Tidyverse & ggplot2 - Bar Plots

Jen Abinette

TidyVerse Assignment

Exploring dplyr package

Dataset

2022 U.S. Primary Election data from FiveThirtyEight repository - This directory contains demographic, electoral and endorsement data for Senate, House and governor candidates in the 2022 primary elections (except Louisiana's).

https://github.com/fivethirtyeight/data/tree/master/primary-project-2022

Importing Data from Github

```
df_rep <- read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/primary-project-2022/
df_dem <- read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/primary-project-2022/
head(df_dem)
```

```
##              Candidate Gender Race.1         Race.2 Race.3 Incumbent
## 1           Gavin Dass   Male  White Asian (Indian)               No
## 2       Victor D. Dunn   Male  Black                              No
## 3 Jrmar "JJ" Jefferson   Male  Black                              No
## 4        Stephen Kocen   Male  White                              No
## 5        Robin Fulford Female  White                              No
## 6           Doc Shelby   Male  Black                              No
##   Incumbent.Challenger State Primary.Date         Office District Primary.Votes
## 1                   No Texas       3/1/22 Representative        1         1,881
## 2                   No Texas       3/1/22 Representative        1         4,554
## 3                   No Texas       3/1/22 Representative        1         7,411
## 4                   No Texas       3/1/22 Representative        1         2,457
## 5                   No Texas       3/1/22 Representative        2        17,160
## 6                   No Texas       3/1/22 Representative        3         8,531
##   Primary.. Primary.Outcome Runoff.Votes Runoff.. Runoff.Outcome EMILY.s.List
## 1       12%            Lost          N/A      N/A            N/A          N/A
## 2       28%     Made runoff        1,783      24%           Lost          N/A
## 3       45%     Made runoff        5,607      76%            Won          N/A
## 4       15%            Lost          N/A      N/A            N/A          N/A
## 5      100%             Won          N/A      N/A            N/A          N/A
## 6       38%            Lost          N/A      N/A            N/A          N/A
##   Justice.Dems Indivisible PCCC Our.Revolution Sunrise Sanders AOC
## 1          N/A         N/A  N/A            N/A     N/A     N/A N/A
```

```
## 2          N/A         N/A  N/A            N/A     N/A     N/A N/A
## 3          N/A         N/A  N/A            N/A     N/A     N/A N/A
## 4          N/A         N/A  N/A            N/A     N/A     N/A N/A
## 5          N/A         N/A  N/A            N/A     N/A     N/A N/A
## 6          N/A         N/A  N/A            N/A     N/A     N/A N/A
##   Party.Committee
## 1             N/A
## 2             N/A
## 3             N/A
## 4             N/A
## 5             N/A
## 6             N/A
```

Creating new columns

```
df_dem <- df_dem %>%
  mutate(political_party = "Democrat")
df_rep <- df_rep %>%
  mutate(political_party = "Republican")
```

Counting unique variable in each column

```
df_dem %>%
  filter("Gender" != "") %>%
  count(Gender, name  = "Count")
```

```
##      Gender Count
## 1    Female   379
## 2      Male   697
## 3 Nonbinary     1
```

```
df_rep %>%
  filter("Gender" != "") %>%
  count(Gender, name  = "Count")
```

```
##   Gender Count
## 1 Female   338
## 2   Male  1261
```

Combining data frame into one

```
df_bind <- bind_rows(df_rep, df_dem)
```

Creating a subset of specific columns into a new data frame

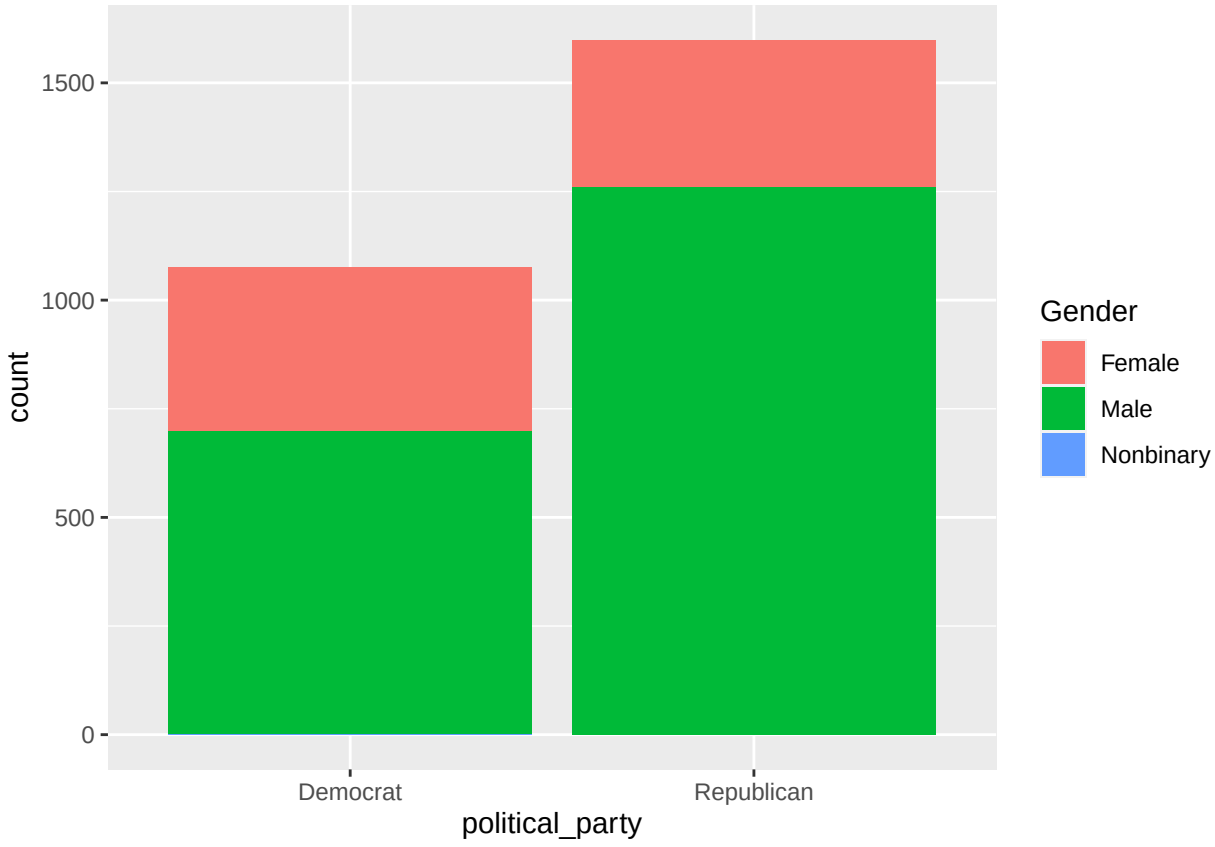
```r
df_subset_bind <- df_bind %>%
  select(Candidate,
         Gender,
         Race.1,
         Race.2,
         Race.3,
         Incumbent,
         State,
         Office,
         District,
         political_party)
```

Exploring the ggplot2 package

```r
library(ggplot2)
```

Creating a Bar Plot for gender and political party

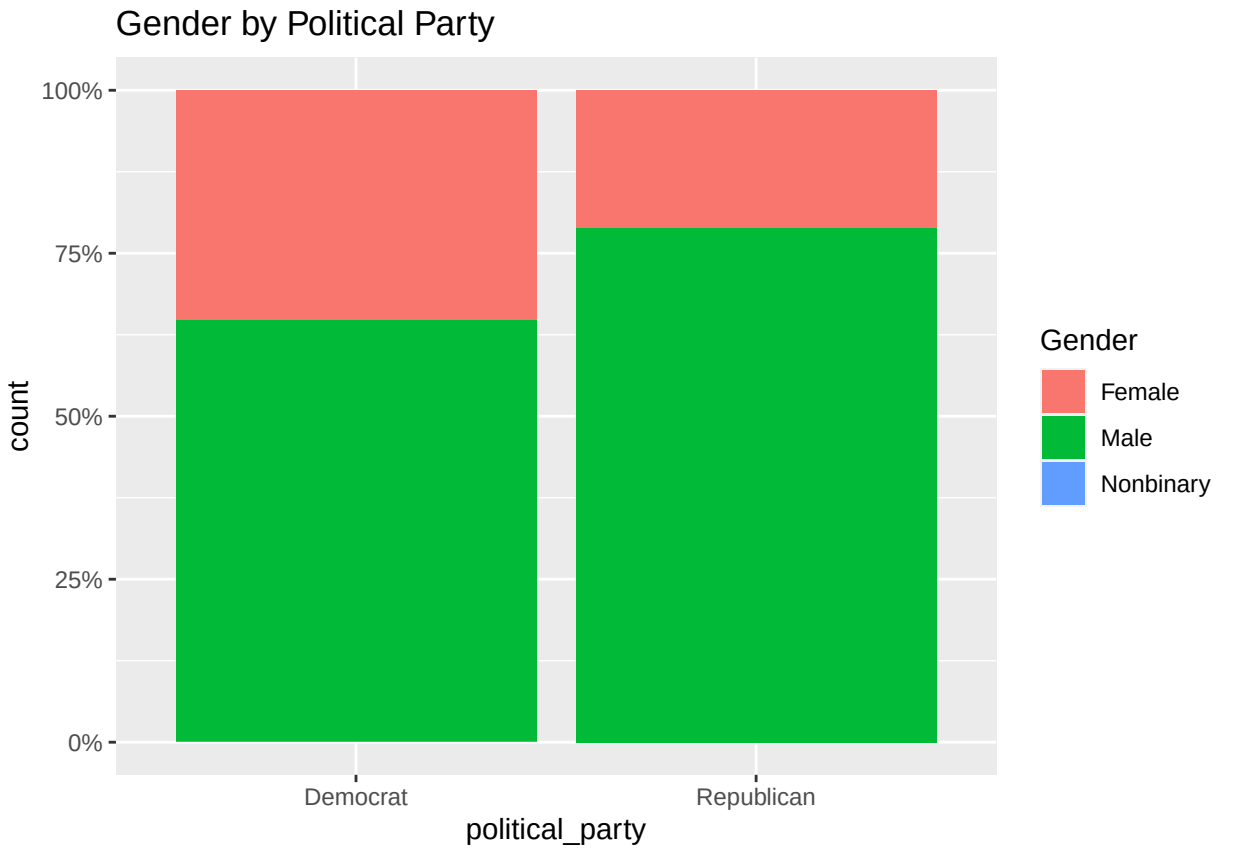
```r
ggplot(data = df_subset_bind, aes(x = political_party, fill = Gender) ) + geom_bar()
```



How can this be improved?

In this case, adjusting our scale to a percent rather than a count will better show which political party has more woman as a percent of the whole. Let's also add a title while we are at it using labs.

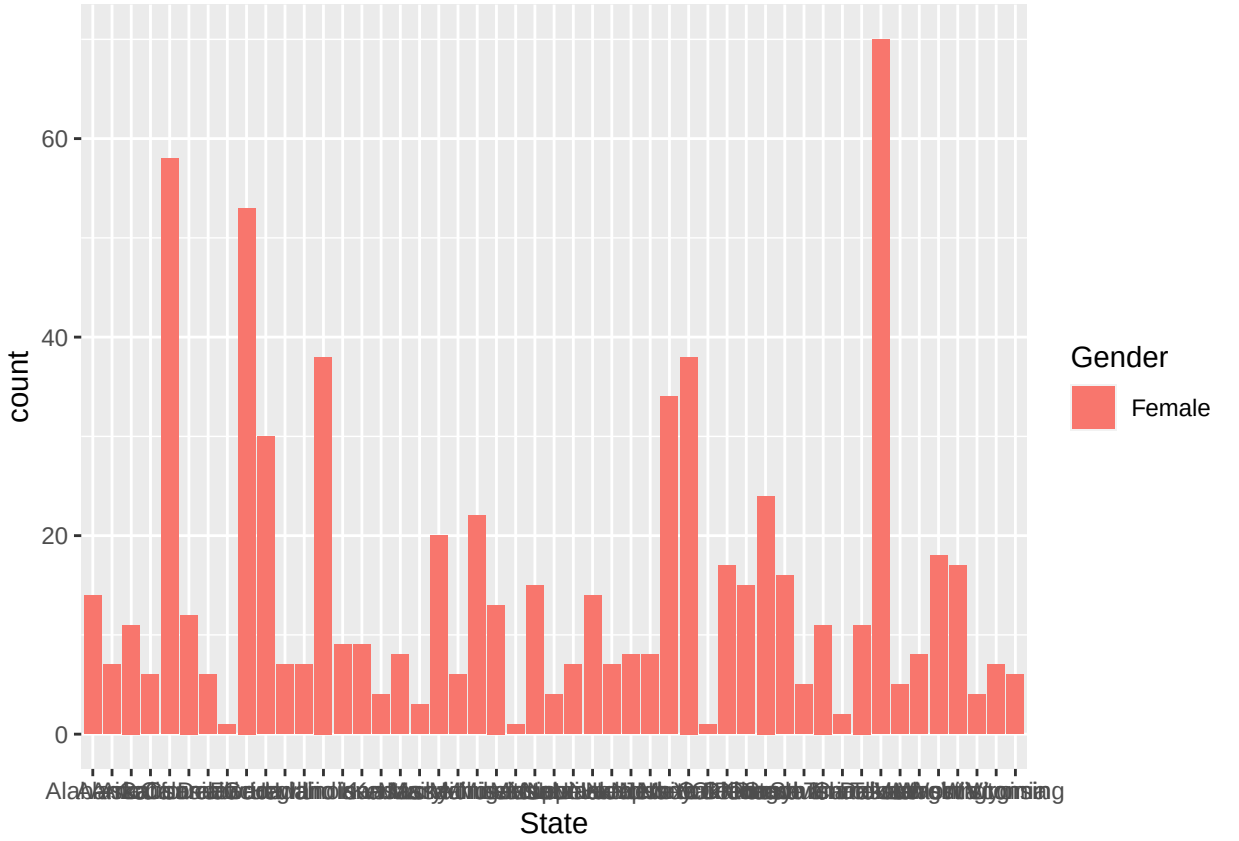
```
ggplot(data = df_subset_bind, aes(x=political_party,fill=Gender)) + geom_bar(position="fill") + scale_y
```



The above plot gives a clear indication to the audience that there is a greater percentage of female candidates in the Democratic Party than Republican.

Let's explore the count of female candidates by State

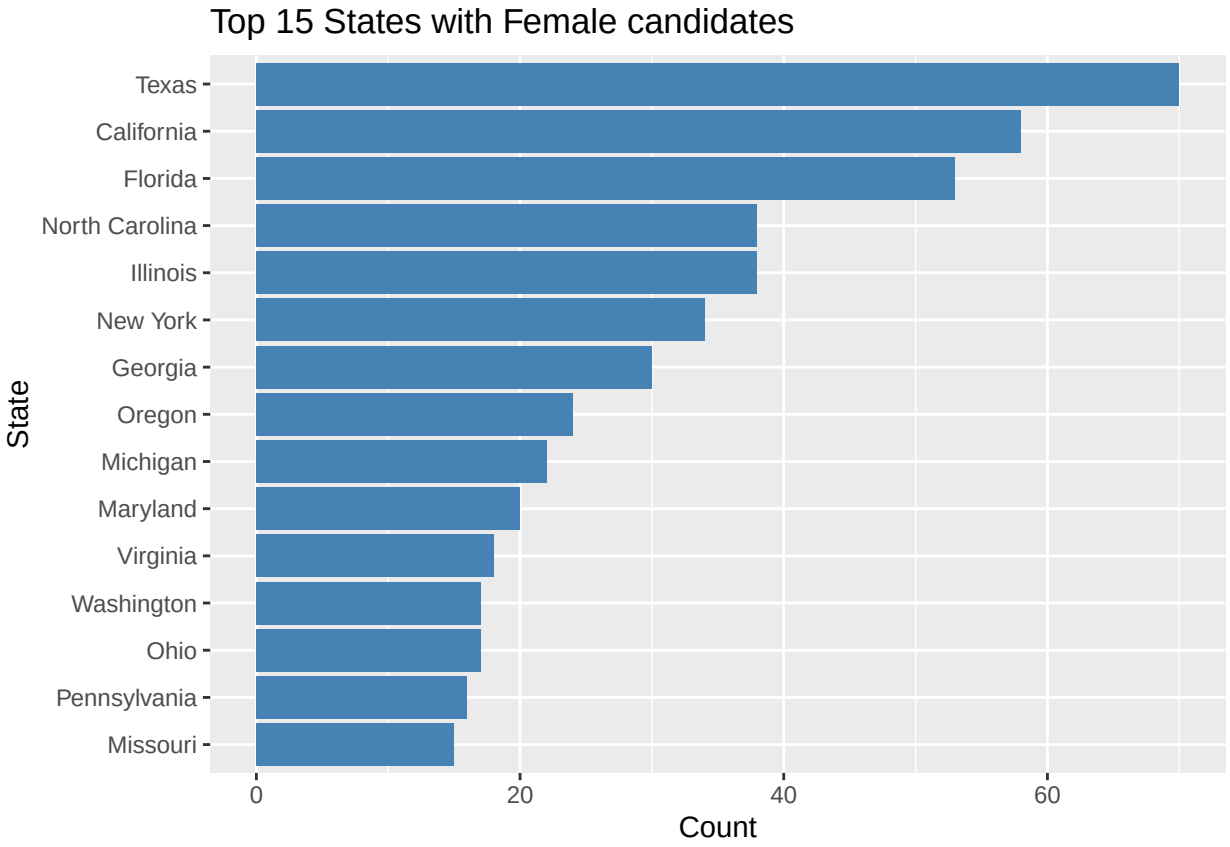
```
df_subset_bind_f <- subset(df_subset_bind, Gender == "Female")
ggplot(data = df_subset_bind_f, aes(x = State, fill=Gender)) + geom_bar()
```



How can we improve the graph above?

Currently, we are unable to read the State labels given there are so many and the x-axis offers little space. Thus we can use coord_flip to list the States on the y axis and let's limit our output to the 15 States with the highest count

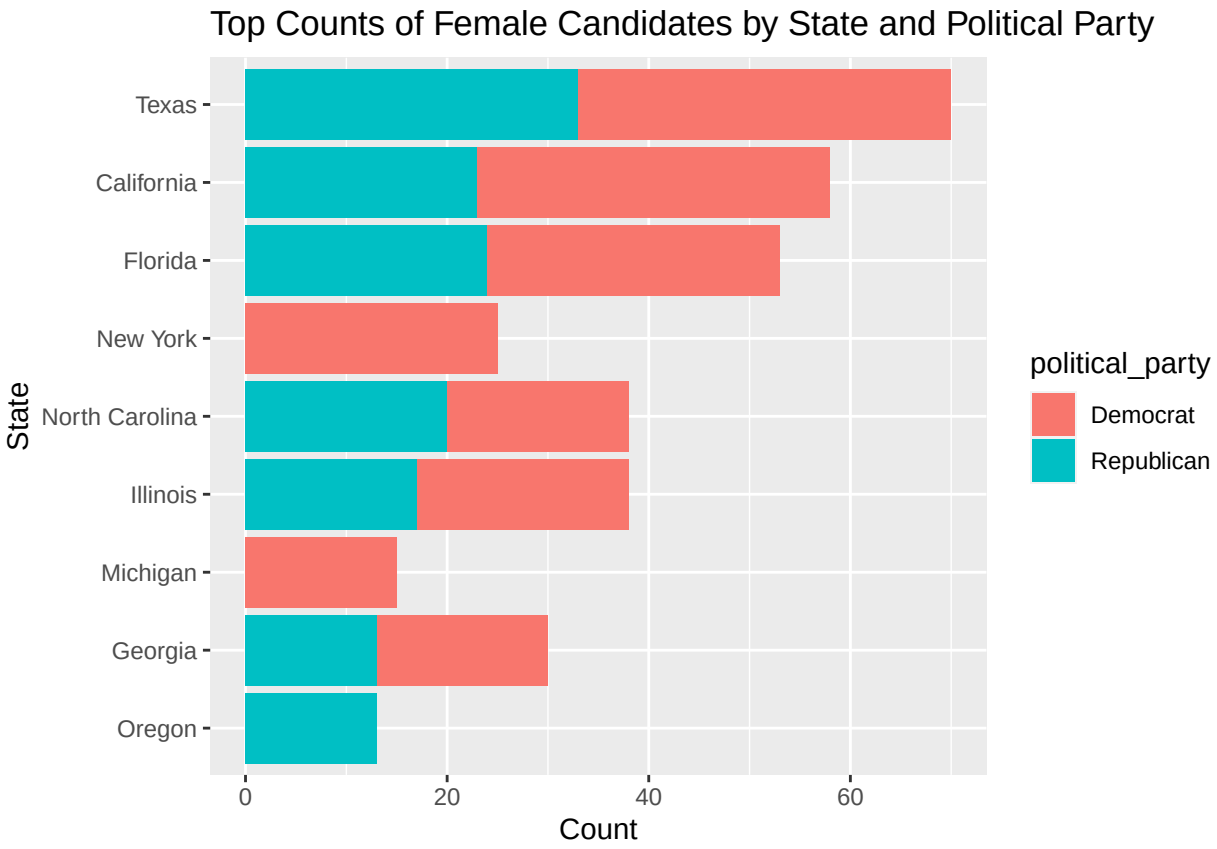
```
df_subset_bind_f %>%
  group_by(State) %>%
  summarise(n=n()) %>%  # Summarize by Count
  arrange(desc(n))  %>% # Arrange in descending order
  head(15)%>%           # Show only first 15 States
  # Use reorder to show States in descending order, adjust bar plot default as we have already summariz
  # and add in steel blue as the bar color, use coord_flip to show State on the y-axis, and label our g
  ggplot( aes(x = reorder(State, n), y = n) ) +  geom_bar(stat="identity", fill="steelblue") + coord_fli
```



What about if we also group by Political Party?

```
df_subset_bind_f %>%
  group_by(State, political_party) %>% # Add political_party
  summarise(n=n()) %>%
  arrange(desc(n))  %>%
  head(15)%>%
# Note that as we grouped by State and political party, that head(15) will cause our plot to show the f
  ggplot( aes(x = reorder(State, n), y = n, fill = political_party) ) +  geom_bar(stat="identity") + co
        x ="State", y = "Count")
```

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



These are just a few ways to use the ggplot2 package to visualize results and adjusting plots to better fit our data.