POL390 Graphing with ggplot2

Getting Started

First we'll practice using ggplot to create a barplot and a histogram using the same variables we used yesterday in base R. Then in the next portion of the lab you'll work on your own or in small groups to create additional graphs (without my demonstration, but I'll be around to help you).

```
# Load the same data as last time
getwd()
```

```
## [1] "/Users/hyunmyungchoi/study_R/POL390"
```

```
setwd("/Users/hyunmyungchoi/study_R/POL390")
library(gridExtra)

dat <- read.csv("anes_timeseries_2020_csv_20220210.csv")

# head(dat)
dim(dat)
```

```
## [1] 8280 1771
```

```
# str(dat)
str(dat)
```

```
## 'data.frame':    8280 obs. of  1771 variables:
##  $ version        : chr  "ANES2020TimeSeries_20220210" "ANES2020TimeSeries_20220210" "ANES2020TimeSe
##  $ V200001        : int  200015 200022 200039 200046 200053 200060 200084 200091 200107 200114 ...
##  $ V160001_orig   : int  401318 300261 400181 300171 405145 400374 407013 407174 406264 402782 ...
##  $ V200002        : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ V200003        : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ V200004        : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ V200005        : int  0 0 0 0 1 0 0 0 0 1 ...
##  $ V200006        : int  -2 4 -2 -2 -2 -2 -2 -2 -2 4 ...
##  $ V200007        : int  -2 -1 -2 -2 -2 -2 -2 -2 -2 -1 ...
##  $ V200008        : int  -2 3 -2 -2 -2 -2 -2 -2 -2 3 ...
##  $ V200009        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ V200010a       : num  0.963 1.069 0.683 0.501 1.262 ...
##  $ V200010b       : num  1.006 1.163 0.769 0.521 0.966 ...
##  $ V200010c       : int  2 2 1 2 1 2 1 2 2 2 ...
##  $ V200010d       : int  9 26 41 29 23 37 7 37 32 41 ...
##  $ V200011a       : num  0.689 1.089 0.689 0.732 1.014 ...
##  $ V200011b       : num  0.881 1.072 0.609 0.742 0.943 ...
##  $ V200011c       : int  2 2 1 2 1 2 1 2 2 2 ...
```

```
##  $ V200011d       : int  9 26 41 29 23 37 7 37 32 41 ...
##  $ V200012a       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200012b       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200012c       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200012d       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200013a       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200013b       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200013c       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200013d       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200014a       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200014b       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200014c       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200014d       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200015a       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200015b       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200015c       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200015d       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ V200016a       : num  0.997 1.059 0.681 0.474 1.269 ...
##  $ V200016b       : num  0.974 1.149 0.756 0.495 0.935 ...
##  $ V200016c       : int  2 2 1 2 1 2 1 2 2 2 ...
##  $ V200016d       : int  9 26 41 29 23 37 7 37 32 41 ...
##  $ V201001        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ V201002a       : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201002b       : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201003        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201004        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201005        : int  2 4 1 2 2 1 4 4 3 2 ...
##  $ V201006        : int  2 3 2 3 2 1 2 3 2 2 ...
##  $ V201007a       : int  1 1 2 2 2 1 2 1 1 1 ...
##  $ V201007b       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ V201007c       : int  1 1 1 1 1 1 0 1 1 1 ...
##  $ V201008        : int  1 1 1 1 1 1 2 3 1 1 ...
##  $ V201009        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201010        : int  -1 -1 -1 -1 -1 -1 1 -1 -1 -1 ...
##  $ V201011        : int  -1 -1 -1 -1 -1 -1 1 -1 -1 -1 ...
##  $ V201012        : int  -1 -1 -1 -1 -1 -1 1 -1 -1 -1 ...
##  $ V201013a       : int  -1 -1 -1 -1 -1 -1 55 -1 -1 -1 ...
##  $ V201013b       : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201014a       : int  1 1 1 1 1 1 1 -1 1 1 ...
##  $ V201014b       : int  40 16 51 6 8 48 55 -1 4 6 ...
##  $ V201014c       : int  1 1 1 0 1 1 0 -2 1 0 ...
##  $ V201014d       : int  0 0 0 0 0 0 0 -2 0 0 ...
##  $ V201014e       : int  1 1 0 1 1 0 1 -2 1 1 ...
##  $ V201015        : int  -1 -1 -1 -1 -1 -1 1 -1 -1 -1 ...
##  $ V201015z       : int  -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 ...
##  $ V201016        : int  3 3 3 3 2 2 2 -1 3 3 ...
##  $ V201017        : int  -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 ...
##  $ V201018        : int  2 4 -1 2 4 -1 -1 -1 4 4 ...
##  $ V201018z       : int  -3 -3 -3 -3 -3 -3 -3 -3 -3 -3 ...
##  $ V201019        : int  -1 -1 -1 -1 -1 -1 -1 1 -1 -1 ...
##  $ V201020        : int  1 1 1 2 1 2 2 2 2 2 ...
##  $ V201021        : int  8 10 6 -1 8 -1 -1 -1 -1 -1 ...
##  $ V201022        : int  2 2 2 2 2 2 2 2 1 2 ...
##  $ V201023        : int  -1 -1 -1 -1 -1 -1 -1 -1 1 -1 ...
```

```
##  $ V201024        : int  -1 -1 -1 -1 -1 -1 -1 -1 2 -1 ...
##  $ V201025x       : int  3 3 3 3 3 3 3 2 4 3 ...
##  $ V201026        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201028        : int  -1 -1 -1 -1 -1 -1 -1 -1 1 -1 ...
##  $ V201029        : int  -1 -1 -1 -1 -1 -1 -1 -1 1 -1 ...
##  $ V201029z       : int  -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...
##  $ V201030        : int  -1 -1 -1 -1 -1 -1 -1 -1 1 -1 ...
##  $ V201031        : int  -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...
##  $ V201031y       : int  -1 -1 -1 -1 -1 -1 -1 -1 6 -1 ...
##  $ V201032        : int  1 1 1 1 1 1 1 1 -1 1 ...
##  $ V201033        : int  2 3 1 1 2 1 2 2 -1 11 ...
##  $ V201033z       : int  -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...
##  $ V201034        : int  1 1 1 1 1 1 1 1 -1 1 ...
##  $ V201035        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201036        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201036z       : int  -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...
##  $ V201037        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201038        : int  -1 -1 -1 -1 -1 -1 -1 -1 2 -1 ...
##  $ V201039        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201039z       : int  -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...
##  $ V201040        : int  -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...
##  $ V201040y       : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ V201041        : int  1 1 1 1 1 1 1 2 -1 1 ...
##  $ V201042        : int  2 2 1 2 2 6 -1 -1 -1 2 ...
##  $ V201042z       : int  -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...
##  $ V201043        : int  -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 ...
##  $ V201043y       : int  -1 -1 -1 -1 -1 -1 7 -1 -1 -1 ...
##    [list output truncated]
```

Barplot in ggplot 2

First I'll demonstrate the basics (and some bells & whistles) of creating a barplot in ggplot2.

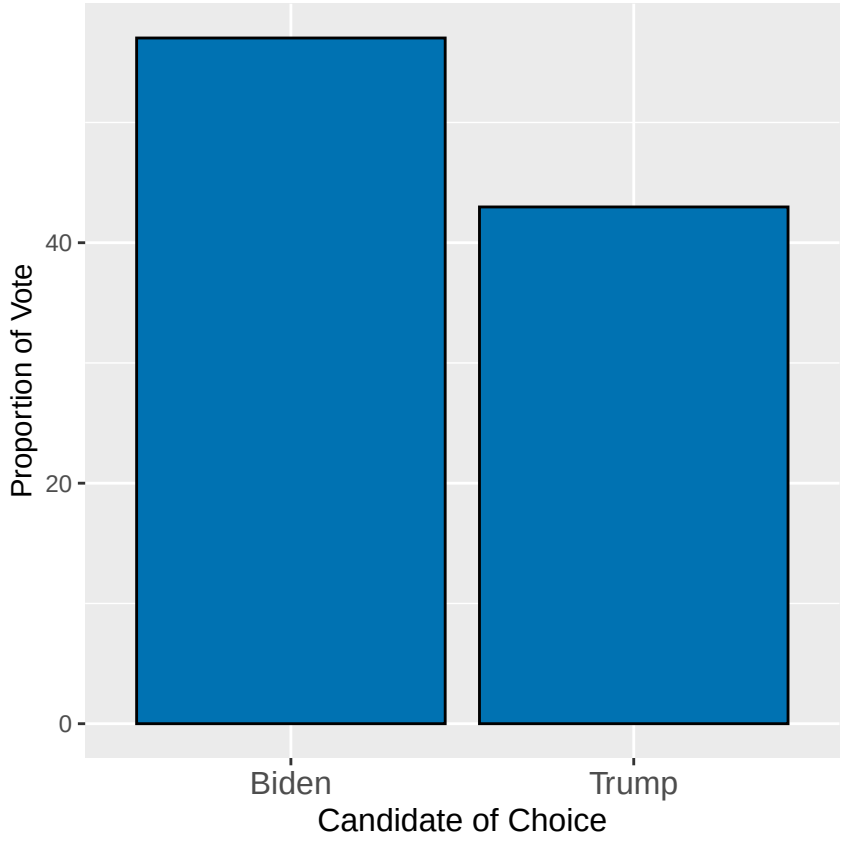
```
# Look at the data and recode
table(dat$V202073)
```

```
##
##   -9   -7   -6   -1    1    2    3    4    5    7    8   11   12
##   53   77  754 1497 3267 2462   69   23   56    1    3    2   16
```

```
# Recode V202073 POST: FOR WHOM DID R VOTE FOR PRESIDENT
dat$vote_trump <- dat$V202073
dat$vote_trump[dat$V202073 < 0 | dat$V202073 > 2] <- NA
dat$vote_trump[dat$V202073 == 1] <- 0
dat$vote_trump[dat$V202073 == 2] <- 1

dat$vote_trump <- as.factor(dat$vote_trump)
table(dat$vote_trump)
```

```
##
##    0    1
## 3267 2462
```

```r
# install.packages("ggplot2") # install ggplot2 (only need to do once!)
library(ggplot2) # call it from our library (must do every time)
## it is asking what kind of graph I want to make
# always get from ggplot
##dat = subset -> na delete
## aes -> x,y axis set
ggplot(dat = subset(dat, !is.na(vote_trump)),
       aes(x = vote_trump, y = (after_stat(count) / sum(after_stat(count))* 100))) +    #gives proporti
  geom_bar(fill = "#0072B2", color = "black") +                    ## bar graph set
  labs(x = "Candidate of Choice", y = "Proportion of Vote") +    ## x axis y axis name set
  scale_x_discrete(labels = c("0" = "Biden", "1" = "Trump"))  +   ## change label name
  theme(aspect.ratio = 1, # how ratio height is
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 12))
```



Now it's your turn! Re-create the barplot you created yesterday but use ggplot2 this time. Play around with the functionality of ggplot2 to make to barplot prettier.

```r
# Student code here

# Code here
table(dat$V202557)
```

```
##
##   -9   -7   -6   -5    1    2
##   67   77  754  102 3374 3906
```

```r
str(dat$V202557)
```

```
##  int [1:8280] 1 1 2 1 1 2 1 2 2 2 ...
```

```r
dat$isLab <- dat$V202557
dat$isLab[dat$V202557 < 1] <- NA
dat$isLab[dat$V202557 == 2] <- 0
dat$isLab[dat$V202557 == 1] <- 1
str(dat$isLab)
```

```
##  num [1:8280] 1 1 0 1 1 0 1 0 0 0 ...
```

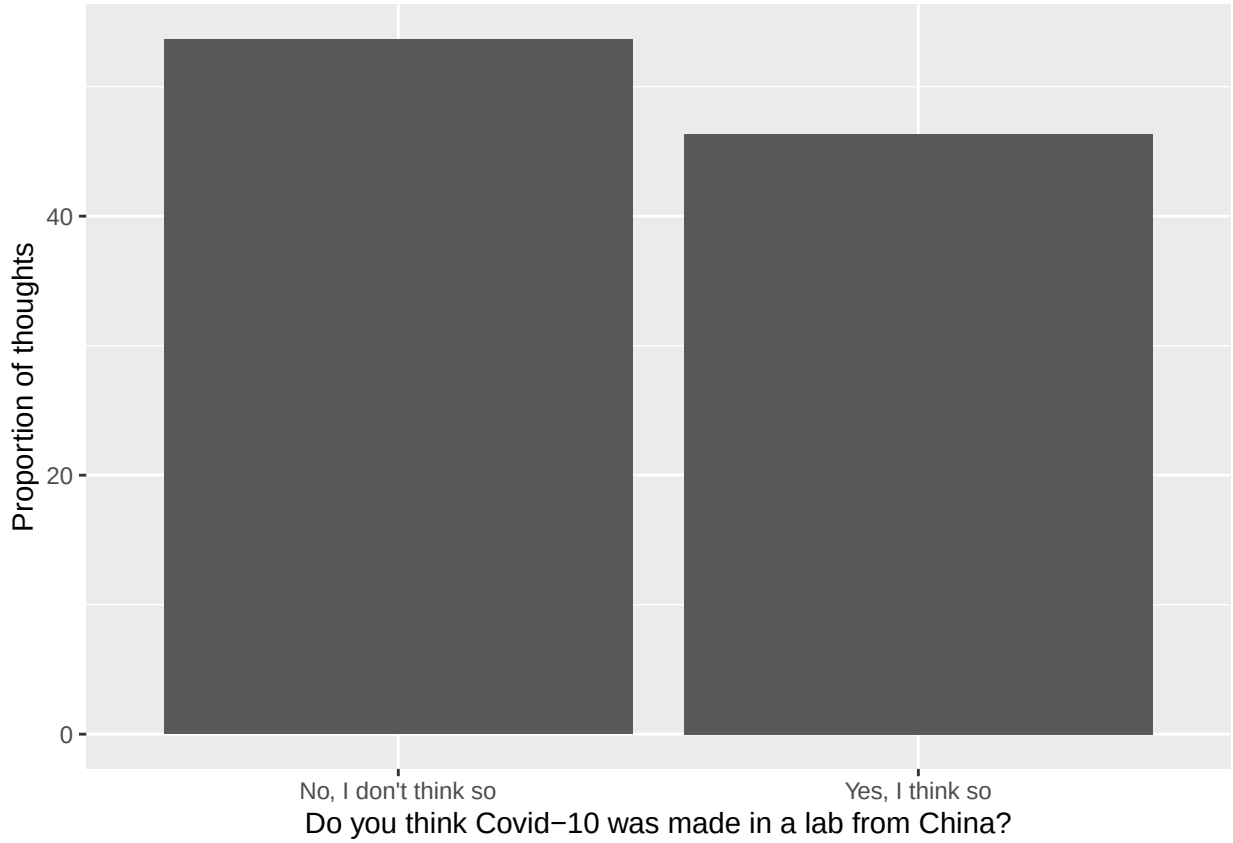
```r
summary(dat$isLab)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.0000  0.0000  0.4635  1.0000  1.0000    1000
```

```r
dat$isLab <- factor(dat$isLab, labels = c("isn't Lab", "is Lab"), levels = c(0,1))
## 46% is proportion of "is lab"
table(dat$isLab)
```

```
##
## isn't Lab    is Lab
##      3906      3374
```

```r
ggplot(dat = subset(dat, !is.na(isLab)),
       aes(x = isLab, y = (after_stat(count) / sum(after_stat(count)) * 100))) +
  geom_bar() +
  labs(x = "Do you think Covid-10 was made in a lab from China?", y = "Proportion of thoughts") +
  scale_x_discrete(labels = c("isn't Lab" = "No, I don't think so", "is Lab" = "Yes, I think so"))
```



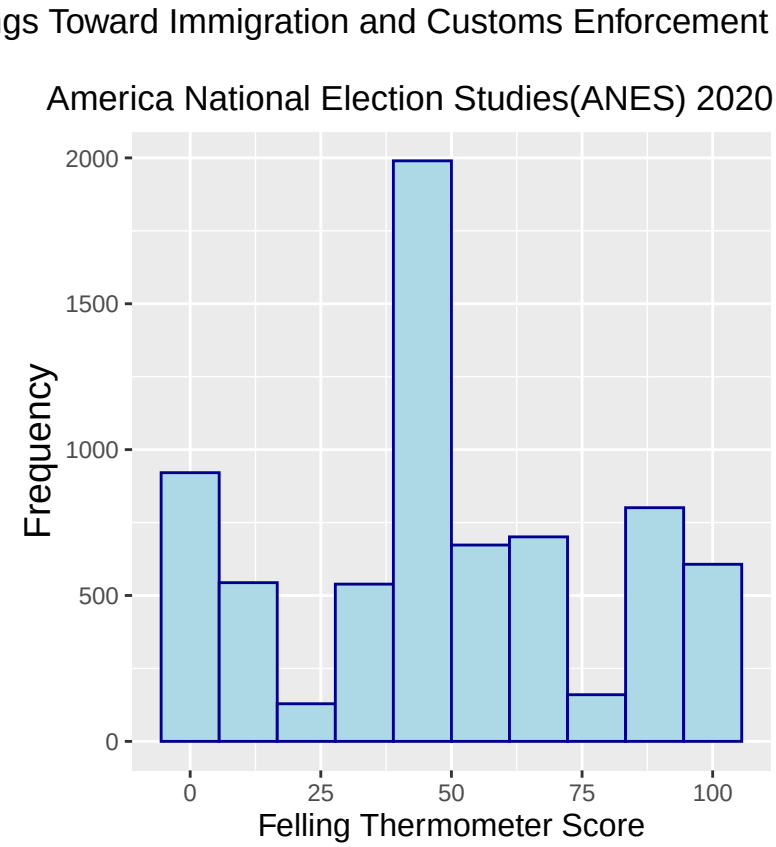
Histogram in ggplot2

First I'll demonstrate the basics of creating a histogram in ggplot2.

```
# Look at the data and recode
# V202182 POST: FEELING THERMOMETER: IMMIGRATION AND CUSTOMS ENFORCEMENT (ICE) AGENCY
summary(dat$V202182)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -9.00   15.00   50.00   75.98   70.00  999.00
```

```
dat$feel_ice <- dat$V202182
dat$feel_ice[dat$V202182 < 0 | dat$V202182 > 100] <- NA

ggplot(dat = subset(dat, !is.na(feel_ice)), aes(x = feel_ice)) +
  geom_histogram(bins = 10, color = "darkblue", fill="lightblue") +
  labs(x = "Felling Thermometer Score",
       y = "Frequency",
       title = "Feelings Toward Immigration and Customs Enforcement Agency(ICE),
       \nAmerica National Election Studies(ANES) 2020 Data") +
  theme(aspect.ratio = 1,
       axis.title.y = element_text(size = 14),
       axis.title.x = element_text(size = 12),
       plot.title = element_text(hjust = 0.5))
```



Now it's your turn! Re-create the histogram you created yesterday but use ggplot2 this time.

```
# Your code here
str(dat$V202158)
```

```
##  int [1:8280] 50 50 100 100 0 85 0 -9 30 50 ...
```

```
table(dat$V202158)
```

```
##
##   -9   -7   -6   -5   -4    0    1    2    3    4    5    6    7    8    9   10
##  126   77  754    9    1  424    5    6    2    3   18    3    2    1    4   36
##   12   15   16   20   23   25   30   35   39   40   45   48   49   50   51   52
##    1  236    1   25    1   25  292   12    1  316   16    1    6 1095    1    1
##   53   55   59   60   65   69   70   72   75   77   78   79   80   83   84   85
##    1   14    1  488   32    1  629    1   85    1    1    1  109    1    1 1060
##   86   87   88   89   90   92   95   96   97   98   99  100  998  999
##    4    1    7    4  222    1  114    2    2   12   15 1949    5   15
```

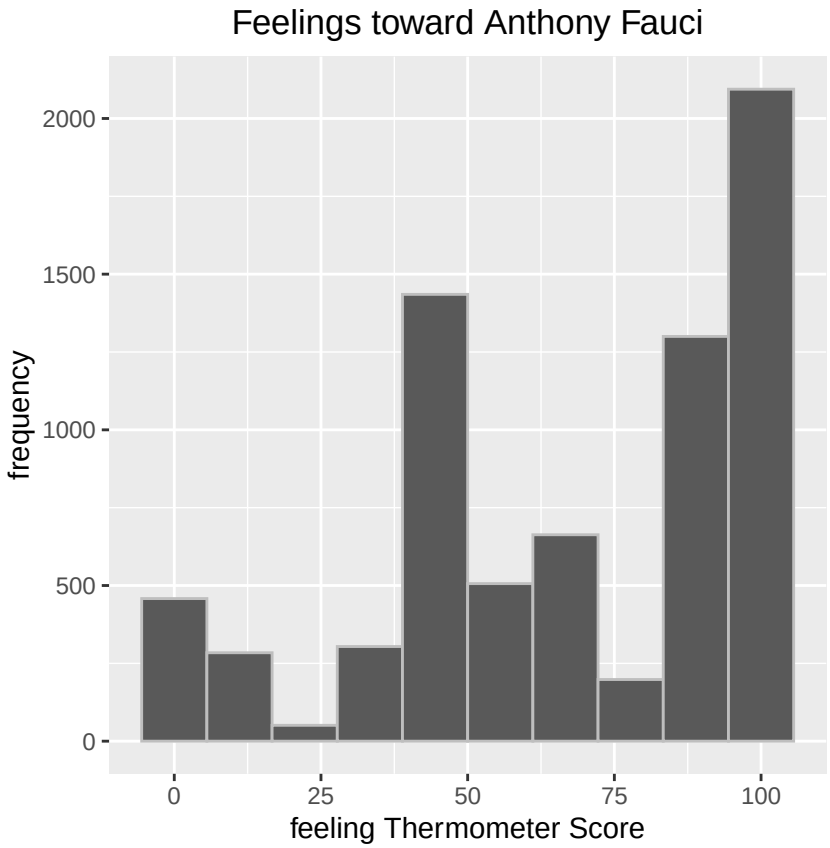
```
summary(dat$V202158)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -9.00   40.00   70.00   61.48   95.00  999.00
```

```
dat$feelsIce <- dat$V202158
dat$feelsIce[dat$V202158 < 0 | dat$V202158 > 100] <- NA
summary(dat$feelsIce)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   50.00   70.00   67.92  100.00  100.00     987
```

```
ggplot(dat = subset(dat, !is.na(feelsIce)), aes(x = feelsIce)) +
  geom_histogram(bins = 10, color = "grey") +
  labs(x = "feeling Thermometer Score",
       y = "frequency",
       title = "Feelings toward Anthony Fauci") +
  theme(aspect.ratio = 1,
        plot.title = element_text(hjust = 0.5))
```



Extending Your Knowledge

Now that you have the basics, extend your knowledge by creating graphs using ggplot2 on your own or in small groups. Use the textbook or the internet (google, Stackoverflow, ChatGPT) to help you figure out the code. I'll be here if you get stuck.

Two factor variables

Create a graph comparing two factor variables. Probably the easiest way to do this is to create a stacked barchart. Just like before, you'll have the factor variables along the x-axis and the y-axis with show the percent or number in each category. Except this time you'll have two factor variables along the x-axis, either stacked on top of each other or (better yet) next to one another. In the text below the R chunk, interpret the graphs

```
# Two factor variables: Stacked barchart  = put two barchart to one

# `V201393`: Were the limits placed on activities due to covid-19 too strict or not? Recode this variab
# `V202310`: How important should science be for decisions about covid? Recode this variable so that 1 =
str(dat$V201393)
```

```
##  int [1:8280] 1 3 3 4 1 5 3 3 3 4 ...
```

```
str(dat$V202310)
```

```
##  int [1:8280] 5 4 5 5 2 5 4 5 4 5 ...
```

```
dat$limitStrict <- dat$V201393
dat$scienceMatter <- dat$V202310

dat$limitStrict[dat$V201393 < 1] <- NA
dat$limitStrict[dat$V201393 == 1] <- 1
dat$limitStrict[dat$V201393 == 2] <- 2
dat$limitStrict[dat$V201393 == 3] <- 3
dat$limitStrict[dat$V201393 == 4] <- 4
dat$limitStrict[dat$V201393 == 5] <- 5
#dat$limitStrict <- dat$limitStrict[!is.na(dat$V201393)]
dat$scienceMatter[dat$V202310 < 1] <- NA
dat$limitStrict[dat$V202310 == 1] <- 1

summary(dat$limitStrict)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   3.000   3.000   3.169   4.000   5.000      24
```

```
summary(dat$scienceMatter)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   4.000   5.000   4.222   5.000   5.000     901
```

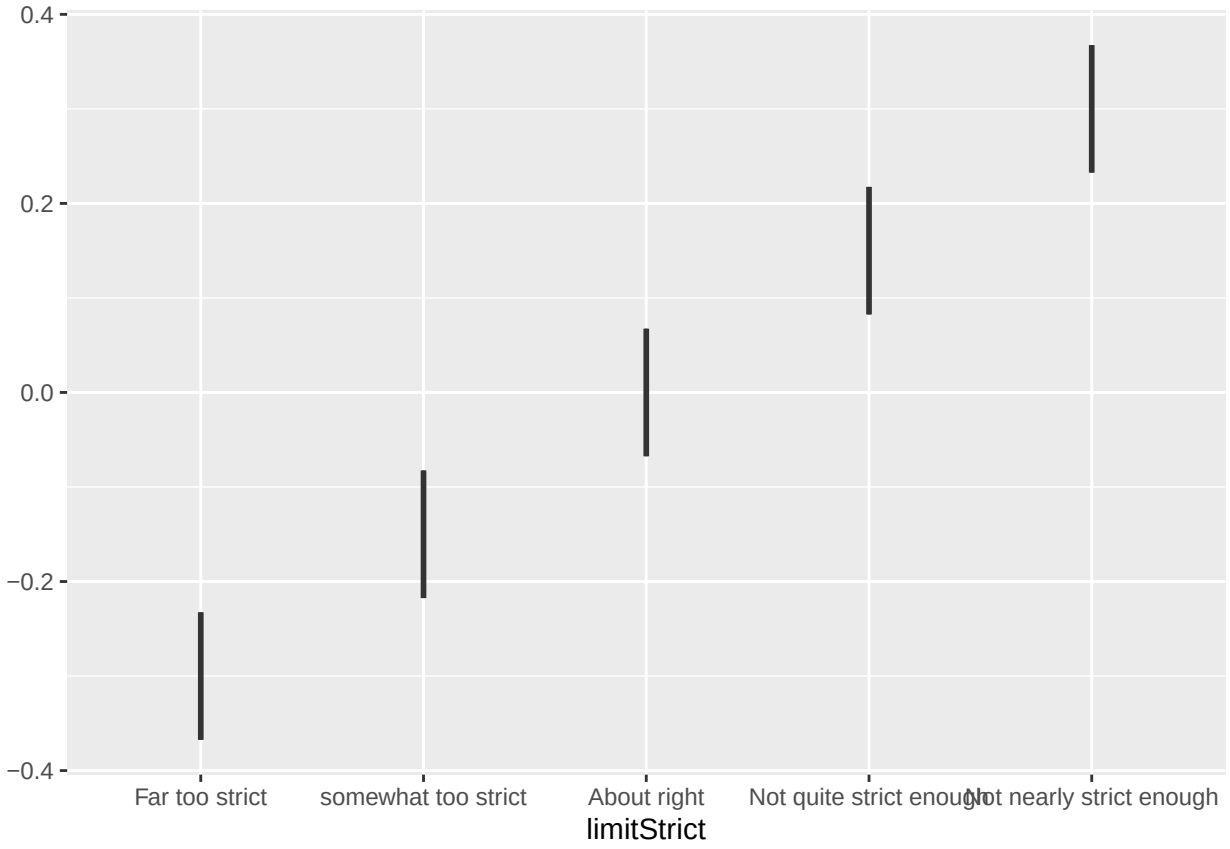
```
#groups <- data.frame(group=rep)

#ggplot(dat = (subset(dat, !is.na(limitStrict))), aes(x=limitStrict, y=scienceMatter, fill=group)) +
  #geom_bar(stat="identity")
```

[Write your interpretation here.] I am very sorry but I really have no idea about it. I searched google, chatgpt, anything and they all say different things. I am sorry. ## A numeric (or integer) variable and a factor variable Show the relationship between one numeric (or integer) variable and one factor variable. Create two different graphs to show this relationship (you can use the same factor/ numeric variables for both or use different variables). First, create a boxplot.

```r
# Boxplot here
dat$limitStrict <- dat$V201393
dat$limitStrict[dat$V201393 < 1] <- NA
dat$limitStrict[dat$V201393 == 1] <- 1
dat$limitStrict[dat$V201393 == 2] <- 2
dat$limitStrict[dat$V201393 == 3] <- 3
dat$limitStrict[dat$V201393 == 4] <- 4
dat$limitStrict[dat$V201393 == 5] <- 5
dat$limitStrict <- factor(dat$limitStrict,
                    labels = c("Far too strict", "somewhat too strict", "About right", "Not quite stric
                    levels = c(1,2,3,4,5))
summary(dat$limitStrict)
```

```
##           Far too strict      somewhat too strict             About right
##                      788                     1022                    3511
##  Not quite strict enough Not nearly strict enough                    NA's
##                     1756                     1178                      25
```

```r
ggplot(dat=(subset(dat,!is.na(limitStrict))), aes(x=limitStrict)) +
  geom_boxplot()
```



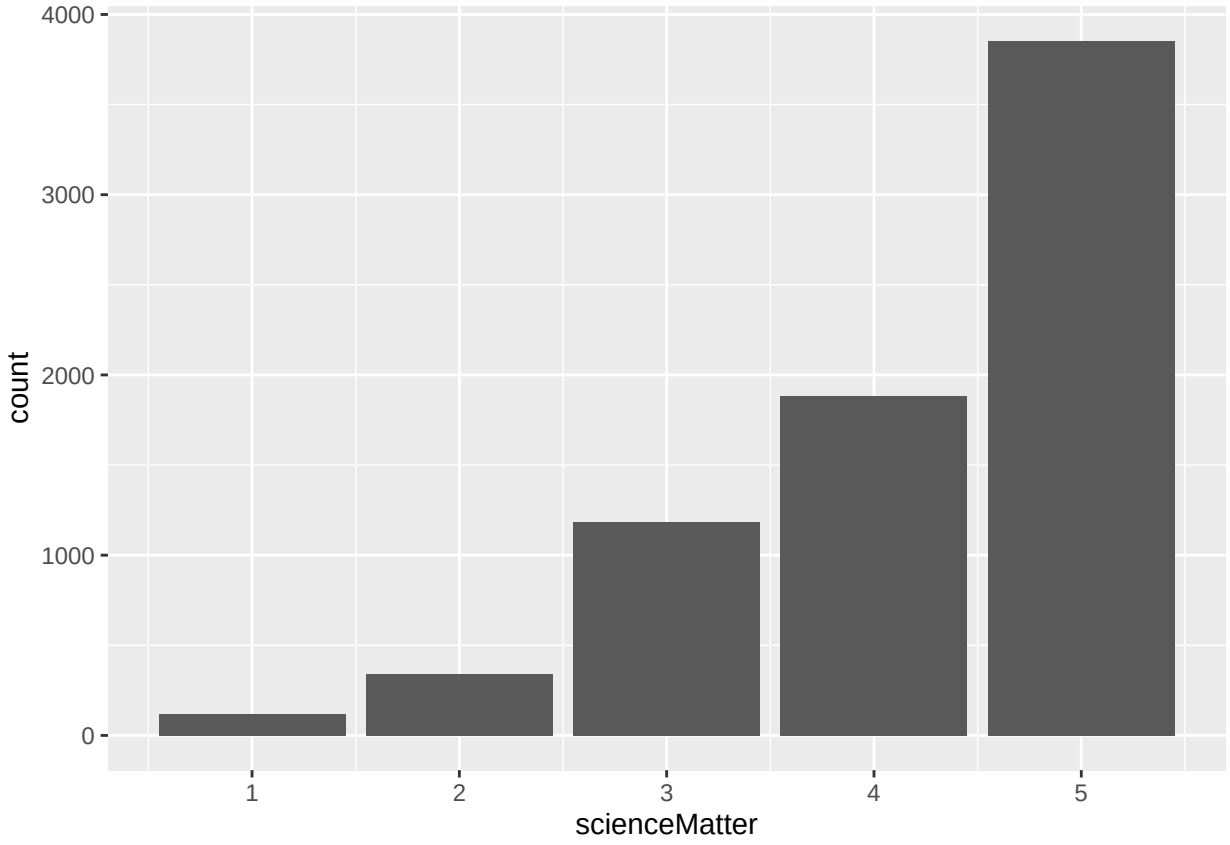
[Boxplot interpretation here]

Second, create a barchart with the factor variable along the x-axis and the numeric (or integer) variable along the y-axis. Your interpretation of this barchart will be a little different than the one you created before. Alternatively, you can create a dotplot with the numeric (or integer) variable along the x-axis and the factor variable along the y-axis.

```
# Barchart or dotplot here
str(dat$V202310)
```

```
##  int [1:8280] 5 4 5 5 2 5 4 5 4 5 ...
```

```
dat$scienceMatter <- dat$V202310
dat$scienceMatter[dat$V202310 < 1] <- NA

ggplot(dat = subset(dat, !is.na(dat$scienceMatter)), aes(x=scienceMatter)) + geom_bar()
```



[Barchart2 or dotplot interpretation here]

Two numeric (or integer) variables

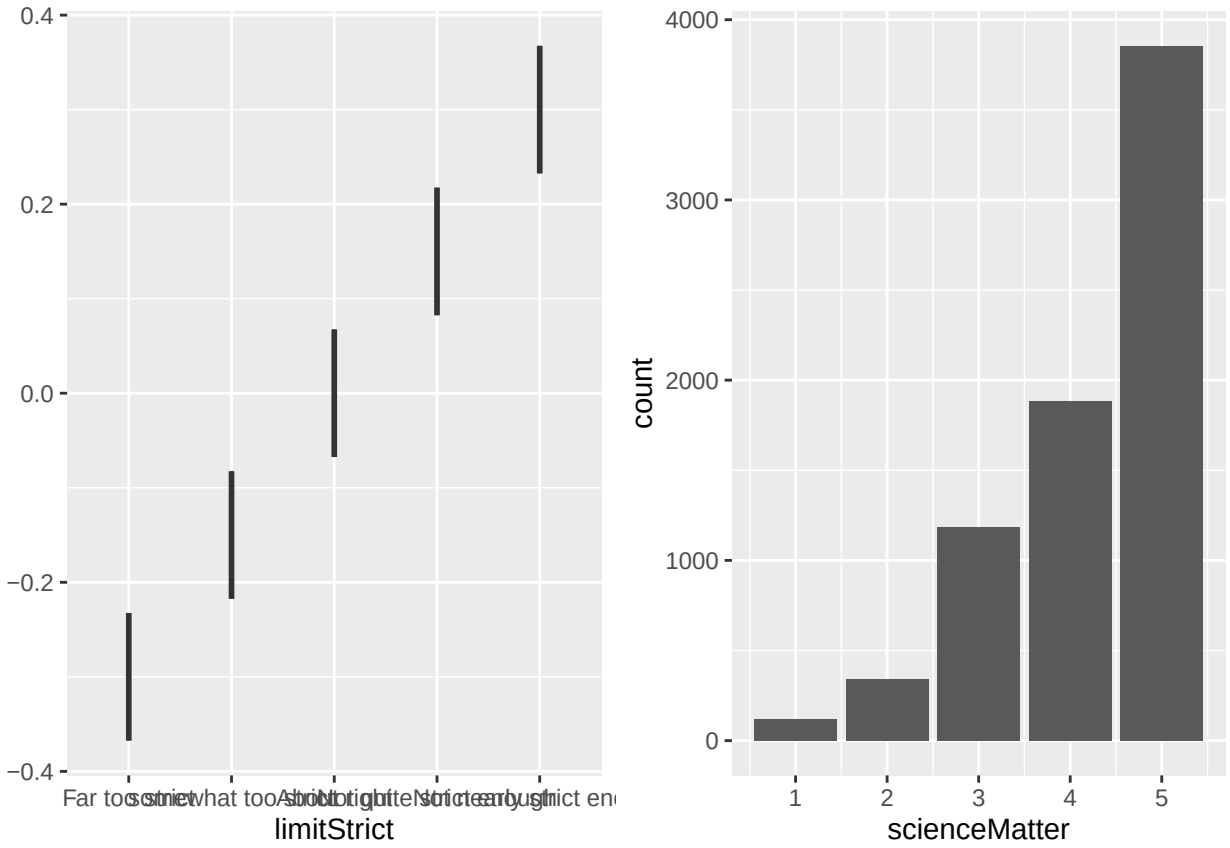
Now use ggplot2 to create a graph showing the relationship between numeric (or integer) variables. Probably the best time of graph to use is a scatterplot and/or a fitted regression line.

```
# Code here
dat$limitStrict <- dat$V201393
dat$limitStrict[dat$V201393 < 1] <- NA
dat$limitStrict[dat$V201393 == 1] <- 1
dat$limitStrict[dat$V201393 == 2] <- 2
dat$limitStrict[dat$V201393 == 3] <- 3
dat$limitStrict[dat$V201393 == 4] <- 4
```

```
dat$limitStrict[dat$V201393 == 5] <- 5
dat$limitStrict <- factor(dat$limitStrict,
                    labels = c("Far too strict", "somewhat too strict", "About right", "Not quite strict
                    levels = c(1,2,3,4,5))


graphL <- ggplot(dat=(subset(dat,!is.na(dat$limitStrict))), aes(x=limitStrict)) +
  geom_boxplot()

str(dat$V202310)
```

```
##  int [1:8280] 5 4 5 5 2 5 4 5 4 5 ...
```

```
dat$scienceMatter <- dat$V202310
dat$scienceMatter[dat$V202310 < 1] <- NA

graphS <- ggplot(dat = subset(dat, !is.na(dat$scienceMatter)), aes(x=scienceMatter)) + geom_bar()


grid.arrange(graphL, graphS, ncol=2)
```



[Interpretation here]