Zeyd Khalil, HW10, October 29, 2020

Exercise 1 - create a tibble that has both deaths and total cases per state, arranged by the total number of deaths in descending order

```
state.1 <-
  dat %>%
  group_by(state, date) %>%
  summarize(total_deaths = sum(deaths), total_cases = sum(cases))
```

```
## ‘summarise()‘ regrouping output by ’state’ (override with ‘.groups‘ argument)
```

```
state.1
```

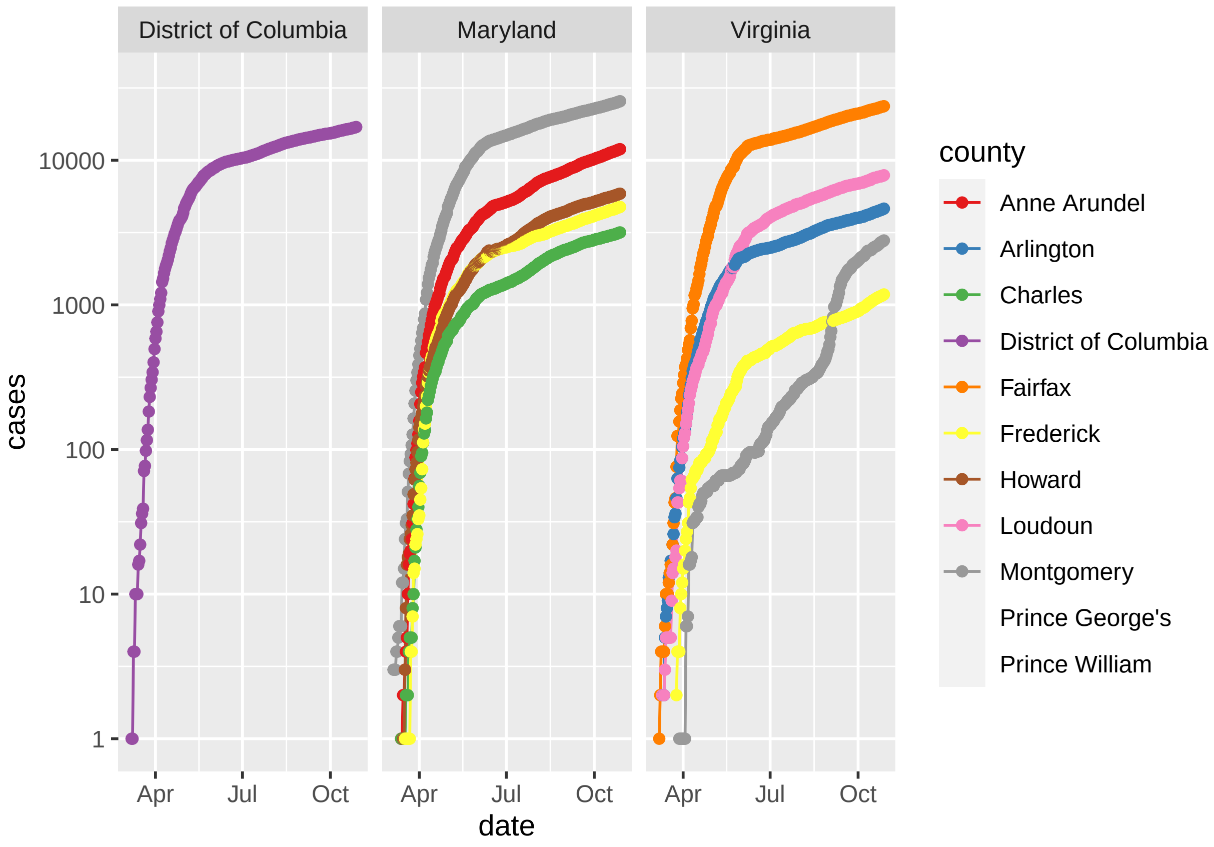
```
## # A tibble: 13,214 x 4
## # Groups:   state [55]
##    state   date       total_deaths total_cases
##    <chr>   <date>            <dbl>       <dbl>
##  1 Alabama 2020-03-13            0           6
##  2 Alabama 2020-03-14            0          12
##  3 Alabama 2020-03-15            0          23
##  4 Alabama 2020-03-16            0          29
##  5 Alabama 2020-03-17            0          39
##  6 Alabama 2020-03-18            0          51
##  7 Alabama 2020-03-19            0          78
##  8 Alabama 2020-03-20            0         106
##  9 Alabama 2020-03-21            0         131
## 10 Alabama 2020-03-22            0         157
## # ... with 13,204 more rows
```

```
deaths <-
  state.1 %>%
  group_by(state) %>%
  filter(total_deaths == max(total_deaths), total_cases == max(total_cases)) %>%
  arrange(desc(total_deaths)) %>%
  distinct()

deaths
```

```
## # A tibble: 55 x 4
## # Groups:   state [53]
##    state          date       total_deaths total_cases
##    <chr>          <date>            <dbl>       <dbl>
##  1 New York       2020-10-28        33107      505416
##   2 Texas         2020-10-28        18251      931113
##   3 California    2020-10-28        17541      922680
##   4 Florida       2020-10-28        16570      790418
##   5 New Jersey    2020-10-28        16324      234790
##   6 Massachusetts 2020-10-28         9924      154218
##   7 Illinois      2020-10-28         9912      395204
##   8 Pennsylvania  2020-10-28         8789      205852
##   9 Georgia       2020-10-28         7692      367126
##  10 Michigan      2020-10-28         7606      185818
## # ... with 45 more rows
```

Exercise 2 -

```
dat_dmv <- dat %>%
  filter(state == "District of Columbia" | state == "Virginia" | state == "Maryland", county == "Anne A

dat_dmv %>%
   ggplot(aes(x = date, y = cases, group = county, col = county)) +
   geom_line() +
   geom_point() +
   facet_wrap(~ state) +
   scale_y_log10() +
   scale_color_brewer(palette = "Set1")
```

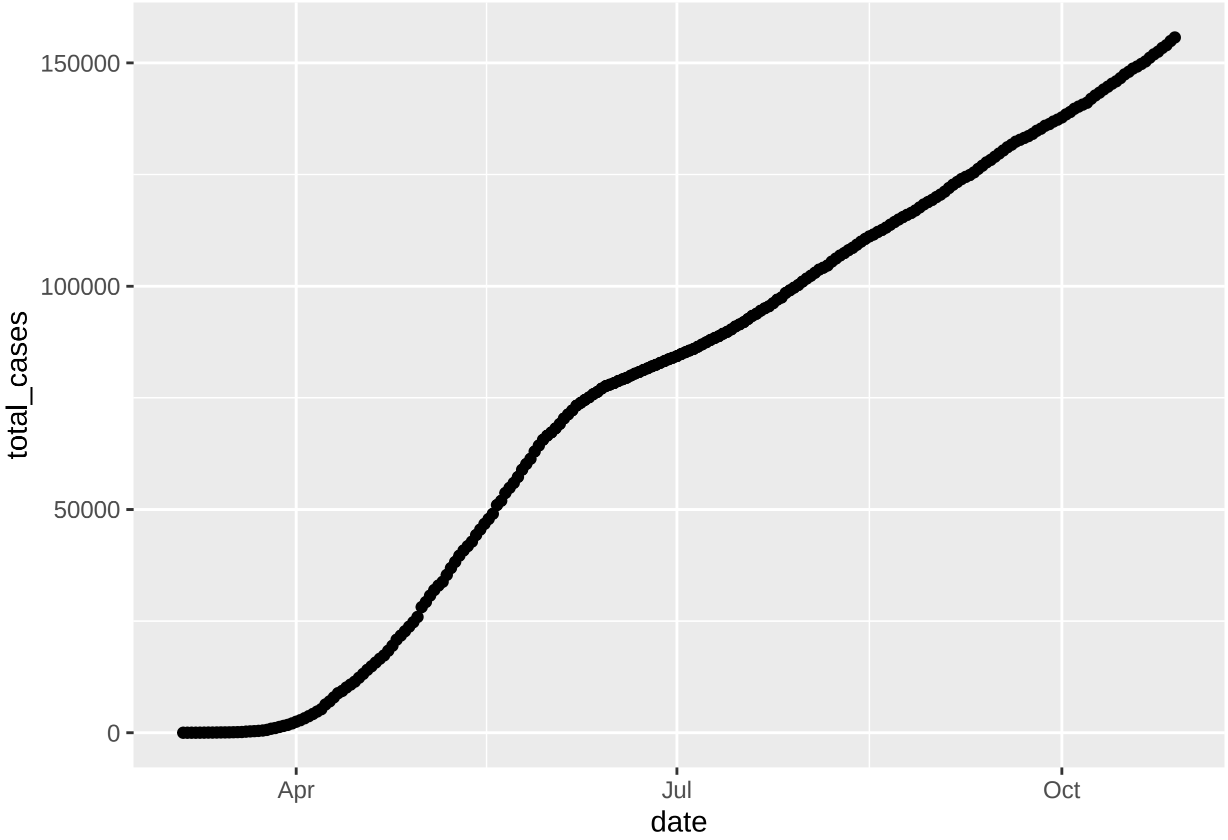


Exercise 3 -

```
dmv_total_cases <-
  dat_dmv %>%
  group_by(date) %>%
  summarize(total_cases = sum(cases))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

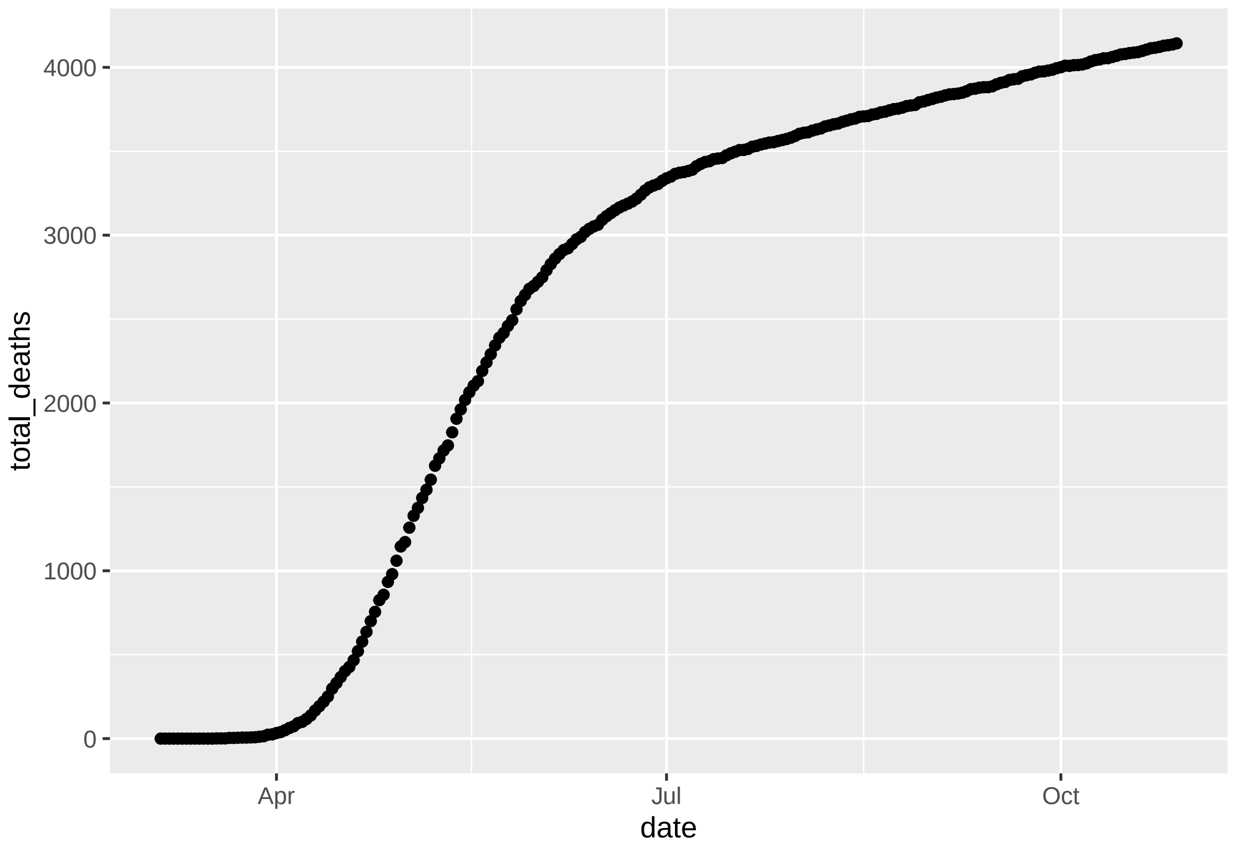
```
dmv_total_cases %>%
  ggplot(aes(x = date, y = total_cases)) +
  geom_point()
```



```
dmv_total_deaths <-
  dat_dmv %>%
  group_by(date) %>%
  summarise(total_deaths = sum(deaths))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
dmv_total_deaths %>%
  ggplot(aes(x = date, y = total_deaths)) +
  geom_point()
```



Ask TA About this tomorrow

Exercise 4 - Read in and tidy both the global and US datasets. For the US data, produce a second tidy dataset called US_by_state that has the total of Confirmed cases, deaths and population for each date for each state.

```
cases_global <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_1
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   ‘Province/State‘ = col_character(),
##   ‘Country/Region‘ = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
deaths_global <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   ‘Province/State‘ = col_character(),
##   ‘Country/Region‘ = col_character()
## )
## See spec(...) for full column specifications.
```

```
cases_us <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_dat
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   iso2 = col_character(),
##   iso3 = col_character(),
##   Admin2 = col_character(),
##   Province_State = col_character(),
##   Country_Region = col_character(),
##   Combined_Key = col_character()
## )
## See spec(...) for full column specifications.
```

```
deaths_us <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_da
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   iso2 = col_character(),
##   iso3 = col_character(),
##   Admin2 = col_character(),
##   Province_State = col_character(),
##   Country_Region = col_character(),
##   Combined_Key = col_character()
## )
## See spec(...) for full column specifications.
```

Tidying the datasets

```
cases_global <-
  cases_global %>%
  pivot_longer(cols = c(‘1/22/20‘:‘10/28/20‘), names_to = "date", values_to = "cases")

deaths_global <-
  deaths_global %>%
  pivot_longer(cols = c(‘1/22/20‘:‘10/28/20‘), names_to = "date", values_to = "Deaths")

cases_us <-
  cases_us %>%
  pivot_longer(cols = c(‘1/22/20‘:‘10/28/20‘), names_to = "date", values_to = "cases")

deaths_us <-
  deaths_us %>%
  pivot_longer(cols = c(‘1/22/20‘:‘10/28/20‘), names_to = "date", values_to = "Deaths")
```

```
global <- cases_global %>% full_join(deaths_global) %>%
  rename(Country_Region = ‘Country/Region‘, Province_State = ‘Province/State‘)
```

```
## Joining, by = c("Province/State", "Country/Region", "Lat", "Long", "date")
```

```
global
```

```
## # A tibble: 75,308 x 7
##    Province_State Country_Region   Lat  Long date    cases Deaths
##    <chr>          <chr>          <dbl> <dbl> <chr>   <dbl>  <dbl>
##  1 <NA>           Afghanistan     33.9  67.7 1/22/20     0      0
##  2 <NA>           Afghanistan     33.9  67.7 1/23/20     0      0
##  3 <NA>           Afghanistan     33.9  67.7 1/24/20     0      0
##  4 <NA>           Afghanistan     33.9  67.7 1/25/20     0      0
##  5 <NA>           Afghanistan     33.9  67.7 1/26/20     0      0
##  6 <NA>           Afghanistan     33.9  67.7 1/27/20     0      0
##  7 <NA>           Afghanistan     33.9  67.7 1/28/20     0      0
##  8 <NA>           Afghanistan     33.9  67.7 1/29/20     0      0
##  9 <NA>           Afghanistan     33.9  67.7 1/30/20     0      0
## 10 <NA>           Afghanistan     33.9  67.7 1/31/20     0      0
## # ... with 75,298 more rows
```

```
US <- deaths_us %>%
  full_join(cases_us, by = c("Combined_Key", "date", "Admin2", "Province_State", "Country_Region")) %>%
    rename(Long = Long_.x, Lat = Lat.x)  %>%
    select(Admin2, Province_State, Country_Region, Lat, Long, Population, date, cases, Deaths)
```

```
US_by_state <- US %>% group_by(Province_State, Country_Region, date) %>% summarize(cases = sum(cases), 
```

```
## ‘summarise()‘ regrouping output by ’Province_State’, ’Country_Region’ (override with ‘.groups‘ argume
```

```
US_by_state
```

```
## # A tibble: 16,298 x 9
##    Province_State Country_Region date  cases Deaths Deaths_per_mill Population
##    <chr>          <chr>          <chr> <dbl>  <dbl>           <dbl>      <dbl>
##  1 Alabama        US             1/22~     0      0               0    4903185
##  2 Alabama        US             1/23~     0      0               0    4903185
##  3 Alabama        US             1/24~     0      0               0    4903185
##  4 Alabama        US             1/25~     0      0               0    4903185
##  5 Alabama        US             1/26~     0      0               0    4903185
##  6 Alabama        US             1/27~     0      0               0    4903185
##  7 Alabama        US             1/28~     0      0               0    4903185
##  8 Alabama        US             1/29~     0      0               0    4903185
##  9 Alabama        US             1/30~     0      0               0    4903185
## 10 Alabama        US             1/31~     0      0               0    4903185
## # ... with 16,288 more rows, and 2 more variables: Lat <dbl>, Long <dbl>
```

Exercise 5 - Replace the US observations in the global dataset with the US data. Add a new variable called continent to the dataset. Be sure there are no NA's for continent. Also create a new variable Country_State that comines the Province_State with Country_Region.

```
## Joining, by = c("Province_State", "Country_Region", "date", "cases", "Deaths", "Lat", "Long")
```

Exercise 7 -

```
top_25 <- cases_global %>% select(‘Country/Region‘, cases) %>% group_by(‘Country/Region‘) %>% summarize
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```