Challenge 11

Heng Javier

2023-11-01

```
#1. Downloading API from website
library(httr)
library(jsonlite)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()  masks stats::filter()
## x purrr::flatten() masks jsonlite::flatten()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

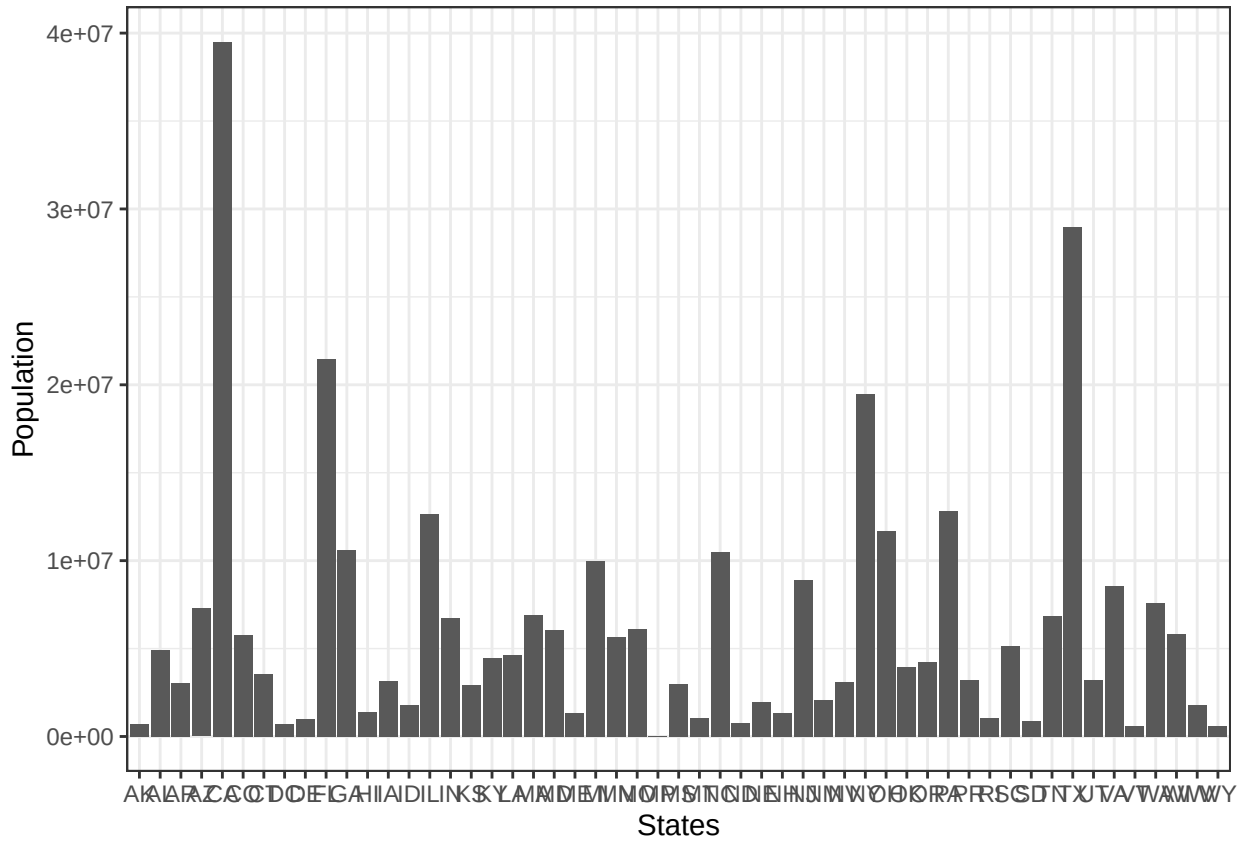
```
historic_state_data_url <- "https://api.covidactnow.org/v2/states.timeseries.json?apiKey=81af7dd43b4c4f5
raw_data <- GET(historic_state_data_url)
#2. Extracting data from JSON into Data Frame
data <- fromJSON(rawToChar(raw_data$content))
glimpse(data)
```

```
## Rows: 53
## Columns: 25
## $ fips                          <chr> "02", "01", "05", "04", "06", "08", "09~
## $ country                       <chr> "US", "US", "US", "US", "US", "US", "US~
## $ state                         <chr> "AK", "AL", "AR", "AZ", "CA", "CO", "CT~
## $ county                        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ hsa                           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ hsaName                       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ level                         <chr> "state", "state", "state", "state", "st~
## $ lat                           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ locationId                    <chr> "iso1:us#iso2:us-ak", "iso1:us#iso2:us-~
## $ long                          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ population                    <int> 731545, 4903185, 3017804, 7278717, 3951~
## $ hsaPopulation                 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ metrics                       <df[,14]> <data.frame[26 x 14]>
## $ riskLevels                    <df[,6]> <data.frame[26 x 6]>
## $ cdcTransmissionLevel          <int> 3, 4, 3, 3, 1, 4, 4, 1, 4, 4, 2, 3,~
## $ communityLevels               <df[,2]> <data.frame[26 x 2]>
```

```
## $ actuals                          <df[,19]> <data.frame[26 x 19]>
## $ annotations                      <df[,30]> <data.frame[26 x 30]>
## $ lastUpdatedDate                  <chr> "2023-11-01", "2023-11-01", "2023-11~
## $ url                              <chr> "https://covidactnow.org/us/alaska-ak",~
## $ metricsTimeseries                <list> [<data.frame[1336 x 14]>], [<data.fr~
## $ actualsTimeseries                <list> [<data.frame[1336 x 20]>], [<data.f~
## $ riskLevelsTimeseries             <list> [<data.frame[1336 x 3]>], [<data.fr~
## $ cdcTransmissionLevelTimeseries <list> [<data.frame[1336 x 2]>], [<data.frame[~
## $ communityLevelsTimeseries        <list> [<data.frame[1336 x 3]>], [<data.frame[~
```

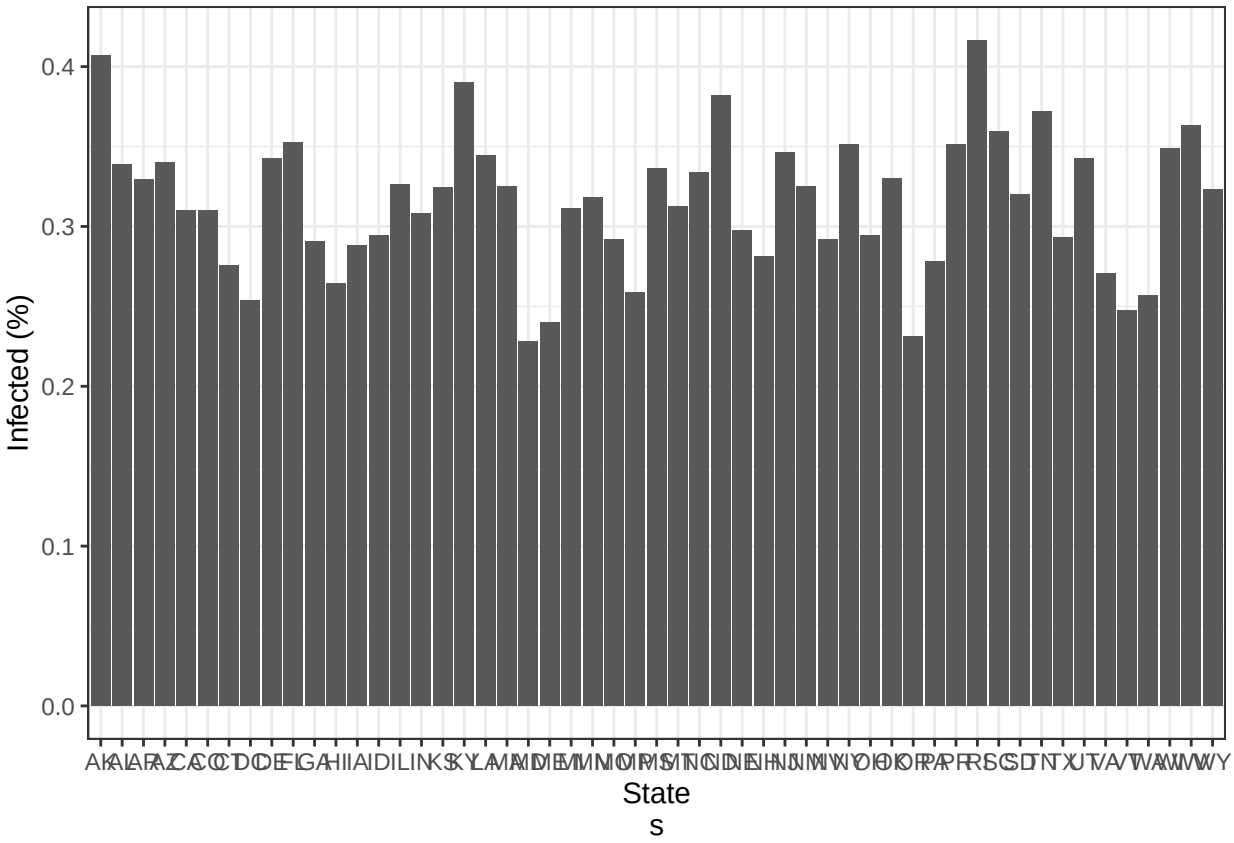
#i. What is the population in various states of U.S.A?

```
ggplot(data, aes(x=state,y=population)) + geom_bar(stat="identity") +labs(x="States",y="Population") + t
```



#ii. What fraction of the population was infected ?

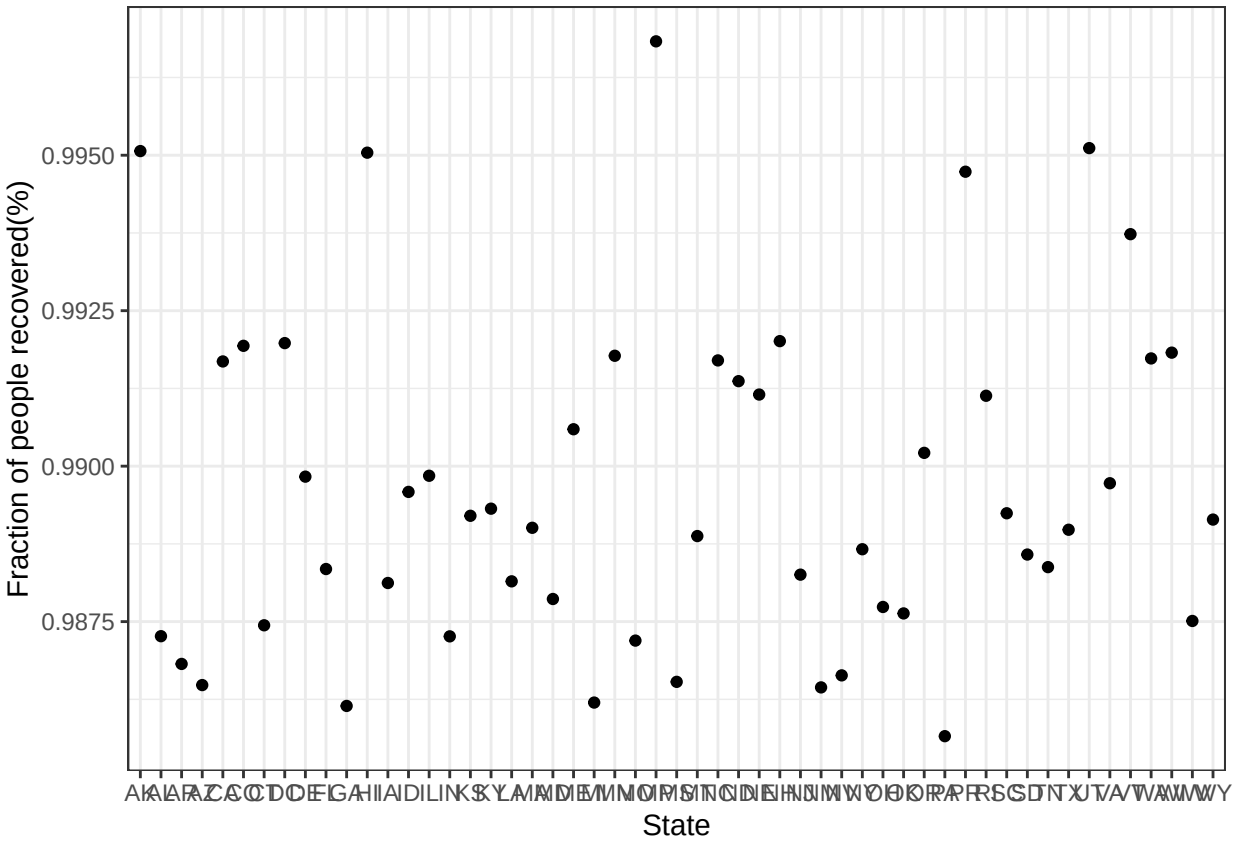
```
ggplot(data, aes(x=state,y=(data$actuals$cases/population))) + geom_bar(stat="identity") + labs(x="State
s",y="Infected (%)")+theme_bw()
```



#iii. What fraction of infected persons recovered ?

```
ggplot(data, aes(x=(data$state),y=((data$actuals$cases  -data$actuals$deaths)/data$actuals$cases))) + g
```

```
## Warning: Use of ‘data$state‘ is discouraged.
## i Use ‘state‘ instead.
```

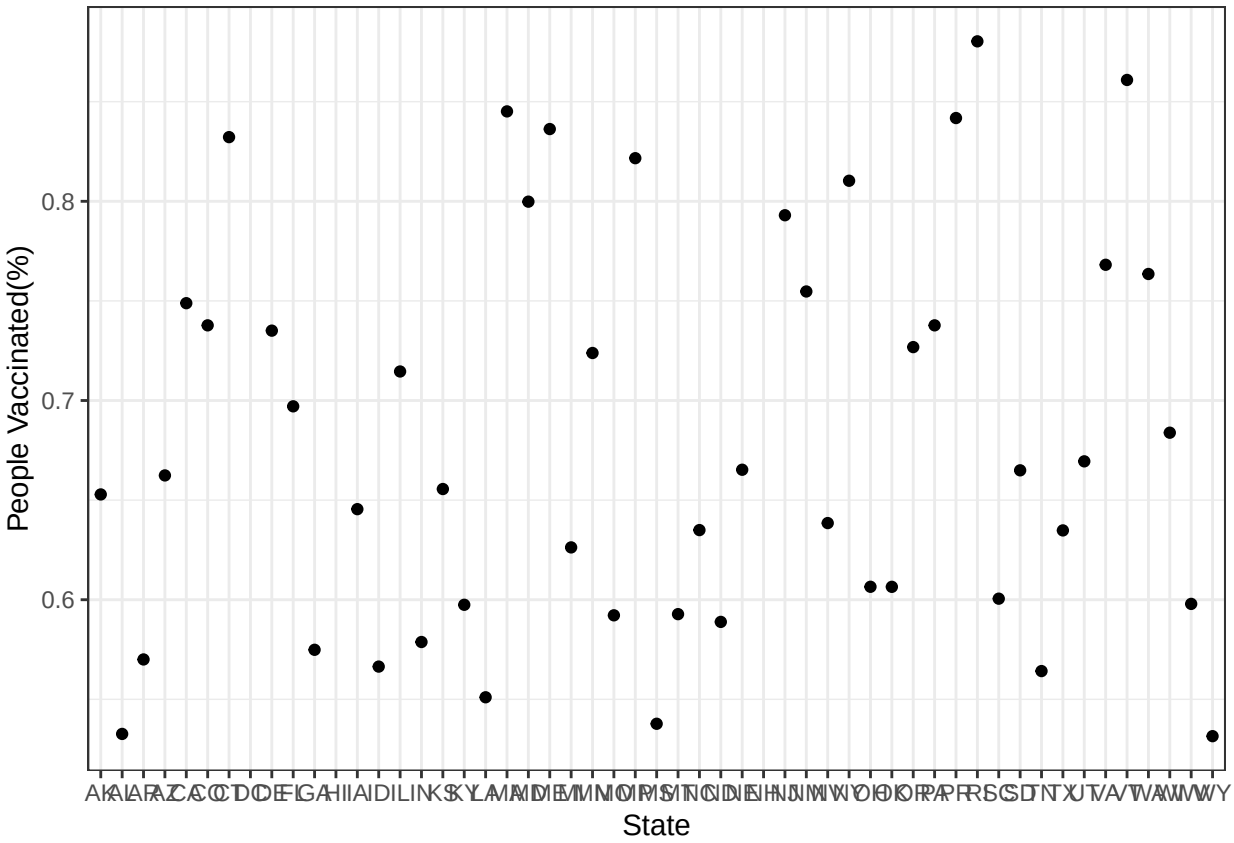


iv. What fraction of the population is currently vaccinated ?

```
ggplot(data, aes(x=(data$state),y=(data$actuals$vaccinationsCompleted/population))) + geom_point(stat="
```

```
## Warning: Use of ‘data$state‘ is discouraged.
## i Use ‘state‘ instead.
```

```
## Warning: Removed 3 rows containing missing values (‘geom_point()‘).
```

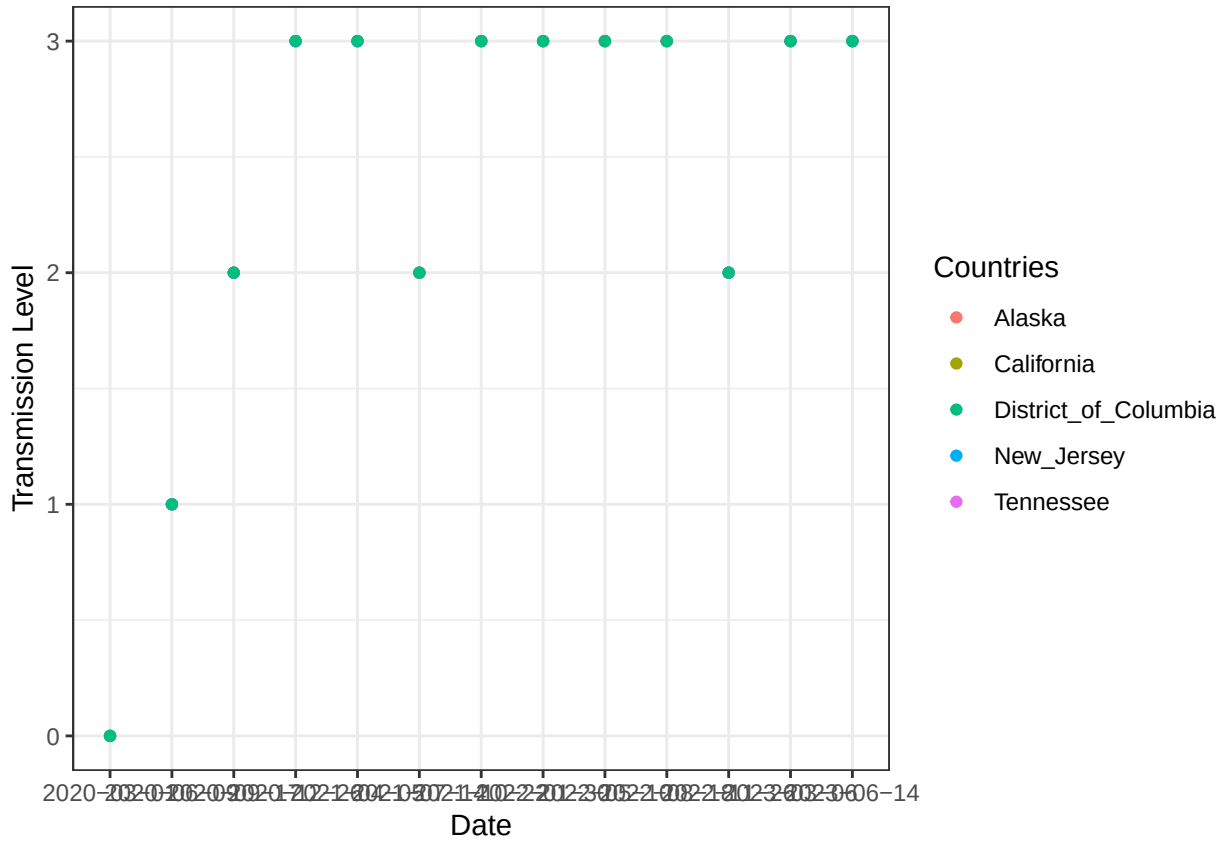


v. What was the transmission-like in the various states ?

```
#extracting data from the variable time_series and exploring actualsTimeseries
time_series <- data %>%
 unnest(actualsTimeseries) # <- to unravel the contents of a dataframe within a dataframe, use unnest
#Creating a new dataframe with needed data
time_series_transmission <- tibble(Date=time_series$cdcTransmissionLevelTimeseries[[which(data$state=="
# Transmission levels in each state
time_series_transmission$Alaska <- time_series$cdcTransmissionLevelTimeseries[[which(data$state=="AK")]]
cdcTransmissionLevel
time_series_transmission$California <- time_series$cdcTransmissionLevelTimeseries[[which(data$state=="C
time_series_transmission$New_Jersey <- time_series$cdcTransmissionLevelTimeseries[[which(data$state=="N
time_series_transmission$Tennessee <- time_series$cdcTransmissionLevelTimeseries[[which(data$state=="TN
time_series_transmission$District_of_Columbia <- time_series$cdcTransmissionLevelTimeseries[[which(data
print(head(time_series_transmission))
```

```
## # A tibble: 6 x 6
##   Date       Alaska California New_Jersey Tennessee District_of_Columbia
##   <chr>       <int>      <int>      <int>     <int>                <int>
## 1 2020-03-01      0          0          0         0                    0
## 2 2020-03-02      0          0          0         0                    0
## 3 2020-03-03      0          0          0         0                    0
## 4 2020-03-04      0          0          0         0                    0
## 5 2020-03-05      0          0          0         0                    0
## 6 2020-03-06      0          0          0         0                    0
```

```r
#plotting v
time_series_transmission[seq(1,1300,by=100),]%>%
 pivot_longer(cols=Alaska:District_of_Columbia,names_to="Countries",values_to="Transmission")%>%
ggplot(aes(x=Date,y=Transmission,colour=Countries,group=Countries)) +
geom_point(show.legend=TRUE) + labs(x="Date",y="Transmission Level")+theme_bw()
```




```r
#Notice, how the values are all identical. Always check the veracity of the data in the data-set before
```

vi. How did the disease progress since it started ?

```r
# New data-frame with dates
time_series_cases <- list(Alaska = time_series %>% filter(state=="AK") %>% select(date,cases))
# Cases of each state
time_series_cases$California <- time_series %>% filter(state=="CA") %>% select(date,cases)
time_series_cases$New_Jersey <- time_series %>% filter(state=="NJ") %>% select(date,cases)
time_series_cases$Tennessee <- time_series %>% filter(state=="TN") %>% select(date,cases)
time_series_cases$District_of_Columbia <- time_series %>% filter(state=="DC") %>% select(date,cases)
#Creating a dataframe for ggplot
data_to_plot <- tibble(Date_Alaska = time_series_cases$Alaska$date[seq(1,1300,by=100)],
 Cases_Alaska = time_series_cases$Alaska$cases[seq(1,1300,by=100)],
 Date_California = time_series_cases$California$date[seq(1,1300,by=100)],
 Cases_California = time_series_cases$California$cases[seq(1,1300,by=100)],
 Date_New_Jersey = time_series_cases$New_Jersey$date[seq(1,1300,by=100)],
 Cases_New_Jersey = time_series_cases$New_Jersey$cases[seq(1,1300,by=100)],
```

```
 Date_Tennessee = time_series_cases$Tennessee$date[seq(1,1300,by=100)],
 Cases_Tennessee = time_series_cases$Tennessee$cases[seq(1,1300,by=100)],
 Date_District_of_Columbia = time_series_cases$District_of_Columbia$date[seq(1,1300,by=100)],
 Cases_District_of_Columbia = time_series_cases$District_of_Columbia$cases[seq(1,1300,by=100)])
data_to_plot
```

```
## # A tibble: 13 x 10
##    Date_Alaska Cases_Alaska Date_California Cases_California Date_New_Jersey
##    <chr>              <int> <chr>                      <int> <chr>
##  1 2020-03-01            NA 2020-01-25                     1 2020-03-01
##  2 2020-06-09           620 2020-05-04                 56333 2020-06-09
##  3 2020-09-17          7413 2020-08-12                595097 2020-09-17
##  4 2020-12-26         45247 2020-11-20               1096427 2020-12-26
##  5 2021-04-05         63486 2021-02-28               3569578 2021-04-05
##  6 2021-07-14         71539 2021-06-08               3798225 2021-07-14
##  7 2021-10-22        132393 2021-09-16               4629146 2021-10-22
##  8 2022-01-30        211117 2021-12-25               5291605 2022-01-30
##  9 2022-05-10        252847 2022-04-04               9110544 2022-05-10
## 10 2022-08-18        289203 2022-07-13              10365785 2022-08-18
## 11 2022-11-26        299841 2022-10-21              11338846 2022-11-26
## 12 2023-03-06        307377 2023-01-29              11980312 2023-03-06
## 13 2023-06-14            NA 2023-05-09              12242634 2023-06-14
## # i 5 more variables: Cases_New_Jersey <int>, Date_Tennessee <chr>,
## #   Cases_Tennessee <int>, Date_District_of_Columbia <chr>,
## #   Cases_District_of_Columbia <int>
```

```
#From the data, we can notice that the data we have is for different dates. Hence, we cannot plot them
```

```
library(cowplot)
```

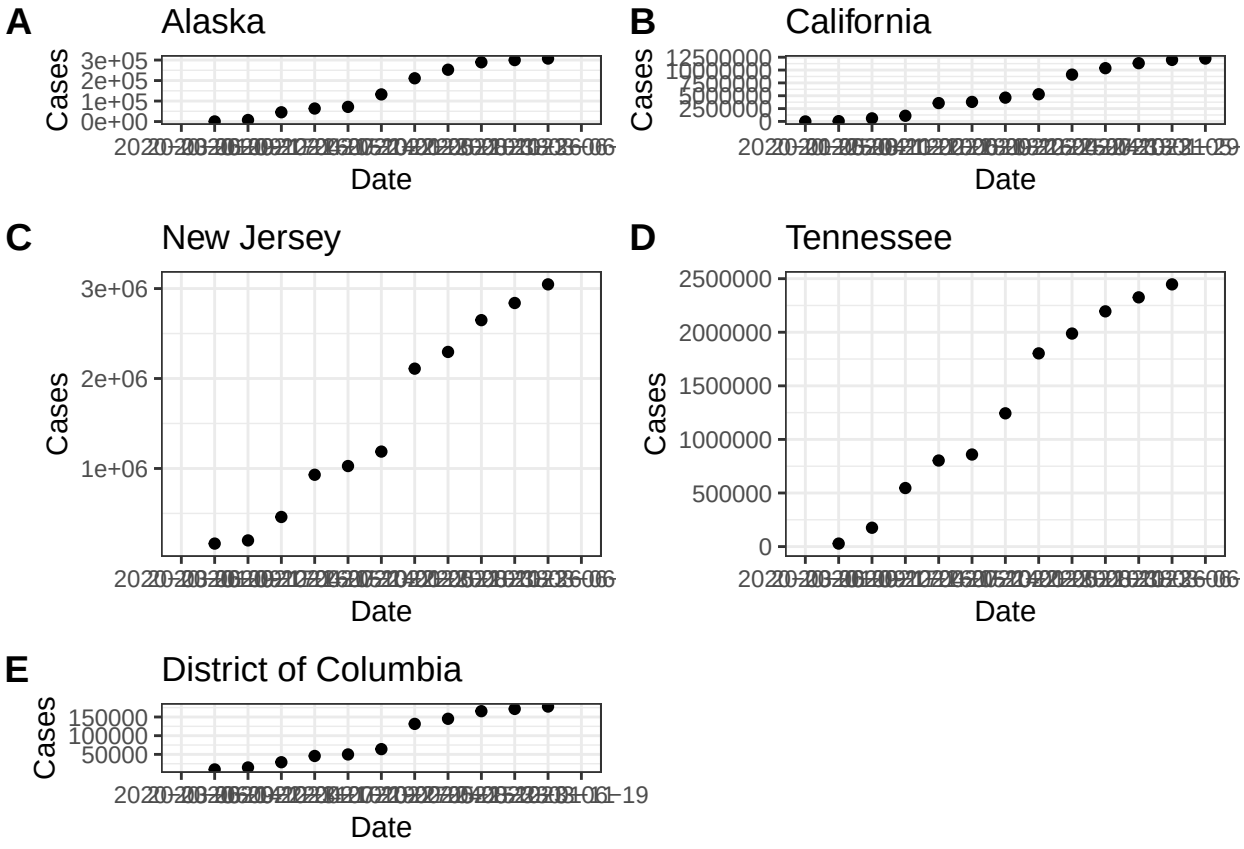
```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
fig1<- ggplot(data_to_plot, aes(x=Date_Alaska,y=Cases_Alaska)) +
geom_point() + labs(x="Date",y="Cases", title="Alaska") + theme_bw()
fig2<- ggplot(data_to_plot, aes(x=Date_California,y=Cases_California)) +
geom_point() + labs(x="Date",y="Cases", title="California") + theme_bw()
fig3<- ggplot(data_to_plot, aes(x=Date_New_Jersey,y=Cases_New_Jersey)) +
geom_point() + labs(x="Date",y="Cases", title="New Jersey") + theme_bw()
fig4<- ggplot(data_to_plot, aes(x=Date_Tennessee,y=Cases_Tennessee)) +
geom_point() + labs(x="Date",y="Cases", title="Tennessee") + theme_bw()
fig5<- ggplot(data_to_plot, aes(x=Date_District_of_Columbia,y=Cases_District_of_Columbia)) +
geom_point() + labs(x="Date",y="Cases", title="District of Columbia") + theme_bw()
plot_grid(fig1 + theme(legend.justification = c(0,1)),
 fig2 + theme(legend.justification = c(1,0)),
 fig3 + theme(legend.justification = c(0,1)),
 fig4 + theme(legend.justification = c(1,0)),
```

```
fig5 + theme(legend.justification = c(0,1)),
align = "v", axis = "lr", nrow=3,
ncol = 2,labels = LETTERS[1:5],
rel_heights = c(1,2))
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
## Removed 2 rows containing missing values (‘geom_point()‘).
## Removed 2 rows containing missing values (‘geom_point()‘).
## Removed 2 rows containing missing values (‘geom_point()‘).
```



#Note, how these questions require different resolutions of data. Questions (i) - (iv) do not need historical data, we could have used the current data available on the webpage for this. But, (v) - (vi) needs us to plot the values of transmission and cases on a periodical basis, therefore requiring time-series values #https://api.covidactnow.org/v2/states.timeseries.json?apiKey=81af7dd43b4c4f5b9f4c8518fff8b0fe