Assignment9

samriddh Gupta

libraries

```
library(countrycode)
```

```
## Warning: package 'countrycode' was built under R version 3.6.3
```

```
library(tidyverse)
```

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

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 3.6.3

## Warning: package 'readr' was built under R version 3.6.3

## Warning: package 'dplyr' was built under R version 3.6.3

## Warning: package 'stringr' was built under R version 3.6.3

## Warning: package 'forcats' was built under R version 3.6.3

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(lubridate)
```

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

## The following object is masked from 'package:base':
##
##     date
```

```
theme_set(theme_bw())

dat <-
  read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties.csv")
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

```
deaths_by_state <- dat %>%
  group_by(state,date) %>%
  summarize(total_deaths = sum(deaths)) %>%
  ungroup() %>%
  filter(date == max(date)) %>%
  arrange(desc(total_deaths))
deaths_by_state
```

```
## # A tibble: 55 x 3
##    state         date       total_deaths
##    <chr>         <date>            <dbl>
##  1 New York      2020-04-12         9385
##  2 New Jersey    2020-04-12         2350
##  3 Michigan      2020-04-12         1488
##  4 Louisiana     2020-04-12          840
##  5 Massachusetts 2020-04-12          756
##  6 Illinois      2020-04-12          727
##  7 California    2020-04-12          676
##  8 Connecticut   2020-04-12          554
##  9 Pennsylvania  2020-04-12          525
## 10 Washington    2020-04-12          511
## # ... with 45 more rows
```

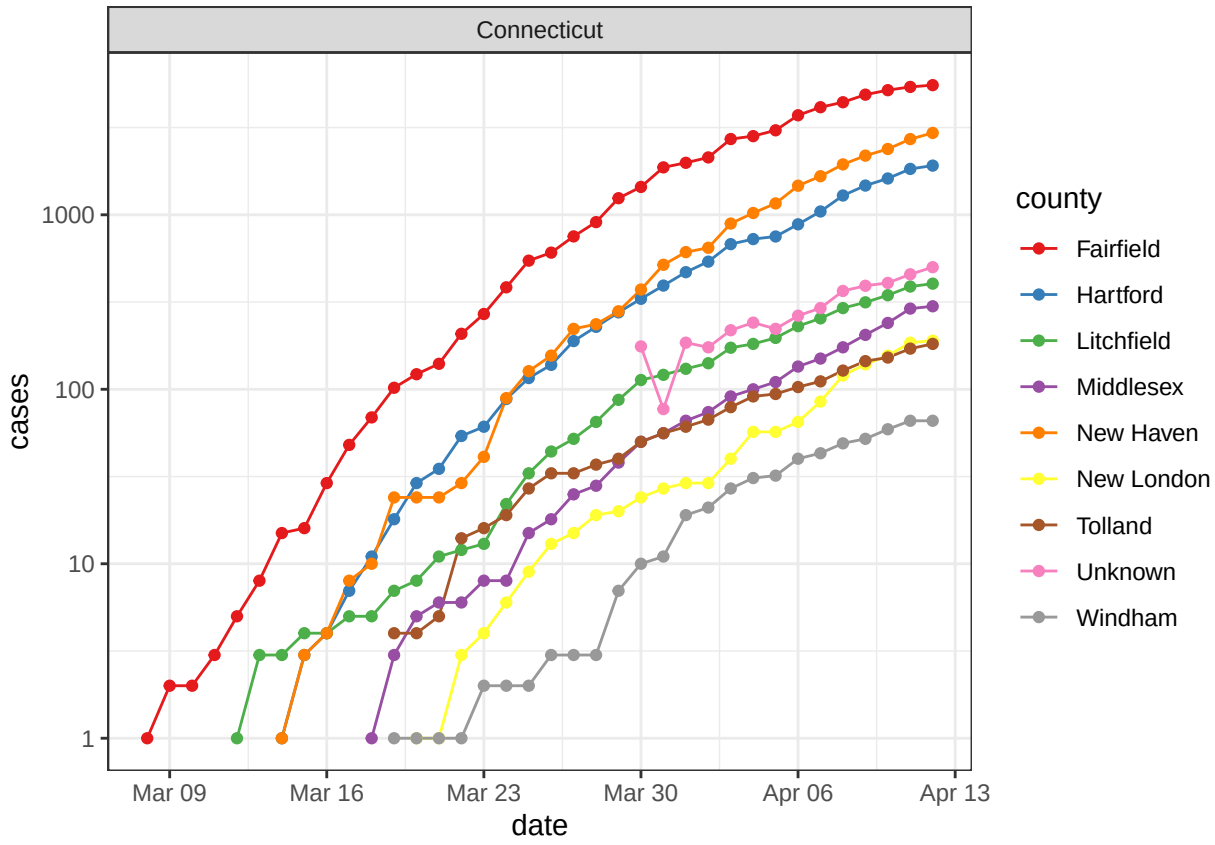
EX1

```
deaths_by_date<-dat %>%
  group_by(state,date) %>%
  summarize(total_deaths = sum(deaths),
            Total_cases=sum(cases)) %>%
  arrange(desc(Total_cases))
deaths_by_date
```

```
## # A tibble: 2,269 x 4
## # Groups:   state [55]
##    state    date       total_deaths Total_cases
```

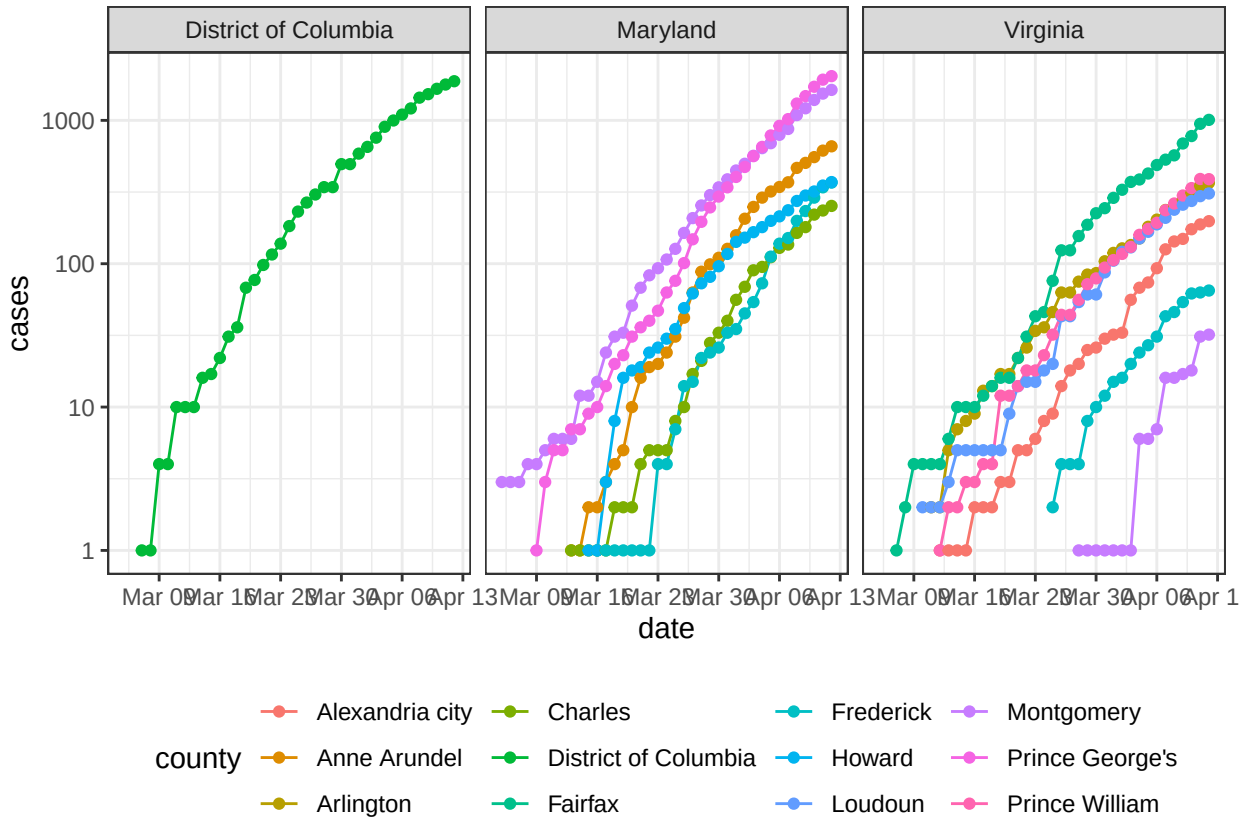
```
##    <chr>    <date>            <dbl>       <dbl>
##  1 New York 2020-04-12         9385      188694
##  2 New York 2020-04-11         8627      180458
##  3 New York 2020-04-10         7844      170512
##  4 New York 2020-04-09         7067      159937
##  5 New York 2020-04-08         6268      149401
##  6 New York 2020-04-07         5563      140081
##  7 New York 2020-04-06         5505      130703
##  8 New York 2020-04-05         4161      122911
##  9 New York 2020-04-04         3568      114996
## 10 New York 2020-04-03         2935      102945
## # ... with 2,259 more rows
```

```
dat_small <-
  dat %>%
  filter(state %in% c("South Carolina", "Connecticut")) %>%
  mutate(county = factor(county))

dat_small %>%
  filter(state == "Connecticut") %>%
  ggplot(aes(x = date, y = cases, group = county, col = county)) +
  geom_line() +
  geom_point() +
  facet_wrap(~ state) +
  scale_y_log10() +
  scale_color_brewer(palette = "Set1")
```



Ex2

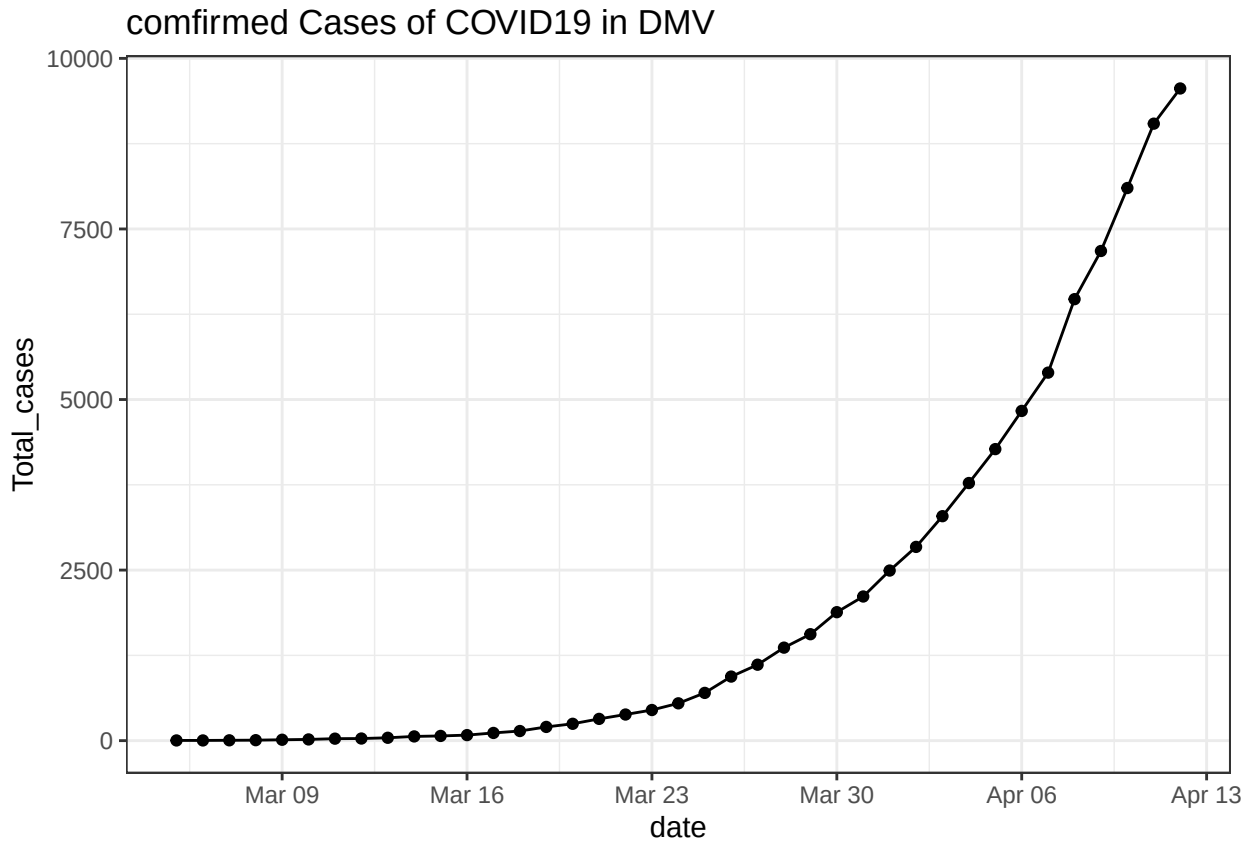
```
dat_small <-
  dat %>%
  filter(state %in% c("District of Columbia", "Maryland","Virginia")) %>%
  filter(county %in% c("Anne Arundel","Charles","Alexandria city","District of Columbia","Frederick","H
  mutate(county = factor(county),
         date = as.Date(date,format = "%Y-%m-%d"))

dat_small %>%
  ggplot(aes(x = date, y = cases, group = county, col = county)) +
  geom_line() +
  geom_point() +
  facet_wrap(~ state) +
  scale_y_log10() +
  theme(legend.position = "bottom")
```



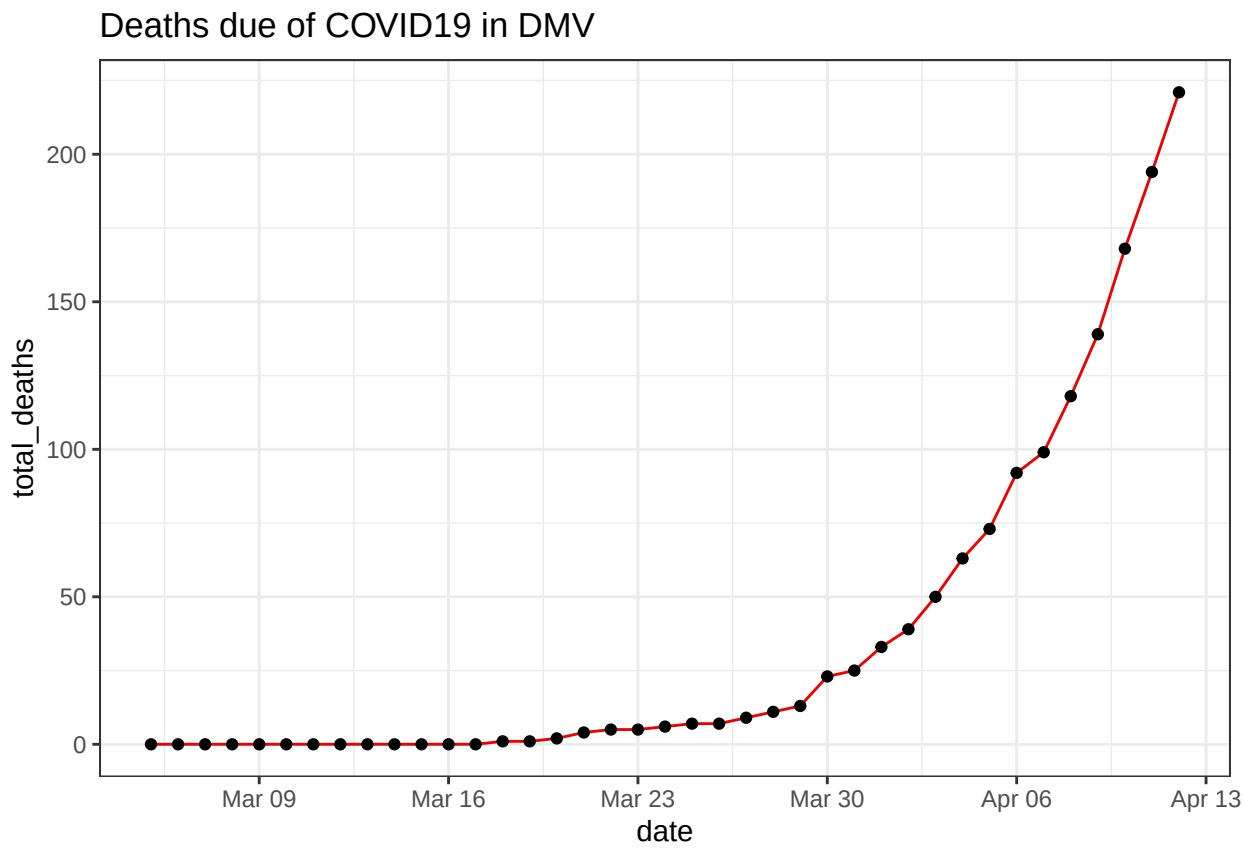
EX3

```
DMV<-dat_small %>%
  group_by(date) %>%
  summarize(total_deaths = sum(deaths),
```

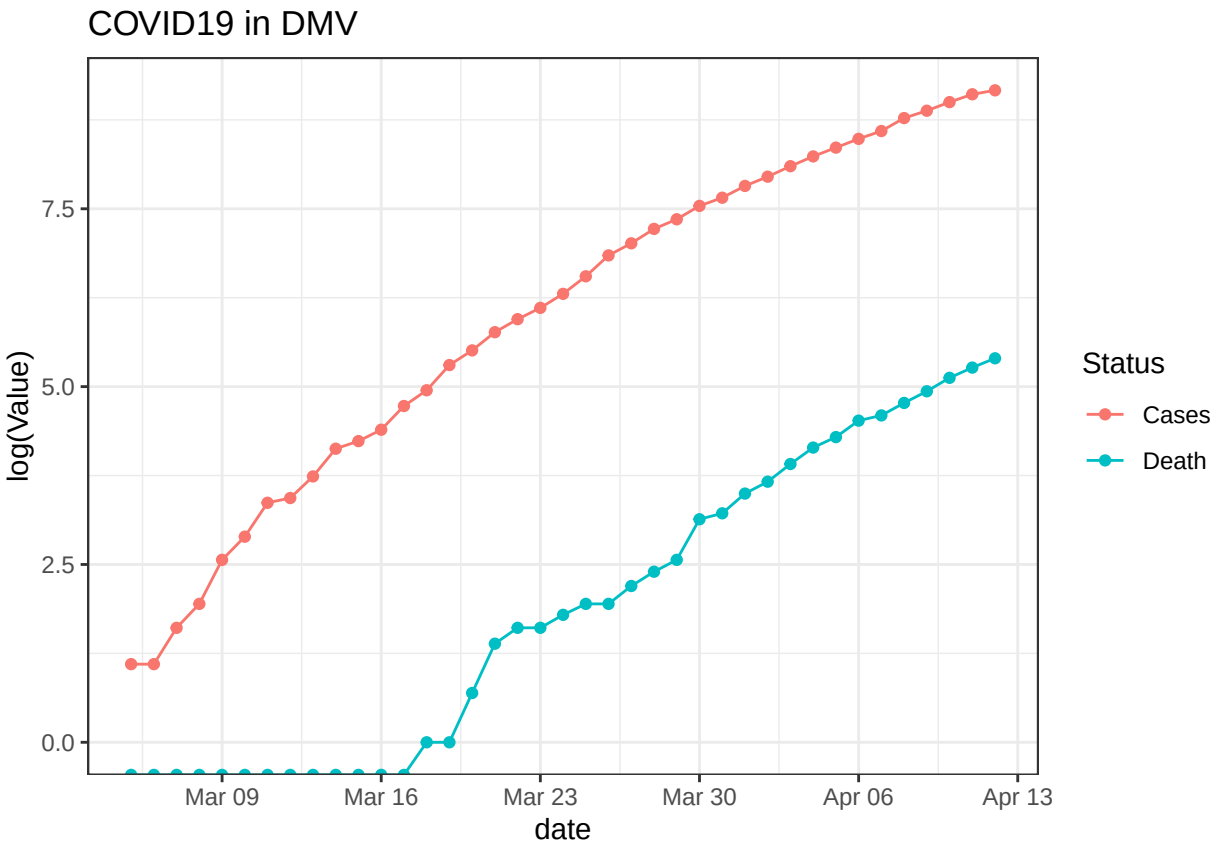
```
            Total_cases=sum(cases))
ggplot(DMV,aes(x=date,y=Total_cases))+
  geom_line()+
  geom_point()+
  ggtitle("comfirmed Cases of COVID19 in DMV ")
```




```
ggplot(DMV,aes(x=date,y=total_deaths))+
  geom_line(color="#EE0000")+
  geom_point()+
  ggtitle("Deaths due of COVID19 in DMV ")
```




```
DMV<-DMV %>%
  mutate(Cases=Total_cases,
         Death=total_deaths) %>%
  select(date,Cases,Death) %>%
  pivot_longer(-date,names_to = "Status",values_to = "Value")

ggplot(DMV,aes(x=date,y=log(Value),color=Status))+
  geom_line()+
  geom_point()+
  ggtitle("COVID19 in DMV ")
```



EX4

```
dat <-
  read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/archived_data/archived_tim
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Province/State` = col_character(),
##   `Country/Region` = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
new_dat<-dat %>%
  pivot_longer(-c(`Province/State`,`Country/Region`,Lat,Long),names_to = "date",values_to = "Confirmed_
```

```
new_dat$`Country/State`= paste(new_dat$`Province/State`,new_dat$`Country/Region`,sep = ",")
new_dat
```

```
## # A tibble: 31,062 x 7
##    `Province/State` `Country/Region`   Lat  Long date  Confirmed_Cases
```

```
##    <chr>            <chr>            <dbl> <dbl> <chr>           <dbl>
##  1 <NA>             Thailand            15   101 1/22~               2
##  2 <NA>             Thailand            15   101 1/23~               3
##  3 <NA>             Thailand            15   101 1/24~               5
##  4 <NA>             Thailand            15   101 1/25~               7
##  5 <NA>             Thailand            15   101 1/26~               8
##  6 <NA>             Thailand            15   101 1/27~               8
##  7 <NA>             Thailand            15   101 1/28~              14
##  8 <NA>             Thailand            15   101 1/29~              14
##  9 <NA>             Thailand            15   101 1/30~              14
## 10 <NA>             Thailand            15   101 1/31~              19
## # ... with 31,052 more rows, and 1 more variable: `Country/State` <chr>
```

Ex5

```
new_dat1<-new_dat %>%
  mutate(continent=countrycode(sourcevar = `Country/Region`,
                               origin = "country.name",
                               destination = "continent")) %>%
    mutate(continent=case_when(`Country/Region`=="South Yemen (former)"~"Asia",
                               `Country/Region`=="Akrotiri and Dhekelia"~"Europe",
                               `Country/Region`=="Central African Rep."~"Africa",
                               `Country/Region`=="Channel Islands"~"Europe",
                               `Country/Region`=="Cocos Island"~"Asia",
                               `Country/Region`=="Czechoslovakia"~"Europe",
                               `Country/Region`=="East Germany"~"Europe",
                               `Country/Region`=="Eritrea and Ethiopia"~"Africa",
                               `Country/Region`=="Kosovo"~"Europe",
                               `Country/Region`=="North Yemen (former)"~"Asia",
                               `Country/Region`=="North Yemen (former)"~"Americas",
                               `Country/Region`=="St. Martin"~"Asia",
                               `Country/Region`=="Yugoslavia"~"Europe",
                               `Country/Region`=="Serbia and Montenegro"~"Asia",
                               TRUE~continent))
```

```
## Warning in countrycode(sourcevar = `Country/Region`, origin = "country.name", : Some values were not
```

```
new_dat1
```

```
## # A tibble: 31,062 x 8
##    `Province/State` `Country/Region`   Lat  Long date  Confirmed_Cases
##    <chr>            <chr>            <dbl> <dbl> <chr>           <dbl>
##  1 <NA>             Thailand            15   101 1/22~               2
##  2 <NA>             Thailand            15   101 1/23~               3
##  3 <NA>             Thailand            15   101 1/24~               5
##  4 <NA>             Thailand            15   101 1/25~               7
##  5 <NA>             Thailand            15   101 1/26~               8
##  6 <NA>             Thailand            15   101 1/27~               8
##  7 <NA>             Thailand            15   101 1/28~              14
##  8 <NA>             Thailand            15   101 1/29~              14
##  9 <NA>             Thailand            15   101 1/30~              14
```

```
## 10 <NA>              Thailand            15   101 1/31~              19
## # ... with 31,052 more rows, and 2 more variables: `Country/State` <chr>,
## #   continent <chr>
```

EX6

```
new_dat2<-new_dat1 %>%
  select(`Country/Region`,Confirmed_Cases,`Country/State`) %>%
  group_by(`Country/State`) %>%
  summarize(Confirmed_Cases_sum=sum(Confirmed_Cases)) %>%
  ungroup() %>%
  arrange(desc(Confirmed_Cases_sum))



new_dat2<-head(new_dat2,25)
new_dat2
```

```
## # A tibble: 25 x 2
##    `Country/State` Confirmed_Cases_sum
##    <chr>                         <dbl>
##  1 Hubei,China                 2894885
##  2 NA,Italy                     497959
##  3 NA,Iran                      252770
##  4 NA,Spain                     186200
##  5 NA,Korea, South              181699
##  6 NA,Germany                   160974
##  7 France,France                117724
##  8 Guangdong,China               67015
##  9 New York,US                   64538
## 10 Henan,China                   61811
## # ... with 15 more rows
```

```
country<-new_dat2$`Country/State`
```

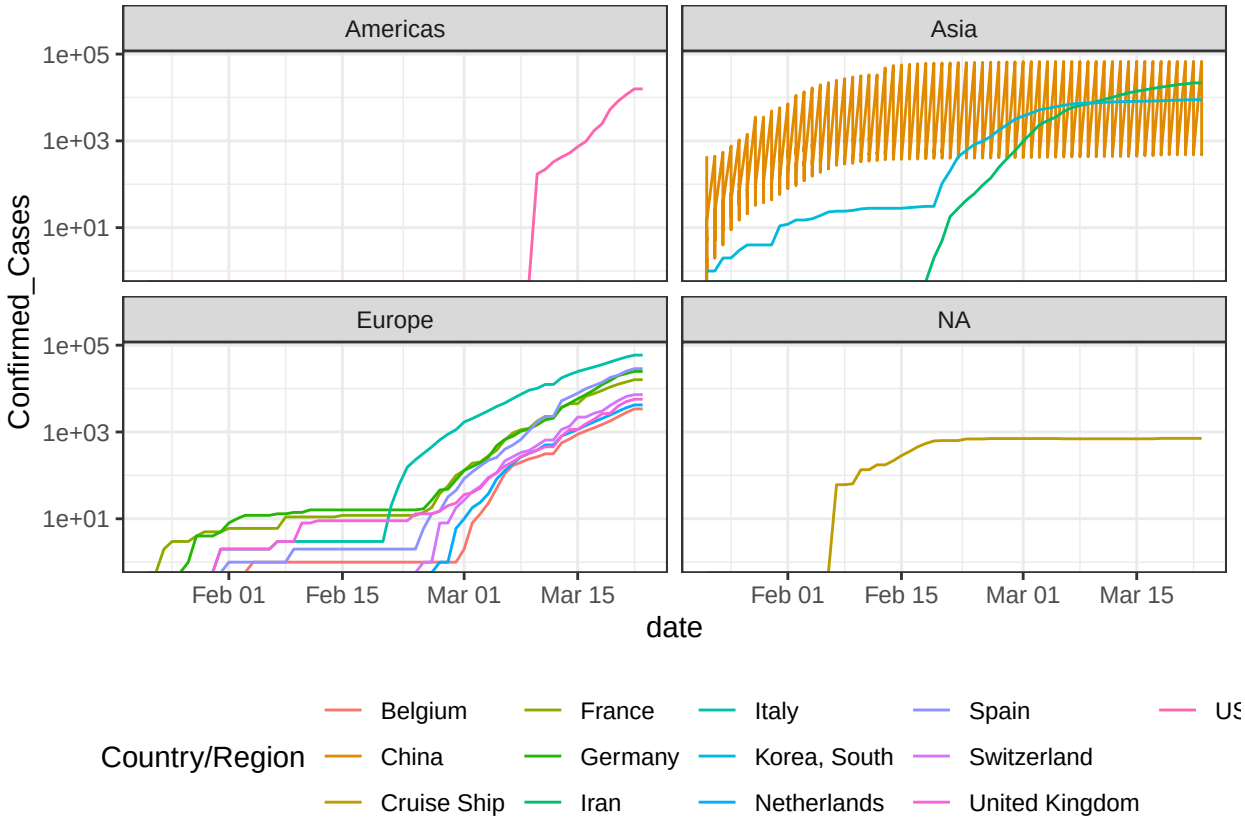
EX7

```
new_dat3<-new_dat %>%
  mutate(continent=countrycode(sourcevar = `Country/Region`,
                               origin = "country.name",
                               destination = "continent")) %>%
  mutate(continent=case_when(`Country/Region`=="Kosovo"~"Europe",
                             TRUE~continent),
         date=as.Date(date,format = "%m/%d/%Y")) %>%
  filter(`Country/State` %in% country)
```

```
## Warning in countrycode(sourcevar = `Country/Region`, origin = "country.name", : Some values were not
```

```
new_dat3 %>%
  ggplot(aes(x = date, y = Confirmed_Cases,color=`Country/Region`)) +
  geom_line() +
  facet_wrap(~ continent) +
  scale_y_log10() +
  theme(legend.position = "bottom")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```



EX8

```
data<-new_dat1 %>%
  filter(`Province/State`=="New York") %>%
  select(date,Confirmed_Cases) %>%
  mutate(date=as.Date(date,format = "%m/%d/%y"))

data1<-deaths_by_date %>%
  select(date,state,Total_cases) %>%
  filter(state=="New York") %>%
  mutate(date=as.Date(date,format = "%Y-%m-%d")) %>%
  select(date,Total_cases)
```

```
## Adding missing grouping variables: `state`
```

```
data3<-full_join(data,data1,all=T)
```

```
## Joining, by = "date"
```

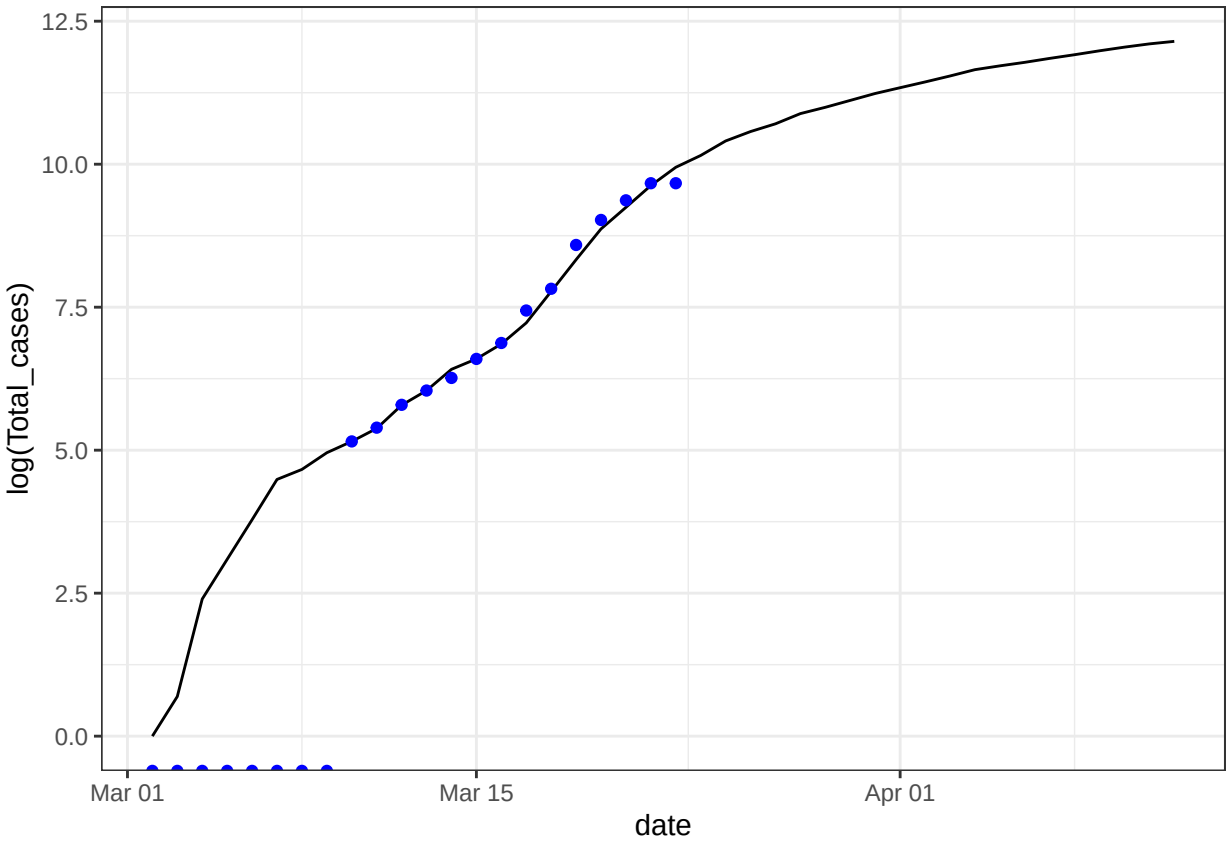
```
data3
```

```
## # A tibble: 82 x 4
##    date       Confirmed_Cases state Total_cases
##    <date>               <dbl> <chr>       <dbl>
##  1 2020-01-22               0 <NA>           NA
##  2 2020-01-23               0 <NA>           NA
##  3 2020-01-24               0 <NA>           NA
##  4 2020-01-25               0 <NA>           NA
##  5 2020-01-26               0 <NA>           NA
##  6 2020-01-27               0 <NA>           NA
##  7 2020-01-28               0 <NA>           NA
##  8 2020-01-29               0 <NA>           NA
##  9 2020-01-30               0 <NA>           NA
## 10 2020-01-31               0 <NA>           NA
## # ... with 72 more rows
```

I think that there is change in number because the tells today number of positive cases in first part and second one tells the total cases after people are recovered.

```
data3 %>%
  filter(date>ymd(20200301)) %>%
  ggplot(aes(x=date,y=log(Total_cases)))+
  geom_line()+
  geom_point(aes(y=log(Confirmed_Cases)),color="blue")
```

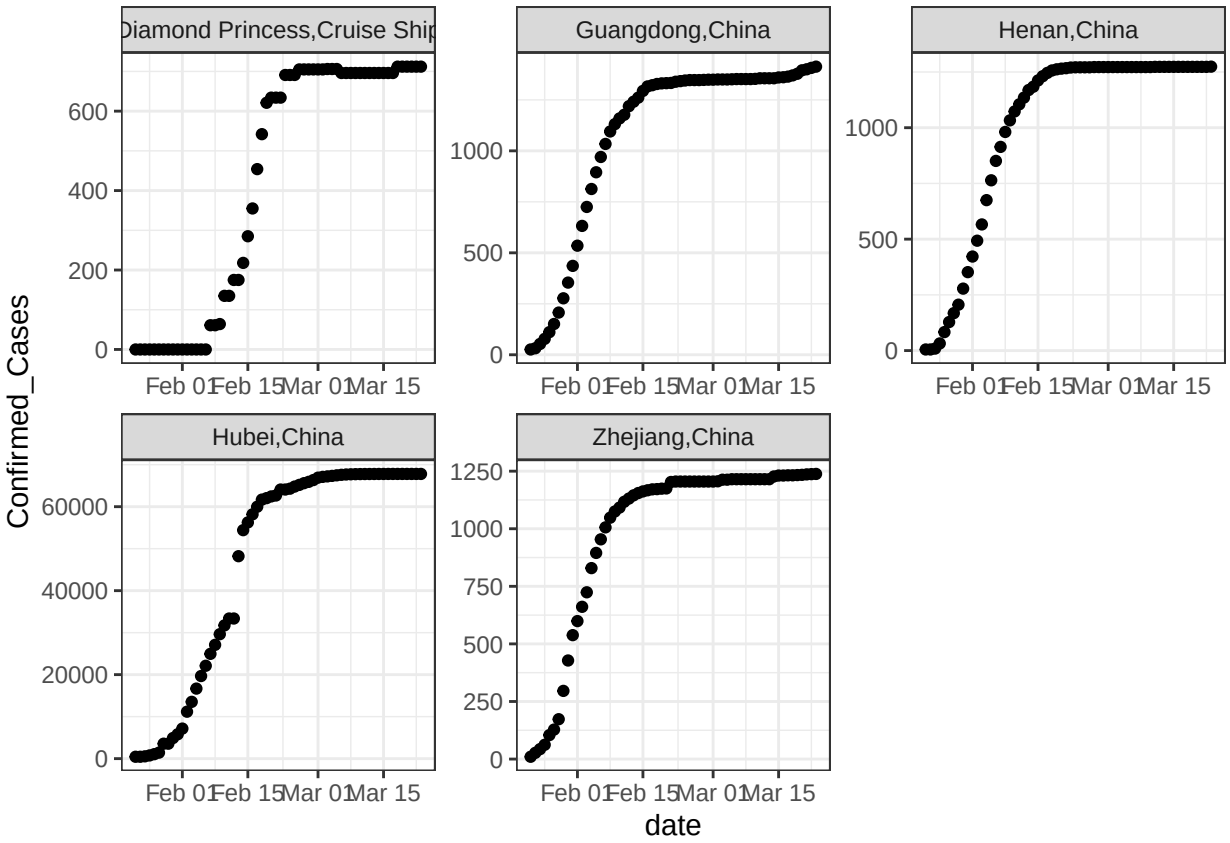
```
## Warning: Removed 20 rows containing missing values (geom_point).
```



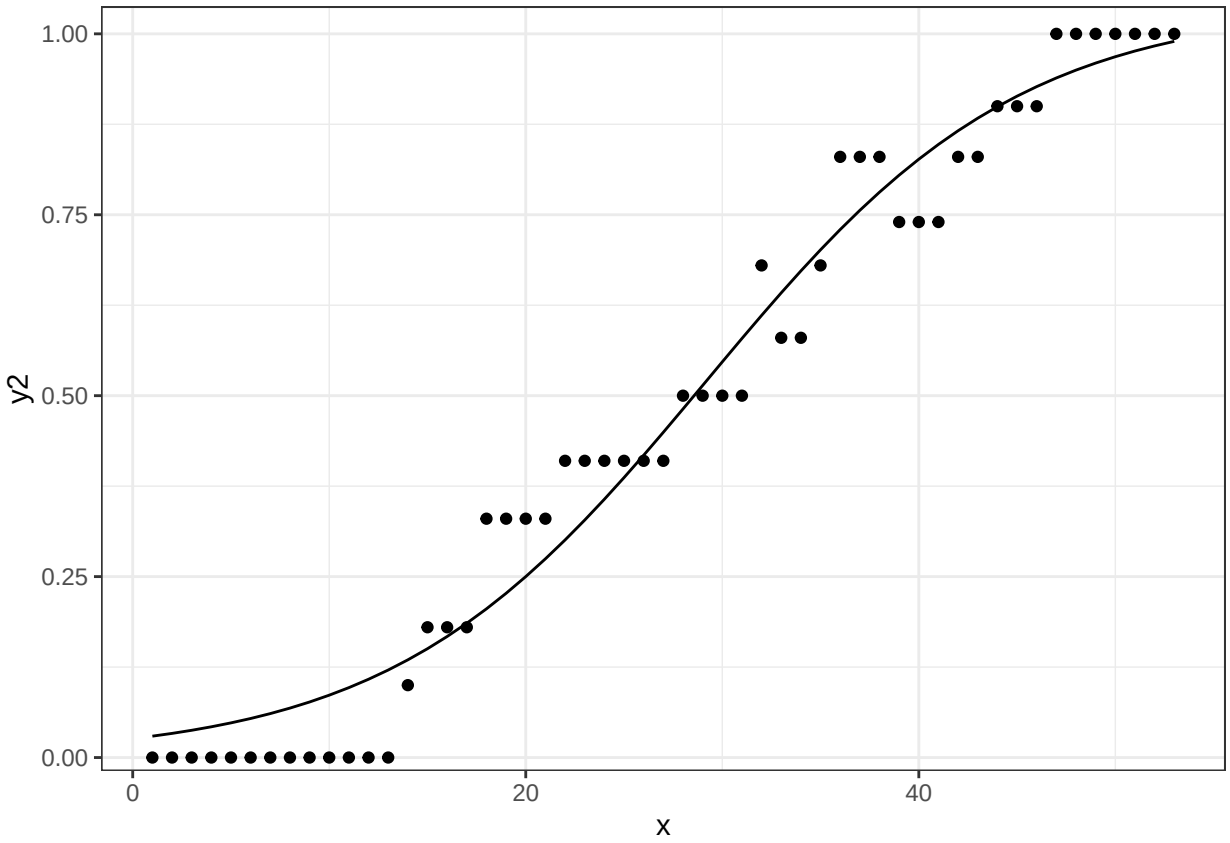

```
  ggtitle("Deaths due of COVID19 in DMV ")
```

```
## $title
## [1] "Deaths due of COVID19 in DMV "
##
## attr(,"class")
## [1] "labels"
```

```
Slowed_cases <- new_dat1 %>%
  filter(`Country/State` %in% c("Hubei,China", "Diamond Princess,Cruise Ship","Guangdong,China","Henan,C
  mutate(date=as.Date(date,format = "%m/%d/%y"))

Slowed_cases %>%
  ggplot(aes(x = date, y = Confirmed_Cases)) +
    geom_point() +
  facet_wrap(~ `Country/State`, scales = "free")
```



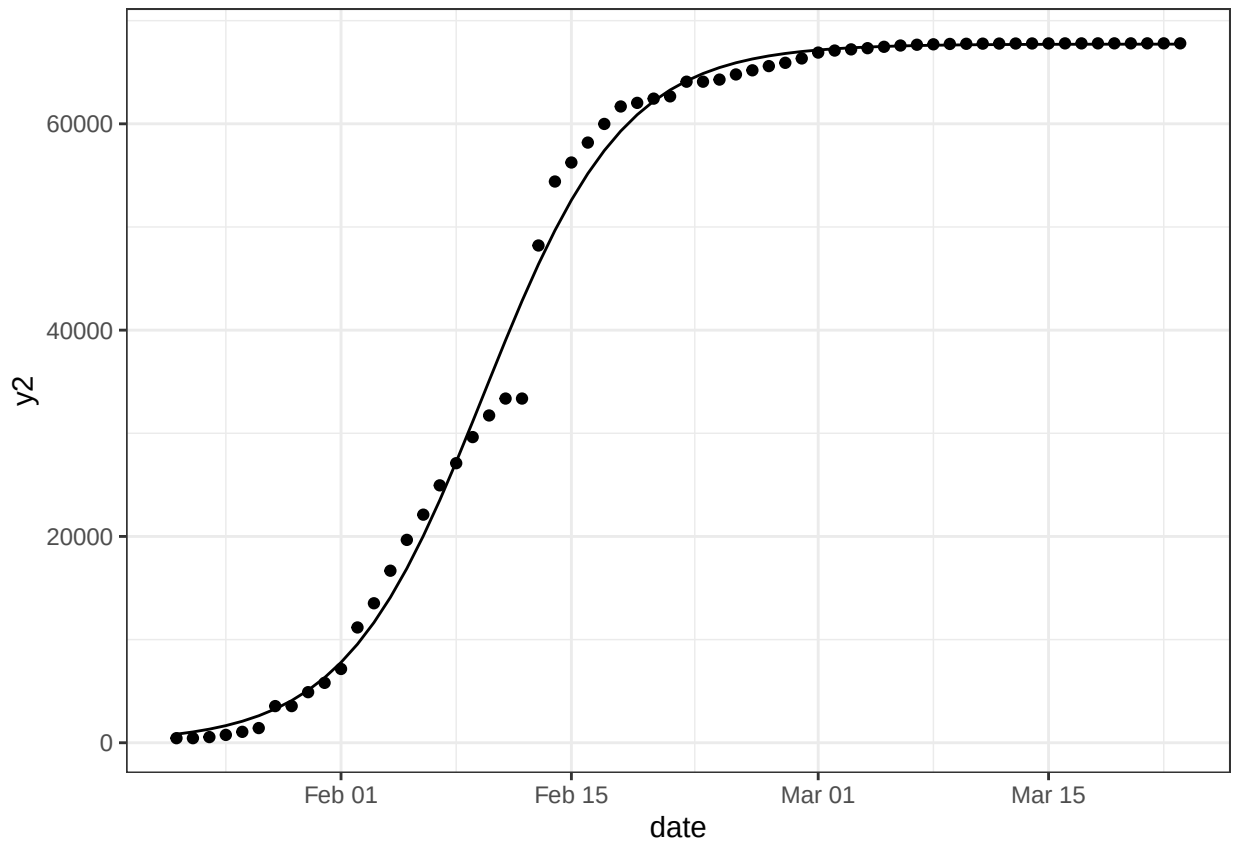

```
# function needed for visualization purposes
sigmoid = function(x, params) {
  params[1] / (1 + exp(-params[2] * (x - params[3])))
}

x = 1:53
y = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0.1,0.18,0.18,0.18,0.33,0.33,0.33,0.33,0.41,
  0.41,0.41,0.41,0.41,0.41,0.5,0.5,0.5,0.5,0.68,0.58,0.58,0.68,0.83,0.83,0.83,
  0.74,0.74,0.74,0.83,0.83,0.9,0.9,0.9,1,1,1,1,1,1,1)
df <- tibble(x = x, y = y)
# fitting code
fitmodel <- nls(y ~ a /(1 + exp(-b * (x - c))), data = df,
                start = list(a = 1, b = 0.5, c = 25))

# visualization code
# get the coefficients using the coef function
params=coef(fitmodel)

df$y2 <- sigmoid(x, params)
df %>% ggplot(aes(x, y2)) + geom_line() + geom_point(y = y)
```



EX10

```
sigmoid = function(x, params) {
  params[1] / (1 + exp(-params[2] * (x - params[3])))
}

Slowed_cases <- new_dat1 %>%
  filter(`Country/State` %in% c("Hubei,China"))%>%
  mutate(date=as.Date(date,format = "%m/%d/%y"),
         date_int=unclass(date))


startK <- max(Slowed_cases$Confirmed_Cases)
fitmodel <-nls(Confirmed_Cases ~ K /( (1 +
                                        exp(-B * (date_int - t0))) ) ,
                data = Slowed_cases,
                start = list(K = startK,  B = .25, t0 = 18300),
                control = list(maxiter = 1000, warnOnly = TRUE))

params=coef(fitmodel)

Slowed_cases$y2 <- sigmoid(Slowed_cases$date_int, params)

ggplot(Slowed_cases,aes(date, y2)) +
```

```
  geom_line() +
  geom_point(aes(y = Confirmed_Cases))
```




```
summary(fitmodel)
```

```
##
## Formula: Confirmed_Cases ~ K/((1 + exp(-B * (date_int - t0))))
##
## Parameters:
##     Estimate Std. Error   t value Pr(>|t|)
## K  6.773e+04  3.839e+02    176.44   <2e-16 ***
## B  2.348e-01  7.916e-03     29.66   <2e-16 ***
## t0 1.830e+04  1.660e-01 110241.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2002 on 59 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 8.777e-07
```

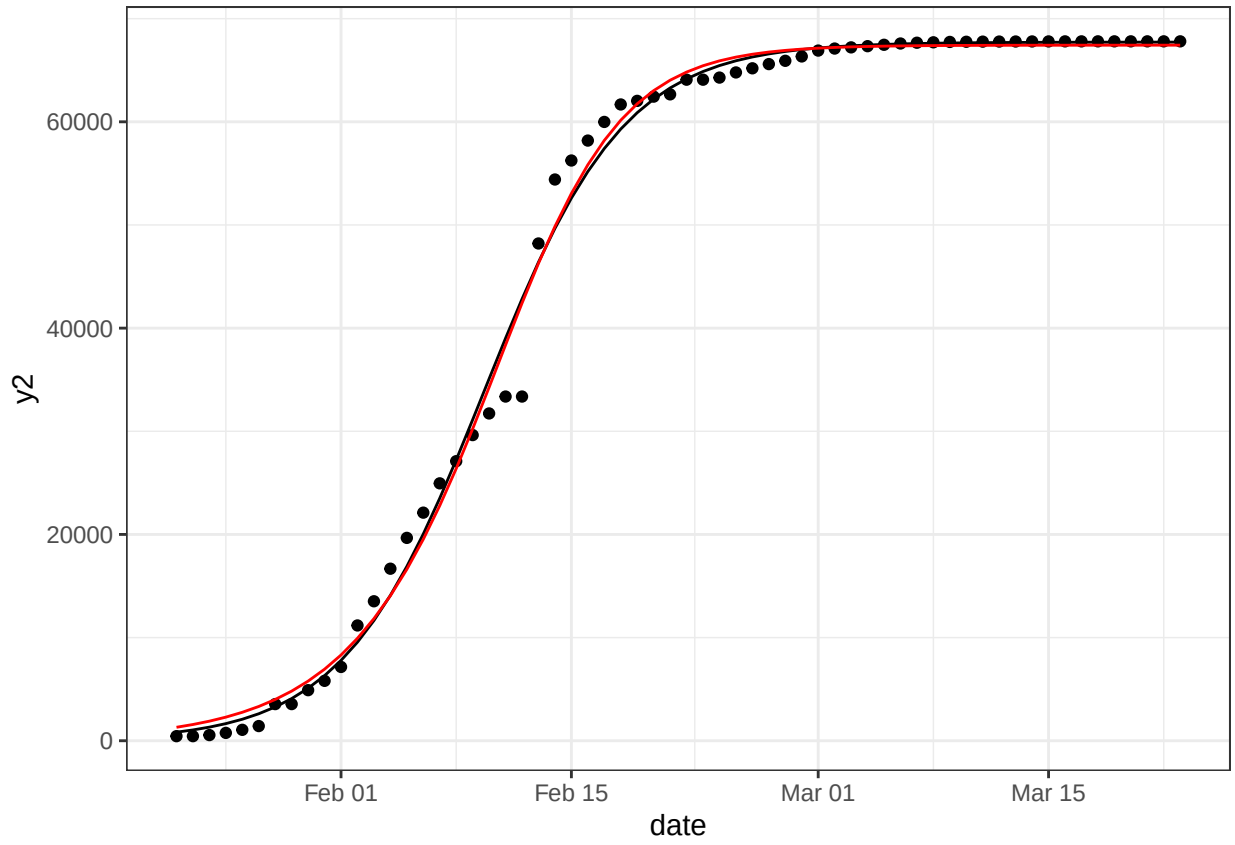
```
broom::glance(fitmodel)
```

```
## # A tibble: 1 x 8
```

```
##   sigma isConv      finTol logLik   AIC   BIC   deviance df.residual
##   <dbl> <lgl>        <dbl>  <dbl> <dbl> <dbl>      <dbl>       <int>
## 1 2002. TRUE   0.000000878  -558. 1123. 1132. 236424289.          59
```

EX11

```
sigmoid = function(x, params) {
  params[1] / (1 + exp(-params[2] * (x - params[3])))^(1/params[4])
}

startK <- max(Slowed_cases$Confirmed_Cases)
fitmodel <-nls(Confirmed_Cases ~ K /(1 +exp(-B * (date_int - t0)))^(1/v),
               data = Slowed_cases,
               start = list(K = startK,  B = .25, t0 = 18300,v=1),
               control = list(maxiter = 1000, warnOnly = TRUE))

params=coef(fitmodel)

Slowed_cases$y3 <- sigmoid(Slowed_cases$date_int, params)

ggplot(Slowed_cases,aes(date, y2)) +
  geom_line() +
  geom_point(aes(y = Confirmed_Cases))+
  geom_line(aes(y = y3),color="red")
```



```r
summary(fitmodel)
```

```
##
## Formula: Confirmed_Cases ~ K/(1 + exp(-B * (date_int - t0)))^(1/v)
##
## Parameters:
##     Estimate Std. Error   t value Pr(>|t|)
## K  6.743e+04  3.909e+02   172.490  < 2e-16 ***
## B  2.799e-01  2.746e-02    10.196 1.50e-14 ***
## t0 1.830e+04  1.018e+00 17983.849  < 2e-16 ***
## v  1.490e+00  2.885e-01     5.164 3.09e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1976 on 58 degrees of freedom
##
## Number of iterations to convergence: 11
## Achieved convergence tolerance: 9.751e-06
```

```r
broom::glance(fitmodel)
```

```
## # A tibble: 1 x 8
##   sigma isConv     finTol logLik   AIC   BIC   deviance df.residual
##   <dbl> <lgl>       <dbl>  <dbl> <dbl> <dbl>      <dbl>       <int>
## 1 1976. TRUE   0.00000975  -556. 1123. 1133. 226420388.          58
```

EX12

```r
Hubei <- new_dat1 %>%
  filter(`Country/State` %in% c("Hubei,China"))%>%
  mutate(date=as.Date(date,format = "%m/%d/%y"),
         date_int=unclass(date))

Guangdong <- new_dat1 %>%
  filter(`Country/State` %in% c("Guangdong,China"))%>%
  mutate(date=as.Date(date,format = "%m/%d/%y"),
         date_int=unclass(date))

Diamond_Princess <- new_dat1 %>%
  filter(`Country/State` %in% c("Diamond Princess,Cruise Ship"))%>%
  mutate(date=as.Date(date,format = "%m/%d/%y"),
         date_int=unclass(date))

Henan <- new_dat1 %>%
  filter(`Country/State` %in% c("Henan,China"))%>%
  mutate(date=as.Date(date,format = "%m/%d/%y"),
         date_int=unclass(date))

Zhejiang <- new_dat1 %>%
  filter(`Country/State` %in% c("Zhejiang,China"))%>%
```

```
  mutate(date=as.Date(date,format = "%m/%d/%y"),
         date_int=unclass(date))


country_mod2 <- function(df){
  startK <- max(df$Confirmed_Cases)
  nls(Confirmed_Cases ~ K /( (1 +  exp(-B * (date_int - t0))) ) ^(1/v) ,
                data = df,
                start = list(K = startK,  B = .25, t0 = 18300, v = 1),
                control = list(maxiter = 1000, warnOnly = TRUE))
}

data<-c(Zhejiang,Henan,Diamond_Princess,Guangdong,Hubei)

#map(data,country_mod2)

country_mod2(Zhejiang)
```

```
## Warning in nls(Confirmed_Cases ~ K/((1 + exp(-B * (date_int - t0))))^(1/v), :
## step factor 0.000488281 reduced below 'minFactor' of 0.000976562
```

```
## Nonlinear regression model
##   model: Confirmed_Cases ~ K/((1 + exp(-B * (date_int - t0))))^(1/v)
##    data: df
##         K         B        t0         v
## 1.228e+03 1.066e-01 1.826e+04 2.625e-02
##  residual sum-of-squares: 1093304
##
## Number of iterations till stop: 40
## Achieved convergence tolerance: 4.631
## Reason stopped: step factor 0.000488281 reduced below 'minFactor' of 0.000976563
```

```
country_mod2(Henan)
```

```
## Nonlinear regression model
##   model: Confirmed_Cases ~ K/((1 + exp(-B * (date_int - t0))))^(1/v)
##    data: df
##         K         B        t0         v
## 1.276e+03 2.375e-01 1.829e+04 5.262e-01
##  residual sum-of-squares: 5229
##
## Number of iterations to convergence: 9
## Achieved convergence tolerance: 2.736e-06
```

```
country_mod2(Diamond_Princess)
```

```
## Nonlinear regression model
##   model: Confirmed_Cases ~ K/((1 + exp(-B * (date_int - t0))))^(1/v)
##    data: df
##         K         B        t0         v
## 7.012e+02 6.322e-01 1.831e+04 2.480e+00
```

```
##  residual sum-of-squares: 13272
##
## Number of iterations to convergence: 13
## Achieved convergence tolerance: 2.321e-06
```

```
country_mod2(Guangdong)
```

```
## Nonlinear regression model
##   model: Confirmed_Cases ~ K/((1 + exp(-B * (date_int - t0))))^(1/v)
##    data: df
##         K         B        t0         v
## 1.362e+03 2.111e-01 1.829e+04 2.865e-01
##  residual sum-of-squares: 11478
##
## Number of iterations to convergence: 18
## Achieved convergence tolerance: 5.174e-07
```

```
country_mod2(Hubei)
```

```
## Nonlinear regression model
##   model: Confirmed_Cases ~ K/((1 + exp(-B * (date_int - t0))))^(1/v)
##    data: df
##         K         B        t0         v
## 6.743e+04 2.799e-01 1.830e+04 1.490e+00
##  residual sum-of-squares: 226420388
##
## Number of iterations to convergence: 11
## Achieved convergence tolerance: 9.751e-06
```