Bioinformatics FYE

Madeleine Duquette (A59019542)

COVID-19 Variants Plot

Install and load ggplot2, lubridate, and dplyr. Will also need to load readr.

```
# install.packages("ggplot2")
# install.packages("lubridate")
# install.packages("dplyr")
```

```
library(ggplot2)
```

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

```
library(lubridate)
```

```
Warning: package 'lubridate' was built under R version 4.2.2
```

```
Loading required package: timechange
```

```
Warning: package 'timechange' was built under R version 4.2.2
```

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

```
The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
library(dplyr)
```

```
Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(readr)
```

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

Read in the csv file

```
variants <- read_csv("covid19_variants.csv")
```

```
Rows: 8840 Columns: 8
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): area, area_type, variant_name
dbl  (4): specimens, percentage, specimens_7d_avg, percentage_7d_avg
date (1): date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Check out what's in there

```
head(variants)
```

```
# A tibble: 6 x 8
  date       area   area_type variant_name specimens percentage specimens_7d_avg
  <date>     <chr>  <chr>     <chr>            <dbl>      <dbl>            <dbl>
1 2021-01-01 Calif~ State     Alpha                1       1.67               NA
2 2021-01-01 Calif~ State     Other               29      48.3                NA
3 2021-01-01 Calif~ State     Delta                0       0                  NA
4 2021-01-01 Calif~ State     Gamma                0       0                  NA
5 2021-01-01 Calif~ State     Omicron              1       1.67               NA
6 2021-01-01 Calif~ State     Total               60     100                  NA
# i 1 more variable: percentage_7d_avg <dbl>
```

I don't need columns 7 and 8 because they do not contain data so I will create a dataframe and then subset columns 1-6

```
# -c(7,8) removes columns 7 and 8 from the datafram created in the first line
df <- data.frame(variants)
df_subset <- df[, -c(7,8)]
head(df_subset)
```

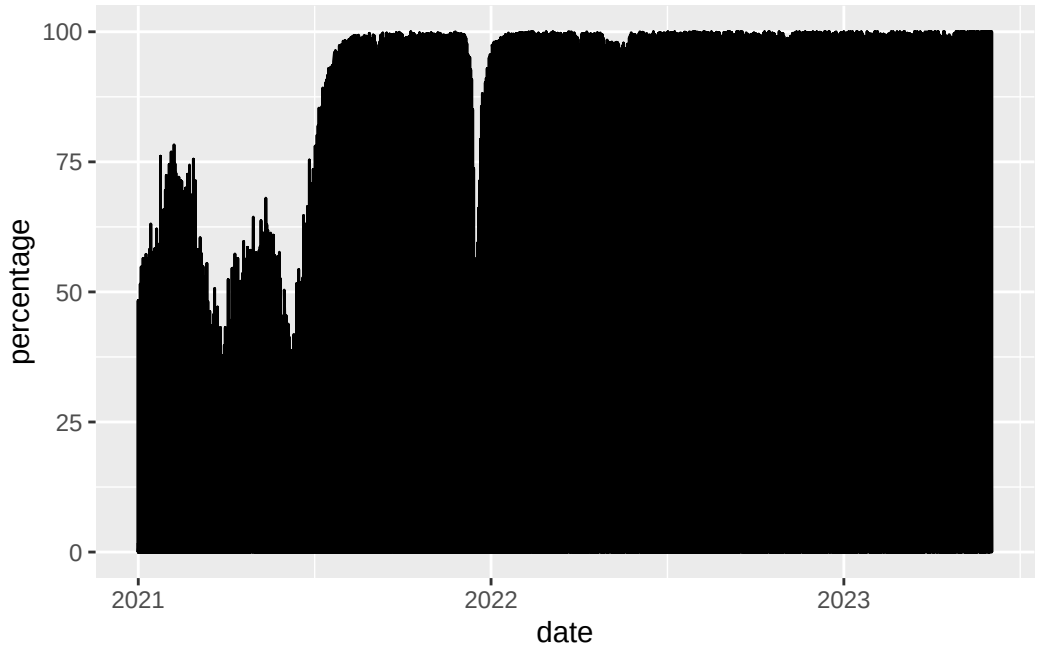
```
       date       area area_type variant_name specimens percentage
1 2021-01-01 California     State        Alpha         1       1.67
2 2021-01-01 California     State        Other        29      48.33
3 2021-01-01 California     State        Delta         0       0.00
4 2021-01-01 California     State        Gamma         0       0.00
5 2021-01-01 California     State      Omicron         1       1.67
6 2021-01-01 California     State        Total        60     100.00
```

The "Total" data is not useful since it will always be 100%. So I will filter it out of my dataset.

```
# filtering the subset dataframe so that rows with variant_name != "Total" are removed
df_nototal <- filter(df_subset, variant_name != "Total")
```

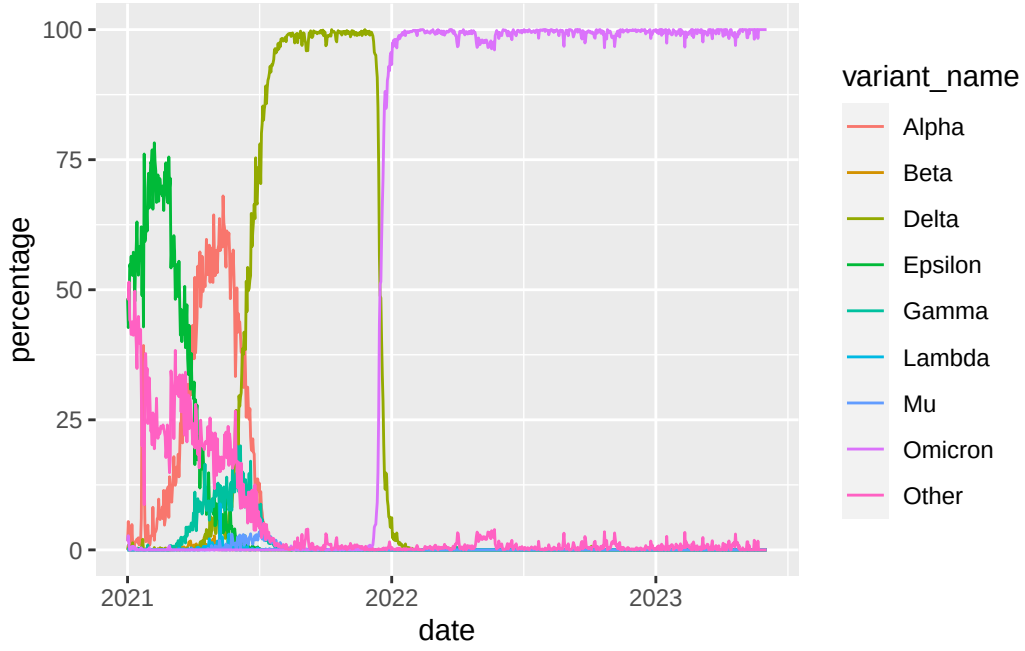
Now let's try graphing something!

```
# setting x axis to date column and y axis equal to percentage column
ggplot(data=df_nototal) +
  aes(x=date, y=percentage) +
  geom_line()
```



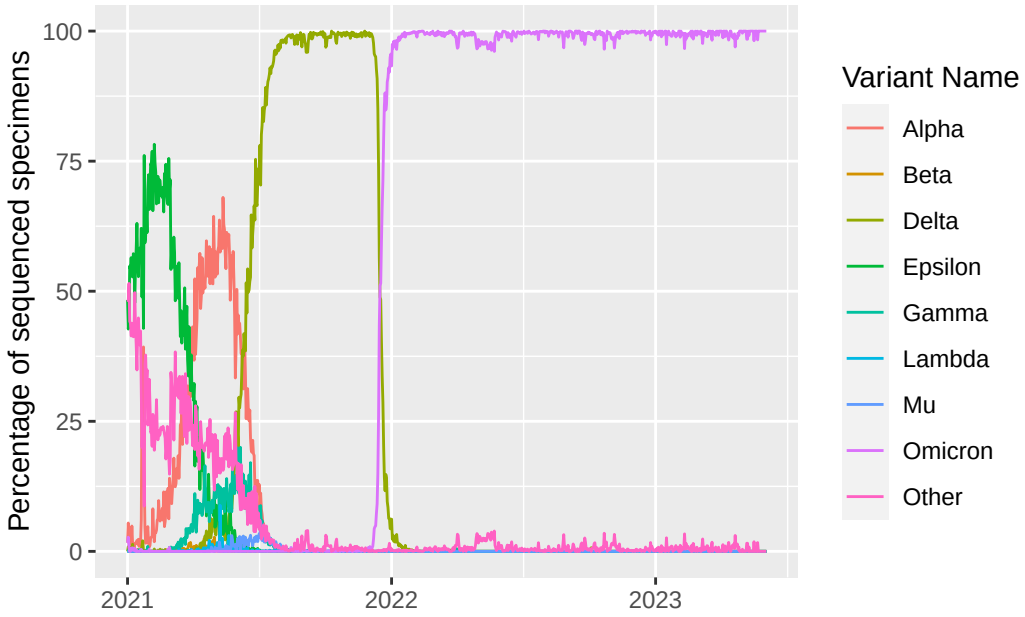
Not much to see there... Let's try changing some colors by the variant strain type

```
# use col=variant_name to color the lines by variant strain
ggplot(data=df_nototal) +
  aes(x=date, y=percentage, col=variant_name) +
  geom_line()
```



Much better! Let's add labels.

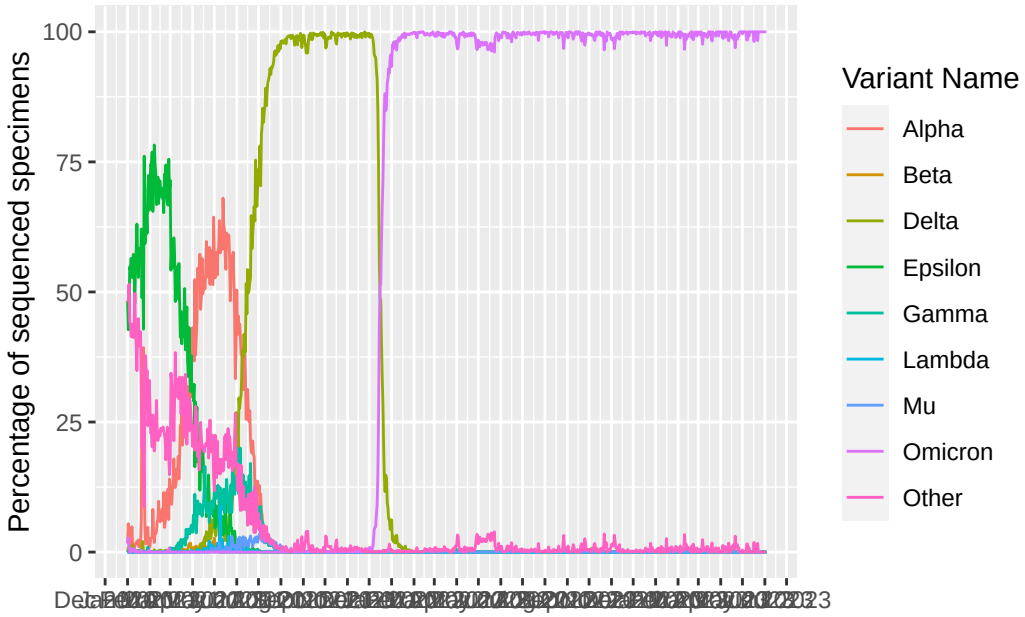
```
# adding labels with labs() and then using xlab("") to remove the unnecessary label on the x-axis beca
ggplot(data=df_nototal) +
  aes(x=date, y=percentage, col=variant_name) +
  geom_line() +
  labs(y="Percentage of sequenced specimens", color="Variant Name") +
  xlab("")
```



We will now add some granularity to the x-axis by scaling by month instead of by year.

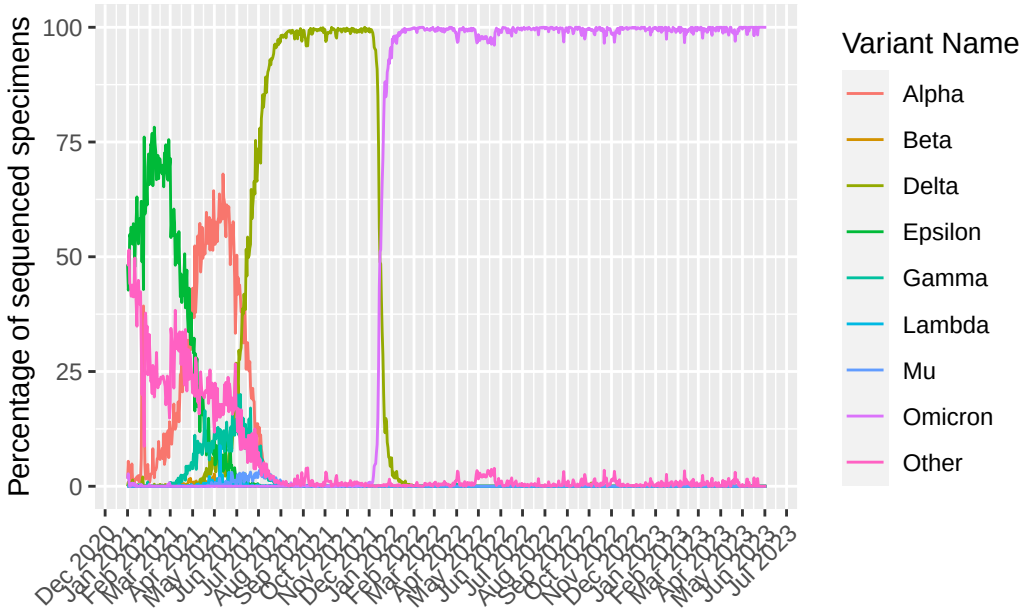
```
# using scale_x_date() to change the x axis ticks
ggplot(data=df_nototal) +
  aes(x=date, y=percentage, col=variant_name) + geom_line() +
```

```
  labs(y="Percentage of sequenced specimens", color="Variant Name") +
  xlab("") +
  scale_x_date(date_labels="%b %Y", date_breaks="1 month")
```



Well, that's definitely not readable. Let's try angling the labels!

```
# using theme() to change the x axis text to be at a 45 degree angle and in line with the ticks
ggplot(data=df_nototal) +
  aes(x=date, y=percentage, col=variant_name) +
  geom_line() +
  labs(y="Percentage of sequenced specimens", color="Variant Name") +
  xlab("") +
  scale_x_date(date_labels="%b %Y", date_breaks="1 month") +
  theme(axis.text.x=element_text(angle=45, hjust=1))
```



Yay!