

Week_11_Challenge

Daniel Tan

2023-10-30

```
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 4.2.3
```

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

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

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

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

```
## 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 4.2.2
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.2      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=fc4688d33ec347"
raw_data <- GET(historic_state_data_url)
```

```
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> 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[~
```

```
time_series <- data %>%
  unnest(actualsTimeseries)
```

```
time_series
```

```
## # A tibble: 70,639 x 44
##   fips country state county hsa hsaName level lat locationId long
##   <chr> <chr> <chr> <lgl> <lgl> <lgl> <chr> <lgl> <chr> <lgl>
## 1 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## 2 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## 3 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## 4 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## 5 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## 6 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## 7 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## 8 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
```

```
## 9 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## 10 02 US AK NA NA NA state NA iso1:us#iso2:us-ak NA
## # i 70,629 more rows
## # i 34 more variables: population <int>, hsaPopulation <int>,
## # metrics <df[,14]>, riskLevels <df[,6]>, cdcTransmissionLevel <int>,
## # communityLevels <df[,2]>, actuals <df[,19]>, annotations <df[,30]>,
## # lastUpdatedDate <chr>, url <chr>, metricsTimeseries <list>, cases <int>,
## # deaths <int>, positiveTests <int>, negativeTests <int>,
## # contactTracers <int>, hospitalBeds <df[,4]>, hsaHospitalBeds <df[,4]>, ...
```

```
# Creating a new dataframe with needed data
```

```
# Save date
```

```
time_series_transmission <- tibble(Date=time_series$cdcTransmissionLevelTimeseries[[which(data$state=="AK")]]
```

```
# 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=="CA")]]
```

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

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

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

```
# New data-frame with dates
```

```
time_series_cases <- list(Alaska = time_series %>% filter(state=="AK") %>% select(date,cases))
```

```
# Cases of each state
```

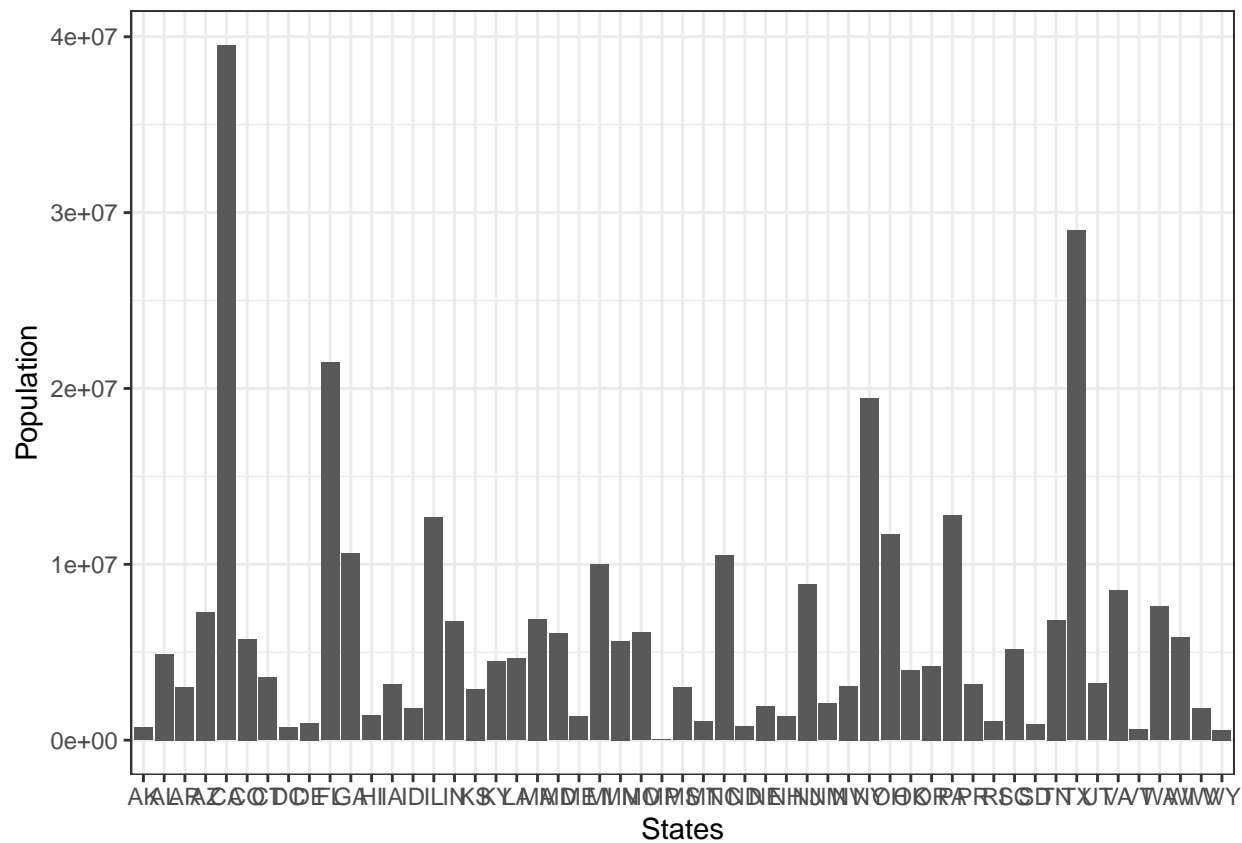
```
time_series_cases$California <- time_series %>% filter(state=="CA") %>% select(date,cases)
```

```
time_series_cases$New_Jersey <- time_series %>% filter(state=="NJ") %>% select(date,cases)
```

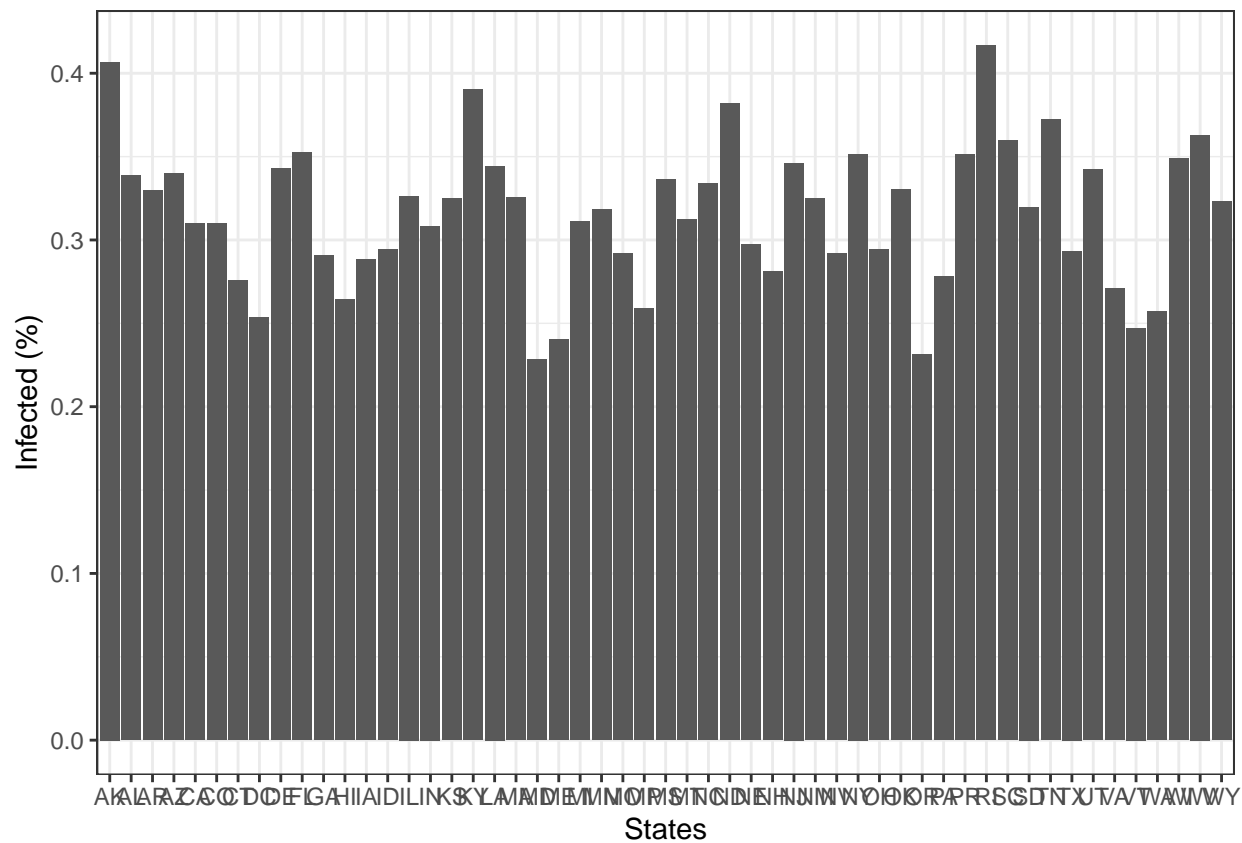
```
time_series_cases$Tennessee <- time_series %>% filter(state=="TN") %>% select(date,cases)
```

```
time_series_cases$District_of_Columbia <- time_series %>% filter(state=="DC") %>% select(date,cases)
```

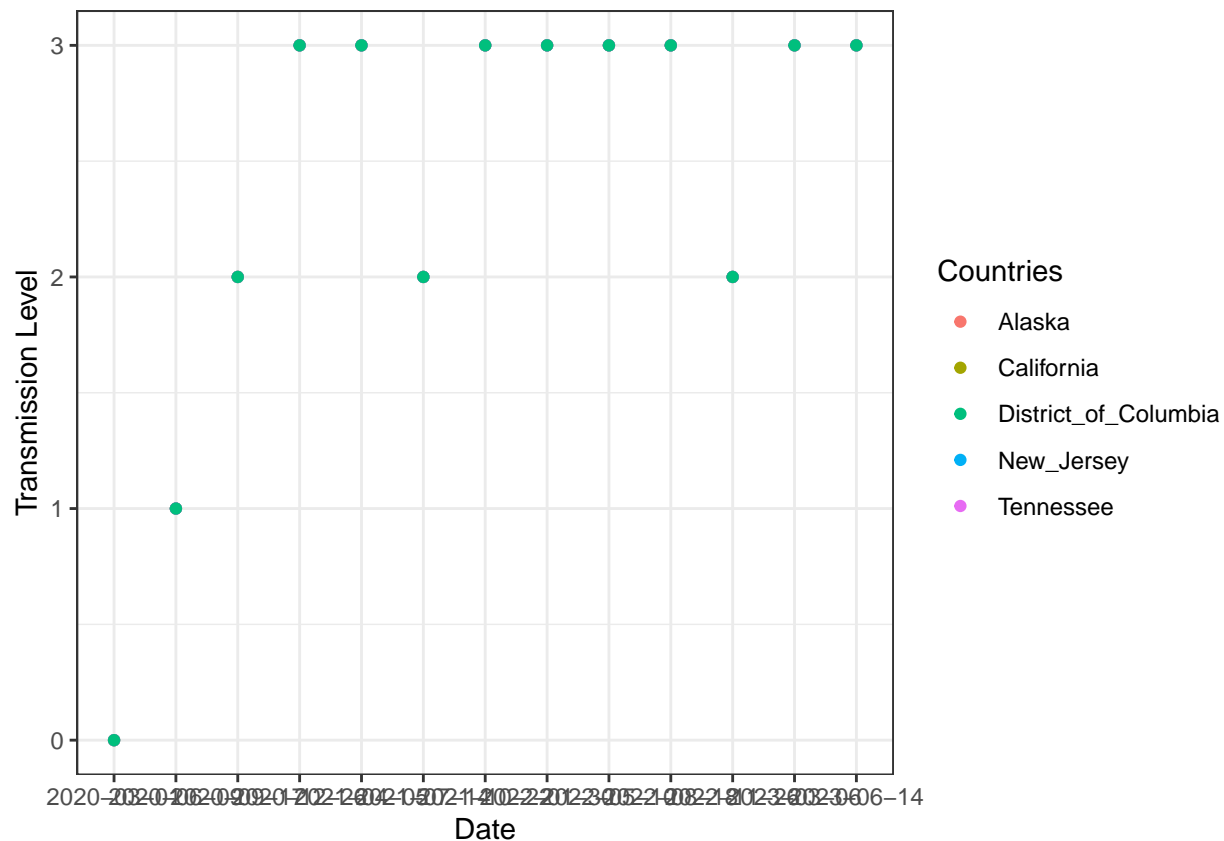
```
ggplot(data, aes(x=state,y=population)) + geom_bar(stat="identity") +labs(x="States",y="Population") +
```



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



```
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()
```



```
data_to_plot <- tibble(Date_Alaska = time_series_cases$Alaska$date[seq(1,1300,by=150)],
Cases_Alaska = time_series_cases$Alaska$cases[seq(1,1300,by=150)],
Date_California = time_series_cases$California$date[seq(1,1300,by=150)],
Cases_California = time_series_cases$California$cases[seq(1,1300,by=150)],
Date_New_Jersey = time_series_cases$New_Jersey$date[seq(1,1300,by=150)],
Cases_New_Jersey = time_series_cases$New_Jersey$cases[seq(1,1300,by=150)],
Date_Tennessee = time_series_cases$Tennessee$date[seq(1,1300,by=150)],
Cases_Tennessee = time_series_cases$Tennessee$cases[seq(1,1300,by=150)],
Date_District_of_Columbia = time_series_cases$District_of_Columbia$date[seq(1,1300,by=150)],
Cases_District_of_Columbia = time_series_cases$District_of_Columbia$cases[seq(1,1300,by=150)])
data_to_plot
```

```
## # A tibble: 9 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 2020-07-29    3440 2020-06-23    191039 2020-07-29
## 3 2020-12-26   45247 2020-11-20   1096427 2020-12-26
## 4 2021-05-25   69476 2021-04-19   3720922 2021-05-25
## 5 2021-10-22  132393 2021-09-16   4629146 2021-10-22
## 6 2022-03-21  243672 2022-02-13   8800780 2022-03-21
## 7 2022-08-18  289203 2022-07-13  10365785 2022-08-18
## 8 2023-01-15  302604 2022-12-10  11620250 2023-01-15
## 9 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>
```

```
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
```

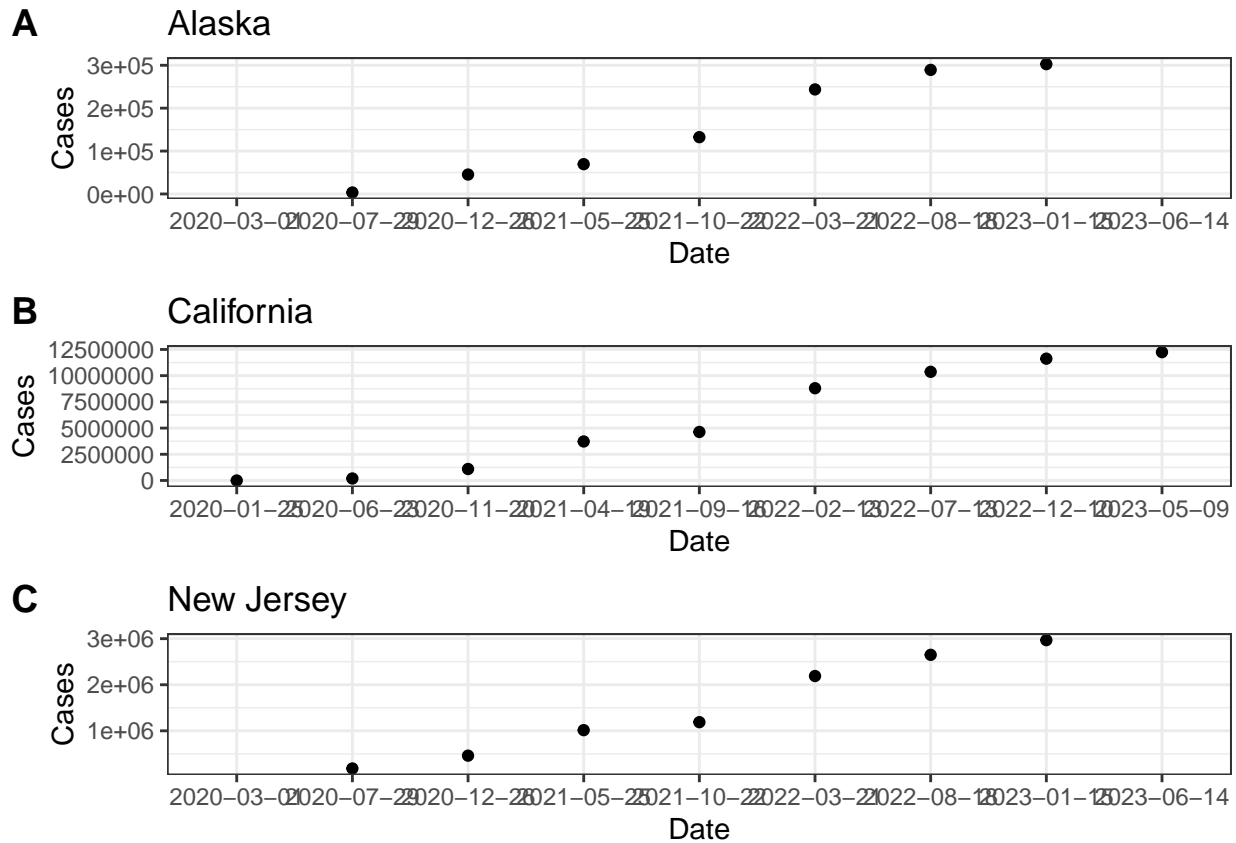
```
fig1<- ggplot(data_to_plot, aes(x=Date_Alaska,y=Cases_Alaska)) +  
geom_point() + labs(x="Date",y="Cases", title="Alaska") + theme_bw()  
fig2<- ggplot(data_to_plot, aes(x=Date_California,y=Cases_California)) +  
geom_point() + labs(x="Date",y="Cases", title="California") + theme_bw()  
fig3<- ggplot(data_to_plot, 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 = 1,labels = LETTERS[1:5])
```

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

```
## Removed 2 rows containing missing values ('geom_point()').
```

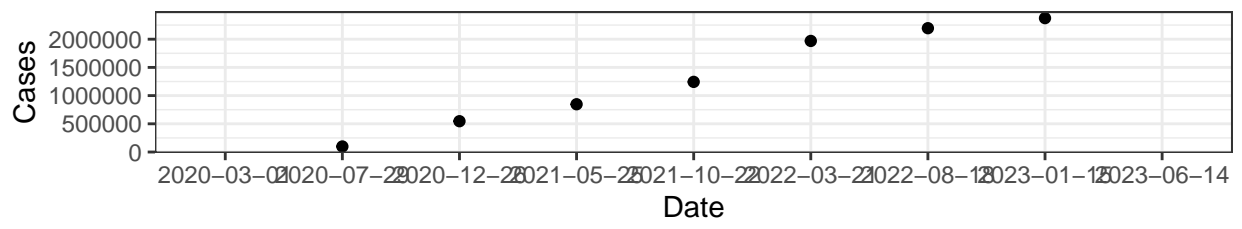
```
## Removed 2 rows containing missing values ('geom_point()').
```

```
## Removed 2 rows containing missing values ('geom_point()').
```



```
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(
fig4 + theme(legend.justification = c(1,0)),
fig5 + theme(legend.justification = c(0,1)),
align = "v", axis = "lr", nrow=3,
ncol = 1,labels = LETTERS[4:5])
```

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


D**Tennessee****E****District of Columbia**