Qian Feng Final

Qian Feng

4/18/2021

Objective

1.Analysis the relationship among fatality Rate, positive Rate, and recovery Rate of covid-19

2.Find out which rate will affect covid-19 daily safety level the most

3.Use external data, analyze the effects of covid-19 on the stock marketing among these AMC, Grubhub, Las Vegas Sands Corp, and Zoom.

Package

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

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

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

```
library(arules)
```

```
## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz)
library(e1071)
```

```
## Registered S3 methods overwritten by 'proxy':
##   method               from
##   print.registry_field registry
##   print.registry_entry registry
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:arules':
##
##     intersect, setdiff, union

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

```
library(grid)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.1.0     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x gridExtra::combine()     masks dplyr::combine()
## x lubridate::date()        masks base::date()
## x tidyr::expand()          masks Matrix::expand()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks arules::intersect(), base::intersect()
## x dplyr::lag()             masks stats::lag()
## x tidyr::pack()            masks Matrix::pack()
## x arules::recode()         masks dplyr::recode()
## x lubridate::setdiff()     masks arules::setdiff(), base::setdiff()
## x lubridate::union()       masks arules::union(), base::union()
## x tidyr::unpack()          masks Matrix::unpack()
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.14.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
```

```
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

Part 1:Analysis the relationship among fatality Rate, positive Rate, and recovery Rate of covid-19

Data

```
modify_uscovid <- read.csv("~/Desktop/bigdata/Qian Feng project/modify/modify_uscovid.csv")
uscovid<- na.omit(modify_uscovid)
uscovid$Date<-as.Date(uscovid$Date,"%m/%d/%y")
```

The dataset I use is covid-19 relative information from April 1, 2020 to December 6, 2020

Step1:Pre-processing

```
uscovid$fatalityRate<-(uscovid$death/uscovid$positive)*100
uscovid$positiveRate<-(uscovid$positive/uscovid$totalTestResults)*100
uscovid$recoveryRate<-(uscovid$recovered/uscovid$positive)*100
uscovid$dailylevelnum<-(uscovid$positiveIncrease/uscovid$totalTestResultsIncrease)*100
```

Fatality Rate represents:The daily case fatality rate of covid-19

Positive Rate represents:The daily case positive rate of covid-19

Recovery Rate represents:The daily case recovery rate of covid-19

dailylevel number reprents:The ratio of daily covid-19 positive cases to the total daily covid-19 cases

Relationship among fatalityRate, positiveRate, and recoveryRate

Step2:Correlation Analysis

```
cor(uscovid$fatalityRate,uscovid$positiveRate)
```

```
## [1] 0.5835276
```

```
cor(uscovid$fatalityRate,uscovid$recoveryRate)
```

```
## [1] -0.6308485
```

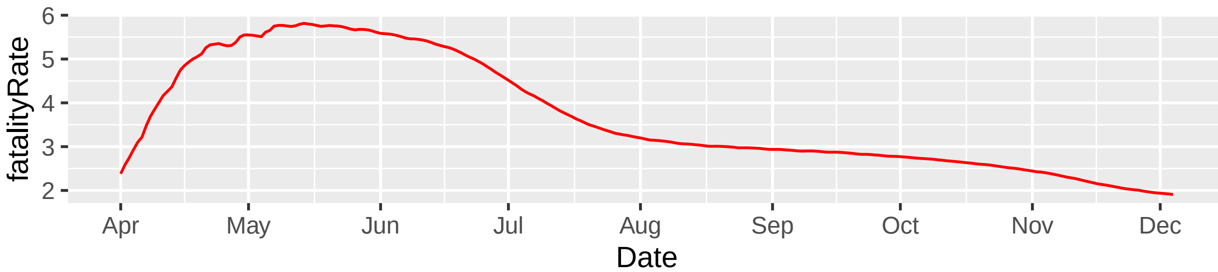
```
cor(uscovid$positiveRate,uscovid$recoveryRate)
```

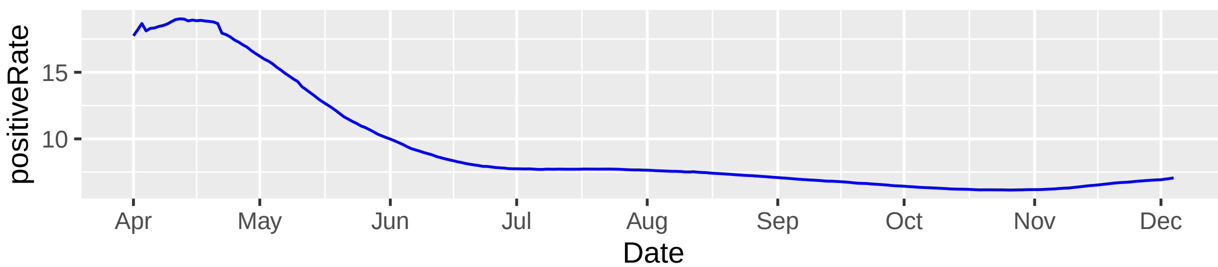
```
## [1] -0.9790366
```

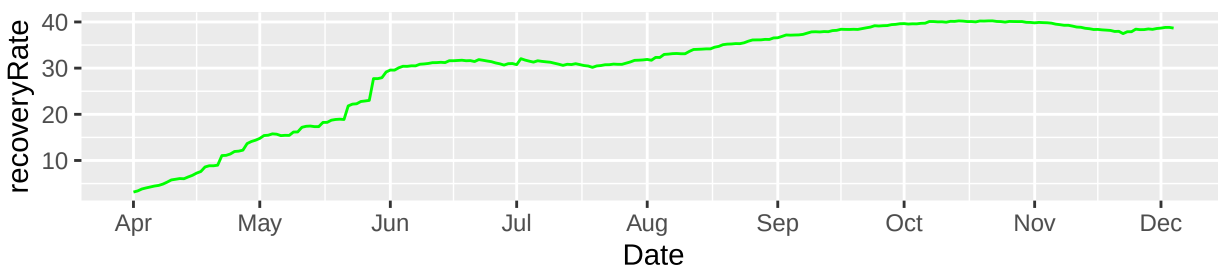
From the correlation analysis, we can see that the correlation between fatality Rate and positive Rate seems relative, but others correlation seems unreasonable.

Step3:Plot all three rates into graph

```
fatalityRate_graph<-ggplot(uscovid, aes(x=Date,y=fatalityRate))+ geom_line(colour="red")+scale_x_date(b

positiveRate_graph<-ggplot(uscovid, aes(x=Date,y=positiveRate))+ geom_line(colour="blue")+scale_x_date(

recoveryRate_graph<-ggplot(uscovid, aes(x=Date,y=recoveryRate))+ geom_line(colour="green")+scale_x_date

grid.arrange(fatalityRate_graph, positiveRate_graph,recoveryRate_graph)
```







During the period from April 1, 2020, to December 6, 2020. The fatality rate of U.S covid-19 first increases in a short period of time and then continues to decrease. The positive rate of U.S covid-19 continues to decrease. The recovery rate of U.S covid-19 continues to increase.

The positive rate and the fatality rate are proportional, but the trend of the fatality rate has a couple of days later than the positive rates trend.The recovery rate is inversely proportional to the positive rate

Part 2:Find out which rate will affect covid-19 daily safety level the most

Step1:Correlation Analysis

```
cor(uscovid$fatalityRate,uscovid$dailylevelnum)
```

```
## [1] 0.02624678
```

```
cor(uscovid$positiveRate,uscovid$dailylevelnum)
```

```
## [1] 0.7449588
```

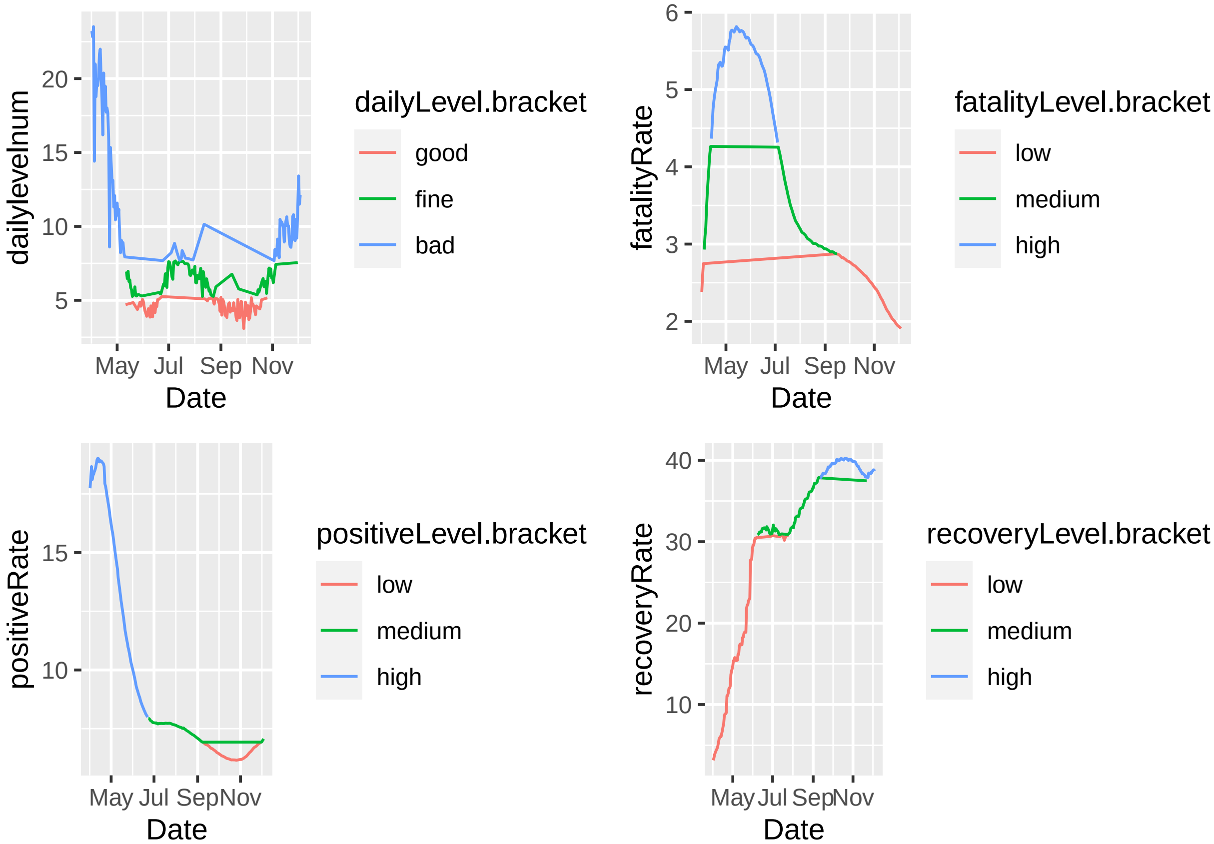
```
cor(uscovid$recoveryRate,uscovid$dailylevelnum)
```

```
## [1] -0.7180937
```

From the correlation analysis, we can see that only the correlation between positive rate and covid-19 daily safety level is relative, but others correlation seems unreasonable.

Step2:Plot all four level into graph

```
uscovid$dailyLevel.bracket<-discretize(uscovid$dailylevelnum, method = "frequency",breaks = 3,labels = c
uscovid$fatalityLevel.bracket<-discretize(uscovid$fatalityRate, method = "frequency",breaks = 3,labels =
uscovid$positiveLevel.bracket<-discretize(uscovid$positiveRate, method = "frequency",breaks = 3,labels =
uscovid$recoveryLevel.bracket<-discretize(uscovid$recoveryRate, method = "frequency",breaks = 3,labels =

dailyLeveLnum_graph<-ggplot(uscovid, aes(x=Date, y=dailylevelnum, color=dailyLevel.bracket)) + geom_line

fatalityLevel_graph<-ggplot(uscovid, aes(x=Date, y=fatalityRate, color=fatalityLevel.bracket)) + geom_li

positiveLevel_graph<-ggplot(uscovid, aes(x=Date, y=positiveRate, color=positiveLevel.bracket)) + geom_li

recoveryLevel_graph<-ggplot(uscovid, aes(x=Date, y=recoveryRate, color=recoveryLevel.bracket)) + geom_li

grid.arrange(dailyLeveLnum_graph, fatalityLevel_graph,positiveLevel_graph,recoveryLevel_graph)
```



I separate the dailylevel number into three levels "good", "fine", "bad". Indicate the daily level of covid-19 in the U.S. Then I apply the same method on Fatality Rate, Positive Rate, and Recovery Rate.

From this graph, I did not see anything directly, so let me use the other way to find the relationship among them

Step3:Association rule

pre-processing the data

```
rule_uscovid<-subset(uscovid, select = c("dailyLevel.bracket","fatalityLevel.bracket", "positiveLevel.b

rule_uscovid_rules<- apriori(rule_uscovid, parameter = list(support = 0.01, confidence = 0.5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[12 item(s), 248 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [168 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
dailyLevel_good <- apriori(data=rule_uscovid, parameter=list (support=0.01,conf = 0.8), appearance = li
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[12 item(s), 248 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
```

```
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
dailyLevel_bad <- apriori(data=rule_uscovid, parameter=list (support=0.01,conf = 0.8), appearance = lis
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[12 item(s), 248 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [5 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

Check the Association rule

```
inspect(head(sort(dailyLevel_good , by = "lift"), 3))
```

```
##     lhs                                 rhs                             support confidence   coverage
## [1] {fatalityLevel.bracket=medium,
##      positiveLevel.bracket=low}    => {dailyLevel.bracket=good} 0.02822581          1 0.02822581 2.98
## [2] {fatalityLevel.bracket=medium,
##      recoveryLevel.bracket=high}   => {dailyLevel.bracket=good} 0.01612903          1 0.01612903 2.98
## [3] {fatalityLevel.bracket=medium,
##      positiveLevel.bracket=low,
##      recoveryLevel.bracket=medium} => {dailyLevel.bracket=good} 0.01209677          1 0.01209677 2.98
```

```
inspect(head(sort(dailyLevel_bad , by = "lift"), 3))
```

```
##     lhs                                 rhs                            support confidence   coverage     l
## [1] {fatalityLevel.bracket=medium,
##      positiveLevel.bracket=high}   => {dailyLevel.bracket=bad} 0.03629032          1 0.03629032 2.987
## [2] {fatalityLevel.bracket=low,
##      positiveLevel.bracket=high}   => {dailyLevel.bracket=bad} 0.01209677          1 0.01209677 2.987
## [3] {fatalityLevel.bracket=low,
##      recoveryLevel.bracket=low}    => {dailyLevel.bracket=bad} 0.01209677          1 0.01209677 2.987
```

From the table above, we can see that when the daily covid-19 safety level is good. The recovery rate level is high, the fatality rate level is medium or low, and the positive rate level is low.

It explains the daily safety level is directly proportional to the recovery rate level. But it is inversely proportional to the fatality rate level and the positive rate level

Step4:Determine the worst period of U.S covid-19

```
wrost_condition<-uscovid %>% filter(uscovid$dailyLevel.bracket == "bad"&uscovid$fatalityLevel.bracket==

show(wrost_condition$Date)
```

```
##  [1] "2020-04-13" "2020-04-14" "2020-04-15" "2020-04-16" "2020-04-17"
##  [6] "2020-04-18" "2020-04-19" "2020-04-20" "2020-04-21" "2020-04-22"
## [11] "2020-04-23" "2020-04-24" "2020-04-25" "2020-04-26" "2020-04-27"
## [16] "2020-04-28" "2020-04-29" "2020-04-30" "2020-05-01" "2020-05-02"
## [21] "2020-05-03" "2020-05-04" "2020-05-05" "2020-05-06" "2020-05-07"
## [26] "2020-05-08" "2020-05-09" "2020-05-10"
```

```
wroststart.date = as.Date("2020-04-11")
wrostend.date   = as.Date("2020-05-11")
```

From the data shown above, we can see that the worst covid-19 period (from April 1, 2020 to December 6, 2020)is approximately between April and May. so we can set the worst period start date as April 11th and the worst period end date as May 11th.

Part 3:Analyze the effects of covid-19 on the stock marketing among these AMC, Grubhub, Las Vegas Sands Corp, and Zoom.

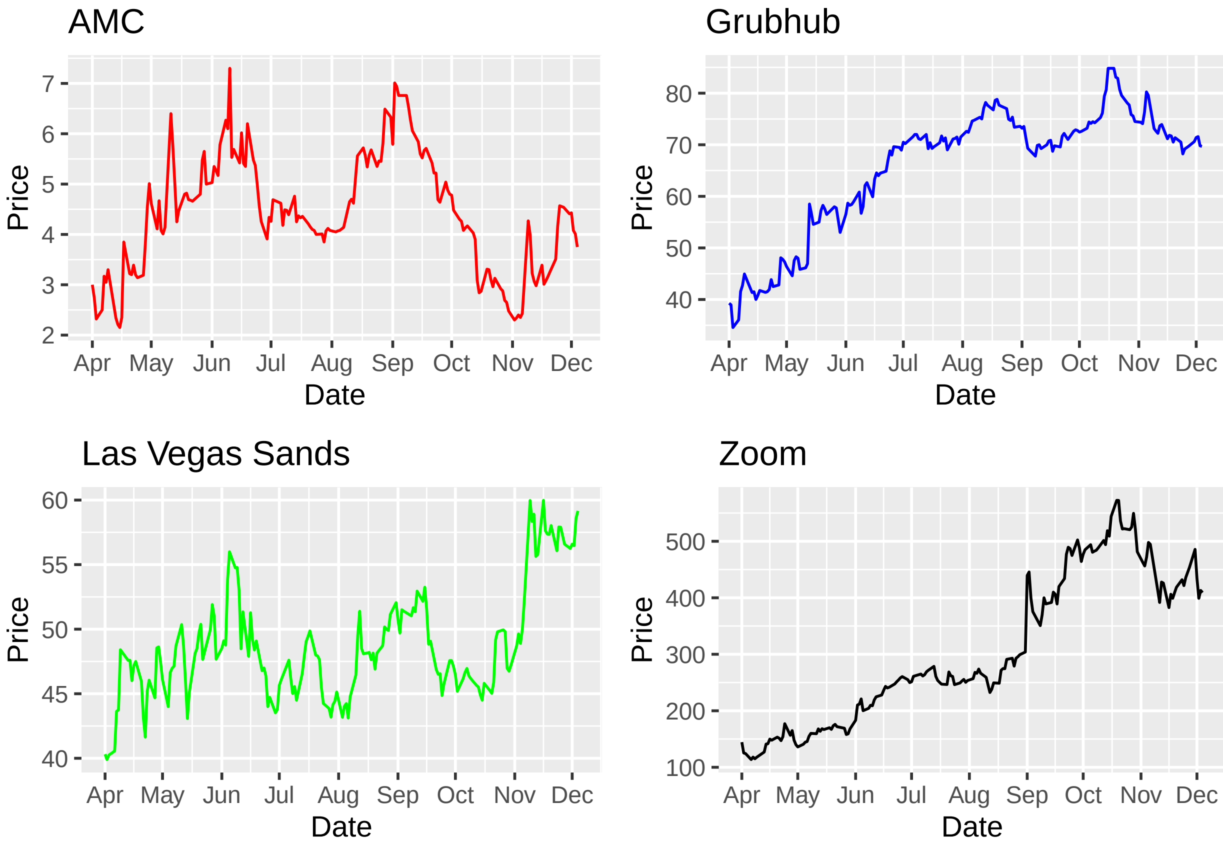
Step1:import external data

```
modify_AMC <- read.csv("~/Desktop/bigdata/Qian Feng project/modify/modify_AMC.csv")
modify_Grubhub <- read.csv("~/Desktop/bigdata/Qian Feng project/modify/modify_Grubhub.csv")
modify_LasVegas <- read.csv("~/Desktop/bigdata/Qian Feng project/modify/modify_LasVegas.csv")
modify_zoom <- read.csv("~/Desktop/bigdata/Qian Feng project/modify/modify_zoom.csv")
AMC<- na.omit(modify_AMC)
AMC$Date<-as.Date(AMC$Date,"%m/%d/%y")
Grubhub<- na.omit(modify_Grubhub)
Grubhub$Date<-as.Date(Grubhub$Date,"%m/%d/%y")
LasVegas<- na.omit(modify_LasVegas)
LasVegas$Date<-as.Date(LasVegas$Date,"%m/%d/%y")
zoom<- na.omit(modify_zoom)
zoom$Date<-as.Date(zoom$Date,"%m/%d/%y")
```

Step2:Plot four company's stock marketing trend

```
AMC_graph<-ggplot(AMC, aes(x=Date,y=Price))+ geom_line(colour="red")+ggtitle("AMC")+scale_x_date(breaks

Grubhub_graph<-ggplot(Grubhub, aes(x=Date,y=Price))+ geom_line(colour="blue")+ggtitle("Grubhub")+scale_

LasVegas_graph<-ggplot(LasVegas, aes(x=Date,y=Price))+ geom_line(colour="green")+ggtitle("Las Vegas San

zoom_graph<-ggplot(zoom, aes(x=Date,y=Price))+ geom_line(colour="black")+ggtitle("Zoom")+scale_x_date(b
```

```
grid.arrange(AMC_graph, Grubhub_graph,LasVegas_graph,zoom_graph)
```



In general, we can see that during the period from April 1, 2020, to December 6, 2020, the stock price of AMC first increases in a short period of time then decreases in a short period of time, then increases in a short period of time again. And finally continues to decrease.

During the period, the stock price of Las Vegas Sands first increases in a short period of time then decreases in a short period of time, then increases in a short period of time again, then decreases in a short period of time once again. And finally continues to increase

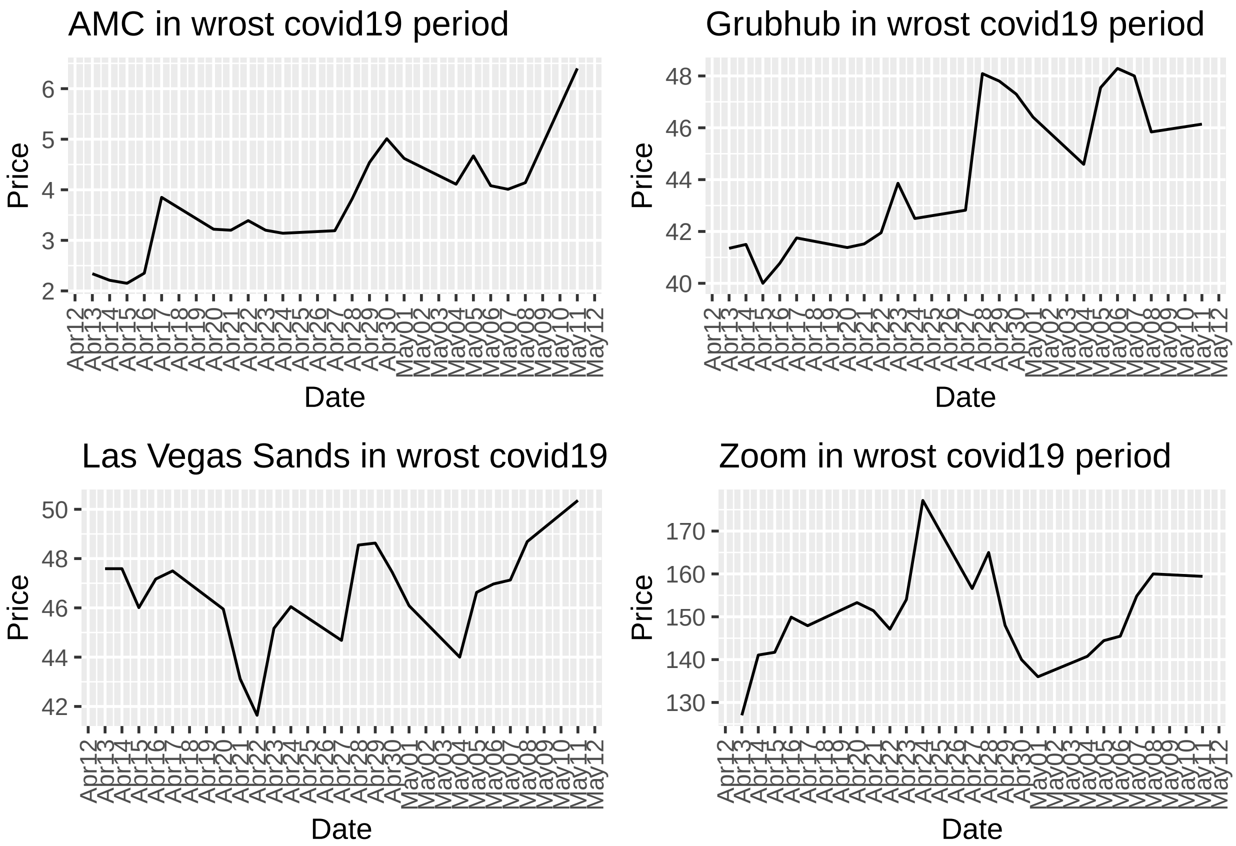
During the period,the stock price of Grubhub is continues to increase

During the period,the stock price of Zoom is continues to increase

Step3:Four company's stock marketing trend in worst covid-19 period

```
AMC_filtered = data=AMC %>% filter(Date >= wroststart.date & Date <= wrostend.date)
Grubhub_filtered = data=Grubhub %>% filter(Date >= wroststart.date & Date <= wrostend.date)
LasVegas_filtered = data=LasVegas %>% filter(Date >= wroststart.date & Date <= wrostend.date)
zoom_filtered = data=zoom %>% filter(Date >= wroststart.date & Date <= wrostend.date)

AMC_wrost<-ggplot(AMC_filtered, aes(x=Date,y=Price))+ geom_line(colour="black")+ggtitle("AMC in wrost c

Grubhub_wrost<-ggplot(Grubhub_filtered, aes(x=Date,y=Price))+ geom_line(colour="black")+ggtitle("Grubhu

LasVegas_wrost<-ggplot(LasVegas_filtered, aes(x=Date,y=Price))+ geom_line(colour="black")+ggtitle("Las
```

```
zoom_wrost<-ggplot(zoom_filtered, aes(x=Date,y=Price))+ geom_line(colour="black")+ggtitle("Zoom in wros

grid.arrange(AMC_wrost, Grubhub_wrost,LasVegas_wrost,zoom_wrost)
```



Step4:Association rule

seperate stock price into three level

```
AMC$Price.bracket<-discretize(AMC$Price, method = "frequency",breaks = 3,labels = c("low","medium","hig
Grubhub$Price.bracket<-discretize(Grubhub$Price, method = "frequency",breaks = 3,labels = c("low","medi
LasVegas$Price.bracket<-discretize(LasVegas$Price, method = "frequency",breaks = 3,labels = c("low","me
zoom$Price.bracket<-discretize(zoom$Price, method = "frequency",breaks = 3,labels = c("low","medium","h
```

I use the same pattern in Part 1 Step 2 to separate the AMC's stock open price into three levels "low", "medium", "high". Indicate the daily situation of AMC's stock market more clearly. Then I apply the same method for Grubhub, Las Vegas Sand, and Zoom.

export the data and assembly all together

```
rule_uscovidwithTime<-subset(uscovid, select = c("Date","dailyLevel.bracket","fatalityLevel.bracket", "
write.csv(rule_uscovidwithTime,"rule_uscovidwithTime.csv", row.names = FALSE)
write.csv(AMC,"AMCedit.csv", row.names = FALSE)
write.csv(Grubhub,"Grubhubedit.csv", row.names = FALSE)
write.csv(LasVegas,"LasVegasedit.csv", row.names = FALSE)
write.csv(zoom,"zoomedit.csv", row.names = FALSE)
```

import the revised data

```r
rule_uscovidwithTimenew <- read.csv("~/Desktop/bigdata/Final/rule_uscovidwithTime copy.csv")
Allfactordata<- na.omit(rule_uscovidwithTimenew)
```

processing data

```r
Allfactordata$dailyLevel.bracket <- as.factor(Allfactordata$dailyLevel.bracket)
Allfactordata$fatalityLevel.bracket <- as.factor(Allfactordata$fatalityLevel.bracket)
Allfactordata$positiveLevel.bracket <- as.factor(Allfactordata$positiveLevel.bracket)
Allfactordata$recoveryLevel.bracket <- as.factor(Allfactordata$recoveryLevel.bracket)
Allfactordata$LasVegas.bracket <- as.factor(Allfactordata$LasVegas.bracket)
Allfactordata$Grubhub.bracket <- as.factor(Allfactordata$Grubhub.bracket)
Allfactordata$AMC.bracket <- as.factor(Allfactordata$AMC.bracket)
Allfactordata$Zoom.bracket <- as.factor(Allfactordata$Zoom.bracket)
```

Build Association rule

```r
all_rules<- apriori(Allfactordata, parameter = list(support = 0.01, confidence = 0.5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[24 item(s), 173 transaction(s)] done [0.00s].
## sorting and recoding items ... [24 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
## writing ... [18324 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
control_good <- apriori(data=Allfactordata, parameter=list (supp=0.01,conf = 0.8), appearance = list (l
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
```

```
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[4 item(s)] done [0.00s].
## set transactions ...[24 item(s), 173 transaction(s)] done [0.00s].
## sorting and recoding items ... [24 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [14 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
AMC_good <- apriori(data=Allfactordata, parameter=list (supp=0.01,conf = 0.8), appearance = list (rhs="
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[24 item(s), 173 transaction(s)] done [0.00s].
## sorting and recoding items ... [24 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
## writing ... [610 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
LasVegas_good <- apriori(data=Allfactordata, parameter=list (supp=0.01,conf = 0.8), appearance = list (
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[24 item(s), 173 transaction(s)] done [0.00s].
## sorting and recoding items ... [24 item(s)] done [0.00s].
```

```
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
## writing ... [437 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
Grubhub_good <- apriori(data=Allfactordata, parameter=list (supp=0.01,conf = 0.8), appearance = list (rh
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[24 item(s), 173 transaction(s)] done [0.00s].
## sorting and recoding items ... [24 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
## writing ... [332 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
Zoom_good <- apriori(data=Allfactordata, parameter=list (supp=0.01,conf = 0.8), appearance = list (rhs=
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[24 item(s), 173 transaction(s)] done [0.00s].
## sorting and recoding items ... [24 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
## writing ... [763 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

check the rules

```
inspect(head(sort(control_good, by = "support"), 3))
```

```
##     lhs                                rhs                     support confidence  coverage     lift coun
## [1] {recoveryLevel.bracket=high} => {Zoom.bracket=high} 0.3179191  0.9322034 0.3410405 2.733410    55
## [2] {fatalityLevel.bracket=low}  => {Zoom.bracket=high} 0.3063584  0.8833333 0.3468208 2.590113    53
## [3] {fatalityLevel.bracket=low,
##      recoveryLevel.bracket=high} => {Zoom.bracket=high} 0.3063584  0.9298246 0.3294798 2.726435    53
```

```
inspect(head(sort(AMC_good, by = "support"), 3))
```

```
##     lhs                                  rhs                    support confidence  coverage     lift cou
## [1] {dailyLevel.bracket=good,
##      recoveryLevel.bracket=medium} => {AMC.bracket=high} 0.1502890  1.0000000 0.1502890 2.883333    2
## [2] {dailyLevel.bracket=good,
##      LasVegas.bracket=high}        => {AMC.bracket=high} 0.1387283  0.9230769 0.1502890 2.661538    2
## [3] {dailyLevel.bracket=good,
##      LasVegas.bracket=medium}      => {AMC.bracket=high} 0.1156069  0.9090909 0.1271676 2.621212    2
```

```
inspect(head(sort(LasVegas_good, by = "support"), 3))
```

```
##     lhs                                rhs                         support confidence  coverage     lift c
## [1] {dailyLevel.bracket=bad,
##      recoveryLevel.bracket=high} => {LasVegas.bracket=high} 0.1156069  0.9523810 0.1213873 2.840722
## [2] {dailyLevel.bracket=bad,
##      fatalityLevel.bracket=low}  => {LasVegas.bracket=high} 0.1156069  0.8333333 0.1387283 2.485632
## [3] {dailyLevel.bracket=bad,
##      fatalityLevel.bracket=low,
##      recoveryLevel.bracket=high} => {LasVegas.bracket=high} 0.1156069  0.9523810 0.1213873 2.840722
```

```
inspect(head(sort(Grubhub_good, by = "support"), 3))
```

```
##     lhs                                rhs                        support confidence  coverage     lift co
## [1] {positiveLevel.bracket=low,
##      AMC.bracket=low}            => {Grubhub.bracket=high} 0.1387283        0.8 0.1734104 2.386207
## [2] {positiveLevel.bracket=low,
##      recoveryLevel.bracket=high,
##      AMC.bracket=low}            => {Grubhub.bracket=high} 0.1387283        0.8 0.1734104 2.386207
## [3] {fatalityLevel.bracket=low,
##      positiveLevel.bracket=low,
##      AMC.bracket=low}            => {Grubhub.bracket=high} 0.1387283        0.8 0.1734104 2.386207
```

```
inspect(head(sort(Zoom_good, by = "support"), 3))
```

```
##     lhs                                rhs                     support confidence  coverage     lift coun
## [1] {recoveryLevel.bracket=high} => {Zoom.bracket=high} 0.3179191  0.9322034 0.3410405 2.733410    55
## [2] {fatalityLevel.bracket=low}  => {Zoom.bracket=high} 0.3063584  0.8833333 0.3468208 2.590113    53
## [3] {fatalityLevel.bracket=low,
##      recoveryLevel.bracket=high} => {Zoom.bracket=high} 0.3063584  0.9298246 0.3294798 2.726435    53
```

From the above data, we can see that 1.The stock marketing of Zoom is positive relative to the U.S covid-19 daily safety level(good).

2.The stock marketing of AMC is positive relative to the U.S covid-19 daily safety level(good) and Las Vegas Sands Stock marketing(high).

3.The stock marketing of Las Vegas Sands is positive relative to the recovery rate level and fatality rate(low) of U.S covid-19. But negative relative to the covid-19 daily safety level(bad).

4.The stock marketing of Grubhub is positive relative to the positive rate level(low) and fatality rate level(low) of U.S covid-19. But negative relative to AMC's stock marketing(low).

5.The stock marketing of Zoom is positive relative to the recovery rate level(high) and positive rate level(low) of U.S covid-19.

Conclusion

The impact of covid-19 on all four companies are massive. For companies like AMC and Las Vegas Sands. It is because they all depend on offline services, so their stock marketing will fluctuate with the U.S. covid-19 situation. If the government controls well in that period, their stock price will rise. For companies like Zoom and Grubhub, which focus on online services. They are more and more popular in the U.S during the epidemic. As a result, their stock price is increased a lot.

In addition, the stock marketing of the four companies will become better with the covid-19 in the United States. It shows that if the epidemic is well controlled, both offline services companies and online services companies in the United States will benefit