Bioinformatics First Year Exam Q15

Mirte Ciz Marieke Kuijpers

2022-06-15

Set Up

First packages and the data need to be loaded into the R working environment.

```
# Load data
data <- read.csv("covid19_variants.csv", header = T)

# Check data
head(data, n = 2)
```

```
##         date       area area_type variant_name specimens percentage
## 1 2021-01-01 California     State        Total        59     100.00
## 2 2021-01-01 California     State        Alpha         1       1.69
##   specimens_7d_avg percentage_7d_avg
## 1               NA                NA
## 2               NA                NA
```

```
tail(data, n = 2)
```

```
##            date       area area_type variant_name specimens percentage
## 5059 2022-05-21 California     State       Lambda         0          0
## 5060 2022-05-21 California     State        Alpha         0          0
##      specimens_7d_avg percentage_7d_avg
## 5059                0                 0
## 5060                0                 0
```

```
summary(data)
```

```
##      date               area            area_type         variant_name
##  Length:5060        Length:5060        Length:5060        Length:5060
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    specimens         percentage     specimens_7d_avg   percentage_7d_avg
##  Min.   :   0.00   Min.   :  0.00   Min.   :   0.000   Min.   :  0.0000
##  1st Qu.:   0.00   1st Qu.:  0.00   1st Qu.:   0.000   1st Qu.:  0.0000
##  Median :   0.00   Median :  0.00   Median :   0.571   Median :  0.0862
##  Mean   : 179.93   Mean   : 20.00   Mean   : 181.336   Mean   : 20.0000
##  3rd Qu.:  37.25   3rd Qu.: 13.51   3rd Qu.:  40.143   3rd Qu.: 12.7339
##  Max.   :5776.00   Max.   :100.00   Max.   :3255.429   Max.   :100.0000
##                                     NA’s   :60         NA’s   :60
```

```
# Load packages
library("ggplot2")
```

```
## Warning: package ’ggplot2’ was built under R version 4.1.3
```

```
library("dplyr")
```

```
## Warning: package ’dplyr’ was built under R version 4.1.3
```

```
##
## 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("lubridate")
```

```
##
## Attaching package: ’lubridate’
```

```
## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

Format data

Note that `summary` shows us that `R` is treating the date column data as character data. This should be changed so that `R` can recognize this as time/date information.

```
# State of date before fixing
class(data$date)
```

```
## [1] "character"
```

```
# Use a Lubridate function to tell R that the date column holds dates not characters
data$date <- ymd(data$date)

# Check this was successful
class(data$date)
```

```
## [1] "Date"
```

Another thing to check is whether the variants present in the data match those in the plot I am trying to recreate, and remove any inappropriate categories.

```
# Check the variants present in the data
unique(data$variant_name)
```

```
##  [1] "Total"   "Alpha"   "Beta"    "Mu"      "Delta"   "Gamma"   "Lambda" 
##  [8] "Epsilon" "Omicron" "Other"
```

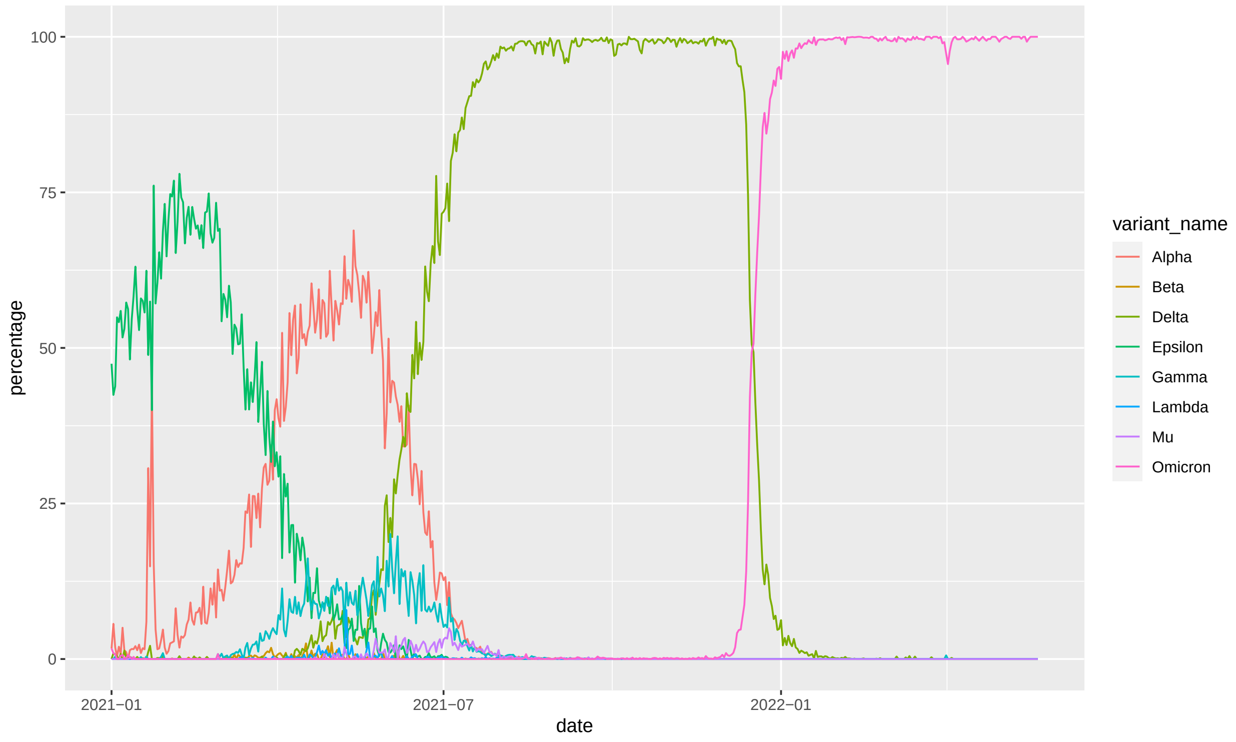
```
# Remove "Total" and "Other", which are not in the template plot
#### not (!) rows with Other or Total in the variant column
dat <- data[!data$variant_name %in% c("Other", "Total"), ]

# Check if successful
unique(dat$variant_name)
```

```
## [1] "Alpha"   "Beta"    "Mu"      "Delta"   "Gamma"   "Lambda"  "Epsilon"
## [8] "Omicron"
```

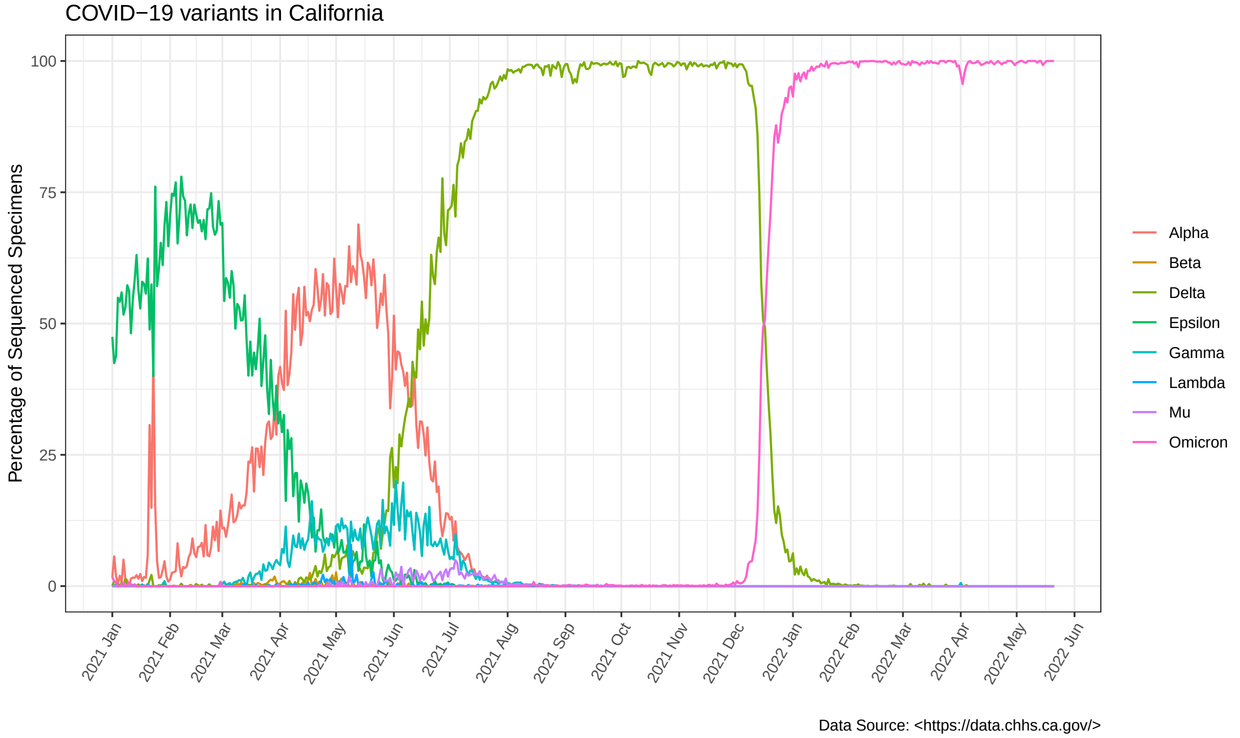
Plot data

Now that the data has been properly formatted it is possible to plot it. I will focus on first plotting the data correctly, and when this is complete I will make a secondary plot with appropriate aesthetics.

```
# Basic plot
ggplot(dat, aes(date, percentage, col = variant_name)) +
  geom_line()
```



```r
# Plot with aesthetics
ggplot(dat, aes(date, percentage, col = variant_name)) +
  geom_line(cex = 0.6) +
  theme_bw() + # basic theme to build upon
  scale_x_date(date_labels = "%Y %b", date_breaks = "1 month") + # format axis
  theme(axis.text.x=element_text(angle=60, hjust=1)) + # customize bw theme
  labs(x = "", y = "Percentage of Sequenced Specimens",
       title = "COVID-19 variants in California", col = "",
       caption = "Data Source: <https://data.chhs.ca.gov/>")
```



COVID−19 variants in California

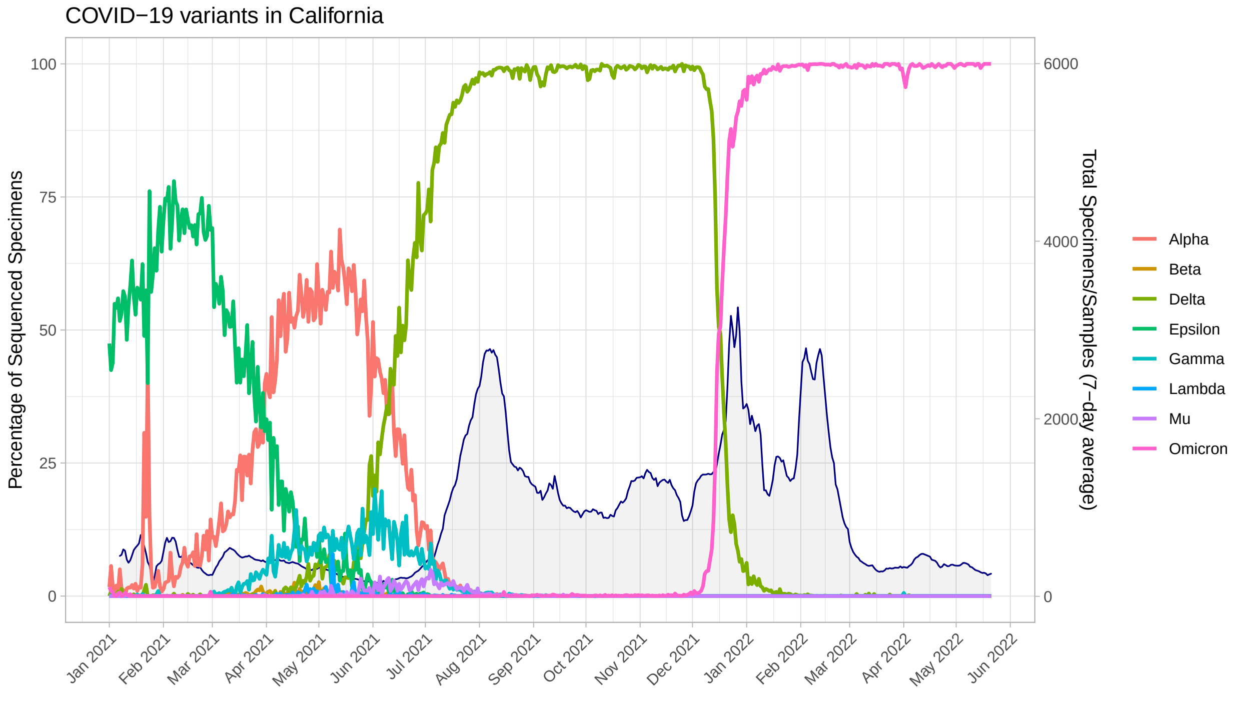
Data Source: <https://data.chhs.ca.gov/>

Note that within the “{r}” of this code chunk I added the following `fig.width=10, fig.height= 6` to ensure that the plots were knitted at a reasonable size into the final document. Without this customization they were a little smaller than I thought they should be.

Now that the template plot has been replicated I can also play around with my own styles.

```r
# Subset the data for just the total numbers
tot <- data[data$variant_name == "Total",]

# My plot
ggplot(dat, aes(x = date, col = variant_name)) +
  geom_area(data = tot, aes(x = date, y = specimens_7d_avg/60), col = "navy",
            cex = 0.45, alpha = 0.05, fill = "black") +
  geom_line(aes(y = percentage), cex = 1) +
  scale_y_continuous(
    # Features of the first axis
    name = "Percentage of Sequenced Specimens",
    # Add a second axis and specify its features
    sec.axis = sec_axis( trans=~.*60, name="Total Specimens/Samples (7-day average)")
  ) +
  theme_light() + # basic theme to build upon
  scale_x_date(date_labels = "%b %Y", date_breaks = "1 month") + # format axis
  theme(axis.text.x=element_text(angle=45, hjust=1)) + # customize bw theme
  labs(x = "", title = "COVID-19 variants in California", col = "",
       caption = "Data Source: <https://data.chhs.ca.gov/>")
```

```
## Warning: Removed 6 rows containing missing values (position_stack).
```



Data Source: <https://data.chhs.ca.gov/>

Session Information

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
```

```
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] lubridate_1.8.0 dplyr_1.0.9     ggplot2_3.3.6
##
## loaded via a namespace (and not attached):
##  [1] highr_0.9        pillar_1.7.0     compiler_4.1.2   tools_4.1.2
##  [5] digest_0.6.29    evaluate_0.15    lifecycle_1.0.1  tibble_3.1.7
##  [9] gtable_0.3.0     pkgconfig_2.0.3  rlang_1.0.2      cli_3.2.0
## [13] DBI_1.1.2        rstudioapi_0.13  yaml_2.3.5       xfun_0.29
## [17] fastmap_1.1.0    withr_2.5.0      stringr_1.4.0    knitr_1.39
## [21] generics_0.1.2   vctrs_0.4.1      grid_4.1.2       tidyselect_1.1.2
## [25] glue_1.6.2       R6_2.5.1         fansi_1.0.3      rmarkdown_2.14
## [29] farver_2.1.0     purrr_0.3.4      magrittr_2.0.3   scales_1.2.0
## [33] ellipsis_0.3.2   htmltools_0.5.2  assertthat_0.2.1 colorspace_2.0-3
## [37] labeling_0.4.2   utf8_1.2.2       stringi_1.7.6    munsell_0.5.0
## [41] crayon_1.5.1
```