Grp3\_S03

Project 1

Samriti Malhotra

June 25, 2020

Table of Contents

## Series 3

### Import the necessary libraries

### Load the dataset

data\_project <- readxl::read\_excel("./project1data/Data Set for class.xls")  
head(data\_project)

## # A tibble: 6 x 7  
## SeriesInd group Var01 Var02 Var03 Var05 Var07  
## <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 40669 S03 30.6 123432400 30.3 30.5 30.6  
## 2 40669 S02 10.3 60855800 10.0 10.2 10.3  
## 3 40669 S01 26.6 10369300 25.9 26.2 26.0  
## 4 40669 S06 27.5 39335700 26.8 27.0 27.3  
## 5 40669 S05 69.3 27809100 68.2 68.7 69.2  
## 6 40669 S04 17.2 16587400 16.9 16.9 17.1

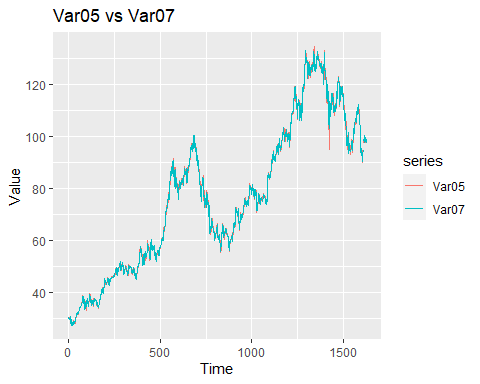
### Series 3 Subset the dataset

S03 <- subset(data\_project, group == 'S03', select = c(SeriesInd, Var05, Var07))  
head(S03)

## # A tibble: 6 x 3  
## SeriesInd Var05 Var07  
## <dbl> <dbl> <dbl>  
## 1 40669 30.5 30.6  
## 2 40670 30.7 30.6  
## 3 40671 30.6 30.1  
## 4 40672 30.2 30.1  
## 5 40673 30.0 30.3  
## 6 40676 30.4 30.0

### Visualization of variables

predictobs <- 1623:1762  
S3 <- ts(S03[-predictobs, 2:3])  
  
autoplot(S3) + ylab('Value') + xlab('Time') + ggtitle('Var05 vs Var07')

 Both the series in S03 look alike, Most probably the forecasts for both series will be almost identical.The data show strong trend but no evident seasonal pattern, which will be bases of our modelling.

### Data Cleaning and Exploration

#### Get subsets of Var05 and Var07

var05 <- S03 %>% filter(SeriesInd <= 43021) %>% select(Var05)  
var07 <- S03 %>% filter(SeriesInd <= 43021) %>% select(Var07)

#### Explore Var05

summary(var05)

## Var05   
## Min. : 27.48   
## 1st Qu.: 53.30   
## Median : 75.59   
## Mean : 76.90   
## 3rd Qu.: 98.55   
## Max. :134.46   
## NA's :4

Var05 has 4 missing value

#### Explore Var07

summary(var07)

## Var07   
## Min. : 27.44   
## 1st Qu.: 53.46   
## Median : 75.71   
## Mean : 76.87   
## 3rd Qu.: 98.61   
## Max. :133.00   
## NA's :4

Var07 has 4 missing values

#### Impute Missing values.

var05 <- na\_interpolation(var05)  
summary(var05)

## Var05   
## Min. : 27.48   
## 1st Qu.: 53.34   
## Median : 75.66   
## Mean : 76.95   
## 3rd Qu.: 98.53   
## Max. :134.46

var07 <- na\_interpolation(var07)  
summary(var07)

## Var07   
## Min. : 27.44   
## 1st Qu.: 53.53   
## Median : 75.76   
## Mean : 76.91   
## 3rd Qu.: 98.51   
## Max. :133.00

var05 <- ts(var05)  
str(var05)

## Time-Series [1:1622, 1] from 1 to 1622: 30.5 30.7 30.6 30.2 30 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr "Var05"

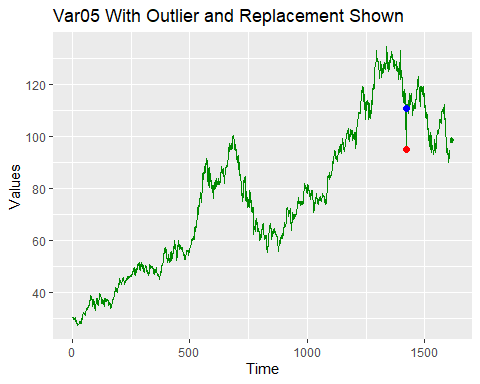
var07 <- ts(var07)  
str(var07)

## Time-Series [1:1622, 1] from 1 to 1622: 30.6 30.6 30.1 30.1 30.3 ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr "Var07"

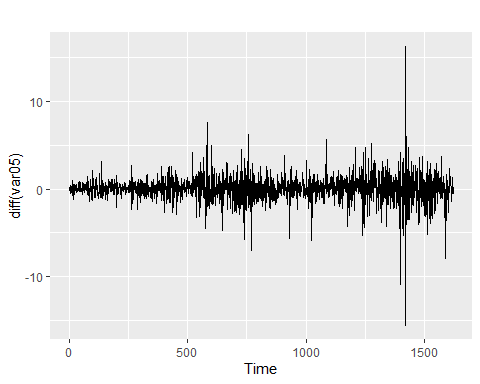
#### Outliers

The series show only one outlier between them which is shown in the above timesries plot. As, series are so similar and this is one of the defining differences of the series, We are not removing outliers.

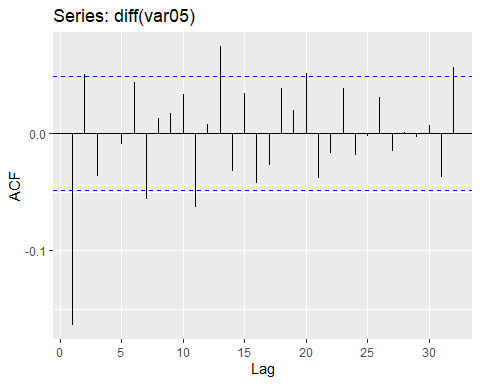
s31out <- tsoutliers(var05)  
s32out <- tsoutliers(var07)  
  
data.frame(S3) %>% ggplot() +  
 geom\_line(aes(x = 1:length(var05), y = Var05), color = 'green4') +  
 geom\_point(data = data.frame(s31out), aes(x = index, y = replacements),  
 color = 'blue', size = 2) +  
 geom\_point(aes(x = s31out$index, y = Var05[s31out$index]),  
 color = 'red', size = 2) +  
 xlab('Time') + ylab('Values') +  
 ggtitle('Var05 With Outlier and Replacement Shown')

 #### ACF for VAr5

par(mfrow=c(1,2))  
autoplot(diff(var05))

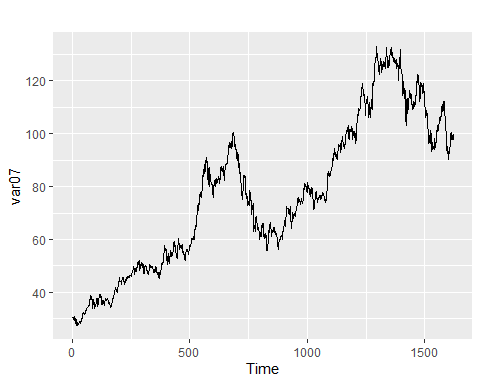


ggAcf(diff(var05))

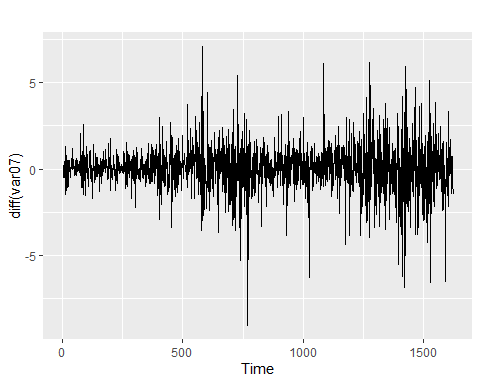


### ACF for VAr7

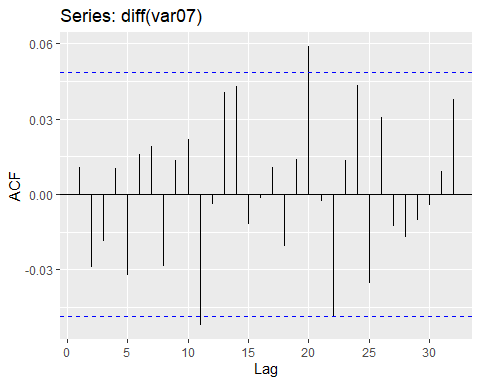
par(mfrow=c(1,2))  
autoplot(var07)



autoplot(diff(var07))



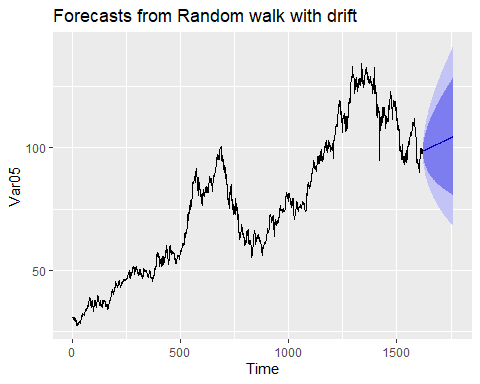
ggAcf(diff(var07))



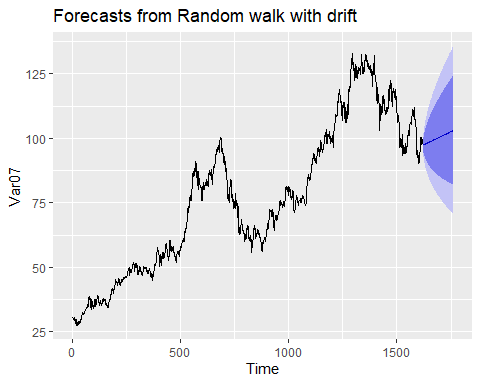
### Random Walk with Drift

Since the data has a clear trend component but no seasonality component, applying simple random walk with drift for baseline model.

s03v5 <- var05  
s03v7 <- var07  
  
rwf(s03v5, h = 140, drift = T) %>% autoplot() + ylab('Var05')

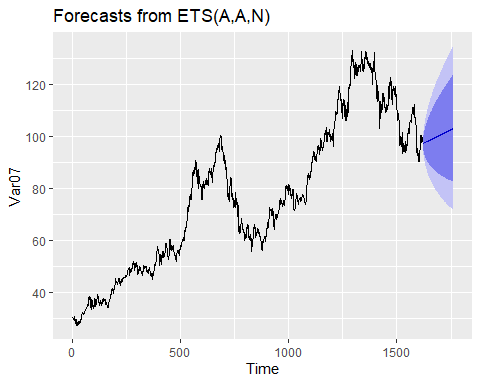
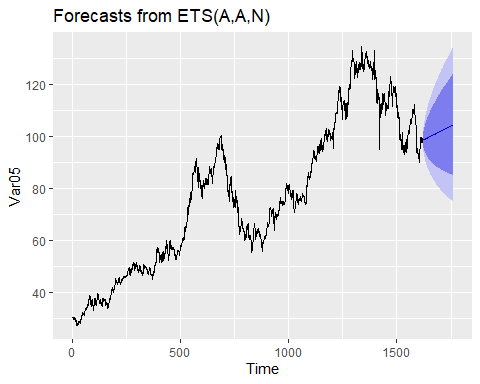


rwf(s03v7, h = 140, drift = T) %>% autoplot() + ylab('Var07')



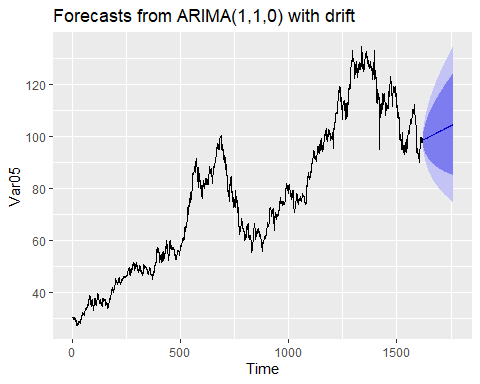
### Exponential Smoothing model

Applying linear trend model for the data.

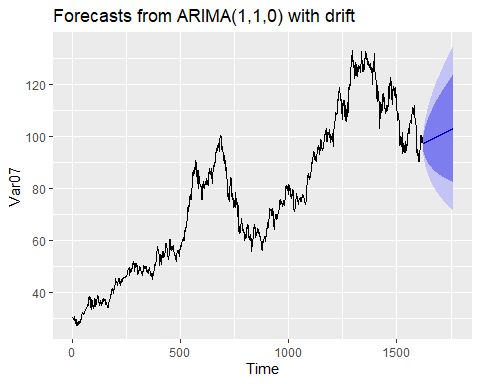


### ARIMA Model

Applying ARIMA model: ARIMA(1,1,0) choice of model parameters is confirmed by data visualization and by the auto arima function.



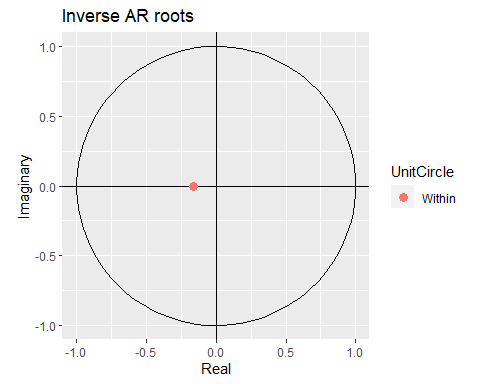
## Series: s03v5   
## ARIMA(1,1,0) with drift   
##   
## Coefficients:  
## ar1 drift  
## -0.1640 0.0421  
## s.e. 0.0245 0.0320  
##   
## sigma^2 estimated as 2.245: log likelihood=-2954.42  
## AIC=5914.84 AICc=5914.86 BIC=5931.02  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 2.6104e-05 1.496811 1.001878 -0.009345759 1.322568 0.9899828  
## ACF1  
## Training set 0.003934698



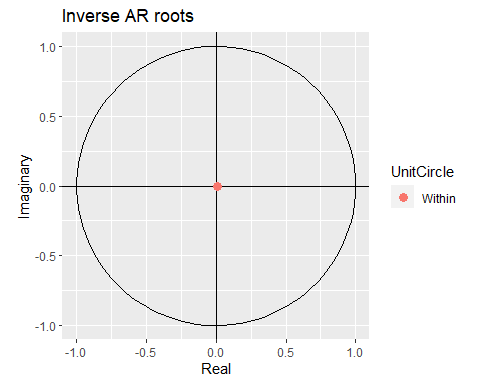
## Series: s03v7   
## ARIMA(1,1,0) with drift   
##   
## Coefficients:  
## ar1 drift  
## 0.0109 0.0412  
## s.e. 0.0248 0.0337  
##   
## sigma^2 estimated as 1.805: log likelihood=-2777.76  
## AIC=5561.52 AICc=5561.53 BIC=5577.69  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE  
## Training set 1.074076e-05 1.342269 0.9292027 -0.005290655 1.223764  
## MASE ACF1  
## Training set 0.9990103 0.0003231799

### Forecast

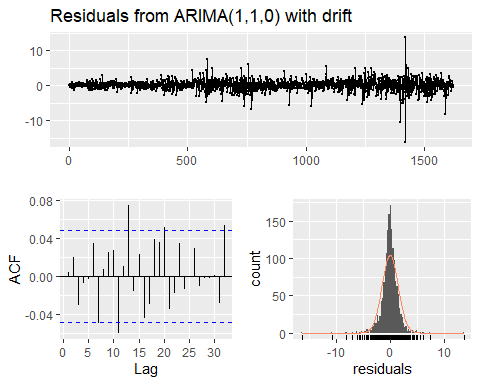
autoplot(arima\_s3v5)



autoplot(arima\_s3v7)

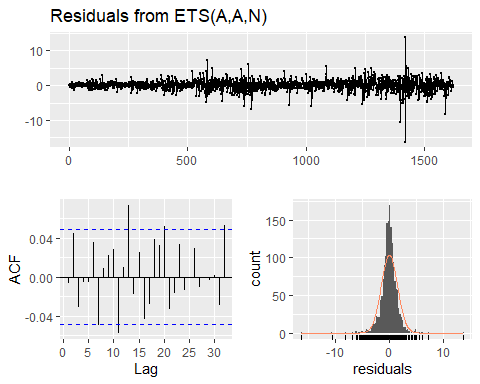
 ### Check the residuals

checkresiduals(arima\_s3v5)



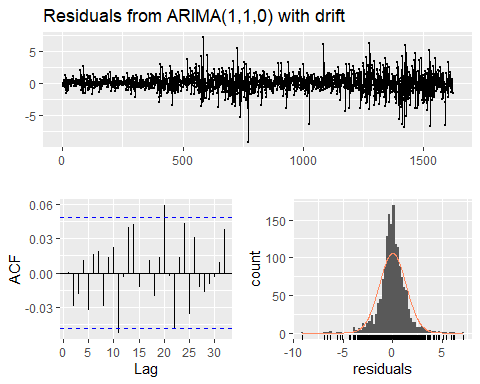
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,0) with drift  
## Q\* = 10.657, df = 8, p-value = 0.2219  
##   
## Model df: 2. Total lags used: 10

checkresiduals(fit)



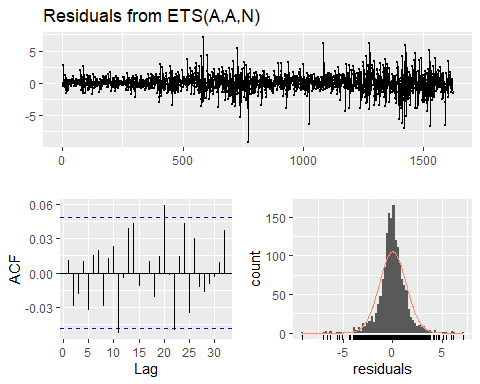
##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,A,N)  
## Q\* = 13.339, df = 6, p-value = 0.03795  
##   
## Model df: 4. Total lags used: 10

checkresiduals(arima\_s3v7)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,0) with drift  
## Q\* = 7.3382, df = 8, p-value = 0.5006  
##   
## Model df: 2. Total lags used: 10

checkresiduals(fit2)



##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,A,N)  
## Q\* = 7.5532, df = 6, p-value = 0.2727  
##   
## Model df: 4. Total lags used: 10

Both sets of residuals appear to resemble a normal distribution with some strong outliers. Exponential smoothing model does not pass the Ljung-Box test with a p-value of 0.03795, which indicates that residuals from the exponential smoothing model may be correlated and model can be improved. ARIMA model further confirms our decision to use the ARIMA model for this series.

### MAPE Calculation:

print(paste0("MAPE for S03 Var05 is ", MLmetrics::MAPE(arima\_s3v5$fitted,s03v5)))

## [1] "MAPE for S03 Var05 is 0.0132256809974202"

print(paste0("MAPE for S03 Var07 is ", MLmetrics::MAPE(arima\_s3v7$fitted,s03v7)))

## [1] "MAPE for S03 Var07 is 0.0122376449235163"

### Writing forcast of V07 to csv

fc <- forecast(s03v7,h=140)  
fc$mean<-fc$mean  
fc$upper<-fc$upper  
fc$lower<-fc$lower  
fc$x<-fc$x  
  
fc

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1623 97.34015 95.23692 99.44337 94.12354 100.5568  
## 1624 97.34015 94.36568 100.31462 92.79109 101.8892  
## 1625 97.34015 93.69698 100.98331 91.76841 102.9119  
## 1626 97.34015 93.13312 101.54717 90.90606 103.7742  
## 1627 97.34015 92.63624 102.04406 90.14614 104.5342  
## 1628 97.34015 92.18693 102.49337 89.45897 105.2213  
## 1629 97.34015 91.77365 102.90665 88.82692 105.8534  
## 1630 97.34015 91.38890 103.29140 88.23850 106.4418  
## 1631 97.34015 91.02746 103.65284 87.68572 106.9946  
## 1632 97.34015 90.68552 103.99478 87.16277 107.5175  
## 1633 97.34015 90.36023 104.32007 86.66528 108.0150  
## 1634 97.34015 90.04934 104.63095 86.18983 108.4905  
## 1635 97.34015 89.75110 104.92919 85.73371 108.9466  
## 1636 97.34015 89.46407 105.21623 85.29473 109.3856  
## 1637 97.34015 89.18706 105.49324 84.87107 109.8092  
## 1638 97.34015 88.91908 105.76122 84.46123 110.2191  
## 1639 97.34015 88.65929 106.02100 84.06393 110.6164  
## 1640 97.34015 88.40699 106.27331 83.67806 111.0022  
## 1641 97.34015 88.16155 106.51875 83.30269 111.3776  
## 1642 97.34015 87.92244 106.75786 82.93700 111.7433  
## 1643 97.34015 87.68918 106.99111 82.58027 112.1000  
## 1644 97.34015 87.46137 107.21893 82.23186 112.4484  
## 1645 97.34015 87.23863 107.44166 81.89121 112.7891  
## 1646 97.34015 87.02064 107.65966 81.55782 113.1225  
## 1647 97.34015 86.80709 107.87320 81.23123 113.4491  
## 1648 97.34015 86.59774 108.08256 80.91105 113.7692  
## 1649 97.34015 86.39232 108.28797 80.59690 114.0834  
## 1650 97.34015 86.19064 108.48966 80.28844 114.3919  
## 1651 97.34015 85.99248 108.68782 79.98539 114.6949  
## 1652 97.34015 85.79767 108.88263 79.68745 114.9928  
## 1653 97.34015 85.60604 109.07426 79.39438 115.2859  
## 1654 97.34015 85.41743 109.26286 79.10593 115.5744  
## 1655 97.34015 85.23171 109.44858 78.82190 115.8584  
## 1656 97.34015 85.04875 109.63155 78.54208 116.1382  
## 1657 97.34015 84.86842 109.81188 78.26628 116.4140  
## 1658 97.34015 84.69060 109.98969 77.99434 116.6860  
## 1659 97.34015 84.51521 110.16509 77.72610 116.9542  
## 1660 97.34015 84.34213 110.33817 77.46139 117.2189  
## 1661 97.34015 84.17128 110.50902 77.20010 117.4802  
## 1662 97.34015 84.00257 110.67773 76.94208 117.7382  
## 1663 97.34015 83.83592 110.84438 76.68721 117.9931  
## 1664 97.34015 83.67125 111.00905 76.43537 118.2449  
## 1665 97.34015 83.50850 111.17180 76.18646 118.4938  
## 1666 97.34015 83.34759 111.33270 75.94038 118.7399  
## 1667 97.34015 83.18847 111.49182 75.69703 118.9833  
## 1668 97.34015 83.03108 111.64922 75.45632 119.2240  
## 1669 97.34015 82.87535 111.80494 75.21815 119.4621  
## 1670 97.34015 82.72124 111.95905 74.98246 119.6978  
## 1671 97.34015 82.56870 112.11160 74.74916 119.9311  
## 1672 97.34015 82.41767 112.26263 74.51818 120.1621  
## 1673 97.34015 82.26811 112.41219 74.28945 120.3908  
## 1674 97.34015 82.11998 112.56032 74.06290 120.6174  
## 1675 97.34015 81.97323 112.70706 73.83848 120.8418  
## 1676 97.34015 81.82783 112.85246 73.61611 121.0642  
## 1677 97.34015 81.68375 112.99655 73.39575 121.2845  
## 1678 97.34015 81.54093 113.13937 73.17733 121.5030  
## 1679 97.34015 81.39936 113.28094 72.96081 121.7195  
## 1680 97.34015 81.25899 113.42131 72.74614 121.9342  
## 1681 97.34015 81.11979 113.56050 72.53326 122.1470  
## 1682 97.34015 80.98175 113.69855 72.32213 122.3582  
## 1683 97.34015 80.84481 113.83548 72.11271 122.5676  
## 1684 97.34015 80.70897 113.97132 71.90496 122.7753  
## 1685 97.34015 80.57419 114.10610 71.69883 122.9815  
## 1686 97.34015 80.44045 114.23985 71.49429 123.1860  
## 1687 97.34015 80.30772 114.37258 71.29130 123.3890  
## 1688 97.34015 80.17598 114.50432 71.08981 123.5905  
## 1689 97.34015 80.04520 114.63509 70.88981 123.7905  
## 1690 97.34015 79.91537 114.76492 70.69125 123.9890  
## 1691 97.34015 79.78647 114.89383 70.49411 124.1862  
## 1692 97.34015 79.65846 115.02183 70.29835 124.3820  
## 1693 97.34015 79.53134 115.14895 70.10393 124.5764  
## 1694 97.34015 79.40509 115.27521 69.91085 124.7695  
## 1695 97.34015 79.27968 115.40061 69.71905 124.9612  
## 1696 97.34015 79.15511 115.52519 69.52853 125.1518  
## 1697 97.34015 79.03134 115.64895 69.33925 125.3410  
## 1698 97.34015 78.90837 115.77192 69.15118 125.5291  
## 1699 97.34015 78.78619 115.89411 68.96431 125.7160  
## 1700 97.34015 78.66476 116.01553 68.77861 125.9017  
## 1701 97.34015 78.54409 116.13620 68.59406 126.0862  
## 1702 97.34015 78.42415 116.25614 68.41063 126.2697  
## 1703 97.34015 78.30494 116.37536 68.22831 126.4520  
## 1704 97.34015 78.18643 116.49386 68.04707 126.6332  
## 1705 97.34015 78.06862 116.61167 67.86689 126.8134  
## 1706 97.34015 77.95149 116.72880 67.68776 126.9925  
## 1707 97.34015 77.83504 116.84526 67.50965 127.1706  
## 1708 97.34015 77.71924 116.96106 67.33255 127.3477  
## 1709 97.34015 77.60408 117.07621 67.15644 127.5239  
## 1710 97.34015 77.48957 117.19073 66.98130 127.6990  
## 1711 97.34015 77.37567 117.30462 66.80712 127.8732  
## 1712 97.34015 77.26239 117.41790 66.63387 128.0464  
## 1713 97.34015 77.14972 117.53058 66.46155 128.2187  
## 1714 97.34015 77.03764 117.64266 66.29014 128.3902  
## 1715 97.34015 76.92614 117.75416 66.11962 128.5607  
## 1716 97.34015 76.81522 117.86508 65.94997 128.7303  
## 1717 97.34015 76.70486 117.97544 65.78120 128.8991  
## 1718 97.34015 76.59506 118.08524 65.61327 129.0670  
## 1719 97.34015 76.48580 118.19450 65.44617 129.2341  
## 1720 97.34015 76.37708 118.30321 65.27991 129.4004  
## 1721 97.34015 76.26890 118.41140 65.11445 129.5658  
## 1722 97.34015 76.16123 118.51906 64.94979 129.7305  
## 1723 97.34015 76.05408 118.62621 64.78592 129.8944  
## 1724 97.34015 75.94744 118.73286 64.62282 130.0575  
## 1725 97.34015 75.84129 118.83900 64.46049 130.2198  
## 1726 97.34015 75.73564 118.94466 64.29890 130.3814  
## 1727 97.34015 75.63047 119.04983 64.13806 130.5422  
## 1728 97.34015 75.52578 119.15452 63.97795 130.7023  
## 1729 97.34015 75.42156 119.25874 63.81856 130.8617  
## 1730 97.34015 75.31780 119.36250 63.65987 131.0204  
## 1731 97.34015 75.21450 119.46580 63.50189 131.1784  
## 1732 97.34015 75.11165 119.56864 63.34460 131.3357  
## 1733 97.34015 75.00925 119.67105 63.18798 131.4923  
## 1734 97.34015 74.90728 119.77301 63.03204 131.6483  
## 1735 97.34015 74.80575 119.87455 62.87676 131.8035  
## 1736 97.34015 74.70464 119.97565 62.72213 131.9582  
## 1737 97.34015 74.60396 120.07634 62.56814 132.1122  
## 1738 97.34015 74.50369 120.17661 62.41479 132.2655  
## 1739 97.34015 74.40383 120.27647 62.26207 132.4182  
## 1740 97.34015 74.30437 120.37593 62.10997 132.5703  
## 1741 97.34015 74.20531 120.47498 61.95847 132.7218  
## 1742 97.34015 74.10665 120.57364 61.80758 132.8727  
## 1743 97.34015 74.00838 120.67191 61.65729 133.0230  
## 1744 97.34015 73.91049 120.76980 61.50758 133.1727  
## 1745 97.34015 73.81299 120.86731 61.35846 133.3218  
## 1746 97.34015 73.71585 120.96444 61.20991 133.4704  
## 1747 97.34015 73.61909 121.06121 61.06192 133.6184  
## 1748 97.34015 73.52269 121.15760 60.91449 133.7658  
## 1749 97.34015 73.42666 121.25364 60.76762 133.9127  
## 1750 97.34015 73.33098 121.34932 60.62129 134.0590  
## 1751 97.34015 73.23565 121.44464 60.47550 134.2048  
## 1752 97.34015 73.14067 121.53962 60.33025 134.3500  
## 1753 97.34015 73.04604 121.63425 60.18552 134.4948  
## 1754 97.34015 72.95175 121.72855 60.04131 134.6390  
## 1755 97.34015 72.85779 121.82250 59.89762 134.7827  
## 1756 97.34015 72.76417 121.91613 59.75443 134.9259  
## 1757 97.34015 72.67088 122.00942 59.61175 135.0685  
## 1758 97.34015 72.57791 122.10239 59.46957 135.2107  
## 1759 97.34015 72.48526 122.19504 59.32787 135.3524  
## 1760 97.34015 72.39293 122.28737 59.18667 135.4936  
## 1761 97.34015 72.30091 122.37938 59.04594 135.6344  
## 1762 97.34015 72.20921 122.47109 58.90569 135.7746

write.csv(fc,"s03v07.csv")