 40676 S03 Var01 30.4  
## 7 40677 S03 Var01 30.0  
## 8 40678 S03 Var01 30.1  
## 9 40679 S03 Var01 30.1  
## 10 40680 S03 Var01 30.2  
## # ℹ 1,752 more rows  
##   
## $S04.Var01  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S04 Var01 17.2  
## 2 40670 S04 Var01 17.2  
## 3 40671 S04 Var01 17.3  
## 4 40672 S04 Var01 16.9  
## 5 40673 S04 Var01 16.8  
## 6 40676 S04 Var01 16.8  
## 7 40677 S04 Var01 16.9  
## 8 40678 S04 Var01 17.0  
## 9 40679 S04 Var01 17.2  
## 10 40680 S04 Var01 17.2  
## # ℹ 1,752 more rows  
##   
## $S05.Var01  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S05 Var01 69.3  
## 2 40670 S05 Var01 69.4  
## 3 40671 S05 Var01 70.6  
## 4 40672 S05 Var01 70.1  
## 5 40673 S05 Var01 69.8  
## 6 40676 S05 Var01 70.5  
## 7 40677 S05 Var01 70.0  
## 8 40678 S05 Var01 70.0  
## 9 40679 S05 Var01 69.7  
## 10 40680 S05 Var01 69.7  
## # ℹ 1,752 more rows  
##   
## $S06.Var01  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S06 Var01 27.5  
## 2 40670 S06 Var01 28.2  
## 3 40671 S06 Var01 28.3  
## 4 40672 S06 Var01 29.4  
## 5 40673 S06 Var01 29.4  
## 6 40676 S06 Var01 29.2  
## 7 40677 S06 Var01 28.5  
## 8 40678 S06 Var01 28.8  
## 9 40679 S06 Var01 29.2  
## 10 40680 S06 Var01 28.9  
## # ℹ 1,752 more rows  
##   
## $S01.Var02  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S01 Var02 10369300  
## 2 40670 S01 Var02 10943800  
## 3 40671 S01 Var02 8933800  
## 4 40672 S01 Var02 10775400  
## 5 40673 S01 Var02 12875600  
## 6 40676 S01 Var02 11677000  
## 7 40677 S01 Var02 21165300  
## 8 40678 S01 Var02 18809200  
## 9 40679 S01 Var02 22908400  
## 10 40680 S01 Var02 20359100  
## # ℹ 1,752 more rows  
##   
## $S02.Var02  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S02 Var02 60855800  
## 2 40670 S02 Var02 215620200  
## 3 40671 S02 Var02 200070600  
## 4 40672 S02 Var02 130201700  
## 5 40673 S02 Var02 130463000  
## 6 40676 S02 Var02 170626200  
## 7 40677 S02 Var02 162995900  
## 8 40678 S02 Var02 154527100  
## 9 40679 S02 Var02 116531200  
## 10 40680 S02 Var02 96149800  
## # ℹ 1,752 more rows  
##   
## $S03.Var02  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S03 Var02 123432400  
## 2 40670 S03 Var02 150476200  
## 3 40671 S03 Var02 138040000  
## 4 40672 S03 Var02 119282800  
## 5 40673 S03 Var02 111902700  
## 6 40676 S03 Var02 115557400  
## 7 40677 S03 Var02 148614900  
## 8 40678 S03 Var02 151473000  
## 9 40679 S03 Var02 108223500  
## 10 40680 S03 Var02 148516900  
## # ℹ 1,752 more rows  
##   
## $S04.Var02  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S04 Var02 16587400  
## 2 40670 S04 Var02 11718100  
## 3 40671 S04 Var02 16422000  
## 4 40672 S04 Var02 31816300  
## 5 40673 S04 Var02 15470000  
## 6 40676 S04 Var02 16181900  
## 7 40677 S04 Var02 15672400  
## 8 40678 S04 Var02 16955600  
## 9 40679 S04 Var02 16715600  
## 10 40680 S04 Var02 18415000  
## # ℹ 1,752 more rows  
##   
## $S05.Var02  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S05 Var02 27809100  
## 2 40670 S05 Var02 30174700  
## 3 40671 S05 Var02 35044700  
## 4 40672 S05 Var02 27192100  
## 5 40673 S05 Var02 24891800  
## 6 40676 S05 Var02 30685000  
## 7 40677 S05 Var02 31496700  
## 8 40678 S05 Var02 24884400  
## 9 40679 S05 Var02 18630800  
## 10 40680 S05 Var02 29411900  
## # ℹ 1,752 more rows  
##   
## $S06.Var02  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S06 Var02 39335700  
## 2 40670 S06 Var02 55416000  
## 3 40671 S06 Var02 33237000  
## 4 40672 S06 Var02 61649000  
## 5 40673 S06 Var02 35508700  
## 6 40676 S06 Var02 24387800  
## 7 40677 S06 Var02 39795000  
## 8 40678 S06 Var02 36715300  
## 9 40679 S06 Var02 31540700  
## 10 40680 S06 Var02 47255900  
## # ℹ 1,752 more rows  
##   
## $S01.Var03  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S01 Var03 25.9  
## 2 40670 S01 Var03 25.7  
## 3 40671 S01 Var03 25.6  
## 4 40672 S01 Var03 25.4  
## 5 40673 S01 Var03 25.5  
## 6 40676 S01 Var03 25.7  
## 7 40677 S01 Var03 24.7  
## 8 40678 S01 Var03 24.7  
## 9 40679 S01 Var03 24.5  
## 10 40680 S01 Var03 24.2  
## # ℹ 1,752 more rows  
##   
## $S02.Var03  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S02 Var03 10.0  
## 2 40670 S02 Var03 10.4  
## 3 40671 S02 Var03 11.1  
## 4 40672 S02 Var03 11.3  
## 5 40673 S02 Var03 11.5  
## 6 40676 S02 Var03 11.8  
## 7 40677 S02 Var03 11.7  
## 8 40678 S02 Var03 11.5  
## 9 40679 S02 Var03 11.5  
## 10 40680 S02 Var03 11.6  
## # ℹ 1,752 more rows  
##   
## $S03.Var03  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S03 Var03 30.3  
## 2 40670 S03 Var03 30.5  
## 3 40671 S03 Var03 30.1  
## 4 40672 S03 Var03 29.9  
## 5 40673 S03 Var03 29.9  
## 6 40676 S03 Var03 29.8  
## 7 40677 S03 Var03 29.5  
## 8 40678 S03 Var03 29.2  
## 9 40679 S03 Var03 29.9  
## 10 40680 S03 Var03 29.4  
## # ℹ 1,752 more rows  
##   
## $S04.Var03  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S04 Var03 16.9  
## 2 40670 S04 Var03 17   
## 3 40671 S04 Var03 17.1  
## 4 40672 S04 Var03 16.6  
## 5 40673 S04 Var03 16.6  
## 6 40676 S04 Var03 16.5  
## 7 40677 S04 Var03 16.6  
## 8 40678 S04 Var03 16.6  
## 9 40679 S04 Var03 16.8  
## 10 40680 S04 Var03 16.8  
## # ℹ 1,752 more rows  
##   
## $S05.Var03  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S05 Var03 68.2  
## 2 40670 S05 Var03 68.8  
## 3 40671 S05 Var03 69.3  
## 4 40672 S05 Var03 69.4  
## 5 40673 S05 Var03 69.2  
## 6 40676 S05 Var03 69.7  
## 7 40677 S05 Var03 69.5  
## 8 40678 S05 Var03 69.3  
## 9 40679 S05 Var03 69.3  
## 10 40680 S05 Var03 68.7  
## # ℹ 1,752 more rows  
##   
## $S06.Var03  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S06 Var03 26.8  
## 2 40670 S06 Var03 27.2  
## 3 40671 S06 Var03 27.8  
## 4 40672 S06 Var03 27.9  
## 5 40673 S06 Var03 28.6  
## 6 40676 S06 Var03 28.6  
## 7 40677 S06 Var03 27.5  
## 8 40678 S06 Var03 27.8  
## 9 40679 S06 Var03 28.5  
## 10 40680 S06 Var03 27.9  
## # ℹ 1,752 more rows  
##   
## $S01.Var05  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S01 Var05 26.2  
## 2 40670 S01 Var05 26.0  
## 3 40671 S01 Var05 25.9  
## 4 40672 S01 Var05 25.6  
## 5 40673 S01 Var05 25.6  
## 6 40676 S01 Var05 26.3  
## 7 40677 S01 Var05 26.0  
## 8 40678 S01 Var05 25.0  
## 9 40679 S01 Var05 24.9  
## 10 40680 S01 Var05 24.7  
## # ℹ 1,752 more rows  
##   
## $S02.Var05  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S02 Var05 10.2  
## 2 40670 S02 Var05 10.4  
## 3 40671 S02 Var05 11.2  
## 4 40672 S02 Var05 11.5  
## 5 40673 S02 Var05 11.7  
## 6 40676 S02 Var05 11.9  
## 7 40677 S02 Var05 12.0  
## 8 40678 S02 Var05 11.9  
## 9 40679 S02 Var05 11.6  
## 10 40680 S02 Var05 11.7  
## # ℹ 1,752 more rows  
##   
## $S03.Var05  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S03 Var05 30.5  
## 2 40670 S03 Var05 30.7  
## 3 40671 S03 Var05 30.6  
## 4 40672 S03 Var05 30.2  
## 5 40673 S03 Var05 30.0  
## 6 40676 S03 Var05 30.4  
## 7 40677 S03 Var05 29.9  
## 8 40678 S03 Var05 29.7  
## 9 40679 S03 Var05 30.0  
## 10 40680 S03 Var05 30.1  
## # ℹ 1,752 more rows  
##   
## $S04.Var05  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S04 Var05 16.9  
## 2 40670 S04 Var05 17.2  
## 3 40671 S04 Var05 17.2  
## 4 40672 S04 Var05 16.8  
## 5 40673 S04 Var05 16.7  
## 6 40676 S04 Var05 16.8  
## 7 40677 S04 Var05 16.6  
## 8 40678 S04 Var05 16.9  
## 9 40679 S04 Var05 16.8  
## 10 40680 S04 Var05 17.2  
## # ℹ 1,752 more rows  
##   
## $S05.Var05  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S05 Var05 68.7  
## 2 40670 S05 Var05 69.2  
## 3 40671 S05 Var05 69.4  
## 4 40672 S05 Var05 69.9  
## 5 40673 S05 Var05 69.7  
## 6 40676 S05 Var05 69.9  
## 7 40677 S05 Var05 69.7  
## 8 40678 S05 Var05 70.0  
## 9 40679 S05 Var05 69.5  
## 10 40680 S05 Var05 69.7  
## # ℹ 1,752 more rows  
##   
## $S06.Var05  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S06 Var05 27.0  
## 2 40670 S06 Var05 27.3  
## 3 40671 S06 Var05 28.0  
## 4 40672 S06 Var05 28.1  
## 5 40673 S06 Var05 28.9  
## 6 40676 S06 Var05 29.1  
## 7 40677 S06 Var05 28.5  
## 8 40678 S06 Var05 28.0  
## 9 40679 S06 Var05 28.5  
## 10 40680 S06 Var05 28.8  
## # ℹ 1,752 more rows  
##   
## $S01.Var07  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S01 Var07 26.0  
## 2 40670 S01 Var07 25.9  
## 3 40671 S01 Var07 25.7  
## 4 40672 S01 Var07 25.8  
## 5 40673 S01 Var07 26.3  
## 6 40676 S01 Var07 26   
## 7 40677 S01 Var07 24.9  
## 8 40678 S01 Var07 25.0  
## 9 40679 S01 Var07 24.7  
## 10 40680 S01 Var07 24.5  
## # ℹ 1,752 more rows  
##   
## $S02.Var07  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S02 Var07 10.3  
## 2 40670 S02 Var07 11.0  
## 3 40671 S02 Var07 11.4  
## 4 40672 S02 Var07 11.7  
## 5 40673 S02 Var07 11.7  
## 6 40676 S02 Var07 12.1  
## 7 40677 S02 Var07 11.9  
## 8 40678 S02 Var07 11.7  
## 9 40679 S02 Var07 11.8  
## 10 40680 S02 Var07 11.6  
## # ℹ 1,752 more rows  
##   
## $S03.Var07  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S03 Var07 30.6  
## 2 40670 S03 Var07 30.6  
## 3 40671 S03 Var07 30.1  
## 4 40672 S03 Var07 30.1  
## 5 40673 S03 Var07 30.3  
## 6 40676 S03 Var07 30.0  
## 7 40677 S03 Var07 29.7  
## 8 40678 S03 Var07 30.1  
## 9 40679 S03 Var07 29.9  
## 10 40680 S03 Var07 29.4  
## # ℹ 1,752 more rows  
##   
## $S04.Var07  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S04 Var07 17.1  
## 2 40670 S04 Var07 17.2  
## 3 40671 S04 Var07 17.2  
## 4 40672 S04 Var07 16.7  
## 5 40673 S04 Var07 16.7  
## 6 40676 S04 Var07 16.7  
## 7 40677 S04 Var07 16.7  
## 8 40678 S04 Var07 16.9  
## 9 40679 S04 Var07 17.1  
## 10 40680 S04 Var07 16.8  
## # ℹ 1,752 more rows  
##   
## $S05.Var07  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S05 Var07 69.2  
## 2 40670 S05 Var07 69.4  
## 3 40671 S05 Var07 70.0  
## 4 40672 S05 Var07 69.8  
## 5 40673 S05 Var07 69.5  
## 6 40676 S05 Var07 70.3  
## 7 40677 S05 Var07 69.9  
## 8 40678 S05 Var07 69.7  
## 9 40679 S05 Var07 69.7  
## 10 40680 S05 Var07 69.1  
## # ℹ 1,752 more rows  
##   
## $S06.Var07  
## # A tibble: 1,762 × 4  
## SeriesInd category key value  
## <dbl> <chr> <chr> <dbl>  
## 1 40669 S06 Var07 27.3  
## 2 40670 S06 Var07 28.1  
## 3 40671 S06 Var07 28.1  
## 4 40672 S06 Var07 29.1  
## 5 40673 S06 Var07 28.9  
## 6 40676 S06 Var07 28.8  
## 7 40677 S06 Var07 28.1  
## 8 40678 S06 Var07 28.6  
## 9 40679 S06 Var07 29.0  
## 10 40680 S06 Var07 28.1  
## # ℹ 1,752 more rows

myts is now a list of clean time series objects – all the ones we need to forecast for

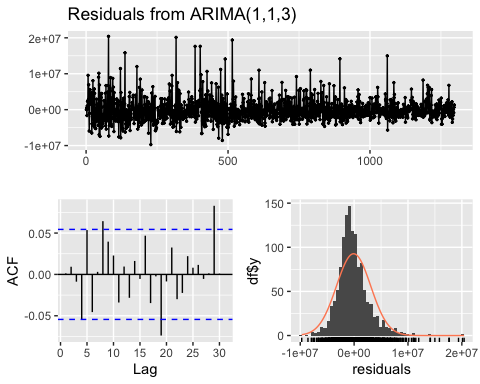


Using the funciton ndiffs we can see that each series requries 1 differencing to become stationary.

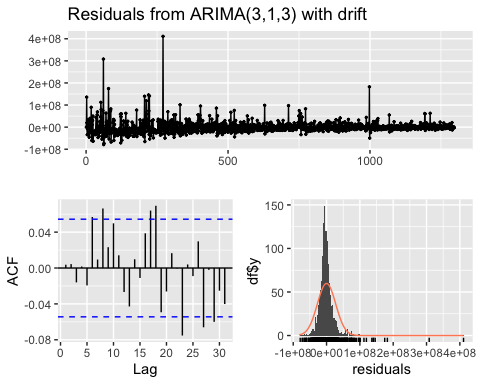




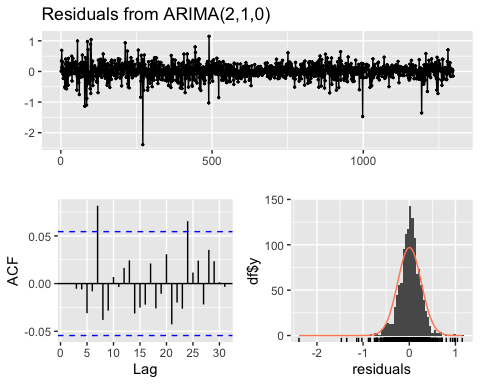
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1) with drift  
## Q\* = 11.711, df = 9, p-value = 0.2301  
##   
## Model df: 1. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1) with drift  
## Q\* = 11.711, df = 9, p-value = 0.2301



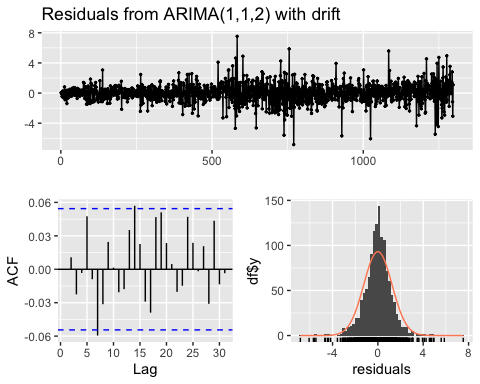
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 18.686, df = 6, p-value = 0.004728  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 18.686, df = 6, p-value = 0.004728



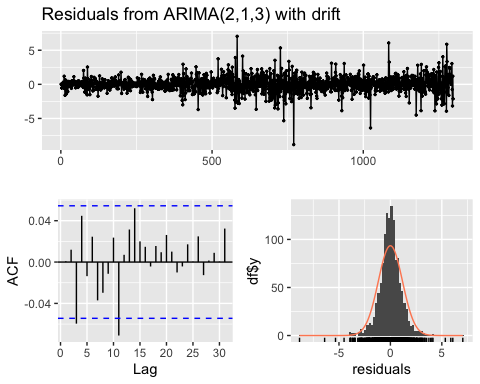
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,1,3) with drift  
## Q\* = 14.923, df = 4, p-value = 0.004863  
##   
## Model df: 6. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,1,3) with drift  
## Q\* = 14.923, df = 4, p-value = 0.004863



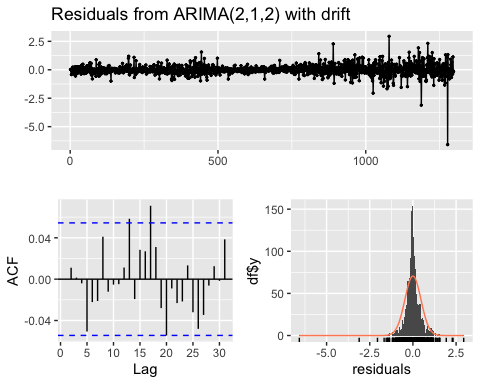
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,0)  
## Q\* = 13.102, df = 8, p-value = 0.1084  
##   
## Model df: 2. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,0)  
## Q\* = 13.102, df = 8, p-value = 0.1084



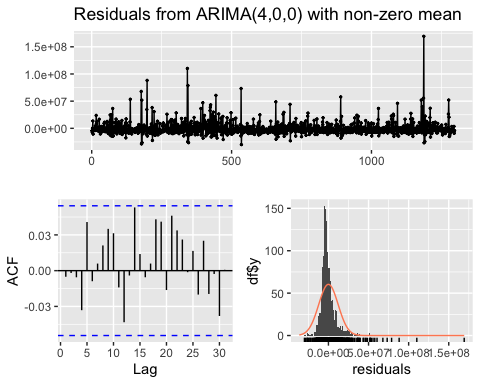
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,2) with drift  
## Q\* = 10.572, df = 7, p-value = 0.1584  
##   
## Model df: 3. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,2) with drift  
## Q\* = 10.572, df = 7, p-value = 0.1584



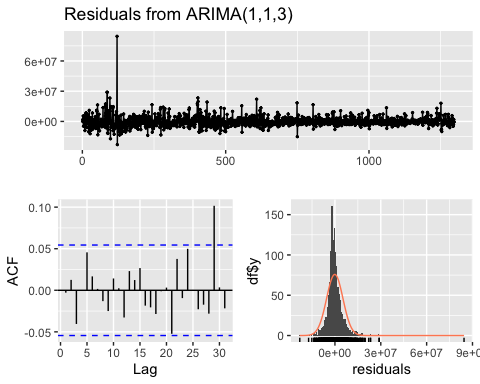
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,3) with drift  
## Q\* = 12.289, df = 5, p-value = 0.03103  
##   
## Model df: 5. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,3) with drift  
## Q\* = 12.289, df = 5, p-value = 0.03103



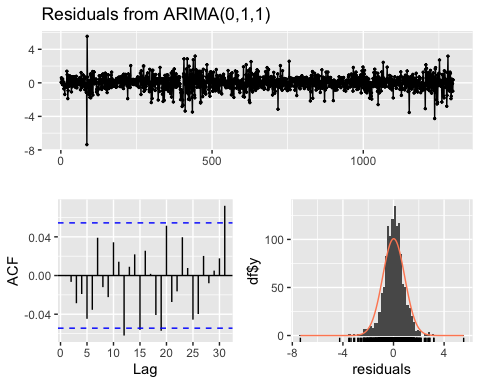
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 7.1813, df = 6, p-value = 0.3044  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 7.1813, df = 6, p-value = 0.3044



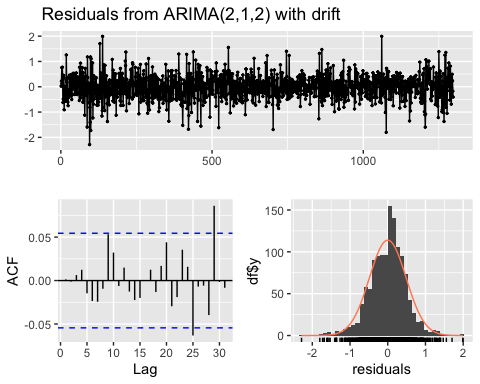
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,0,0) with non-zero mean  
## Q\* = 7.3155, df = 6, p-value = 0.2926  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,0,0) with non-zero mean  
## Q\* = 7.3155, df = 6, p-value = 0.2926



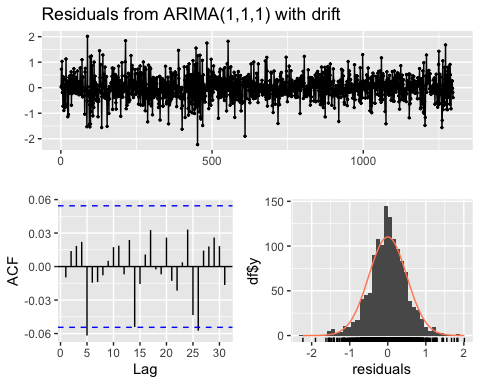
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 6.7384, df = 6, p-value = 0.3457  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 6.7384, df = 6, p-value = 0.3457



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1)  
## Q\* = 10.229, df = 9, p-value = 0.3323  
##   
## Model df: 1. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1)  
## Q\* = 10.229, df = 9, p-value = 0.3323



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 7.2131, df = 6, p-value = 0.3016  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 7.2131, df = 6, p-value = 0.3016



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1) with drift  
## Q\* = 7.464, df = 8, p-value = 0.4875  
##   
## Model df: 2. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1) with drift  
## Q\* = 7.464, df = 8, p-value = 0.4875

## list\_vars SES\_MAPE Holt\_MAPE Arima\_MAPE  
## [1,] "S01.Var01" "9%" "10%" "16%"   
## [2,] "S01.Var02" "30%" "30%" "33%"   
## [3,] "S02.Var02" "30%" "30%" "52%"   
## [4,] "S02.Var03" "16%" "16%" "16%"   
## [5,] "S03.Var05" "16%" "20%" "28%"   
## [6,] "S03.Var07" "15%" "18%" "26%"   
## [7,] "S04.Var01" "23%" "23%" "33%"   
## [8,] "S04.Var02" "28%" "28%" "75%"   
## [9,] "S05.Var02" "26%" "26%" "32%"   
## [10,] "S05.Var03" "9%" "9%" "9%"   
## [11,] "S06.Var05" "6%" "6%" "11%"   
## [12,] "S06.Var07" "6%" "6%" "11%"

list\_vars

SES\_MAPE

Holt\_MAPE

Arima\_MAPE

S01.Var01

9%

10%

16%

S01.Var02

30%

30%

33%

S02.Var02

30%

30%

52%

S02.Var03

16%

16%

16%

S03.Var05

16%

20%

28%

S03.Var07

15%

18%

26%

S04.Var01

23%

23%

33%

S04.Var02

28%

28%

75%

S05.Var02

26%

26%

32%

S05.Var03

9%

9%

9%

S06.Var05

6%

6%

11%

S06.Var07

6%

6%

11%

knitr::opts\_chunk$set(fig.path='Figs/', echo=FALSE, warning=FALSE, message=FALSE, cache=TRUE)  
library(httr)  
library(kableExtra)  
library(fpp2)  
library(imputeTS)  
library(tidyverse)  
library(urca)  
library(ggfortify)  
library(gridExtra)  
library(scales)  
set.seed(123)  
  
github\_link <- "https://github.com/klgriffen96/summer23\_data624/raw/main/project\_1/Data%20Set%20for%20Class.xls"  
temp\_file <- tempfile(fileext = ".xls")  
req <- GET(github\_link,   
 # write result to disk  
 write\_disk(path = temp\_file))  
  
df <- readxl::read\_excel(temp\_file)  
  
head(df, 10) |>  
 kbl(caption = "Series Provided") |>  
 kable\_classic(full\_width = F, html\_font = "Cambria")  
  
# where given ends and predict starts on next  
break\_given\_predict <- 43021  
  
df\_all\_given <- df |> filter(SeriesInd <= break\_given\_predict)  
df\_predict <- df |> filter(SeriesInd > break\_given\_predict)  
  
df\_all\_given <- split(df\_all\_given, f = df$category)  
df\_predict <- split(df\_predict, f = df$category)  
  
head(df\_all\_given$S01, 5) |>  
 kbl(caption = "S01") |>  
 kable\_classic(full\_width = F, html\_font = "Cambria")  
df\_long <- df %>% gather(key, value, -SeriesInd, -category)  
split\_data <- split(df\_long, f=list(df\_long$category, df\_long$key))  
  
#Split the data into data frames by category and var name  
split\_data  
  
#put the combo into a list so it can be run through  
list\_vars <- c(  
 "S01.Var01", "S01.Var02",  
 "S02.Var02", "S02.Var03",  
 "S03.Var05", "S03.Var07",  
 "S04.Var01", "S04.Var02",  
 "S05.Var02", "S05.Var03",  
 "S06.Var05", "S06.Var07"  
)  
  
  
#select list items based on the list vars and then turn each list item into a clean ts  
myts <- lapply(split\_data[list\_vars], function(x) {  
 x %>%   
 dplyr::select(value) %>%   
 slice(1:1622) %>% #removes the missing values we need to predict  
 ts() %>%   
 tsclean(lambda = "auto") %>%  
 na\_ma()   
})  
  
  
for (i in 1:length(list\_vars)){  
 var <- list\_vars[i]  
 ts <- myts[[var]]   
   
 plot <- autoplot(ts) +  
 ggtitle(paste("Plot for", var))  
   
  
 print(plot)  
}  
  
test\_split <- function(x){  
 # Determine the index to split the time series into train and test sets  
 split.index <- floor(0.8 \* length(ts)) # 80% for training, 20% for testing  
   
 # Split the time series into train and test sets  
 train <- window(ts, end = split.index)  
 test <- window(ts, start = split.index + 1)  
   
 # Set the horizon  
 horizon <- length(test)  
  
 return(list(train, test, horizon))  
}  
# Create the empty vectors  
smooth\_results <- vector(mode = "list", length = length(myts))  
ses\_MAPE <- vector("numeric", length = length(list\_vars))  
holt\_MAPE <- vector("numeric", length = length(list\_vars))  
ses\_p <- vector("numeric", length = length(list\_vars))  
holt\_p <- vector("numeric", length = length(list\_vars))  
  
# Create a function to test the ses and holt forecasts with test and train data  
ses\_test <- function(x, i) {  
 #ts <- diff(x)  
 ts <- x  
 # Determine the index to split the time series into train and test sets  
 split.index <- floor(0.8 \* length(ts)) # 80% for training, 20% for testing  
   
 # Split the time series into train and test sets  
 train <- window(ts, end = split.index)  
 test <- window(ts, start = split.index + 1)  
   
 # Set the horizon  
 horizon <- length(test)  
   
 # Ses fit with training data  
 ses.fit <- ses(train, h = horizon)  
 ses.p <- Box.test(residuals(ses.fit))$p.value  
   
 # Test with test data  
 ses\_res <- accuracy(ses.fit, test)['Test set', 'MAPE']  
   
  
 # Holt fit with training data  
 holt.fit <- holt(train, damped = TRUE, h = horizon)  
 holt.p <- Box.test(residuals(holt.fit))$p.value  
  
 # Test with test data  
 holt\_res <- accuracy(holt.fit, test)['Test set', 'MAPE']  
   
 result <- list(ses\_MAPE = ses\_res,   
 ses.p = ses.p,   
 holt\_MAPE = holt\_res,   
 holt.p=holt.p)  
   
 # plot  
 p <- autoplot(train) +  
 autolayer(ses.fit, series = "ses") +  
 autolayer(holt.fit, alpha = 0.4, series = "holt") +  
 autolayer(test, series = "test data") +  
 ggtitle(list\_vars[i])  
   
 print(p)  
   
 return(result)  
}  
  
for (i in seq\_along(myts)) {  
 result<- ses\_test(myts[[i]], i)  
 ses\_MAPE[i] <- result[1]  
 ses\_p[i] <- result[2]  
 holt\_MAPE[i] <- result[3]  
 holt\_p[i] <- result[4]  
}  
  
  
#apply scales::percent for formatting  
SES\_MAPE <- lapply(ses\_MAPE, function(x) percent(x, scale = 1))  
Holt\_MAPE <- lapply(holt\_MAPE, function(x) percent(x, scale = 1))  
# Create the empty vectors  
arima\_MAPE <- vector(mode = "list", length = length(myts))  
  
# Create a function to test the ses and holt forecasts with test and train data  
arima\_test <- function(x, i) {  
   
 # Determine the index to split the time series into train and test sets  
 split.index <- floor(0.8 \* length(x)) # 80% for training, 20% for testing  
   
 # Split the time series into train and test sets  
 train <- window(x, end = split.index)  
 test <- window(x, start = split.index + 1)  
   
 # Set the horizon  
 horizon <- length(test)  
   
 #auto arima fit  
 arima.fc <- train %>%   
 auto.arima() %>%  
 forecast(h=horizon)  
   
 # test results  
 result <- accuracy(arima.fc, test)['Test set', 'MAPE']  
  
   
 p <- checkresiduals(arima.fc)  
  
 print(p)  
   
 return(result)  
}  
  
  
## Run   
for (i in seq\_along(myts)) {  
 arima\_MAPE[i] <- arima\_test(myts[[i]], i)  
}  
  
Arima\_MAPE <- lapply(arima\_MAPE, function(x) percent(x, scale = 1))  
results\_df <- cbind(  
 list\_vars,  
 SES\_MAPE,  
 Holt\_MAPE,  
 Arima\_MAPE  
)  
  
# Print the results as a table  
print(results\_df, format = "markdown", digits = 2) %>%   
 kable()