title: ” ”
author: “Your Name”
date: ‘2023-02-20’
output:
html_document:
df_print: paged

```
library(data.table)
library(dplyr)
library(dplyr)
library(tidyr)
library(plotly)
library(lubridate)
```

In this homework you should use plotly unless said otherwise.

To create pdf version of your homework, knit it first to html and then print it to pdf. Interactive plotly plots can be difficult sometimes to convert to static images suitable for insertion to LaTex documents (that is knitting to PDF).

Look for questions in R-chunks as comments and plain text (they are prefixed as Q.).

Part 1. Iris Dataset. (20 points)

> “The Iris flower data set or Fisher’s Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis” https://en.wikipedia.org/wiki/Iris_flower_data_set (https://en.wikipedia.org/wiki/Iris_flower_data_set)

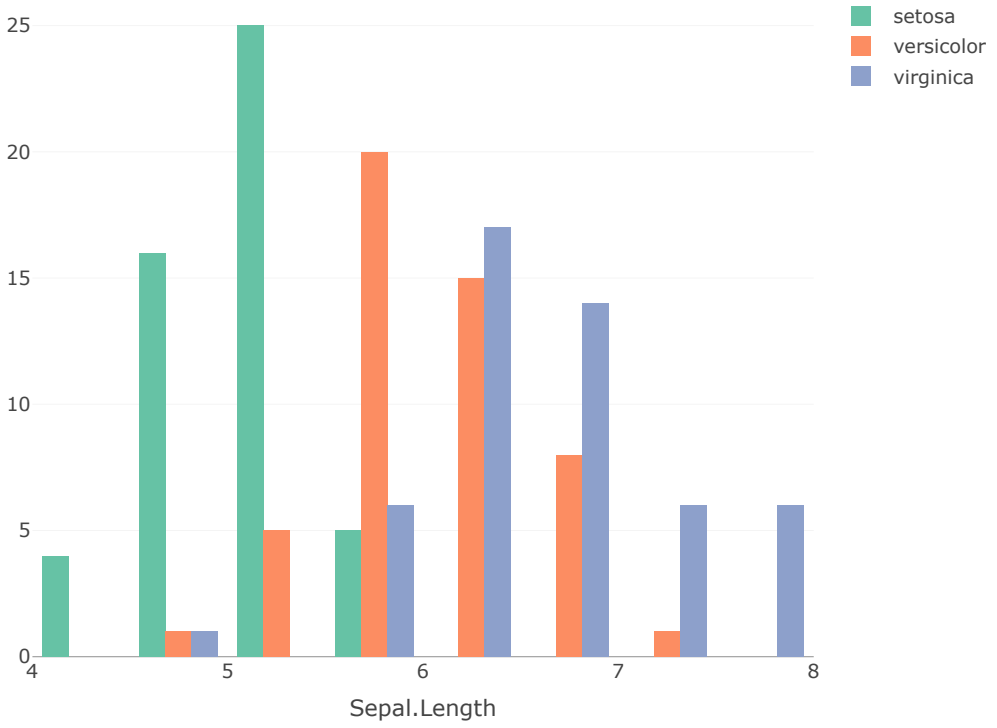
```
# Q1.1. Read the iris.csv file  (2 points)
# hint: use fread from data.table, it is significantly faster than default methods
#       be sure to have strings as factors (see stringsAsFactors argument)
```

```
# Q1.2. Show some values from data frame (2 points)
```

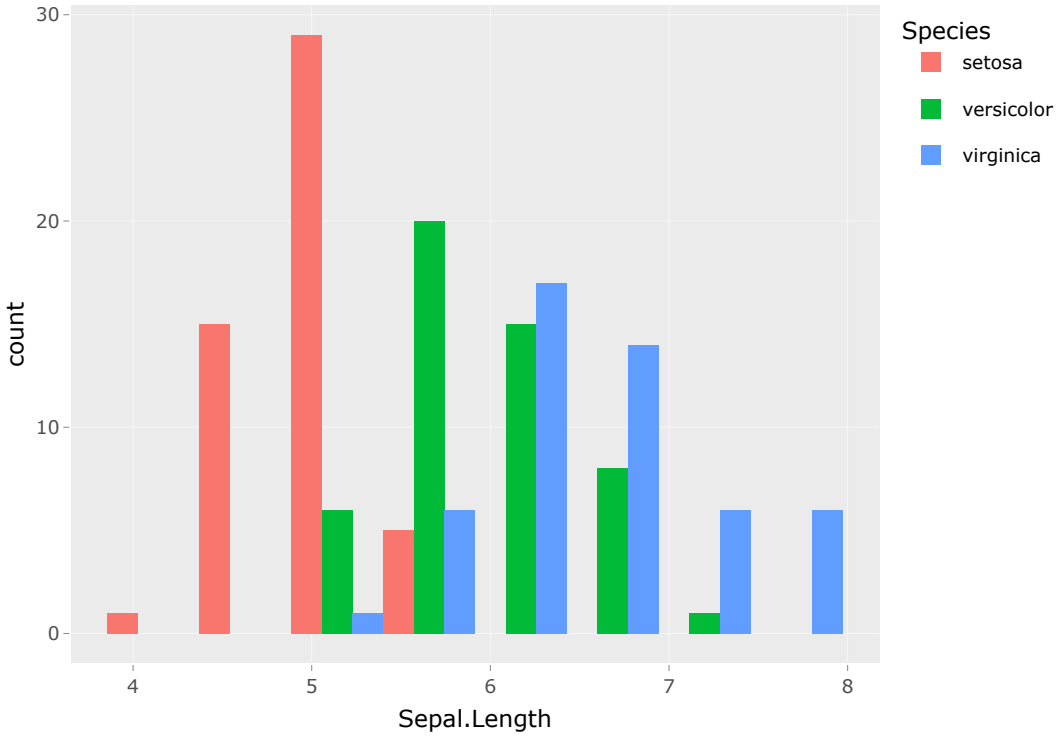
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

```
# Q1.3. Build histogram plot for Sepal.Length variable for each species using plot_ly
# (use color argument for grouping) (2 points)
# should be one plot
```

```
## No trace type specified:
##   Based on info supplied, a 'histogram' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#histogram
```

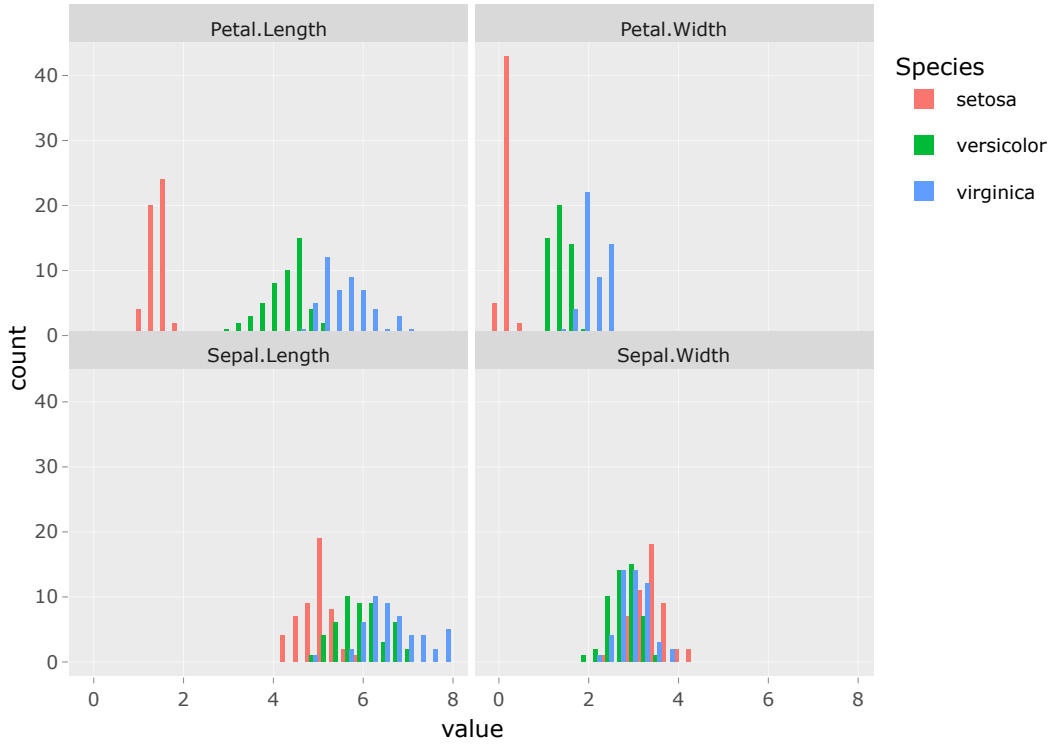



```
# Q1.4. Repeat previous plot with ggplot2 and convert it to plotly with ggplotly (2 points)
```



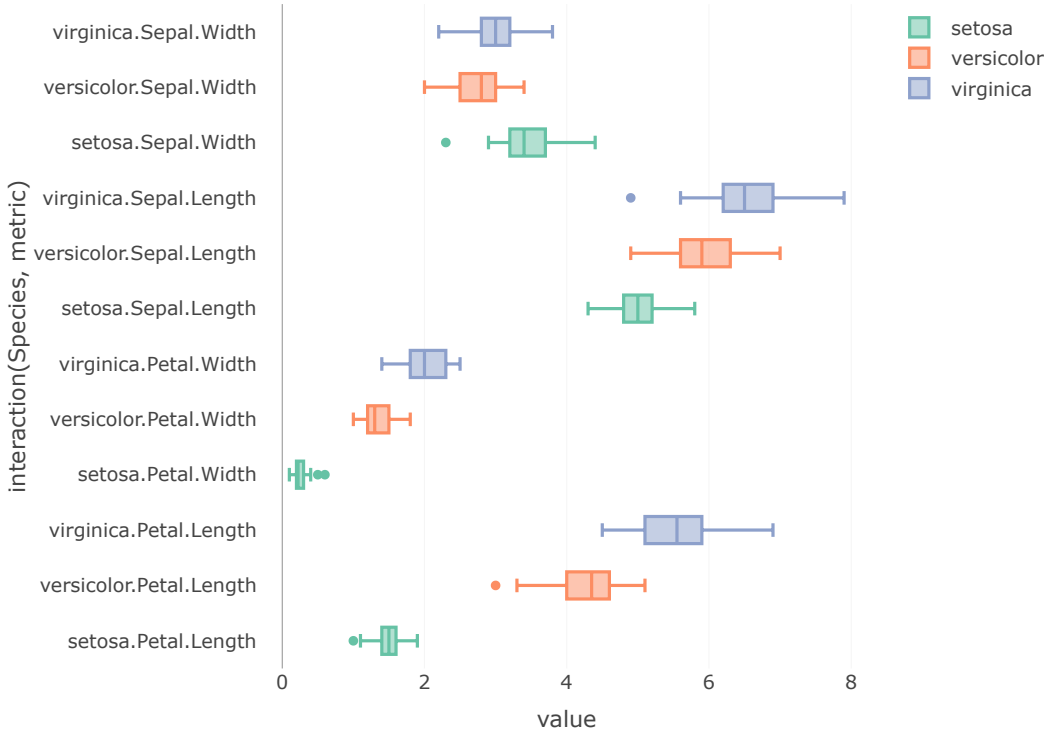

```
# Q1.5. Create facet 2 by 2 plot with histograms similar to previous but for each metric
# (2 points)
# hint:
#   following conversion to long format can be useful:
#   iris %>% gather(key = "metric", value = "value",-Species)
#
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

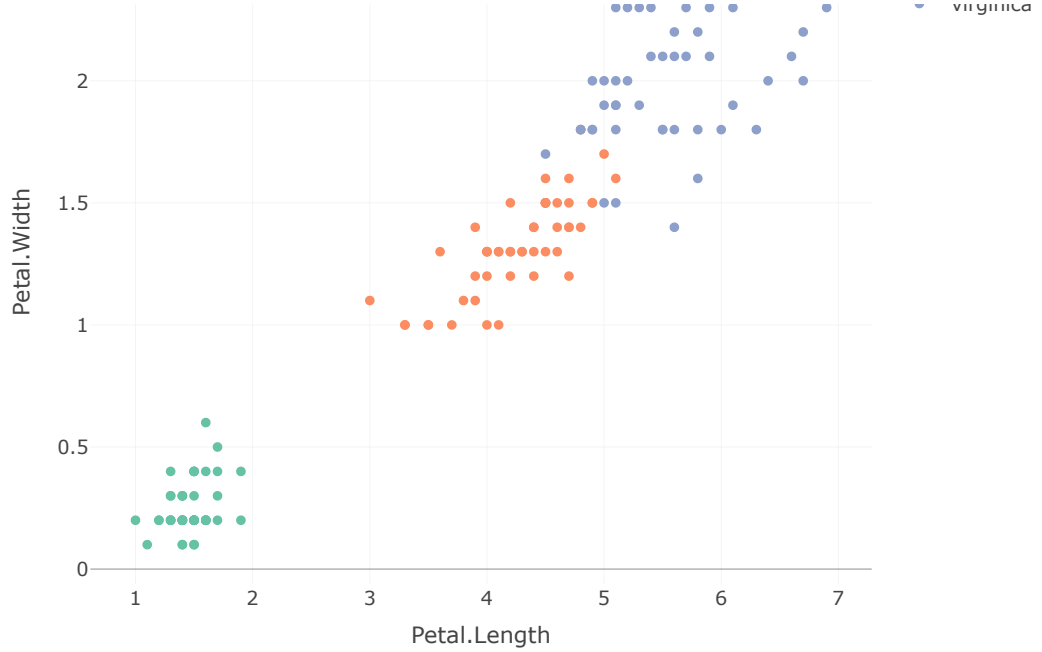


Q1.6. Which metrics has best species separations? (2 points)

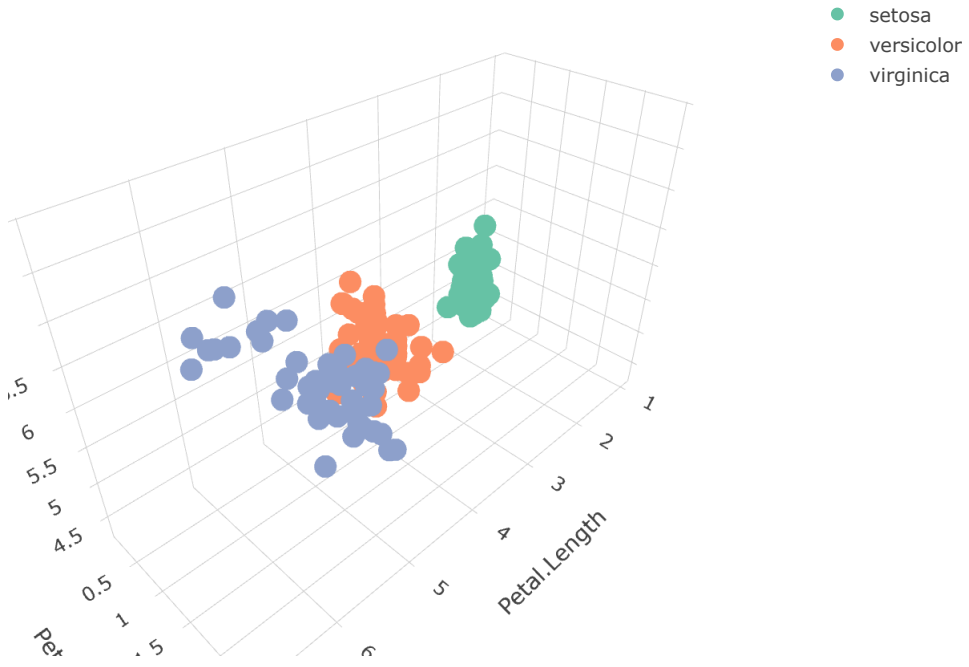
```
# Q1.7. Repeat above plot but using box plot (2 points)
```




```
# Q1.8. Choose two metrics which separates species the most and use it to make scatter plot
# color points by species (2 points)
```




```
# Q1.9. Choose three metrics which separates species the most and use it to make 3d plot
# color points by species (2 points)
```



Q1.10. Comment on species separation (2 points):

Part 2. Covid-19 Dataset. (20 points)

Download us-states.csv (https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-states.csv) (there is also a copy in homework assignment) from https://github.com/nytimes/covid-19-data/ (https://github.com/nytimes/covid-19-data/). README.md (https://github.com/nytimes/covid-19-data/blob/master/README.md) for details on file content.

```
# Q2.1. Read us-states.csv (2 points)
```

```
# Q2.2. Show some values from dataframe (2 points)
```

date	state	fips	cases	deaths
2020-01-21	Washington	53	1	0
2020-01-22	Washington	53	1	0
2020-01-23	Washington	53	1	0
2020-01-24	Illinois	17	1	0
2020-01-24	Washington	53	1	0
2020-01-25	California	6	1	0

```
# Q2.3. Create new dataframe with new cases per month for each state (2 points)
# hint:
#    is cases column cumulative or not cumulative?
```

This table should help with identification of cumulative or not cumulative format of case column

date <date>	state <chr>	fips <int>	cases <int>	deaths <int>
2020-03-01	New York	36	1	0
2020-03-02	New York	36	1	0
2020-03-03	New York	36	2	0
2020-03-04	New York	36	11	0
2020-03-05	New York	36	22	0
2020-03-06	New York	36	44	0
2020-03-07	New York	36	89	0
2020-03-08	New York	36	106	0
2020-03-09	New York	36	142	0
2020-03-10	New York	36	173	0

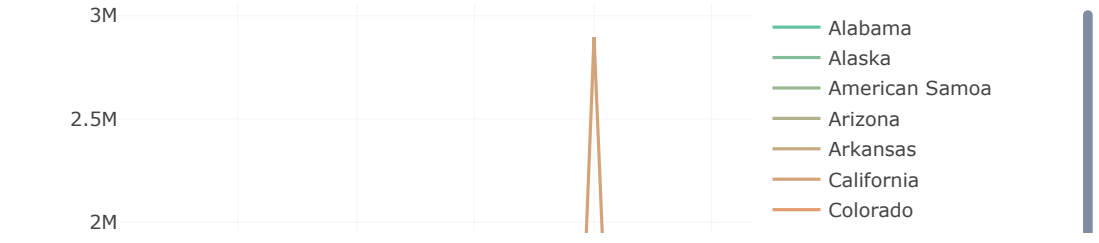
1-10 of 917 rows Previous **1** 2 3 4 5 6 … 92 Next

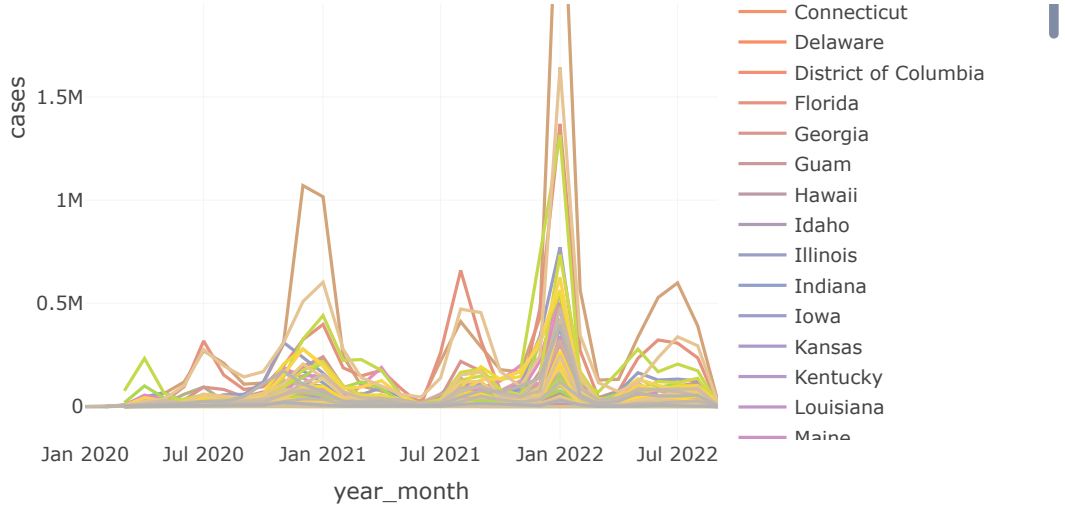
TODO HERE WE NEED TO AKS TO SHOW SOMETHING

```
# Q2.4.Using previous dataframe plot new monthly cases in states, group by states
# The resulting plot is busy, use interactive plotly capabilities to limit number
# of displayed states
# (2 points)
```

```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors

## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```

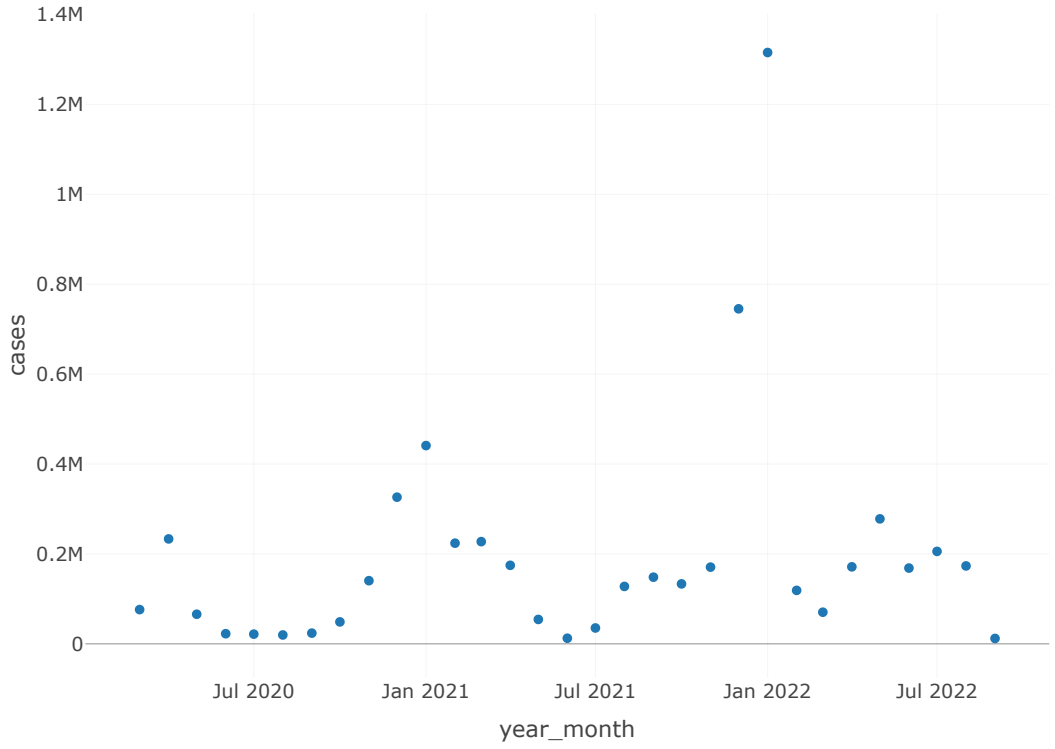





```
# Q2.5.Plot new monthly cases only in NY state
# (2 points)
```

```
## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```

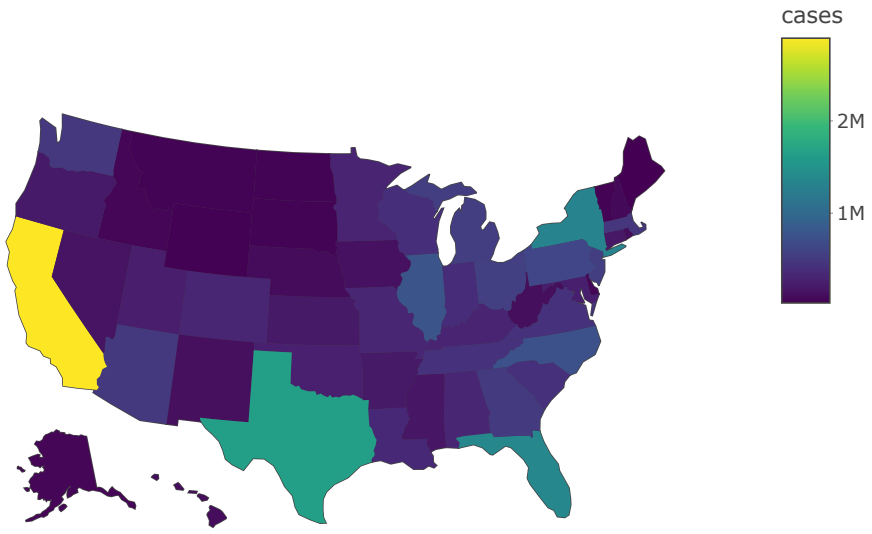



```
# Q2.6. Found the year-month with highest cases in NY state
# (2 points)
```

state <chr>	year_month <date>	fips <int>	cases_cum <int>	deaths_cum <int>	cases <dbl>
New York	2022-01-01	36	4789532	64247	1315562

1 row

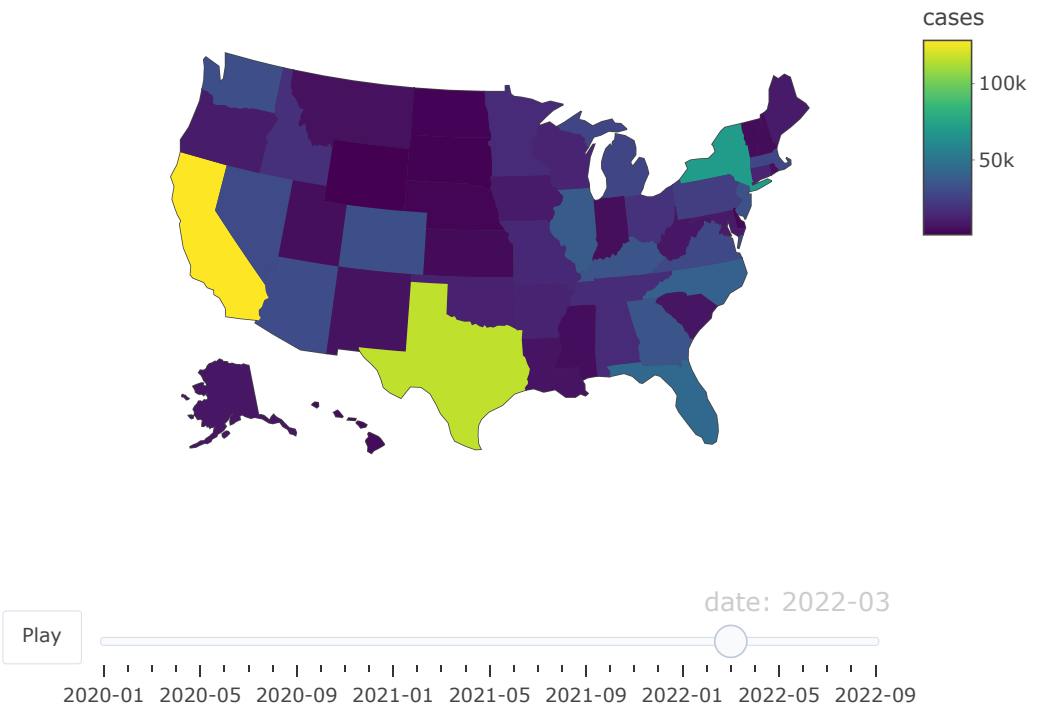
```
# Q2.7. Plot new cases in determined above year-month
# using USA state map, color each state by number of cases  (3 points)
# hint:
#   there two build in constants in R: state.abb and state.name
#   to convert full name to abbreviation
```




```
# Q2.8. Add animation capability (3 points)
# hint:
#     for variable frame you need either integer or character/factorial so
#     convert date to character or factorial
```

state <chr>	year_month <date>	fips <int>	cases_cum <int>	deaths_cum <int>	cases <dbl>	date <chr>	state_abb <chr>
Alabama	2020-03-01	1	999	14	999	2020-03	AL
Alabama	2020-04-01	1	7068	272	6069	2020-04	AL
Alabama	2020-05-01	1	17952	630	10884	2020-05	AL
Alabama	2020-06-01	1	38045	950	20093	2020-06	AL
Alabama	2020-07-01	1	87723	1580	49678	2020-07	AL
Alabama	2020-08-01	1	126058	2182	38335	2020-08	AL
Alabama	2020-09-01	1	154701	2540	28643	2020-09	AL
Alabama	2020-10-01	1	192285	2967	37584	2020-10	AL
Alabama	2020-11-01	1	249524	3578	57239	2020-11	AL
Alabama	2020-12-01	1	361226	4827	111702	2020-12	AL

1-10 of 1,564 rows Previous 1 2 3 4 5 6 … 157 Next



Q2.9. Compare animated plot from Q2.8 to plots from Q2.4/Q2.5 (When you would prefer one or another?) (2 points)