Final Project - STAT 508

Joseph Abraham, Pallavi Surana, Xue Zhang

5/5/2020

Intro

For this final project, we will be analyzing data from two different sources regarding the Coronavirus (COVID-19) pandemic, comprised of information from many countries and regions. The onset of this strain of the coronavirus is arguably the most internationally debilitating event since the Spanish Flu in 1918.

Now, in the midst of the ongoing situation, we will be analyzing data to validate commonly held notions about the disease and attempt to meaningfully model and segment its spread throughout the world using statistical techniques including logistic regression, non-linear regression, principal components analysis, and regression trees.

Data

The data used in this analysis is two sets of data regarding the spread of the coronavirus. One set, obtained from Kaggle, and the data, contains information on individuals in China, regarding their age, gender, survival status, and more.

These statistics from many different countries depicts the wild and proliferous spread of the coronavirus (covid-19) that has become a worldwide pandemic affecting nearly every country.

Structure of Larger Dataset

The data has many rows and 7 columns, with records through Feb 9th, 2020. It is updated on a regular basis.

Each column has the following details: 1. Province/State - Province or State 2. Country/Region - Country or region 3. Lat - latitude 4. Long - longitude 5. Date - Date when the observations were recorded 6. cases - Cumulative number of cases reported 7. type - 3 types of cases, confirmed, recovered or death

Sources and References Used for Data

- Source 1 - https://github.com/RamiKrispin/coronavirus
- Reference 1 - https://covid19r.github.io/coronavirus/
- Reference 2 - https://github.com/CSSEGISandData/COVID-19
- Source of kaggle dataset- https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset#covid_19_data.csv

```
covid=read.csv("/Users/josephabraham/Documents/THE LAST SEMESTER/STAT 508/Final Project/novel-corona-vi

## Data on multiple individual cases, mostly from China.
head(covid)
```

```
##   id reporting.date           location country gender age visiting.Wuhan
## 1  1        1/20/20 Shenzhen, Guangdong   China   male  66              1
## 2  2        1/20/20            Shanghai   China female  56              0
```

```
## 3  3        1/21/20            Zhejiang   China   male  46             0
## 4  4        1/21/20             Tianjin   China female  60             1
## 5  5        1/21/20             Tianjin   China   male  58             0
## 6  6        1/21/20           Chongqing   China female  44             0
##   from.Wuhan death recovered
## 1          0     0         0
## 2          1     0         0
## 3          1     0         0
## 4          0     0         0
## 5          0     0         0
## 6          1     0         0
```

```
## Regional coronavirus spread over days from different countries.
head(coronavirus)
```

```
## # A tibble: 6 x 7
##   Province.State Country.Region   Lat  Long date       cases type
##   <chr>          <chr>          <dbl> <dbl> <date>     <int> <chr>
## 1 ""             Japan           35.7  140. 2020-01-22     2 confirmed
## 2 ""             South Korea     37.6  127. 2020-01-22     1 confirmed
## 3 ""             Thailand        13.8  101. 2020-01-22     2 confirmed
## 4 "Anhui"        Mainland China  31.8  117. 2020-01-22     1 confirmed
## 5 "Beijing"      Mainland China  40.2  116. 2020-01-22    14 confirmed
## 6 "Chongqing"    Mainland China  30.1  108. 2020-01-22     6 confirmed
```

Analysis

First, the effect of age and gender on death will be tested using a logistic regression model on the individual case data.

```
no_age=is.na(covid$age)
covid_age=covid[-no_age,]
death_var=rep(1,length(covid_age$id))
death_var[covid_age$death=="0"]=0
log.model=glm(death_var~gender*age,family=poisson(link="log"),data=covid_age)
summary(log.model)
```

```
##
## Call:
## glm(formula = death_var ~ gender * age, family = poisson(link = "log"),
##     data = covid_age)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1710  -0.3746  -0.2236  -0.1292   2.5888
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -8.29833    1.39793  -5.936 2.92e-09 ***
## gendermale      2.09353    1.56871   1.335    0.182
## age             0.08251    0.01917   4.303 1.68e-05 ***
## gendermale:age -0.01800    0.02176  -0.827    0.408
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 307.83  on 823  degrees of freedom
## Residual deviance: 229.09  on 820  degrees of freedom
##   (260 observations deleted due to missingness)
## AIC: 353.09
##
## Number of Fisher Scoring iterations: 6
```

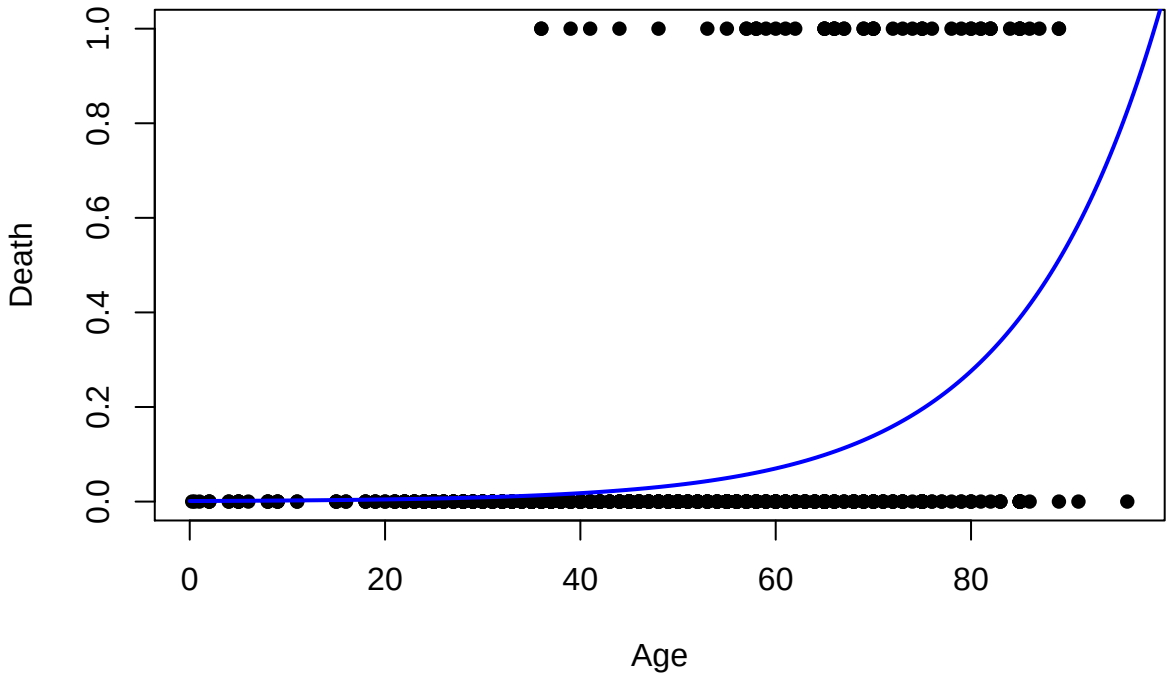
The interaction between gender and age was not significant at the $\alpha = 0.05$ level and is therefore removed.

```
log.model=glm(death_var~gender+age,family=poisson(link="log"),data=covid_age)
summary(log.model)
```

```
##
## Call:
## glm(formula = death_var ~ gender + age, family = poisson(link = "log"),
##     data = covid_age)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0219  -0.3643  -0.2395  -0.1335   2.4701
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.33380    0.68034 -10.780  < 2e-16 ***
## gendermale   0.83885    0.30685   2.734  0.00626 **
## age          0.06877    0.00910   7.557 4.12e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 307.83  on 823  degrees of freedom
## Residual deviance: 229.79  on 821  degrees of freedom
##   (260 observations deleted due to missingness)
## AIC: 351.79
##
## Number of Fisher Scoring iterations: 6
```

It appears that both gender and age are significant predictors of the likelihood of death, as men seem to be $exp(0.83885) = 2.314$ times as likely to die, all other predictors held constant, and for each one year incrase in age, a person is $exp(0.06877) = 1.07119$ times as likely to die. Below is a graphical representation of the model considering just the main effect on age.

```
age.model=glm(death_var~age,family=poisson(link="log"),data=covid_age)
xage <- seq(0,100, 0.1)
yage <- predict(age.model, list(age=xage),type="response")
plot(covid_age$age,death_var, pch = 16, xlab = "Age",ylab="Death")
lines(xage, yage, col= "blue", lwd = 2)
```



USA State Rate of Change for every 3 days

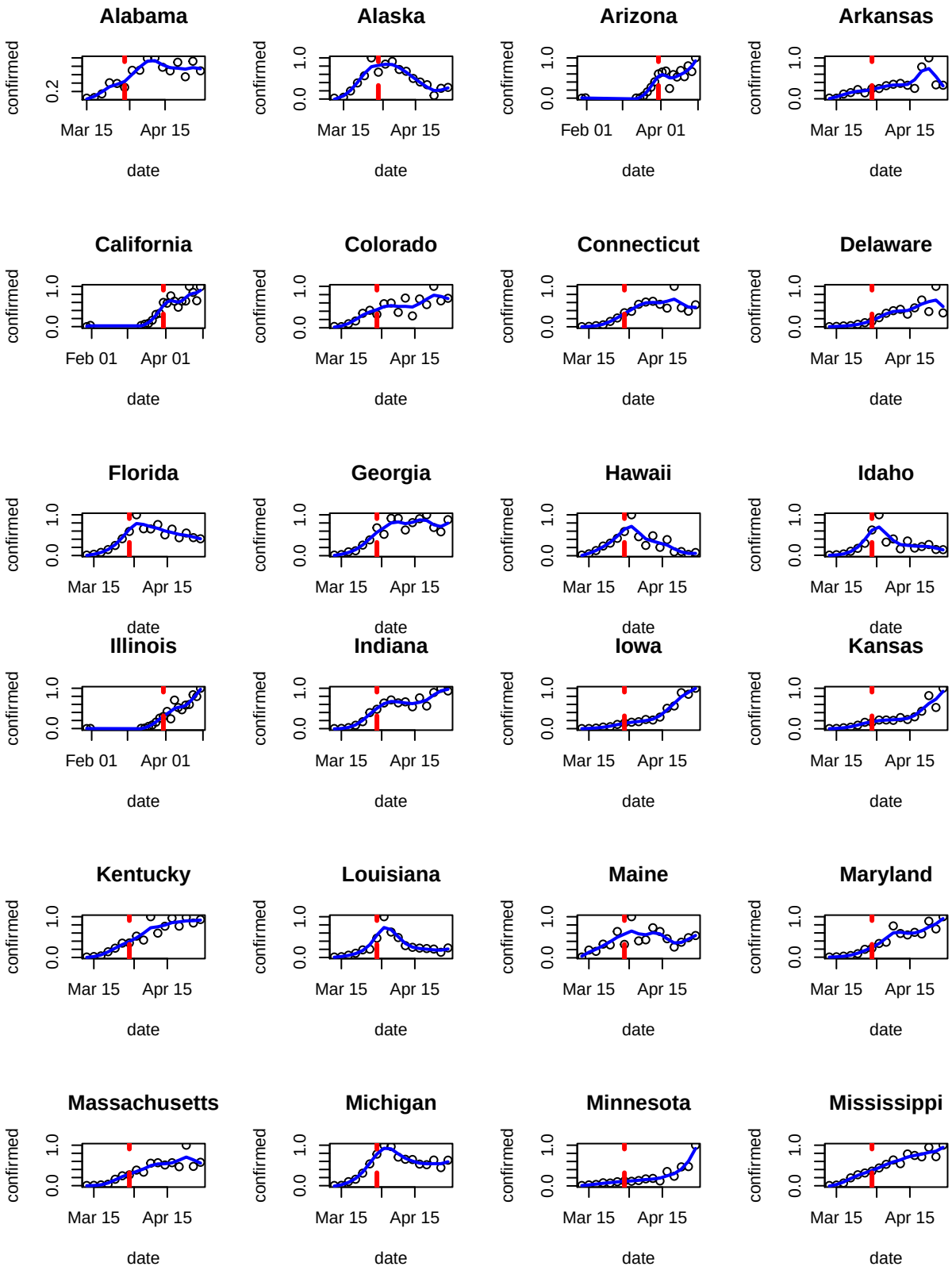
```
data <- read.csv("/Users/josephabraham/Documents/THE LAST SEMESTER/STAT 508/Final Project/novel-corona-
data$ObservationDate <- as.Date(as.character(data$ObservationDate), format="%m/%d/%Y")
str(data)
```

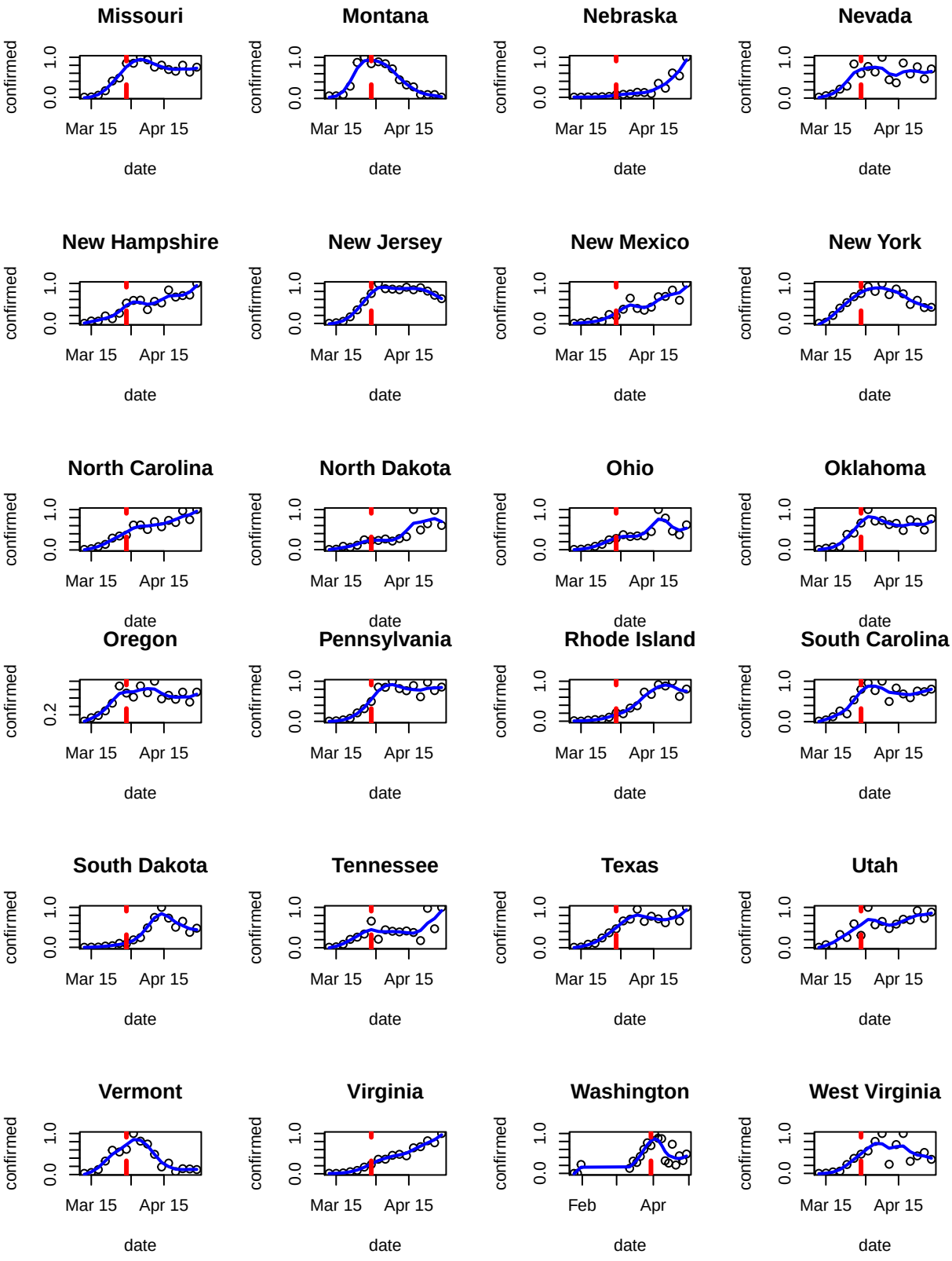
```
## 'data.frame':    20574 obs. of  8 variables:
##  $ SNo            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ ObservationDate: Date, format: "2020-01-22" "2020-01-22" ...
##  $ Province.State : chr  "Anhui" "Beijing" "Chongqing" "Fujian" ...
##  $ Country.Region : chr  "Mainland China" "Mainland China" "Mainland China" "Mainland China" ...
##  $ Last.Update    : chr  "1/22/2020 17:00" "1/22/2020 17:00" "1/22/2020 17:00" "1/22/2020 17:00" ...
##  $ Confirmed      : num  1 14 6 1 0 26 2 1 4 1 ...
##  $ Deaths         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Recovered      : num  0 0 0 0 0 0 0 0 0 0 ...
```

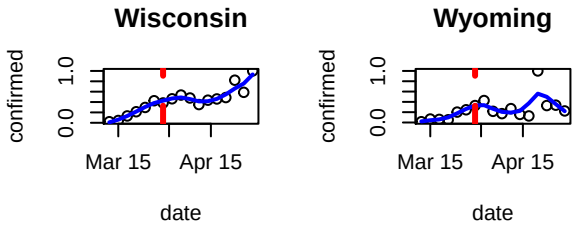
```
#subset U.S.
data.us <- subset(data, Country.Region=="US")
data.us <- subset(data.us,Province.State %in% state.name)

#################### create the function
rate.confirmed <- function(data.ny){
  #calculate increase per day
  data.increase <- data.frame(cbind(data.frame(data.ny[2:nrow(data.ny),2]),apply( data.ny[,6:8], 2 , di
  colnames(data.increase) <- c("ObservationDate","Confirmed","Recovered","Deaths")
  # replace below 0 with 0
  data.increase[data.increase$Confirmed<0,4] <- 0
  # take 3 days average
  if(nrow(data.increase)%%3==0)
    a<- data.increase
  if(nrow(data.increase)%%3==1)
    a<- data.increase[2:nrow(data.increase),]
```

```
  if(nrow(data.increase)%%3==2)
    a<- data.increase[3:nrow(data.increase),]
  day3average <- colSums(matrix(a$Confirmed, nrow=3))

  a$day <- 1:nrow(a)
  date <- a[a$day%%3==1,1]

  # standardization
  confirmed <- cbind(date,data.frame(day3average/max(day3average)))
  colnames(confirmed) <- c("date","rate")
  return(confirmed)
}

# automatically fit and plot the curve
curve <- function(confirmed,name){
  #fit=smooth.spline(confirmed$rate~confirmed$date,cv=T)
  fit=smooth.spline(confirmed$rate~confirmed$date,df=8)
  days.grid=seq(from=1,to=nrow(confirmed))
  pred <- predict(fit, newdata=list(days=days.grid),se=T)
  a <-cor((nrow(confirmed)-10):nrow(confirmed),confirmed$rate[(nrow(confirmed)-10):nrow(confirmed)])
  plot(y=confirmed$rate,x=confirmed$date,ylab="confirmed",xlab="date",main=paste(name))
  lines(confirmed$date,pred$y,lwd=2,col="blue")
  abline(v=(confirmed$date[nrow(confirmed)]-30),col="red", lwd=3, lty=2)
}

######function end######################
```

```
# automatically subset
states <- data.frame()
par(mfrow=c(3,4))
for(i in 1:length(state.name)){
  data.i <- subset(data.us,Province.State == state.name[i])
  confirmed<- rate.confirmed(data.i)
  curve(confirmed,state.name[i])
  states[i,1]<- state.name[i]
  states[i,2] <- cor((nrow(confirmed)-10):nrow(confirmed),confirmed$rate[(nrow(confirmed)-10):nrow(confi
  colnames(states) <- c("name","cor")
}
```


Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
Florida
Georgia
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Maryland
Massachusetts
Michigan
Minnesota
Mississippi
confirmed
date
Mar 15
Apr 15
Feb 01
Apr 01
0.0
0.2
1.0


Missouri
Montana
Nebraska
Nevada
New Hampshire
New Jersey
New Mexico
New York
North Carolina
North Dakota
Ohio
Oklahoma
Oregon
Pennsylvania
Rhode Island
South Carolina
South Dakota
Tennessee
Texas
Utah
Vermont
Virginia
Washington
West Virginia
confirmed
date
Mar 15
Apr 15
Feb
Apr



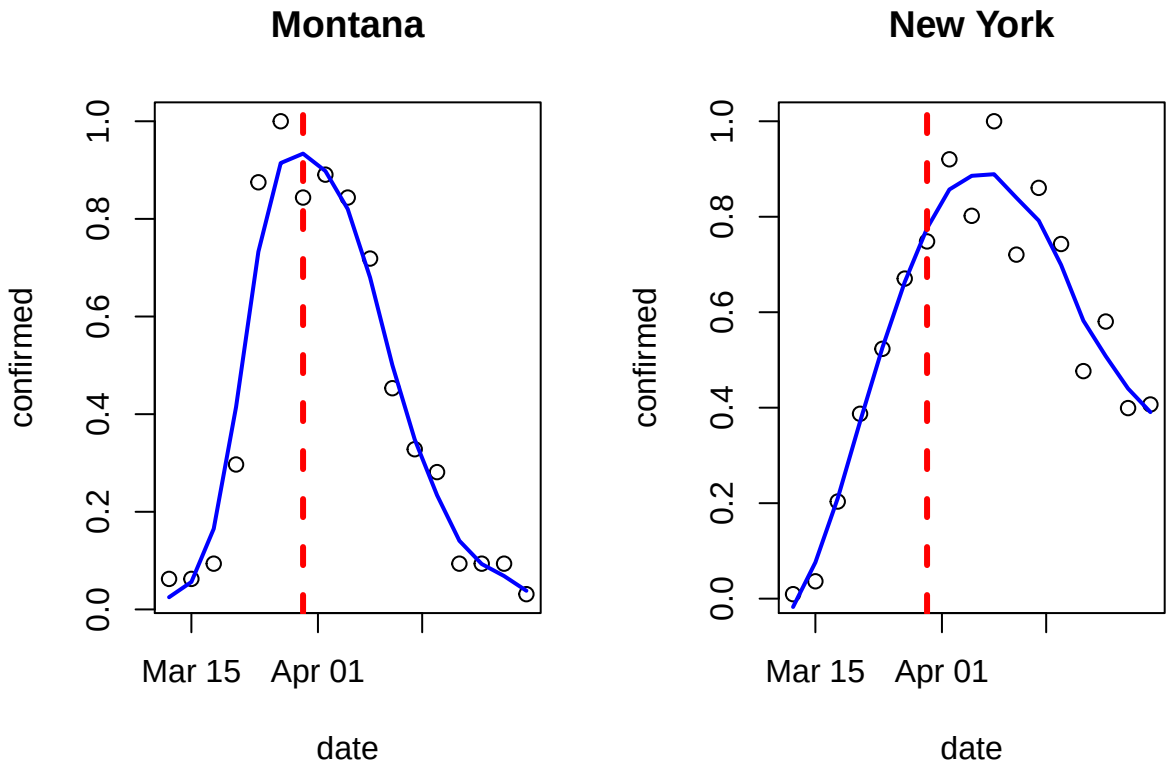

```
states[order(states$cor),]
```

```
##              name        cor
## 26        Montana -0.9632084
## 2          Alaska -0.8779573
## 45        Vermont -0.8700223
## 11         Hawaii -0.8225192
## 32       New York -0.8066119
## 18      Louisiana -0.7720557
## 22       Michigan -0.7704760
## 12          Idaho -0.7391417
## 9          Florida -0.7051189
## 25        Missouri -0.6874513
## 47     Washington -0.6434150
## 30     New Jersey -0.5592740
## 36       Oklahoma -0.3683294
## 40 South Carolina -0.3154857
## 37         Oregon -0.3045642
## 48  West Virginia -0.2846251
## 19          Maine -0.2423299
## 28         Nevada -0.1073032
## 50        Wyoming  0.1212372
## 10        Georgia  0.1909360
## 7     Connecticut  0.2246938
## 38   Pennsylvania  0.2308366
## 1         Alabama  0.2729840
## 42      Tennessee  0.4277872
## 41   South Dakota  0.4373216
## 4         Arkansas  0.4390794
## 44           Utah  0.4686706
## 43          Texas  0.4835310
## 35           Ohio  0.4849369
## 3          Arizona  0.5015626
## 6         Colorado  0.5508674
## 21  Massachusetts  0.5942081
## 5       California  0.5944509
## 8         Delaware  0.6241092
## 49      Wisconsin  0.7044753
## 29  New Hampshire  0.7246503
## 34   North Dakota  0.7446893
## 17        Kentucky  0.7453143
## 14         Indiana  0.7502646
## 23       Minnesota  0.8076373
## 31     New Mexico  0.8085538
## 39   Rhode Island  0.8197784
## 20        Maryland  0.8370391
## 33 North Carolina  0.8374804
```

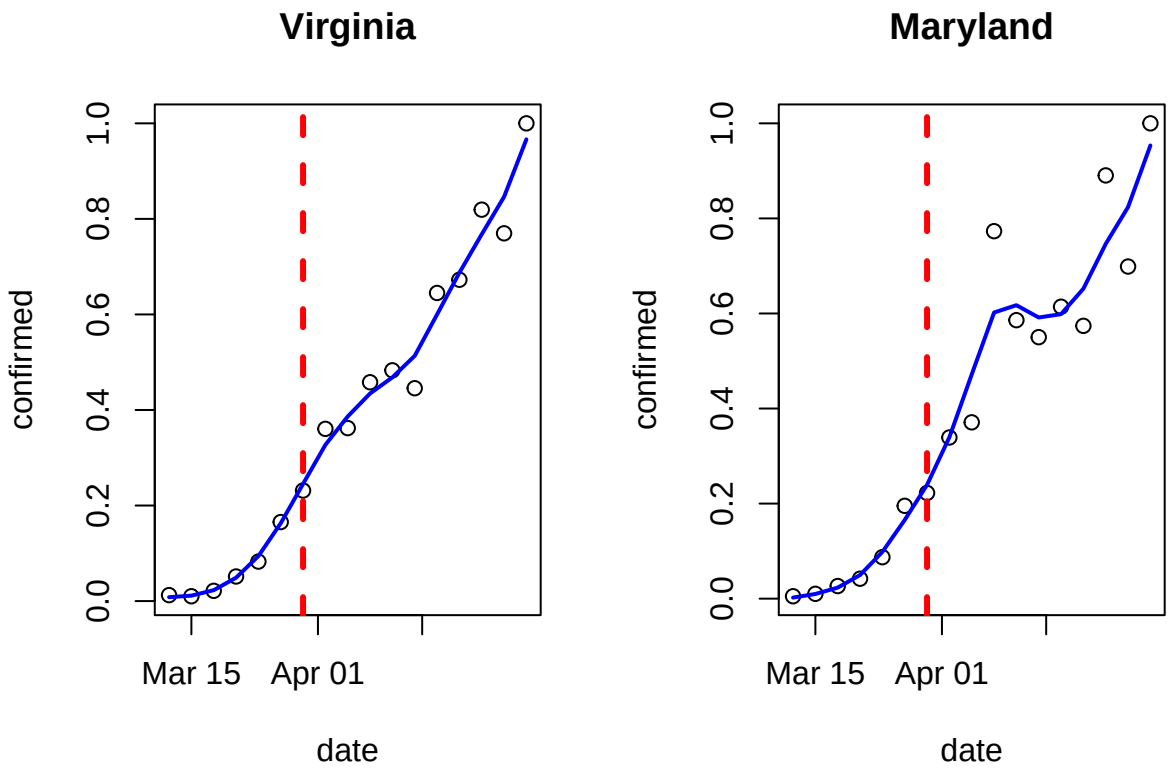
```
## 16         Kansas  0.8390534
## 27       Nebraska  0.8525308
## 13       Illinois  0.8595394
## 24    Mississippi  0.8986104
## 15           Iowa  0.9415724
## 46       Virginia  0.9653384
```

plots

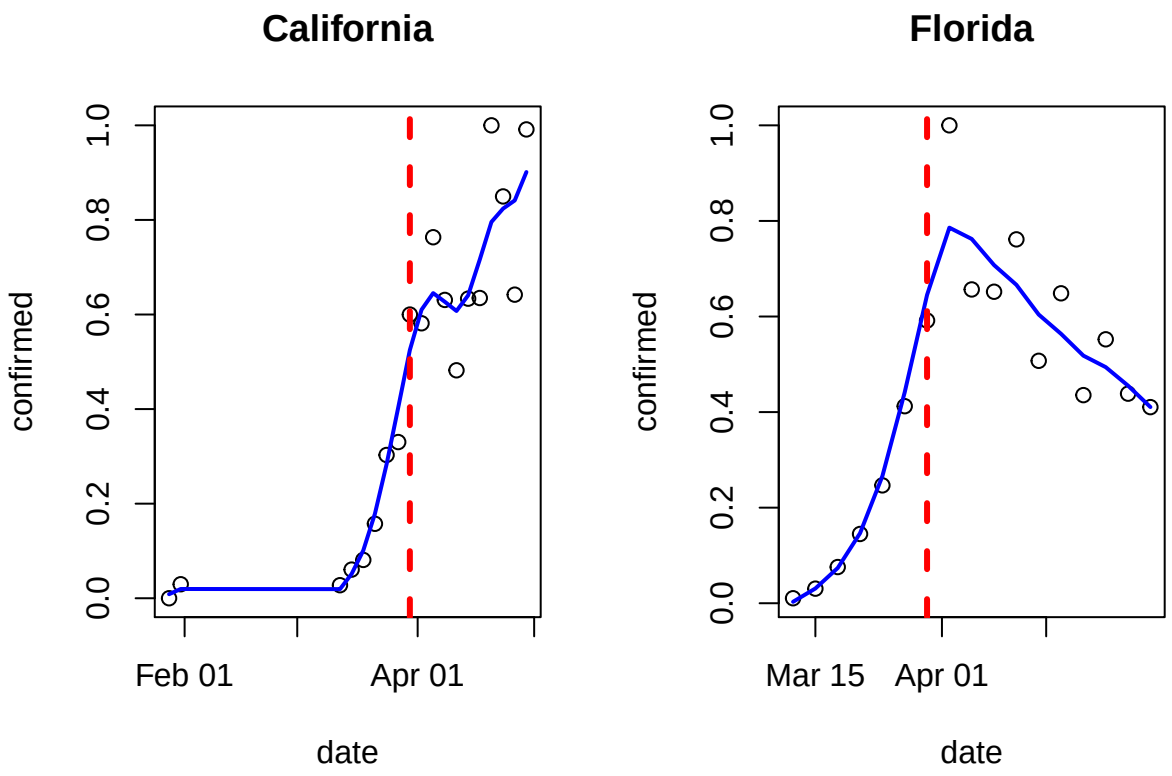
```
par(mfrow=c(1,2))
data.i <- subset(data.us,Province.State == "Montana")
confirmed<- rate.confirmed(data.i)
curve(confirmed,"Montana")

data.i <- subset(data.us,Province.State == "New York")
confirmed<- rate.confirmed(data.i)
curve(confirmed,"New York")
```



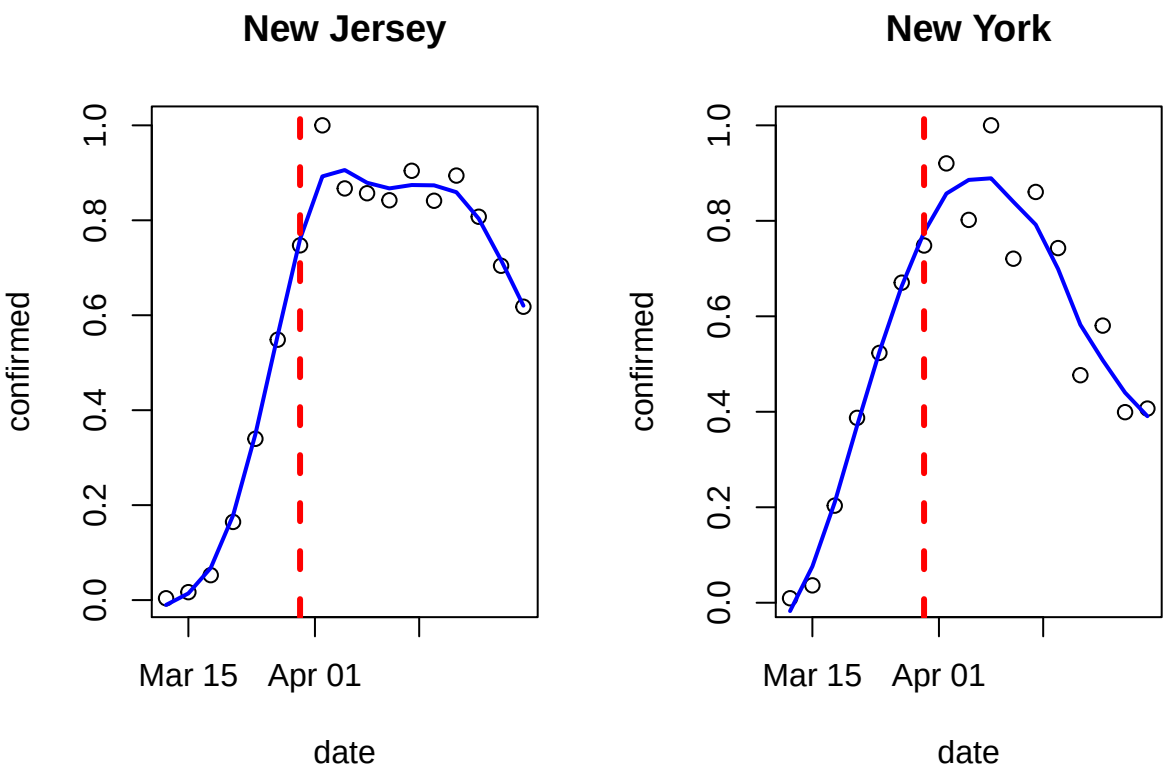

```
data.i <- subset(data.us,Province.State == "Virginia")
confirmed<- rate.confirmed(data.i)
curve(confirmed,"Virginia")

data.i <- subset(data.us,Province.State == "Maryland")
confirmed<- rate.confirmed(data.i)
curve(confirmed,"Maryland")
```




```
data.i <- subset(data.us,Province.State == "California")
confirmed<- rate.confirmed(data.i)
curve(confirmed,"California")

data.i <- subset(data.us,Province.State == "Florida")
confirmed<- rate.confirmed(data.i)
curve(confirmed,"Florida")
```




```
data.i <- subset(data.us,Province.State == "New Jersey")
confirmed<- rate.confirmed(data.i)
curve(confirmed,"New Jersey")

data.i <- subset(data.us,Province.State == "New York")
confirmed<- rate.confirmed(data.i)
curve(confirmed,"New York")
```



The graphs above are from conducting a non-linear regression on the rate of increase in different states over the last 30 days and comparing the correlation of the standerized confirmed cases of recent 30 days. If the correlation is negative, is means the recent trend of confirmed case is declining; if it is high, it means there is a trend of increasing in the recent 30 days.

It shows that Montana, Alaska, Vermont, Hawaii, and New York are having a trend of declining in new confirmed cases, as their correlation are all smaller than -0.8.

On the other hand, the top states that have an increasing new confirmed cases trend are: Virginia, Iowa, Mississippi, Illinois, Nebraska, Kansas, North Carolina, Maryland, Rhode Island, New Mexico and Minnesota, and they have correlation of more than 0.8.

PCA

The result seem good in seperating some of the countries or areas, but we have to disgard the date variable for PCA.

Countries

```
library(reshape2)
```

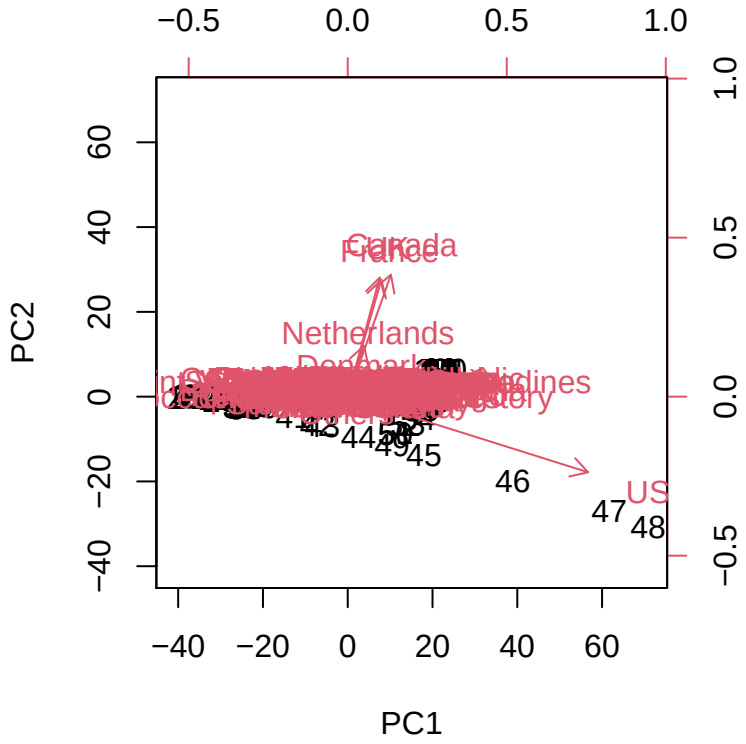
```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

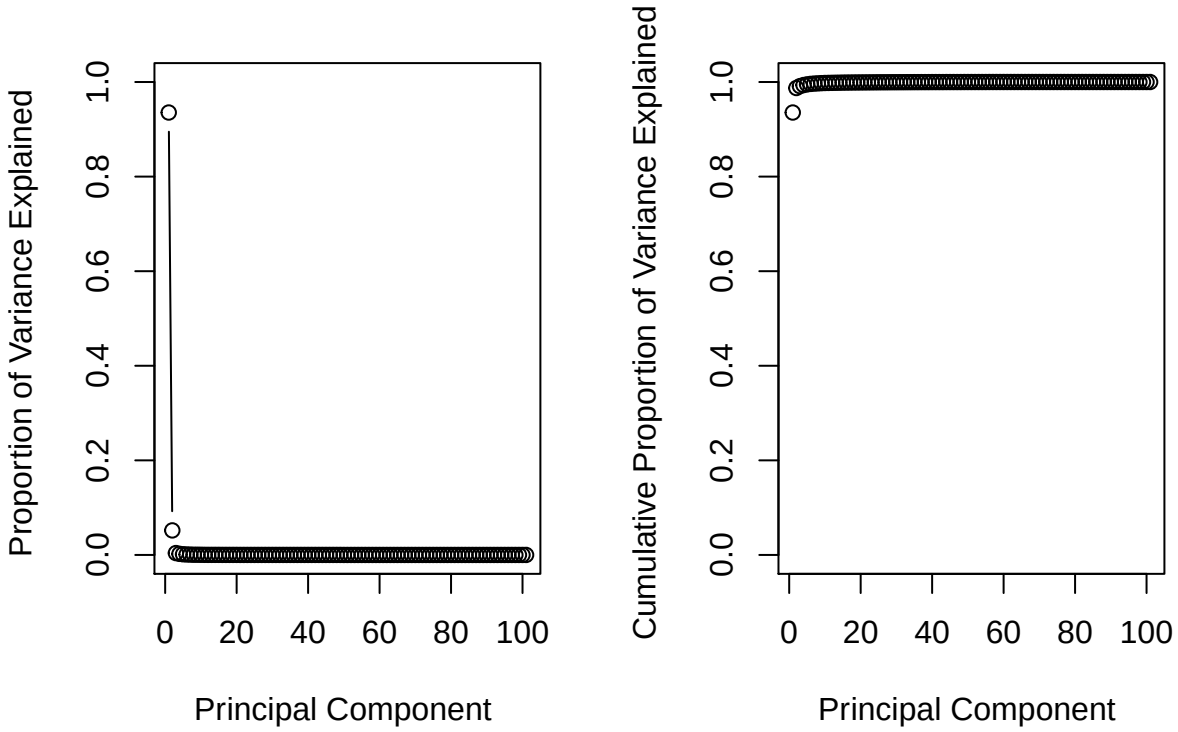
```
# Country.Region
a <- data[,c(2,4,6)]
b <- dcast(a,ObservationDate~Country.Region,value.var = "Confirmed")
```

```
## Aggregation function missing: defaulting to length
```

```
b <- b[,2:221]
b <- b[,colSums(b)>0]

pr.out=prcomp(b)

biplot(pr.out, scale=0)
```




```
par(mfrow=c(1,2))
pr.var=pr.out$sdev ^2
pve=pr.var/sum(pr.var)
plot(pve, xlab="Principal Component ", ylab="Proportion of Variance Explained ", ylim=c(0,1),type="b")
plot(cumsum(pve), xlab="Principal Component ", ylab="Cumulative Proportion of Variance Explained ", yli
     type="b")
```



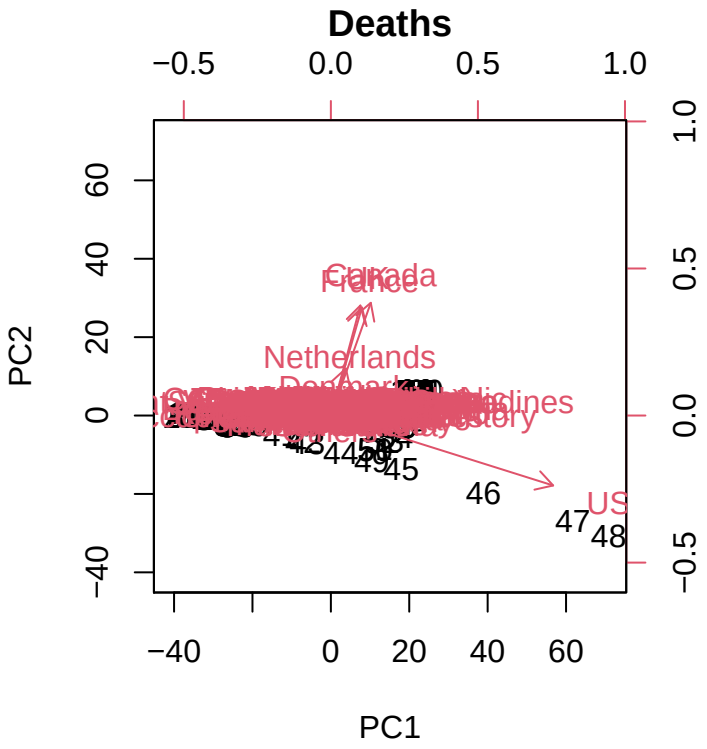
The first two PCs explained more than 90% of the errors. It shows that the U.S. performs differently than the rest of the countries in the first PC. Canada, France and Netherland performs differently than the rest of the countries in the second PC.

Deaths

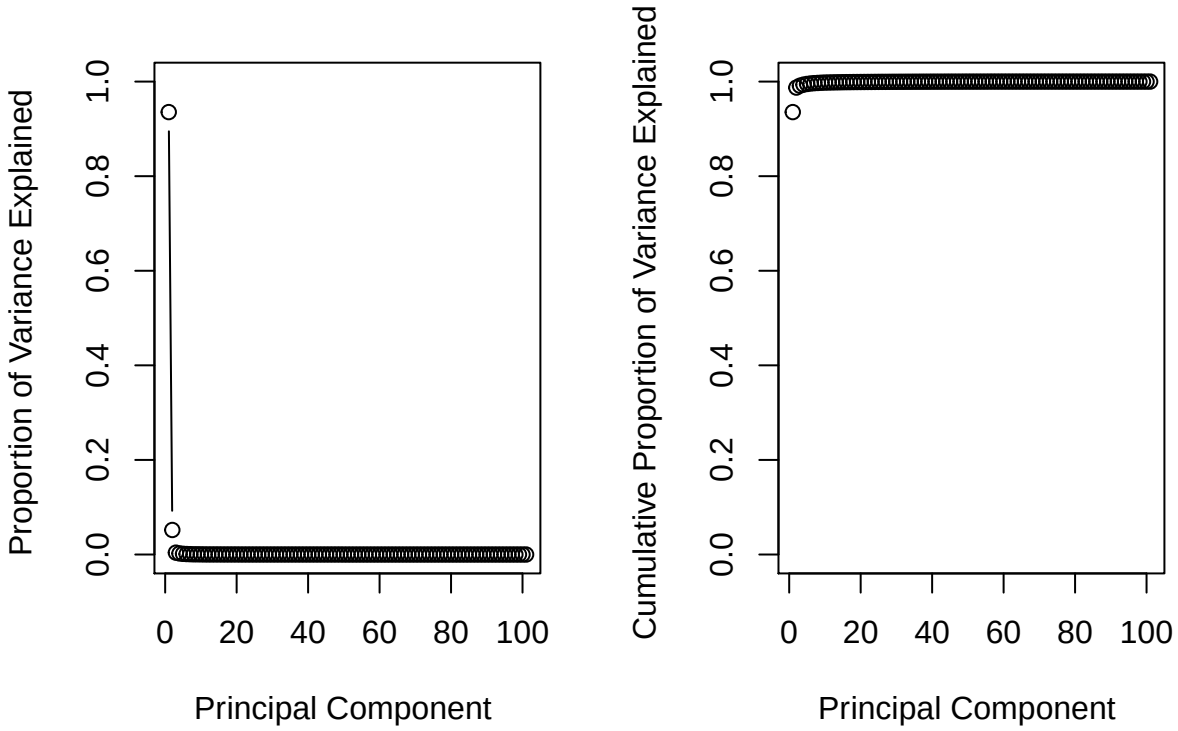
```
library(reshape2)
# Country.Region
a <- data[,c(2,4,7)]
b <- dcast(a,ObservationDate~Country.Region,value.var = "Deaths")
```

```
## Aggregation function missing: defaulting to length
```

```
b <- b[,2:221]
b <- b[,colSums(b)>0]

pr.out=prcomp(b)

biplot(pr.out, scale=0,main = "Deaths")
```




```
par(mfrow=c(1,2))
pr.var=pr.out$sdev ^2
pve=pr.var/sum(pr.var)
plot(pve, xlab="Principal Component ", ylab="Proportion of Variance Explained ", ylim=c(0,1),type="b")
plot(cumsum(pve), xlab="Principal Component ", ylab="Cumulative Proportion of Variance Explained ", yli
     type="b")
```

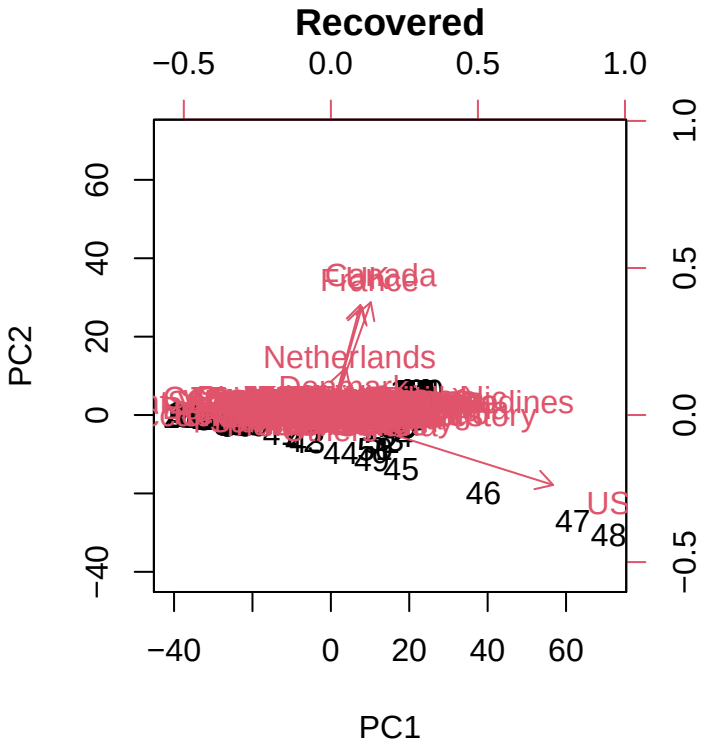


Recovered

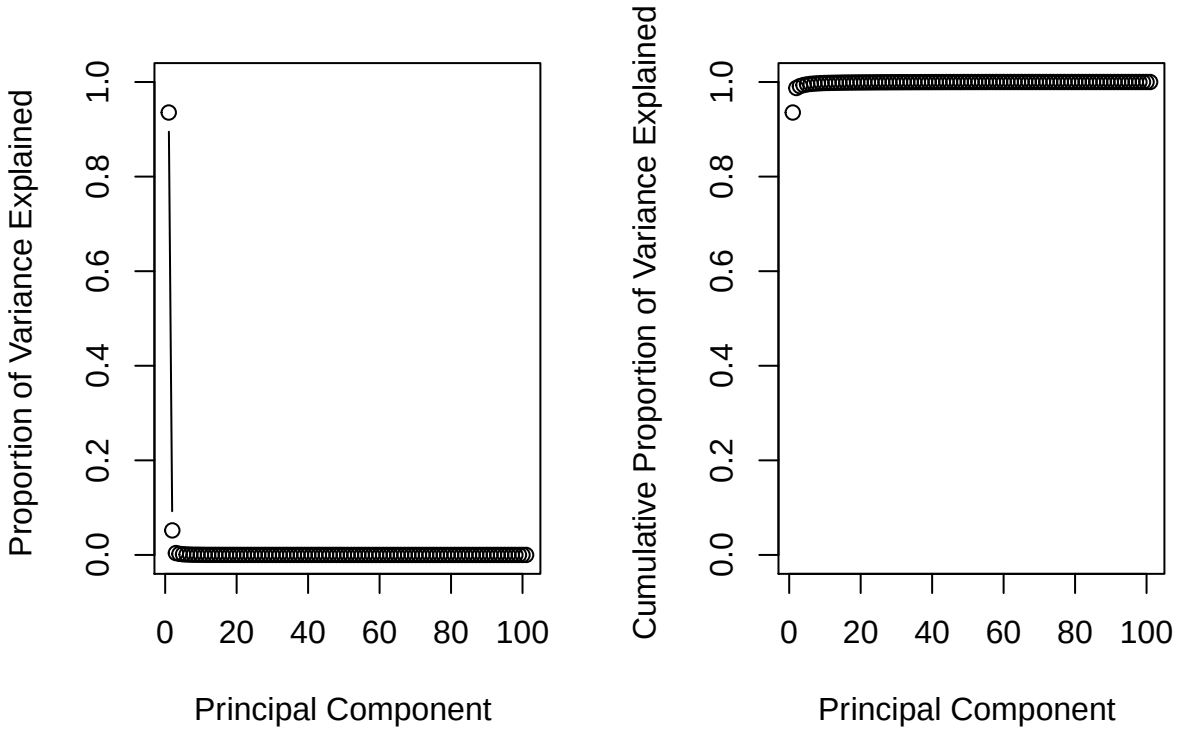
```
library(reshape2)
# Country.Region
a <- data[,c(2,4,8)]
b <- dcast(a,ObservationDate~Country.Region,value.var = "Recovered")
```

```
## Aggregation function missing: defaulting to length
```

```
b <- b[,2:221]
b <- b[,colSums(b)>0]

pr.out=prcomp(b)

biplot(pr.out, scale=0,main = "Recovered")
```

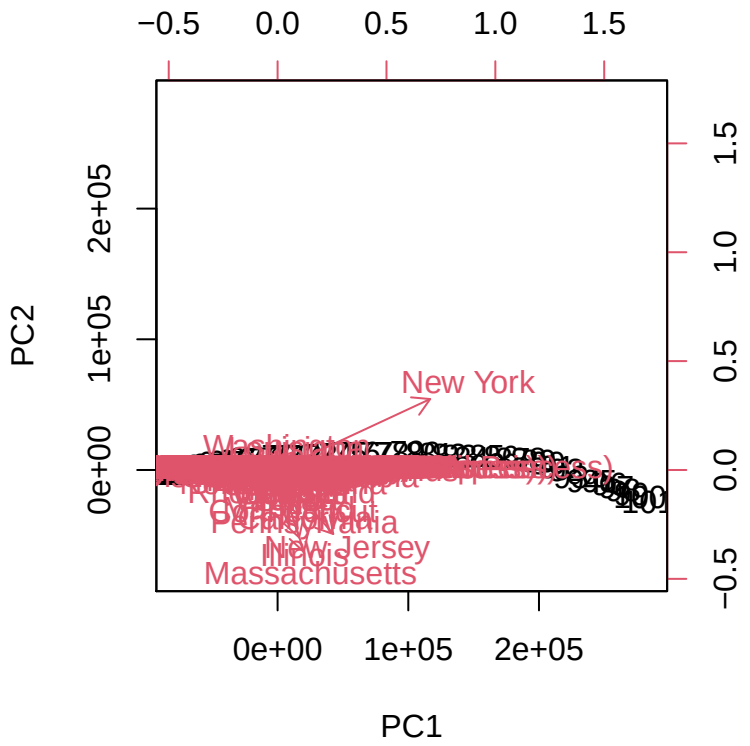



```
par(mfrow=c(1,2))
pr.var=pr.out$sdev ^2
pve=pr.var/sum(pr.var)
plot(pve, xlab="Principal Component ", ylab="Proportion of Variance Explained ", ylim=c(0,1),type="b")
plot(cumsum(pve), xlab="Principal Component ", ylab="Cumulative Proportion of Variance Explained ", yli
     type="b")
```

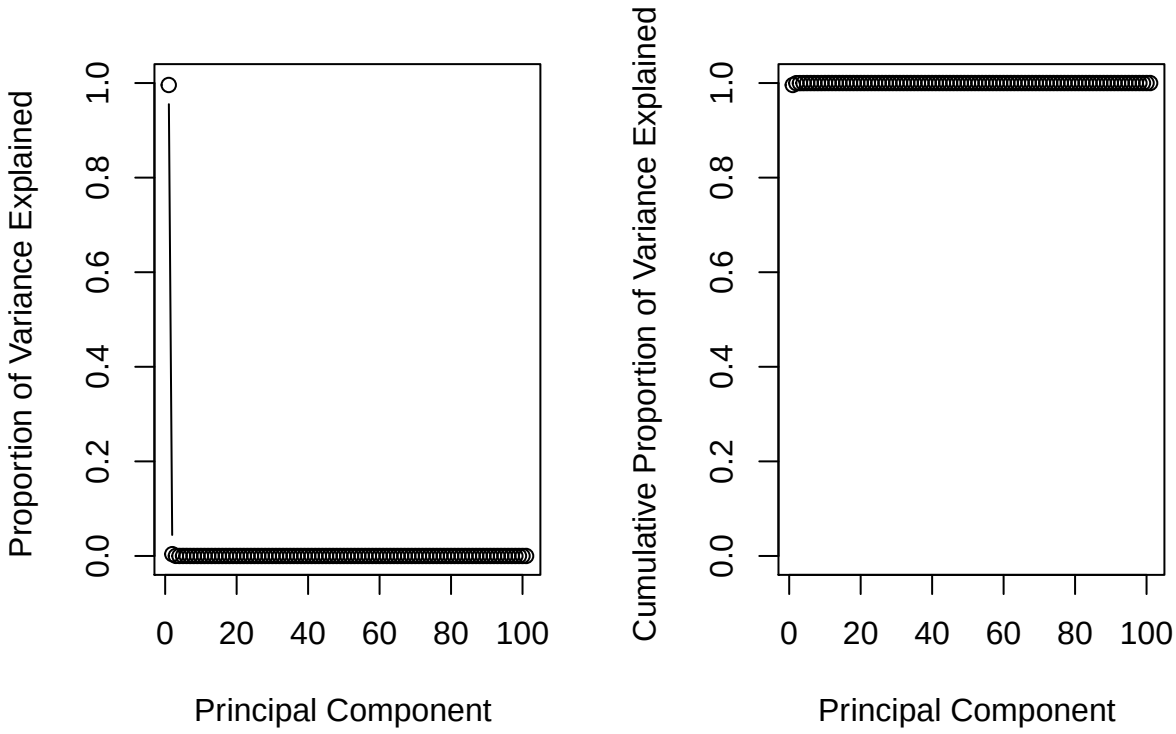


U.S.

```
a<- data[data$Country.Region=="US",c(2,3,6)]
b <- dcast(a,ObservationDate~Province.State,value.var = "Confirmed")
b <- b[,2:200]
b[is.na(b)] <-0
b <- b[,colSums(b)>0]

pr.out=prcomp(b)

biplot(pr.out, scale=0)
```

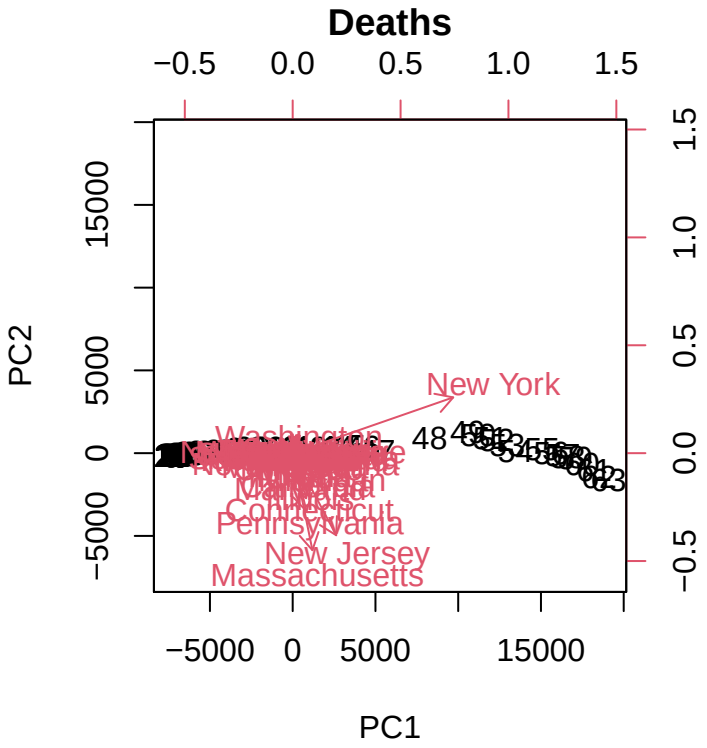



```
pr.var=pr.out$sdev ^2
par(mfrow=c(1,2))
pve=pr.var/sum(pr.var)
plot(pve, xlab="Principal Component ", ylab="Proportion of Variance Explained ", ylim=c(0,1),type="b")
plot(cumsum(pve), xlab="Principal Component ", ylab="Cumulative Proportion of Variance Explained ", yli
     type="b")
```

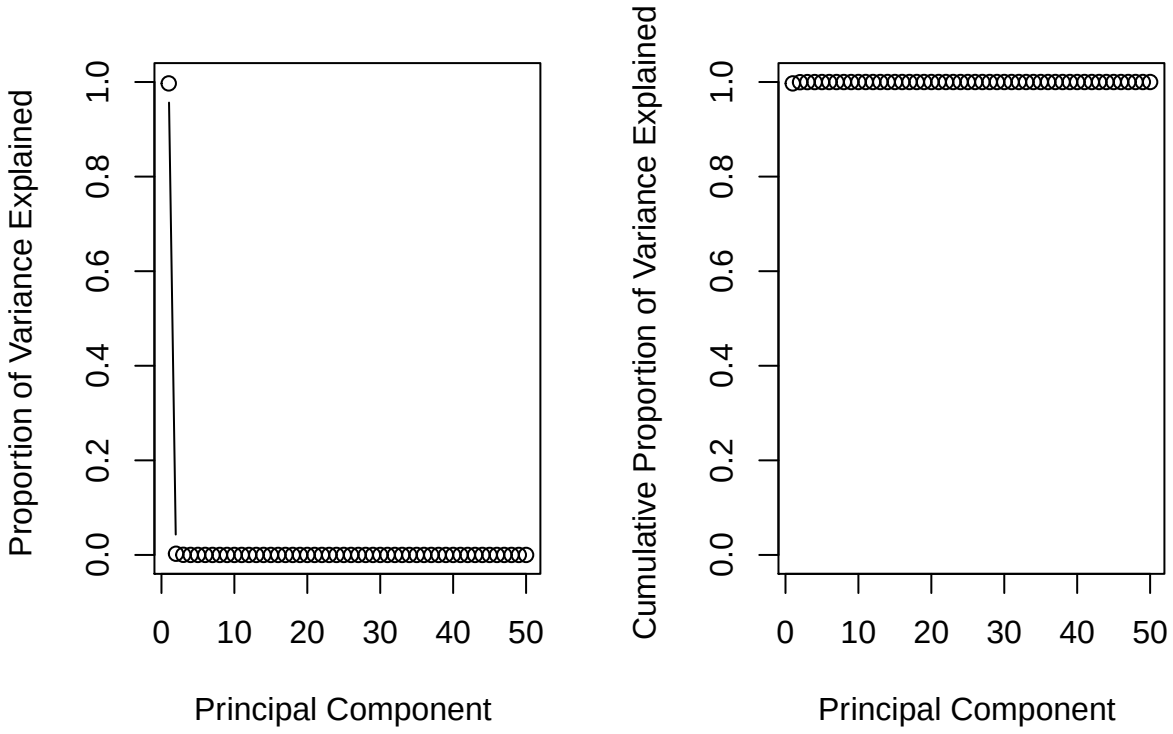


Deaths

```
a<- data.us[data.us$Country.Region=="US",c(2,3,7)]
b <- dcast(a,ObservationDate~Province.State,value.var = "Deaths")
b[is.na(b)] <-0
b <- b[,2:51]
b <- b[,colSums(b)>0]

pr.out=prcomp(b)

biplot(pr.out, scale=0,main = "Deaths")
```

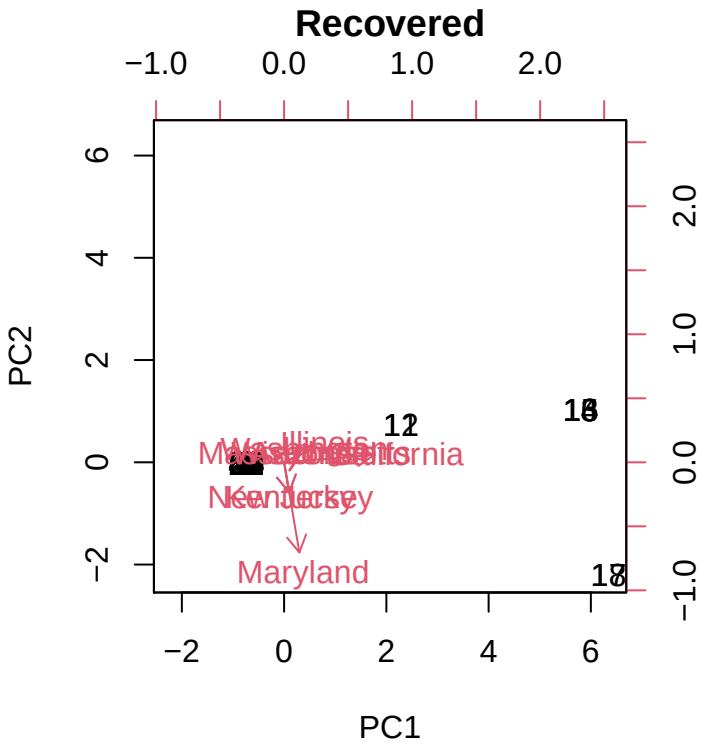



```
pr.var=pr.out$sdev ^2
par(mfrow=c(1,2))
pve=pr.var/sum(pr.var)
plot(pve, xlab="Principal Component ", ylab="Proportion of Variance Explained ", ylim=c(0,1),type="b")
plot(cumsum(pve), xlab="Principal Component ", ylab="Cumulative Proportion of Variance Explained ", yli
     type="b")
```

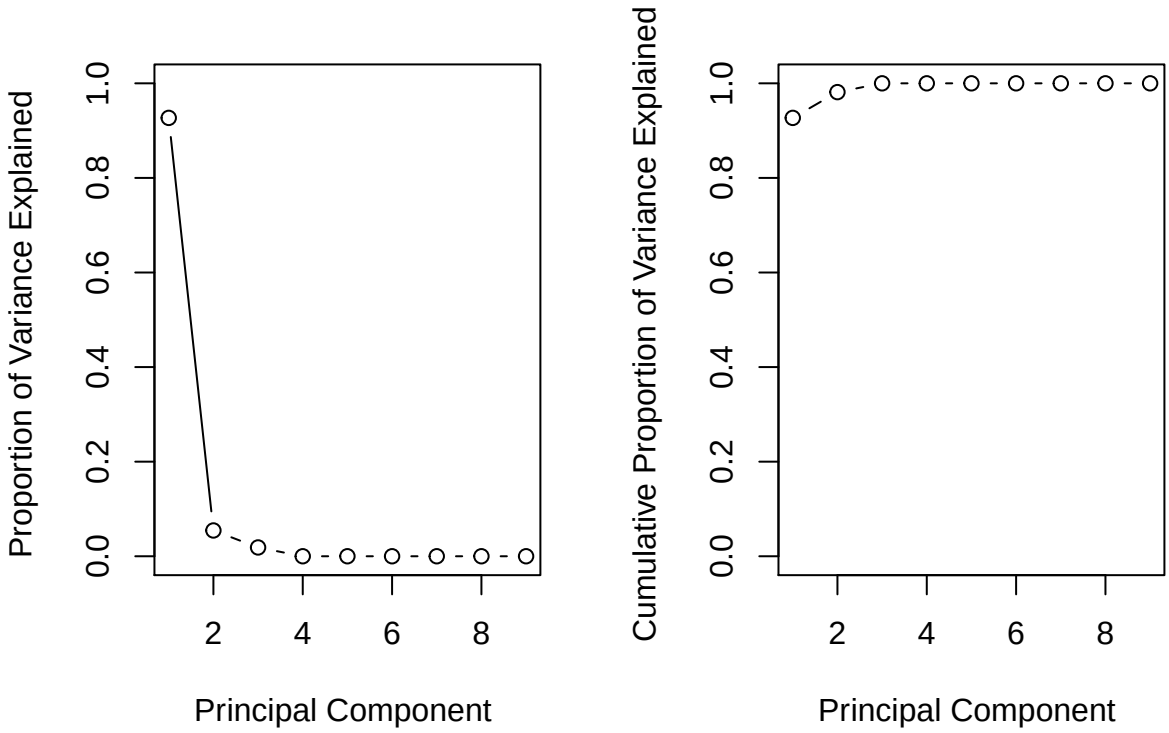


Recovered

```
a<- data.us[data.us$Country.Region=="US",c(2,3,8)]
b <- dcast(a,ObservationDate~Province.State,value.var = "Recovered")
b[is.na(b)] <-0
b <- b[,2:51]
b <- b[,colSums(b)>0]

pr.out=prcomp(b)

biplot(pr.out, scale=0,main = "Recovered")
```




```
pr.var=pr.out$sdev ^2
par(mfrow=c(1,2))
pve=pr.var/sum(pr.var)
plot(pve, xlab="Principal Component ", ylab="Proportion of Variance Explained ", ylim=c(0,1),type="b")
plot(cumsum(pve), xlab="Principal Component ", ylab="Cumulative Proportion of Variance Explained ", yli
     type="b")
```



This results shows that the first PC explains most of the errors, and New York performs different comparing with the rest of the states.

Now, to assess the variables that affect recovery, we will attempt to classify the data using classification trees.

Decision Trees

To estimate recovery trend, create "recover" column. Now if recovery is reported the corresponding row in the column gets a value of 1 otherwise 0.

```
recover <- c(1:1507)
coronavirus=cbind(coronavirus, recover)
```

For simplicity, we are recording the response here as binary. We can see we have 493 recovered cases recorded.

```
coronavirus$recover [coronavirus$type == "recovered"] <- 1
coronavirus$recover [coronavirus$type != "recovered"] <- 0

table(coronavirus$recover)
```

```
##
##    0    1
## 1014  493
```

```
# coronavirus
```

Split data for plotting decision trees

```
# train and test data
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

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

```
set.seed(1)
trainIndex = createDataPartition(coronavirus$recover, p = .80,
                                 list = FALSE)
train.data = coronavirus[ trainIndex, ]
test.data = coronavirus [ -trainIndex, ]
```

We have split the data into train and test with 80:20 ratio. The test data has 301 observations and train has 1206 observations. All data points of recover are converted to factors from int to compute decision trees.

```
dim (train.data)
```

```
## [1] 1206    8
```

```
dim(test.data)
```

```
## [1] 301   8
```

```
train.data$recover = factor(train.data$recover)
test.data$recover = factor(test.data$recover)
```

Decision tree model (Classification Tree)

1st model - Classification tree

```
# build decision tree model
rpart = rpart(recover ~ cases + Country.Region,
    data = train.data,
    method="class", # classification tree
    parms=list(split="information"),
    control=rpart.control(cp = 0.001))

# Generate a textual view of the Decision Tree model.
print(rpart)
```

```
## n= 1206
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 1206 391 0 (0.67578773 0.32421227)
##     2) Country.Region=Australia,Belgium,Canada,Egypt,Germany,Hong Kong,Italy,Japan,Malaysia,Others,Pl
##     3) Country.Region=Cambodia,Finland,France,India,Macau,Mainland China,Nepal,Russia,South Korea,Sri
##       6) cases>=11.5 389 100 0 (0.74293059 0.25706941)
##        12) cases>=1331 14   0 0 (1.00000000 0.00000000) *
##        13) cases< 1331 375 100 0 (0.73333333 0.26666667)
##          26) cases< 270 365  93 0 (0.74520548 0.25479452)
##            52) cases>=63.5 43   4 0 (0.90697674 0.09302326) *
##            53) cases< 63.5 322  89 0 (0.72360248 0.27639752)
##             106) cases< 23.5 161  39 0 (0.75776398 0.24223602) *
##             107) cases>=23.5 161  50 0 (0.68944099 0.31055901)
##               214) cases>=32.5 93  24 0 (0.74193548 0.25806452)
##                 428) cases< 37.5 28   5 0 (0.82142857 0.17857143) *
##                 429) cases>=37.5 65  19 0 (0.70769231 0.29230769)
##                   858) cases>=39.5 57  14 0 (0.75438596 0.24561404) *
##                   859) cases< 39.5 8   3 1 (0.37500000 0.62500000) *
##               215) cases< 32.5 68  26 0 (0.61764706 0.38235294)
##                 430) cases< 31.5 61  22 0 (0.63934426 0.36065574) *
##                 431) cases>=31.5 7   3 1 (0.42857143 0.57142857) *
##          27) cases>=270 10   3 1 (0.30000000 0.70000000) *
##       7) cases< 11.5 679 271 0 (0.60088365 0.39911635)
##        14) Country.Region=France,India,Macau,South Korea,Thailand,UK 52  15 0 (0.71153846 0.28846154)
##          28) cases< 1.5 24   4 0 (0.83333333 0.16666667) *
##          29) cases>=1.5 28  11 0 (0.60714286 0.39285714)
##            58) cases>=4.5 7   1 0 (0.85714286 0.14285714) *
##            59) cases< 4.5 21  10 0 (0.52380952 0.47619048)
##             118) Country.Region=Macau,South Korea,UK 11   4 0 (0.63636364 0.36363636) *
##             119) Country.Region=France,India,Thailand 10   4 1 (0.40000000 0.60000000) *
##        15) Country.Region=Cambodia,Finland,Mainland China,Nepal,Russia,Sri Lanka,United Arab Emirates
##          30) cases>=3.5 308 114 0 (0.62987013 0.37012987) *
##          31) cases< 3.5 319 142 0 (0.55485893 0.44514107)
##            62) Country.Region=Mainland China,United Arab Emirates 312 137 0 (0.56089744 0.43910256) *
##            63) Country.Region=Cambodia,Finland,Nepal,Russia,Sri Lanka 7   2 1 (0.28571429 0.71428571)
```

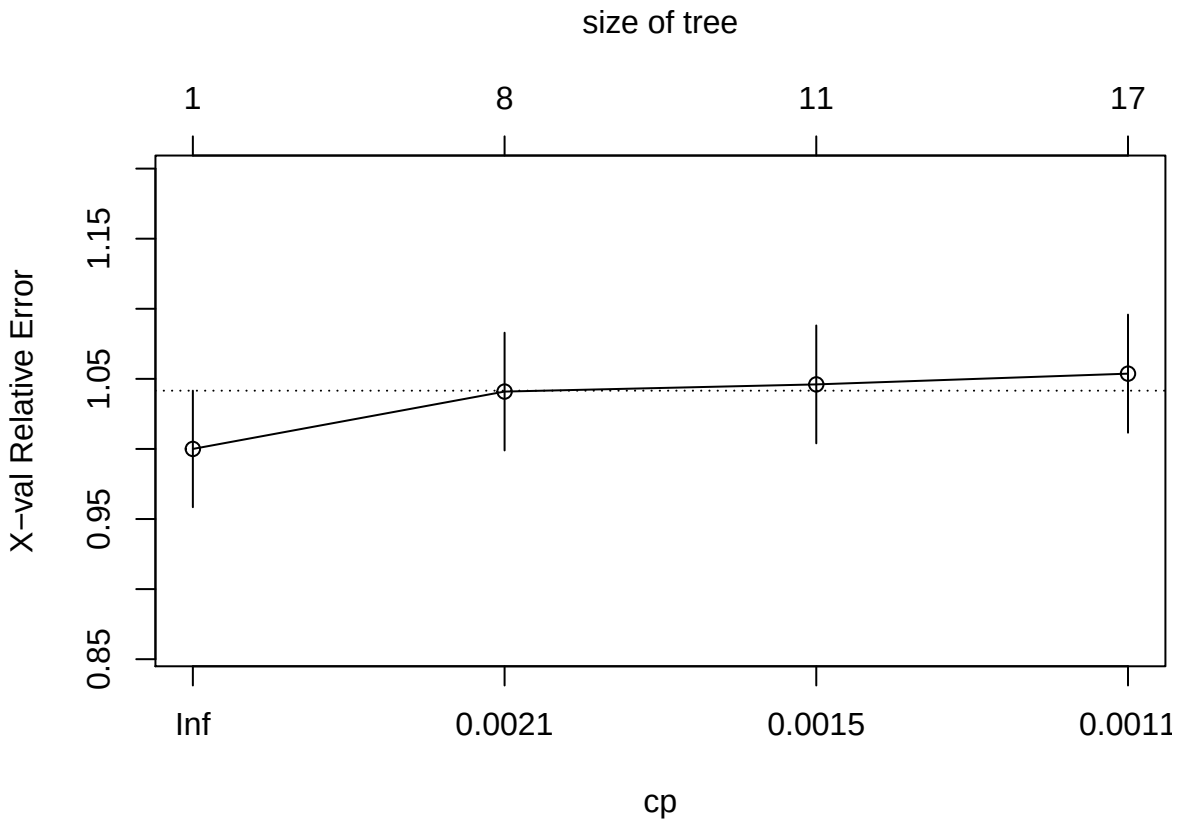
When rpart grows a tree it performs 10-fold cross validation on the data. Use printcp() to see the cross validation results. When we see the variable imporatnce from the tree computed, it makes sense that the

recovery is more dependent on the number of cases and not region wise as the importance is more for number of cases.

```
printcp(rpart)
```

```
##
## Classification tree:
## rpart(formula = recover ~ cases + Country.Region, data = train.data,
##     method = "class", parms = list(split = "information"), control = rpart.control(cp = 0.001))
##
## Variables actually used in tree construction:
## [1] cases          Country.Region
##
## Root node error: 391/1206 = 0.32421
##
## n= 1206
##
##          CP nsplit rel error xerror     xstd
## 1 0.0025575      0   1.00000 1.0000 0.041574
## 2 0.0017050      7   0.98210 1.0409 0.041997
## 3 0.0012788     10   0.97698 1.0460 0.042048
## 4 0.0010000     16   0.96931 1.0537 0.042122
```

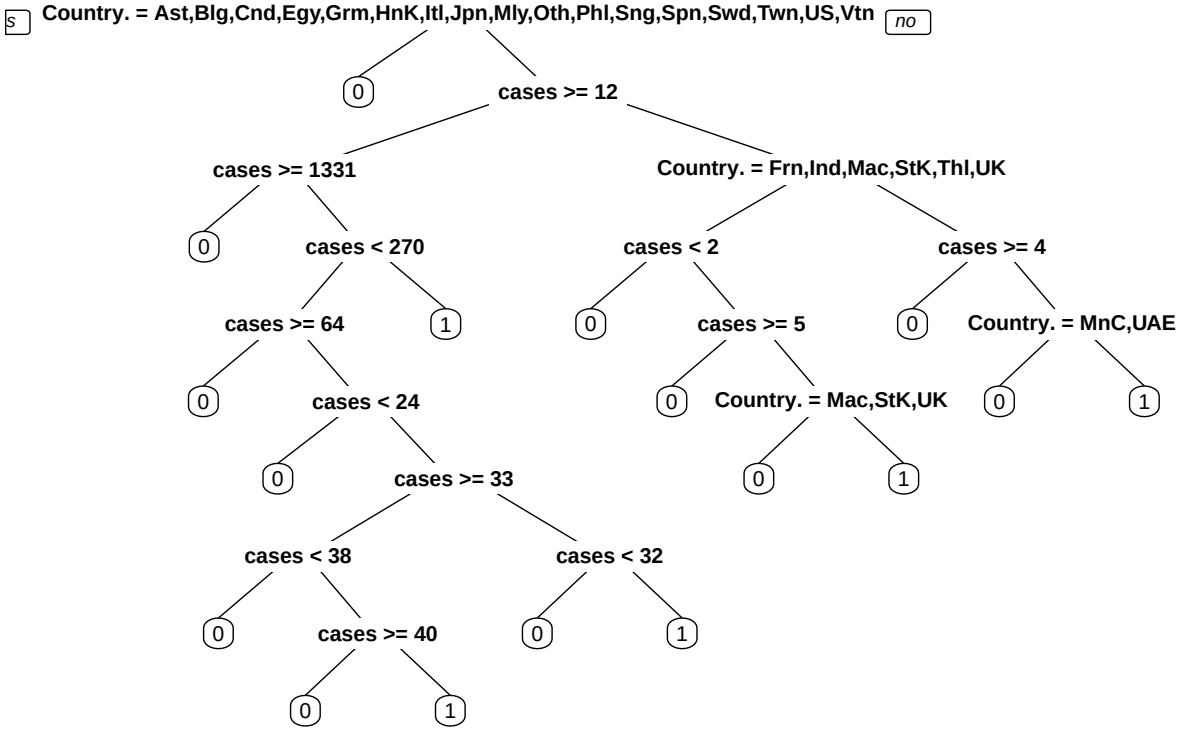
```
# visualize cross-validation results
plotcp(rpart)
```




```
rpart$variable.importance
```

```
##          cases Country.Region
##       34.54142       16.38460
```

Th decision tree plots teels us in countries like Egypt, Canada, US, UK, Germany among others split into 2 branches when the cases are more than or = 12. When the cases are greater than 12, and split into a subtree where it is greater than 1331 cases there is no recovery seen and in such cases the recovery is expected around 300 cases overall. Similarly in countries like India and Japan among others, for instance in UAE, recovery is reported when cases are less than 10.



This is another way of depicting the above tree.

```
dev.new()
fancyRpartPlot(rpart, main="Decision Tree Graph")
```

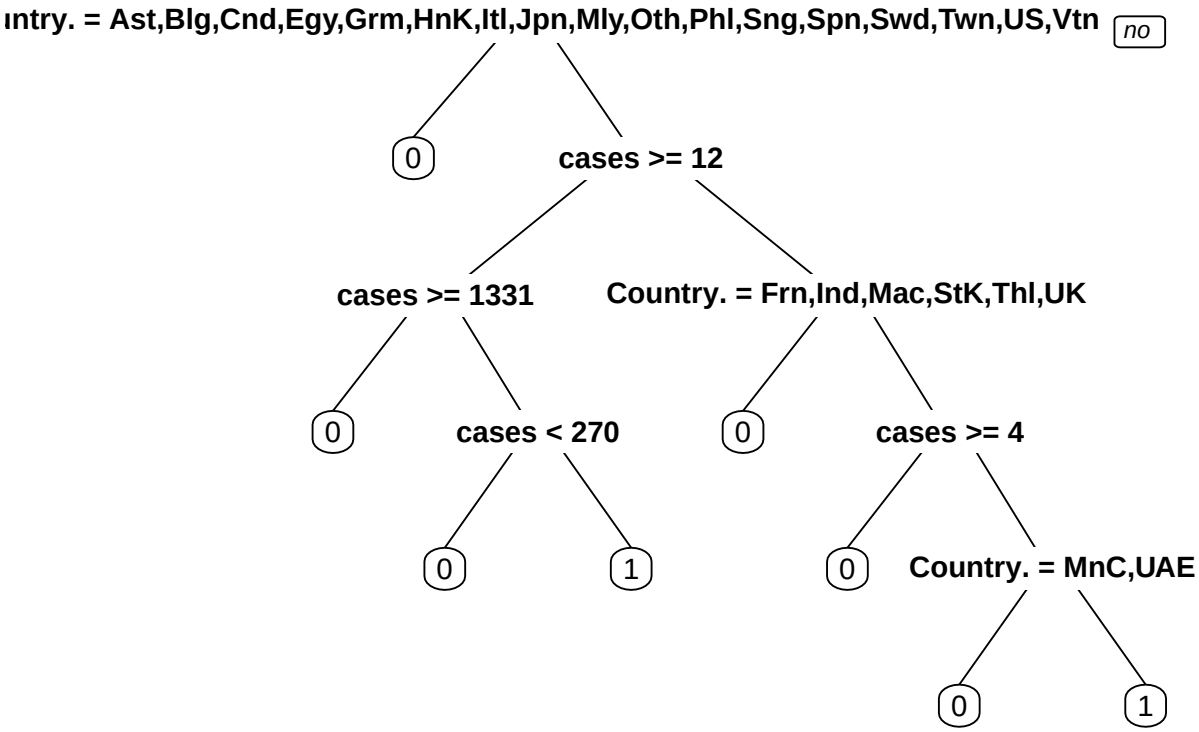
Pruning tree for better model performance and to check if it matches the classification tree we got earlier. The minimum calculated is in sync with the first reading of the cross-validation results. This is calculated based on the cp table which has the least xerror and xstd values and is chosen for pruning. In this case the cp values used for pruning will be

- CP nsplit rel error xerror xstd 0.0019582 0 1.00000 1.0000 0.042211

It's usually a good idea to prune a decision tree. Fully grown trees don't perform well against data not in the training set because they tend to be over-fitted so pruning is used to reduce their complexity by keeping only the most important splits.

Pruning shows that when cases are less than about 300 recovery chances are higher in comparison to otherwise in the counties shown in the tree.

```
# Pruning tree
rpart.prune = prune(rpart, cp = 0.0019582)

prp(rpart.prune)
```



Make predictions and calculate accuracy from confusion matrix. Accuracy = number correct divided by number total instances Accuracy = 67.77%

```
rpart1 = rpart(recover ~ cases + Country.Region,
               data = test.data,
               method="class", # classification tree
               parms=list(split="information"),
               control=rpart.control(cp = 0.001))
rpart.prune1 = prune(rpart1, cp = 0.0019582)
pred.rpart1 = factor(predict(rpart.prune1, type = "class", data = test.data))

# making the confusion matrix for test #
confusionMatrix(test.data$recover, pred.rpart1)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 160  39
##          1  58  44
##
##                Accuracy : 0.6777
##                  95% CI : (0.6217, 0.7302)
##     No Information Rate : 0.7243
##     P-Value [Acc > NIR] : 0.96774
##
##                   Kappa : 0.2466
##
##  Mcnemar's Test P-Value : 0.06761
##
##             Sensitivity : 0.7339
```

```
##             Specificity : 0.5301
##          Pos Pred Value : 0.8040
##          Neg Pred Value : 0.4314
##              Prevalence : 0.7243
##          Detection Rate : 0.5316
##    Detection Prevalence : 0.6611
##       Balanced Accuracy : 0.6320
##
##        'Positive' Class : 0
##
```

Conclusion

This project is timely and important in the way that the pandemic has taken over the world and we are all living in it. This study concludes the following:

Logistic Regression

The effect of age and gender on death was tested using a logistic regression model on the individual case data.

We observe that both gender and age are significant predictors of death due to coronavirus. Men seem 2.314 more likely to die from infection and for every one-year increase in age a person is about 1.07 times more likely.

Non-Linear Regression

We did the non-linear regression to the data, and it turns out not all the states have the same trend. some of them had peaks and the curve is falling down, while others are still climbing up.

We are more concerned about the states most recent performance and compared the correlation of the standardized confirmed cases of recent 30 days- which is at the right of the red vertical dotted line. If the correlation is negative, it means the recent trend of confirmed case is declining; if it is positive, it means there is a trend of increasing in the recent 30 days.

It shows that Montana, Alaska, Vermont, Hawaii, and New York are having a trend of declining in new confirmed cases, as their correlation are all smaller than -0.8. For example, among them Montana has the smallest correlation which is -0.96, and that of New York is -0.81. Those states all shows that they have passed the peak of the curve and now the new confirmed cases are falling down.

On the other hand, the states that have fastest increasing new confirmed cases trend are: Virginia, Iowa, Mississippi, Illinois, Nebraska, Kansas, North Carolina, Maryland, Rhode Island, New Mexico and Minnesota, and they have correlation of more than 0.8. For example, the largest correlation is from Virginia which is 0.97, and we could see from the non-linear curve as well that the new confirmed cases are increasing sharply recently. The increase of confirmed cases might due to the increase of the test ability recently or other reasons, and the data is telling us that the situations in those states need some attention because of their recent increase of new confirmed cases.

Principal Component Analysis on the coronavirus package

The result seem good in separating some of the countries or areas, but we have to disregard the date variable for PCA.

For the PCA of world wide by countries. It shows similar trends in confirmed number of cases, deaths and recovered. The U.S. performs differently than the rest of the countries in the first PC. Canada, France and Netherlands performs differently than the rest of the countries in the second PC.

For the PCA of the U.S., it shows in the confirmed case numbers and deaths numbers, the first PC explains most of the errors, and New York performs different comparing with the rest of the states. In the recovered data, Maryland performs different, following by New Jersy and Kansas.

Decision trees

We try to predict if the recovery cases trend in terms of greater than the other, is based on number of cases in country or the region of infection.

The decision tree plots shows us in countries like Egypt, Canada, US, UK, Germany among others split into 2 branches when the cases are more than or = 12. When the cases are greater than 12, and split into a subtree where it is greater than 1331 cases there is no recovery seen and in such cases the recovery is expected around 300 cases overall. Similarly in countries like India and Japan among others, for instance in UAE, recovery is reported when cases are less than 10.

Confusion matrix and statistics related to the tree performance show us that the accuracy of the classification tree plotted is about 67.7% and Cohen's kappa value of 0.246, with a sensitivity of 73.39%. Overall the classification tree is a good metric to predict recovery cases and the lower the number of cases in a country, better the possibility of patients recovery.