Amphibians Final

Hannah Hagearty

Frogs, Salamanders and Toads YAY!!!

#load in the packages needed

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(ggplot2)
library(knitr)
library(dplyr)
```

#loading in data

```
amph_data <- read.csv("Amphibians_on_NPS_Lands_20231201.csv")
view(amph_data)
```

#clean up data

```r
str(amph_data)
```

```
'data.frame':   3472 obs. of  22 variables:
 $ Park_Code             : chr  "ABLI" "ABLI" "ABLI" "ABLI" ...
 $ Park_Name             : chr  "Abraham Lincoln Birthplace National Historical Park" "Abraham
 $ IM_Network            : chr  "CUPN" "CUPN" "CUPN" "CUPN" ...
 $ CASC_Region           : chr  "NE" "NE" "NE" "NE" ...
 $ State                 : chr  "KY" "KY" "KY" "KY" ...
 $ TSN                   : int  173520 208204 173598 173590 173591 173592 773511 773520 17358
 $ Order                 : chr  "Anura" "Caudata" "Caudata" "Caudata" ...
 $ Family                : chr  "Hylidae" "Ambystomatidae" "Ambystomatidae" "Ambystomatidae"
 $ Scientific_Name       : chr  "Acris crepitans" "Ambystoma barbouri" "Ambystoma jeffersonian
 $ Common_Names          : chr  "Northern Cricket Frog, Eastern Cricket Frog" "Streamside Sala
 $ Verified_Date         : chr  "9/8/2020" "9/8/2020" "9/8/2020" "9/8/2020" ...
 $ Park_OccurrenceStatus: chr  "Present" "Adjacent" "Adjacent" "Adjacent" ...
 $ Notes                 : chr  "" "" "Expected but not observed" "Expected but not observed"
 $ Verified_Source       : chr  "https://irma.nps.gov/DataStore/DownloadFile/420948" "https:/
 $ Park_Synonyms         : chr  "Acris crepitans crepitans" "" "" "" ...
 $ NPSpecies_Occurrence : chr  "Present" "Not In Park" "Unconfirmed" "Unconfirmed" ...
 $ NPSpecies_OccTag      : chr  "" "Adjacent" "" "" ...
 $ Abundance             : chr  "Common" "" "" "" ...
 $ Nativeness            : chr  "Native" "Native" "Native" "Native" ...
 $ GRank                 : chr  "G5" "G4" "G4" "G5" ...
 $ SRank                 : chr  "KY: S5" "KY: S4" "KY: S4" "KY: S5" ...
 $ IUCNRank              : chr  "LC" "NT" "LC" "LC" ...
```

#Make a map with individuals by points - Common Name (number of ones that occur in a state), graphed on the states the parks occur in, showing how many species of amphibian were recorded per park in the present day. Will switch up by by Park occurance status.

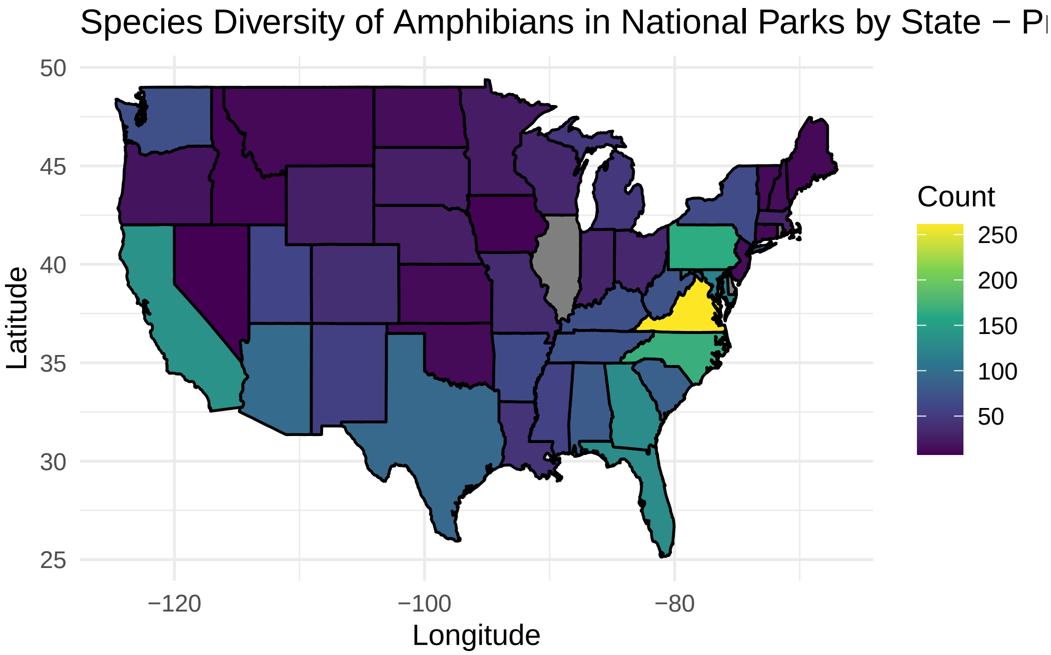
```r
# I got help from Chat GTP on this question, mostly i would get an error and ask it how to fi

library(ggplot2)
library(maps)
```

```
Attaching package: 'maps'
```

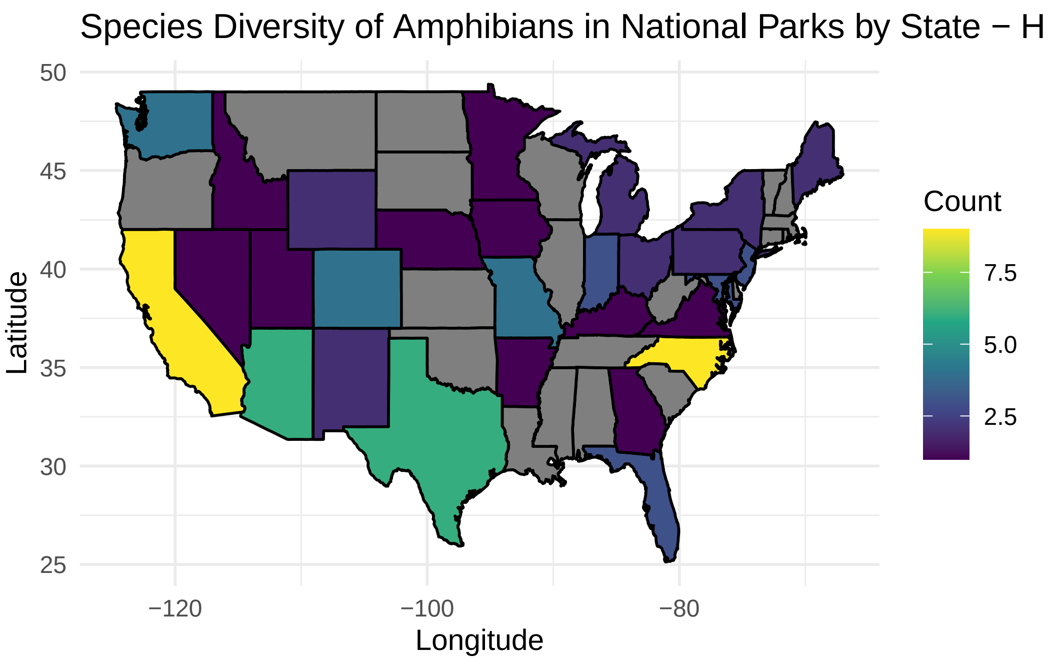
```
The following object is masked from 'package:purrr':

    map
```

```r
library(dplyr)

# Get the map of the USA - with help from chat GTP
usa_map <- map_data("state")

View(usa_map)

map_data <- amph_data %>%
  select(State, Common_Names, Park_OccurrenceStatus) %>%
  filter(Park_OccurrenceStatus == 'Present') %>% #filtering for present information
  group_by(State) %>%
  summarise(count = n())
# ok so this is taking the number of common names found in each state and it is going to hope

state_abbreviation_to_name <- data.frame(
  abbreviation = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA",
                   "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
                   "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
                   "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
                   "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY"),
  name = c("alabama", "alaska", "arizona", "arkansas", "california", "colorado",
           "connecticut", "delaware", "florida", "georgia", "hawaii", "idaho",
           "illinois", "indiana", "iowa", "kansas", "kentucky", "louisiana",
           "maine", "maryland", "massachusetts", "michigan", "minnesota",
           "mississippi", "missouri", "montana", "nebraska", "nevada", "new hampshire",
           "new jersey", "new mexico", "new york", "north carolina", "north dakota",
           "ohio", "oklahoma", "oregon", "pennsylvania", "rhode island", "south carolina",
           "south dakota", "tennessee", "texas", "utah", "vermont", "virginia",
           "washington", "west virginia", "wisconsin", "wyoming")
) # change the labels in the maps_data, to lowercase state names because i need that to merge

map_data <- map_data %>%
  left_join(state_abbreviation_to_name, by = c("State" = "abbreviation"))

View(map_data)

map_data <- map_data %>%
  drop_na() #get rid of stuff that is not labeled as a state (like territories and islands)


#couldn't get this to work, so asked chat GTP and it said that my merge was not working becau
```

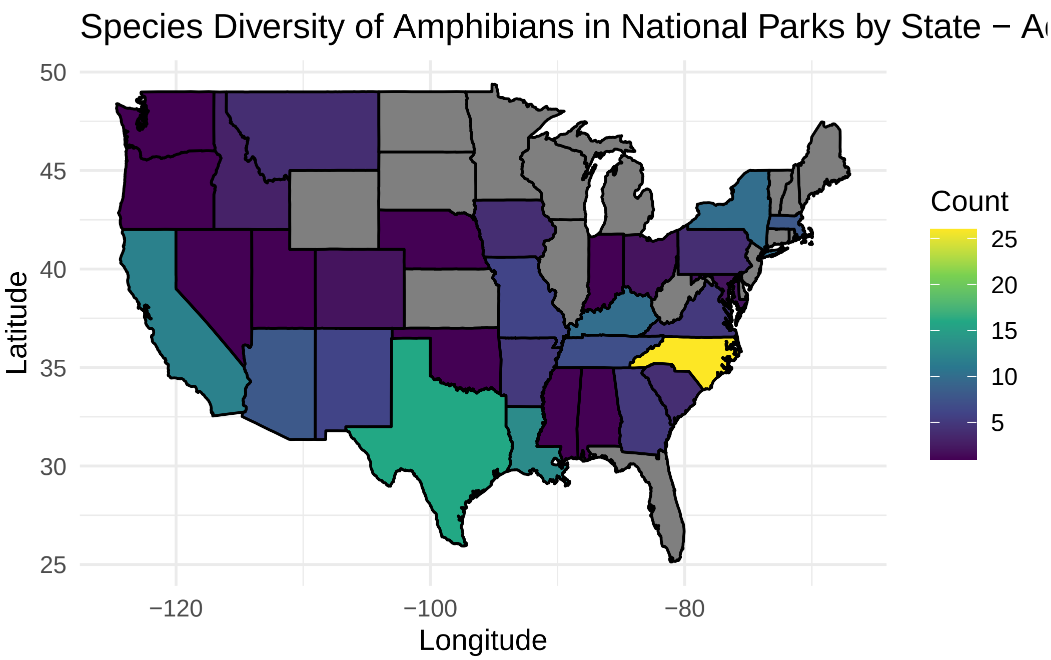
```
View(map_data)

usa_map_data <- usa_map %>%
  left_join(map_data, by = c("region" = "name")) #join them together

View(usa_map_data)


#get the map into R.
#got help from chat gtp on this
#ggplot(data = usa_map, aes(x = long, y = lat, group = group)) +
#  geom_polygon(fill = "lightgray", color = "black") +

ggplot(data = usa_map_data, aes(x = long, y = lat, group = group, fill = count)) +
  geom_polygon(color = "black") +  # Borders of the states
  scale_fill_viridis_c(option = "D") +  # said this one was color blind friendly

  #scale_fill_continuous() +
  labs(title = "Species Diversity of Amphibians in National Parks by State - Present day", fi
  theme_minimal() +
  theme(legend.position = "right")+
  ylab("Latitude") +
  xlab("Longitude")
```


Species Diversity of Amphibians in National Parks by State − Pr
50
45
40
35
30
25
Latitude
−120
−100
−80
Longitude
Count
250
200
150
100
50

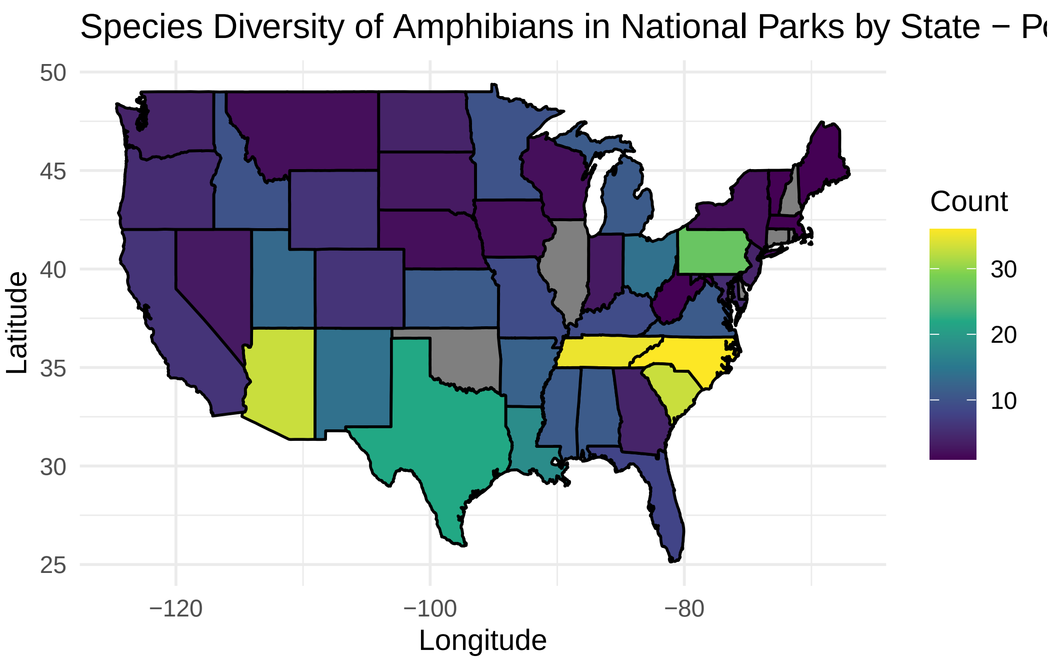
```
usa_map_h <- map_data("state")

View(usa_map_h)

map_data_h <- amph_data %>%
  select(State, Common_Names, Park_OccurrenceStatus) %>%
  filter(Park_OccurrenceStatus == 'Historical') %>% #filtering for present day information
  group_by(State) %>%
  summarise(count = n())
# ok so this is taking the number of common names found in each state and it is going to hope

state_abbreviation_to_name_h <- data.frame(
  abbreviation = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA",
                   "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
                   "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
                   "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
                   "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY"),
  name = c("alabama", "alaska", "arizona", "arkansas", "california", "colorado",
           "connecticut", "delaware", "florida", "georgia", "hawaii", "idaho",
           "illinois", "indiana", "iowa", "kansas", "kentucky", "louisiana",
           "maine", "maryland", "massachusetts", "michigan", "minnesota",
           "mississippi", "missouri", "montana", "nebraska", "nevada", "new hampshire",
           "new jersey", "new mexico", "new york", "north carolina", "north dakota",
           "ohio", "oklahoma", "oregon", "pennsylvania", "rhode island", "south carolina",
           "south dakota", "tennessee", "texas", "utah", "vermont", "virginia",
           "washington", "west virginia", "wisconsin", "wyoming")
) # change the labels in the maps_data, to lowercase state names because i need that to merg

map_data_h <- map_data_h %>%
  left_join(state_abbreviation_to_name_h, by = c("State" = "abbreviation"))

View(map_data_h)

map_data_h <- map_data_h %>%
  drop_na() #get rid of stuff that is not labeled as a state (like territories and islands)


#couldn't get this to work, so asked chat GTP and it said that my merge was not working becau

View(map_data_h)

usa_map_data_h <- usa_map_h %>%
  left_join(map_data_h, by = c("region" = "name")) #join them together

View(usa_map_data_h)


#get the map into R.
#got help from chat gtp on this
#ggplot(data = usa_map, aes(x = long, y = lat, group = group)) +
#  geom_polygon(fill = "lightgray", color = "black") +

ggplot(data = usa_map_data_h, aes(x = long, y = lat, group = group, fill = count)) +
  geom_polygon(color = "black") +  # Borders of the states
  scale_fill_viridis_c(option = "D") +  # said this one was color blind friendly

  #scale_fill_continuous() +
  labs(title = "Species Diversity of Amphibians in National Parks by State - Historical", fil
  theme_minimal() +
  theme(legend.position = "right")+
  ylab("Latitude") +
  xlab("Longitude")
```


Species Diversity of Amphibians in National Parks by State − H
50
45
40
35
30
25
Latitude
−120
−100
−80
Longitude
Count
7.5
5.0
2.5

```
usa_map_Adj <- map_data("state")

View(usa_map_Adj)

map_data_Adj <- amph_data %>%
  select(State, Common_Names, Park_OccurrenceStatus) %>%
  filter(Park_OccurrenceStatus == 'Adjacent') %>% #filtering for present day information
  group_by(State) %>%
  summarise(count = n())
# ok so this is taking the number of common names found in each state and it is going to hop

state_abbreviation_to_name_Adj <- data.frame(
  abbreviation = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA",
                   "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
                   "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
                   "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
                   "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY"),
  name = c("alabama", "alaska", "arizona", "arkansas", "california", "colorado",
           "connecticut", "delaware", "florida", "georgia", "hawaii", "idaho",
           "illinois", "indiana", "iowa", "kansas", "kentucky", "louisiana",
           "maine", "maryland", "massachusetts", "michigan", "minnesota",
           "mississippi", "missouri", "montana", "nebraska", "nevada", "new hampshire",
           "new jersey", "new mexico", "new york", "north carolina", "north dakota",
           "ohio", "oklahoma", "oregon", "pennsylvania", "rhode island", "south carolina",
           "south dakota", "tennessee", "texas", "utah", "vermont", "virginia",
           "washington", "west virginia", "wisconsin", "wyoming")
) # change the labels in the maps_data, to lowercase state names because i need that to merge

map_data_Adj <- map_data_Adj %>%
  left_join(state_abbreviation_to_name_Adj, by = c("State" = "abbreviation"))

View(map_data_Adj)

map_data_Adj <- map_data_Adj %>%
  drop_na() #get rid of stuff that is not labeled as a state (like territories and islands)


#couldn't get this to work, so asked chat GTP and it said that my merge was not working becau

View(map_data_Adj)

usa_map_data_Adj <- usa_map_Adj %>%
  left_join(map_data_Adj, by = c("region" = "name")) #join them together

View(usa_map_data_Adj)


#get the map into R.
#got help from chat gtp on this
#ggplot(data = usa_map, aes(x = long, y = lat, group = group)) +
#  geom_polygon(fill = "lightgray", color = "black") +

ggplot(data = usa_map_data_Adj, aes(x = long, y = lat, group = group, fill = count)) +
  geom_polygon(color = "black") +  # Borders of the states
  scale_fill_viridis_c(option = "D") +  # said this one was color blind friendly

  #scale_fill_continuous() +
  labs(title = "Species Diversity of Amphibians in National Parks by State - Adjacent", fill
  theme_minimal() +
  theme(legend.position = "right")+
  ylab("Latitude") +
  xlab("Longitude")
```

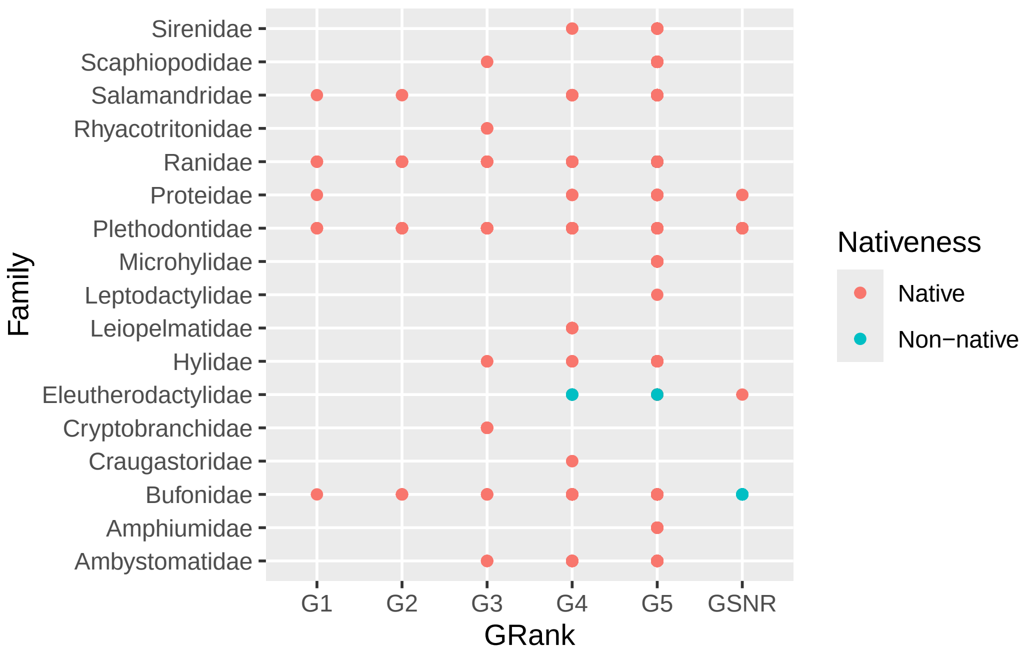

Species Diversity of Amphibians in National Parks by State − A
Latitude
50
45
40
35
30
25
−120
−100
−80
Longitude
Count
25
20
15
10
5

```
usa_map_PO <- map_data("state")

View(usa_map_PO)

map_data_PO <- amph_data %>%
  select(State, Common_Names, Park_OccurrenceStatus) %>%
  filter(Park_OccurrenceStatus == 'Possible') %>% #filtering for present day information
  group_by(State) %>%
  summarise(count = n())
# ok so this is taking the number of common names found in each state and it is going to hope

state_abbreviation_to_name_Adj <- data.frame(
  abbreviation = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA",
                   "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
                   "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
                   "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
                   "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY"),
  name = c("alabama", "alaska", "arizona", "arkansas", "california", "colorado",
           "connecticut", "delaware", "florida", "georgia", "hawaii", "idaho",
           "illinois", "indiana", "iowa", "kansas", "kentucky", "louisiana",
           "maine", "maryland", "massachusetts", "michigan", "minnesota",
           "mississippi", "missouri", "montana", "nebraska", "nevada", "new hampshire",
           "new jersey", "new mexico", "new york", "north carolina", "north dakota",
           "ohio", "oklahoma", "oregon", "pennsylvania", "rhode island", "south carolina",
           "south dakota", "tennessee", "texas", "utah", "vermont", "virginia",
           "washington", "west virginia", "wisconsin", "wyoming")
) # change the labels in the maps_data, to lowercase state names because i need that to merg

map_data_PO <- map_data_PO %>%
  left_join(state_abbreviation_to_name_Adj, by = c("State" = "abbreviation"))

View(map_data_PO)

map_data_PO <- map_data_PO %>%
  drop_na() #get rid of stuff that is not labeled as a state (like territories and islands)


#couldn't get this to work, so asked chat GTP and it said that my merge was not working becau

View(map_data_PO)

usa_map_data_PO <- usa_map_PO %>%
  left_join(map_data_PO, by = c("region" = "name")) #join them together

View(usa_map_data_PO)


#get the map into R.
#got help from chat gtp on this
#ggplot(data = usa_map, aes(x = long, y = lat, group = group)) +
#  geom_polygon(fill = "lightgray", color = "black") +

ggplot(data = usa_map_data_PO, aes(x = long, y = lat, group = group, fill = count)) +
  geom_polygon(color = "black") +  # Borders of the states
  scale_fill_viridis_c(option = "D") +  # said this one was color blind friendly

  #scale_fill_continuous() +
  labs(title = "Species Diversity of Amphibians in National Parks by State - Possible", fill
  theme_minimal() +
  theme(legend.position = "right")+
  ylab("Latitude") +
  xlab("Longitude")
```


Species Diversity of Amphibians in National Parks by State − P
Latitude
50
45
40
35
30
25
−120
−100
−80
Longitude
Count
30
20
10

Globabl Conservation Status By Family

```
#Trying to graph Global Conservation Status by Family of amphibian, color indicates native o

Global_conservation_status_rank_by_family <- amph_data %>%
  select(GRank, Family, Nativeness)


view(Global_conservation_status_rank_by_family)

ggplot(data = Global_conservation_status_rank_by_family, aes(x = GRank , y = Family, color =
  geom_point()
```

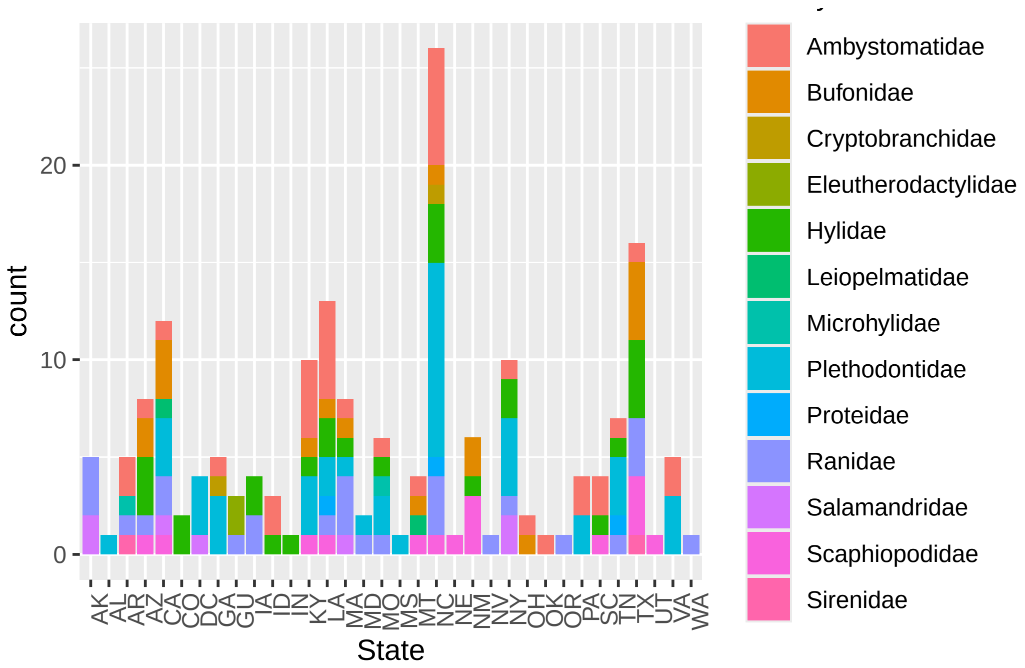


#These are the Stacked Bar Plots that each go with one of the Maps.

```
#Family diversity by State, this should go with the Map that is Adjacent

map_data_Adj_bar <- amph_data %>%
  select(State, Common_Names, Park_OccurrenceStatus, Family) %>%
  filter(Park_OccurrenceStatus == 'Adjacent') %>% #filtering for Adjacent information
  filter(State != "Mariana Islands") %>%
  filter( State != "Trail (ME to GA)") %>% # got rid of stuff on a trail or stuff on a randon
  group_by(State, Family) %>%
  summarise(count = n() )
```

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

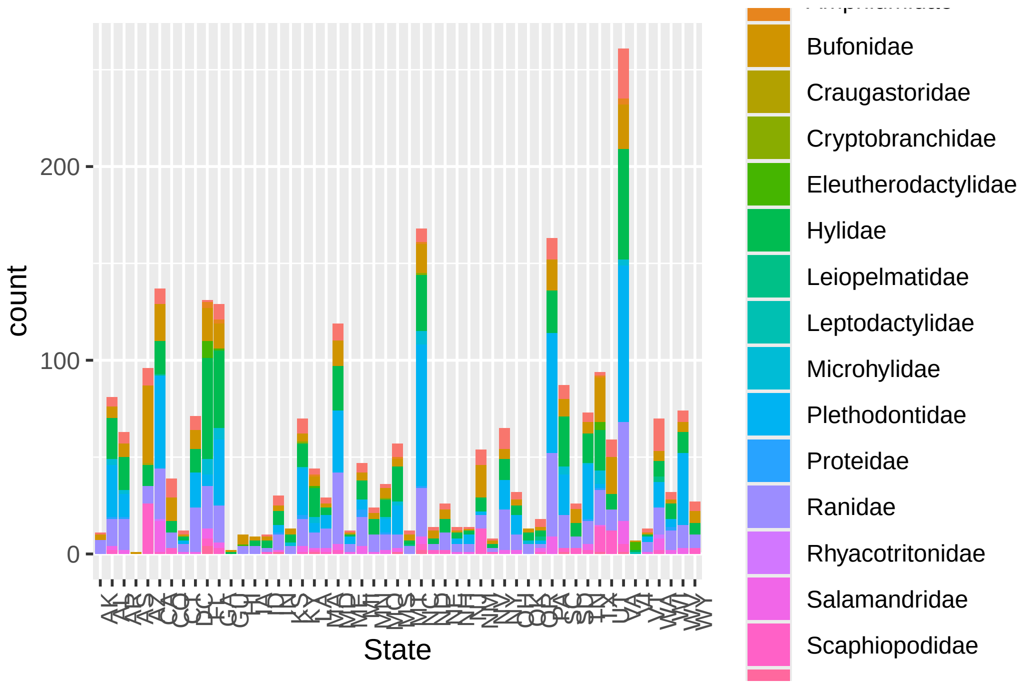
```
view(map_data_Adj_bar)


ggplot(data = map_data_Adj_bar, aes(fill= Family , y=count, x=State)) +
    geom_bar(position="stack", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
#Family diversity by State, this should go with the Map that is Present

map_data_bar <- amph_data %>%
  select(State, Common_Names, Park_OccurrenceStatus, Family) %>%
  filter(Park_OccurrenceStatus == 'Present') %>%#filtering for present information
  filter(State != "Mariana Islands") %>%
  filter( State != "Trail (ME to GA)") %>% # got rid of stuff on a trail or stuff on a rando
  group_by(State, Family) %>%
  summarise(count = n() )
```

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

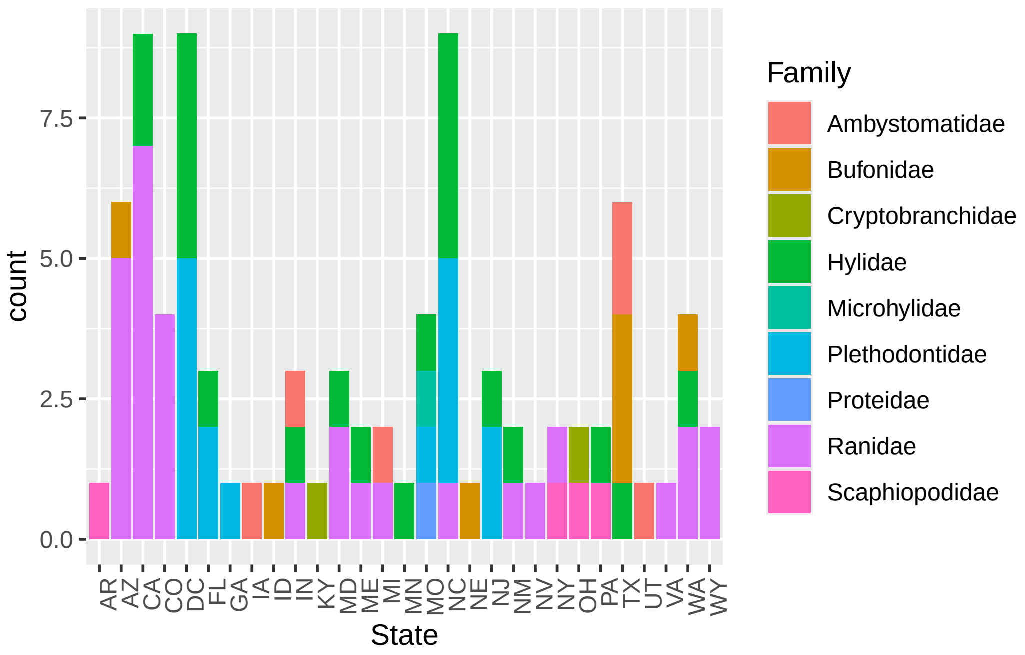
```
view(map_data_bar)


ggplot(data = map_data_bar, aes(fill= Family , y=count, x=State)) +
    geom_bar(position="stack", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
map_data_his_bar <- amph_data %>%
  select(State, Common_Names, Park_OccurrenceStatus, Family) %>%
  filter(Park_OccurrenceStatus == 'Historical') %>%#filtering for Historical information
  filter(State != "Mariana Islands") %>%
  filter( State != "Trail (ME to GA)") %>% # got rid of stuff on a trail or stuff on a rando
  group_by(State, Family) %>%
  summarise(count = n() )
```

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

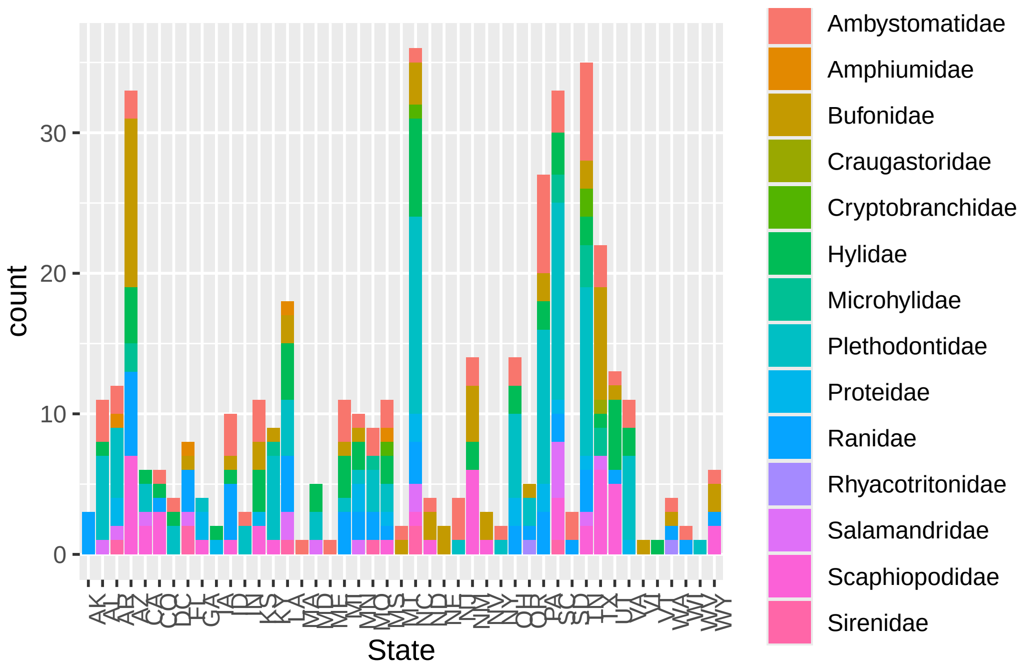
```
view(map_data_his_bar)


ggplot(data = map_data_his_bar, aes(fill= Family , y=count, x=State)) +
    geom_bar(position="stack", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
map_data_pos_bar <- amph_data %>%
  select(State, Common_Names, Park_OccurrenceStatus, Family) %>%
  filter(Park_OccurrenceStatus == 'Possible') %>% #filtering for Possible information
  filter(State != "Mariana Islands") %>%
  filter( State != "Trail (ME to GA)") %>% # got rid of stuff on a trail or stuff on a randon
  group_by(State, Family) %>%
  summarise(count = n() )
```

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

```
view(map_data_pos_bar)


ggplot(data = map_data_pos_bar, aes(fill= Family , y=count, x=State)) +
    geom_bar(position="stack", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
#working on this

#library(dplyr)
#library(lubridate)


#class(amph_data$Verified_Date)
#amph_data$Verified_Date <- as.Date(amph_data$Verified_Date) # chat gtp helped me figure out
#amph_data$Verified_Date <- mdy(amph_data$Verified_Date)
 #view(amph_data) # had to change up the order of day month and year so that everything got

#native <- amph_data %>%
#  select(State, Nativeness, Verified_Date) %>%
#  mutate(
 #    YEAR = year(Verified_Date),
 #    MONTH = month(Verified_Date),
 #    DAY = day(Verified_Date)
 # ) %>%
 # group_by(State, Nativeness) %>%
 # summarise(count = n())


#view(native)
```

```
#ggplot(data = native, aes(y=count, x=State, color = Nativeness)) +
#    geom_point() +
#  theme(axis.text.x = element_text(angle = 90, hjust = 1))


#native <- amph_data %>%
#  select(State, Nativeness, Verified_Date) %>%
#  mutate(
#    YEAR = year(Verified_Date),
#    MONTH = month(Verified_Date),
 #   DAY = day(Verified_Date)
 # ) %>%
 # group_by(State, Nativeness, Year) %>%
 # summarise(count = n())
```