Week 10 Challenge

Ariel Quek

2023-10-30

```
knitr::opts_chunk$set(echo = TRUE)
```

Week 10 Challenge

Step 1

Downloading data set (API)

```
library(httr)
```

```
## Warning: package 'httr' was built under R version 4.2.3
```

```
library(jsonlite)
```

```
## Warning: package 'jsonlite' was built under R version 4.2.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.1
```

```
## Warning: package 'tidyr' was built under R version 4.2.2
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.0     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.3     ✔ tibble    3.1.8
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1     
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter()  masks stats::filter()
## ✖ purrr::flatten() masks jsonlite::flatten()
## ✖ dplyr::lag()     masks stats::lag()
## ℹ Use the □]8;;http://conflicted.r-lib.org/□conflicted package□]8;;□ to force all conflict
s to become errors
```

Retrieving data

```
historic_state_data_url <- "https://api.covidactnow.org/v2/states.timeseries.json?apiKey=aee461090f09499f86335e3630089532"

raw_data <- GET(historic_state_data_url)
```

Step 2

Converting data to a dataframe

```
data <- fromJSON(rawToChar(raw_data$content))
```

Step 3

Get a glimpse of data-set

```
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> 2, 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> "2023-10-30", "2023-10-30", "2023-10…
## $ url                            <chr> "https://covidactnow.org/us/alaska-ak",…
## $ metricsTimeseries              <list> [<data.frame[1334 x 14]>], [<data.fr…
## $ actualsTimeseries              <list> [<data.frame[1334 x 20]>], [<data.f…
## $ riskLevelsTimeseries           <list> [<data.frame[1334 x 3]>], [<data.fr…
## $ cdcTransmissionLevelTimeseries <list> [<data.frame[1334 x 2]>], [<data.frame[…
## $ communityLevelsTimeseries      <list> [<data.frame[1334 x 3]>], [<data.frame[…
```

Step 4

We will work on the following questions:

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

ii. What fraction of the population was infected ?

iii. What fraction of infected persons recovered ?

iv. What fraction of the population is currently vaccinated ? *the above do not need historical data*

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

vi. How did the disease progress since it started ? *the above needs us to plot values of transmission and cases on a periodical basis – requires time-series values*

Step 5

Extracting time-series data from the data-frame

```
time_series <- data %>% unnest(actualsTimeseries)
# <- to unravel the contents of a dataframe within a dataframe, use unnest
```

Creating a new dataframe with the needed data

```
time_series_transmission <- tibble(Date=time_series$cdcTransmissionLevelTimeseries[[which(dat
a$state=="CA")]]$date)

# Transmission levels in each state
time_series_transmission$Alaska <- time_series$cdcTransmissionLevelTimeseries[[which(data$sta
te=="AK")]]$cdcTransmissionLevel

time_series_transmission$California <- time_series$cdcTransmissionLevelTimeseries[[which(data
$state=="CA")]]$cdcTransmissionLevel

time_series_transmission$New_Jersey <- time_series$cdcTransmissionLevelTimeseries[[which(data
$state=="NJ")]]$cdcTransmissionLevel

time_series_transmission$Tennessee <- time_series$cdcTransmissionLevelTimeseries[[which(data
$state=="TN")]]$cdcTransmissionLevel

time_series_transmission$District_of_Columbia <- time_series$cdcTransmissionLevelTimeseries
[[which(data$state=="DC")]]$cdcTransmissionLevel

print(head(time_series_transmission))
```

```
## # A tibble: 6 × 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
```

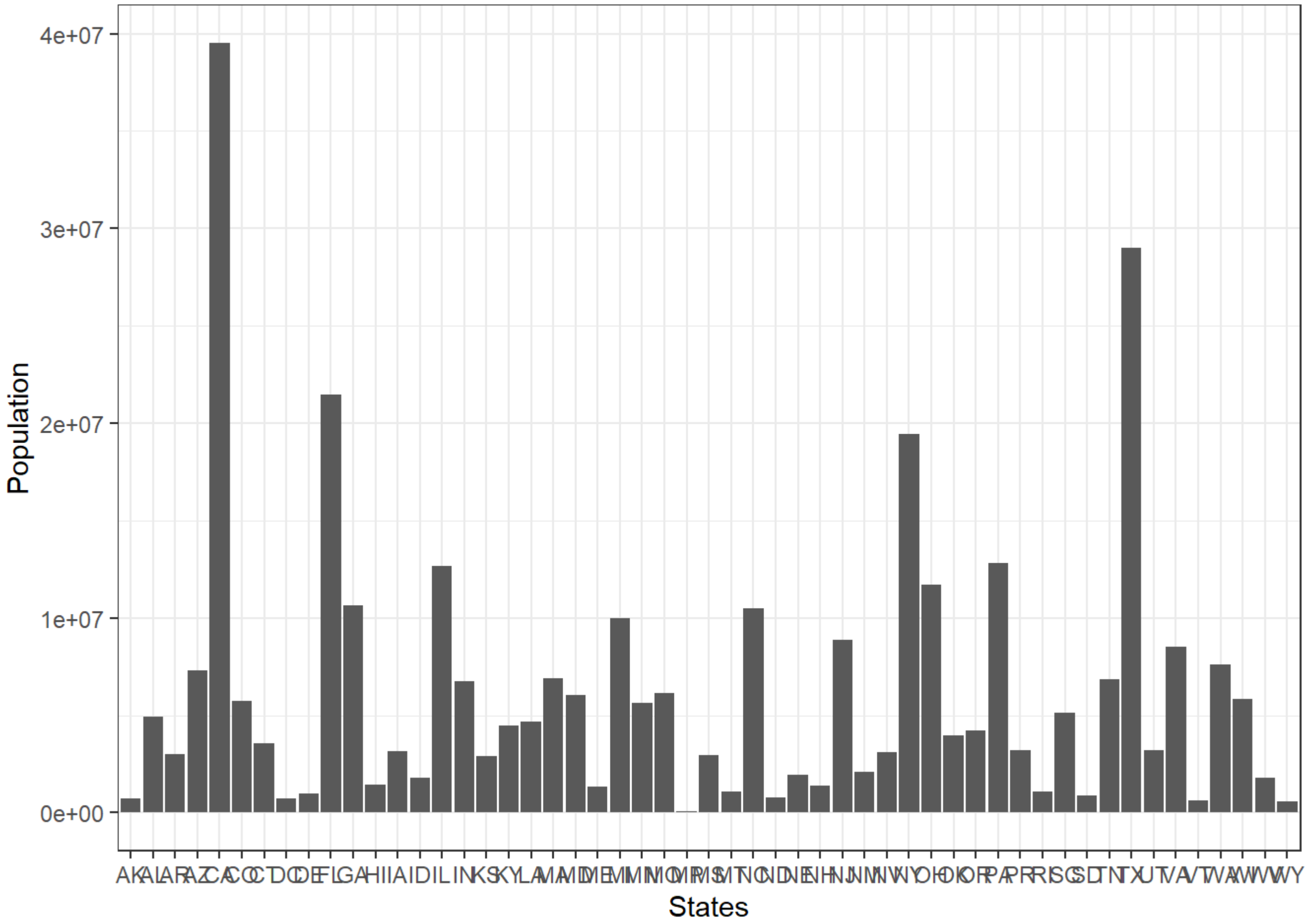
Selecting cases of each states from a new data-frame with dates

```
# New data-frame with dates
time_series_cases <- list(Alaska = time_series %>% filter(state=="AK") %>% select(date,case
s))
# 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(dat
e,cases)
```

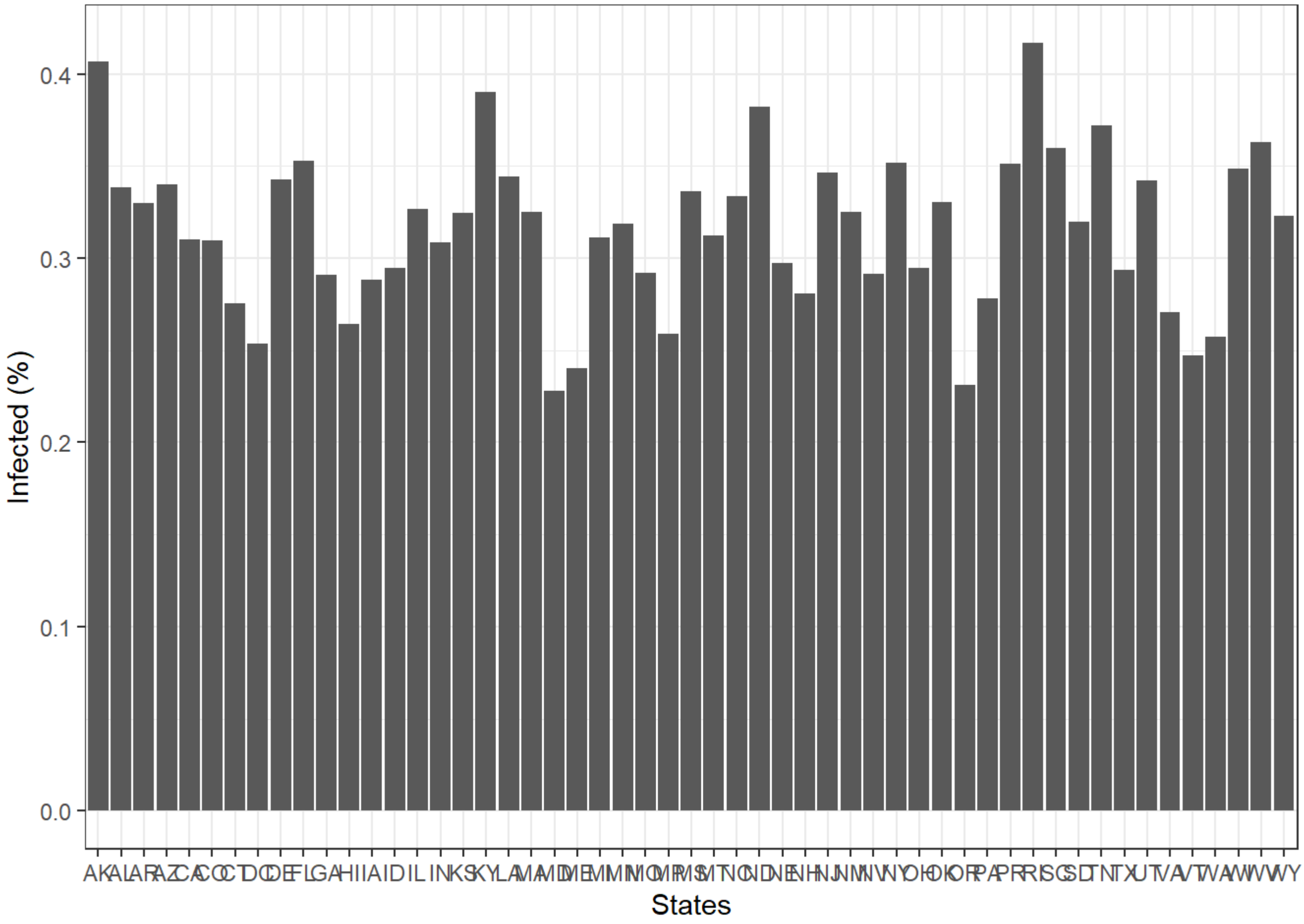
Step 6

Visualising the data

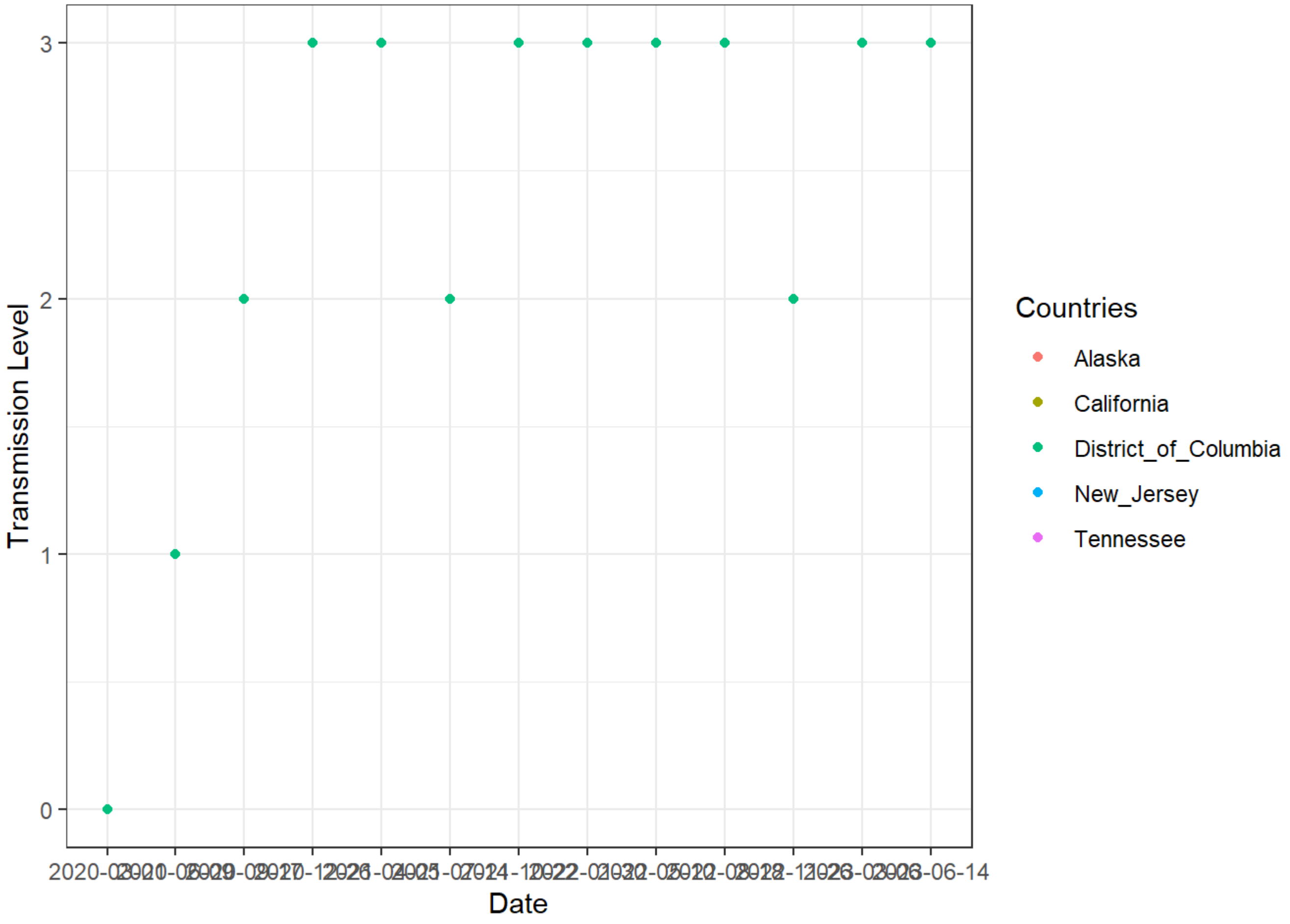
```
ggplot(data, aes(x=state,y=population)) + geom_bar(stat="identity") +labs(x="States",y="Popul
ation") + theme_bw()
```



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



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



Representing the data

```
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=10
0)],
                       Cases_California = time_series_cases$California$cases[seq(1,1300,by=10
0)],
                       Date_New_Jersey = time_series_cases$New_Jersey$date[seq(1,1300,by=10
0)],
                       Cases_New_Jersey = time_series_cases$New_Jersey$cases[seq(1,1300,by=10
0)],
                       Date_Tennessee = time_series_cases$Tennessee$date[seq(1,1300,by=100)],
                       Cases_Tennessee = time_series_cases$Tennessee$cases[seq(1,1300,by=10
0)],
                       Date_District_of_Columbia = time_series_cases$District_of_Columbia$dat
e[seq(1,1300,by=100)],
                       Cases_District_of_Columbia = time_series_cases$District_of_Columbia$ca
ses[seq(1,1300,by=100)])

data_to_plot
```

```
## # A tibble: 13 × 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>
```

Ploting subplots

```
install.packages("cowplot", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Ariel/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'cowplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Ariel\AppData\Local\Temp\RtmpALSzl9\downloaded_packages
```

```
library(cowplot)
```

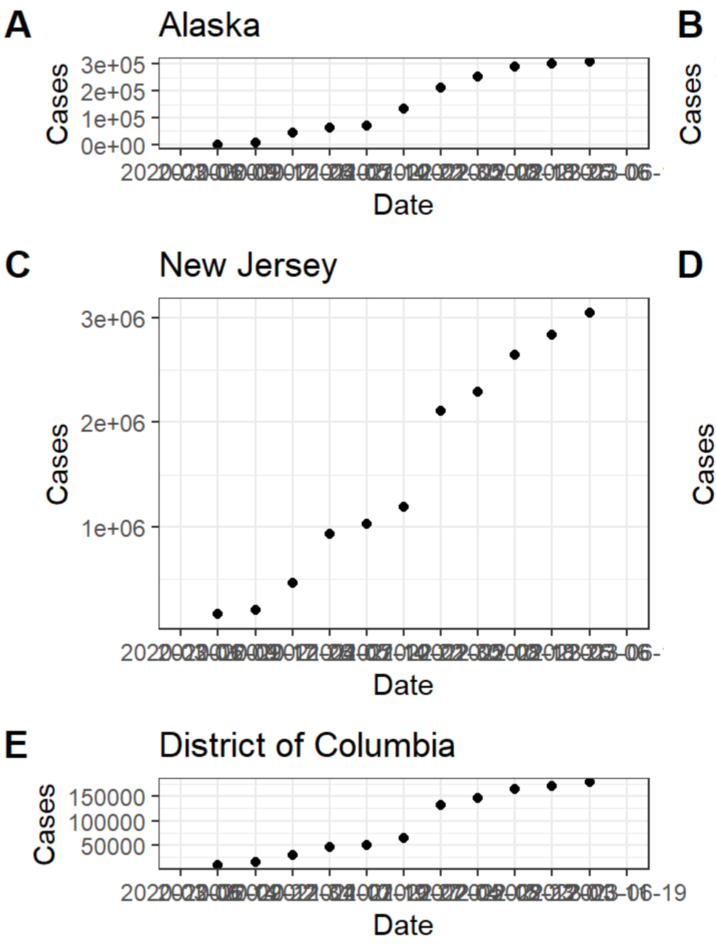
```
## Warning: package 'cowplot' was built under R version 4.2.3
```

```
##
## 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()`).
```



Varying the size to play with the resolution

```
new_resolution <-
  plot_grid(
  fig1 + theme(legend.justification = c(0, 1), axis.text.x = element_text(size = 3)),
  fig2 + theme(legend.justification = c(1, 0), axis.text.x = element_text(size = 3)),
  fig3 + theme(legend.justification = c(0, 1), axis.text.x = element_text(size = 3)),
  fig4 + theme(legend.justification = c(1, 0), axis.text.x = element_text(size = 3)),
  fig5 + theme(legend.justification = c(0, 1), axis.text.x = element_text(size = 3)),
  align = "v", axis = "lr", nrow = 3, ncol = 2, labels = LETTERS[1:5], rel_heights = c(40, 5
0)
)
```

```
## 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()`).
```

```
ggsave("new_resolution.png", new_resolution, width = 10, height = 8, units = "in")
```

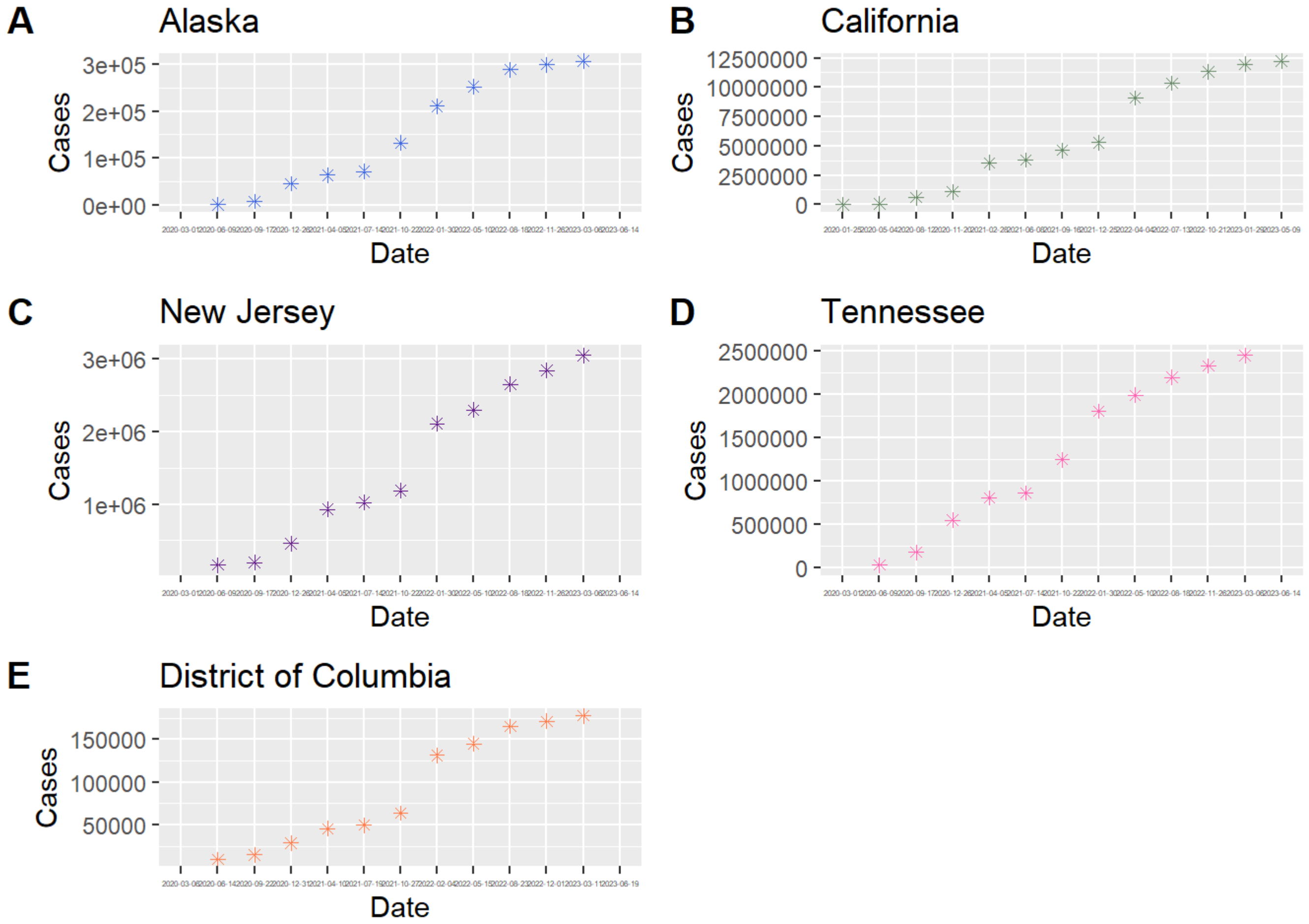
Varying the colours

```
# Modify the color for each plot using the fill color for points as an example
fig1<- ggplot(data_to_plot, aes(x=Date_Alaska,y=Cases_Alaska)) +
  geom_point(color="royalblue", shape=8) + labs(x="Date",y="Cases", title="Alaska")

fig2<- ggplot(data_to_plot, aes(x=Date_California,y=Cases_California)) +
  geom_point(color="darkseagreen4", shape=8) + labs(x="Date",y="Cases", title="California")

fig3<- ggplot(data_to_plot, aes(x=Date_New_Jersey,y=Cases_New_Jersey)) +
  geom_point(color="darkorchid4", shape=8) + labs(x="Date",y="Cases", title="New Jersey")

fig4<- ggplot(data_to_plot, aes(x=Date_Tennessee,y=Cases_Tennessee)) +
  geom_point(color="hotpink", shape=8) + labs(x="Date",y="Cases", title="Tennessee")

fig5<- ggplot(data_to_plot, aes(x=Date_District_of_Columbia,y=Cases_District_of_Columbia)) +
  geom_point(color="coral", shape=8) + labs(x="Date",y="Cases", title="District of Columbia")

new_with_colors <-
  plot_grid(
  fig1 + theme(legend.justification = c(0, 1), axis.text.x = element_text(size = 3)),
  fig2 + theme(legend.justification = c(1, 0), axis.text.x = element_text(size = 3)),
  fig3 + theme(legend.justification = c(0, 1), axis.text.x = element_text(size = 3)),
  fig4 + theme(legend.justification = c(1, 0), axis.text.x = element_text(size = 3)),
  fig5 + theme(legend.justification = c(0, 1), axis.text.x = element_text(size = 3)),
  align = "v", axis = "lr", nrow = 3, ncol = 2, labels = LETTERS[1:5], rel_heights = c(40, 5
0)
)
```

```
## 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()`).
```

```
# Save the combined plot with increased size
ggsave("new_with_colors.png", new_with_colors, width = 10, height = 8, units = "in")
```

```
new_with_colors
```


A
Alaska
Cases
3e+05
2e+05
1e+05
0e+00
Date
B
California
Cases
Date
C
New Jersey
Cases
3e+06
2e+06
1e+06
Date
D
Tennessee
Cases
Date
E
District of Columbia
Cases
Date