Project\_1\_compiled

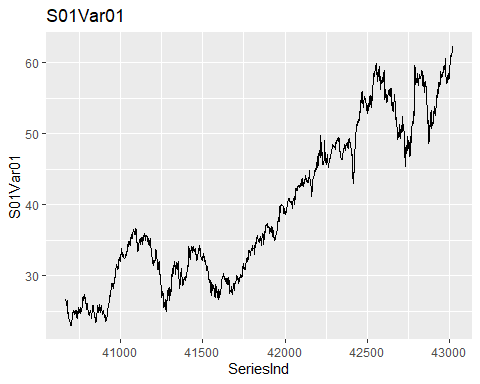
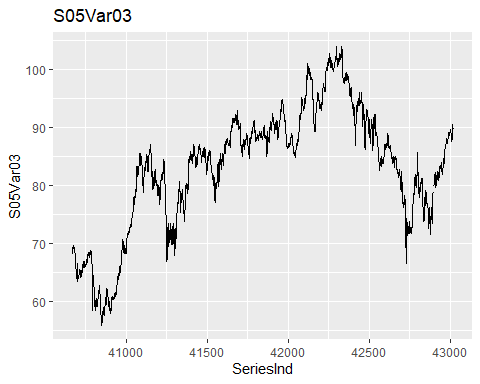
Samuel Kigamba, Lin Li, Patrick Maloney, Daniel Moscoe, David Moste

6/27/2021

### Introduction

This project was completed by Samuel Kigamba, Lin Li, Patrick Maloney, Daniel Moscoe, and David Moste. In this assignment, we were given a collection of de-identified timeseries and instructed to forecast 140 periods into the future. The fact that no context about the data-generation process was given meant that we would have to fully rely on our technical skills to identify the correct approach for the forecasts.

### Unraveling the Mystery

After looking at the visualizations and running some tests, we came to the conclusion that this appears to be stock market data, with each variable possibly representing a different stock price over time. We came to this conclusion due to the way the points were generally close together in value over adjacent periods, but appeared to also be occurring in any direction (up or down) on any given period . Those in the data science world will recognise this as a random walk, meaning that the value for one period doesn’t necessarily affect the value for the following period. See the example below for a couple of our variables, which all had these properties in common. 

As we can see from these examples, the data appears to be a random walk and has a trend but no clear seasonality or cylclicity. This is generally typical of stock market data.

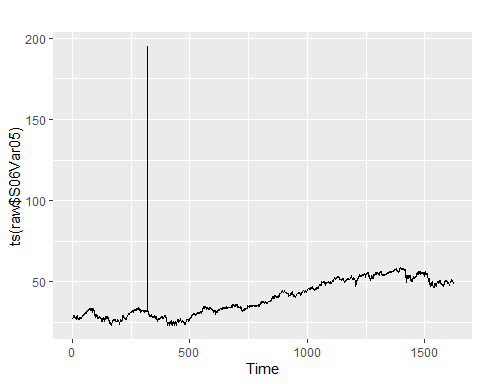
The second clue was that the variable SeriesInd contains numerous gaps where data is missing. However, our analysis showed that these gaps occurred at regular intervals and the differences in values between periods did not suggest a missing data. In fact, the gaps usually occurred after five consecutive periods, with the typical gap lasting two periods, along with other single gaps added in intermittently. This pattern closely resembles the stock market schedule of operation, which is closed on weekends and holidays. Therefore, we continued with the analysis with the understanding that we were likely dealing with stock market data.

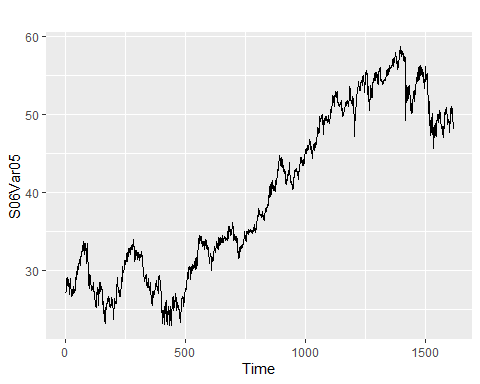
### Methodology

Since all the variables we were assigned to analyze had very similar attributes, and we assume that they all demonstrate a stock price over time, we followe a general methodology for producing each forecast, running the unique data through each step. In some cases, the varaibles were so similar that a single model fit each stock. In this space, our general methodolgy will be presented, but for further technical details on each stock forecast, please see the Appendix.

##### Exploratory Visualization

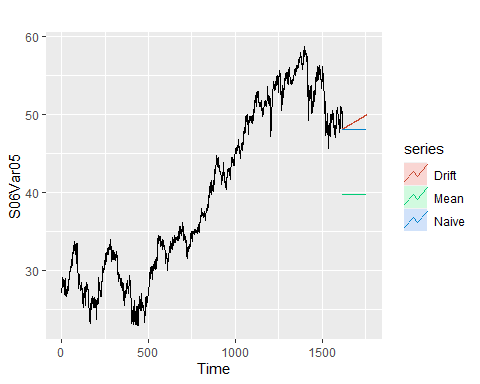
Once again, we looked at each timeseries to determine the best course of action for our forecast. Below is an example for S06Var05:

 We can see a clear outlier that had to be removed to be able to effectively analyze the rest of the series.

 Now that the outliers are removed, we can see that the points are reasonably close together in sequential periods for the most part and we can comfortably say that the lack of any clear pattern suggests this is a random walk.

#### Simple Forecasts

The next step is to produce a simple random walk forecast, a naive model that uses the previous period to predict the value of the following period.



Here is our example of our baseline Naive random walk forecast model, with the mean models and drift models included for comparison. For those not familiar, a random walk with drift essentially allows the forecast to follow the trend line, while the mean just takes the average value.

Random walks such as these are notoriously challenging to forecast. We will examine several possible methods to improve on these forecasts with more sophisticated techniques, but if the performance is nearly the same, we will favor the simpler model, as these often perform better in the wild over longer periods.

Additional models were fit with various techniques including simple exponential smoothing (SES), a technique that utilizes moving averages to forecast along a trend line, a Holt method, and ARIMA models, which allow for auto-regression on non-stationary data.

In the end, however, these techniques largely failed to significantly outperform the random walk forecasts without drift. There were some cases where the SES method marginally outperformed the naive method, but not by enough to justify adopting the more complicated method, generally speaking. On a case by case basis, other models were sometimes used, but the gains were marginal over the naive model. For full statistical analysis of each forecast, please see the Appendix.

### Results

Here are the first few values of the final forecasts selected for each variable:

## ï..SereisInd S01Var01 S01Var02 S02Var02 S02Var03 S03Var05 S03Var07 S04Var01  
## 1 43022 62.31 5876662 24203892 13.07357 98.66671 97.34015 36.91  
## 2 43023 62.31 5640500 27105957 13.07357 98.66671 97.34015 36.91  
## 3 43024 62.31 5554749 27660981 13.07357 98.66671 97.34015 36.91  
## 4 43025 62.31 5509240 27805714 13.07357 98.66671 97.34015 36.91  
## 5 43028 62.31 5475856 28477919 13.07357 98.66671 97.34015 36.91  
## 6 43029 62.31 5447280 28422323 13.07357 98.66671 97.34015 36.91  
## S04Var02 S05Var S05Var.1 S06Var S06Var.1  
## 1 6519800 11370000 89.7 48.13 47.97  
## 2 6519800 11370000 89.7 48.13 47.97  
## 3 6519800 11370000 89.7 48.13 47.97  
## 4 6519800 11370000 89.7 48.13 47.97  
## 5 6519800 11370000 89.7 48.13 47.97  
## 6 6519800 11370000 89.7 48.13 47.97

As we can see in the table above, all but two ended up using the naive random walk method.

## Appendix

Below is all code and in depth analysis for this project

s01\_v01\_ts <- ts(window(raw$S01Var01, end = 1622))  
s01\_v02\_ts <- ts(window(raw$S01Var02, end = 1622))  
s02\_v02\_ts <- ts(window(raw$S02Var02, end = 1622))  
  
# Check for NAs and general data summary  
summary(s01\_v01\_ts)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 23.01 29.85 35.66 39.41 48.70 62.31 2

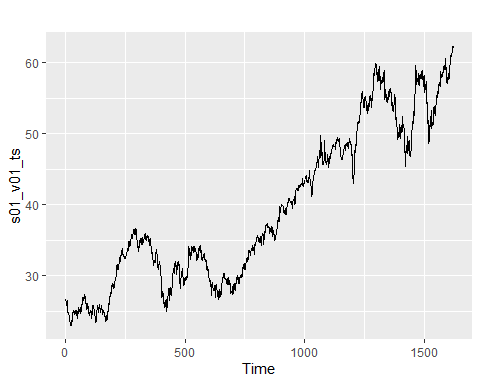
summary(s01\_v02\_ts)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1339900 5347550 7895050 8907092 11321675 48477500

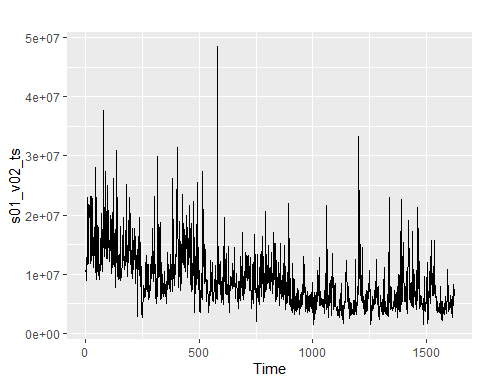
summary(s02\_v02\_ts)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 7128800 27880300 39767500 50633098 59050900 480879500

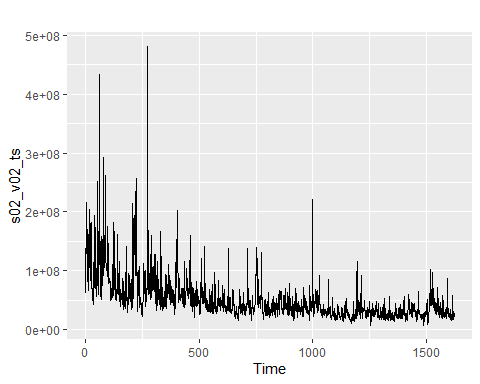
# Impute missing values from s01\_v01  
s01\_v01\_ts <- forecast::na.interp(s01\_v01\_ts)  
  
# Trying to get a visual of the data  
autoplot(s01\_v01\_ts)



autoplot(s01\_v02\_ts)



autoplot(s02\_v02\_ts)



# Checking if volume changes significantly after gaps  
gaps <- diff(raw$SeriesInd) > 1  
gaps <- c(FALSE, gaps)  
gaps\_df <- data.frame("SeriesInd" = raw$SeriesInd, "AfterGap" = gaps, "S01V02" = raw$S01Var02)  
  
after\_gap <- gaps\_df %>%  
 filter(AfterGap) %>%  
 drop\_na()  
all\_others <- gaps\_df %>%  
 filter(AfterGap == FALSE) %>%  
 drop\_na()  
  
gap\_comparison <- data.frame("Situation" = c("After Gap", "Others"),  
 "Mean" = c(mean(after\_gap$S01V02),mean(all\_others$S01V02)),  
 "sd" = c(sd(after\_gap$S01V02),sd(all\_others$S01V02)),  
 "Variance" = c(var(after\_gap$S01V02),var(all\_others$S01V02)))

# Check performance of random walk  
rwf\_nodrift <- tsCV(s01\_v01\_ts, rwf, drift = FALSE, h = 1)  
rmse\_rwf\_nodrift <- sqrt(mean(rwf\_nodrift^2, na.rm = TRUE))  
rwf\_drift <- tsCV(s01\_v01\_ts, rwf, drift = TRUE, h = 1)  
rmse\_rwf\_drift <- sqrt(mean(rwf\_drift^2, na.rm = TRUE))  
meanf <- tsCV(s01\_v01\_ts, meanf, h = 1)  
rmse\_meanf <- sqrt(mean(meanf^2, na.rm = TRUE))

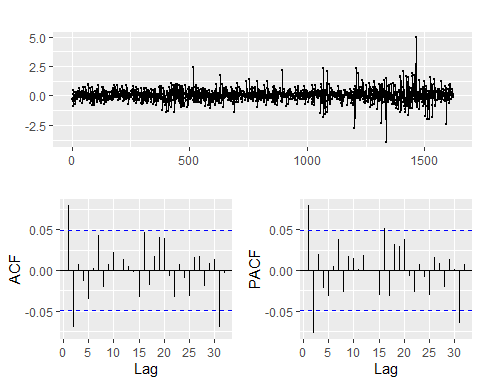
# The random walk with no drift has the lowest rmse  
  
# Try ses  
s01\_v01\_ses <- ses(s01\_v01\_ts, h = 140)  
summary(s01\_v01\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = s01\_v01\_ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.9999   
##   
## Initial states:  
## l = 26.6152   
##   
## sigma: 0.5144  
##   
## AIC AICc BIC   
## 9836.684 9836.699 9852.859   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.02200885 0.5141259 0.3498239 0.04403019 0.9147055 0.9994088  
## ACF1  
## Training set 0.07947628  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1623 62.30999 61.65070 62.96927 61.30170 63.31828  
## 1624 62.30999 61.37766 63.24231 60.88412 63.73585  
## 1625 62.30999 61.16815 63.45183 60.56369 64.05628  
## 1626 62.30999 60.99152 63.62846 60.29356 64.32642  
## 1627 62.30999 60.83590 63.78408 60.05556 64.56441  
## 1628 62.30999 60.69521 63.92477 59.84040 64.77958  
## 1629 62.30999 60.56583 64.05414 59.64253 64.97744  
## 1630 62.30999 60.44541 64.17457 59.45836 65.16161  
## 1631 62.30999 60.33231 64.28767 59.28539 65.33459  
## 1632 62.30999 60.22533 64.39464 59.12178 65.49819  
## 1633 62.30999 60.12358 64.49639 58.96617 65.65380  
## 1634 62.30999 60.02637 64.59361 58.81749 65.80249  
## 1635 62.30999 59.93312 64.68686 58.67488 65.94509  
## 1636 62.30999 59.84340 64.77658 58.53766 66.08231  
## 1637 62.30999 59.75682 64.86315 58.40526 66.21471  
## 1638 62.30999 59.67309 64.94688 58.27721 66.34277  
## 1639 62.30999 59.59194 65.02804 58.15309 66.46688  
## 1640 62.30999 59.51314 65.10684 58.03258 66.58740  
## 1641 62.30999 59.43650 65.18347 57.91537 66.70461  
## 1642 62.30999 59.36185 65.25812 57.80121 66.81877  
## 1643 62.30999 59.28905 65.33093 57.68986 66.93011  
## 1644 62.30999 59.21796 65.40202 57.58114 67.03884  
## 1645 62.30999 59.14847 65.47151 57.47486 67.14511  
## 1646 62.30999 59.08047 65.53950 57.37087 67.24911  
## 1647 62.30999 59.01388 65.60610 57.26902 67.35095  
## 1648 62.30999 58.94860 65.67137 57.16919 67.45078  
## 1649 62.30999 58.88457 65.73541 57.07126 67.54871  
## 1650 62.30999 58.82171 65.79826 56.97513 67.64484  
## 1651 62.30999 58.75997 65.86001 56.88070 67.73927  
## 1652 62.30999 58.69928 65.92069 56.78789 67.83209  
## 1653 62.30999 58.63960 65.98038 56.69661 67.92337  
## 1654 62.30999 58.58087 66.03911 56.60679 68.01319  
## 1655 62.30999 58.52305 66.09693 56.51836 68.10161  
## 1656 62.30999 58.46610 66.15388 56.43127 68.18871  
## 1657 62.30999 58.40998 66.20999 56.34544 68.27453  
## 1658 62.30999 58.35466 66.26532 56.26084 68.35914  
## 1659 62.30999 58.30010 66.31987 56.17740 68.44258  
## 1660 62.30999 58.24628 66.37370 56.09508 68.52490  
## 1661 62.30999 58.19315 66.42682 56.01383 68.60614  
## 1662 62.30999 58.14071 66.47927 55.93362 68.68635  
## 1663 62.30999 58.08891 66.53106 55.85441 68.76556  
## 1664 62.30999 58.03775 66.58223 55.77616 68.84382  
## 1665 62.30999 57.98719 66.63279 55.69883 68.92114  
## 1666 62.30999 57.93721 66.68277 55.62240 68.99757  
## 1667 62.30999 57.88780 66.73218 55.54683 69.07314  
## 1668 62.30999 57.83893 66.78104 55.47210 69.14787  
## 1669 62.30999 57.79060 66.82938 55.39818 69.22180  
## 1670 62.30999 57.74277 66.87720 55.32503 69.29494  
## 1671 62.30999 57.69544 66.92453 55.25265 69.36733  
## 1672 62.30999 57.64859 66.97138 55.18100 69.43898  
## 1673 62.30999 57.60221 67.01777 55.11006 69.50991  
## 1674 62.30999 57.55628 67.06370 55.03982 69.58016  
## 1675 62.30999 57.51079 67.10919 54.97025 69.64973  
## 1676 62.30999 57.46572 67.15425 54.90133 69.71865  
## 1677 62.30999 57.42108 67.19890 54.83304 69.78693  
## 1678 62.30999 57.37683 67.24314 54.76538 69.85460  
## 1679 62.30999 57.33298 67.28699 54.69831 69.92166  
## 1680 62.30999 57.28951 67.33046 54.63183 69.98814  
## 1681 62.30999 57.24642 67.37356 54.56593 70.05405  
## 1682 62.30999 57.20369 67.41629 54.50057 70.11940  
## 1683 62.30999 57.16131 67.45867 54.43577 70.18421  
## 1684 62.30999 57.11928 67.50070 54.37149 70.24849  
## 1685 62.30999 57.07759 67.54239 54.30772 70.31225  
## 1686 62.30999 57.03622 67.58375 54.24446 70.37551  
## 1687 62.30999 56.99518 67.62479 54.18169 70.43828  
## 1688 62.30999 56.95446 67.66552 54.11941 70.50057  
## 1689 62.30999 56.91404 67.70594 54.05759 70.56238  
## 1690 62.30999 56.87392 67.74606 53.99623 70.62374  
## 1691 62.30999 56.83409 67.78588 53.93533 70.68465  
## 1692 62.30999 56.79455 67.82542 53.87486 70.74512  
## 1693 62.30999 56.75530 67.86468 53.81482 70.80515  
## 1694 62.30999 56.71632 67.90366 53.75521 70.86477  
## 1695 62.30999 56.67761 67.94237 53.69600 70.92397  
## 1696 62.30999 56.63916 67.98082 53.63720 70.98277  
## 1697 62.30999 56.60097 68.01900 53.57880 71.04117  
## 1698 62.30999 56.56304 68.05694 53.52079 71.09919  
## 1699 62.30999 56.52535 68.09462 53.46315 71.15682  
## 1700 62.30999 56.48791 68.13206 53.40589 71.21409  
## 1701 62.30999 56.45071 68.16927 53.34899 71.27098  
## 1702 62.30999 56.41374 68.20623 53.29246 71.32752  
## 1703 62.30999 56.37701 68.24297 53.23627 71.38370  
## 1704 62.30999 56.34049 68.27948 53.18044 71.43954  
## 1705 62.30999 56.30421 68.31577 53.12494 71.49504  
## 1706 62.30999 56.26813 68.35184 53.06977 71.55021  
## 1707 62.30999 56.23228 68.38770 53.01493 71.60504  
## 1708 62.30999 56.19663 68.42335 52.96041 71.65956  
## 1709 62.30999 56.16119 68.45879 52.90621 71.71376  
## 1710 62.30999 56.12595 68.49402 52.85232 71.76765  
## 1711 62.30999 56.09092 68.52906 52.79874 71.82124  
## 1712 62.30999 56.05608 68.56390 52.74545 71.87452  
## 1713 62.30999 56.02143 68.59855 52.69247 71.92751  
## 1714 62.30999 55.98697 68.63301 52.63977 71.98021  
## 1715 62.30999 55.95270 68.66728 52.58735 72.03262  
## 1716 62.30999 55.91861 68.70136 52.53522 72.08476  
## 1717 62.30999 55.88470 68.73527 52.48336 72.13661  
## 1718 62.30999 55.85098 68.76900 52.43178 72.18819  
## 1719 62.30999 55.81742 68.80255 52.38047 72.23951  
## 1720 62.30999 55.78404 68.83593 52.32941 72.29056  
## 1721 62.30999 55.75083 68.86915 52.27862 72.34135  
## 1722 62.30999 55.71779 68.90219 52.22809 72.39189  
## 1723 62.30999 55.68491 68.93507 52.17780 72.44217  
## 1724 62.30999 55.65219 68.96779 52.12777 72.49221  
## 1725 62.30999 55.61963 69.00034 52.07797 72.54200  
## 1726 62.30999 55.58724 69.03274 52.02843 72.59155  
## 1727 62.30999 55.55499 69.06498 51.97911 72.64086  
## 1728 62.30999 55.52290 69.09707 51.93003 72.68994  
## 1729 62.30999 55.49096 69.12901 51.88119 72.73879  
## 1730 62.30999 55.45917 69.16080 51.83257 72.78741  
## 1731 62.30999 55.42753 69.19245 51.78417 72.83580  
## 1732 62.30999 55.39603 69.22395 51.73600 72.88398  
## 1733 62.30999 55.36467 69.25530 51.68805 72.93193  
## 1734 62.30999 55.33346 69.28652 51.64031 72.97967  
## 1735 62.30999 55.30238 69.31759 51.59278 73.02720  
## 1736 62.30999 55.27144 69.34853 51.54546 73.07451  
## 1737 62.30999 55.24064 69.37934 51.49835 73.12162  
## 1738 62.30999 55.20997 69.41001 51.45145 73.16853  
## 1739 62.30999 55.17943 69.44054 51.40474 73.21523  
## 1740 62.30999 55.14903 69.47095 51.35824 73.26174  
## 1741 62.30999 55.11875 69.50123 51.31193 73.30804  
## 1742 62.30999 55.08859 69.53138 51.26582 73.35416  
## 1743 62.30999 55.05857 69.56141 51.21990 73.40008  
## 1744 62.30999 55.02866 69.59131 51.17416 73.44581  
## 1745 62.30999 54.99888 69.62109 51.12862 73.49136  
## 1746 62.30999 54.96922 69.65075 51.08326 73.53672  
## 1747 62.30999 54.93968 69.68029 51.03808 73.58190  
## 1748 62.30999 54.91026 69.70971 50.99308 73.62689  
## 1749 62.30999 54.88096 69.73902 50.94826 73.67171  
## 1750 62.30999 54.85176 69.76821 50.90362 73.71636  
## 1751 62.30999 54.82269 69.79729 50.85915 73.76083  
## 1752 62.30999 54.79372 69.82625 50.81485 73.80512  
## 1753 62.30999 54.76487 69.85511 50.77073 73.84925  
## 1754 62.30999 54.73613 69.88385 50.72677 73.89321  
## 1755 62.30999 54.70749 69.91248 50.68297 73.93700  
## 1756 62.30999 54.67896 69.94101 50.63935 73.98063  
## 1757 62.30999 54.65054 69.96943 50.59588 74.02410  
## 1758 62.30999 54.62223 69.99775 50.55257 74.06740  
## 1759 62.30999 54.59402 70.02596 50.50943 74.11055  
## 1760 62.30999 54.56591 70.05407 50.46644 74.15354  
## 1761 62.30999 54.53790 70.08208 50.42360 74.19637  
## 1762 62.30999 54.50999 70.10998 50.38092 74.23905

# The rmse is essentially the same as a random walk with no drift  
  
# Try a holt model  
s01\_v01\_holt <- holt(s01\_v01\_ts, h = 140)  
summary(s01\_v01\_holt)

##   
## Forecast method: Holt's method  
##   
## Model Information:  
## Holt's method   
##   
## Call:  
## holt(y = s01\_v01\_ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.9999   
## beta = 1e-04   
##   
## Initial states:  
## l = 27.5054   
## b = 0.0211   
##   
## sigma: 0.5148  
##   
## AIC AICc BIC   
## 9841.052 9841.089 9868.009   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.000832028 0.5141841 0.3504907 -0.01441626 0.9176757 1.001314  
## ACF1  
## Training set 0.08000527  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1623 62.33123 61.67146 62.99100 61.32220 63.34026  
## 1624 62.35247 61.41942 63.28552 60.92549 63.77945  
## 1625 62.37371 61.23092 63.51650 60.62596 64.12145  
## 1626 62.39495 61.07532 63.71458 60.37674 64.41316  
## 1627 62.41619 60.94073 63.89165 60.15966 64.67271  
## 1628 62.43743 60.82107 64.05379 59.96542 64.90944  
## 1629 62.45867 60.71271 64.20462 59.78846 65.12888  
## 1630 62.47991 60.61331 64.34650 59.62520 65.33462  
## 1631 62.50115 60.52123 64.48107 59.47312 65.52917  
## 1632 62.52239 60.43527 64.60951 59.33042 65.71436  
## 1633 62.54363 60.35453 64.73273 59.19569 65.89156  
## 1634 62.56487 60.27832 64.85142 59.06789 66.06184  
## 1635 62.58611 60.20607 64.96614 58.94616 66.22606  
## 1636 62.60735 60.13735 65.07735 58.82981 66.38489  
## 1637 62.62859 60.07177 65.18541 58.71827 66.53891  
## 1638 62.64983 60.00903 65.29063 58.61107 66.68859  
## 1639 62.67107 59.94886 65.39328 58.50780 66.83433  
## 1640 62.69231 59.89104 65.49358 58.40813 66.97648  
## 1641 62.71355 59.83537 65.59172 58.31176 67.11534  
## 1642 62.73479 59.78169 65.68788 58.21842 67.25115  
## 1643 62.75603 59.72986 65.78220 58.12790 67.38416  
## 1644 62.77727 59.67973 65.87481 58.03999 67.51454  
## 1645 62.79851 59.63119 65.96582 57.95452 67.64249  
## 1646 62.81975 59.58415 66.05534 57.87133 67.76816  
## 1647 62.84099 59.53851 66.14347 57.79028 67.89170  
## 1648 62.86223 59.49418 66.23028 57.71124 68.01322  
## 1649 62.88347 59.45109 66.31585 57.63409 68.13284  
## 1650 62.90471 59.40917 66.40025 57.55874 68.25067  
## 1651 62.92595 59.36836 66.48354 57.48508 68.36681  
## 1652 62.94719 59.32860 66.56577 57.41303 68.48134  
## 1653 62.96843 59.28984 66.64701 57.34251 68.59434  
## 1654 62.98967 59.25203 66.72730 57.27345 68.70588  
## 1655 63.01091 59.21513 66.80668 57.20577 68.81604  
## 1656 63.03215 59.17910 66.88520 57.13941 68.92488  
## 1657 63.05339 59.14389 66.96288 57.07433 69.03244  
## 1658 63.07463 59.10947 67.03978 57.01045 69.13880  
## 1659 63.09586 59.07582 67.11591 56.94773 69.24400  
## 1660 63.11710 59.04289 67.19132 56.88613 69.34807  
## 1661 63.13834 59.01067 67.26602 56.82561 69.45108  
## 1662 63.15958 58.97911 67.34006 56.76611 69.55306  
## 1663 63.18082 58.94821 67.41344 56.70760 69.65405  
## 1664 63.20206 58.91793 67.48620 56.65004 69.75408  
## 1665 63.22330 58.88825 67.55836 56.59341 69.85320  
## 1666 63.24454 58.85915 67.62993 56.53767 69.95142  
## 1667 63.26578 58.83062 67.70095 56.48278 70.04878  
## 1668 63.28702 58.80263 67.77142 56.42873 70.14532  
## 1669 63.30826 58.77516 67.84137 56.37548 70.24105  
## 1670 63.32950 58.74820 67.91081 56.32300 70.33600  
## 1671 63.35074 58.72173 67.97975 56.27128 70.43021  
## 1672 63.37198 58.69574 68.04822 56.22029 70.52368  
## 1673 63.39322 58.67022 68.11623 56.17001 70.61644  
## 1674 63.41446 58.64514 68.18379 56.12041 70.70852  
## 1675 63.43570 58.62050 68.25091 56.07148 70.79992  
## 1676 63.45694 58.59628 68.31760 56.02320 70.89068  
## 1677 63.47818 58.57248 68.38389 55.97555 70.98081  
## 1678 63.49942 58.54907 68.44977 55.92852 71.07033  
## 1679 63.52066 58.52606 68.51527 55.88208 71.15925  
## 1680 63.54190 58.50343 68.58038 55.83622 71.24759  
## 1681 63.56314 58.48116 68.64512 55.79093 71.33536  
## 1682 63.58438 58.45926 68.70950 55.74618 71.42258  
## 1683 63.60562 58.43771 68.77353 55.70198 71.50926  
## 1684 63.62686 58.41650 68.83722 55.65830 71.59542  
## 1685 63.64810 58.39563 68.90057 55.61514 71.68107  
## 1686 63.66934 58.37508 68.96360 55.57247 71.76621  
## 1687 63.69058 58.35485 69.02631 55.53029 71.85087  
## 1688 63.71182 58.33494 69.08870 55.48859 71.93505  
## 1689 63.73306 58.31533 69.15079 55.44735 72.01877  
## 1690 63.75430 58.29601 69.21259 55.40657 72.10203  
## 1691 63.77554 58.27699 69.27409 55.36624 72.18485  
## 1692 63.79678 58.25825 69.33531 55.32633 72.26723  
## 1693 63.81802 58.23979 69.39625 55.28686 72.34918  
## 1694 63.83926 58.22161 69.45691 55.24780 72.43072  
## 1695 63.86050 58.20369 69.51731 55.20915 72.51185  
## 1696 63.88174 58.18603 69.57745 55.17091 72.59257  
## 1697 63.90298 58.16863 69.63733 55.13305 72.67291  
## 1698 63.92422 58.15148 69.69696 55.09558 72.75286  
## 1699 63.94546 58.13458 69.75634 55.05848 72.83244  
## 1700 63.96670 58.11791 69.81549 55.02175 72.91165  
## 1701 63.98794 58.10149 69.87439 54.98539 72.99049  
## 1702 64.00918 58.08529 69.93307 54.94937 73.06898  
## 1703 64.03042 58.06932 69.99151 54.91371 73.14713  
## 1704 64.05166 58.05358 70.04974 54.87839 73.22493  
## 1705 64.07290 58.03806 70.10774 54.84341 73.30239  
## 1706 64.09414 58.02275 70.16553 54.80875 73.37953  
## 1707 64.11538 58.00765 70.22310 54.77442 73.45634  
## 1708 64.13662 57.99276 70.28047 54.74040 73.53283  
## 1709 64.15786 57.97808 70.33764 54.70670 73.60902  
## 1710 64.17910 57.96359 70.39460 54.67330 73.68489  
## 1711 64.20034 57.94931 70.45137 54.64021 73.76047  
## 1712 64.22158 57.93521 70.50794 54.60741 73.83574  
## 1713 64.24282 57.92131 70.56433 54.57490 73.91073  
## 1714 64.26406 57.90759 70.62052 54.54268 73.98543  
## 1715 64.28530 57.89406 70.67653 54.51075 74.05985  
## 1716 64.30654 57.88071 70.73236 54.47909 74.13399  
## 1717 64.32778 57.86754 70.78801 54.44770 74.20786  
## 1718 64.34902 57.85454 70.84349 54.41658 74.28146  
## 1719 64.37026 57.84172 70.89879 54.38572 74.35479  
## 1720 64.39150 57.82907 70.95393 54.35513 74.42787  
## 1721 64.41274 57.81658 71.00889 54.32479 74.50068  
## 1722 64.43398 57.80426 71.06369 54.29470 74.57325  
## 1723 64.45522 57.79210 71.11833 54.26486 74.64557  
## 1724 64.47646 57.78010 71.17281 54.23527 74.71764  
## 1725 64.49770 57.76826 71.22713 54.20592 74.78947  
## 1726 64.51894 57.75658 71.28129 54.17680 74.86107  
## 1727 64.54018 57.74505 71.33531 54.14792 74.93243  
## 1728 64.56142 57.73366 71.38917 54.11927 75.00356  
## 1729 64.58266 57.72243 71.44288 54.09085 75.07446  
## 1730 64.60390 57.71135 71.49644 54.06265 75.14514  
## 1731 64.62513 57.70040 71.54987 54.03467 75.21560  
## 1732 64.64637 57.68960 71.60314 54.00691 75.28584  
## 1733 64.66761 57.67895 71.65628 53.97937 75.35586  
## 1734 64.68885 57.66843 71.70928 53.95204 75.42567  
## 1735 64.71009 57.65804 71.76215 53.92491 75.49528  
## 1736 64.73133 57.64780 71.81487 53.89800 75.56467  
## 1737 64.75257 57.63768 71.86747 53.87128 75.63387  
## 1738 64.77381 57.62770 71.91993 53.84477 75.70286  
## 1739 64.79505 57.61784 71.97226 53.81846 75.77165  
## 1740 64.81629 57.60812 72.02447 53.79234 75.84025  
## 1741 64.83753 57.59852 72.07655 53.76641 75.90865  
## 1742 64.85877 57.58904 72.12850 53.74068 75.97687  
## 1743 64.88001 57.57969 72.18033 53.71513 76.04489  
## 1744 64.90125 57.57046 72.23204 53.68978 76.11273  
## 1745 64.92249 57.56135 72.28363 53.66460 76.18039  
## 1746 64.94373 57.55236 72.33510 53.63961 76.24786  
## 1747 64.96497 57.54349 72.38646 53.61479 76.31516  
## 1748 64.98621 57.53473 72.43770 53.59015 76.38227  
## 1749 65.00745 57.52609 72.48882 53.56569 76.44922  
## 1750 65.02869 57.51755 72.53983 53.54140 76.51599  
## 1751 65.04993 57.50914 72.59073 53.51728 76.58258  
## 1752 65.07117 57.50083 72.64152 53.49333 76.64902  
## 1753 65.09241 57.49263 72.69220 53.46955 76.71528  
## 1754 65.11365 57.48454 72.74277 53.44593 76.78138  
## 1755 65.13489 57.47655 72.79323 53.42247 76.84731  
## 1756 65.15613 57.46867 72.84359 53.39918 76.91309  
## 1757 65.17737 57.46090 72.89385 53.37604 76.97870  
## 1758 65.19861 57.45322 72.94400 53.35306 77.04416  
## 1759 65.21985 57.44565 72.99405 53.33024 77.10946  
## 1760 65.24109 57.43818 73.04400 53.30757 77.17461  
## 1761 65.26233 57.43081 73.09385 53.28506 77.23961  
## 1762 65.28357 57.42354 73.14360 53.26269 77.30445

# The rmse is still the same  
  
# Checking arima  
s01\_v01\_ts %>% diff() %>% ggtsdisplay(main="")



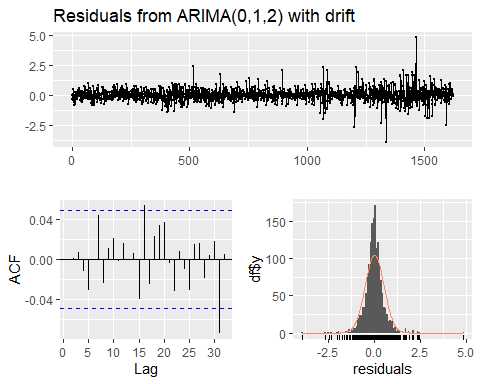
s01\_v01\_ts %>% diff() %>% ur.kpss() %>% summary()

##   
## #######################   
## # KPSS Unit Root Test #   
## #######################   
##   
## Test is of type: mu with 8 lags.   
##   
## Value of test-statistic is: 0.1001   
##   
## Critical value for a significance level of:   
## 10pct 5pct 2.5pct 1pct  
## critical values 0.347 0.463 0.574 0.739

s01\_v01\_arima <- auto.arima(s01\_v01\_ts, seasonal = FALSE,  
 stepwise = FALSE, approximation = FALSE)  
  
summary(s01\_v01\_arima)

## Series: s01\_v01\_ts   
## ARIMA(0,1,2) with drift   
##   
## Coefficients:  
## ma1 ma2 drift  
## 0.0875 -0.0731 0.0220  
## s.e. 0.0248 0.0250 0.0129  
##   
## sigma^2 estimated as 0.2612: log likelihood=-1210.39  
## AIC=2428.77 AICc=2428.79 BIC=2450.33  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 1.925779e-05 0.5103994 0.3471124 -0.01588171 0.9089467 0.9916625  
## ACF1  
## Training set -0.00036009

checkresiduals(s01\_v01\_arima)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,2) with drift  
## Q\* = 6.7638, df = 7, p-value = 0.4539  
##   
## Model df: 3. Total lags used: 10

The rmse is slightly better with the arima model.

# Check performance of random walk  
rwf\_nodrift <- tsCV(s01\_v02\_ts, rwf, drift = FALSE, h = 1)  
rmse\_rwf\_nodrift <- sqrt(mean(rwf\_nodrift^2, na.rm = TRUE))  
rwf\_drift <- tsCV(s01\_v02\_ts, rwf, drift = TRUE, h = 1)  
rmse\_rwf\_drift <- sqrt(mean(rwf\_drift^2, na.rm = TRUE))  
meanf <- tsCV(s01\_v02\_ts, meanf, h = 1)  
rmse\_meanf <- sqrt(mean(meanf^2, na.rm = TRUE))

The random walk with no drift has the lowest rmse

# Try ses  
s01\_v02\_ses <- ses(s01\_v02\_ts, h = 140)  
summary(s01\_v02\_ses)

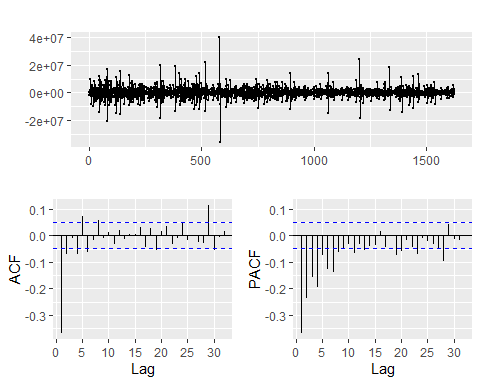
##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = s01\_v02\_ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.2836   
##   
## Initial states:  
## l = 14881687.8294   
##   
## sigma: 3429485  
##   
## AIC AICc BIC   
## 60808.33 60808.34 60824.50   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -19256.35 3427370 2260049 -10.20043 27.58586 0.8949302 0.1438804  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1623 6024892 1629829.903 10419953 -696775.2 12746558  
## 1624 6024892 1456545.197 10593238 -961791.3 13011574  
## 1625 6024892 1289597.481 10760186 -1217115.8 13266899  
## 1626 6024892 1128338.534 10921445 -1463740.1 13513523  
## 1627 6024892 972223.645 11077559 -1702497.2 13752280  
## 1628 6024892 820789.852 11228993 -1934095.2 13983878  
## 1629 6024892 673639.734 11376143 -2159141.9 14208925  
## 1630 6024892 530429.108 11519354 -2378163.6 14427947  
## 1631 6024892 390857.556 11658926 -2591619.9 14641403  
## 1632 6024892 254661.003 11795122 -2799914.5 14849698  
## 1633 6024892 121605.847 11928177 -3003404.8 15053188  
## 1634 6024892 -8515.747 12058299 -3202408.6 15252192  
## 1635 6024892 -135889.661 12185673 -3397210.2 15446993  
## 1636 6024892 -260682.938 12310466 -3588065.1 15637848  
## 1637 6024892 -383046.353 12432829 -3775203.8 15824987  
## 1638 6024892 -503116.545 12552900 -3958835.3 16008618  
## 1639 6024892 -621017.809 12670801 -4139149.7 16188933  
## 1640 6024892 -736863.602 12786647 -4316320.6 16366104  
## 1641 6024892 -850757.819 12900541 -4490506.7 16540290  
## 1642 6024892 -962795.891 13012579 -4661854.2 16711637  
## 1643 6024892 -1073065.712 13122849 -4830497.3 16880280  
## 1644 6024892 -1181648.453 13231432 -4996560.3 17046343  
## 1645 6024892 -1288619.258 13338402 -5160158.0 17209941  
## 1646 6024892 -1394047.852 13443831 -5321397.1 17371180  
## 1647 6024892 -1497999.076 13547782 -5480376.8 17530160  
## 1648 6024892 -1600533.349 13650316 -5637189.5 17686973  
## 1649 6024892 -1701707.082 13751490 -5791921.3 17841704  
## 1650 6024892 -1801573.037 13851356 -5944653.1 17994436  
## 1651 6024892 -1900180.655 13949964 -6095460.5 18145244  
## 1652 6024892 -1997576.336 14047359 -6244414.3 18294197  
## 1653 6024892 -2093803.696 14143587 -6391581.3 18441364  
## 1654 6024892 -2188903.798 14238687 -6537024.4 18586808  
## 1655 6024892 -2282915.353 14332698 -6680802.7 18730586  
## 1656 6024892 -2375874.907 14425658 -6822972.0 18872755  
## 1657 6024892 -2467817.004 14517600 -6963585.3 19013368  
## 1658 6024892 -2558774.339 14608557 -7102692.6 19152476  
## 1659 6024892 -2648777.894 14698561 -7240341.1 19290124  
## 1660 6024892 -2737857.056 14787640 -7376576.0 19426359  
## 1661 6024892 -2826039.738 14875823 -7511439.7 19561223  
## 1662 6024892 -2913352.471 14963136 -7644973.0 19694756  
## 1663 6024892 -2999820.506 15049604 -7777214.5 19826998  
## 1664 6024892 -3085467.895 15135251 -7908200.9 19957984  
## 1665 6024892 -3170317.569 15220101 -8037967.2 20087750  
## 1666 6024892 -3254391.411 15304175 -8166547.1 20216330  
## 1667 6024892 -3337710.320 15387493 -8293972.4 20343755  
## 1668 6024892 -3420294.277 15470077 -8420273.6 20470057  
## 1669 6024892 -3502162.392 15551945 -8545480.1 20595263  
## 1670 6024892 -3583332.966 15633116 -8669619.8 20719403  
## 1671 6024892 -3663823.528 15713607 -8792719.5 20842503  
## 1672 6024892 -3743650.889 15793434 -8914804.9 20964588  
## 1673 6024892 -3822831.177 15872614 -9035900.7 21085684  
## 1674 6024892 -3901379.876 15951163 -9156030.6 21205814  
## 1675 6024892 -3979311.864 16029095 -9275217.2 21325000  
## 1676 6024892 -4056641.442 16106425 -9393482.6 21443266  
## 1677 6024892 -4133382.368 16183165 -9510847.8 21560631  
## 1678 6024892 -4209547.883 16259331 -9627332.9 21677116  
## 1679 6024892 -4285150.740 16334934 -9742957.4 21792741  
## 1680 6024892 -4360203.228 16409986 -9857740.3 21907523  
## 1681 6024892 -4434717.194 16484500 -9971699.6 22021483  
## 1682 6024892 -4508704.065 16558487 -10084852.8 22134636  
## 1683 6024892 -4582174.873 16631958 -10197216.7 22247000  
## 1684 6024892 -4655140.268 16704923 -10308807.6 22358591  
## 1685 6024892 -4727610.538 16777394 -10419641.3 22469424  
## 1686 6024892 -4799595.628 16849379 -10529733.0 22579516  
## 1687 6024892 -4871105.155 16920888 -10639097.4 22688880  
## 1688 6024892 -4942148.420 16991932 -10747748.7 22797532  
## 1689 6024892 -5012734.428 17062518 -10855700.7 22905484  
## 1690 6024892 -5082871.895 17132655 -10962966.7 23012750  
## 1691 6024892 -5152569.265 17202352 -11069559.6 23119343  
## 1692 6024892 -5221834.720 17271618 -11175492.0 23225275  
## 1693 6024892 -5290676.192 17340459 -11280775.9 23330559  
## 1694 6024892 -5359101.371 17408884 -11385423.2 23435206  
## 1695 6024892 -5427117.721 17476901 -11489445.2 23539228  
## 1696 6024892 -5494732.484 17544516 -11592853.1 23642636  
## 1697 6024892 -5561952.688 17611736 -11695657.5 23745441  
## 1698 6024892 -5628785.162 17678568 -11797869.0 23847652  
## 1699 6024892 -5695236.540 17745020 -11899497.6 23949281  
## 1700 6024892 -5761313.266 17811096 -12000553.2 24050336  
## 1701 6024892 -5827021.607 17876805 -12101045.4 24150829  
## 1702 6024892 -5892367.657 17942151 -12200983.6 24250767  
## 1703 6024892 -5957357.342 18007140 -12300376.8 24350160  
## 1704 6024892 -6021996.431 18071780 -12399233.7 24449017  
## 1705 6024892 -6086290.538 18136074 -12497563.1 24547346  
## 1706 6024892 -6150245.126 18200028 -12595373.2 24645156  
## 1707 6024892 -6213865.519 18263649 -12692672.2 24742455  
## 1708 6024892 -6277156.902 18326940 -12789468.0 24839251  
## 1709 6024892 -6340124.327 18389907 -12885768.3 24935551  
## 1710 6024892 -6402772.718 18452556 -12981580.8 25031364  
## 1711 6024892 -6465106.876 18514890 -13076912.7 25126696  
## 1712 6024892 -6527131.482 18576915 -13171771.1 25221554  
## 1713 6024892 -6588851.103 18638634 -13266163.1 25315946  
## 1714 6024892 -6650270.194 18700053 -13360095.5 25409879  
## 1715 6024892 -6711393.103 18761176 -13453574.9 25503358  
## 1716 6024892 -6772224.074 18822007 -13546607.9 25596391  
## 1717 6024892 -6832767.249 18882550 -13639200.7 25688984  
## 1718 6024892 -6893026.677 18942810 -13731359.5 25781143  
## 1719 6024892 -6953006.309 19002789 -13823090.4 25872874  
## 1720 6024892 -7012710.007 19062493 -13914399.4 25964182  
## 1721 6024892 -7072141.545 19121925 -14005292.1 26055075  
## 1722 6024892 -7131304.610 19181088 -14095774.2 26145557  
## 1723 6024892 -7190202.809 19239986 -14185851.2 26235634  
## 1724 6024892 -7248839.668 19298623 -14275528.5 26325312  
## 1725 6024892 -7307218.634 19357002 -14364811.4 26414595  
## 1726 6024892 -7365343.081 19415126 -14453705.1 26503488  
## 1727 6024892 -7423216.310 19472999 -14542214.6 26591998  
## 1728 6024892 -7480841.548 19530625 -14630344.8 26680128  
## 1729 6024892 -7538221.959 19588005 -14718100.5 26767884  
## 1730 6024892 -7595360.634 19645144 -14805486.6 26855270  
## 1731 6024892 -7652260.606 19702044 -14892507.6 26942291  
## 1732 6024892 -7708924.839 19758708 -14979168.1 27028951  
## 1733 6024892 -7765356.241 19815139 -15065472.5 27115256  
## 1734 6024892 -7821557.657 19871341 -15151425.1 27201208  
## 1735 6024892 -7877531.878 19927315 -15237030.3 27286813  
## 1736 6024892 -7933281.636 19983065 -15322292.2 27372075  
## 1737 6024892 -7988809.610 20038593 -15407214.9 27456998  
## 1738 6024892 -8044118.427 20093902 -15491802.4 27541586  
## 1739 6024892 -8099210.660 20148994 -15576058.7 27625842  
## 1740 6024892 -8154088.835 20203872 -15659987.7 27709771  
## 1741 6024892 -8208755.428 20258539 -15743593.0 27793376  
## 1742 6024892 -8263212.866 20312996 -15826878.5 27876662  
## 1743 6024892 -8317463.533 20367247 -15909847.7 27959631  
## 1744 6024892 -8371509.766 20421293 -15992504.3 28042287  
## 1745 6024892 -8425353.859 20475137 -16074851.7 28124635  
## 1746 6024892 -8478998.063 20528781 -16156893.4 28206677  
## 1747 6024892 -8532444.587 20582228 -16238632.8 28288416  
## 1748 6024892 -8585695.602 20635479 -16320073.2 28369856  
## 1749 6024892 -8638753.238 20688536 -16401217.9 28451001  
## 1750 6024892 -8691619.585 20741403 -16482070.0 28531853  
## 1751 6024892 -8744296.698 20794080 -16562632.7 28612416  
## 1752 6024892 -8796786.594 20846570 -16642909.1 28692692  
## 1753 6024892 -8849091.257 20898874 -16722902.1 28772685  
## 1754 6024892 -8901212.632 20950996 -16802614.9 28852398  
## 1755 6024892 -8953152.634 21002936 -16882050.3 28931833  
## 1756 6024892 -9004913.143 21054696 -16961211.1 29010994  
## 1757 6024892 -9056496.006 21106279 -17040100.3 29089883  
## 1758 6024892 -9107903.041 21157686 -17118720.6 29168504  
## 1759 6024892 -9159136.034 21208919 -17197074.7 29246858  
## 1760 6024892 -9210196.739 21259980 -17275165.3 29324948  
## 1761 6024892 -9261086.884 21310870 -17352995.1 29402778  
## 1762 6024892 -9311808.167 21361591 -17430566.6 29480350

# The rmse is much better with ses  
  
# Try a holt model  
s01\_v02\_holt <- holt(s01\_v02\_ts, h = 140)  
summary(s01\_v02\_holt)

##   
## Forecast method: Holt's method  
##   
## Model Information:  
## Holt's method   
##   
## Call:  
## holt(y = s01\_v02\_ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.3645   
## beta = 0.0085   
##   
## Initial states:  
## l = 7092880.4164   
## b = 1533384.8743   
##   
## sigma: 3503140  
##   
## AIC AICc BIC   
## 60879.26 60879.30 60906.22   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -109131.3 3498818 2311365 -9.706392 27.69387 0.9152499 0.1051293  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1623 6278408 1788952.65 10767862 -587621.2 13144436  
## 1624 6302020 1510363.89 11093676 -1026185.6 13630225  
## 1625 6325632 1236972.98 11414292 -1456800.5 14108065  
## 1626 6349245 967646.77 11730843 -1881199.1 14579688  
## 1627 6372857 701496.80 12044217 -2300740.0 15046454  
## 1628 6396469 437813.58 12355125 -2716508.3 15509447  
## 1629 6420082 176021.76 12664142 -3129384.0 15969548  
## 1630 6443694 -84351.34 12971740 -3540089.9 16427478  
## 1631 6467307 -343698.49 13278312 -3949226.8 16883840  
## 1632 6490919 -602349.18 13584187 -4357298.6 17339137  
## 1633 6514531 -860582.19 13889645 -4764731.5 17793794  
## 1634 6538144 -1118635.15 14194923 -5171889.1 18248177  
## 1635 6561756 -1376711.92 14500224 -5579083.1 18702596  
## 1636 6585369 -1634988.44 14805726 -5986582.6 19157320  
## 1637 6608981 -1893617.28 15111579 -6394620.9 19612583  
## 1638 6632593 -2152731.38 15417918 -6803401.4 20068588  
## 1639 6656206 -2412447.05 15724859 -7213101.9 20525514  
## 1640 6679818 -2672866.39 16032503 -7623878.5 20983515  
## 1641 6703431 -2934079.32 16340941 -8035868.9 21442730  
## 1642 6727043 -3196165.27 16650251 -8449194.4 21903280  
## 1643 6750655 -3459194.56 16960505 -8863962.7 22365273  
## 1644 6774268 -3723229.60 17271765 -9280269.0 22828805  
## 1645 6797880 -3988325.86 17584086 -9698198.4 23293959  
## 1646 6821493 -4254532.73 17897518 -10117826.3 23760811  
## 1647 6845105 -4521894.26 18212104 -10539220.2 24229430  
## 1648 6868717 -4790449.75 18527884 -10962440.0 24699875  
## 1649 6892330 -5060234.33 18844894 -11387539.6 25172199  
## 1650 6915942 -5331279.38 19163164 -11814566.8 25646451  
## 1651 6939555 -5603612.97 19482722 -12243564.8 26122674  
## 1652 6963167 -5877260.20 19803594 -12674571.7 26600906  
## 1653 6986779 -6152243.49 20125802 -13107622.0 27081181  
## 1654 7010392 -6428582.91 20449366 -13542746.3 27563530  
## 1655 7034004 -6706296.34 20774305 -13979972.0 28047980  
## 1656 7057617 -6985399.75 21100633 -14419323.5 28534557  
## 1657 7081229 -7265907.36 21428365 -14860822.5 29023280  
## 1658 7104841 -7547831.80 21757514 -15304488.4 29514171  
## 1659 7128454 -7831184.27 22088092 -15750338.2 30007246  
## 1660 7152066 -8115974.68 22420107 -16198387.2 30502519  
## 1661 7175678 -8402211.74 22753569 -16648648.7 31000006  
## 1662 7199291 -8689903.10 23088485 -17101134.3 31499716  
## 1663 7222903 -8979055.45 23424862 -17555854.3 32001661  
## 1664 7246516 -9269674.54 23762706 -18012817.5 32505849  
## 1665 7270128 -9561765.35 24102021 -18472031.5 33012288  
## 1666 7293740 -9855332.09 24442813 -18933502.7 33520984  
## 1667 7317353 -10150378.32 24785084 -19397236.6 34031942  
## 1668 7340965 -10446906.94 25128837 -19863237.7 34545168  
## 1669 7364578 -10744920.30 25474076 -20331509.4 35060665  
## 1670 7388190 -11044420.23 25820800 -20802054.7 35578435  
## 1671 7411802 -11345408.08 26169013 -21274875.5 36098480  
## 1672 7435415 -11647884.76 26518714 -21749973.3 36620803  
## 1673 7459027 -11951850.76 26869905 -22227348.9 37145403  
## 1674 7482640 -12257306.23 27222585 -22707002.3 37672282  
## 1675 7506252 -12564250.96 27576755 -23188933.4 38201437  
## 1676 7529864 -12872684.42 27932413 -23673141.3 38732870  
## 1677 7553477 -13182605.81 28289559 -24159624.9 39266578  
## 1678 7577089 -13494014.06 28648192 -24648382.3 39802561  
## 1679 7600702 -13806907.85 29008311 -25139411.7 40340815  
## 1680 7624314 -14121285.66 29369914 -25632710.7 40881339  
## 1681 7647926 -14437145.72 29732998 -26128276.7 41424129  
## 1682 7671539 -14754486.12 30097564 -26626106.6 41969184  
## 1683 7695151 -15073304.76 30463607 -27126197.2 42516500  
## 1684 7718764 -15393599.35 30831126 -27628545.2 43066072  
## 1685 7742376 -15715367.50 31200119 -28133146.8 43617899  
## 1686 7765988 -16038606.67 31570583 -28639998.1 44171975  
## 1687 7789601 -16363314.18 31942516 -29149095.0 44728296  
## 1688 7813213 -16689487.25 32315913 -29660433.3 45286860  
## 1689 7836825 -17017123.01 32690774 -30174008.6 45847660  
## 1690 7860438 -17346218.47 33067094 -30689816.3 46410692  
## 1691 7884050 -17676770.58 33444871 -31207851.8 46975952  
## 1692 7907663 -18008776.19 33824102 -31728110.2 47543436  
## 1693 7931275 -18342232.08 34204782 -32250586.6 48113137  
## 1694 7954887 -18677134.99 34586910 -32775276.1 48685051  
## 1695 7978500 -19013481.57 34970481 -33302173.4 49259173  
## 1696 8002112 -19351268.43 35355493 -33831273.5 49835498  
## 1697 8025725 -19690492.14 35741941 -34362571.0 50414020  
## 1698 8049337 -20031149.21 36129823 -34896060.7 50994735  
## 1699 8072949 -20373236.12 36519135 -35431737.1 51577636  
## 1700 8096562 -20716749.33 36909873 -35969594.9 52162719  
## 1701 8120174 -21061685.23 37302034 -36509628.5 52749977  
## 1702 8143787 -21408040.23 37695613 -37051832.4 53339406  
## 1703 8167399 -21755810.68 38090609 -37596201.0 53930999  
## 1704 8191011 -22104992.93 38487016 -38142728.9 54524752  
## 1705 8214624 -22455583.30 38884831 -38691410.2 55120658  
## 1706 8238236 -22807578.12 39284050 -39242239.5 55718712  
## 1707 8261849 -23160973.66 39684671 -39795211.0 56318908  
## 1708 8285461 -23515766.23 40086688 -40350319.0 56921241  
## 1709 8309073 -23871952.11 40490099 -40907558.0 57525705  
## 1710 8332686 -24229527.57 40894899 -41466922.1 58132294  
## 1711 8356298 -24588488.88 41301085 -42028405.7 58741002  
## 1712 8379911 -24948832.32 41708653 -42592003.0 59351824  
## 1713 8403523 -25310554.15 42117600 -43157708.5 59964754  
## 1714 8427135 -25673650.64 42527921 -43725516.3 60579787  
## 1715 8450748 -26038118.07 42939614 -44295420.8 61196916  
## 1716 8474360 -26403952.72 43352673 -44867416.2 61816136  
## 1717 8497973 -26771150.86 43767096 -45441497.0 62437442  
## 1718 8521585 -27139708.78 44182879 -46017657.3 63060827  
## 1719 8545197 -27509622.79 44600017 -46595891.6 63686286  
## 1720 8568810 -27880889.19 45018509 -47176194.2 64313814  
## 1721 8592422 -28253504.28 45438348 -47758559.5 64943404  
## 1722 8616034 -28627464.40 45859533 -48342981.8 65575051  
## 1723 8639647 -29002765.87 46282060 -48929455.5 66208749  
## 1724 8663259 -29379405.05 46705924 -49517975.1 66844494  
## 1725 8686872 -29757378.29 47131122 -50108534.9 67482278  
## 1726 8710484 -30136681.96 47557650 -50701129.5 68122098  
## 1727 8734096 -30517312.46 47985505 -51295753.2 68763946  
## 1728 8757709 -30899266.16 48414684 -51892400.6 69407818  
## 1729 8781321 -31282539.50 48845182 -52491066.3 70053709  
## 1730 8804934 -31667128.90 49276996 -53091744.6 70701612  
## 1731 8828546 -32053030.80 49710123 -53694430.3 71351522  
## 1732 8852158 -32440241.66 50144559 -54299117.9 72003435  
## 1733 8875771 -32828757.96 50580300 -54905801.9 72657344  
## 1734 8899383 -33218576.18 51017343 -55514477.1 73313244  
## 1735 8922996 -33609692.84 51455684 -56125138.0 73971129  
## 1736 8946608 -34002104.46 51895320 -56737779.5 74630995  
## 1737 8970220 -34395807.58 52336248 -57352396.1 75292837  
## 1738 8993833 -34790798.77 52778464 -57968982.6 75956648  
## 1739 9017445 -35187074.60 53221965 -58587533.9 76622424  
## 1740 9041058 -35584631.66 53666747 -59208044.6 77290160  
## 1741 9064670 -35983466.57 54112806 -59830509.6 77959849  
## 1742 9088282 -36383575.96 54560141 -60454923.7 78631488  
## 1743 9111895 -36784956.47 55008746 -61081281.9 79305071  
## 1744 9135507 -37187604.78 55458619 -61709579.0 79980593  
## 1745 9159120 -37591517.56 55909757 -62339809.9 80658049  
## 1746 9182732 -37996691.53 56362155 -62971969.7 81337434  
## 1747 9206344 -38403123.40 56815812 -63606053.2 82018742  
## 1748 9229957 -38810809.92 57270723 -64242055.6 82701969  
## 1749 9253569 -39219747.83 57726886 -64879971.8 83387110  
## 1750 9277182 -39629933.93 58184297 -65519797.0 84074160  
## 1751 9300794 -40041365.00 58642953 -66161526.1 84763114  
## 1752 9324406 -40454037.86 59102850 -66805154.4 85453967  
## 1753 9348019 -40867949.34 59563987 -67450677.1 86146714  
## 1754 9371631 -41283096.29 60026358 -68098089.2 86841351  
## 1755 9395243 -41699475.58 60489963 -68747386.0 87537873  
## 1756 9418856 -42117084.10 60954796 -69398562.8 88236275  
## 1757 9442468 -42535918.76 61420855 -70051614.8 88936551  
## 1758 9466081 -42955976.47 61888138 -70706537.2 89638699  
## 1759 9489693 -43377254.19 62356640 -71363325.5 90342712  
## 1760 9513305 -43799748.87 62826360 -72021975.0 91048586  
## 1761 9536918 -44223457.48 63297293 -72682481.1 91756317  
## 1762 9560530 -44648377.03 63769437 -73344839.1 92465900

The rmse is still better with ses

# Checking arima  
s01\_v02\_ts %>% diff() %>% ggtsdisplay(main="")



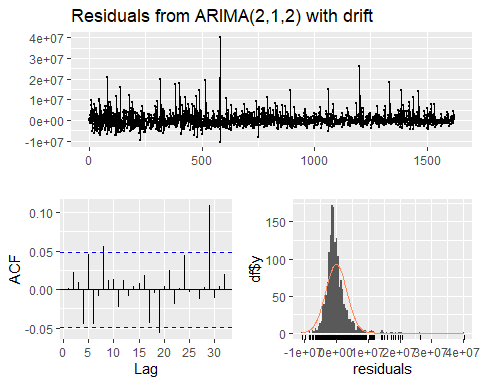
s01\_v02\_ts %>% diff() %>% ur.kpss() %>% summary()

##   
## #######################   
## # KPSS Unit Root Test #   
## #######################   
##   
## Test is of type: mu with 8 lags.   
##   
## Value of test-statistic is: 0.0042   
##   
## Critical value for a significance level of:   
## 10pct 5pct 2.5pct 1pct  
## critical values 0.347 0.463 0.574 0.739

s01\_v02\_arima <- auto.arima(s01\_v02\_ts, seasonal = FALSE,  
 stepwise = FALSE, approximation = FALSE)  
  
summary(s01\_v02\_arima)

## Series: s01\_v02\_ts   
## ARIMA(2,1,2) with drift   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 drift  
## 1.1522 -0.2277 -1.7683 0.7708 -6232.062  
## s.e. 0.0516 0.0369 0.0427 0.0422 2817.412  
##   
## sigma^2 estimated as 1.096e+13: log likelihood=-26634.43  
## AIC=53280.85 AICc=53280.91 BIC=53313.2  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 11012.69 3305058 2160962 -10.72258 26.67912 0.8556937 0.001150632

checkresiduals(s01\_v02\_arima)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 16.603, df = 5, p-value = 0.005318  
##   
## Model df: 5. Total lags used: 10

# This is the lowest rmse

# Check performance of random walk  
rwf\_nodrift <- tsCV(s02\_v02\_ts, rwf, drift = FALSE, h = 1)  
rmse\_rwf\_nodrift <- sqrt(mean(rwf\_nodrift^2, na.rm = TRUE))  
rwf\_drift <- tsCV(s02\_v02\_ts, rwf, drift = TRUE, h = 1)  
rmse\_rwf\_drift <- sqrt(mean(rwf\_drift^2, na.rm = TRUE))  
meanf <- tsCV(s02\_v02\_ts, meanf, h = 1)  
rmse\_meanf <- sqrt(mean(meanf^2, na.rm = TRUE))

The random walk with no drift has the lowest rmse

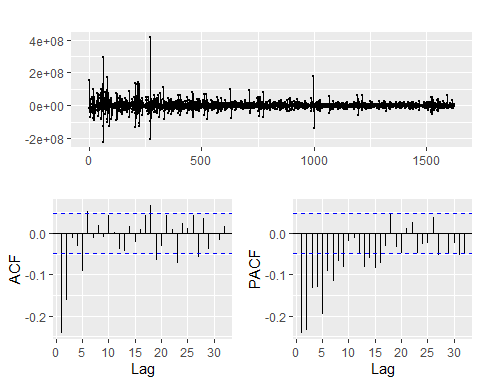
# Try ses  
s02\_v02\_ses <- ses(s02\_v02\_ts, h = 140)  
summary(s02\_v02\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = s02\_v02\_ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.4478   
##   
## Initial states:  
## l = 143804147.3047   
##   
## sigma: 26905675  
##   
## AIC AICc BIC   
## 67490.73 67490.75 67506.91   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -166302.6 26889082 14946398 -9.704775 28.34966 0.9549655 0.1500061  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1623 23006041 -11474968 57487051 -29728112 75740194  
## 1624 23006041 -14774650 60786733 -34774540 80786623  
## 1625 23006041 -17808433 63820515 -39414310 85426393  
## 1626 23006041 -20631809 66643891 -43732291 89744373  
## 1627 23006041 -23283294 69295377 -47787388 93799471  
## 1628 23006041 -25790918 71803000 -51622468 97634550  
## 1629 23006041 -28175829 74187912 -55269875 101281957  
## 1630 23006041 -30454454 76466536 -58754730 104766813  
## 1631 23006041 -32639850 78651932 -62097005 108109087  
## 1632 23006041 -34742602 80754684 -65312887 111324970  
## 1633 23006041 -36771433 82783516 -68415717 114427799  
## 1634 23006041 -38733631 84745713 -71416639 117428721  
## 1635 23006041 -40635358 86647441 -74325080 120337163  
## 1636 23006041 -42481884 88493967 -77149098 123161180  
## 1637 23006041 -44277753 90289836 -79895643 125907725  
## 1638 23006041 -46026920 92039002 -82570761 128582844  
## 1639 23006041 -47732847 93744929 -85179752 131191835  
## 1640 23006041 -49398592 95410675 -87727290 133739372  
## 1641 23006041 -51026868 97038950 -90217522 136229605  
## 1642 23006041 -52620093 98632176 -92654151 138666233  
## 1643 23006041 -54180440 100192522 -95040495 141052577  
## 1644 23006041 -55709863 101721945 -97379545 143391628  
## 1645 23006041 -57210130 103222213 -99674006 145686089  
## 1646 23006041 -58682849 104694932 -101926336 147938418  
## 1647 23006041 -60129483 106141566 -104138772 150150854  
## 1648 23006041 -61551372 107563454 -106313362 152325445  
## 1649 23006041 -62949742 108961825 -108451986 154464068  
## 1650 23006041 -64325725 110337807 -110556369 156568452  
## 1651 23006041 -65680362 111692444 -112628107 158640190  
## 1652 23006041 -67014616 113026698 -114668673 160680755  
## 1653 23006041 -68329381 114341463 -116679433 162691515  
## 1654 23006041 -69625487 115637569 -118661656 164673738  
## 1655 23006041 -70903706 116915788 -120616523 166628606  
## 1656 23006041 -72164759 118176842 -122545138 168557220  
## 1657 23006041 -73409320 119421402 -124448529 170460612  
## 1658 23006041 -74638019 120650102 -126327662 172339745  
## 1659 23006041 -75851448 121863530 -128183442 174195524  
## 1660 23006041 -77050162 123062245 -130016717 176028799  
## 1661 23006041 -78234684 124246767 -131828287 177840369  
## 1662 23006041 -79405507 125417589 -133618905 179630988  
## 1663 23006041 -80563094 126575176 -135389282 181401365  
## 1664 23006041 -81707885 127719968 -137140090 183152172  
## 1665 23006041 -82840296 128852378 -138871962 184884044  
## 1666 23006041 -83960719 129972801 -140585501 186597583  
## 1667 23006041 -85069526 131081609 -142281276 188293358  
## 1668 23006041 -86167074 132179156 -143959829 189971912  
## 1669 23006041 -87253696 133265778 -145621675 191633757  
## 1670 23006041 -88329713 134341796 -147267302 193279384  
## 1671 23006041 -89395431 135407513 -148897176 194909258  
## 1672 23006041 -90451138 136463221 -150511741 196523823  
## 1673 23006041 -91497112 137509195 -152111421 198123503  
## 1674 23006041 -92533618 138545700 -153696619 199708701  
## 1675 23006041 -93560907 139572990 -155267722 201279805  
## 1676 23006041 -94579222 140591304 -156825100 202837183  
## 1677 23006041 -95588793 141600876 -158369106 204381189  
## 1678 23006041 -96589843 142601925 -159900079 205912161  
## 1679 23006041 -97582582 143594664 -161418343 207430426  
## 1680 23006041 -98567215 144579298 -162924210 208936292  
## 1681 23006041 -99543938 145556020 -164417978 210430060  
## 1682 23006041 -100512937 146525019 -165899934 211912017  
## 1683 23006041 -101474394 147486476 -167370355 213382438  
## 1684 23006041 -102428481 148440563 -168829506 214841588  
## 1685 23006041 -103375365 149387448 -170277641 216289723  
## 1686 23006041 -104315208 150327291 -171715006 217727089  
## 1687 23006041 -105248164 151260247 -173141839 219153922  
## 1688 23006041 -106174382 152186465 -174558368 220570450  
## 1689 23006041 -107094007 153106089 -175964812 221976894  
## 1690 23006041 -108007176 154019259 -177361384 223373466  
## 1691 23006041 -108914025 154926107 -178748289 224760372  
## 1692 23006041 -109814682 155826764 -180125725 226137808  
## 1693 23006041 -110709272 156721355 -181493883 227505966  
## 1694 23006041 -111597918 157610000 -182852949 228865031  
## 1695 23006041 -112480734 158492817 -184203101 230215183  
## 1696 23006041 -113357836 159369918 -185544512 231556594  
## 1697 23006041 -114229332 160241414 -186877350 232889433  
## 1698 23006041 -115095328 161107411 -188201777 234213860  
## 1699 23006041 -115955928 161968011 -189517951 235530034  
## 1700 23006041 -116811231 162823313 -190826024 236838106  
## 1701 23006041 -117661333 163673416 -192126143 238138225  
## 1702 23006041 -118506329 164518411 -193418452 239430535  
## 1703 23006041 -119346309 165358391 -194703091 240715173  
## 1704 23006041 -120181361 166193443 -195980193 241992275  
## 1705 23006041 -121011572 167023654 -197249891 243261973  
## 1706 23006041 -121837024 167849106 -198512311 244524393  
## 1707 23006041 -122657798 168669880 -199767577 245779659  
## 1708 23006041 -123473974 169486056 -201015809 247027892  
## 1709 23006041 -124285626 170297709 -202257125 248269208  
## 1710 23006041 -125092831 171104914 -203491639 249503721  
## 1711 23006041 -125895660 171907742 -204719459 250731542  
## 1712 23006041 -126694184 172706266 -205940696 251952778  
## 1713 23006041 -127488470 173500552 -207155452 253167534  
## 1714 23006041 -128278587 174290669 -208363831 254375913  
## 1715 23006041 -129064598 175076680 -209565931 255578014  
## 1716 23006041 -129846567 175858649 -210761850 256773933  
## 1717 23006041 -130624556 176636639 -211951682 257963764  
## 1718 23006041 -131398626 177410708 -213135519 259147601  
## 1719 23006041 -132168834 178180916 -214313450 260325533  
## 1720 23006041 -132935238 178947320 -215485564 261497647  
## 1721 23006041 -133697893 179709976 -216651945 262664028  
## 1722 23006041 -134456855 180468938 -217812677 263824760  
## 1723 23006041 -135212176 181224259 -218967841 264979924  
## 1724 23006041 -135963909 181975991 -220117517 266129599  
## 1725 23006041 -136712103 182724186 -221261782 267273864  
## 1726 23006041 -137456809 183468891 -222400711 268412793  
## 1727 23006041 -138198075 184210157 -223534379 269546461  
## 1728 23006041 -138935947 184948030 -224662857 270674940  
## 1729 23006041 -139670473 185682555 -225786218 271798300  
## 1730 23006041 -140401697 186413779 -226904528 272916611  
## 1731 23006041 -141129663 187141746 -228017857 274029939  
## 1732 23006041 -141854415 187866498 -229126269 275138352  
## 1733 23006041 -142575995 188588078 -230229831 276241913  
## 1734 23006041 -143294444 189306527 -231328603 277340686  
## 1735 23006041 -144009803 190021885 -232422650 278434732  
## 1736 23006041 -144722110 190734192 -233512030 279524112  
## 1737 23006041 -145431405 191443488 -234596803 280608886  
## 1738 23006041 -146137726 192149809 -235677028 281689110  
## 1739 23006041 -146841110 192853192 -236752760 282764843  
## 1740 23006041 -147541592 193553675 -237824056 283836138  
## 1741 23006041 -148239210 194251292 -238890970 284903052  
## 1742 23006041 -148933997 194946079 -239953555 285965637  
## 1743 23006041 -149625987 195638070 -241011863 287023945  
## 1744 23006041 -150315215 196327297 -242065946 288078028  
## 1745 23006041 -151001713 197013795 -243115854 289127936  
## 1746 23006041 -151685513 197697596 -244161636 290173718  
## 1747 23006041 -152366647 198378730 -245203341 291215423  
## 1748 23006041 -153045146 199057228 -246241015 292253097  
## 1749 23006041 -153721040 199733122 -247274705 293286787  
## 1750 23006041 -154394358 200406441 -248304457 294316540  
## 1751 23006041 -155065131 201077213 -249330316 295342398  
## 1752 23006041 -155733387 201745469 -250352324 296364406  
## 1753 23006041 -156399153 202411235 -251370526 297382608  
## 1754 23006041 -157062458 203074540 -252384963 298397046  
## 1755 23006041 -157723328 203735411 -253395677 299407760  
## 1756 23006041 -158381791 204393873 -254402709 300414791  
## 1757 23006041 -159037872 205049954 -255406098 301418181  
## 1758 23006041 -159691597 205703679 -256405884 302417967  
## 1759 23006041 -160342991 206355073 -257402105 303414188  
## 1760 23006041 -160992079 207004161 -258394800 304406882  
## 1761 23006041 -161638885 207650968 -259384005 305396087  
## 1762 23006041 -162283434 208295516 -260369757 306381839

# The rmse is much better with ses  
  
# Try a holt model  
s02\_v02\_holt <- holt(s02\_v02\_ts, h = 140)  
summary(s02\_v02\_holt)

##   
## Forecast method: Holt's method  
##   
## Model Information:  
## Holt's method   
##   
## Call:  
## holt(y = s02\_v02\_ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.4611   
## beta = 0.0014   
##   
## Initial states:  
## l = 159309106.262   
## b = -2175070.1275   
##   
## sigma: 27008730  
##   
## AIC AICc BIC   
## 67505.13 67505.17 67532.09   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 944074.3 26975407 14782557 -7.560724 27.67563 0.9444972 0.1412501  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1623 22823428 -11789652 57436508 -30112710 75759566  
## 1624 22771521 -15364137 60907179 -35551935 81094977  
## 1625 22719614 -18658282 64097510 -40562417 86001645  
## 1626 22667707 -21733595 67069009 -45238224 90573638  
## 1627 22615800 -24632152 69863752 -49643707 94875307  
## 1628 22563893 -27384230 72512016 -53825169 98952955  
## 1629 22511986 -30012468 75036440 -57817234 102841206  
## 1630 22460079 -32534315 77454473 -61646589 106566747  
## 1631 22408172 -34963554 79779898 -65334312 110150656  
## 1632 22356265 -37311294 82023824 -68897392 113609922  
## 1633 22304358 -39586643 84195359 -72349760 116958476  
## 1634 22252451 -41797175 86302077 -75702999 120207901  
## 1635 22200544 -43949272 88350360 -78966869 123367957  
## 1636 22148637 -46048366 90345640 -82149678 126446952  
## 1637 22096730 -48099127 92292587 -85258569 129452028  
## 1638 22044823 -50105604 94195250 -88299733 132389379  
## 1639 21992916 -52071335 96057167 -91278581 135264413  
## 1640 21941009 -53999431 97881449 -94199871 138081889  
## 1641 21889102 -55892644 99670848 -97067813 140846017  
## 1642 21837195 -57753425 101427815 -99886154 143560543  
## 1643 21785288 -59583964 103154540 -102658245 146228820  
## 1644 21733381 -61386232 104852994 -105387097 148853859  
## 1645 21681474 -63162005 106524952 -108075429 151438377  
## 1646 21629567 -64912892 108172026 -110725703 153984836  
## 1647 21577660 -66640358 109795678 -113340156 156495476  
## 1648 21525753 -68345739 111397245 -115920833 158972339  
## 1649 21473846 -70030257 112977948 -118469602 161417294  
## 1650 21421939 -71695033 114538911 -120988180 163832058  
## 1651 21370032 -73341101 116081165 -123478146 166218210  
## 1652 21318125 -74969414 117605664 -125940958 168577208  
## 1653 21266218 -76580854 119113290 -128377965 170910401  
## 1654 21214311 -78176238 120604859 -130790416 173219038  
## 1655 21162404 -79756325 122081133 -133179473 175504281  
## 1656 21110497 -81321823 123542817 -135546218 177767212  
## 1657 21058590 -82873392 124990571 -137891659 180008839  
## 1658 21006683 -84411647 126425012 -140216739 182230105  
## 1659 20954776 -85937165 127846717 -142522341 184431892  
## 1660 20902869 -87450488 129256226 -144809291 186615028  
## 1661 20850962 -88952124 130654047 -147078366 188780289  
## 1662 20799055 -90442549 132040658 -149330297 190928406  
## 1663 20747148 -91922214 133416510 -151565771 193060067  
## 1664 20695241 -93391543 134782025 -153785439 195175920  
## 1665 20643334 -94850937 136137604 -155989911 197276578  
## 1666 20591427 -96300774 137483627 -158179767 199362620  
## 1667 20539520 -97741412 138820451 -160355555 201434594  
## 1668 20487613 -99173191 140148416 -162517794 203493020  
## 1669 20435706 -100596433 141467845 -164666977 205538389  
## 1670 20383799 -102011444 142779041 -166803572 207571169  
## 1671 20331892 -103418514 144082298 -168928022 209591805  
## 1672 20279985 -104817920 145377889 -171040751 211600720  
## 1673 20228078 -106209925 146666081 -173142161 213598316  
## 1674 20176171 -107594780 147947122 -175232636 215584977  
## 1675 20124264 -108972724 149221252 -177312542 217561069  
## 1676 20072357 -110343987 150488700 -179382228 219526942  
## 1677 20020450 -111708785 151749684 -181442029 221482928  
## 1678 19968543 -113067327 153004413 -183492263 223429348  
## 1679 19916636 -114419815 154253086 -185533236 225366507  
## 1680 19864729 -115766437 155495895 -187565240 227294697  
## 1681 19812822 -117107379 156733022 -189588555 229214198  
## 1682 19760915 -118442815 157964644 -191603450 231125279  
## 1683 19709008 -119772913 159190929 -193610182 233028197  
## 1684 19657101 -121097837 160412038 -195609000 234923201  
## 1685 19605194 -122417740 161628127 -197600140 236810527  
## 1686 19553287 -123732773 162839346 -199583831 238690404  
## 1687 19501380 -125043079 164045838 -201560293 240563052  
## 1688 19449473 -126348795 165247741 -203529736 242428682  
## 1689 19397566 -127650057 166445188 -205492366 244287497  
## 1690 19345659 -128946990 167638307 -207448377 246139694  
## 1691 19293752 -130239720 168827223 -209397958 247985461  
## 1692 19241845 -131528364 170012053 -211341291 249824981  
## 1693 19189938 -132813038 171192913 -213278553 251658428  
## 1694 19138031 -134093853 172369914 -215209912 253485974  
## 1695 19086124 -135370916 173543163 -217135533 255307780  
## 1696 19034217 -136644329 174712762 -219055573 257124007  
## 1697 18982310 -137914194 175878813 -220970186 258934805  
## 1698 18930403 -139180606 177041411 -222879518 260740324  
## 1699 18878496 -140443659 178200651 -224783714 262540705  
## 1700 18826589 -141703444 179356621 -226682910 264336087  
## 1701 18774682 -142960047 180509410 -228577241 266126604  
## 1702 18722775 -144213554 181659103 -230466837 267912386  
## 1703 18670868 -145464047 182805782 -232351822 269693557  
## 1704 18618961 -146711604 183949525 -234232319 271470240  
## 1705 18567054 -147956304 185090411 -236108445 273242552  
## 1706 18515147 -149198221 186228514 -237980314 275010607  
## 1707 18463240 -150437426 187363905 -239848038 276774517  
## 1708 18411332 -151673992 188496657 -241711724 278534389  
## 1709 18359425 -152907985 189626836 -243571475 280290326  
## 1710 18307518 -154139472 190754509 -245427395 282042432  
## 1711 18255611 -155368518 191879741 -247279580 283790803  
## 1712 18203704 -156595184 193002593 -249128127 285535536  
## 1713 18151797 -157819533 194123128 -250973128 287276723  
## 1714 18099890 -159041622 195241403 -252814674 289014455  
## 1715 18047983 -160261509 196357476 -254652852 290748819  
## 1716 17996076 -161479250 197471403 -256487748 292479901  
## 1717 17944169 -162694899 198583238 -258319446 294207785  
## 1718 17892262 -163908509 199693034 -260148024 295932549  
## 1719 17840355 -165120132 200800843 -261973564 297654274  
## 1720 17788448 -166329817 201906714 -263796140 299373037  
## 1721 17736541 -167537614 203010697 -265615827 301088910  
## 1722 17684634 -168743569 204112838 -267432699 302801968  
## 1723 17632727 -169947730 205213185 -269246826 304512280  
## 1724 17580820 -171150140 206311781 -271058276 306219917  
## 1725 17528913 -172350845 207408672 -272867117 307924944  
## 1726 17477006 -173549887 208503899 -274673415 309627428  
## 1727 17425099 -174747307 209597506 -276477233 311327432  
## 1728 17373192 -175943146 210689531 -278278634 313025019  
## 1729 17321285 -177137445 211780016 -280077679 314720250  
## 1730 17269378 -178330242 212868999 -281874426 316413183  
## 1731 17217471 -179521574 213956517 -283668934 318103877  
## 1732 17165564 -180711479 215042608 -285461259 319792388  
## 1733 17113657 -181899993 216127308 -287251456 321478771  
## 1734 17061750 -183087151 217210652 -289039579 323163080  
## 1735 17009843 -184272987 218292674 -290825681 324845368  
## 1736 16957936 -185457535 219373408 -292609813 326525686  
## 1737 16906029 -186640828 220452887 -294392026 328204084  
## 1738 16854122 -187822898 221531142 -296172367 329880612  
## 1739 16802215 -189003776 222608206 -297950886 331555317  
## 1740 16750308 -190183492 223684109 -299727629 333228246  
## 1741 16698401 -191362078 224758880 -301502642 334899445  
## 1742 16646494 -192539561 225832550 -303275970 336568958  
## 1743 16594587 -193715971 226905146 -305047656 338236830  
## 1744 16542680 -194891336 227976697 -306817743 339903104  
## 1745 16490773 -196065683 229047230 -308586274 341567821  
## 1746 16438866 -197239039 230116771 -310353289 343231022  
## 1747 16386959 -198411430 231185348 -312118829 344892747  
## 1748 16335052 -199582882 232252986 -313882932 346553036  
## 1749 16283145 -200753420 233319710 -315645637 348211928  
## 1750 16231238 -201923068 234385545 -317406982 349869459  
## 1751 16179331 -203091852 235450514 -319167004 351525667  
## 1752 16127424 -204259793 236514642 -320925739 353180588  
## 1753 16075517 -205426916 237577951 -322683222 354834256  
## 1754 16023610 -206593244 238640464 -324439488 356486708  
## 1755 15971703 -207758797 239702204 -326194570 358137976  
## 1756 15919796 -208923599 240763191 -327948502 359788095  
## 1757 15867889 -210087670 241823448 -329701317 361437096  
## 1758 15815982 -211251031 242882995 -331453046 363085011  
## 1759 15764075 -212413702 243941853 -333203722 364731872  
## 1760 15712168 -213575705 245000041 -334953373 366377710  
## 1761 15660261 -214737058 246057580 -336702031 368022554  
## 1762 15608354 -215897781 247114489 -338449726 369666434

# The rmse is still better with ses  
  
# Checking arima  
s02\_v02\_ts %>% diff() %>% ggtsdisplay(main="")



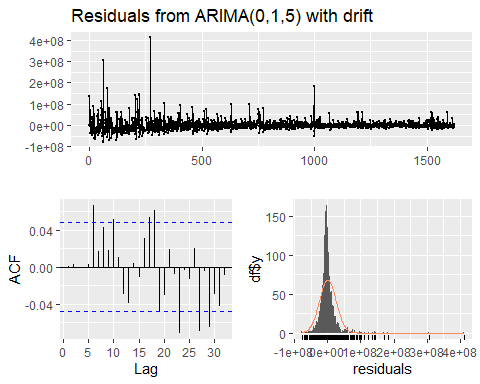
s02\_v02\_ts %>% diff() %>% ur.kpss() %>% summary()

##   
## #######################   
## # KPSS Unit Root Test #   
## #######################   
##   
## Test is of type: mu with 8 lags.   
##   
## Value of test-statistic is: 0.0042   
##   
## Critical value for a significance level of:   
## 10pct 5pct 2.5pct 1pct  
## critical values 0.347 0.463 0.574 0.739

s02\_v02\_arima <- auto.arima(s02\_v02\_ts, seasonal = FALSE,  
 stepwise = FALSE, approximation = FALSE)  
  
summary(s02\_v02\_arima)

## Series: s02\_v02\_ts   
## ARIMA(0,1,5) with drift   
##   
## Coefficients:  
## ma1 ma2 ma3 ma4 ma5 drift  
## -0.4560 -0.2656 -0.0655 -0.0645 -0.0943 -55595.76  
## s.e. 0.0251 0.0278 0.0271 0.0275 0.0253 34603.13  
##   
## sigma^2 estimated as 6.348e+14: log likelihood=-29923.36  
## AIC=59860.73 AICc=59860.8 BIC=59898.46  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -5031.419 25140281 14011965 -10.13021 27.2437 0.895262  
## ACF1  
## Training set 0.0003958196

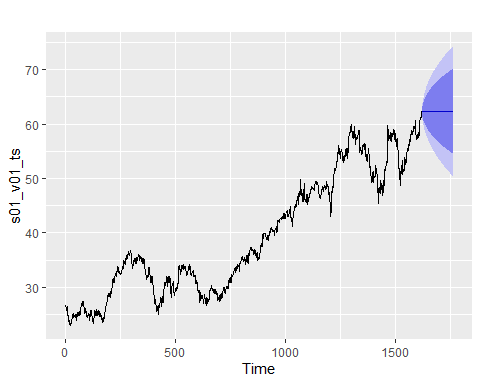
checkresiduals(s02\_v02\_arima)



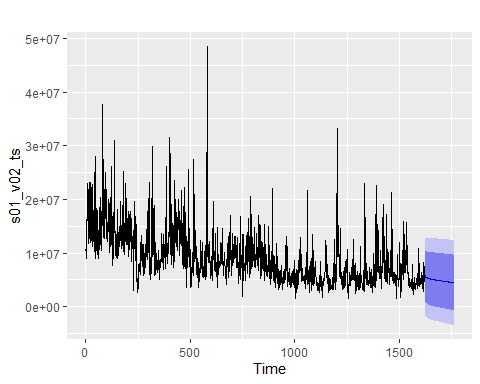
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,5) with drift  
## Q\* = 15.657, df = 4, p-value = 0.003516  
##   
## Model df: 6. Total lags used: 10

This is the lowest rmse

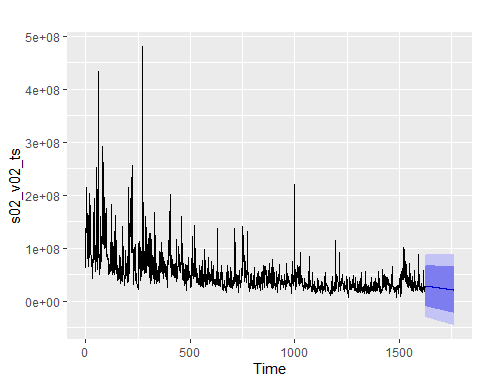
# Forecasts  
s01\_v01\_forecast <- rwf(s01\_v01\_ts, h = 140)  
s01\_v02\_forecast <- forecast(s01\_v02\_arima, h = 140)  
s02\_v02\_forecast <- forecast(s02\_v02\_arima, h = 140)  
  
autoplot(s01\_v01\_ts) +  
 autolayer(s01\_v01\_forecast)



autoplot(s01\_v02\_ts) +  
 autolayer(s01\_v02\_forecast)



autoplot(s02\_v02\_ts) +  
 autolayer(s02\_v02\_forecast)



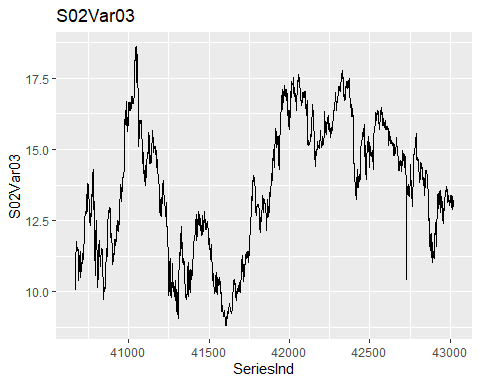
df = raw[c("SeriesInd", "S02Var03", "S03Var05", "S03Var07")]  
summary(df)

## SeriesInd S02Var03 S03Var05 S03Var07   
## Min. :40669 Min. : 8.82 Min. : 27.48 Min. : 27.44   
## 1st Qu.:41253 1st Qu.:11.82 1st Qu.: 53.30 1st Qu.: 53.46   
## Median :41846 Median :13.76 Median : 75.59 Median : 75.71   
## Mean :41843 Mean :13.68 Mean : 76.90 Mean : 76.87   
## 3rd Qu.:42430 3rd Qu.:15.52 3rd Qu.: 98.55 3rd Qu.: 98.61   
## Max. :43021 Max. :38.28 Max. :134.46 Max. :133.00   
## NA's :4 NA's :4 NA's :4

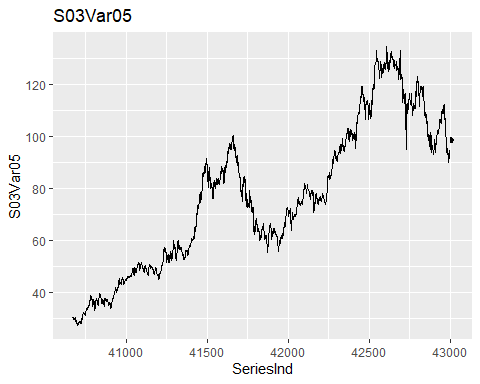
S03VAr05 and S03VAr07 looks almost identical from the summary above.

Visualize the data and look for seasonality or trend within the different data sets of data.

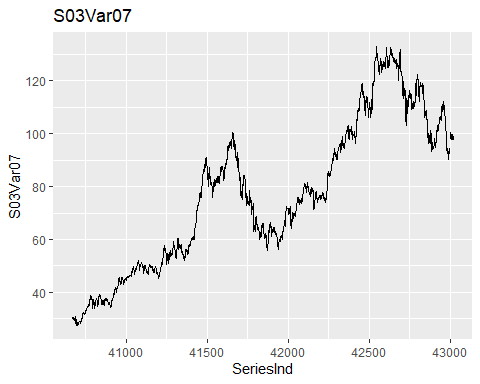
# Data under S02var03 has an outlier which we eliminate before creating visualizations.  
df %>%  
 filter(S02Var03 < 20) %>%  
 ggplot(aes(x = SeriesInd, y = S02Var03)) +  
 geom\_line() +  
 ggtitle("S02Var03")



df %>%  
 ggplot(aes(x = SeriesInd, y = S03Var05)) +  
 geom\_line() +  
 ggtitle("S03Var05")



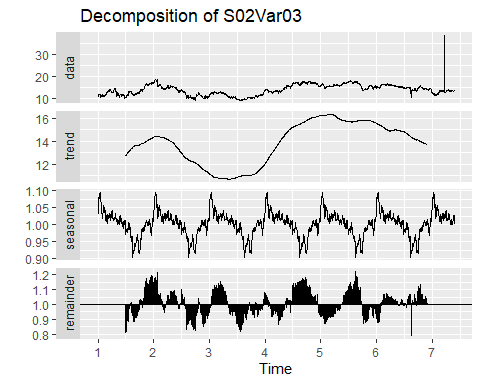
df %>%  
 ggplot(aes(x = SeriesInd, y = S03Var07)) +  
 geom\_line() +  
 ggtitle("S03Var07")



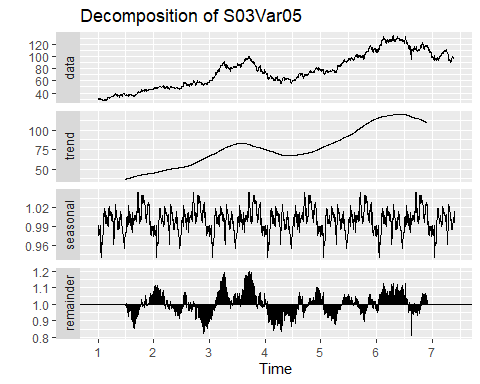
# Convert to time series  
SeriesInd.ts <- ts(raw$SeriesInd)  
S02Var03.ts <- na.remove(ts(raw$S02Var03, frequency = 253))  
S03Var05.ts <- na.remove(ts(raw$S03Var05, frequency = 253))  
S03Var07.ts <- na.remove(ts(raw$S03Var07, frequency = 253))

Decompose data and check for trend and seasonality

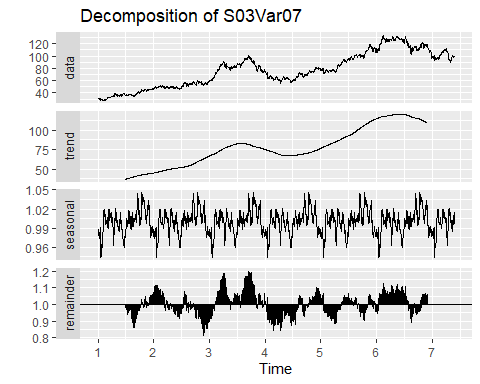
S02Var03.ts %>%  
 decompose(type="multiplicative")%>%  
 autoplot() + xlab("Time") +  
 ggtitle("Decomposition of S02Var03")



S03Var05.ts %>%  
 decompose(type="multiplicative") %>%  
 autoplot() + xlab("Time") +  
 ggtitle("Decomposition of S03Var05")



S03Var07.ts %>%  
 decompose(type="multiplicative") %>%  
 autoplot() + xlab("Time") +  
 ggtitle("Decomposition of S03Var07")



None of the three variables exhibits obvious seasonality or cyclicity. S02Var03 has an otlier that seems to compress the graph downwards.

Evaluating performance of simple models using cross-validation and RMSE:

S02Var03\_rmse\_rwf\_nodrift <- tsCV(S02Var03.ts, rwf, drift = FALSE, h = 1)  
S02Var03\_rmse\_rwf\_nodrift <- sqrt(mean(S02Var03\_rmse\_rwf\_nodrift^2, na.rm = TRUE))  
S02Var03\_rmse\_rwf\_drift <- tsCV(S02Var03.ts, rwf, drift = TRUE, h = 1)  
S02Var03\_rmse\_rwf\_drift <- sqrt(mean(S02Var03\_rmse\_rwf\_drift^2, na.rm = TRUE))  
S02Var03\_rmse\_meanf <- tsCV(S02Var03.ts, meanf, h = 1)  
S02Var03\_rmse\_meanf <- sqrt(mean(S02Var03\_rmse\_meanf^2, na.rm = TRUE))  
S03Var05\_rmse\_rwf\_nodrift <- tsCV(S03Var05.ts, rwf, drift = FALSE, h = 1)  
S03Var05\_rmse\_rwf\_nodrift <- sqrt(mean(S03Var05\_rmse\_rwf\_nodrift^2, na.rm = TRUE))  
S03Var05\_rmse\_rwf\_drift <- tsCV(S03Var05.ts, rwf, drift = TRUE, h = 1)  
S03Var05\_rmse\_rwf\_drift <- sqrt(mean(S03Var05\_rmse\_rwf\_drift^2, na.rm = TRUE))  
S03Var05\_rmse\_meanf <- tsCV(S03Var05.ts, meanf, h = 1)  
S03Var05\_rmse\_meanf <- sqrt(mean(S03Var05\_rmse\_meanf^2, na.rm = TRUE))  
S03Var07\_rmse\_rwf\_nodrift <- tsCV(S03Var07.ts, rwf, drift = FALSE, h = 1)  
S03Var07\_rmse\_rwf\_nodrift <- sqrt(mean(S03Var07\_rmse\_rwf\_nodrift^2, na.rm = TRUE))  
S03Var07\_rmse\_rwf\_drift <- tsCV(S03Var07.ts, rwf, drift = TRUE, h = 1)  
S03Var07\_rmse\_rwf\_drift <- sqrt(mean(S03Var07\_rmse\_rwf\_drift^2, na.rm = TRUE))  
S03Var07\_rmse\_meanf <- tsCV(S03Var07.ts, meanf, h = 1)  
S03Var07\_rmse\_meanf <- sqrt(mean(S03Var07\_rmse\_meanf^2, na.rm = TRUE))

S02Var03\_rmse\_rwf\_nodrift

## [1] 0.9358236

S02Var03\_rmse\_rwf\_drift

## [1] 0.9366271

S02Var03\_rmse\_meanf

## [1] NaN

S03Var05\_rmse\_rwf\_nodrift

## [1] 1.52121

S03Var05\_rmse\_rwf\_drift

## [1] 1.522178

S03Var05\_rmse\_meanf

## [1] NaN

S03Var07\_rmse\_rwf\_nodrift

## [1] 1.347304

S03Var07\_rmse\_rwf\_drift

## [1] 1.348208

S03Var07\_rmse\_meanf

## [1] NaN

From the RMSE analysis of three models, namely: rwf with no drift, rmf with draft and mean, the rfw is the slightly better performing model.

We will use time series cross-validation to compare the one-step forecast accuracy using Simple Exponential Smoothing, Holts Linear trend method and Holds dumped trend methods.

S02Var03

S02Var03\_ses <- tsCV(S02Var03.ts, ses, h=140)  
S02Var03\_holt <- tsCV(S02Var03.ts, holt, h=140)  
S02Var03\_holtdmpd <- tsCV(S02Var03.ts, holt, damped=TRUE, h=140)  
print("S02Var03")

## [1] "S02Var03"

print("MSE")

## [1] "MSE"

# Compare MSE:  
print(c("ses:", mean(S02Var03\_ses^2, na.rm=TRUE)))

## [1] "ses:" "4.62659362002589"

print(c("Holt:", mean(S02Var03\_holt^2, na.rm=TRUE)))

## [1] "Holt:" "12.5592562250959"

print(c("Holt damped:", mean(S02Var03\_holtdmpd^2, na.rm=TRUE)))

## [1] "Holt damped:" "6.983167294729"

print("MAE")

## [1] "MAE"

# Compare MAE:  
print(c("ses:", mean(abs(S02Var03\_ses), na.rm=TRUE)))

## [1] "ses:" "1.58534065211624"

print(c("Holt:", mean(abs(S02Var03\_holt), na.rm=TRUE)))

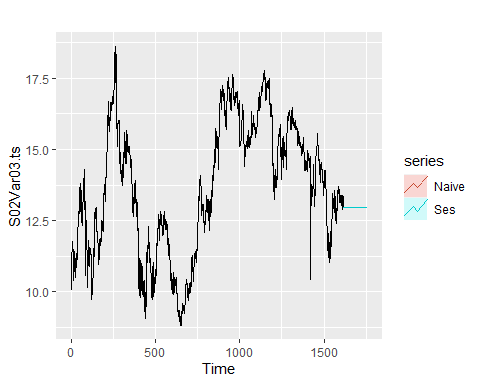
## [1] "Holt:" "2.15879529723051"

print(c("Holt damped:", mean(abs(S02Var03\_holtdmpd), na.rm=TRUE)))

## [1] "Holt damped:" "1.66965061485577"

Simple Exponential Smoothing method is best whether you compare MAE or MSE values. So we will proceed with using the SES method and apply it to the whole data set to get forecasts for future periods. We will also include the three

S02Var03.ts <- raw %>%  
 filter(S02Var03 < 20) %>%  
 select(S02Var03) %>%  
 ts()  
# Random walk forecasts without drift  
S02Var03\_rwf <- rwf(S02Var03.ts, h = 140)  
#S02Var03\_drwf <- rwf(S02Var03.ts, h = 140, drift = TRUE)  
#S02Var03\_mean <- meanf(S02Var03.ts, h = 140)  
# Exponential Smoothing forecast  
S02Var03\_ses <- ses(S02Var03.ts, h = 140)  
autoplot(S02Var03.ts) +  
 autolayer(S02Var03\_rwf, series = "Naive", PI = FALSE) +  
 #autolayer(S02Var03\_drwf, series = "Drift", PI = FALSE) +  
 #autolayer(S02Var03\_mean, series = "Mean", PI = FALSE) +  
 autolayer(S02Var03\_ses, series = "Ses", PI = FALSE)



Both simple exponential smoothing and naive random walk methods provide identical forecasts.

# Exponential Smoothing  
S02Var03\_ses <- ses(S02Var03.ts, h = 140)  
S02Var03\_ses[["model"]]

## Simple exponential smoothing   
##   
## Call:  
## ses(y = S02Var03.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.9999   
##   
## Initial states:  
## l = 10.0494   
##   
## sigma: 0.2683  
##   
## AIC AICc BIC   
## 7696.239 7696.254 7712.404

The value of alpha of is very close to one, showing that the level reacts strongly to each new observation. This also makes the SES method almost indistinguishable from random walk forecast.

# Estimate parameters  
summary(S02Var03\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = S02Var03.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.9999   
##   
## Initial states:  
## l = 10.0494   
##   
## sigma: 0.2683  
##   
## AIC AICc BIC   
## 7696.239 7696.254 7712.404   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.001800194 0.2681444 0.1790439 -0.006437297 1.370394 0.9993951  
## ACF1  
## Training set 0.03650273  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1618 12.96001 12.616156 13.30386 12.434130 13.48589  
## 1619 12.96001 12.473751 13.44627 12.216342 13.70368  
## 1620 12.96001 12.364477 13.55554 12.049222 13.87080  
## 1621 12.96001 12.272354 13.64766 11.908331 14.01169  
## 1622 12.96001 12.191191 13.72883 11.784204 14.13581  
## 1623 12.96001 12.117814 13.80220 11.671983 14.24804  
## 1624 12.96001 12.050337 13.86968 11.568785 14.35123  
## 1625 12.96001 11.987530 13.93249 11.472731 14.44729  
## 1626 12.96001 11.928541 13.99148 11.382515 14.53750  
## 1627 12.96001 11.872747 14.04727 11.297186 14.62283  
## 1628 12.96001 11.819680 14.10034 11.216027 14.70399  
## 1629 12.96001 11.768976 14.15104 11.138480 14.78154  
## 1630 12.96001 11.720343 14.19968 11.064103 14.85591  
## 1631 12.96001 11.673547 14.24647 10.992536 14.92748  
## 1632 12.96001 11.628395 14.29162 10.923482 14.99654  
## 1633 12.96001 11.584725 14.33529 10.856693 15.06332  
## 1634 12.96001 11.542399 14.37762 10.791962 15.12806  
## 1635 12.96001 11.501301 14.41872 10.729108 15.19091  
## 1636 12.96001 11.461329 14.45869 10.667976 15.25204  
## 1637 12.96001 11.422397 14.49762 10.608434 15.31158  
## 1638 12.96001 11.384425 14.53559 10.550362 15.36966  
## 1639 12.96001 11.347348 14.57267 10.493657 15.42636  
## 1640 12.96001 11.311104 14.60891 10.438227 15.48179  
## 1641 12.96001 11.275640 14.64438 10.383989 15.53603  
## 1642 12.96001 11.240908 14.67911 10.330870 15.58915  
## 1643 12.96001 11.206863 14.71316 10.278804 15.64121  
## 1644 12.96001 11.173467 14.74655 10.227729 15.69229  
## 1645 12.96001 11.140684 14.77933 10.177592 15.74243  
## 1646 12.96001 11.108481 14.81154 10.128342 15.79168  
## 1647 12.96001 11.076829 14.84319 10.079934 15.84008  
## 1648 12.96001 11.045700 14.87432 10.032327 15.88769  
## 1649 12.96001 11.015070 14.90495 9.985481 15.93454  
## 1650 12.96001 10.984914 14.93510 9.939362 15.98066  
## 1651 12.96001 10.955212 14.96481 9.893936 16.02608  
## 1652 12.96001 10.925943 14.99407 9.849174 16.07084  
## 1653 12.96001 10.897090 15.02293 9.805047 16.11497  
## 1654 12.96001 10.868635 15.05138 9.761528 16.15849  
## 1655 12.96001 10.840562 15.07946 9.718594 16.20142  
## 1656 12.96001 10.812855 15.10716 9.676221 16.24380  
## 1657 12.96001 10.785502 15.13452 9.634388 16.28563  
## 1658 12.96001 10.758489 15.16153 9.593074 16.32694  
## 1659 12.96001 10.731803 15.18822 9.552262 16.36776  
## 1660 12.96001 10.705433 15.21459 9.511932 16.40809  
## 1661 12.96001 10.679367 15.24065 9.472069 16.44795  
## 1662 12.96001 10.653597 15.26642 9.432656 16.48736  
## 1663 12.96001 10.628111 15.29191 9.393679 16.52634  
## 1664 12.96001 10.602901 15.31712 9.355123 16.56490  
## 1665 12.96001 10.577957 15.34206 9.316975 16.60304  
## 1666 12.96001 10.553272 15.36675 9.279222 16.64080  
## 1667 12.96001 10.528838 15.39118 9.241853 16.67817  
## 1668 12.96001 10.504646 15.41537 9.204856 16.71516  
## 1669 12.96001 10.480691 15.43933 9.168219 16.75180  
## 1670 12.96001 10.456965 15.46305 9.131934 16.78808  
## 1671 12.96001 10.433462 15.48656 9.095989 16.82403  
## 1672 12.96001 10.410175 15.50984 9.060375 16.85964  
## 1673 12.96001 10.387100 15.53292 9.025084 16.89493  
## 1674 12.96001 10.364229 15.55579 8.990106 16.92991  
## 1675 12.96001 10.341558 15.57846 8.955434 16.96458  
## 1676 12.96001 10.319082 15.60094 8.921059 16.99896  
## 1677 12.96001 10.296795 15.62322 8.886975 17.03304  
## 1678 12.96001 10.274693 15.64532 8.853173 17.06685  
## 1679 12.96001 10.252772 15.66725 8.819647 17.10037  
## 1680 12.96001 10.231027 15.68899 8.786391 17.13363  
## 1681 12.96001 10.209454 15.71056 8.753398 17.16662  
## 1682 12.96001 10.188048 15.73197 8.720661 17.19936  
## 1683 12.96001 10.166807 15.75321 8.688175 17.23184  
## 1684 12.96001 10.145726 15.77429 8.655935 17.26408  
## 1685 12.96001 10.124802 15.79522 8.623934 17.29608  
## 1686 12.96001 10.104031 15.81599 8.592167 17.32785  
## 1687 12.96001 10.083410 15.83661 8.560630 17.35939  
## 1688 12.96001 10.062936 15.85708 8.529318 17.39070  
## 1689 12.96001 10.042605 15.87741 8.498225 17.42179  
## 1690 12.96001 10.022415 15.89760 8.467347 17.45267  
## 1691 12.96001 10.002363 15.91765 8.436680 17.48334  
## 1692 12.96001 9.982446 15.93757 8.406220 17.51380  
## 1693 12.96001 9.962662 15.95736 8.375962 17.54406  
## 1694 12.96001 9.943007 15.97701 8.345902 17.57412  
## 1695 12.96001 9.923479 15.99654 8.316037 17.60398  
## 1696 12.96001 9.904076 16.01594 8.286363 17.63366  
## 1697 12.96001 9.884796 16.03522 8.256876 17.66314  
## 1698 12.96001 9.865635 16.05438 8.227573 17.69245  
## 1699 12.96001 9.846593 16.07343 8.198450 17.72157  
## 1700 12.96001 9.827666 16.09235 8.169504 17.75051  
## 1701 12.96001 9.808853 16.11116 8.140732 17.77929  
## 1702 12.96001 9.790152 16.12987 8.112131 17.80789  
## 1703 12.96001 9.771560 16.14846 8.083697 17.83632  
## 1704 12.96001 9.753076 16.16694 8.055429 17.86459  
## 1705 12.96001 9.734698 16.18532 8.027322 17.89270  
## 1706 12.96001 9.716425 16.20359 7.999375 17.92064  
## 1707 12.96001 9.698253 16.22176 7.971584 17.94843  
## 1708 12.96001 9.680182 16.23984 7.943947 17.97607  
## 1709 12.96001 9.662211 16.25781 7.916462 18.00356  
## 1710 12.96001 9.644336 16.27568 7.889125 18.03089  
## 1711 12.96001 9.626558 16.29346 7.861935 18.05808  
## 1712 12.96001 9.608874 16.31114 7.834890 18.08513  
## 1713 12.96001 9.591282 16.32874 7.807986 18.11203  
## 1714 12.96001 9.573782 16.34624 7.781222 18.13880  
## 1715 12.96001 9.556372 16.36365 7.754596 18.16542  
## 1716 12.96001 9.539051 16.38097 7.728105 18.19191  
## 1717 12.96001 9.521817 16.39820 7.701748 18.21827  
## 1718 12.96001 9.504669 16.41535 7.675522 18.24450  
## 1719 12.96001 9.487605 16.43241 7.649426 18.27059  
## 1720 12.96001 9.470625 16.44939 7.623457 18.29656  
## 1721 12.96001 9.453727 16.46629 7.597614 18.32240  
## 1722 12.96001 9.436911 16.48311 7.571895 18.34812  
## 1723 12.96001 9.420174 16.49984 7.546298 18.37372  
## 1724 12.96001 9.403516 16.51650 7.520822 18.39920  
## 1725 12.96001 9.386935 16.53308 7.495464 18.42455  
## 1726 12.96001 9.370431 16.54959 7.470224 18.44979  
## 1727 12.96001 9.354003 16.56602 7.445099 18.47492  
## 1728 12.96001 9.337649 16.58237 7.420088 18.49993  
## 1729 12.96001 9.321369 16.59865 7.395189 18.52483  
## 1730 12.96001 9.305161 16.61486 7.370401 18.54962  
## 1731 12.96001 9.289025 16.63099 7.345723 18.57430  
## 1732 12.96001 9.272959 16.64706 7.321153 18.59887  
## 1733 12.96001 9.256963 16.66305 7.296689 18.62333  
## 1734 12.96001 9.241036 16.67898 7.272331 18.64769  
## 1735 12.96001 9.225177 16.69484 7.248076 18.67194  
## 1736 12.96001 9.209385 16.71063 7.223924 18.69609  
## 1737 12.96001 9.193659 16.72636 7.199873 18.72014  
## 1738 12.96001 9.177998 16.74202 7.175923 18.74410  
## 1739 12.96001 9.162402 16.75762 7.152071 18.76795  
## 1740 12.96001 9.146870 16.77315 7.128316 18.79170  
## 1741 12.96001 9.131401 16.78862 7.104658 18.81536  
## 1742 12.96001 9.115994 16.80402 7.081095 18.83892  
## 1743 12.96001 9.100649 16.81937 7.057627 18.86239  
## 1744 12.96001 9.085364 16.83465 7.034251 18.88577  
## 1745 12.96001 9.070139 16.84988 7.010967 18.90905  
## 1746 12.96001 9.054974 16.86504 6.987774 18.93224  
## 1747 12.96001 9.039868 16.88015 6.964670 18.95535  
## 1748 12.96001 9.024819 16.89520 6.941655 18.97836  
## 1749 12.96001 9.009828 16.91019 6.918728 19.00129  
## 1750 12.96001 8.994893 16.92512 6.895888 19.02413  
## 1751 12.96001 8.980015 16.94000 6.873133 19.04688  
## 1752 12.96001 8.965192 16.95483 6.850463 19.06955  
## 1753 12.96001 8.950423 16.96959 6.827877 19.09214  
## 1754 12.96001 8.935709 16.98431 6.805374 19.11464  
## 1755 12.96001 8.921049 16.99897 6.782952 19.13707  
## 1756 12.96001 8.906441 17.01358 6.760612 19.15941  
## 1757 12.96001 8.891886 17.02813 6.738352 19.18167

S03Var05

S03Var05\_ses <- tsCV(S03Var05.ts, ses, h=140)  
S03Var05\_holt <- tsCV(S03Var05.ts, holt, h=140)  
S03Var05\_holtdmpd <- tsCV(S03Var05.ts, holt, damped=TRUE, h=140)  
print("S03Var05")

## [1] "S03Var05"

print("MSE")

## [1] "MSE"

# Compare MSE:  
print(c("ses:", mean(S03Var05\_ses^2, na.rm=TRUE)))

## [1] "ses:" "143.619383822223"

print(c("Holt:", mean(S03Var05\_holt^2, na.rm=TRUE)))

## [1] "Holt:" "168.725647965842"

print(c("Holt damped:", mean(S03Var05\_holtdmpd^2, na.rm=TRUE)))

## [1] "Holt damped:" "145.004023990954"

print("MAE")

## [1] "MAE"

# Compare MAE:  
print(c("ses:", mean(abs(S03Var05\_ses), na.rm=TRUE)))

## [1] "ses:" "9.10236584126034"

print(c("Holt:", mean(abs(S03Var05\_holt), na.rm=TRUE)))

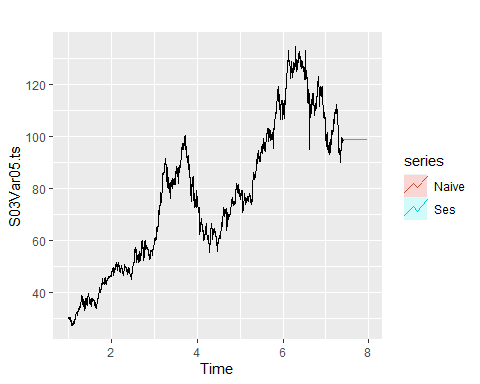
## [1] "Holt:" "9.43398449909038"

print(c("Holt damped:", mean(abs(S03Var05\_holtdmpd), na.rm=TRUE)))

## [1] "Holt damped:" "9.06816242407823"

Simple Exponential Smoothing method is best whether you compare MSE values. Holt damped is slightly better than SES for MAE value but the improvement is not high enough to justify the added complexity. So we will proceed with using the SES method and apply it to the whole data set to get forecasts for future periods.

# Random walk forecasts without drift  
S03Var05\_rwf <- rwf(S03Var05.ts, h = 140)  
#S03Var05\_drwf <- rwf(S03Var05.ts, h = 140, drift = TRUE)  
#S03Var05\_mean <- meanf(S03Var05.ts, h = 140)  
# Exponential Smoothing Forecast  
S03Var05\_ses <- ses(S03Var05.ts, h = 140)  
autoplot(S03Var05.ts) +  
 autolayer(S03Var05\_rwf, series = "Naive", PI = FALSE) +  
 #autolayer(S03Var05\_drwf, series = "Drift", PI = FALSE) +  
 #autolayer(S03Var05\_mean, series = "Mean", PI = FALSE) +  
 autolayer(S03Var05\_ses, series = "Ses", PI = FALSE)



Both simple exponential smoothing and naive random walk methods provide identical forcasts.

# Exponential Smoothing: Check for alpha  
  
S03Var05\_ses <- ses(S03Var05.ts, h = 140)  
S03Var05\_ses[["model"]]

## Simple exponential smoothing   
##   
## Call:  
## ses(y = S03Var05.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.8471   
##   
## Initial states:  
## l = 30.5127   
##   
## sigma: 1.5027  
##   
## AIC AICc BIC   
## 13277.28 13277.30 13293.45

The value of alpha of is very close to one, showing that the level reacts strongly to each new observation. This also makes the ses method almost indistinguishable fom random walk forecast.

# Estimate parameters  
summary(S03Var05\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = S03Var05.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.8471   
##   
## Initial states:  
## l = 30.5127   
##   
## sigma: 1.5027  
##   
## AIC AICc BIC   
## 13277.28 13277.30 13293.45   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.04972741 1.501802 1.007913 0.06594036 1.331037 0.04354704  
## ACF1  
## Training set -0.007611238  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 7.411077 98.66671 96.74088 100.5925 95.72141 101.6120  
## 7.415039 98.66671 96.14283 101.1906 94.80677 102.5266  
## 7.419002 98.66671 95.66153 101.6719 94.07068 103.2627  
## 7.422964 98.66671 95.24731 102.0861 93.43720 103.8962  
## 7.426926 98.66671 94.87812 102.4553 92.87256 104.4609  
## 7.430889 98.66671 94.54184 102.7916 92.35826 104.9752  
## 7.434851 98.66671 94.23098 103.1024 91.88284 105.4506  
## 7.438813 98.66671 93.94052 103.3929 91.43863 105.8948  
## 7.442776 98.66671 93.66691 103.6665 91.02017 106.3132  
## 7.446738 98.66671 93.40751 103.9259 90.62346 106.7100  
## 7.450700 98.66671 93.16032 104.1731 90.24542 107.0880  
## 7.454663 98.66671 92.92376 104.4097 89.88363 107.4498  
## 7.458625 98.66671 92.69657 104.6368 89.53616 107.7973  
## 7.462587 98.66671 92.47771 104.8557 89.20145 108.1320  
## 7.466550 98.66671 92.26633 105.0671 88.87817 108.4552  
## 7.470512 98.66671 92.06171 105.2717 88.56523 108.7682  
## 7.474475 98.66671 91.86324 105.4702 88.26170 109.0717  
## 7.478437 98.66671 91.67040 105.6630 87.96678 109.3666  
## 7.482399 98.66671 91.48273 105.8507 87.67977 109.6536  
## 7.486362 98.66671 91.29985 106.0336 87.40007 109.9333  
## 7.490324 98.66671 91.12139 106.2120 87.12715 110.2063  
## 7.494286 98.66671 90.94706 106.3864 86.86053 110.4729  
## 7.498249 98.66671 90.77658 106.5568 86.59980 110.7336  
## 7.502211 98.66671 90.60971 106.7237 86.34459 110.9888  
## 7.506173 98.66671 90.44622 106.8872 86.09456 111.2389  
## 7.510136 98.66671 90.28593 107.0475 85.84941 111.4840  
## 7.514098 98.66671 90.12864 107.2048 85.60885 111.7246  
## 7.518060 98.66671 89.97419 107.3592 85.37265 111.9608  
## 7.522023 98.66671 89.82245 107.5110 85.14058 112.1928  
## 7.525985 98.66671 89.67326 107.6602 84.91242 112.4210  
## 7.529947 98.66671 89.52651 107.8069 84.68798 112.6454  
## 7.533910 98.66671 89.38208 107.9513 84.46709 112.8663  
## 7.537872 98.66671 89.23986 108.0936 84.24958 113.0838  
## 7.541834 98.66671 89.09975 108.2337 84.03531 113.2981  
## 7.545797 98.66671 88.96167 108.3718 83.82413 113.5093  
## 7.549759 98.66671 88.82552 108.5079 83.61591 113.7175  
## 7.553721 98.66671 88.69123 108.6422 83.41053 113.9229  
## 7.557684 98.66671 88.55873 108.7747 83.20788 114.1255  
## 7.561646 98.66671 88.42794 108.9055 83.00786 114.3256  
## 7.565608 98.66671 88.29880 109.0346 82.81036 114.5231  
## 7.569571 98.66671 88.17125 109.1622 82.61528 114.7181  
## 7.573533 98.66671 88.04523 109.2882 82.42255 114.9109  
## 7.577496 98.66671 87.92068 109.4127 82.23208 115.1013  
## 7.581458 98.66671 87.79757 109.5358 82.04379 115.2896  
## 7.585420 98.66671 87.67583 109.6576 81.85762 115.4758  
## 7.589383 98.66671 87.55543 109.7780 81.67348 115.6599  
## 7.593345 98.66671 87.43632 109.8971 81.49131 115.8421  
## 7.597307 98.66671 87.31846 110.0150 81.31106 116.0224  
## 7.601270 98.66671 87.20181 110.1316 81.13266 116.2008  
## 7.605232 98.66671 87.08633 110.2471 80.95605 116.3774  
## 7.609194 98.66671 86.97200 110.3614 80.78119 116.5522  
## 7.613157 98.66671 86.85877 110.4746 80.60803 116.7254  
## 7.617119 98.66671 86.74662 110.5868 80.43651 116.8969  
## 7.621081 98.66671 86.63551 110.6979 80.26658 117.0668  
## 7.625044 98.66671 86.52542 110.8080 80.09821 117.2352  
## 7.629006 98.66671 86.41632 110.9171 79.93136 117.4021  
## 7.632968 98.66671 86.30818 111.0252 79.76598 117.5674  
## 7.636931 98.66671 86.20099 111.1324 79.60203 117.7314  
## 7.640893 98.66671 86.09470 111.2387 79.43948 117.8939  
## 7.644855 98.66671 85.98931 111.3441 79.27829 118.0551  
## 7.648818 98.66671 85.88478 111.4486 79.11844 118.2150  
## 7.652780 98.66671 85.78110 111.5523 78.95987 118.3735  
## 7.656742 98.66671 85.67825 111.6552 78.80258 118.5308  
## 7.660705 98.66671 85.57621 111.7572 78.64652 118.6869  
## 7.664667 98.66671 85.47496 111.8585 78.49167 118.8418  
## 7.668630 98.66671 85.37448 111.9589 78.33799 118.9954  
## 7.672592 98.66671 85.27475 112.0587 78.18547 119.1479  
## 7.676554 98.66671 85.17576 112.1577 78.03408 119.2993  
## 7.680517 98.66671 85.07749 112.2559 77.88379 119.4496  
## 7.684479 98.66671 84.97993 112.3535 77.73458 119.5988  
## 7.688441 98.66671 84.88305 112.4504 77.58643 119.7470  
## 7.692404 98.66671 84.78685 112.5466 77.43930 119.8941  
## 7.696366 98.66671 84.69132 112.6421 77.29320 120.0402  
## 7.700328 98.66671 84.59643 112.7370 77.14808 120.1853  
## 7.704291 98.66671 84.50218 112.8312 77.00394 120.3295  
## 7.708253 98.66671 84.40856 112.9249 76.86075 120.4727  
## 7.712215 98.66671 84.31554 113.0179 76.71849 120.6149  
## 7.716178 98.66671 84.22312 113.1103 76.57715 120.7563  
## 7.720140 98.66671 84.13129 113.2021 76.43671 120.8967  
## 7.724102 98.66671 84.04004 113.2934 76.29714 121.0363  
## 7.728065 98.66671 83.94935 113.3841 76.15845 121.1750  
## 7.732027 98.66671 83.85922 113.4742 76.02060 121.3128  
## 7.735989 98.66671 83.76963 113.5638 75.88359 121.4498  
## 7.739952 98.66671 83.68058 113.6528 75.74740 121.5860  
## 7.743914 98.66671 83.59205 113.7414 75.61201 121.7214  
## 7.747876 98.66671 83.50404 113.8294 75.47741 121.8560  
## 7.751839 98.66671 83.41654 113.9169 75.34359 121.9898  
## 7.755801 98.66671 83.32954 114.0039 75.21054 122.1229  
## 7.759763 98.66671 83.24303 114.0904 75.07823 122.2552  
## 7.763726 98.66671 83.15700 114.1764 74.94666 122.3868  
## 7.767688 98.66671 83.07145 114.2620 74.81582 122.5176  
## 7.771651 98.66671 82.98636 114.3471 74.68569 122.6477  
## 7.775613 98.66671 82.90173 114.4317 74.55626 122.7772  
## 7.779575 98.66671 82.81756 114.5159 74.42752 122.9059  
## 7.783538 98.66671 82.73383 114.5996 74.29947 123.0339  
## 7.787500 98.66671 82.65053 114.6829 74.17208 123.1613  
## 7.791462 98.66671 82.56767 114.7657 74.04535 123.2881  
## 7.795425 98.66671 82.48523 114.8482 73.91928 123.4141  
## 7.799387 98.66671 82.40321 114.9302 73.79384 123.5396  
## 7.803349 98.66671 82.32160 115.0118 73.66903 123.6644  
## 7.807312 98.66671 82.24040 115.0930 73.54484 123.7886  
## 7.811274 98.66671 82.15960 115.1738 73.42126 123.9122  
## 7.815236 98.66671 82.07919 115.2542 73.29828 124.0351  
## 7.819199 98.66671 81.99916 115.3343 73.17590 124.1575  
## 7.823161 98.66671 81.91952 115.4139 73.05410 124.2793  
## 7.827123 98.66671 81.84026 115.4932 72.93288 124.4005  
## 7.831086 98.66671 81.76137 115.5720 72.81222 124.5212  
## 7.835048 98.66671 81.68284 115.6506 72.69213 124.6413  
## 7.839010 98.66671 81.60468 115.7287 72.57259 124.7608  
## 7.842973 98.66671 81.52687 115.8065 72.45359 124.8798  
## 7.846935 98.66671 81.44942 115.8840 72.33513 124.9983  
## 7.850897 98.66671 81.37231 115.9611 72.21721 125.1162  
## 7.854860 98.66671 81.29554 116.0379 72.09980 125.2336  
## 7.858822 98.66671 81.21911 116.1143 71.98292 125.3505  
## 7.862784 98.66671 81.14302 116.1904 71.86654 125.4669  
## 7.866747 98.66671 81.06725 116.2662 71.75066 125.5828  
## 7.870709 98.66671 80.99181 116.3416 71.63529 125.6981  
## 7.874672 98.66671 80.91669 116.4167 71.52040 125.8130  
## 7.878634 98.66671 80.84189 116.4915 71.40599 125.9274  
## 7.882596 98.66671 80.76739 116.5660 71.29207 126.0413  
## 7.886559 98.66671 80.69321 116.6402 71.17862 126.1548  
## 7.890521 98.66671 80.61933 116.7141 71.06563 126.2678  
## 7.894483 98.66671 80.54576 116.7877 70.95310 126.3803  
## 7.898446 98.66671 80.47248 116.8609 70.84103 126.4924  
## 7.902408 98.66671 80.39949 116.9339 70.72941 126.6040  
## 7.906370 98.66671 80.32680 117.0066 70.61823 126.7152  
## 7.910333 98.66671 80.25439 117.0790 70.50750 126.8259  
## 7.914295 98.66671 80.18227 117.1512 70.39719 126.9362  
## 7.918257 98.66671 80.11042 117.2230 70.28732 127.0461  
## 7.922220 98.66671 80.03885 117.2946 70.17786 127.1556  
## 7.926182 98.66671 79.96756 117.3659 70.06883 127.2646  
## 7.930144 98.66671 79.89654 117.4369 69.96021 127.3732  
## 7.934107 98.66671 79.82579 117.5076 69.85200 127.4814  
## 7.938069 98.66671 79.75530 117.5781 69.74420 127.5892  
## 7.942031 98.66671 79.68507 117.6483 69.63679 127.6966  
## 7.945994 98.66671 79.61510 117.7183 69.52979 127.8036  
## 7.949956 98.66671 79.54539 117.7880 69.42317 127.9102  
## 7.953918 98.66671 79.47593 117.8575 69.31694 128.0165  
## 7.957881 98.66671 79.40672 117.9267 69.21109 128.1223  
## 7.961843 98.66671 79.33776 117.9957 69.10563 128.2278

S03Var07

S03Var07\_ses <- tsCV(S03Var07.ts, ses, h=140)  
S03Var07\_holt <- tsCV(S03Var07.ts, holt, h=140)  
S03Var07\_holtdmpd <- tsCV(S03Var07.ts, holt, damped=TRUE, h=140)  
print("S03Var07")

## [1] "S03Var07"

print("MSE")

## [1] "MSE"

# Compare MSE:  
print(c("ses:", mean(S03Var07\_ses^2, na.rm=TRUE)))

## [1] "ses:" "142.718636070367"

print(c("Holt:", mean(S03Var07\_holt^2, na.rm=TRUE)))

## [1] "Holt:" "165.172476563924"

print(c("Holt damped:", mean(S03Var07\_holtdmpd^2, na.rm=TRUE)))

## [1] "Holt damped:" "143.640660491007"

print("MAE")

## [1] "MAE"

# Compare MAE:  
print(c("ses:", mean(abs(S03Var07\_ses), na.rm=TRUE)))

## [1] "ses:" "9.07887197232571"

print(c("Holt:", mean(abs(S03Var07\_holt), na.rm=TRUE)))

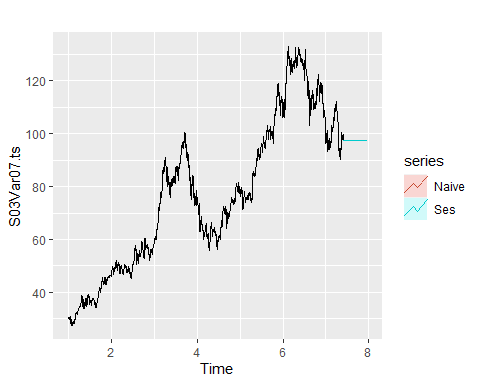
## [1] "Holt:" "9.28328222745204"

print(c("Holt damped:", mean(abs(S03Var07\_holtdmpd), na.rm=TRUE)))

## [1] "Holt damped:" "9.05125492620748"

Simple Exponential Smoothing method is best whether you compare MSE values. Holt damped is slightly better than ses for MAE value but the improvement is not high enough to justify the added complexity. So we will proceed with using the SES method and apply it to the whole data set to get forecasts for future periods.

# Random walk forecasts without drift  
S03Var07\_rwf <- rwf(S03Var07.ts, h = 140)  
#S03Var07\_drwf <- rwf(S03Var07.ts, h = 140, drift = TRUE)  
#S03Var07\_mean <- meanf(S03Var07.ts, h = 140)  
# Exponential Smoothing Forecast  
S03Var07\_ses <- ses(S03Var07.ts, h = 140)  
autoplot(S03Var07.ts) +  
 autolayer(S03Var07\_rwf, series = "Naive", PI = FALSE) +  
 #autolayer(S03Var07\_drwf, series = "Drift", PI = FALSE) +  
 #autolayer(S03Var07\_mean, series = "Mean", PI = FALSE) +  
 autolayer(S03Var07\_ses, series = "Ses", PI = FALSE)



Both simple exponential smoothing and naive random walk methods provide identical forcasts with ses having a slight edge.

# Exponential Smoothing: Check for alpha  
S03Var07\_ses <- ses(S03Var07.ts, h = 140)  
S03Var07\_ses[["model"]]

## Simple exponential smoothing   
##   
## Call:  
## ses(y = S03Var07.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.9999   
##   
## Initial states:  
## l = 30.5857   
##   
## sigma: 1.3477  
##   
## AIC AICc BIC   
## 12924.99 12925.00 12941.16

The value of alpha of is very close to one, showing that the level reacts strongly to each new observation. This also makes the ses method almost indistinguishable fom random walk forecast.

# Estimate parameters  
summary(S03Var07\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = S03Var07.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.9999   
##   
## Initial states:  
## l = 30.5857   
##   
## sigma: 1.3477  
##   
## AIC AICc BIC   
## 12924.99 12925.00 12941.16   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.04126163 1.34689 0.9318533 0.05725115 1.228263 0.04037496  
## ACF1  
## Training set 0.0103716  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 7.411077 97.34015 95.61297 99.06733 94.69866 99.98164  
## 7.415039 97.34015 94.89768 99.78262 93.60471 101.07559  
## 7.419002 97.34015 94.34880 100.33150 92.76527 101.91503  
## 7.422964 97.34015 93.88606 100.79424 92.05758 102.62272  
## 7.426926 97.34015 93.47838 101.20192 91.43409 103.24621  
## 7.430889 97.34015 93.10981 101.57049 90.87040 103.80989  
## 7.434851 97.34015 92.77087 101.90943 90.35204 104.32825  
## 7.438813 97.34015 92.45540 102.22490 89.86956 104.81073  
## 7.442776 97.34015 92.15909 102.52120 89.41641 105.26389  
## 7.446738 97.34015 91.87884 102.80146 88.98780 105.69250  
## 7.450700 97.34015 91.61229 103.06801 88.58014 106.10016  
## 7.454663 97.34015 91.35760 103.32270 88.19063 106.48967  
## 7.458625 97.34015 91.11332 103.56698 87.81703 106.86327  
## 7.462587 97.34015 90.87826 103.80203 87.45755 107.22275  
## 7.466550 97.34015 90.65147 104.02883 87.11069 107.56961  
## 7.470512 97.34015 90.43211 104.24819 86.77521 107.90508  
## 7.474475 97.34015 90.21951 104.46079 86.45007 108.23023  
## 7.478437 97.34015 90.01307 104.66723 86.13435 108.54595  
## 7.482399 97.34015 89.81229 104.86800 85.82729 108.85301  
## 7.486362 97.34015 89.61674 105.06356 85.52821 109.15209  
## 7.490324 97.34015 89.42601 105.25429 85.23651 109.44378  
## 7.494286 97.34015 89.23977 105.44053 84.95169 109.72861  
## 7.498249 97.34015 89.05772 105.62258 84.67326 110.00704  
## 7.502211 97.34015 88.87958 105.80072 84.40083 110.27947  
## 7.506173 97.34015 88.70512 105.97518 84.13401 110.54629  
## 7.510136 97.34015 88.53411 106.14618 83.87248 110.80782  
## 7.514098 97.34015 88.36637 106.31393 83.61593 111.06437  
## 7.518060 97.34015 88.20170 106.47860 83.36409 111.31621  
## 7.522023 97.34015 88.03994 106.64036 83.11671 111.56359  
## 7.525985 97.34015 87.88095 106.79934 82.87356 111.80674  
## 7.529947 97.34015 87.72459 106.95570 82.63443 112.04587  
## 7.533910 97.34015 87.57074 107.10956 82.39912 112.28118  
## 7.537872 97.34015 87.41926 107.26103 82.16747 112.51283  
## 7.541834 97.34015 87.27007 107.41023 81.93929 112.74100  
## 7.545797 97.34015 87.12306 107.55724 81.71445 112.96584  
## 7.549759 97.34015 86.97813 107.70217 81.49280 113.18750  
## 7.553721 97.34015 86.83520 107.84510 81.27421 113.40609  
## 7.557684 97.34015 86.69418 107.98611 81.05855 113.62175  
## 7.561646 97.34015 86.55502 108.12528 80.84571 113.83459  
## 7.565608 97.34015 86.41762 108.26268 80.63558 114.04471  
## 7.569571 97.34015 86.28193 108.39837 80.42807 114.25223  
## 7.573533 97.34015 86.14789 108.53241 80.22307 114.45723  
## 7.577496 97.34015 86.01543 108.66487 80.02049 114.65981  
## 7.581458 97.34015 85.88451 108.79579 79.82026 114.86004  
## 7.585420 97.34015 85.75506 108.92524 79.62229 115.05801  
## 7.589383 97.34015 85.62705 109.05325 79.42650 115.25379  
## 7.593345 97.34015 85.50041 109.17988 79.23284 115.44746  
## 7.597307 97.34015 85.37512 109.30517 79.04122 115.63908  
## 7.601270 97.34015 85.25113 109.42917 78.85159 115.82871  
## 7.605232 97.34015 85.12840 109.55190 78.66389 116.01641  
## 7.609194 97.34015 85.00688 109.67341 78.47805 116.20225  
## 7.613157 97.34015 84.88656 109.79374 78.29403 116.38627  
## 7.617119 97.34015 84.76738 109.91292 78.11176 116.56854  
## 7.621081 97.34015 84.64933 110.03097 77.93121 116.74909  
## 7.625044 97.34015 84.53236 110.14794 77.75232 116.92797  
## 7.629006 97.34015 84.41645 110.26385 77.57506 117.10524  
## 7.632968 97.34015 84.30157 110.37873 77.39936 117.28094  
## 7.636931 97.34015 84.18769 110.49260 77.22520 117.45509  
## 7.640893 97.34015 84.07480 110.60550 77.05254 117.62776  
## 7.644855 97.34015 83.96285 110.71745 76.88134 117.79896  
## 7.648818 97.34015 83.85183 110.82846 76.71155 117.96875  
## 7.652780 97.34015 83.74172 110.93857 76.54315 118.13715  
## 7.656742 97.34015 83.63250 111.04780 76.37611 118.30419  
## 7.660705 97.34015 83.52414 111.15616 76.21038 118.46992  
## 7.664667 97.34015 83.41662 111.26368 76.04594 118.63435  
## 7.668630 97.34015 83.30992 111.37038 75.88277 118.79753  
## 7.672592 97.34015 83.20403 111.47627 75.72082 118.95947  
## 7.676554 97.34015 83.09893 111.58137 75.56008 119.12021  
## 7.680517 97.34015 82.99460 111.68570 75.40052 119.27978  
## 7.684479 97.34015 82.89102 111.78928 75.24211 119.43819  
## 7.688441 97.34015 82.78818 111.89212 75.08483 119.59547  
## 7.692404 97.34015 82.68606 111.99424 74.92865 119.75165  
## 7.696366 97.34015 82.58464 112.09565 74.77355 119.90675  
## 7.700328 97.34015 82.48392 112.19638 74.61951 120.06079  
## 7.704291 97.34015 82.38388 112.29642 74.46651 120.21379  
## 7.708253 97.34015 82.28450 112.39580 74.31452 120.36578  
## 7.712215 97.34015 82.18577 112.49452 74.16353 120.51676  
## 7.716178 97.34015 82.08769 112.59261 74.01352 120.66678  
## 7.720140 97.34015 81.99023 112.69007 73.86447 120.81583  
## 7.724102 97.34015 81.89338 112.78692 73.71636 120.96394  
## 7.728065 97.34015 81.79714 112.88316 73.56917 121.11113  
## 7.732027 97.34015 81.70149 112.97881 73.42288 121.25742  
## 7.735989 97.34015 81.60642 113.07388 73.27749 121.40281  
## 7.739952 97.34015 81.51192 113.16837 73.13297 121.54733  
## 7.743914 97.34015 81.41799 113.26231 72.98930 121.69100  
## 7.747876 97.34015 81.32460 113.35570 72.84648 121.83382  
## 7.751839 97.34015 81.23176 113.44854 72.70449 121.97581  
## 7.755801 97.34015 81.13944 113.54085 72.56331 122.11699  
## 7.759763 97.34015 81.04765 113.63264 72.42293 122.25737  
## 7.763726 97.34015 80.95638 113.72392 72.28334 122.39696  
## 7.767688 97.34015 80.86561 113.81469 72.14452 122.53578  
## 7.771651 97.34015 80.77534 113.90496 72.00646 122.67384  
## 7.775613 97.34015 80.68556 113.99474 71.86915 122.81115  
## 7.779575 97.34015 80.59625 114.08404 71.73257 122.94773  
## 7.783538 97.34015 80.50743 114.17287 71.59672 123.08357  
## 7.787500 97.34015 80.41907 114.26123 71.46159 123.21871  
## 7.791462 97.34015 80.33116 114.34913 71.32715 123.35315  
## 7.795425 97.34015 80.24371 114.43658 71.19341 123.48689  
## 7.799387 97.34015 80.15671 114.52359 71.06035 123.61995  
## 7.803349 97.34015 80.07014 114.61016 70.92795 123.75235  
## 7.807312 97.34015 79.98401 114.69629 70.79622 123.88408  
## 7.811274 97.34015 79.89830 114.78200 70.66514 124.01516  
## 7.815236 97.34015 79.81301 114.86729 70.53470 124.14560  
## 7.819199 97.34015 79.72813 114.95217 70.40489 124.27541  
## 7.823161 97.34015 79.64366 115.03664 70.27570 124.40460  
## 7.827123 97.34015 79.55959 115.12071 70.14713 124.53317  
## 7.831086 97.34015 79.47592 115.20438 70.01916 124.66114  
## 7.835048 97.34015 79.39263 115.28767 69.89179 124.78851  
## 7.839010 97.34015 79.30973 115.37056 69.76501 124.91529  
## 7.842973 97.34015 79.22721 115.45308 69.63881 125.04149  
## 7.846935 97.34015 79.14507 115.53523 69.51318 125.16712  
## 7.850897 97.34015 79.06329 115.61700 69.38811 125.29219  
## 7.854860 97.34015 78.98188 115.69842 69.26360 125.41670  
## 7.858822 97.34015 78.90083 115.77947 69.13964 125.54065  
## 7.862784 97.34015 78.82013 115.86017 69.01623 125.66407  
## 7.866747 97.34015 78.73978 115.94051 68.89335 125.78695  
## 7.870709 97.34015 78.65978 116.02051 68.77099 125.90930  
## 7.874672 97.34015 78.58012 116.10018 68.64916 126.03113  
## 7.878634 97.34015 78.50080 116.17950 68.52785 126.15245  
## 7.882596 97.34015 78.42181 116.25849 68.40704 126.27326  
## 7.886559 97.34015 78.34314 116.33715 68.28674 126.39356  
## 7.890521 97.34015 78.26481 116.41549 68.16693 126.51337  
## 7.894483 97.34015 78.18679 116.49351 68.04761 126.63269  
## 7.898446 97.34015 78.10909 116.57121 67.92878 126.75152  
## 7.902408 97.34015 78.03170 116.64860 67.81042 126.86988  
## 7.906370 97.34015 77.95462 116.72568 67.69254 126.98776  
## 7.910333 97.34015 77.87784 116.80245 67.57512 127.10518  
## 7.914295 97.34015 77.80137 116.87893 67.45817 127.22213  
## 7.918257 97.34015 77.72520 116.95510 67.34167 127.33863  
## 7.922220 97.34015 77.64932 117.03098 67.22562 127.45468  
## 7.926182 97.34015 77.57373 117.10657 67.11002 127.57028  
## 7.930144 97.34015 77.49843 117.18187 66.99485 127.68544  
## 7.934107 97.34015 77.42341 117.25689 66.88013 127.80017  
## 7.938069 97.34015 77.34868 117.33162 66.76583 127.91447  
## 7.942031 97.34015 77.27422 117.40608 66.65196 128.02834  
## 7.945994 97.34015 77.20004 117.48026 66.53851 128.14179  
## 7.949956 97.34015 77.12613 117.55417 66.42547 128.25482  
## 7.953918 97.34015 77.05249 117.62781 66.31285 128.36745  
## 7.957881 97.34015 76.97912 117.70118 66.20064 128.47966  
## 7.961843 97.34015 76.90601 117.77429 66.08883 128.59147

S04 Var01, Var02 S05 Var 02

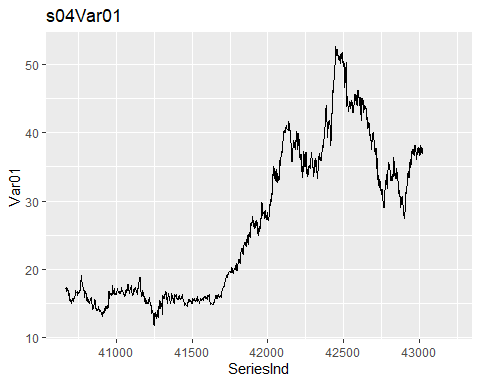
all\_set <- read.csv("https://raw.githubusercontent.com/lincarrieli/Data624/main/Project1\_Group2.csv", header = TRUE)  
s04 <- subset(all\_set, group == "S04")  
s05 <- subset(all\_set, group == "S05")

s04Var01 <- s04$Var01  
s04Var02 <- s04$Var02  
s05Var02 <- s05$Var02

Initial visualization for S04 Var02, Var03 and S04 Var 02

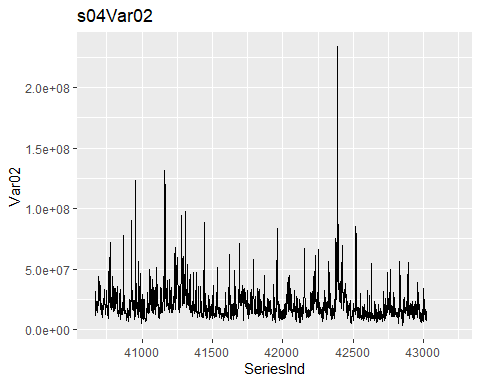
s04 %>%  
 ggplot(aes(x=SeriesInd, y=Var01)) + geom\_line() + ggtitle("s04Var01")

## Warning: Removed 140 row(s) containing missing values (geom\_path).



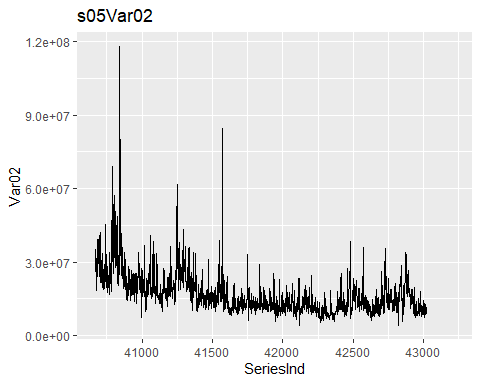
s04 %>%  
 ggplot(aes(x=SeriesInd, y=Var02)) + geom\_line() + ggtitle("s04Var02")

## Warning: Removed 140 row(s) containing missing values (geom\_path).

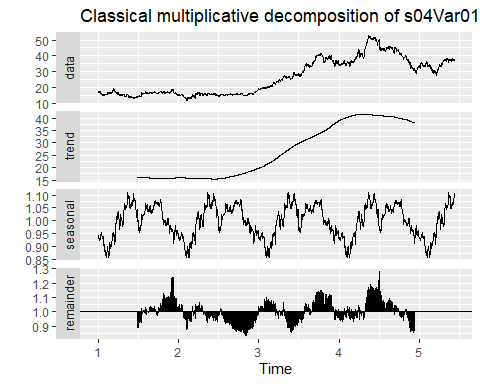


s05 %>%  
 ggplot(aes(x=SeriesInd, y=Var02)) + geom\_line() + ggtitle("s05Var02")

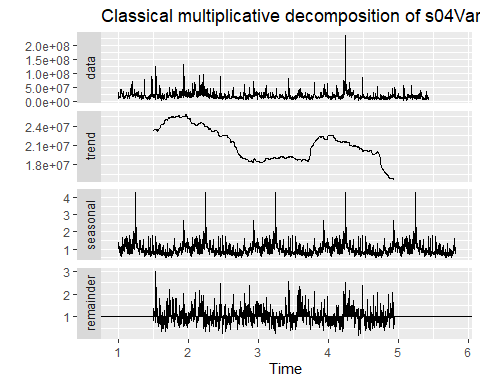
## Warning: Removed 140 row(s) containing missing values (geom\_path).



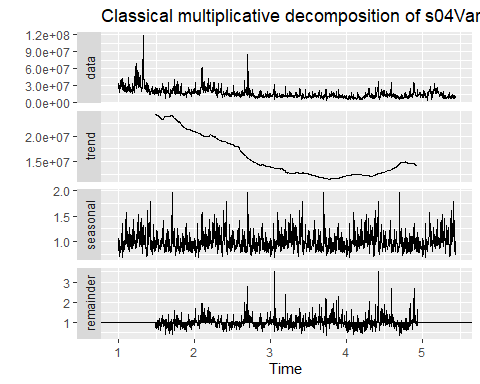
library(tseries)  
library(tidyverse)  
library(fpp2)  
library(readxl)  
s04Var01.ts <- ts(s04Var01, frequency = 365)  
s04Var01.ts1 <- na.remove(s04Var01.ts)  
s04Var01.ts1 %>%  
 decompose(type="multiplicative")%>%  
 autoplot() + xlab("Time") +  
 ggtitle("Classical multiplicative decomposition of s04Var01")



s04Var02.ts <- ts(s04Var02, frequency = 365)  
s04Var02.ts %>%  
 decompose(type="multiplicative") %>%  
 autoplot() + xlab("Time") +  
 ggtitle("Classical multiplicative decomposition of s04Var02")



s05Var02.ts <- ts(s05Var02, frequency = 365)  
s05Var02.ts1 <- na.remove(s05Var02.ts)  
s05Var02.ts1 %>%  
 decompose(type="multiplicative") %>%  
 autoplot() + xlab("Time") +  
 ggtitle("Classical multiplicative decomposition of s04Var02")



Each variable seems to display distinct characteristics. s04Var02 has no apparent trend but do exhibit seasonality based on the decomposition plot; it also has a few outliers. s04Var03 has a upward trend and starting to go downward with clear seasonality. s05Var02 shows a downward trend at the beginning of the series and seemed to remine flat before starting to go upward; three extreme outliers can be observed from the plot.

Examine the nature of gaps/missing values. Using the approach of computing the =average squared difference across gaps

SeriesInd.ts <- ts(s04$SeriesInd)  
gaps <- diff(SeriesInd.ts) > 1  
gaps <- c(FALSE, gaps)  
gaps.df <- data.frame("SeriesInd" = s04$SeriesInd, "AfterGap" = gaps)  
gaps.df <- gaps.df %>%  
 mutate("s04Var01" = s04$Var01, "s04Var01.diff" = s04$Var01 - lag(s04$Var01))  
sqdiff\_across\_gaps\_s04Var01 <- gaps.df %>%  
 filter(AfterGap) %>%  
 filter(s04Var01.diff > -50) %>%  
 select(s04Var01.diff)  
sqdiff\_across\_gaps\_s04Var01 <- sqdiff\_across\_gaps\_s04Var01^2  
sqdiff\_across\_gaps\_s04Var01 <-  
 mean(sqdiff\_across\_gaps\_s04Var01$s04Var01.diff)  
sqdiff\_across\_gaps\_s04Var01

## [1] 0.2402264

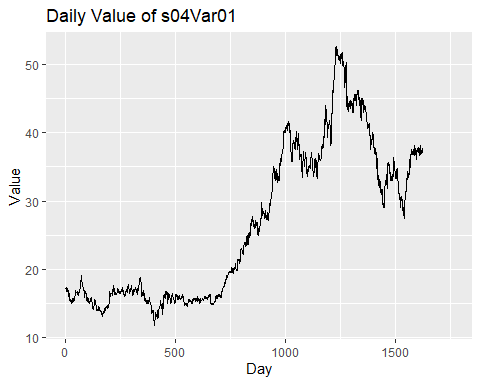
Computing the average squared difference between successive entries for s04Var01:

sqdiff\_across\_all <- gaps.df %>%  
 filter(s04Var01.diff < 50) %>%  
 select(s04Var01.diff)  
sqdiff\_across\_all <- sqdiff\_across\_all^2  
sqdiff\_across\_all <- mean(sqdiff\_across\_all$s04Var01.diff)  
sqdiff\_across\_all

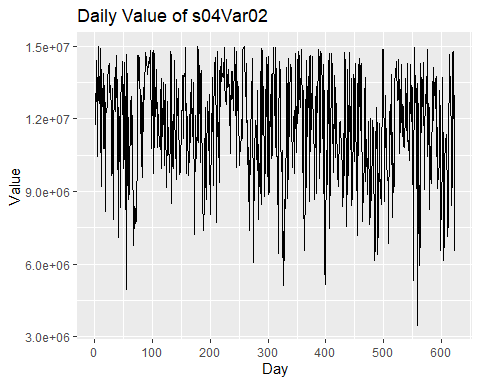
## [1] 0.2561081

The small difference in values of mean squared difference between gaps and successive values are small enough to suggest a pause in data generating process, and not missing data.

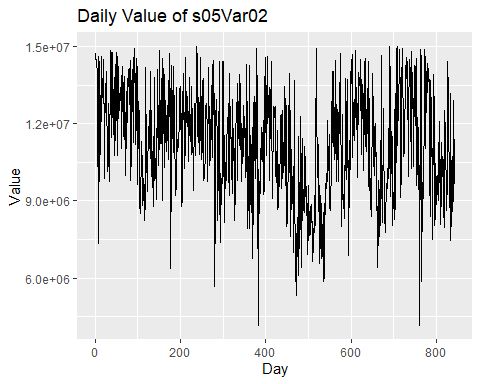
s04Var01.ts <-ts(s04Var01)  
autoplot(s04Var01.ts) +  
 xlab("Day") +  
 ylab("Value") +  
 ggtitle("Daily Value of s04Var01")



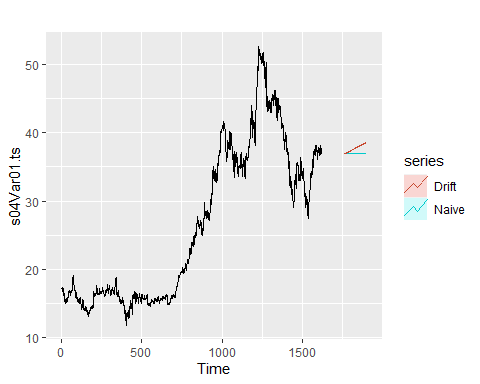
s04Var02.ts <- s04 %>%  
 filter(Var02 < 15000000) %>%  
 select(Var02) %>%  
 ts()  
autoplot(s04Var02.ts) +  
 xlab("Day") +  
 ylab("Value") +  
 ggtitle("Daily Value of s04Var02")



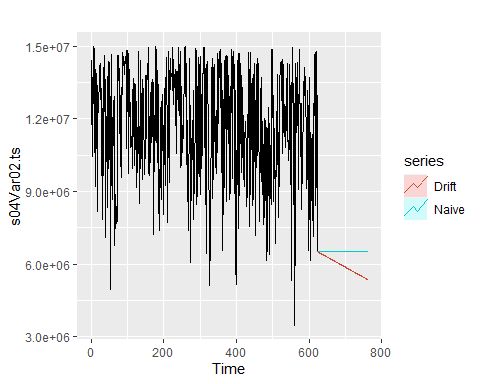
s05Var02.ts <- s05 %>%  
 filter(Var02 < 15000000) %>%  
 select(Var02) %>%  
 ts()  
autoplot(s05Var02.ts) +  
 xlab("Day") +  
 ylab("Value") +  
 ggtitle("Daily Value of s05Var02")



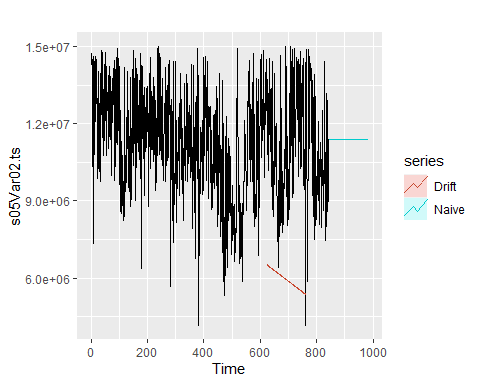
library(forecast)  
library(ggplot2)  
s04Var01\_rwf <- rwf(s04Var01.ts, h = 140)  
s04Var02\_rwf <- rwf(s04Var02.ts, h = 140)  
s05Var02\_rwf <- rwf(s05Var02.ts, h = 140)  
s04Var01\_drwf <- rwf(s04Var01.ts, h = 140, drift = TRUE)  
s04Var02\_drwf <- rwf(s04Var02.ts, h = 140, drift = TRUE)  
s05Var02\_drwf <- rwf(s04Var02.ts, h = 140, drift = TRUE)  
#s04Var01\_mean <- mean(s04Var01.ts, h = 140)  
#s04Var02\_mean <- mean(s04Var02.ts, h = 140)  
#s05Var02\_mean <- mean(s05Var02.ts, h = 140)  
autoplot(s04Var01.ts) +  
 autolayer(s04Var01\_rwf, series = "Naive", PI = FALSE) +  
 autolayer(s04Var01\_drwf, series = "Drift", PI = FALSE)



autoplot(s04Var02.ts) +  
 autolayer(s04Var02\_rwf, series = "Naive", PI = FALSE) +  
 autolayer(s04Var02\_drwf, series = "Drift", PI = FALSE)



autoplot(s05Var02.ts) +  
 autolayer(s05Var02\_rwf, series = "Naive", PI = FALSE) +  
 autolayer(s05Var02\_drwf, series = "Drift", PI = FALSE)



s04Var01\_rmse\_rwf\_nodrift <- tsCV(s04Var01.ts, rwf, drift = FALSE, h = 1)  
s04Var01\_rmse\_rwf\_nodrift <- sqrt(mean(s04Var01\_rmse\_rwf\_nodrift^2, na.rm = TRUE))  
s04Var01\_rmse\_rwf\_drift <- tsCV(s04Var01.ts, rwf, drift = TRUE, h = 1)  
s04Var01\_rmse\_rwf\_drift <- sqrt(mean(s04Var01\_rmse\_rwf\_drift^2, na.rm = TRUE))  
s04Var02\_rmse\_rwf\_nodrift <- tsCV(s04Var02.ts, rwf, drift = FALSE, h = 1)  
s04Var02\_rmse\_rwf\_nodrift <- sqrt(mean(s04Var02\_rmse\_rwf\_nodrift^2, na.rm = TRUE))  
s04Var02\_rmse\_rwf\_drift <- tsCV(s04Var02.ts, rwf, drift = TRUE, h = 1)  
s04Var02\_rmse\_rwf\_drift <- sqrt(mean(s04Var02\_rmse\_rwf\_drift^2, na.rm = TRUE))  
s05Var02\_rmse\_rwf\_nodrift <- tsCV(s05Var02.ts, rwf, drift = FALSE, h = 1)  
s05Var02\_rmse\_rwf\_nodrift <- sqrt(mean(s05Var02\_rmse\_rwf\_nodrift^2, na.rm = TRUE))  
s05Var02\_rmse\_rwf\_drift <- tsCV(s05Var02.ts, rwf, drift = TRUE, h = 1)  
s05Var02\_rmse\_rwf\_drift <- sqrt(mean(s05Var02\_rmse\_rwf\_drift^2, na.rm = TRUE))

s04Var01\_rmse\_rwf\_nodrift

## [1] 0.5059984

s04Var01\_rmse\_rwf\_drift

## [1] 0.5064454

s04Var02\_rmse\_rwf\_nodrift

## [1] 2719240

s04Var02\_rmse\_rwf\_drift

## [1] 2736915

s05Var02\_rmse\_rwf\_nodrift

## [1] 2176955

s05Var02\_rmse\_rwf\_drift

## [1] 2188323

For all models, the better performing model is random walk with no drift.

Exponential soomthing is suitable for data with clear trend or seasonality, and Holt’s linear trend method allows the forecasting of data with a trend. I apply apply both methods to all three variables and would expect Exponential smoothing would better fit s04Var02, and Holt’s would fit better with s04Var01 and s05Var02.

s04Var02\_ses <- ses(s04Var02.ts, h = 140)  
summary(s04Var02\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = s04Var02.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.116   
##   
## Initial states:  
## l = 13128916.9119   
##   
## sigma: 2216753  
##   
## AIC AICc BIC   
## 22218.72 22218.75 22232.02   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -32932.43 2213191 1781116 -4.715647 17.41825 0.8318006 0.1513413  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 624 10749006 7908123 13589888 6404250 15093761  
## 625 10749006 7889074 13608937 6375118 15122893  
## 626 10749006 7870151 13627860 6346178 15151833  
## 627 10749006 7851352 13646659 6317427 15180584  
## 628 10749006 7832674 13665337 6288861 15209150  
## 629 10749006 7814115 13683896 6260478 15237533  
## 630 10749006 7795672 13702339 6232272 15265739  
## 631 10749006 7777344 13720667 6204242 15293769  
## 632 10749006 7759129 13738882 6176383 15321628  
## 633 10749006 7741023 13756988 6148694 15349317  
## 634 10749006 7723026 13774985 6121169 15376842  
## 635 10749006 7705136 13792876 6093808 15404203  
## 636 10749006 7687349 13810662 6066607 15431405  
## 637 10749006 7669666 13828345 6039562 15458449  
## 638 10749006 7652084 13845928 6012672 15485339  
## 639 10749006 7634600 13863411 5985934 15512077  
## 640 10749006 7617215 13880796 5959345 15538666  
## 641 10749006 7599925 13898086 5932903 15565108  
## 642 10749006 7582730 13915281 5906605 15591406  
## 643 10749006 7565628 13932383 5880449 15617562  
## 644 10749006 7548617 13949394 5854433 15643578  
## 645 10749006 7531696 13966315 5828555 15669456  
## 646 10749006 7514863 13983148 5802812 15695199  
## 647 10749006 7498118 13999893 5777202 15720809  
## 648 10749006 7481459 14016552 5751724 15746287  
## 649 10749006 7464884 14033127 5726375 15771636  
## 650 10749006 7448392 14049619 5701153 15796858  
## 651 10749006 7431982 14066029 5676056 15821955  
## 652 10749006 7415653 14082358 5651083 15846928  
## 653 10749006 7399404 14098607 5626232 15871779  
## 654 10749006 7383233 14114778 5601501 15896510  
## 655 10749006 7367139 14130872 5576888 15921123  
## 656 10749006 7351122 14146889 5552391 15945620  
## 657 10749006 7335180 14162831 5528010 15970001  
## 658 10749006 7319312 14178699 5503742 15994270  
## 659 10749006 7303517 14194494 5479585 16018426  
## 660 10749006 7287794 14210217 5455539 16042472  
## 661 10749006 7272142 14225869 5431602 16066410  
## 662 10749006 7256560 14241451 5407771 16090240  
## 663 10749006 7241048 14256963 5384047 16113964  
## 664 10749006 7225604 14272408 5360427 16137584  
## 665 10749006 7210227 14287784 5336910 16161101  
## 666 10749006 7194917 14303095 5313495 16184516  
## 667 10749006 7179672 14318339 5290181 16207830  
## 668 10749006 7164492 14333519 5266965 16231046  
## 669 10749006 7149376 14348635 5243848 16254163  
## 670 10749006 7134324 14363687 5220827 16277184  
## 671 10749006 7119334 14378677 5197902 16300109  
## 672 10749006 7104405 14393606 5175071 16322941  
## 673 10749006 7089538 14408473 5152333 16345679  
## 674 10749006 7074730 14423281 5129687 16368325  
## 675 10749006 7059982 14438029 5107132 16390880  
## 676 10749006 7045293 14452718 5084666 16413345  
## 677 10749006 7030662 14467349 5062290 16435721  
## 678 10749006 7016088 14481923 5040001 16458010  
## 679 10749006 7001571 14496440 5017799 16480212  
## 680 10749006 6987110 14510901 4995682 16502329  
## 681 10749006 6972704 14525307 4973651 16524360  
## 682 10749006 6958353 14539658 4951703 16546309  
## 683 10749006 6944056 14553955 4929837 16568174  
## 684 10749006 6929813 14568199 4908054 16589957  
## 685 10749006 6915622 14582389 4886352 16611660  
## 686 10749006 6901484 14596527 4864729 16633282  
## 687 10749006 6887398 14610613 4843186 16654825  
## 688 10749006 6873362 14624649 4821721 16676290  
## 689 10749006 6859378 14638633 4800333 16697678  
## 690 10749006 6845443 14652568 4779023 16718989  
## 691 10749006 6831559 14666453 4757787 16740224  
## 692 10749006 6817723 14680288 4736627 16761384  
## 693 10749006 6803935 14694076 4715542 16782470  
## 694 10749006 6790196 14707815 4694529 16803482  
## 695 10749006 6776504 14721507 4673589 16824422  
## 696 10749006 6762860 14735151 4652722 16845290  
## 697 10749006 6749261 14748750 4631925 16866086  
## 698 10749006 6735709 14762302 4611199 16886812  
## 699 10749006 6722203 14775808 4590542 16907469  
## 700 10749006 6708741 14789270 4569955 16928056  
## 701 10749006 6695325 14802686 4549436 16948575  
## 702 10749006 6681952 14816059 4528985 16969027  
## 703 10749006 6668624 14829387 4508600 16989411  
## 704 10749006 6655339 14842672 4488282 17009729  
## 705 10749006 6642096 14855915 4468030 17029981  
## 706 10749006 6628897 14869114 4447843 17050168  
## 707 10749006 6615739 14882272 4427720 17070291  
## 708 10749006 6602624 14895387 4407662 17090349  
## 709 10749006 6589549 14908462 4387666 17110345  
## 710 10749006 6576516 14921495 4367733 17130278  
## 711 10749006 6563523 14934488 4347862 17150149  
## 712 10749006 6550570 14947441 4328053 17169958  
## 713 10749006 6537658 14960353 4308305 17189706  
## 714 10749006 6524784 14973227 4288617 17209394  
## 715 10749006 6511950 14986061 4268988 17229023  
## 716 10749006 6499155 14998856 4249420 17248592  
## 717 10749006 6486398 15011613 4229909 17268102  
## 718 10749006 6473679 15024332 4210457 17287554  
## 719 10749006 6460997 15037014 4191063 17306948  
## 720 10749006 6448354 15049658 4171726 17326285  
## 721 10749006 6435747 15062264 4152446 17345566  
## 722 10749006 6423177 15074834 4133221 17364790  
## 723 10749006 6410643 15087368 4114053 17383958  
## 724 10749006 6398146 15099865 4094940 17403072  
## 725 10749006 6385684 15112327 4075881 17422130  
## 726 10749006 6373258 15124753 4056877 17441134  
## 727 10749006 6360867 15137144 4037926 17460085  
## 728 10749006 6348511 15149501 4019029 17478982  
## 729 10749006 6336189 15161822 4000185 17497826  
## 730 10749006 6323902 15174109 3981393 17516618  
## 731 10749006 6311648 15186363 3962654 17535358  
## 732 10749006 6299429 15198582 3943965 17554046  
## 733 10749006 6287243 15210768 3925329 17572683  
## 734 10749006 6275090 15222921 3906742 17591269  
## 735 10749006 6262970 15235041 3888207 17609804  
## 736 10749006 6250883 15247128 3869721 17628290  
## 737 10749006 6238828 15259183 3851285 17646727  
## 738 10749006 6226805 15271206 3832897 17665114  
## 739 10749006 6214815 15283196 3814559 17683452  
## 740 10749006 6202855 15295156 3796269 17701742  
## 741 10749006 6190928 15307083 3778027 17719984  
## 742 10749006 6179031 15318980 3759832 17738179  
## 743 10749006 6167165 15330846 3741685 17756326  
## 744 10749006 6155330 15342681 3723585 17774426  
## 745 10749006 6143525 15354486 3705531 17792480  
## 746 10749006 6131751 15366260 3687524 17810488  
## 747 10749006 6120006 15378005 3669562 17828449  
## 748 10749006 6108291 15389720 3651645 17846366  
## 749 10749006 6096606 15401405 3633774 17864237  
## 750 10749006 6084950 15413061 3615948 17882064  
## 751 10749006 6073323 15424689 3598166 17899846  
## 752 10749006 6061724 15436287 3580428 17917583  
## 753 10749006 6050155 15447856 3562733 17935278  
## 754 10749006 6038614 15459397 3545083 17952928  
## 755 10749006 6027101 15470910 3527475 17970536  
## 756 10749006 6015616 15482395 3509910 17988101  
## 757 10749006 6004159 15493853 3492388 18005623  
## 758 10749006 5992729 15505282 3474908 18023103  
## 759 10749006 5981327 15516684 3457470 18040541  
## 760 10749006 5969952 15528059 3440074 18057937  
## 761 10749006 5958604 15539407 3422718 18075293  
## 762 10749006 5947283 15550728 3405404 18092607  
## 763 10749006 5935988 15562023 3388131 18109880

s05Var02\_ses <- ses(s05Var02.ts, h = 140)  
summary(s05Var02\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = s05Var02.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.1885   
##   
## Initial states:  
## l = 13643262.0247   
##   
## sigma: 1846285  
##   
## AIC AICc BIC   
## 30010.03 30010.05 30024.24   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -21240.31 1844093 1449099 -3.056439 13.92863 0.8593412 0.1636575  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 844 10268890 7902781 12634998 6650238 13887541  
## 845 10268890 7861131 12676648 6586541 13951238  
## 846 10268890 7820190 12717589 6523927 14013852  
## 847 10268890 7779922 12757857 6462343 14075436  
## 848 10268890 7740296 12797483 6401739 14136040  
## 849 10268890 7701281 12836498 6342071 14195708  
## 850 10268890 7662850 12874929 6283296 14254483  
## 851 10268890 7624977 12912802 6225375 14312404  
## 852 10268890 7587640 12950139 6168272 14369507  
## 853 10268890 7550815 12986964 6111954 14425826  
## 854 10268890 7514483 13023296 6056388 14481391  
## 855 10268890 7478623 13059156 6001546 14536233  
## 856 10268890 7443219 13094560 5947399 14590380  
## 857 10268890 7408253 13129526 5893923 14643856  
## 858 10268890 7373709 13164070 5841093 14696686  
## 859 10268890 7339572 13198207 5788885 14748894  
## 860 10268890 7305829 13231950 5737279 14800500  
## 861 10268890 7272466 13265313 5686255 14851525  
## 862 10268890 7239470 13298309 5635792 14901987  
## 863 10268890 7206829 13330950 5585872 14951907  
## 864 10268890 7174533 13363246 5536480 15001299  
## 865 10268890 7142570 13395209 5487597 15050182  
## 866 10268890 7110931 13426848 5439209 15098570  
## 867 10268890 7079606 13458173 5391302 15146477  
## 868 10268890 7048586 13489193 5343860 15193919  
## 869 10268890 7017861 13519918 5296871 15240908  
## 870 10268890 6987424 13550355 5250322 15287457  
## 871 10268890 6957267 13580512 5204200 15333579  
## 872 10268890 6927382 13610397 5158495 15379284  
## 873 10268890 6897762 13640017 5113195 15424584  
## 874 10268890 6868400 13669379 5068290 15469489  
## 875 10268890 6839290 13698490 5023769 15514010  
## 876 10268890 6810424 13727355 4979623 15558156  
## 877 10268890 6781797 13755982 4935842 15601937  
## 878 10268890 6753404 13784376 4892418 15645361  
## 879 10268890 6725237 13812542 4849341 15688438  
## 880 10268890 6697293 13840486 4806605 15731174  
## 881 10268890 6669566 13868213 4764200 15773579  
## 882 10268890 6642051 13895728 4722119 15815660  
## 883 10268890 6614743 13923036 4680355 15857424  
## 884 10268890 6587638 13950141 4638901 15898878  
## 885 10268890 6560731 13977048 4597750 15940029  
## 886 10268890 6534017 14003762 4556895 15980884  
## 887 10268890 6507494 14030285 4516331 16021448  
## 888 10268890 6481156 14056623 4476051 16061728  
## 889 10268890 6455000 14082779 4436048 16101731  
## 890 10268890 6429022 14108757 4396319 16141460  
## 891 10268890 6403218 14134561 4356856 16180923  
## 892 10268890 6377586 14160193 4317655 16220124  
## 893 10268890 6352122 14185657 4278710 16259069  
## 894 10268890 6326822 14210957 4240017 16297762  
## 895 10268890 6301683 14236096 4201571 16336208  
## 896 10268890 6276703 14261076 4163367 16374412  
## 897 10268890 6251878 14285901 4125400 16412379  
## 898 10268890 6227205 14310574 4087667 16450112  
## 899 10268890 6202682 14335097 4050162 16487617  
## 900 10268890 6178306 14359473 4012883 16524896  
## 901 10268890 6154075 14383704 3975824 16561955  
## 902 10268890 6129985 14407794 3938982 16598797  
## 903 10268890 6106035 14431744 3902353 16635426  
## 904 10268890 6082222 14455557 3865934 16671845  
## 905 10268890 6058544 14479235 3829721 16708058  
## 906 10268890 6034998 14502781 3793711 16744069  
## 907 10268890 6011582 14526197 3757899 16779880  
## 908 10268890 5988294 14549485 3722284 16815495  
## 909 10268890 5965132 14572647 3686861 16850918  
## 910 10268890 5942094 14595685 3651628 16886152  
## 911 10268890 5919179 14618600 3616581 16921198  
## 912 10268890 5896383 14641396 3581718 16956061  
## 913 10268890 5873706 14664073 3547036 16990743  
## 914 10268890 5851145 14686634 3512532 17025247  
## 915 10268890 5828698 14709081 3478203 17059576  
## 916 10268890 5806365 14731414 3444047 17093732  
## 917 10268890 5784143 14753636 3410061 17127718  
## 918 10268890 5762030 14775749 3376243 17161537  
## 919 10268890 5740025 14797754 3342589 17195190  
## 920 10268890 5718127 14819652 3309099 17228680  
## 921 10268890 5696333 14841446 3275768 17262011  
## 922 10268890 5674643 14863136 3242596 17295183  
## 923 10268890 5653055 14884724 3209580 17328199  
## 924 10268890 5631568 14906211 3176718 17361061  
## 925 10268890 5610179 14927600 3144007 17393772  
## 926 10268890 5588888 14948891 3111445 17426334  
## 927 10268890 5567694 14970085 3079031 17458748  
## 928 10268890 5546595 14991184 3046763 17491016  
## 929 10268890 5525589 15012190 3014638 17523141  
## 930 10268890 5504676 15033103 2982654 17555125  
## 931 10268890 5483855 15053924 2950811 17586968  
## 932 10268890 5463124 15074655 2919105 17618674  
## 933 10268890 5442482 15095297 2887536 17650243  
## 934 10268890 5421928 15115852 2856101 17681678  
## 935 10268890 5401460 15136319 2824799 17712980  
## 936 10268890 5381078 15156701 2793627 17744152  
## 937 10268890 5360781 15176998 2762586 17775193  
## 938 10268890 5340568 15197211 2731672 17806107  
## 939 10268890 5320437 15217342 2700884 17836895  
## 940 10268890 5300388 15237391 2670221 17867558  
## 941 10268890 5280419 15257360 2639682 17898097  
## 942 10268890 5260530 15277249 2609264 17928515  
## 943 10268890 5240719 15297060 2578967 17958813  
## 944 10268890 5220986 15316793 2548788 17988991  
## 945 10268890 5201330 15336449 2518727 18019052  
## 946 10268890 5181751 15356029 2488782 18048997  
## 947 10268890 5162246 15375533 2458952 18078827  
## 948 10268890 5142815 15394964 2429235 18108544  
## 949 10268890 5123458 15414321 2399631 18138148  
## 950 10268890 5104173 15433606 2370137 18167642  
## 951 10268890 5084960 15452819 2340753 18197026  
## 952 10268890 5065818 15471961 2311478 18226301  
## 953 10268890 5046746 15491033 2282310 18255469  
## 954 10268890 5027744 15510036 2253248 18284531  
## 955 10268890 5008810 15528969 2224292 18313487  
## 956 10268890 4989944 15547835 2195439 18342340  
## 957 10268890 4971145 15566634 2166688 18371091  
## 958 10268890 4952413 15585366 2138040 18399739  
## 959 10268890 4933746 15604033 2109492 18428287  
## 960 10268890 4915145 15622634 2081044 18456736  
## 961 10268890 4896608 15641171 2052694 18485085  
## 962 10268890 4878134 15659645 2024441 18513338  
## 963 10268890 4859724 15678055 1996285 18541494  
## 964 10268890 4841377 15696402 1968225 18569554  
## 965 10268890 4823091 15714688 1940259 18597520  
## 966 10268890 4804866 15732913 1912387 18625393  
## 967 10268890 4786702 15751077 1884607 18653172  
## 968 10268890 4768598 15769181 1856919 18680860  
## 969 10268890 4750553 15787226 1829322 18708457  
## 970 10268890 4732567 15805212 1801815 18735964  
## 971 10268890 4714639 15823140 1774397 18763382  
## 972 10268890 4696769 15841010 1747067 18790712  
## 973 10268890 4678956 15858823 1719824 18817955  
## 974 10268890 4661200 15876579 1692668 18845111  
## 975 10268890 4643500 15894279 1665598 18872181  
## 976 10268890 4625855 15911924 1638613 18899166  
## 977 10268890 4608265 15929514 1611712 18926067  
## 978 10268890 4590730 15947049 1584894 18952885  
## 979 10268890 4573249 15964530 1558158 18979621  
## 980 10268890 4555821 15981958 1531505 19006274  
## 981 10268890 4538446 15999333 1504933 19032847  
## 982 10268890 4521124 16016655 1478440 19059339  
## 983 10268890 4503854 16033925 1452028 19085751

The optimized simple exponential smoothing method computed , making this method almost indistinguishable from the random-walk forecast.

s04Var02\_ses <- ses(s04Var02.ts, h = 140)  
summary(s04Var02\_ses)

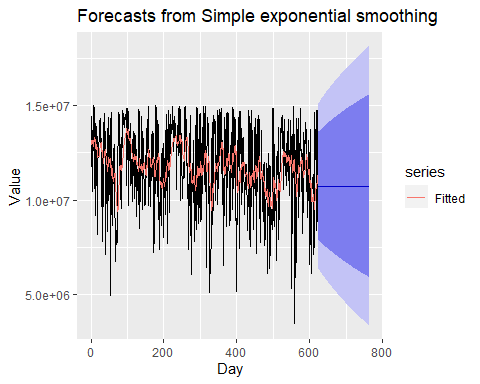
##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = s04Var02.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.116   
##   
## Initial states:  
## l = 13128916.9119   
##   
## sigma: 2216753  
##   
## AIC AICc BIC   
## 22218.72 22218.75 22232.02   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -32932.43 2213191 1781116 -4.715647 17.41825 0.8318006 0.1513413  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 624 10749006 7908123 13589888 6404250 15093761  
## 625 10749006 7889074 13608937 6375118 15122893  
## 626 10749006 7870151 13627860 6346178 15151833  
## 627 10749006 7851352 13646659 6317427 15180584  
## 628 10749006 7832674 13665337 6288861 15209150  
## 629 10749006 7814115 13683896 6260478 15237533  
## 630 10749006 7795672 13702339 6232272 15265739  
## 631 10749006 7777344 13720667 6204242 15293769  
## 632 10749006 7759129 13738882 6176383 15321628  
## 633 10749006 7741023 13756988 6148694 15349317  
## 634 10749006 7723026 13774985 6121169 15376842  
## 635 10749006 7705136 13792876 6093808 15404203  
## 636 10749006 7687349 13810662 6066607 15431405  
## 637 10749006 7669666 13828345 6039562 15458449  
## 638 10749006 7652084 13845928 6012672 15485339  
## 639 10749006 7634600 13863411 5985934 15512077  
## 640 10749006 7617215 13880796 5959345 15538666  
## 641 10749006 7599925 13898086 5932903 15565108  
## 642 10749006 7582730 13915281 5906605 15591406  
## 643 10749006 7565628 13932383 5880449 15617562  
## 644 10749006 7548617 13949394 5854433 15643578  
## 645 10749006 7531696 13966315 5828555 15669456  
## 646 10749006 7514863 13983148 5802812 15695199  
## 647 10749006 7498118 13999893 5777202 15720809  
## 648 10749006 7481459 14016552 5751724 15746287  
## 649 10749006 7464884 14033127 5726375 15771636  
## 650 10749006 7448392 14049619 5701153 15796858  
## 651 10749006 7431982 14066029 5676056 15821955  
## 652 10749006 7415653 14082358 5651083 15846928  
## 653 10749006 7399404 14098607 5626232 15871779  
## 654 10749006 7383233 14114778 5601501 15896510  
## 655 10749006 7367139 14130872 5576888 15921123  
## 656 10749006 7351122 14146889 5552391 15945620  
## 657 10749006 7335180 14162831 5528010 15970001  
## 658 10749006 7319312 14178699 5503742 15994270  
## 659 10749006 7303517 14194494 5479585 16018426  
## 660 10749006 7287794 14210217 5455539 16042472  
## 661 10749006 7272142 14225869 5431602 16066410  
## 662 10749006 7256560 14241451 5407771 16090240  
## 663 10749006 7241048 14256963 5384047 16113964  
## 664 10749006 7225604 14272408 5360427 16137584  
## 665 10749006 7210227 14287784 5336910 16161101  
## 666 10749006 7194917 14303095 5313495 16184516  
## 667 10749006 7179672 14318339 5290181 16207830  
## 668 10749006 7164492 14333519 5266965 16231046  
## 669 10749006 7149376 14348635 5243848 16254163  
## 670 10749006 7134324 14363687 5220827 16277184  
## 671 10749006 7119334 14378677 5197902 16300109  
## 672 10749006 7104405 14393606 5175071 16322941  
## 673 10749006 7089538 14408473 5152333 16345679  
## 674 10749006 7074730 14423281 5129687 16368325  
## 675 10749006 7059982 14438029 5107132 16390880  
## 676 10749006 7045293 14452718 5084666 16413345  
## 677 10749006 7030662 14467349 5062290 16435721  
## 678 10749006 7016088 14481923 5040001 16458010  
## 679 10749006 7001571 14496440 5017799 16480212  
## 680 10749006 6987110 14510901 4995682 16502329  
## 681 10749006 6972704 14525307 4973651 16524360  
## 682 10749006 6958353 14539658 4951703 16546309  
## 683 10749006 6944056 14553955 4929837 16568174  
## 684 10749006 6929813 14568199 4908054 16589957  
## 685 10749006 6915622 14582389 4886352 16611660  
## 686 10749006 6901484 14596527 4864729 16633282  
## 687 10749006 6887398 14610613 4843186 16654825  
## 688 10749006 6873362 14624649 4821721 16676290  
## 689 10749006 6859378 14638633 4800333 16697678  
## 690 10749006 6845443 14652568 4779023 16718989  
## 691 10749006 6831559 14666453 4757787 16740224  
## 692 10749006 6817723 14680288 4736627 16761384  
## 693 10749006 6803935 14694076 4715542 16782470  
## 694 10749006 6790196 14707815 4694529 16803482  
## 695 10749006 6776504 14721507 4673589 16824422  
## 696 10749006 6762860 14735151 4652722 16845290  
## 697 10749006 6749261 14748750 4631925 16866086  
## 698 10749006 6735709 14762302 4611199 16886812  
## 699 10749006 6722203 14775808 4590542 16907469  
## 700 10749006 6708741 14789270 4569955 16928056  
## 701 10749006 6695325 14802686 4549436 16948575  
## 702 10749006 6681952 14816059 4528985 16969027  
## 703 10749006 6668624 14829387 4508600 16989411  
## 704 10749006 6655339 14842672 4488282 17009729  
## 705 10749006 6642096 14855915 4468030 17029981  
## 706 10749006 6628897 14869114 4447843 17050168  
## 707 10749006 6615739 14882272 4427720 17070291  
## 708 10749006 6602624 14895387 4407662 17090349  
## 709 10749006 6589549 14908462 4387666 17110345  
## 710 10749006 6576516 14921495 4367733 17130278  
## 711 10749006 6563523 14934488 4347862 17150149  
## 712 10749006 6550570 14947441 4328053 17169958  
## 713 10749006 6537658 14960353 4308305 17189706  
## 714 10749006 6524784 14973227 4288617 17209394  
## 715 10749006 6511950 14986061 4268988 17229023  
## 716 10749006 6499155 14998856 4249420 17248592  
## 717 10749006 6486398 15011613 4229909 17268102  
## 718 10749006 6473679 15024332 4210457 17287554  
## 719 10749006 6460997 15037014 4191063 17306948  
## 720 10749006 6448354 15049658 4171726 17326285  
## 721 10749006 6435747 15062264 4152446 17345566  
## 722 10749006 6423177 15074834 4133221 17364790  
## 723 10749006 6410643 15087368 4114053 17383958  
## 724 10749006 6398146 15099865 4094940 17403072  
## 725 10749006 6385684 15112327 4075881 17422130  
## 726 10749006 6373258 15124753 4056877 17441134  
## 727 10749006 6360867 15137144 4037926 17460085  
## 728 10749006 6348511 15149501 4019029 17478982  
## 729 10749006 6336189 15161822 4000185 17497826  
## 730 10749006 6323902 15174109 3981393 17516618  
## 731 10749006 6311648 15186363 3962654 17535358  
## 732 10749006 6299429 15198582 3943965 17554046  
## 733 10749006 6287243 15210768 3925329 17572683  
## 734 10749006 6275090 15222921 3906742 17591269  
## 735 10749006 6262970 15235041 3888207 17609804  
## 736 10749006 6250883 15247128 3869721 17628290  
## 737 10749006 6238828 15259183 3851285 17646727  
## 738 10749006 6226805 15271206 3832897 17665114  
## 739 10749006 6214815 15283196 3814559 17683452  
## 740 10749006 6202855 15295156 3796269 17701742  
## 741 10749006 6190928 15307083 3778027 17719984  
## 742 10749006 6179031 15318980 3759832 17738179  
## 743 10749006 6167165 15330846 3741685 17756326  
## 744 10749006 6155330 15342681 3723585 17774426  
## 745 10749006 6143525 15354486 3705531 17792480  
## 746 10749006 6131751 15366260 3687524 17810488  
## 747 10749006 6120006 15378005 3669562 17828449  
## 748 10749006 6108291 15389720 3651645 17846366  
## 749 10749006 6096606 15401405 3633774 17864237  
## 750 10749006 6084950 15413061 3615948 17882064  
## 751 10749006 6073323 15424689 3598166 17899846  
## 752 10749006 6061724 15436287 3580428 17917583  
## 753 10749006 6050155 15447856 3562733 17935278  
## 754 10749006 6038614 15459397 3545083 17952928  
## 755 10749006 6027101 15470910 3527475 17970536  
## 756 10749006 6015616 15482395 3509910 17988101  
## 757 10749006 6004159 15493853 3492388 18005623  
## 758 10749006 5992729 15505282 3474908 18023103  
## 759 10749006 5981327 15516684 3457470 18040541  
## 760 10749006 5969952 15528059 3440074 18057937  
## 761 10749006 5958604 15539407 3422718 18075293  
## 762 10749006 5947283 15550728 3405404 18092607  
## 763 10749006 5935988 15562023 3388131 18109880

s05Var02\_ses <- ses(s05Var02.ts, h = 140)  
summary(s05Var02\_ses)

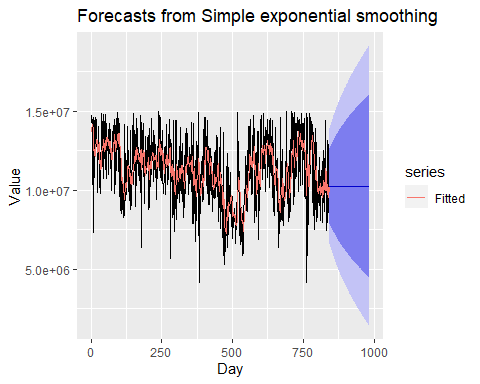
##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = s05Var02.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.1885   
##   
## Initial states:  
## l = 13643262.0247   
##   
## sigma: 1846285  
##   
## AIC AICc BIC   
## 30010.03 30010.05 30024.24   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -21240.31 1844093 1449099 -3.056439 13.92863 0.8593412 0.1636575  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 844 10268890 7902781 12634998 6650238 13887541  
## 845 10268890 7861131 12676648 6586541 13951238  
## 846 10268890 7820190 12717589 6523927 14013852  
## 847 10268890 7779922 12757857 6462343 14075436  
## 848 10268890 7740296 12797483 6401739 14136040  
## 849 10268890 7701281 12836498 6342071 14195708  
## 850 10268890 7662850 12874929 6283296 14254483  
## 851 10268890 7624977 12912802 6225375 14312404  
## 852 10268890 7587640 12950139 6168272 14369507  
## 853 10268890 7550815 12986964 6111954 14425826  
## 854 10268890 7514483 13023296 6056388 14481391  
## 855 10268890 7478623 13059156 6001546 14536233  
## 856 10268890 7443219 13094560 5947399 14590380  
## 857 10268890 7408253 13129526 5893923 14643856  
## 858 10268890 7373709 13164070 5841093 14696686  
## 859 10268890 7339572 13198207 5788885 14748894  
## 860 10268890 7305829 13231950 5737279 14800500  
## 861 10268890 7272466 13265313 5686255 14851525  
## 862 10268890 7239470 13298309 5635792 14901987  
## 863 10268890 7206829 13330950 5585872 14951907  
## 864 10268890 7174533 13363246 5536480 15001299  
## 865 10268890 7142570 13395209 5487597 15050182  
## 866 10268890 7110931 13426848 5439209 15098570  
## 867 10268890 7079606 13458173 5391302 15146477  
## 868 10268890 7048586 13489193 5343860 15193919  
## 869 10268890 7017861 13519918 5296871 15240908  
## 870 10268890 6987424 13550355 5250322 15287457  
## 871 10268890 6957267 13580512 5204200 15333579  
## 872 10268890 6927382 13610397 5158495 15379284  
## 873 10268890 6897762 13640017 5113195 15424584  
## 874 10268890 6868400 13669379 5068290 15469489  
## 875 10268890 6839290 13698490 5023769 15514010  
## 876 10268890 6810424 13727355 4979623 15558156  
## 877 10268890 6781797 13755982 4935842 15601937  
## 878 10268890 6753404 13784376 4892418 15645361  
## 879 10268890 6725237 13812542 4849341 15688438  
## 880 10268890 6697293 13840486 4806605 15731174  
## 881 10268890 6669566 13868213 4764200 15773579  
## 882 10268890 6642051 13895728 4722119 15815660  
## 883 10268890 6614743 13923036 4680355 15857424  
## 884 10268890 6587638 13950141 4638901 15898878  
## 885 10268890 6560731 13977048 4597750 15940029  
## 886 10268890 6534017 14003762 4556895 15980884  
## 887 10268890 6507494 14030285 4516331 16021448  
## 888 10268890 6481156 14056623 4476051 16061728  
## 889 10268890 6455000 14082779 4436048 16101731  
## 890 10268890 6429022 14108757 4396319 16141460  
## 891 10268890 6403218 14134561 4356856 16180923  
## 892 10268890 6377586 14160193 4317655 16220124  
## 893 10268890 6352122 14185657 4278710 16259069  
## 894 10268890 6326822 14210957 4240017 16297762  
## 895 10268890 6301683 14236096 4201571 16336208  
## 896 10268890 6276703 14261076 4163367 16374412  
## 897 10268890 6251878 14285901 4125400 16412379  
## 898 10268890 6227205 14310574 4087667 16450112  
## 899 10268890 6202682 14335097 4050162 16487617  
## 900 10268890 6178306 14359473 4012883 16524896  
## 901 10268890 6154075 14383704 3975824 16561955  
## 902 10268890 6129985 14407794 3938982 16598797  
## 903 10268890 6106035 14431744 3902353 16635426  
## 904 10268890 6082222 14455557 3865934 16671845  
## 905 10268890 6058544 14479235 3829721 16708058  
## 906 10268890 6034998 14502781 3793711 16744069  
## 907 10268890 6011582 14526197 3757899 16779880  
## 908 10268890 5988294 14549485 3722284 16815495  
## 909 10268890 5965132 14572647 3686861 16850918  
## 910 10268890 5942094 14595685 3651628 16886152  
## 911 10268890 5919179 14618600 3616581 16921198  
## 912 10268890 5896383 14641396 3581718 16956061  
## 913 10268890 5873706 14664073 3547036 16990743  
## 914 10268890 5851145 14686634 3512532 17025247  
## 915 10268890 5828698 14709081 3478203 17059576  
## 916 10268890 5806365 14731414 3444047 17093732  
## 917 10268890 5784143 14753636 3410061 17127718  
## 918 10268890 5762030 14775749 3376243 17161537  
## 919 10268890 5740025 14797754 3342589 17195190  
## 920 10268890 5718127 14819652 3309099 17228680  
## 921 10268890 5696333 14841446 3275768 17262011  
## 922 10268890 5674643 14863136 3242596 17295183  
## 923 10268890 5653055 14884724 3209580 17328199  
## 924 10268890 5631568 14906211 3176718 17361061  
## 925 10268890 5610179 14927600 3144007 17393772  
## 926 10268890 5588888 14948891 3111445 17426334  
## 927 10268890 5567694 14970085 3079031 17458748  
## 928 10268890 5546595 14991184 3046763 17491016  
## 929 10268890 5525589 15012190 3014638 17523141  
## 930 10268890 5504676 15033103 2982654 17555125  
## 931 10268890 5483855 15053924 2950811 17586968  
## 932 10268890 5463124 15074655 2919105 17618674  
## 933 10268890 5442482 15095297 2887536 17650243  
## 934 10268890 5421928 15115852 2856101 17681678  
## 935 10268890 5401460 15136319 2824799 17712980  
## 936 10268890 5381078 15156701 2793627 17744152  
## 937 10268890 5360781 15176998 2762586 17775193  
## 938 10268890 5340568 15197211 2731672 17806107  
## 939 10268890 5320437 15217342 2700884 17836895  
## 940 10268890 5300388 15237391 2670221 17867558  
## 941 10268890 5280419 15257360 2639682 17898097  
## 942 10268890 5260530 15277249 2609264 17928515  
## 943 10268890 5240719 15297060 2578967 17958813  
## 944 10268890 5220986 15316793 2548788 17988991  
## 945 10268890 5201330 15336449 2518727 18019052  
## 946 10268890 5181751 15356029 2488782 18048997  
## 947 10268890 5162246 15375533 2458952 18078827  
## 948 10268890 5142815 15394964 2429235 18108544  
## 949 10268890 5123458 15414321 2399631 18138148  
## 950 10268890 5104173 15433606 2370137 18167642  
## 951 10268890 5084960 15452819 2340753 18197026  
## 952 10268890 5065818 15471961 2311478 18226301  
## 953 10268890 5046746 15491033 2282310 18255469  
## 954 10268890 5027744 15510036 2253248 18284531  
## 955 10268890 5008810 15528969 2224292 18313487  
## 956 10268890 4989944 15547835 2195439 18342340  
## 957 10268890 4971145 15566634 2166688 18371091  
## 958 10268890 4952413 15585366 2138040 18399739  
## 959 10268890 4933746 15604033 2109492 18428287  
## 960 10268890 4915145 15622634 2081044 18456736  
## 961 10268890 4896608 15641171 2052694 18485085  
## 962 10268890 4878134 15659645 2024441 18513338  
## 963 10268890 4859724 15678055 1996285 18541494  
## 964 10268890 4841377 15696402 1968225 18569554  
## 965 10268890 4823091 15714688 1940259 18597520  
## 966 10268890 4804866 15732913 1912387 18625393  
## 967 10268890 4786702 15751077 1884607 18653172  
## 968 10268890 4768598 15769181 1856919 18680860  
## 969 10268890 4750553 15787226 1829322 18708457  
## 970 10268890 4732567 15805212 1801815 18735964  
## 971 10268890 4714639 15823140 1774397 18763382  
## 972 10268890 4696769 15841010 1747067 18790712  
## 973 10268890 4678956 15858823 1719824 18817955  
## 974 10268890 4661200 15876579 1692668 18845111  
## 975 10268890 4643500 15894279 1665598 18872181  
## 976 10268890 4625855 15911924 1638613 18899166  
## 977 10268890 4608265 15929514 1611712 18926067  
## 978 10268890 4590730 15947049 1584894 18952885  
## 979 10268890 4573249 15964530 1558158 18979621  
## 980 10268890 4555821 15981958 1531505 19006274  
## 981 10268890 4538446 15999333 1504933 19032847  
## 982 10268890 4521124 16016655 1478440 19059339  
## 983 10268890 4503854 16033925 1452028 19085751

For s04Var02 and s05Var02, the , indicating that some weight is given to observations from the more distant past.

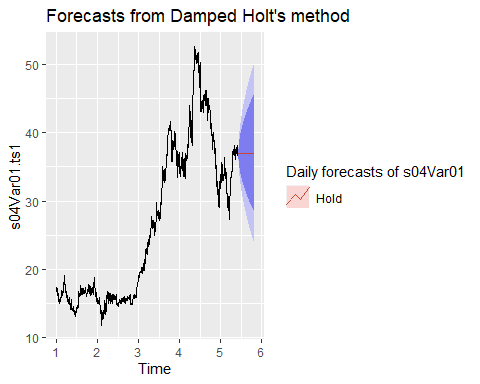
s04Var02\_1.ts <- s04Var02.ts %>%  
 na.remove()  
s04Var02\_ses <- ses(s04Var02\_1.ts, h = 140)  
autoplot(s04Var02\_ses) +  
 autolayer(fitted(s04Var02\_ses), series="Fitted") +  
 ylab("Value") + xlab("Day")



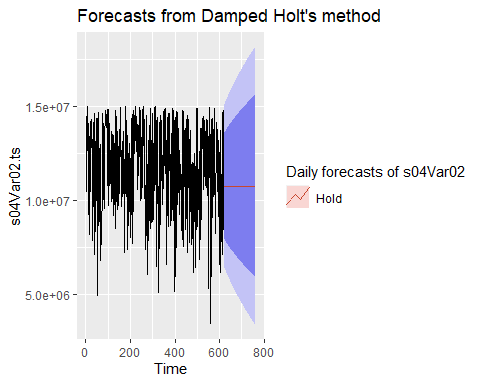
s05Var02\_1.ts <- s05Var02.ts %>%  
 na.remove()  
s05Var02\_ses <- ses(s05Var02\_1.ts, h = 140)  
autoplot(s05Var02\_ses) +  
 autolayer(fitted(s05Var02\_ses), series="Fitted") +  
 ylab("Value") + xlab("Day")



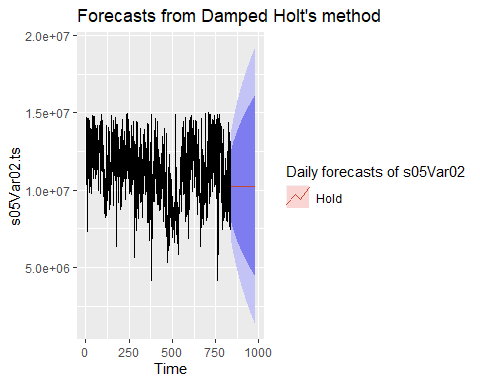
s04VAr01\_hw <- holt(s04Var01.ts1,  
 damped = TRUE, h=140)  
autoplot(s04VAr01\_hw) +  
 autolayer(s04VAr01\_hw, series="Hold", PI=FALSE)+  
 guides(colour=guide\_legend(title="Daily forecasts of s04Var01"))



s04VAr02\_hw <- holt(s04Var02.ts,  
 damped = TRUE, h=140)  
autoplot(s04VAr02\_hw) +  
 autolayer(s04VAr02\_hw, series="Hold", PI=FALSE)+  
 guides(colour=guide\_legend(title="Daily forecasts of s04Var02"))

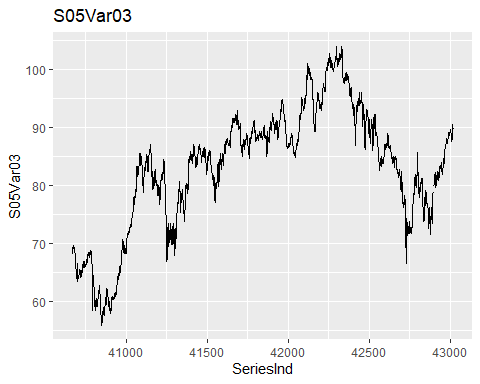


s05VAr02\_hw <- holt(s05Var02.ts,  
 damped = TRUE, h=140)  
autoplot(s05VAr02\_hw) +  
 autolayer(s05VAr02\_hw, series="Hold", PI=FALSE)+  
 guides(colour=guide\_legend(title="Daily forecasts of s05Var02"))

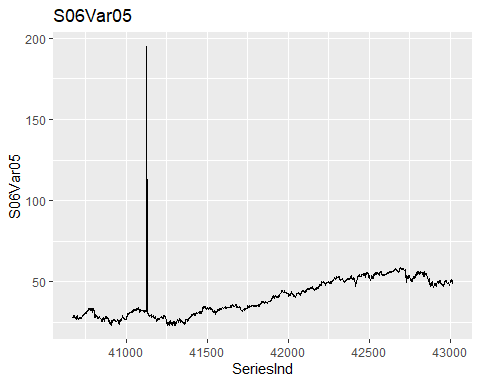


This section contains initial visualizations of S05Var03, S06Var05, and S06Var07. These visualizations provide the basis for initial commentary and suggest a roadmap for the analysis that comprises the remainder of this file.

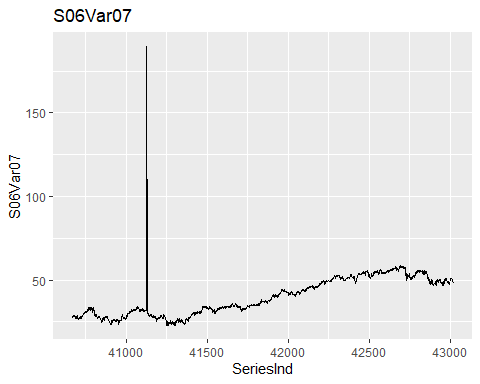
raw %>%  
 ggplot(aes(x = SeriesInd, y = S05Var03)) +  
 geom\_line() +  
 ggtitle("S05Var03")



raw %>%  
 ggplot(aes(x = SeriesInd, y = S06Var05)) +  
 geom\_line() +  
 ggtitle("S06Var05")



raw %>%  
 ggplot(aes(x = SeriesInd, y = S06Var07)) +  
 geom\_line() +  
 ggtitle("S06Var07")



All three variables closely resemble a random walk. The variables from group S06 closely resemble each other. None of the variables exhibits obvious seasonality or cyclicity. None of the variables is stationary. The group S06 variables appear to trend upward after SeriesInd = 41250, but because the data are compressed toward the bottom of the grid due to a small number of extreme outliers, it’s hard to be sure at this early stage. For all three variables, variability in the data does not appear to depend on the level of the data.

Examining the numeric data reveals gaps in the SeriesInd column. These gaps mostly occur at regular intervals, and could represent weekends and holidays in a calendar.

Do the gaps represent a pause in the process that generated this data? Or do the gaps conceal unknown values in the time series? If the average change in value across these gaps is larger than the typical difference between a value and its lag, then there’s reason to think these gaps represent missing data. If the average change across these gaps is approximately equal to the typical change from one value to the next, then the gaps probably represent a pause in the data-generating process.

Computing the average squared difference across gaps for S06Var05:

SeriesInd.ts <- ts(raw$SeriesInd)  
gaps <- diff(SeriesInd.ts) > 1  
gaps <- c(FALSE, gaps)  
gaps.df <- data.frame("SeriesInd" = raw$SeriesInd, "AfterGap" = gaps)  
gaps.df <- gaps.df %>%  
 mutate("S06Var05" = raw$S06Var05, "S06Var05.diff" = raw$S06Var05 - lag(raw$S06Var05))  
sqdiff\_across\_gaps\_S06Var05 <- gaps.df %>%  
 filter(AfterGap) %>%  
 filter(S06Var05.diff > -50) %>%  
 select(S06Var05.diff)  
sqdiff\_across\_gaps\_S06Var05 <- sqdiff\_across\_gaps\_S06Var05^2  
sqdiff\_across\_gaps\_S06Var05 <-  
 mean(sqdiff\_across\_gaps\_S06Var05$S06Var05.diff)

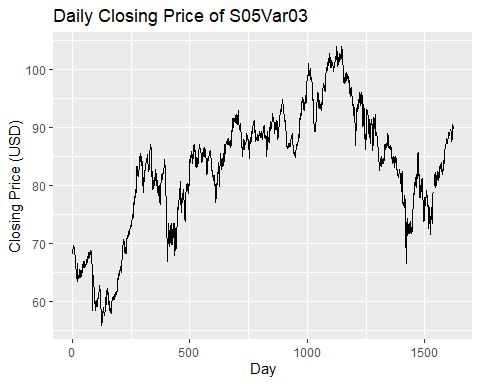
Computing the average squared difference between successive entries for S06Var05:

sqdiff\_across\_all <- gaps.df %>%  
 filter(abs(S06Var05.diff) < 50) %>%  
 select(S06Var05.diff)  
sqdiff\_across\_all <- sqdiff\_across\_all^2  
sqdiff\_across\_all <- mean(sqdiff\_across\_all$S06Var05.diff)

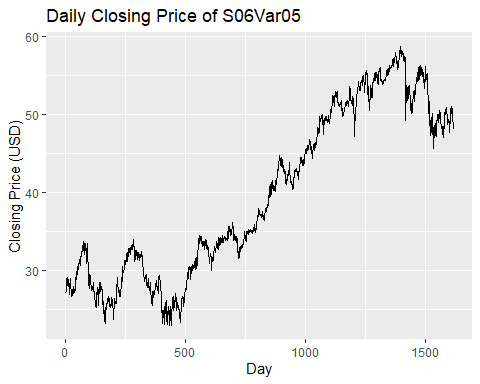
The mean square difference between values across gaps is 0.376, and the mean square difference between all successive values is 0.326. These values are close enough to suggest that missing values in the SeriesInd column represent a pause in the data-generating process, rather than missing data. As a result, we can treat this data as if all the measurements are consecutive, with no missing values.

While it’s not known what process generated these data, the levels and behavior of the data are similar to those of stock prices. For the purpose of this report, I’ll regard each value as a closing stock price of a different company, and I’ll regard the time seriesSeriesInd` values as counting days. Restarting the time series at Day = 0 and dropping outliers from the group S06 data gives us the following:

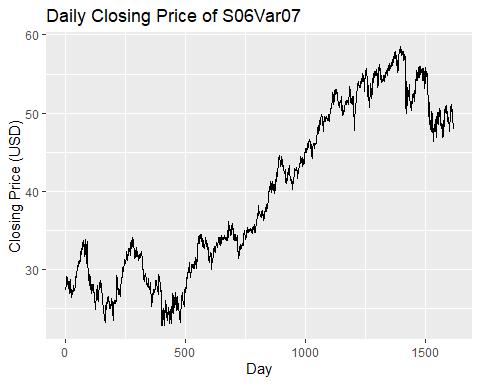
S05Var03.ts <- ts(raw$S05Var03)  
S06Var05.ts <- raw %>%  
 filter(S06Var05 < 100) %>%  
 select(S06Var05) %>%  
 ts()  
S06Var07.ts <- raw %>%  
 filter(S06Var07 < 100) %>%  
 select(S06Var07) %>%  
 ts()  
autoplot(S05Var03.ts) +  
 xlab("Day") +  
 ylab("Closing Price (USD)") +  
 ggtitle("Daily Closing Price of S05Var03")



autoplot(S06Var05.ts) +  
 xlab("Day") +  
 ylab("Closing Price (USD)") +  
 ggtitle("Daily Closing Price of S06Var05")

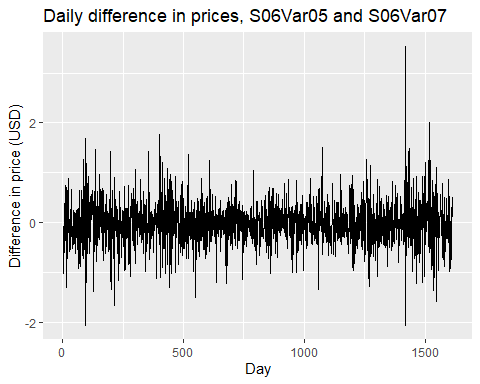


autoplot(S06Var07.ts) +  
 xlab("Day") +  
 ylab("Closing Price (USD)") +  
 ggtitle("Daily Closing Price of S06Var07")



How similar are the time series representing S06Var05 and S06Var07?

autoplot(S06Var05.ts-S06Var07.ts) +  
 xlab("Day") +  
 ylab("Difference in price (USD)") +  
 ggtitle("Daily difference in prices, S06Var05 and S06Var07")

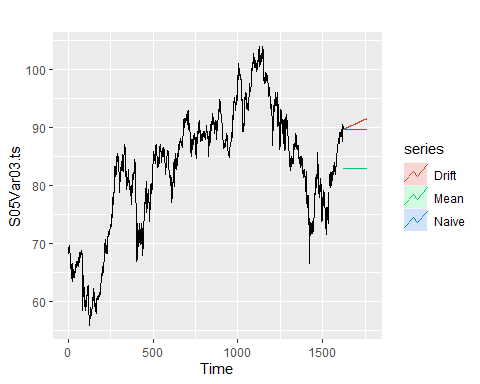


Differences in price are white noise centered at zero. The range of these differences is small compared to the level of each variable. For this reason, I’ll restrict the analysis to only S06Var05, and then apply the best model for S06Var05 to S06Var07 as well.

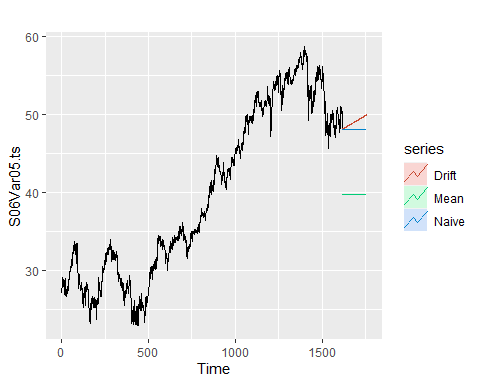
Simple forecasts

Because “a naive forecast is optimal when data follow a random walk” (HA 3.1), I compute some simple forecasts for each variable before performing more complex analysis. The performance of these simple models will provide a benchmark for performance of more sophisticated models. In the event of a tie, I’ll favor these simpler models.

S05Var03\_rwf <- rwf(S05Var03.ts, h = 140)  
S06Var05\_rwf <- rwf(S06Var05.ts, h = 140)  
S06Var07\_rwf <- rwf(S06Var07.ts, h = 140)  
S05Var03\_drwf <- rwf(S05Var03.ts, h = 140, drift = TRUE)  
S06Var05\_drwf <- rwf(S06Var05.ts, h = 140, drift = TRUE)  
S06Var07\_drwf <- rwf(S06Var07.ts, h = 140, drift = TRUE)  
S05Var03\_mean <- meanf(S05Var03.ts, h = 140)  
S06Var05\_mean <- meanf(S06Var05.ts, h = 140)  
S06Var07\_mean <- meanf(S06Var07.ts, h = 140)  
autoplot(S05Var03.ts) +  
 autolayer(S05Var03\_rwf, series = "Naive", PI = FALSE) +  
 autolayer(S05Var03\_drwf, series = "Drift", PI = FALSE) +  
 autolayer(S05Var03\_mean, series = "Mean", PI = FALSE)



autoplot(S06Var05.ts) +  
 autolayer(S06Var05\_rwf, series = "Naive", PI = FALSE) +  
 autolayer(S06Var05\_drwf, series = "Drift", PI = FALSE) +  
 autolayer(S06Var05\_mean, series = "Mean", PI = FALSE)



Evaluating performance of simple models using cross-validation and RMSE:

S05Var03\_rmse\_rwf\_nodrift <- tsCV(S05Var03.ts, rwf, drift = FALSE, h = 1)  
S05Var03\_rmse\_rwf\_nodrift <- sqrt(mean(S05Var03\_rmse\_rwf\_nodrift^2, na.rm = TRUE))  
S05Var03\_rmse\_rwf\_drift <- tsCV(S05Var03.ts, rwf, drift = TRUE, h = 1)  
S05Var03\_rmse\_rwf\_drift <- sqrt(mean(S05Var03\_rmse\_rwf\_drift^2, na.rm = TRUE))  
S05Var03\_rmse\_meanf <- tsCV(S05Var03.ts, meanf, h = 1)  
S05Var03\_rmse\_meanf <- sqrt(mean(S05Var03\_rmse\_meanf^2, na.rm = TRUE))  
S06Var05\_rmse\_rwf\_nodrift <- tsCV(S06Var05.ts, rwf, drift = FALSE, h = 1)  
S06Var05\_rmse\_rwf\_nodrift <- sqrt(mean(S06Var05\_rmse\_rwf\_nodrift^2, na.rm = TRUE))  
S06Var05\_rmse\_rwf\_drift <- tsCV(S06Var05.ts, rwf, drift = TRUE, h = 1)  
S06Var05\_rmse\_rwf\_drift <- sqrt(mean(S06Var05\_rmse\_rwf\_drift^2, na.rm = TRUE))  
S06Var05\_rmse\_meanf <- tsCV(S06Var05.ts, meanf, h = 1)  
S06Var05\_rmse\_meanf <- sqrt(mean(S06Var05\_rmse\_meanf^2, na.rm = TRUE))

For both S05Var03 and S06Var05, the best-performing model is the random walk forecast with no drift. For S05Var03, RMSE = 0.9037. For S06Var05, RMSE = 0.5712.

Exponential smoothing

Below, I fit a simple exponential smoothing method.

S05Var03\_ses <- ses(S05Var03.ts, h = 140)

## Warning in ets(x, "ANN", alpha = alpha, opt.crit = "mse", lambda = lambda, :  
## Missing values encountered. Using longest contiguous portion of time series

summary(S05Var03\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = S05Var03.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.9999   
##   
## Initial states:  
## l = 68.1906   
##   
## sigma: 0.9023  
##   
## AIC AICc BIC   
## 5142.281 5142.311 5156.312   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.02624863 0.901128 0.6446989 0.02598541 0.8551915 0.9987617  
## ACF1  
## Training set 0.08262822  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 795 89.02993 87.87363 90.18623 87.26152 90.79833  
## 796 89.02993 87.39475 90.66510 86.52915 91.53071  
## 797 89.02993 87.02729 91.03256 85.96716 92.09269  
## [ reached 'max' / getOption("max.print") -- omitted 137 rows ]

The optimized simple exponential smoothing method computed , making this method almost indistinguishable from the random-walk forecast. It offers a tiny improvement in performance when measured as RMSE, but this improvement is not sufficient to justify a more complex model.

S06Var05\_ses <- ses(S06Var05.ts, h = 140)  
summary(S06Var05\_ses)

##   
## Forecast method: Simple exponential smoothing  
##   
## Model Information:  
## Simple exponential smoothing   
##   
## Call:  
## ses(y = S06Var05.ts, h = 140)   
##   
## Smoothing parameters:  
## alpha = 0.8676   
##   
## Initial states:  
## l = 27.0651   
##   
## sigma: 0.5665  
##   
## AIC AICc BIC   
## 10105.98 10106.00 10122.15   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.0150733 0.5661707 0.4145355 0.02695357 1.133511 0.9978013  
## ACF1  
## Training set 0.0002050798  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1617 48.19949 47.47347 48.92552 47.08913 49.30985  
## 1618 48.19949 47.23828 49.16070 46.72945 49.66954  
## 1619 48.19949 47.05026 49.34872 46.44189 49.95709  
## [ reached 'max' / getOption("max.print") -- omitted 137 rows ]

For S06Var05, , indicating that optimal simple exponential smoothing does take some account of values earlier than lag-1. Similar to SES’s performance with S05Var03, SES offers only a very small performance gain over the random-walk forecast. This small gain is not sufficient to justify a more complex model.

Does allowing for drift and damping improve the performance of the exponential smoothing models?

S05Var03\_holt <- holt(S05Var03.ts, h = 140, damped = TRUE)

## Warning in ets(x, "AAN", alpha = alpha, beta = beta, phi = phi, damped =  
## damped, : Missing values encountered. Using longest contiguous portion of time  
## series

S06Var05\_holt <- holt(S06Var05.ts, h = 140, damped = TRUE)  
summary(S05Var03\_holt)

##   
## Forecast method: Damped Holt's method  
##   
## Model Information:  
## Damped Holt's method   
##   
## Call:  
## holt(y = S05Var03.ts, h = 140, damped = TRUE)   
##   
## Smoothing parameters:  
## alpha = 0.9999   
## beta = 4e-04   
## phi = 0.8   
##   
## Initial states:  
## l = 68.9316   
## b = 0.159   
##   
## sigma: 0.9044  
##   
## AIC AICc BIC   
## 5149.071 5149.178 5177.134   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.0244788 0.9015767 0.6455891 0.02341696 0.8565096 1.000141  
## ACF1  
## Training set 0.08159178  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 795 89.03017 87.87110 90.18925 87.25753 90.80282  
## 796 89.03037 87.39104 90.66971 86.52322 91.53753  
## 797 89.03053 87.02254 91.03853 85.95957 92.10150  
## [ reached 'max' / getOption("max.print") -- omitted 137 rows ]

summary(S06Var05\_holt)

##   
## Forecast method: Damped Holt's method  
##   
## Model Information:  
## Damped Holt's method   
##   
## Call:  
## holt(y = S06Var05.ts, h = 140, damped = TRUE)   
##   
## Smoothing parameters:  
## alpha = 0.8679   
## beta = 1e-04   
## phi = 0.9733   
##   
## Initial states:  
## l = 27.4044   
## b = 0.0605   
##   
## sigma: 0.5671  
##   
## AIC AICc BIC   
## 10112.03 10112.09 10144.36   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01319427 0.5661796 0.4146892 0.02042205 1.134186 0.9981713  
## ACF1  
## Training set -0.0004246243  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1617 48.19914 47.47242 48.92585 47.08772 49.31055  
## 1618 48.19898 47.23669 49.16127 46.72728 49.67067  
## 1619 48.19883 47.04818 49.34947 46.43907 49.95858  
## [ reached 'max' / getOption("max.print") -- omitted 137 rows ]

For the S05Var03 series, an exponential smoothing model with damping and drift performs marginally better than the random walk forecast with no drift. For this data, the optimal choice for is almost 1, indicating that forecast values are almost entirely dependent only on their lag-1. . This is a low value that rapidly damps forecasts. Together, the high value of and the low value of suggest that exponential smoothing with damping and drift don’t offer much additional insight above the simple random walk forecast.

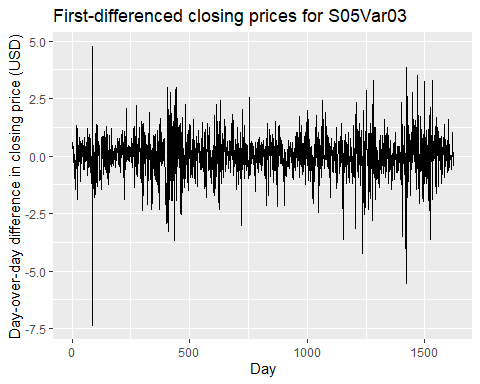
The parameters for the optimal model for the S06Var05 data are somewhat different, with and . This suggests that values earlier than lag-1 carry some importance in generating forecasts, and that damping at a more gradual pace is optimal. Still, RMSE improves only marginally with this more complex model. Simple random walk forecasts with no drift are still the top contenders for all three variables under examination.

ARIMA models

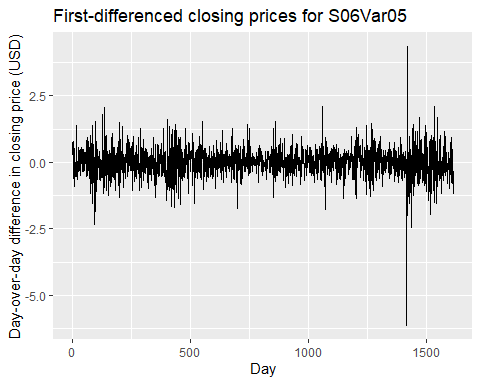
ARIMA models are generally restricted to stationary data. The most common technique for producing stationary data from a trended dataset such as this one is to perform first-differencing. We then perform a Box-Ljung test on the differenced data to determine whether the result is stationary.

Differenced data:

autoplot(diff(S05Var03.ts)) +  
 xlab("Day") +  
 ylab("Day-over-day difference in closing price (USD)") +  
 ggtitle("First-differenced closing prices for S05Var03")

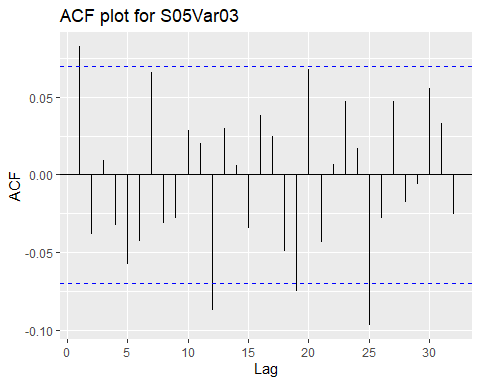
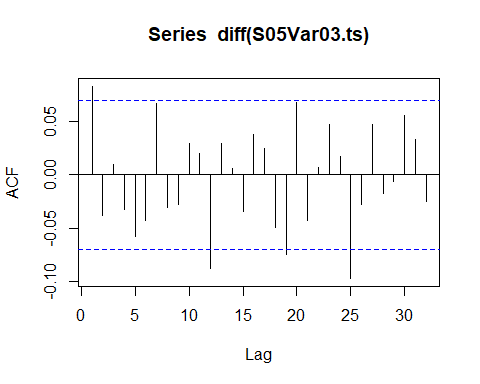


autoplot(diff(S06Var05.ts)) +  
 xlab("Day") +  
 ylab("Day-over-day difference in closing price (USD)") +  
 ggtitle("First-differenced closing prices for S06Var05")

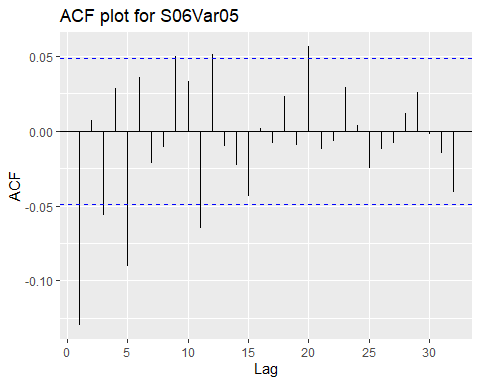
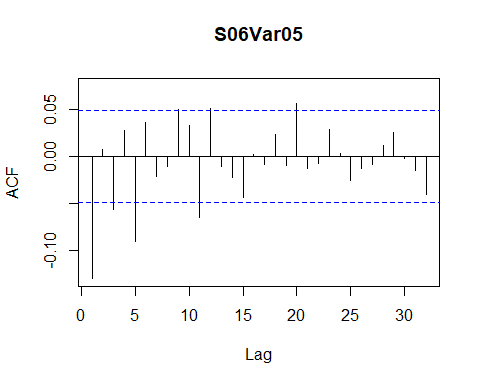


For both variables, this data does appear to be stationary– it has mean near zero, and its variability does not appear to change over time. The values appear to be random. What do the ACF plots show?

autoplot(Acf(diff(S05Var03.ts))) +  
 ggtitle("ACF plot for S05Var03")

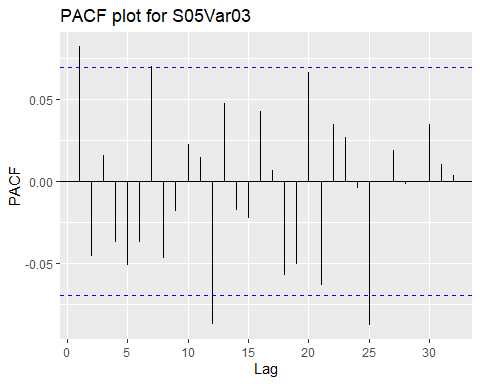
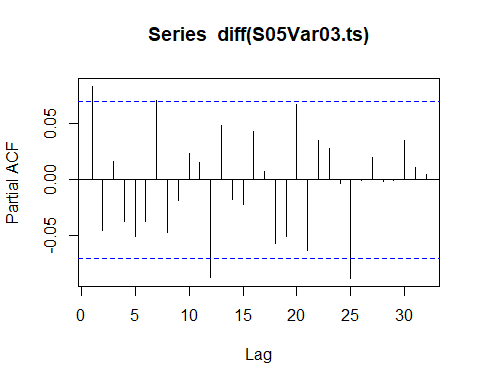


autoplot(Acf(diff(S06Var05.ts))) +  
 ggtitle("ACF plot for S06Var05")

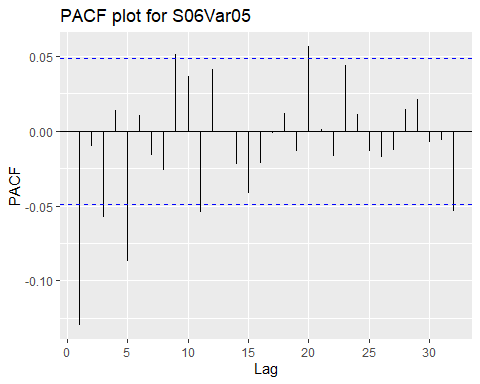
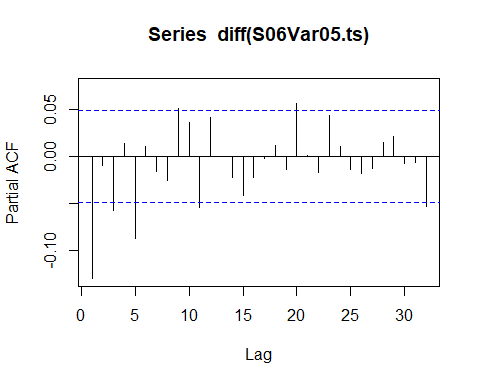


The ACF plots shows many significant lags, which suggest the differenced data may not be stationary.

autoplot(Pacf(diff(S05Var03.ts))) +  
 ggtitle("PACF plot for S05Var03")



autoplot(Pacf(diff(S06Var05.ts))) +  
 ggtitle("PACF plot for S06Var05")



The PACF plots also show several significant values, further casting doubt on the stationarity of this data. A portmanteau test confirms our suspicion:

Box.test(diff(S05Var03.ts, differences = 2), type = 'Ljung-Box')

##   
## Box-Ljung test  
##   
## data: diff(S05Var03.ts, differences = 2)  
## X-squared = 280.31, df = 1, p-value < 2.2e-16

Box.test(diff(S06Var05.ts, differences = 2), type = 'Ljung-Box')

##   
## Box-Ljung test  
##   
## data: diff(S06Var05.ts, differences = 2)  
## X-squared = 508.83, df = 1, p-value < 2.2e-16

The Box-Ljung test confirms that the data should not be considered stationary. Even after two rounds of differencing, the null hypothesis that each observation is independent of its lag is rejected, with a p-value near 0.

Let’s fit an ARIMA model anyway. Since the visual appearance of stationarity is so striking, it might be that the data is close enough to satisfying the assumptions that the model still proves useful.

S05Var03\_arima <- auto.arima(S05Var03.ts)  
S06Var05\_arima <- auto.arima(S06Var05.ts)  
summary(S05Var03\_arima)

## Series: S05Var03.ts   
## ARIMA(0,1,2)   
##   
## Coefficients:  
## ma1 ma2  
## 0.1148 -0.0259  
## s.e. 0.0251 0.0257  
##   
## sigma^2 estimated as 0.8048: log likelihood=-2118.21  
## AIC=4242.41 AICc=4242.43 BIC=4258.59  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01181137 0.8962794 0.6460439 0.009077146 0.7999247 0.9910684  
## ACF1  
## Training set 0.002297292

summary(S06Var05\_arima)

## Series: S06Var05.ts   
## ARIMA(2,1,2)   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2  
## -0.2834 0.5443 0.1567 -0.5871  
## s.e. 0.1655 0.1214 0.1603 0.1067  
##   
## sigma^2 estimated as 0.3189: log likelihood=-1366.62  
## AIC=2743.24 AICc=2743.27 BIC=2770.17  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01721161 0.5637965 0.4142626 0.03161162 1.133366 0.9971444  
## ACF1  
## Training set -0.002473942

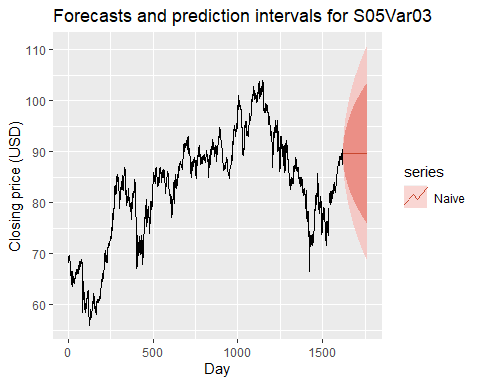
For S05Var03, RMSE for the ARIMA(0,1,2) model is 0.8963. For S06Var05, RMSE for ARIMA(2,1,2) is 0.5637. As with the damped and trended exponential smoothing methods, these models give marginal improvements in RMSE. However, these small improvements are not sufficient to justify their use as forecasting models, since they’re complicated relative to simple forecasting methods.

Had ARIMA models provided a greater reduction in RMSE, then we would proceed from here to an analysis of residuals.

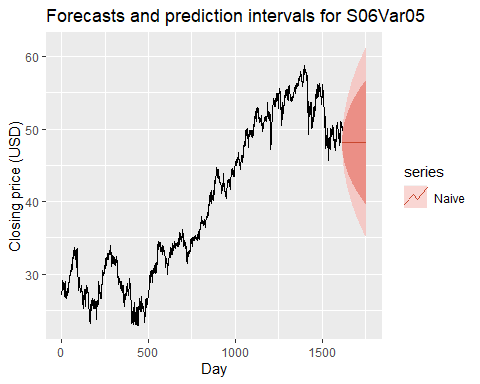
Conclusion

For the variables S05Var03, S06Var05, and S06Var07, the best forecasting model is a random walk forecast without drift. The forecast value for all future times is the most recent value of the time series. As time extends into the future, prediction intervals for these forecasts become wider:

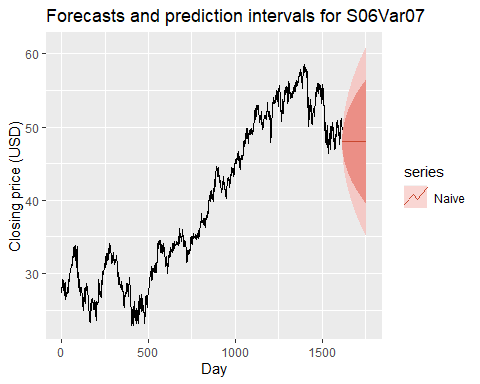
autoplot(S05Var03.ts) +  
 autolayer(S05Var03\_rwf, series = "Naive", PI = TRUE) +  
 xlab("Day") +  
 ylab("Closing price (USD)") +  
 ggtitle("Forecasts and prediction intervals for S05Var03")



autoplot(S06Var05.ts) +  
 autolayer(S06Var05\_rwf, series = "Naive", PI = TRUE) +  
 xlab("Day") +  
 ylab("Closing price (USD)") +  
 ggtitle("Forecasts and prediction intervals for S06Var05")



autoplot(S06Var07.ts) +  
 autolayer(S06Var07\_rwf, series = "Naive", PI = TRUE) +  
 xlab("Day") +  
 ylab("Closing price (USD)") +  
 ggtitle("Forecasts and prediction intervals for S06Var07")



This is consistent with reasonable expectations for stock market price data, which are notoriously resistant to time series forecasting methods. If we had more information about this data– such as information about the process that generated it, or historical data further into the past– it might have been possible to take better advantage of methods incorporating trend. Over the relatively short horizon of this data set, though, long-run trends that appear in stock prices were not wholly evident.

For all future times, the forecast values for each variable are given below.

S05Var03\_rwf\_fc <- forecast(S05Var03\_rwf, h = 140)  
print(S05Var03\_rwf\_fc)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1623 89.7 88.54181 90.85818 87.92870 91.47129  
## 1624 89.7 88.06207 91.33792 87.19501 92.20498  
## 1625 89.7 87.69396 91.70603 86.63203 92.76797  
## [ reached 'max' / getOption("max.print") -- omitted 137 rows ]

S06Var05\_rwf\_fc <- forecast(S06Var05\_rwf, h = 140)  
print(S06Var05\_rwf\_fc)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1617 48.13 47.39796 48.86204 47.01045 49.24956  
## 1618 48.13 47.09474 49.16526 46.54671 49.71329  
## 1619 48.13 46.86207 49.39793 46.19087 50.06913  
## [ reached 'max' / getOption("max.print") -- omitted 137 rows ]

S06Var07\_rwf\_fc <- forecast(S06Var07\_rwf, h = 140)  
print(S06Var07\_rwf\_fc)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1617 47.97 47.24943 48.69057 46.86799 49.07202  
## 1618 47.97 46.95096 48.98904 46.41152 49.52849  
## 1619 47.97 46.72194 49.21806 46.06125 49.87875  
## [ reached 'max' / getOption("max.print") -- omitted 137 rows ]