Stat. 651 Section 1 or 2: Homework 1

Prof. Eric A. Suess

So how should you complete your homework for this class?

- First thing to do is type all of your information about the problems you complete in the text part of your Quarto Notebook.
- Second thing to do is type all of your R code into R chunks that can be run.
- If you load the tidyverse in an R Notebook chunk, be sure to include the "message = FALSE" in the {r}, so {r message = FALSE}.
- Last thing is to spell check your R Notebook. Edit > Check Spelling... or hit the F7 key.

Using an Quarto Notebook produce your solutions to the following questions. Start by making an Quarto Notebook with file name Lastname_Firstname_Stat651_hw1.qmd. Then Render the .qmd file to a Lastname_Firstname_Stat651_hw1.pdf. Use your own last name and first name in the filename. At the top of your first page you should include Name, Class, Section, and homework assignment.

The header of your R Notebooks should include

title: "Stat. 651 Homework 1"

author: "Your name"

date: " October 17, 2022"

Upload one file to Canvas.

Homework 1:

- Chapter 3 Exercises Problem 4, 8, 9, 10

Problem 4

First note that the temperature variable is the average temperature, in Celsius.

```
library(tidyverse)
```

```
library(macleish)
```

```
Loading required package: etl
```

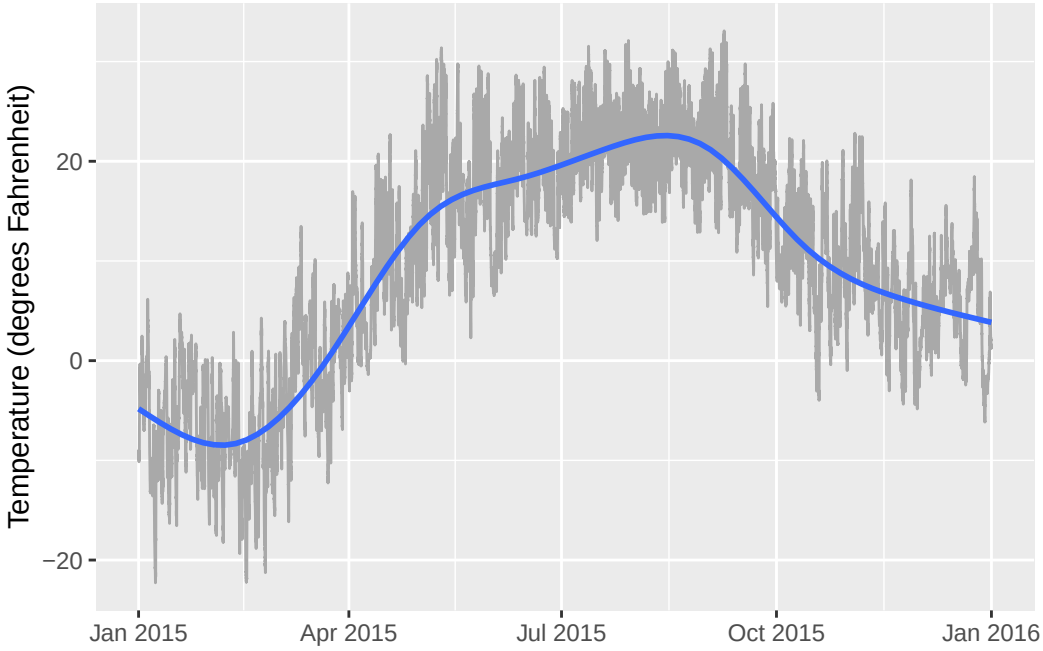
```
whately_2015
```

```
# A tibble: 52,560 x 8
   when                temperature wind_speed wind_dir rel_humidity pressure
   <dttm>                    <dbl>      <dbl>    <dbl>        <dbl>    <int>
 1 2015-01-01 00:00:00       -9.32       1.40     225.         54.6      985
 2 2015-01-01 00:10:00       -9.46       1.51     248.         55.4      985
 3 2015-01-01 00:20:00       -9.44       1.62     258.         56.2      985
 4 2015-01-01 00:30:00       -9.3        1.14     244.         56.4      985
 5 2015-01-01 00:40:00       -9.32       1.22     238.         56.9      984
 6 2015-01-01 00:50:00       -9.34       1.09     242.         57.2      984
 7 2015-01-01 01:00:00       -9.3        1.17     242.         57.7      984
 8 2015-01-01 01:10:00       -9.1        1.31     244.         58.2      984
 9 2015-01-01 01:20:00       -9.07       1.31     226.         59.0      984
10 2015-01-01 01:30:00       -8.99       1.81     220          59.3      984
# ... with 52,550 more rows, and 2 more variables: solar_radiation <dbl>,
#   rainfall <dbl>
```

```
?whately_2015
```

```
whately_2015 %>% ggplot(aes(x = when, y = temperature)) +
  geom_line(color = "darkgrey") +
  geom_smooth() +
  xlab(NULL) +
  ylab("Temperature (degrees Fahrenheit)")
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Problem 8

Which dataframe to use? storms

```
library(nasaweather)
```

```
Attaching package: 'nasaweather'

The following object is masked from 'package:dplyr':

    storms
```

```
head(atmos, 2)                # Atmospheric data.
```

```
# A tibble: 2 x 11
    lat  long  year month surftemp  temp pressure ozone cloudlow cloudmid
  <dbl> <dbl> <int> <int>    <dbl> <dbl>    <dbl> <dbl>    <dbl>    <dbl>
1  36.2 -114.  1995     1     273.  272.      835   304      7.5     34.5
2  33.7 -114.  1995     1     280.  282.      940   304     11.5     32.5
# ... with 1 more variable: cloudhigh <dbl>
```

```r
borders <- ungroup(borders)  # Country borders, ungroup fixes a problem with knitting.
head(elev, 2)                # Elevation.
```

```
# A tibble: 2 x 3
   long   lat  elev
  <dbl> <dbl> <dbl>
1 -114. -21.2     0
2 -114. -18.7     0
```

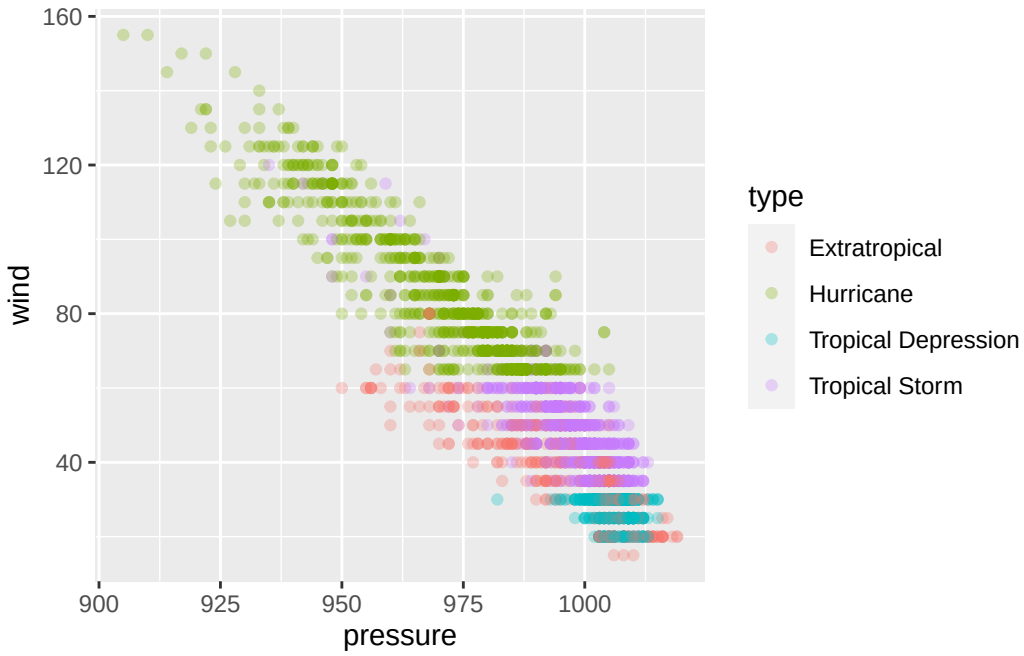
```r
head(glaciers, 2)            # Glacier locations
```

```
# A tibble: 2 x 6
  id          name          lat  long area  country
  <chr>       <chr>       <dbl> <dbl> <chr> <chr>
1 CO1A0101001 RAMIREZ E 4  10.8 -73.6 " NA" CO
2 CO1A0101002 RAMIREZ E 3  10.8 -73.6 " NA" CO
```

```r
head(storms, 2)              # Storm tracks data
```

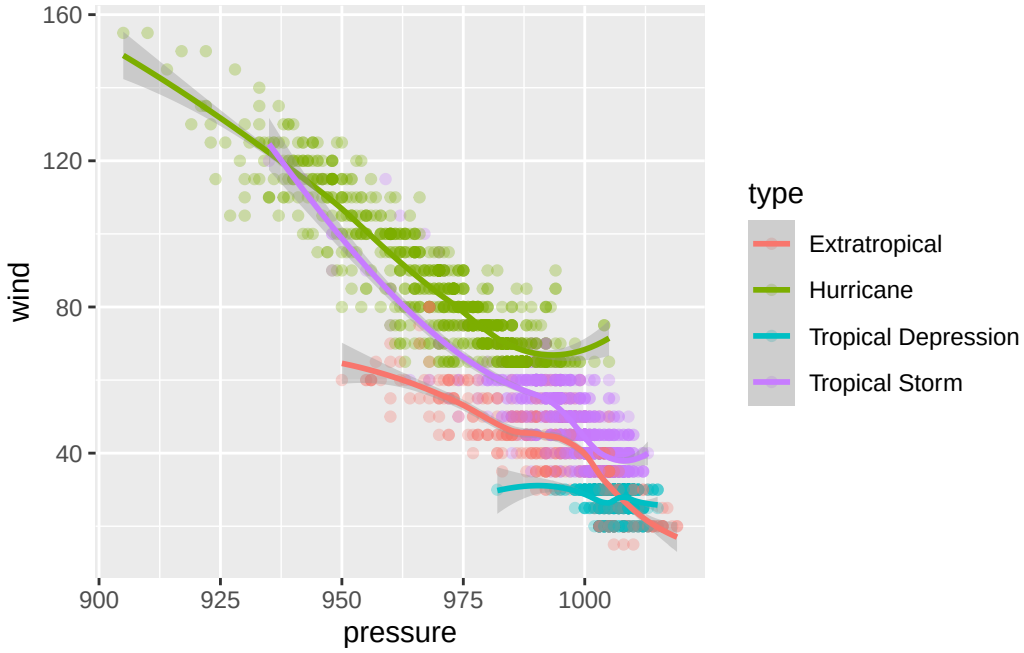
```
# A tibble: 2 x 11
  name     year month   day  hour   lat  long pressure  wind type        seasday
  <chr>   <int> <int> <int> <int> <dbl> <dbl>    <int> <int> <chr>         <int>
1 Allison  1995     6     3     0  17.4 -84.3     1005    30 Tropical D~       3
2 Allison  1995     6     3     6  18.3 -84.9     1004    30 Tropical D~       3
```

```r
p <- storms %>% ggplot(aes(y = wind, x = pressure, color = type)) +
  geom_point(alpha = 0.3)

p
```




```
p + geom_smooth()
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



This solution includes how to plot the paths on a map, while the question does not ask for this.

All storms

```
library(purrr)

bbox <- storms %>%
  select(lat, long) %>%
  map_df(range)
bbox
```

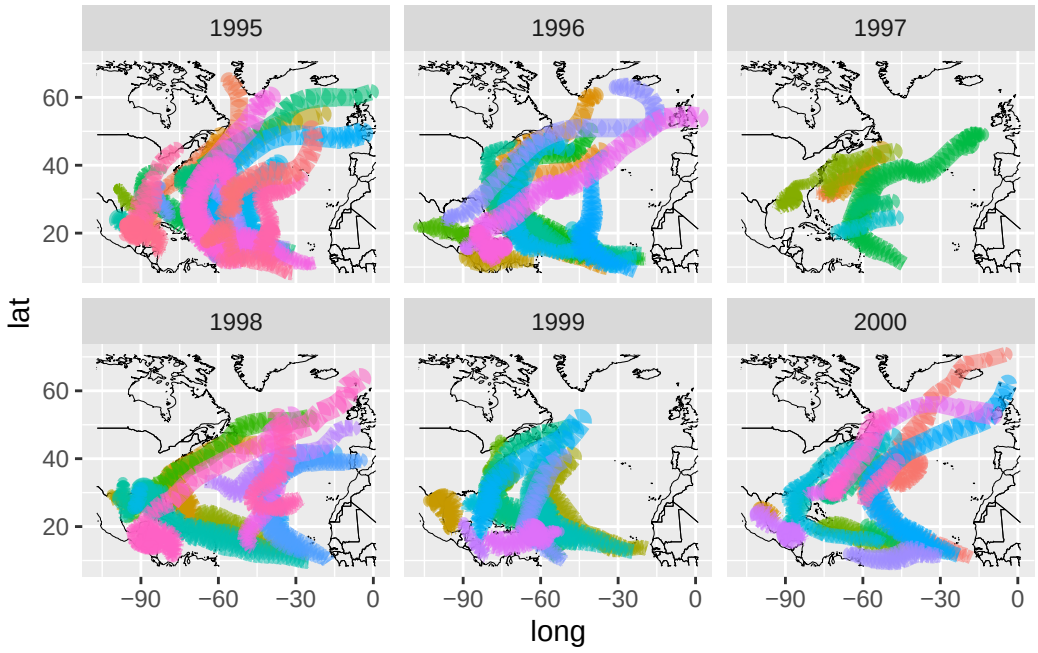
```
# A tibble: 2 x 2
    lat  long
  <dbl> <dbl>
1   8.3 -107.
2  70.7    1
```

```
base_map <- map_data("world") %>% ggplot( aes(x = long, y = lat)) +
  geom_path(aes(group = group), color = "black", size = 0.1) +
  lims(x = bbox$long, y = bbox$lat)

library(lubridate)
```

```
Attaching package: 'lubridate'

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

    date, intersect, setdiff, union
```

```
storms2 <- storms %>%
  unite(the_date, c(year, month, day), sep = "-", remove = FALSE) %>%
  mutate(the_date = ymd(the_date))

base_map <- base_map + geom_path(data = storms2, aes(color = name, alpha = 0.01, size = wi
                      arrow = arrow(length = unit(0.005, "inches"))) +
  facet_wrap(~year)

base_map + theme(legend.position = "none")
```

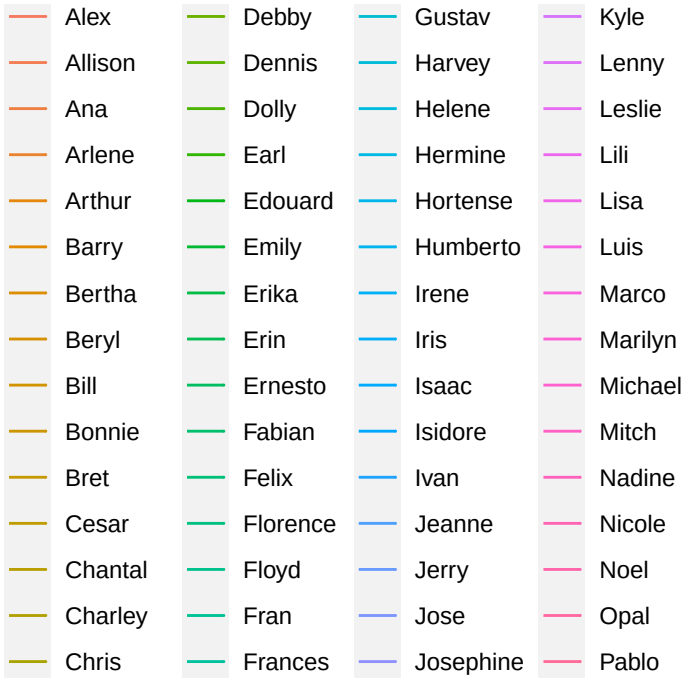
```
Warning: Removed 454346 row(s) containing missing values (geom_path).
```




```
legend <- cowplot::get_legend(base_map)
```

```
Warning: Removed 454346 row(s) containing missing values (geom_path).
```

```
cowplot::plot_grid(legend, scale = .5)
```



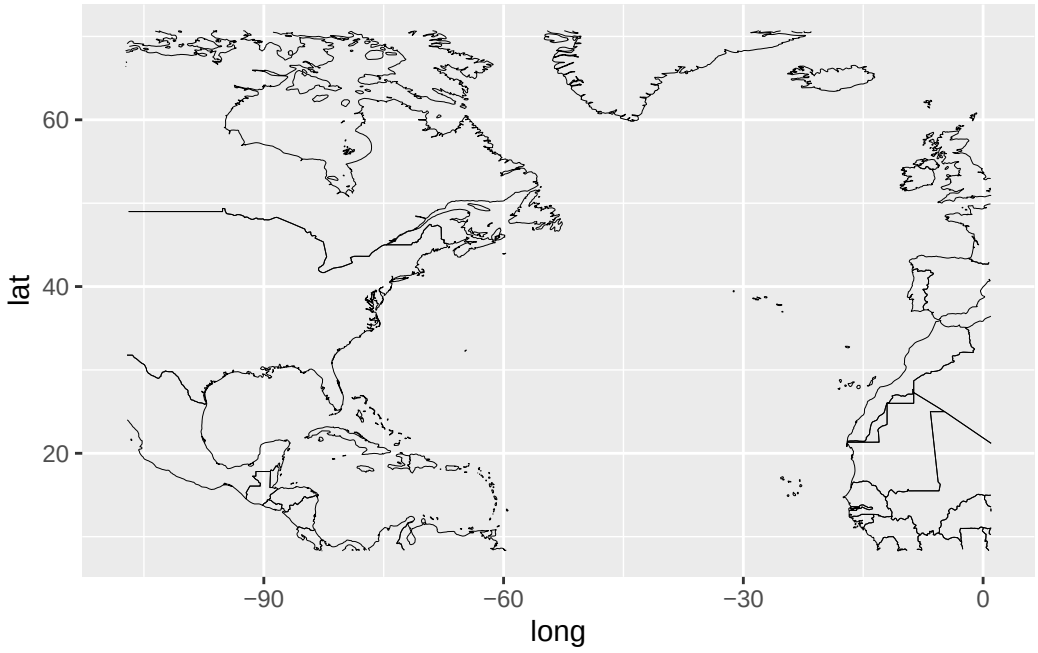
Only Tropical Storms

```
library(purrr)

bbox <- storms %>%
  select(lat, long) %>%
  map_df(range)
bbox
```

```
# A tibble: 2 x 2
    lat  long
  <dbl> <dbl>
1   8.3 -107.
2  70.7    1
```

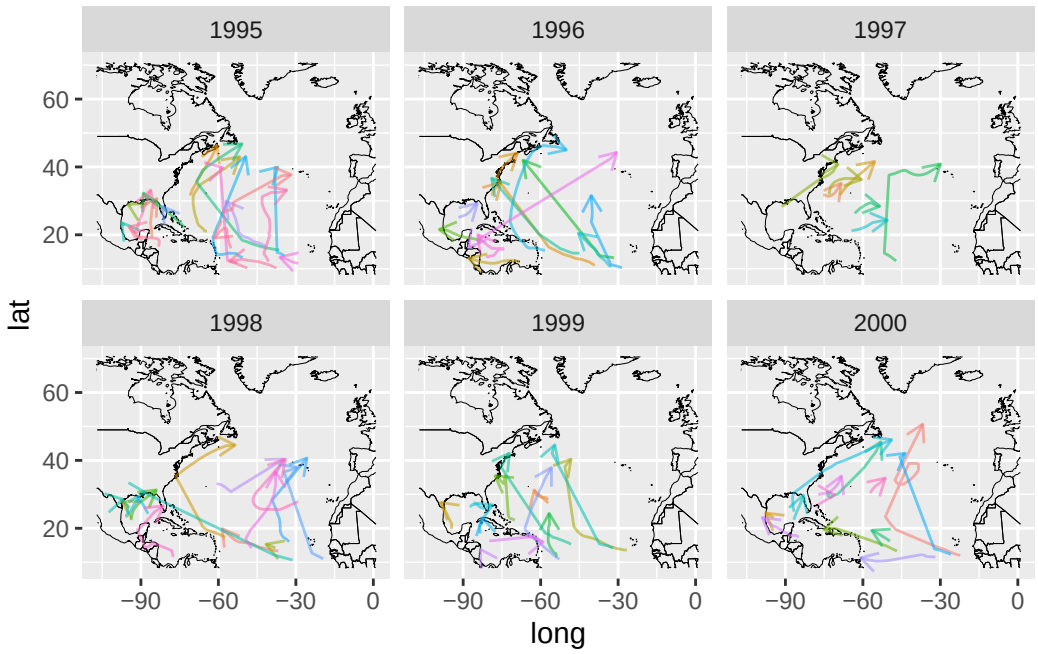
```
base_map <- map_data("world") %>% ggplot( aes(x = long, y = lat)) +
  geom_path(aes(group = group), color = "black", size = 0.1) +
  lims(x = bbox$long, y = bbox$lat)

base_map
```

```
Warning: Removed 78391 row(s) containing missing values (geom_path).
```



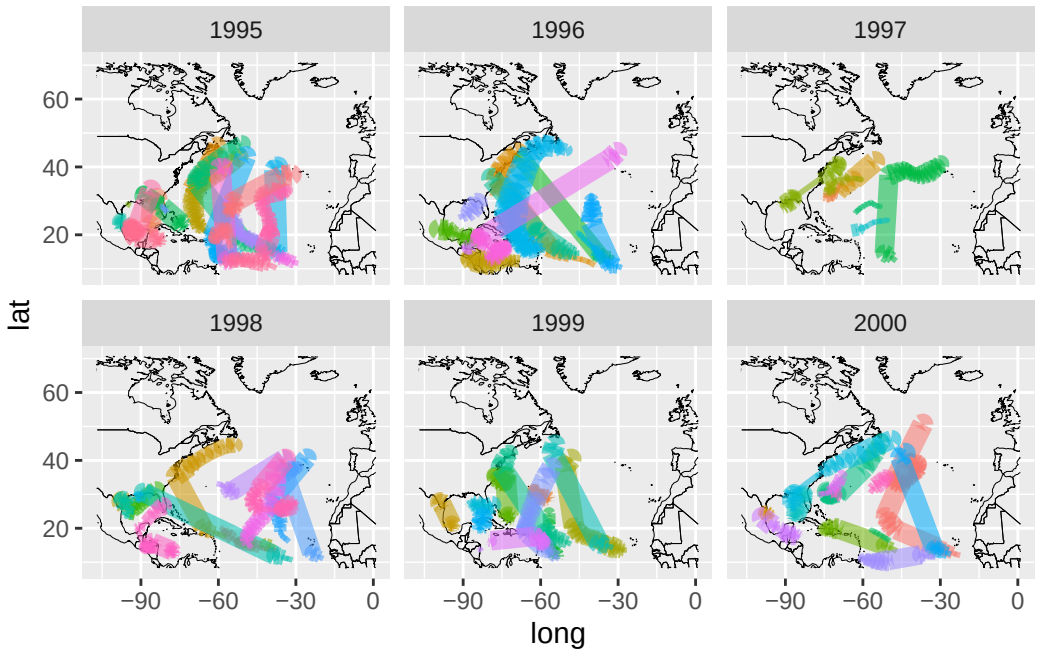

```
storms3 <- storms2 %>% filter(type == "Tropical Storm")

base_map + geom_path(data = storms3, aes(color = name, alpha = 0.01), arrow = arrow(length
  facet_wrap(~year) +
  theme(legend.position = "none")
```

```
Warning: Removed 454346 row(s) containing missing values (geom_path).
```




```
base_map + geom_path(data = storms3, aes(color = name, alpha = 0.01, size = wind), arrow =
  facet_wrap(~year) +
  theme(legend.position = "none")
```

```
Warning: Removed 454346 row(s) containing missing values (geom_path).
```



If you have ggmap working, you can try the following code to plot the storms on a Google map.

```
library(ggmap)
```

```
Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(grid)

area <- c(left = -117.3, bottom = -5.7, right = 11, top = 72.7)

storms_plot <- get_stamenmap(area, zoom = 4) %>%
  ggmap() +
  geom_path(aes(x = long, y = lat, alpha = 0.01, col = name, size = wind), data = storms)
  facet_wrap(~year)
```

```
42 tiles needed, this may take a while (try a smaller zoom).
```

```
Source : http://tile.stamen.com/terrain/4/2/3.png
```

Source : http://tile.stamen.com/terrain/4/3/3.png

Source : http://tile.stamen.com/terrain/4/4/3.png

Source : http://tile.stamen.com/terrain/4/5/3.png

Source : http://tile.stamen.com/terrain/4/6/3.png

Source : http://tile.stamen.com/terrain/4/7/3.png

Source : http://tile.stamen.com/terrain/4/8/3.png

Source : http://tile.stamen.com/terrain/4/2/4.png

Source : http://tile.stamen.com/terrain/4/3/4.png

Source : http://tile.stamen.com/terrain/4/4/4.png

Source : http://tile.stamen.com/terrain/4/5/4.png

Source : http://tile.stamen.com/terrain/4/6/4.png

Source : http://tile.stamen.com/terrain/4/7/4.png

Source : http://tile.stamen.com/terrain/4/8/4.png

Source : http://tile.stamen.com/terrain/4/2/5.png

Source : http://tile.stamen.com/terrain/4/3/5.png

Source : http://tile.stamen.com/terrain/4/4/5.png

Source : http://tile.stamen.com/terrain/4/5/5.png

Source : http://tile.stamen.com/terrain/4/6/5.png

Source : http://tile.stamen.com/terrain/4/7/5.png

Source : http://tile.stamen.com/terrain/4/8/5.png

Source : http://tile.stamen.com/terrain/4/2/6.png

Source : http://tile.stamen.com/terrain/4/3/6.png

Source : http://tile.stamen.com/terrain/4/4/6.png

Source : http://tile.stamen.com/terrain/4/5/6.png

Source : http://tile.stamen.com/terrain/4/6/6.png

Source : http://tile.stamen.com/terrain/4/7/6.png

Source : http://tile.stamen.com/terrain/4/8/6.png

Source : http://tile.stamen.com/terrain/4/2/7.png

Source : http://tile.stamen.com/terrain/4/3/7.png

Source : http://tile.stamen.com/terrain/4/4/7.png

Source : http://tile.stamen.com/terrain/4/5/7.png

Source : http://tile.stamen.com/terrain/4/6/7.png

Source : http://tile.stamen.com/terrain/4/7/7.png

Source : http://tile.stamen.com/terrain/4/8/7.png

Source : http://tile.stamen.com/terrain/4/2/8.png

Source : http://tile.stamen.com/terrain/4/3/8.png

Source : http://tile.stamen.com/terrain/4/4/8.png

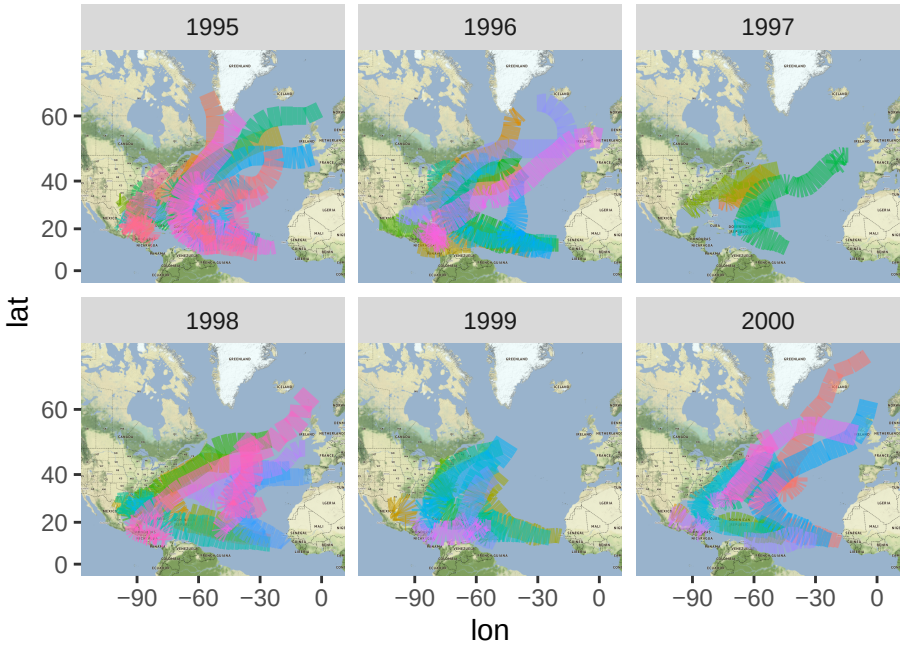
Source : http://tile.stamen.com/terrain/4/5/8.png

Source : http://tile.stamen.com/terrain/4/6/8.png

Source : http://tile.stamen.com/terrain/4/7/8.png

Source : http://tile.stamen.com/terrain/4/8/8.png

```
storms_plot + theme(legend.position = 'None')
```



Problem 9

a) The association between *bill depth* and *bill length* is **positive**.

```
library(palmerpenguins)

head(penguins, 2)
```

```
# A tibble: 2 x 8
  species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g sex
  <fct>   <fct>           <dbl>         <dbl>            <int>       <int> <fct>
1 Adelie  Torge~           39.1          18.7              181        3750 male
2 Adelie  Torge~           39.5          17.4              186        3800 fema~
# ... with 1 more variable: year <int>
```

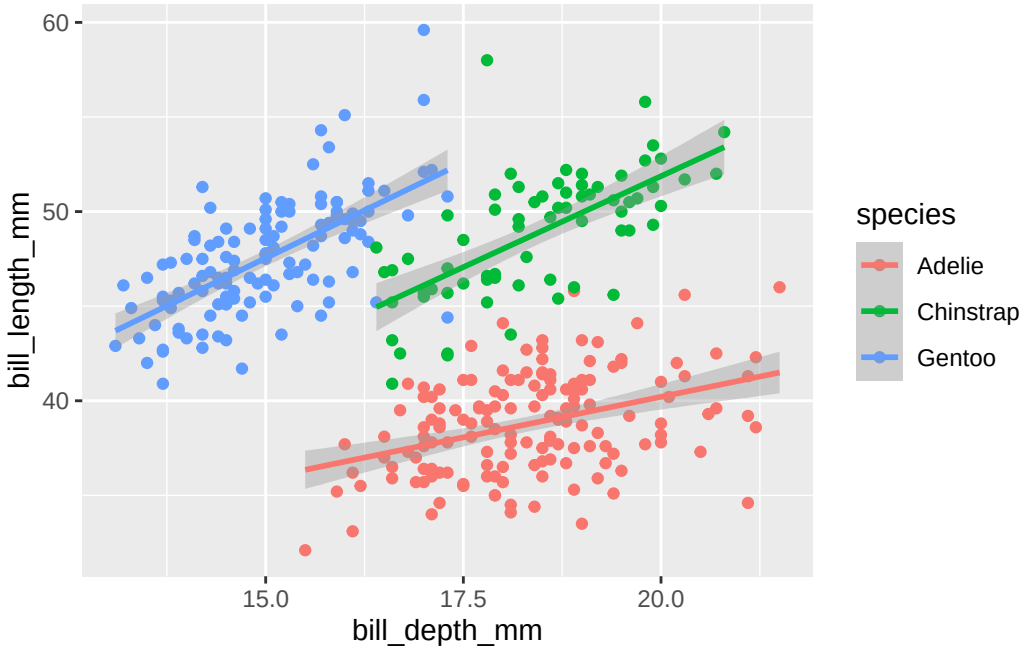
```
penguins %>% select(bill_length_mm, bill_depth_mm, species) %>%
  ggplot(aes(y = bill_length_mm, x = bill_depth_mm, color = species)) +
  geom_point() +
```

```
    geom_smooth(method = "lm")
```

```
`geom_smooth()` using formula 'y ~ x'
```

```
Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

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



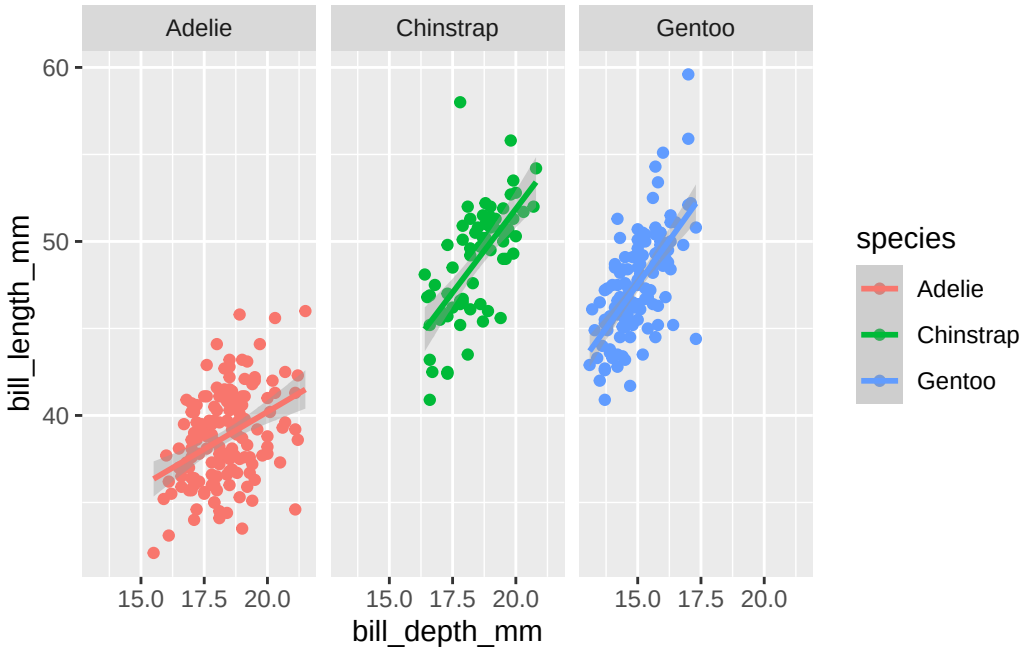
b) The association between *bill depth* and *bill length* is **positive** for each species.

```
penguins %>% select(bill_length_mm, bill_depth_mm, species) %>%
  ggplot(aes(y = bill_length_mm, x = bill_depth_mm, color = species)) +
  geom_point() +
  geom_smooth(method = "lm") +
  facet_wrap(~ species)
```

```
`geom_smooth()` using formula 'y ~ x'
```

```
Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

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



Problem 10

Deadest Names

```
library(mdsr)
baby_names <- make_babynames_dist()
```

Note that the dataset ranges from 1900 to 2010, while the visualization was for names up to 2014. So we will not be able to match the plot.

```
baby_names %>% summarize(range = range(year))
```

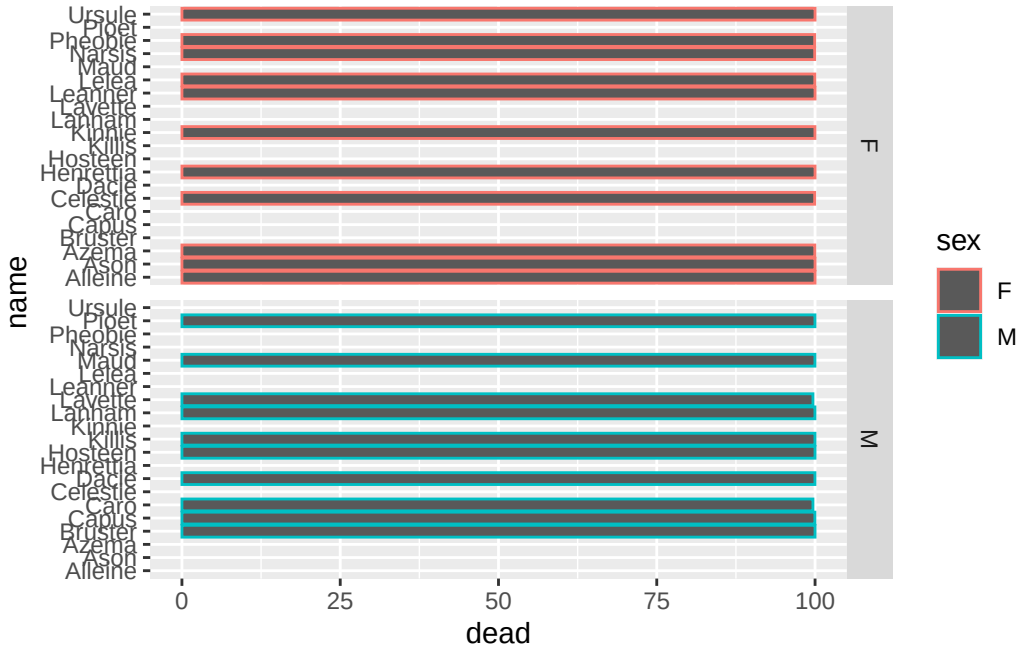
```
# A tibble: 2 x 1
  range
  <dbl>
1  1900
2  2010
```

```
tail(baby_names)
```

```
# A tibble: 6 x 9
   year sex   name           n       prop alive_prob count_thousands age_today
  <dbl> <chr> <chr>      <int>      <dbl>      <dbl>           <dbl>     <dbl>
1  2010 M     Zylin          5 0.00000244      0.993           0.005         4
2  2010 M     Zymaire        5 0.00000244      0.993           0.005         4
3  2010 M     Zyonne         5 0.00000244      0.993           0.005         4
4  2010 M     Zyquarius      5 0.00000244      0.993           0.005         4
5  2010 M     Zyran          5 0.00000244      0.993           0.005         4
6  2010 M     Zzyzx          5 0.00000244      0.993           0.005         4
# ... with 1 more variable: est_alive_today <dbl>
```

```
baby_names %>% group_by(sex, name) %>%
  summarise( dead = (sum(n - est_alive_today)/sum(n))*100 ) %>%
  arrange(desc(dead), .by_group=TRUE) %>%
  top_n(10) %>%
  ggplot(aes(y = name, x = dead, color = sex)) +
  geom_bar(stat = "identity") +
  facet_grid(sex ~ .)
```

```
`summarise()` has grouped output by 'sex'. You can override using the `.groups`
argument.
Selecting by dead
```


name
Ursule
Pioet
Pheobie
Narsis
Maud
Lelea
Leannel
Laverle
Lannam
Kinnie
Killis
Hosteen
Henrettia
Dacie
Celestie
Caro
Capus
Bruster
Azema
Ason
Alleine
F
M
0
25
50
75
100
dead
sex
F
M