SICSS_BootCamp

R Basics Quiz

Question 1

What arguments can you use for the mean function in R?

Syntax: mean(x, trim, na.rm)

x=an R object

trim=the fraction of observations to be trimmed from each end of x before the mean is computed.

na.rm=indicates whether NA values should be stripped before the mean is computed

Question 2

Extract the third element of the vector using numerical indexing.

```
random_vector <- c("R","is","great")
```

```
random_vector[3]
```

```
## [1] "great"
```

Question 3

Use R code to identify the data type of some_vector. What is the largest number in this vector? How about the mean value?

```
some_vector <- c(25555,342343,123123123,4234234,53243234,54324234,5421111,12312312,111231,1231231,12312
```

```
typeof(some_vector)
```

```
## [1] "double"
```

```
max(some_vector)
```

```
## [1] 1111233333
```

```
mean(some_vector)
```

```
## [1] 59756995
```

Question 4

How many rows and columns does the congress dataframe have? Use a function to show its data type. You must use R code to generate these values.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
```

```
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
load(url('https://dssoc.github.io/datasets/congress.RData'))
```

```
nrow(congress)
```

```
## [1] 539
```

```
ncol(congress)
```

```
## [1] 8
```

```
typeof(congress)
```

```
## [1] "list"
```

Question 5

What is the average age of all congress members? What is the data type of the birthyear column?

```
avg_year <- as.integer(mean(congress$birthyear))
avg_age <- 2022-avg_year
avg_age
```

```
## [1] 61
```

```
typeof(congress$birthyear)
```

```
## [1] "double"
```

Question 6

How much older is Sherrod Brown (a member of congress) compared to the average of members of congress? How about Dianne Feinstein?

```
brown_row <- congress |> filter(full_name=="Sherrod Brown")
brown_age <- 2022 - brown_row$birthyear
brown_age - avg_age
```

```
## [1] 9
```

Sherrod Brown is 9 years older than the average members of congress.

```
feinstein_row <- congress |> filter(full_name=="Dianne Feinstein")
feinstein_age <- 2022 - feinstein_row$birthyear
feinstein_age - avg_age
```

```
## [1] 28
```

Diane Feinstein is 28 years older than the average members of congress.

Question 7

Who are the oldest members of congress?

```
congress <- arrange(congress, birthyear)
```

```
head(congress)
```

```
##   bioguide_id             full_name type      party state  birthdate gender
## 1     F000062      Dianne Feinstein  sen   Democrat    CA 1933-06-22      F
## 2     G000386        Chuck Grassley  sen Republican    IA 1933-09-17      M
## 3     Y000033             Don Young  rep Republican    AK 1933-06-09      M
## 4     I000024       James M. Inhofe  sen Republican    OK 1934-11-17      M
## 5     S000320     Richard C. Shelby  sen Republican    AL 1934-05-06      M
## 6     J000126 Eddie Bernice Johnson  rep   Democrat    TX 1935-12-03      F
##   birthyear
## 1      1933
## 2      1933
## 3      1933
## 4      1934
## 5      1934
## 6      1935
```

Dianne Feinstein, Chuck Grassley, and Don Young are the oldest members of congress.

Data Wrangling Quiz

Question 1

Describe what the following tidyverse functions do. Also describe the pipe operator "%>%". You may need to look up the official documentation for each of these.

filter–selects rows

select–selects columns

mutate–allows you to change an existing variable or add a new variable

count–gives you the number of each observations of a given variable_name in a dataframe

arrange–sorts dataframe in alphabetical order by variable_name

gather–transforms data into long structure (wide structure==lots of columns; long==lots of rows)

pipe operator–channel data into a function

Question 2

Create a new dataframe that includes only senators and the columns gender, birthyear, and party. Then use that new dataframe to compute the number of male and female democrats and republicans (the output should be five rows corresponding to female democrats, male democrats, male independents, female republicans, and male republicans).

Limit to senators, then select columns gender, birthyear, party

```
df <- congress |> filter (type=='sen') |> select('gender', 'birthyear', 'party', 'birthdate')
```

compute the number of male and female democrats and republicans

```
count(df, gender, party)
```

```
##   gender       party  n
## 1      F    Democrat 16
## 2      F  Republican  8
## 3      M    Democrat 32
## 4      M Independent  2
## 5      M  Republican 42
```

Question 3

Identify the oldest and youngest male and female democrat senators using tidyverse functions.

```
df2 <- congress |> filter(gender=='M' & party=='Democrat')|> arrange(birthyear)
```

```
df2[1,]
```

```
##   bioguide_id          full_name type    party state  birthdate gender
## 1     P000096 Bill Pascrell, Jr.  rep Democrat    NJ 1937-01-25      M
##   birthyear
## 1      1937
```

The oldest male democratic member of congress is Bill Pascrell, Jr.

```
df2 <- congress |> filter(gender=='F' & party=='Democrat')|>arrange(birthyear)
```

```
df2[1,]
```

```
##   bioguide_id        full_name type    party state  birthdate gender birthyear
## 1     F000062 Dianne Feinstein  sen Democrat    CA 1933-06-22      F      1933
```

The oldest female democratic member of congress is Dianne Feinstein.

Question 4

Using mutate, create a new variable called age which represents the approximate age of each member of congress. How many democratic senators are over 60 years old?

```
df <- congress |> filter(party=='Democrat' & type=='sen') |> mutate (age=2022-birthyear)
```

```
df2 <- df |> filter(age>60)
nrow(df2)
```

```
## [1] 32
```

Question 5

Create a new column that indicates whether or not the member of congress is more than 55 years old, and create a single dataframe showing the number of male and female members of congress that are over and under 55.

```
df <- congress |> mutate(age=2022-birthyear)
df <- df |> mutate(over55=ifelse(age>55,1, 0))
```

```
count(df, gender, over55)
```

```
##   gender over55   n
## 1      F      0  47
## 2      F      1 100
## 3      M      0 121
## 4      M      1 271
```

Question 6

Using gather, create a new dataframe where each row corresponds to a valid twitter, facebook, or youtube social media account, then compute the total number of accounts for each political party. Then do the same with pivot_longer.

```
long_df <- gather(congress_contact, key='name', value='social_account', 'twitter': 'youtube')
```

```
joined_df <- inner_join(congress, long_df)
```

```
## Joining, by = "bioguide_id"
```

```
all3accounts_join <- joined_df |> filter(nchar(social_account)>0)
```

```
all3accounts_joinSummary <- all3accounts_join |>group_by(party)|> summarise(n=n())
all3accounts_joinSummary
```

```
## # A tibble: 3 x 2
##   party           n
##   <fct>       <int>
## 1 Democrat      616
## 2 Independent     6
## 3 Republican    537
```

Perform the same process using pivot_longer

```
pivot_longer_form <- congress_contact |>
  pivot_longer("twitter":"youtube", names_to = "social_account")
```

```
joined_df2 <- inner_join(congress, pivot_longer_form)
```

```
## Joining, by = "bioguide_id"
```

```
all3accounts_join2 <- joined_df2 |> filter(nchar(value)>0)
```

```
all3accounts_join2Summary <- all3accounts_join2 |>group_by(party)|> summarise(n=n())
all3accounts_join2Summary
```

```
## # A tibble: 3 x 2
##   party           n
##   <fct>       <int>
## 1 Democrat      616
## 2 Independent     6
## 3 Republican    537
```

Question 7

Write code to print only the states who implemented both travel restrictions and mask requirements.

```
travel_restrictions <- c("WA", "OR", "NV", "CA", "NM", "MN", "IL", "OH", "MI", "PA", "VA", "NY", "MA",

require_masks <- c("HI", "WA", "OR", "NV", "CA", "MT", "CO", "NM", "KS", "TX", "MN", "AR", "LA", "WI",
```

```
intersect(travel_restrictions, require_masks)
```

```
## [1] "WA" "OR" "NV" "CA" "NM" "MN" "IL" "OH" "MI"
```

Question 8

Write code to print the states who had implemented mask requirements but not travel restrictions.

```
setdiff(require_masks, travel_restrictions)
```

```
##  [1] "HI" "MT" "CO" "KS" "TX" "AR" "LA" "WI" "AL" "IN"
```

Visualization Quiz

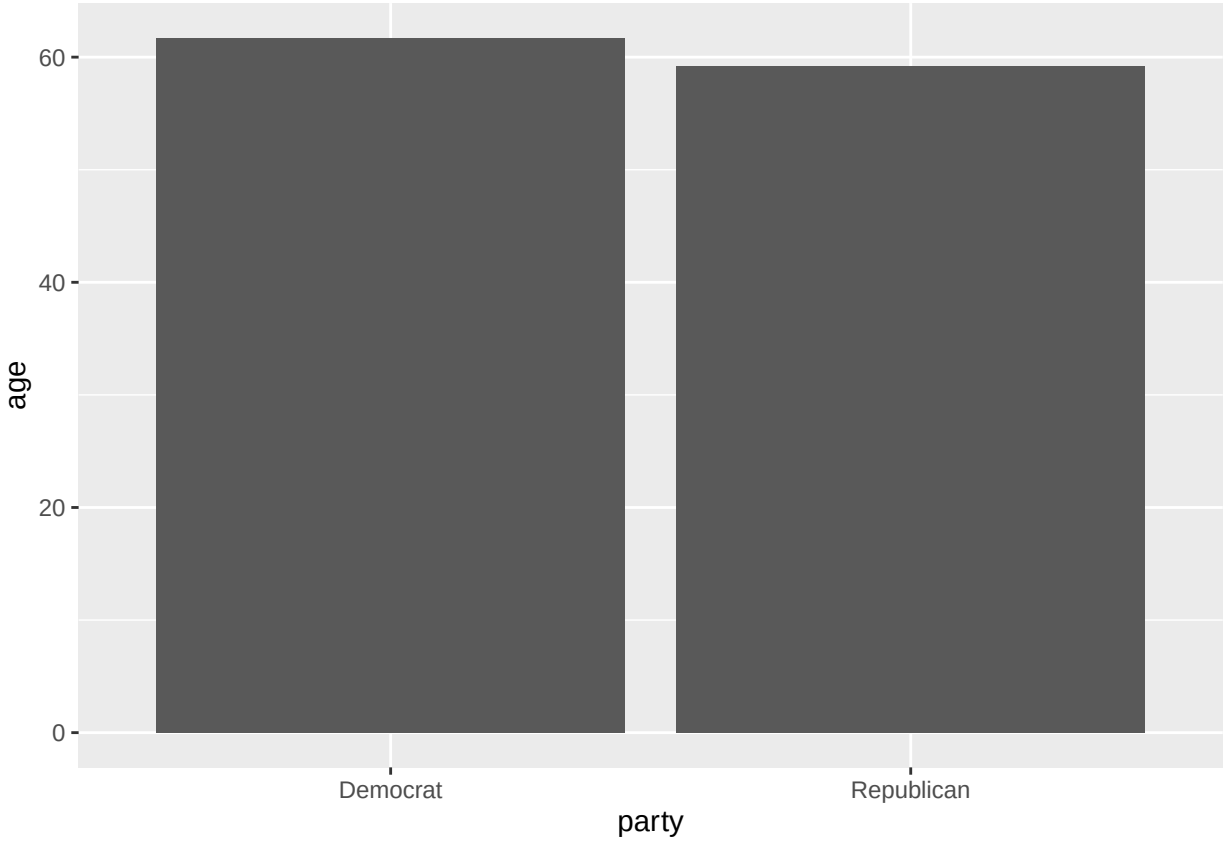
Question 1

Create a bar plot to show the average ages of democrat and republican congress members. Now do the same for M and F genders (this second part should include members of all parties).

```
library(ggplot2)
```

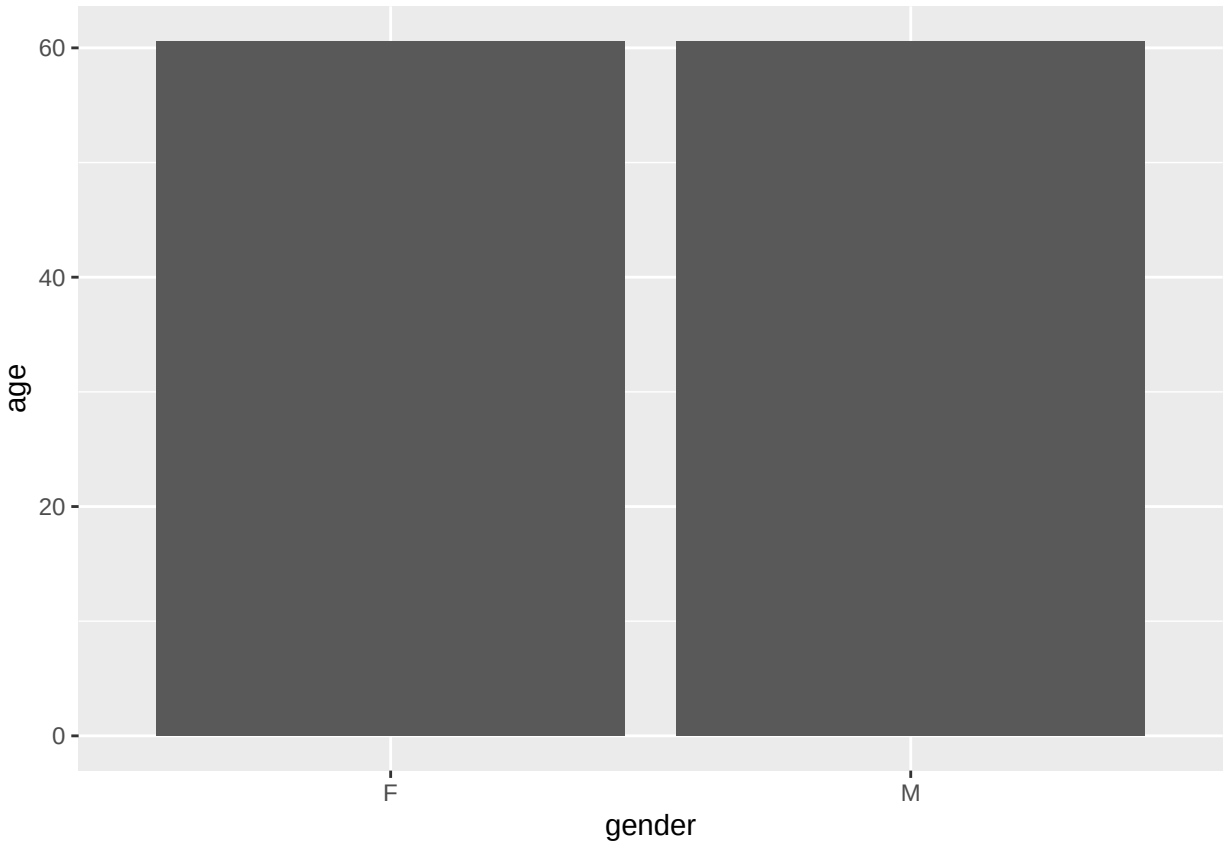
```
plotdf <- df |> filter(party=='Democrat' | party=='Republican')
```

```
# Basic barplot
p<-ggplot(data=plotdf, aes(x=party, y=age)) +
  geom_bar(stat = "summary", fun = "mean")
p
```



Do same thing for gender

```
p<-ggplot(data=df, aes(x=gender, y=age)) +
  geom_bar(stat = "summary", fun = "mean")
p
```



Question 2

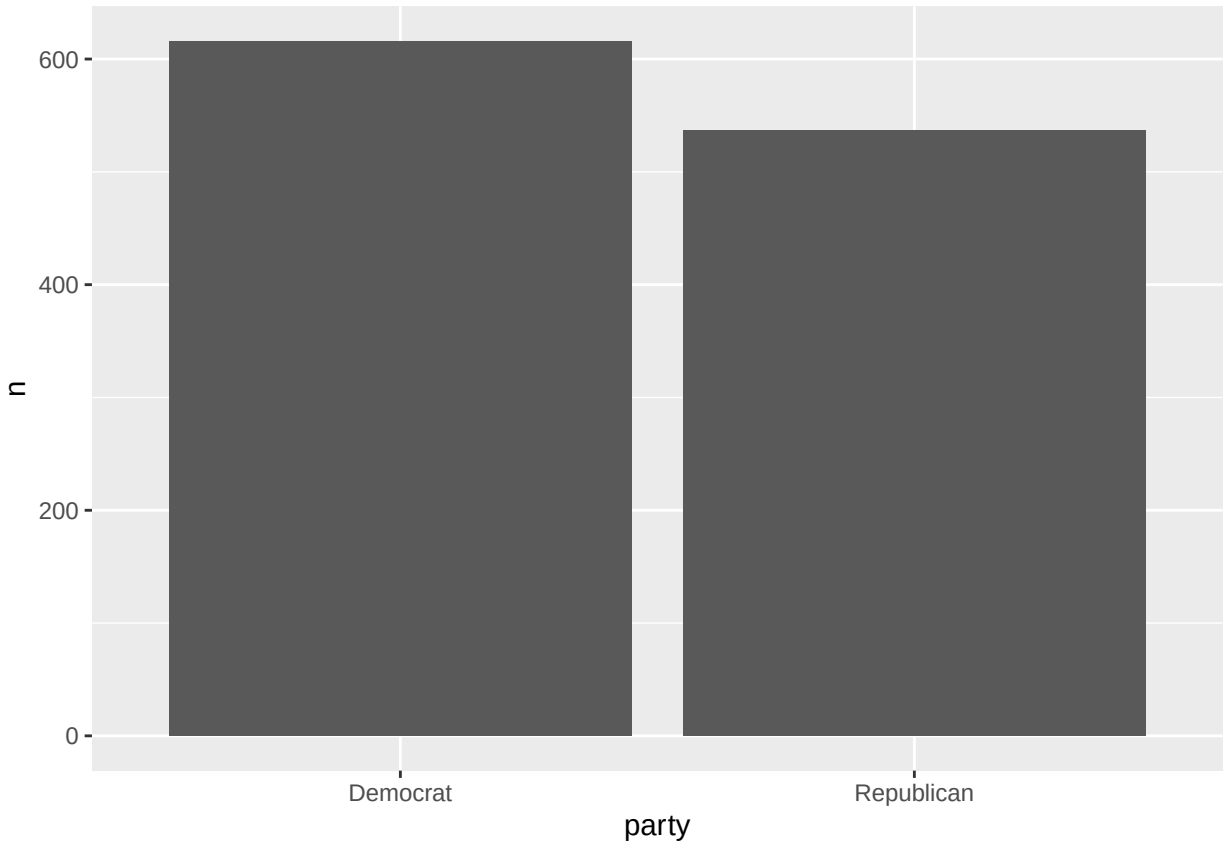
Create two bar charts: one that shows the total number of social media accounts among democrats and republicans (Twitter, Facebook, YouTube), and one that shows the average number of accounts per-politician for each party. Which political party has more social media accounts? Which party has a higher per-politician average?

Note: there are several ways to accomplish this. You could use gather again and then group_by and summarise within politician and then within party, or you could use mutate to get counts for each politican and then average by party. Any other approach is also fine.

```
all3accounts_join <- all3accounts_join |> filter(party=='Democrat' | party=='Republican') |> group_by(pa
```

Graphing total number of accounts by party

```
p<-ggplot(data=all3accounts_join, aes(x=party, y=n)) +
  geom_bar(stat = "identity")
p
```



Democrats have more social media accounts.

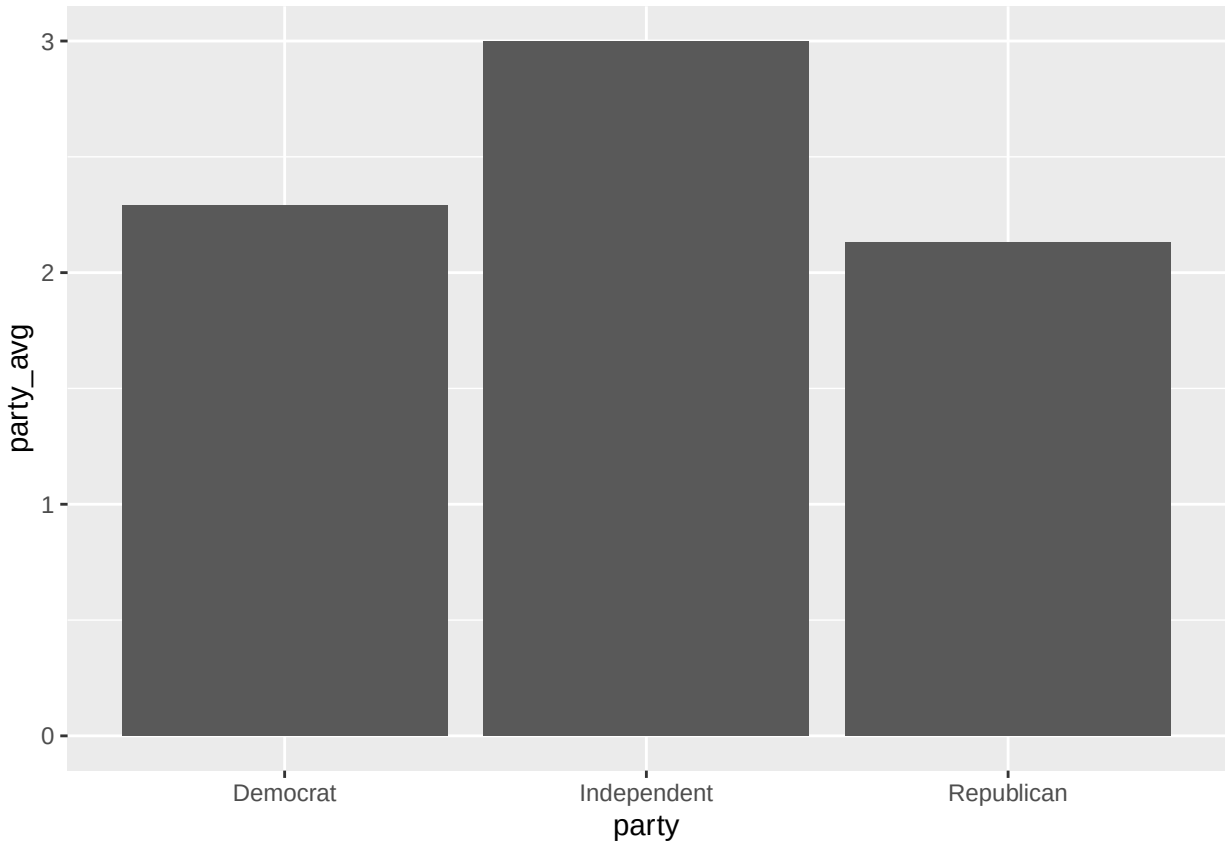
Creating plot that shows the average number of accounts per-politician for each party

```
all3accounts <- all3accounts_join2 |> group_by(bioguide_id, party) |> summarise(n=n()) |> group_by(party
```

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

Graphing avg number of accounts per politician by party

```
p<-ggplot(data=all3accounts, aes(x=party, y=party_avg)) +
  geom_bar(stat = "identity")
p
```



Independents, then democrats, then republicans have highest per politician average.

Question 3

Use an inner join to combine the columns of the committees dataframe with the columns of congress, and create a plot showing the average number of committees that democrats and republicans belong to. Next create a plot showing the averages by gender (note: this second part should include members of other parties as well).

```
load(url('https://dssoc.github.io/datasets/committees.RData'))
```

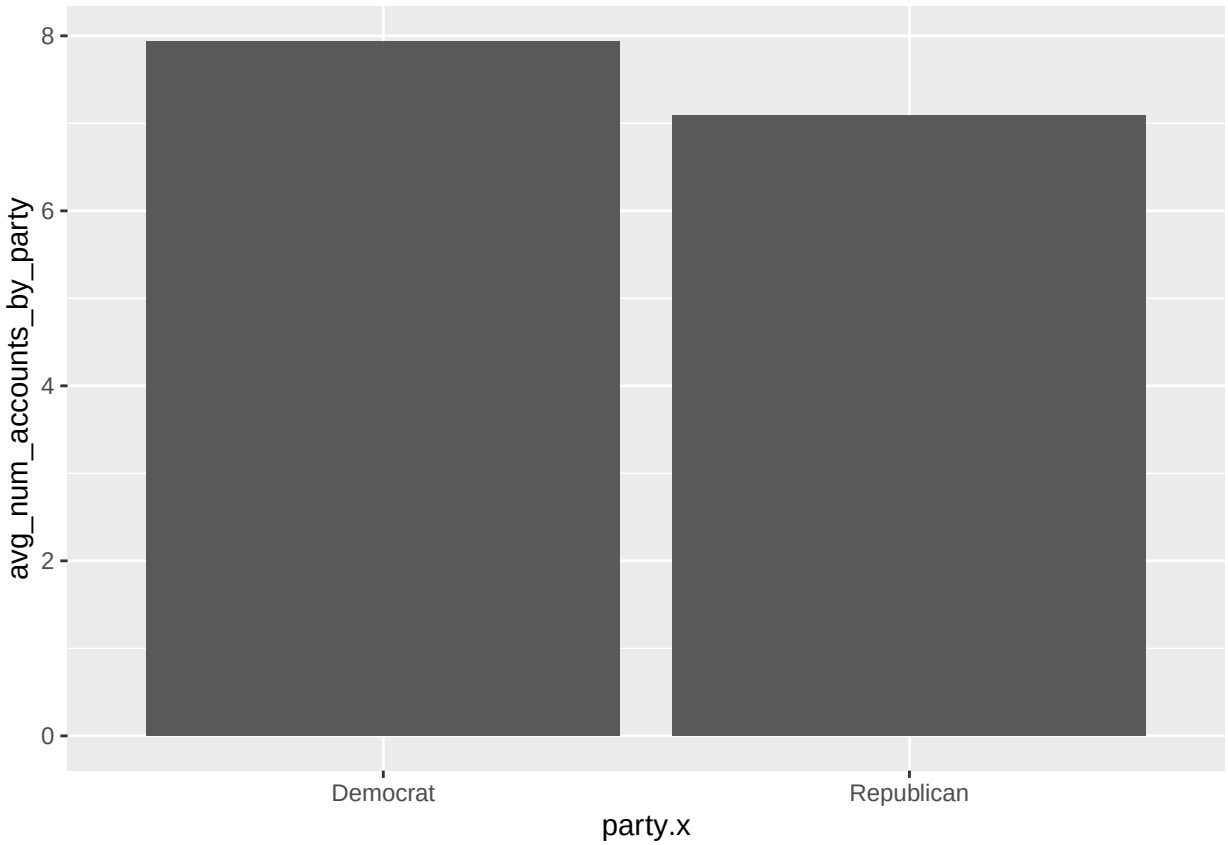
```
combined <- inner_join(congress, committee_memberships, "bioguide_id")
```

```
combined <- combined |> group_by(bioguide_id, party.x) |> summarise(n=n()) |> group_by(party.x) |> summa
```

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

```
combined <- combined |> filter(party.x!='Independent')
```

```
p<-ggplot(data=combined, aes(x=party.x, y=avg_num_accounts_by_party)) +
  geom_bar(stat = "identity")
p
```



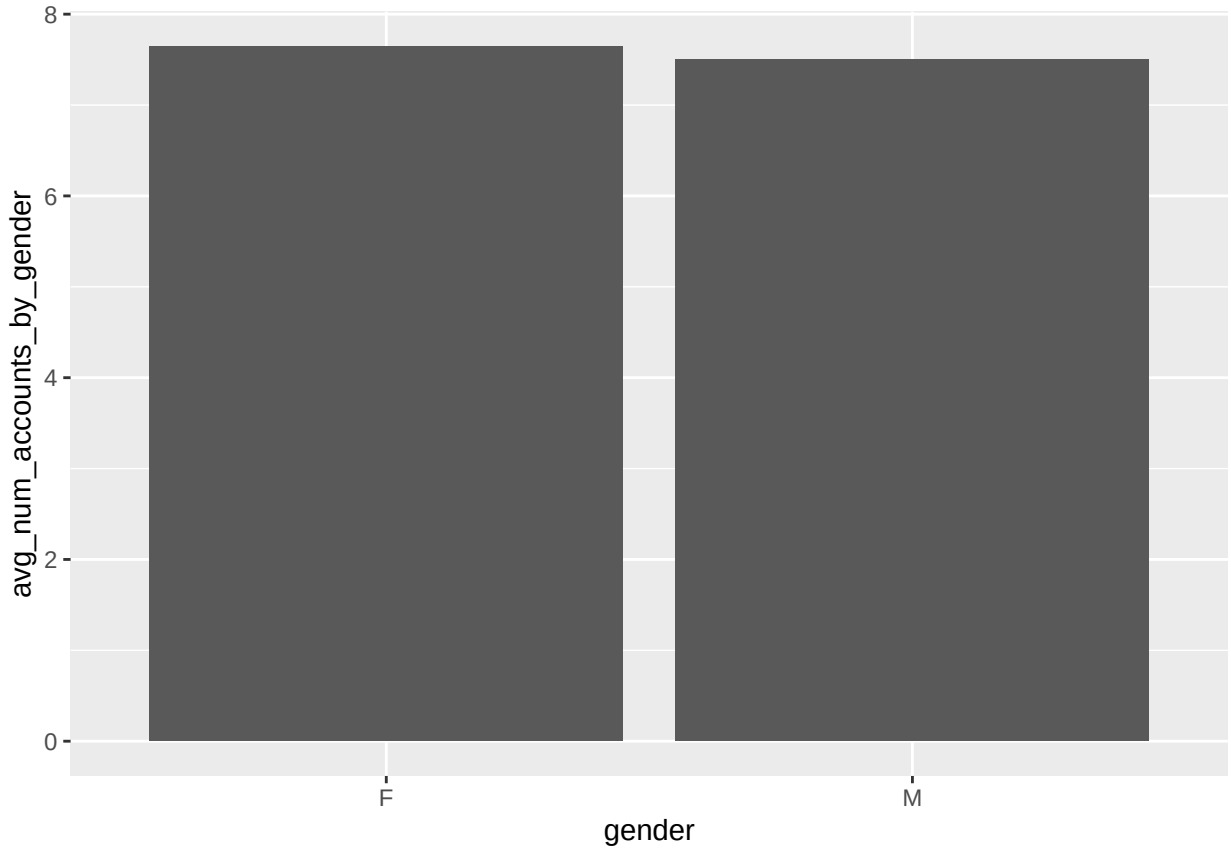
Plotting same average by gender

```
combined <- inner_join(congress, committee_memberships, "bioguide_id")
```

```
combined <- combined |> group_by(bioguide_id, gender) |> summarise(n=n()) |> group_by(gender) |> summari
```

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

```
p<-ggplot(data=combined, aes(x=gender, y=avg_num_accounts_by_gender)) +
  geom_bar(stat = "identity")
p
```



Question 4

Create a bar plot showing the number of members that belong to the 10 largest congressional committees (i.e. committees with the largest number of members). The bars should be sorted based on committee sizes.

Note: Our standard for visualizations is that each plot should have axis labels, all labels must be readable, and we should easily be able to tell what your figure is showing. Failure to do this will result in point deductions.

```
combined <- inner_join(congress, committee_memberships, "bioguide_id")
```

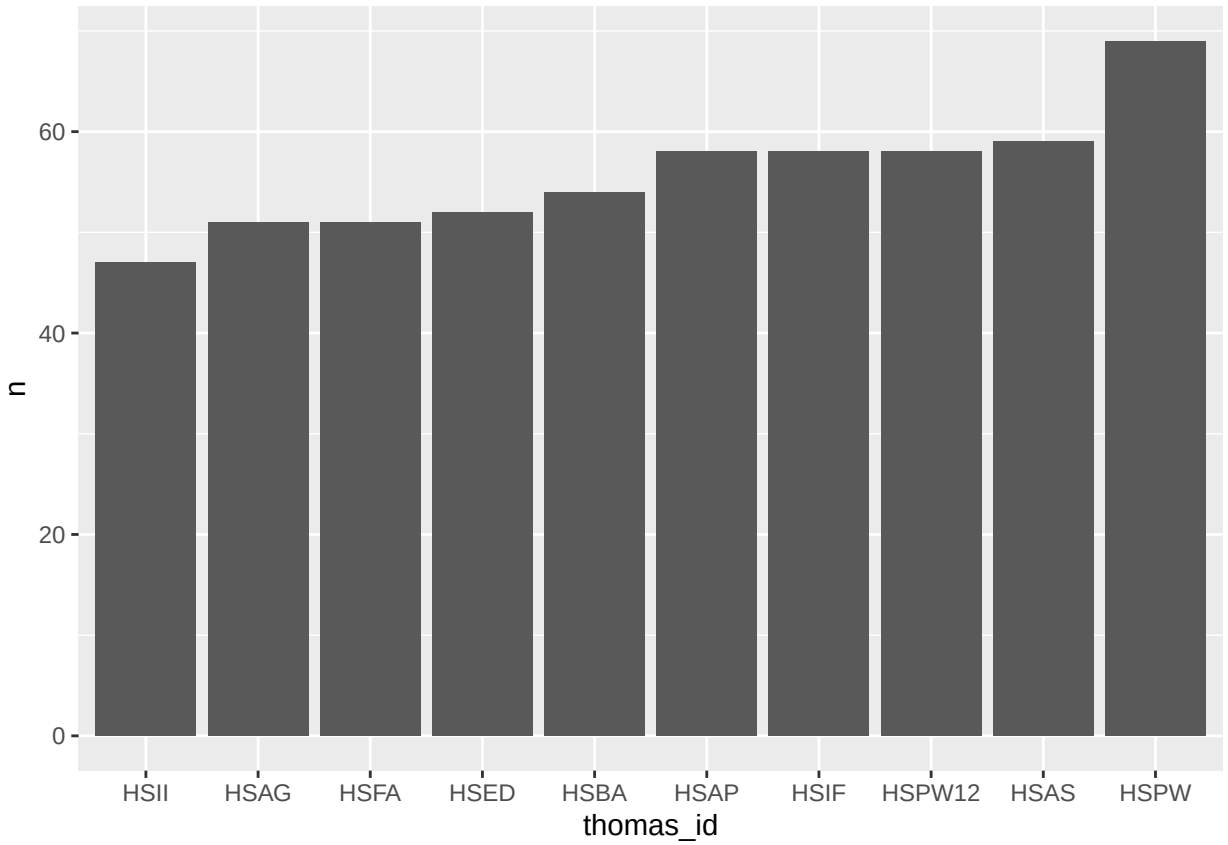
```
combined <- combined |> group_by(thomas_id) |> summarise(n=n())
```

```
combined <- combined[order(combined$n),]
```

```
top_ten <- tail(combined, n=10)
```

```
top_ten$thomas_id <- as.vector(top_ten$thomas_id) #get rid of factors
top_ten$thomas_id <-  factor(top_ten$thomas_id,top_ten$thomas_id)
```

```
p<-ggplot(data=top_ten, aes(x=thomas_id, y=n)) +
  geom_bar(stat = "identity")
p
```



Question 5

Create a single bar plot that shows the average age of the committees with the 5 highest and lowest average ages. The bars should be sorted based on average committee ages. Which committees have the highest and lowest average ages?

```
combined <- inner_join(congress, committee_memberships, "bioguide_id")
```

```
combined <- combined |> mutate (age = as.numeric(Sys.Date()-birthdate)/365)
```

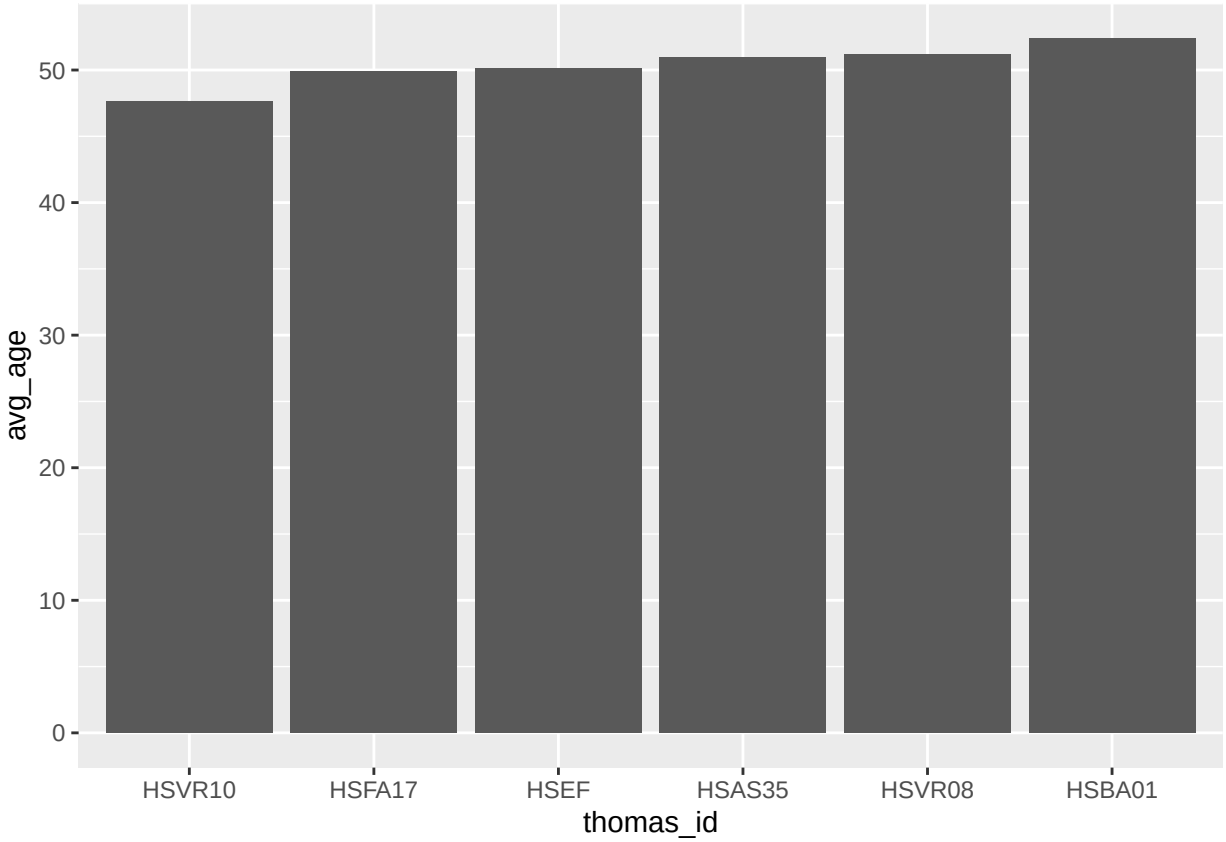
```
combined <- combined |> group_by(thomas_id) |> summarise(avg_age=mean(age))
```

```
combined <- combined[order(combined$avg_age),]
```

```
top_five <- head(combined)
bottom_five <- tail(combined)
```

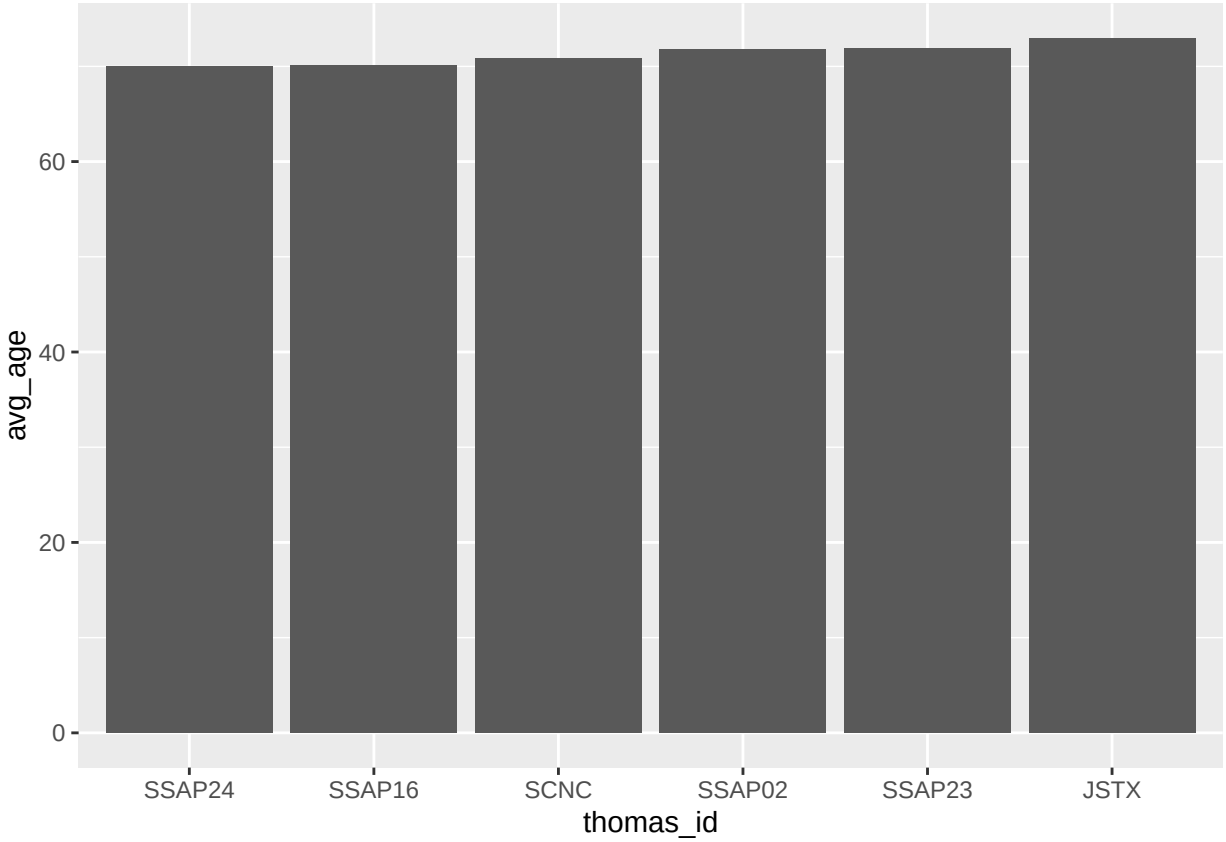
```
top_five$thomas_id <- as.vector(top_five$thomas_id) #get rid of factors
top_five$thomas_id <-  factor(top_five$thomas_id,top_five$thomas_id)
```

```
p<-ggplot(data=top_five, aes(x=thomas_id, y=avg_age)) +
  geom_bar(stat = "identity")
p
```




```
bottom_five$thomas_id <- as.vector(bottom_five$thomas_id) #get rid of factors
bottom_five$thomas_id <-  factor(bottom_five$thomas_id,bottom_five$thomas_id)
```

```
p<-ggplot(data=bottom_five, aes(x=thomas_id, y=avg_age)) +
  geom_bar(stat = "identity")
p
```



Question 6

Create a line graph showing the total number of politician births in each decade since the 1930's, with separate lines for senate and house members (see the type column). The labels on your x-axis should look like "1930's", "1940's", and so on, and your legend should show values "Senator" and "Representative" (i.e. not rep and sen).

Note: The plotted lines may not be continuous if there were no births in some decades.

```
combined <- inner_join(congress, committee_memberships, "bioguide_id")
```

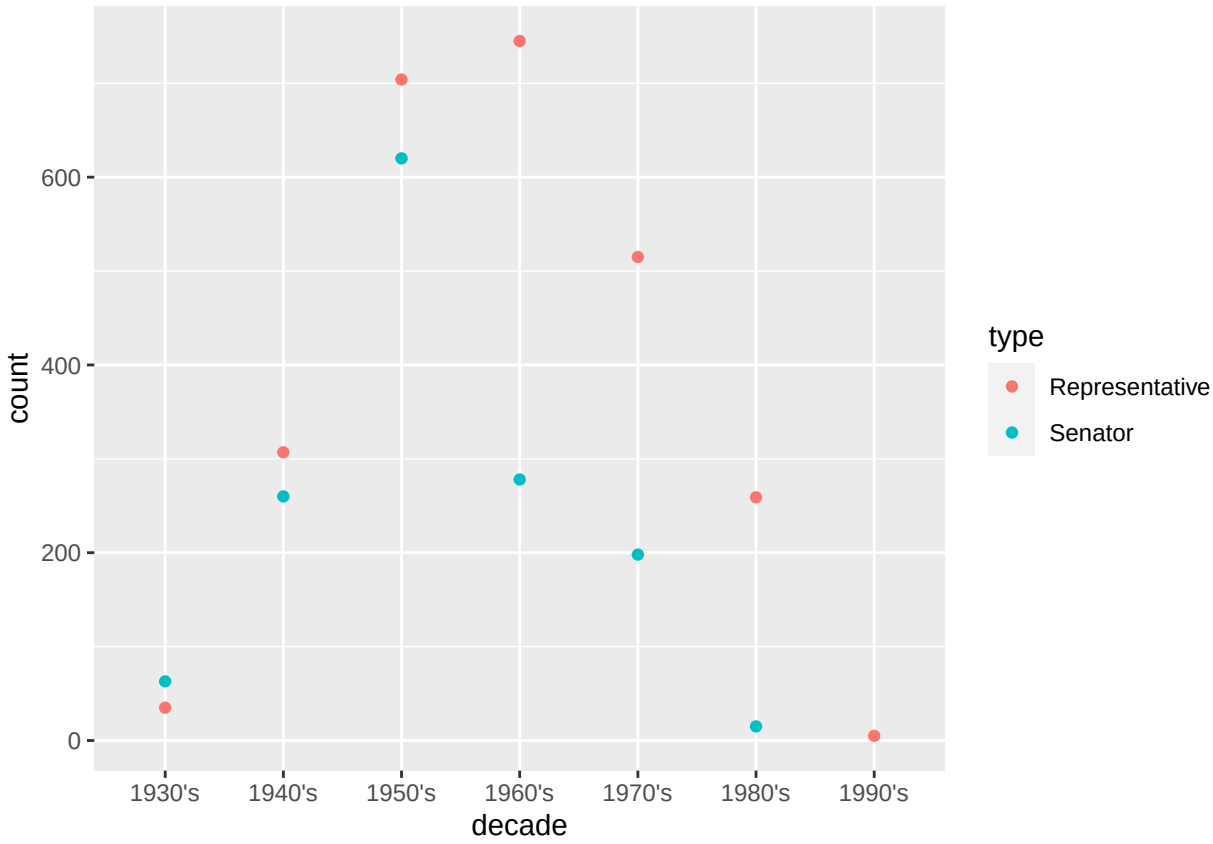
```
combined <- combined |> mutate(decade=case_when(substr(as.character(birthyear), 3, 3) == "3"~"1930's", s
```

```
combined <- combined |> group_by(type, decade) |> summarise(count=n())
```

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

```
combined <- combined |> mutate(type=if_else(type=="sen", "Senator", "Representative"))
```

```
p<-ggplot(data=combined, aes(x=decade, y=count, color=type)) +
  geom_point()
p
```



Question 7

Create a bar chart showing the frequency of politician births by month and another bar chart showing politician births by weekday. The x-labels should be month names and weekday names, respectively, and appear in chronological order.

Note: you can use the lubridate package methods to get weekday and month names.

```
library(lubridate)
```

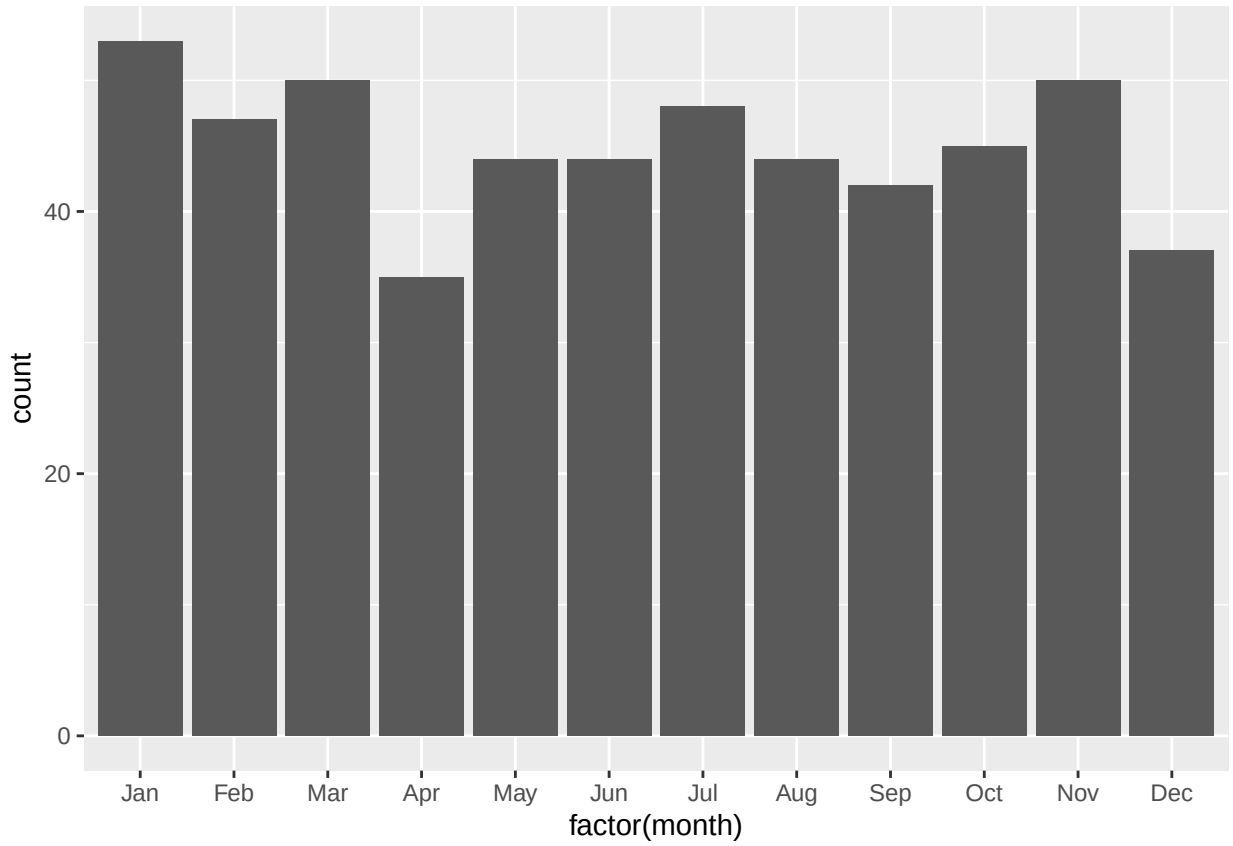
```
##
## Attaching package: 'lubridate'
```

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

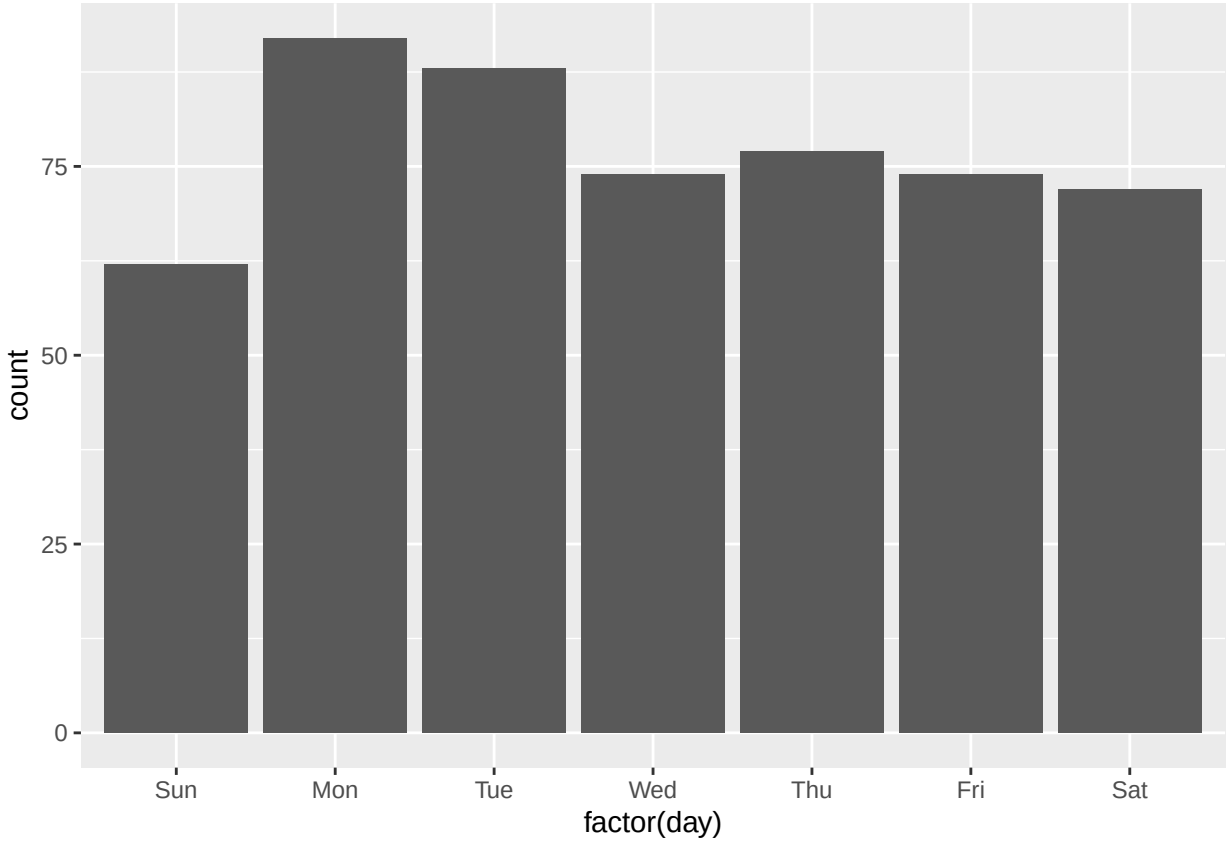
```
#extract month name
df <- congress|> mutate(month=month(birthdate, label=TRUE))

# extract day name
df <- df|> mutate(day=wday(birthdate, label=TRUE))
```

```
#graphing month
p<-ggplot(data=df, aes(x=factor(month))) +
  geom_bar()
p
```




```
#graphing day
p<-ggplot(data=df, aes(x=factor(day))) +
  geom_bar()
p
```



Basic Programming Quiz

Question 1

In your own words, describe what a function is and provide one example of how you might use it in a data science.

A function is a method for performing the same operation on different inputs without having to retype the operative code each time.

Question 2

Packages in R can contain many useful functions/commands. If you didn't know what a certain function did, or how it worked, where within RStudio would you look to learn more / see example code? Where would you look outside RStudio?

Within RStudio: help() function Outside RStudio: R documentation

Question 3

Write a function that takes a character vector as an argument and returns a character vector containing the first letters of each element in the original vector. To show that it works, test it on the character vector sentence defined below.

```
sentence <- c('you', 'only', 'understand', 'data', 'if', 'data', 'is', 'tidy')
```

```
first_char <- function(char_vector) {
  output_char_vector <- c()
  for(word in char_vector){
```

```
    output_char_vector <- append(output_char_vector, substr(word, 1, 1))
  }
  return (output_char_vector)
}
```

```
first_char(sentence)
```

```
## [1] "y" "o" "u" "d" "i" "d" "i" "t"
```

Question 4

Create your own function which accepts a birthyear vector and returns an approximate current age, then use it on the birthyear column of the congress dataframe to create a new age column with mutate.

```
year_to_age <- function(date) {

  age <- as.integer((today()-date)/365)

  return (age)
}
```

```
df <- congress |> mutate (age = year_to_age(birthdate))
```

Question 5

Write a function that accepts a date string and returns the day of the week it corresponds to, then use it to create a new column of congress representing the weekday of the birth of each politician using mutate.

```
date_to_weekday <- function(date) {

  weekday <- wday(date, label=TRUE)

  return (weekday)
}
```

```
df <- congress |> mutate (weekday = date_to_weekday(birthdate))
```

Question 6

Write a function that accepts a dataframe with the columns birthday and full_name, and prints the names and ages of the k oldest representatives in congress (i.e. not including senators) using a "for loop". In this sense, k is an arbitrary number that should be given as an argument to the function - set the default value to be 5. If you use the dataframe as the first argument, you can use the pipe operator ("%>%") to pass the dataframe directly to the function. Define your function such that you can use it like this: congress %>% print_oldest(3).

```
oldest_reps <- function(df, k) {
  df <- inner_join(congress, df, "full_name")
  df <- df|> filter(type=="rep")

  ##sort df by age
  df <- df|> mutate (age = year_to_age(birthdate.x))
  df <- df[order(df$age),]

  last_k <- tail(df, n=k)
  for(row in 1:k){
```

```
    name <- last_k[row, "full_name"]
    age <- as.character(last_k[row, "age"])
    full <- paste0(name, " ", age)
    print(full)
  }
}
```

```
df <- congress |> select(c('full_name', 'birthdate'))
```

```
df |> oldest_reps(10)
```

```
## [1] "Alan S. Lowenthal 81"
## [1] "Nancy Pelosi 82"
## [1] "Maxine Waters 83"
## [1] "Steny H. Hoyer 83"
## [1] "Harold Rogers 84"
## [1] "Grace F. Napolitano 85"
## [1] "Eleanor Holmes Norton 85"
## [1] "Bill Pascrell, Jr. 85"
## [1] "Eddie Bernice Johnson 86"
## [1] "Don Young 89"
```

Question 7

Starting with the function from the previous question, change it such that if k > 5, it only prints the first 5. Test it using this code: congress %>% print_oldest(100).

```
oldest_reps2 <- function(df, k) {
  df <- inner_join(congress, df, "full_name")
  df <- df|> filter(type=="rep")

  ##sort df by age
  df <- df|> mutate (age = year_to_age(birthdate.x))
  df <- df[order(df$age),]
  if (k>5){
    k <- 5
  }
  last_k <- tail(df, n=k)
  for(row in 1:k){
    name <- last_k[row, "full_name"]
    age <- as.character(last_k[row, "age"])
    full <- paste0(name, " ", age)
    print(full)
  }
}
```

```
df <- congress|> select(c('full_name', 'birthdate'))
```

```
df |> oldest_reps2(100)
```

```
## [1] "Grace F. Napolitano 85"
## [1] "Eleanor Holmes Norton 85"
## [1] "Bill Pascrell, Jr. 85"
## [1] "Eddie Bernice Johnson 86"
## [1] "Don Young 89"
```

Modeling Quiz

Question 1

In your own words, describe what statistical modeling means. When is it used? What does it allow data scientists to do?

Statistical modeling allows data scientists to gain a broad sense of trends and correlations in datasets by assessing associations and producing summary statistics.

Question 2

Create three new variables related to our congress dataset: (a) the age of the member, (b) the number of committees they are on, and (c) the percentage of instances where they hold a title in the committees they belong to (i.e. when the title entry in the committee membership dataframe is not empty). You will want to save these new variables for future problems. Then use the summary function to create summary statistics for these new variables.

```
updated_congress <- congress |> mutate (age = year_to_age(birthdate))
```

```
combined <- inner_join(updated_congress, committee_memberships, "bioguide_id")
```

```
num_committees <- combined |> group_by(bioguide_id) |> summarise(n=n())
```

```
combined <- inner_join(combined, num_committees, "bioguide_id")
```

```
perc_titles <- combined |> group_by(bioguide_id) |> mutate(title_count=if_else(!is.na(title), 1, 0)) |>
```

```
combined <- inner_join(combined, perc_titles, "bioguide_id")
```

```
combined <- combined |> mutate(title_perc=title_sum/n)
```

Select only one row for each name

```
new <- data.frame()

for(row in 1:nrow(combined)){
  id <- combined[row, "bioguide_id"]


  if (!(id %in% new$bioguide_id)){

    new <- rbind(new, combined[row, ])

  }
}
```

Creating summary statistics of these variables

```
summary(new)
```

```
##  bioguide_id         full_name          type            party.x
##  Length:531         Length:531         rep:431   Democrat   :270
##  Class :character   Class :character   sen:100   Independent:  2
##  Mode  :character   Mode  :character             Republican :259
##
##
##
##     state              birthdate          gender    birthyear         age
```

```
##  Length:531         Min.   :1933-06-09   F:144   Min.   :1933   Min.   :26.00
##  Class :character   1st Qu.:1953-04-01   M:387   1st Qu.:1953   1st Qu.:51.00
##  Mode  :character   Median :1961-03-07           Median :1961   Median :61.00
##                     Mean   :1961-12-13           Mean   :1961   Mean   :60.04
##                     3rd Qu.:1970-08-11           3rd Qu.:1970   3rd Qu.:69.00
##                     Max.   :1995-08-01           Max.   :1995   Max.   :89.00
##   thomas_id             party.y         rank          title
##  Length:531         majority:272   Min.   : 1.00   Length:531
##  Class :character   minority:259   1st Qu.: 5.00   Class :character
##  Mode  :character                  Median :10.00   Mode  :character
##                                    Mean   :11.53
##                                    3rd Qu.:17.00
##                                    Max.   :37.00
##        n           title_sum        title_perc
##  Min.   : 1.00   Min.   : 0.000   Min.   :0.0000
##  1st Qu.: 5.00   1st Qu.: 0.000   1st Qu.:0.0000
##  Median : 7.00   Median : 1.000   Median :0.1429
##  Mean   : 7.54   Mean   : 1.169   Mean   :0.1562
##  3rd Qu.: 9.00   3rd Qu.: 1.000   3rd Qu.:0.2000
##  Max.   :24.00   Max.   :10.000   Max.   :1.0000
```

Question 3

Create a linear model predicting the number of committees that members belong to from age, then create a scatter plot with a linear trendline. Describe the relationship. What do each of these (the model summary and the plot) show that you cannot see from the other?

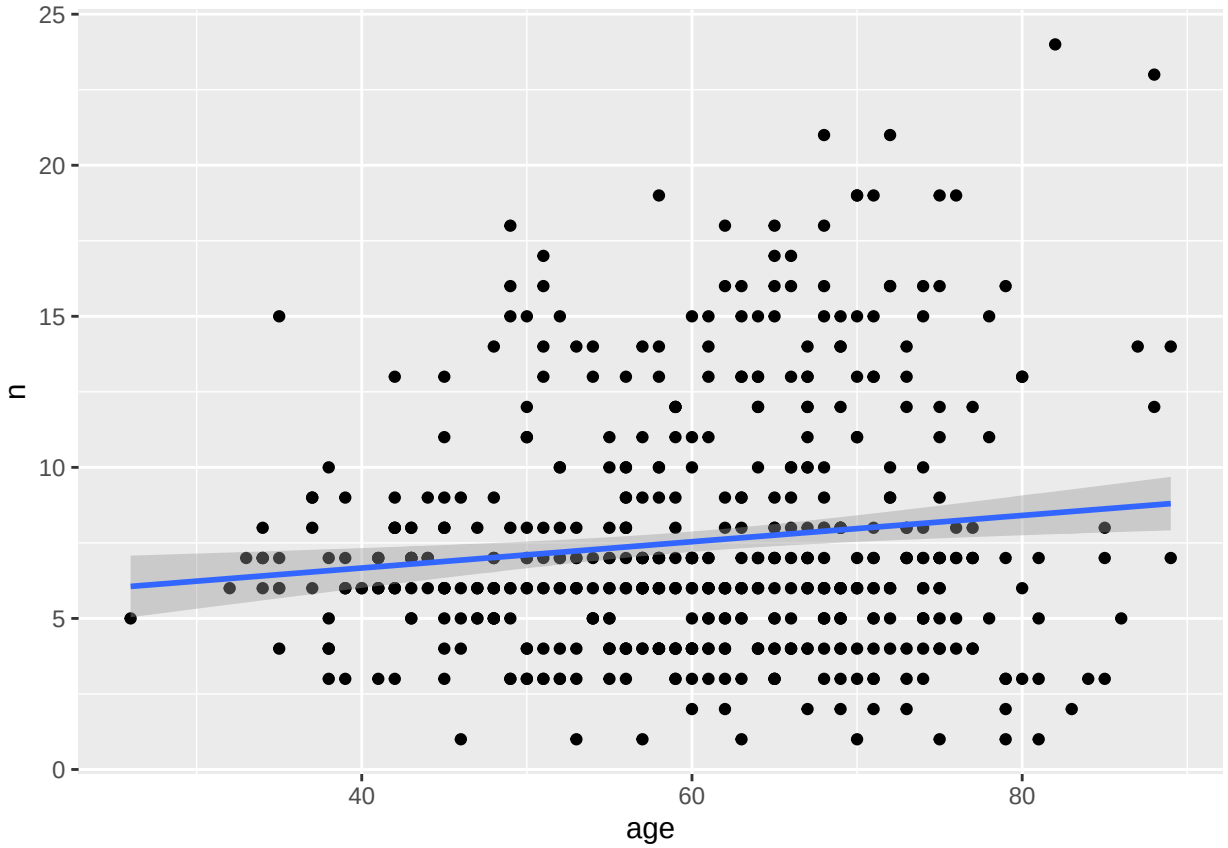
Note: usually we see the dependent variable (number of committees in this case) on the y-axis and the independent variable on the x-axis.

```
m1 <- lm(n~age, new)
summary(m1)
```

```
##
## Call:
## lm(formula = n ~ age, data = new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4524 -2.6257 -0.8861  1.5266 15.5041
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.92815    0.88051   5.597 3.51e-08 ***
## age          0.04351    0.01438   3.025  0.00261 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.961 on 529 degrees of freedom
## Multiple R-squared:  0.017,  Adjusted R-squared:  0.01515
## F-statistic: 9.151 on 1 and 529 DF,  p-value: 0.002607
```

```
p<-ggplot(data=new, aes(x=age, y=n)) +
  geom_point()+
  geom_smooth(method=lm)
p
```

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



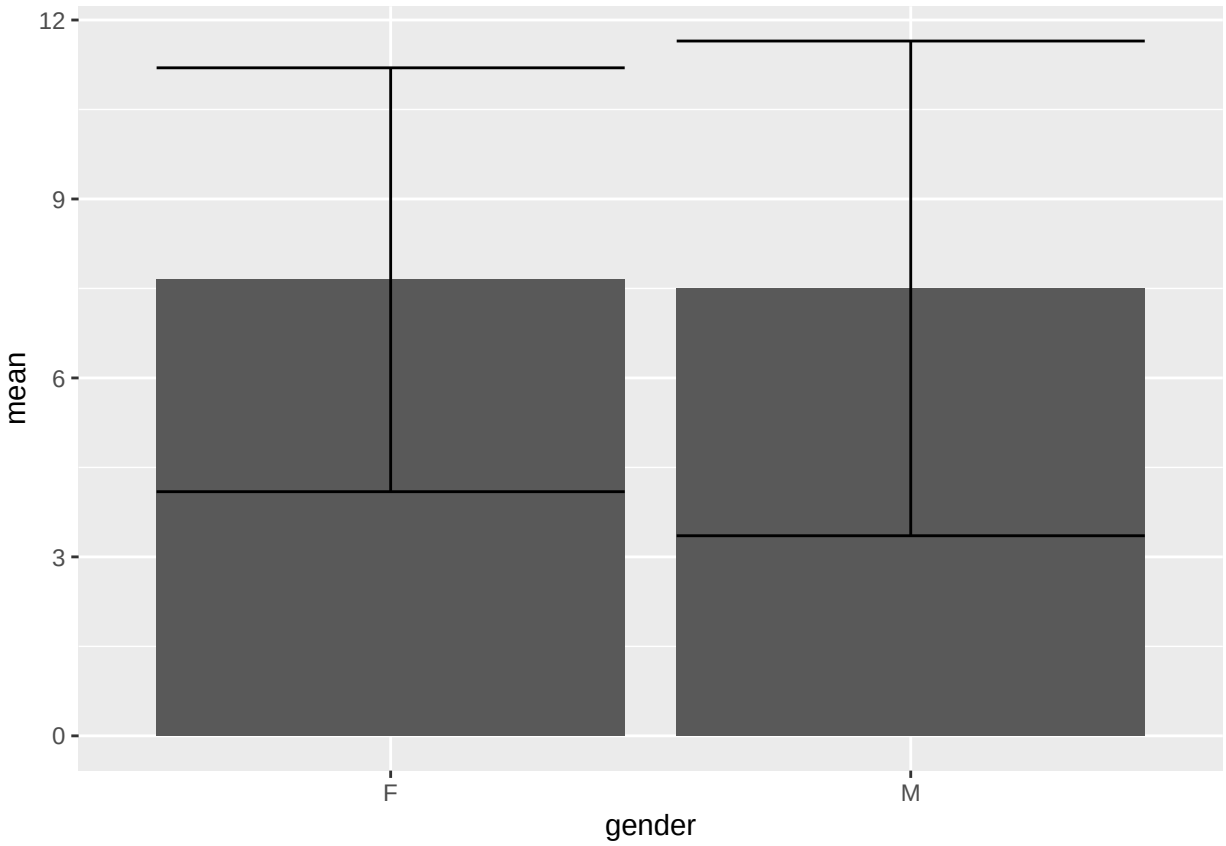
The plot shows the distribution of points while the summary statistics do not.

Question 4

Create a bar graph showing the average number of committees that congress members belong to by gender (i.e. a bar for M and a bar for F) with error bars. What can you see from this visualization? Does there appear to be a significant difference?

```
by_gender <- new |> group_by(gender) |> summarise(sd=sd(n), mean=mean(n))
```

```
# Basic barplot
p<-ggplot(data=by_gender, aes(x=gender, y=mean)) +
  geom_bar(stat="identity") +
  geom_errorbar(aes(ymin=mean-sd, ymax=mean+sd))
p
```



Question 5

Construct a model predicting the percentage of time that a member holds a title in the committees they belong to from age, gender, and political party. Which variables might be related to holding a title? Try removing and adding different variables. Does changing any of the used variables change your original interpretation?

```
m2 <- lm(title_perc~ age + gender + party.x, new)
summary(m2)
```

```
##
## Call:
## lm(formula = title_perc ~ age + gender + party.x, data = new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.23315 -0.11607 -0.02784  0.05950  0.89702
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -0.1254871  0.0427097  -2.938  0.00345 **
## age                 0.0043614  0.0006546   6.663 6.79e-11 ***
## genderM             0.0271658  0.0180992   1.501  0.13397
## party.xIndependent -0.0784006  0.1273992  -0.615  0.53856
## party.xRepublican   0.0006747  0.0161955   0.042  0.96679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1785 on 526 degrees of freedom
## Multiple R-squared:  0.08221,    Adjusted R-squared:  0.07523
## F-statistic: 11.78 on 4 and 526 DF,  p-value: 3.6e-09
```

Age is significant at the 0.001 level, meaning that there is a less than 1% chance that we would see this level of variation in y by random chance (not caused by variation in x), or that there is a 1% chance that we would see results this extreme (this much variation in y) if x and y are independent.

Question 6

Use the model from the previous question to make a scatter plot that includes prediction lines for BOTH F and M Democrats. That is, your plot should include two prediction lines - one for M and one for F, and the visualization (not the model) should only include democrats. This is important because our original model included information about all the variables, but we mainly want to visualize a single relationship, and how it might differ by gender. How do you interpret this plot?

```
library(ggiraph)
```

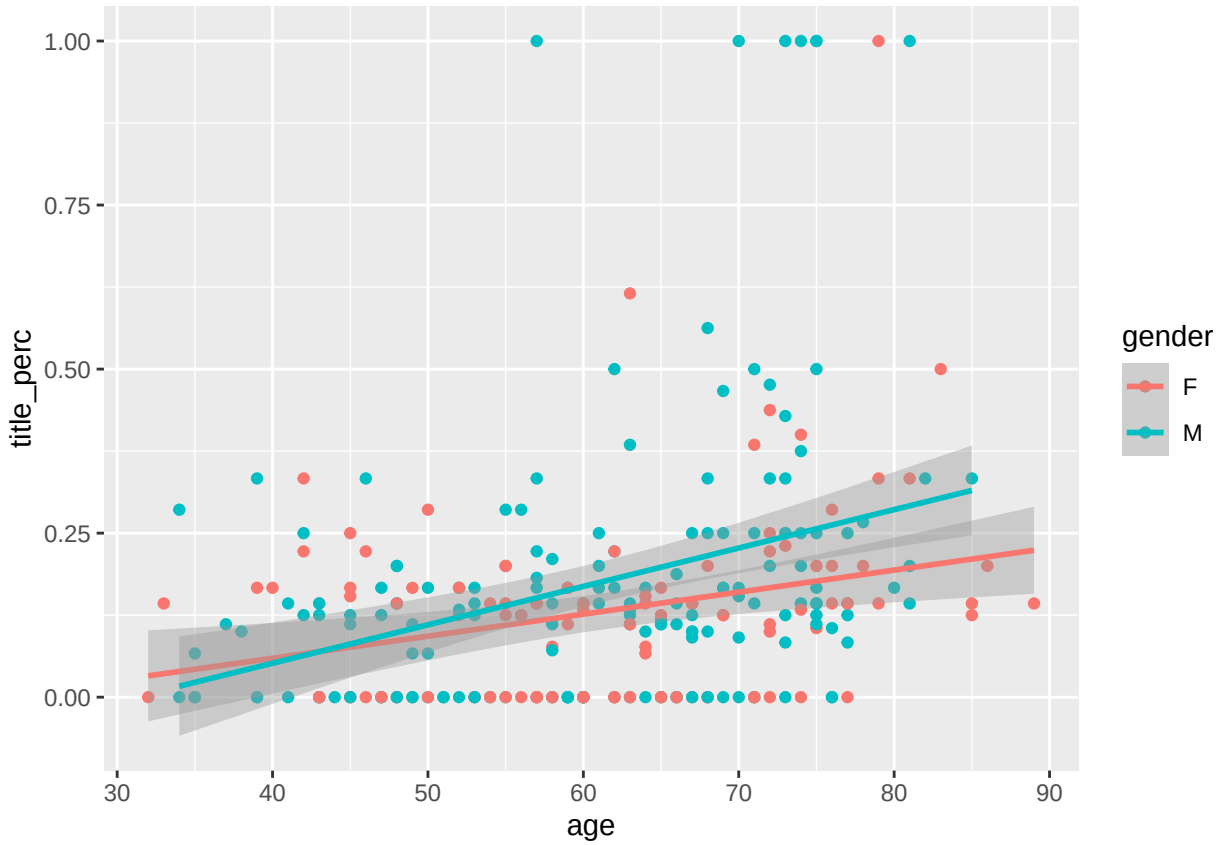
```
## Warning: package 'ggiraph' was built under R version 4.1.2
```

```
library(ggiraphExtra)
```

```
dem_only <- new |> filter(party.x=="Democrat")
```

```
p<-ggplot(data=dem_only, aes(x=age, y=title_perc, color=gender)) +
  geom_point()+
  geom_smooth(method=lm)
p
```

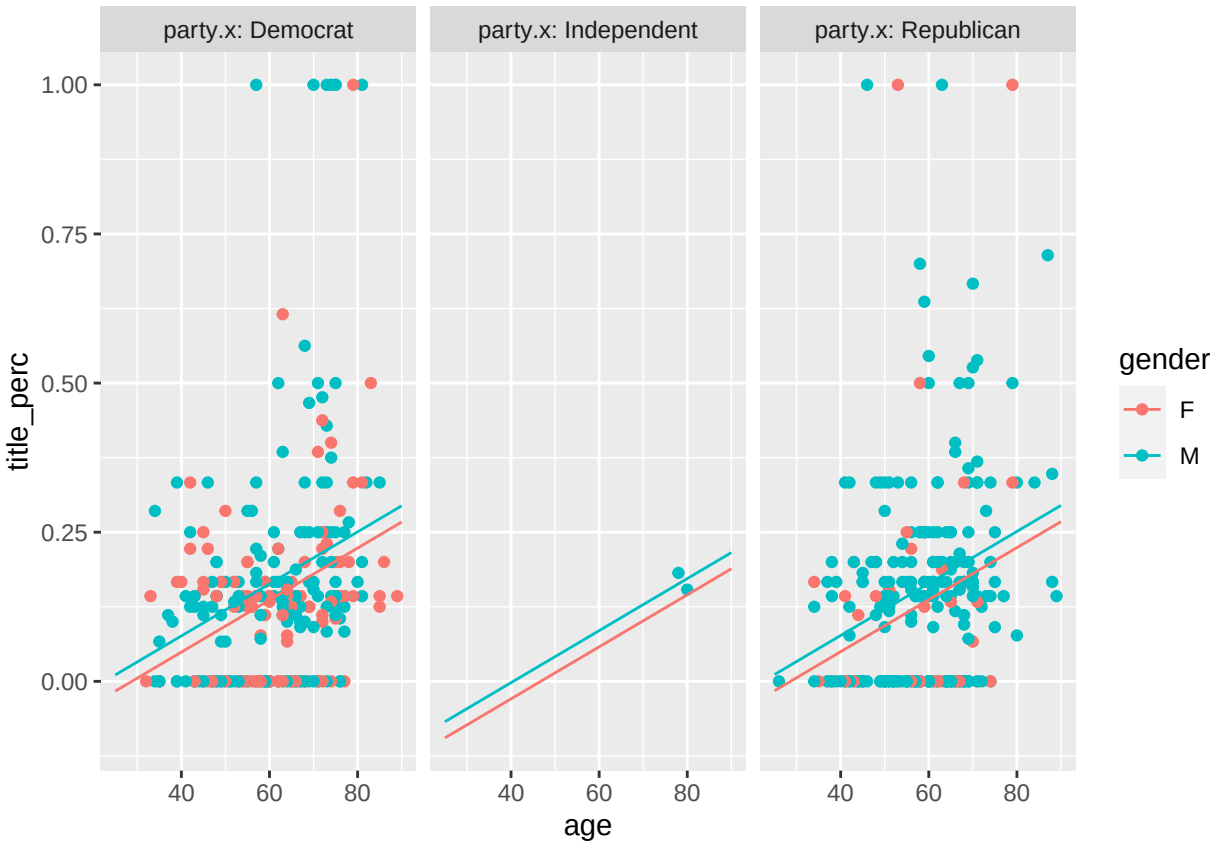
```
## `geom_smooth()` using formula 'y ~ x'
```



The positive relationship between age and percent of time a politician holds a committee title is stronger for men (age has a stronger impact on likelihood of holding a committee title for men).

Below is a plot showing the relationship for all parties.

```
p <- ggPredict(m2)
p
```


party.x: Democrat
party.x: Independent
party.x: Republican
title_perc
1.00
0.75
0.50
0.25
0.00
40
60
80
age
gender
F
M