Programming Assignment 1 Solutions

Soc Methods Camp

September 5th, 2018

Step zero: loading and examining the data

Tasks:

- Load the anespilot16.csv data, view its dimensions, check what type of object it is, and view the first few rows and first 10 variables
- Install (if not already) and load the dplyr and ggplot2 packages

```
##load ANES data
anes <- read.csv("anespilot16.csv")

##view dimensions
dim(anes)
```

```
## [1] 1200  594
```

```
##check what type of object anes is
class(anes)
```

```
## [1] "data.frame"
```

```
##view first 6 rows and first 10 variables
head(anes[, 1:10])
```

```
##                                 version caseid    weight weight_spss
## 1 ANES 2016 Pilot Study version 20160223      1 0.9511600   0.5421612
## 2 ANES 2016 Pilot Study version 20160223      2 2.6701962   1.5220118
## 3 ANES 2016 Pilot Study version 20160223      3 1.4303896   0.8153221
## 4 ANES 2016 Pilot Study version 20160223      4 0.9139662   0.5209607
## 5 ANES 2016 Pilot Study version 20160223      5 0.2639346   0.1504427
## 6 ANES 2016 Pilot Study version 20160223      6 1.4631177   0.8339771
##   follow turnout12 turnout12b vote12 percent16 meet
## 1      1         1          9      2       100    1
## 2      2         2          9      9        50    4
## 3      1         1          9      1       100    1
## 4      1         1          9      2       100    5
## 5      1         1          9      1       100    2
## 6      1         1          9      3       100    3
```

```
##install and/or load the dplyr package
library(dplyr)
library(ggplot2)
```

Step one: creating variables of interest to review different data types

Task one: Create a vector, *varsinclude*, that contains the following variables of interest for our analysis:

Feelings about free trade:

- freetrade: views about free trade

Rankings of which issues are most important:

- ISSUES_OC14_10: where the respondent ranks unemployment as an important issue (out of 21 issues)

Feelings about candidates:

- ftsanders: feeling thermometer for Sanders
- fttrump: feeling thermometer for Trump
- fthrc: feeling thermometer for Hillary Clinton

Demographic variables:

- gender
- race
- educ
- birthyr

Question one: What type of vector should *varsinclude* be? Confirm your answer by using the "class" command

Answer: character vector

```
varsinclude <- c("freetrade",
                 "ISSUES_OC14_10",
                 "ftsanders",
                 "fttrump",
                 "fthrc",
                 "gender",
                 "race",
                 "educ",
                 "birthyr")
class(varsinclude)
```

```
## [1] "character"
```

Task two: Use either the appropriate dplyr command or base R to create a new data.frame, anes2, that only includes those relevant variables. *Hint: if using dplyr, the command one_of may be useful*

Then, check the dimensions (should be 1200 x 9)

For this step, one approach was to use cbind.data.frame with each of the columns. That definitely works, but the advantages of creating a character vector like the one above is that we can then use that character vector to tell R which columns to take (in base R is most straightforward). Below in the code, I wrote out what steps we're changing by saving the column names into a vector

```
##dplyr
anes2 <- select(anes, one_of(varsinclude))

##base R
anes2 <- anes[, varsinclude]

#equivalent to the above expression, but we
#don't need to do because we already created the varsinclude
#vector
testanes2<- anes[,  c("freetrade",
                 "ISSUES_OC14_10",
                 "ftsanders",
```

```
                "fttrump",
                "fthrc",
                "gender",
                "race",
                "educ",
                "birthyr")]
head(testanes2)
```

```
##   freetrade ISSUES_OC14_10 ftsanders fttrump fthrc gender race educ
## 1         4              8        84       1    76      1    1    5
## 2         4              8        13      28    52      2    1    6
## 3         4              8         2     100     1      1    1    5
## 4         5              8        71       0    69      1    1    5
## 5         4              8        13      13     1      1    1    2
## 6         1              2        11      61     1      1    1    6
##   birthyr
## 1    1960
## 2    1957
## 3    1963
## 4    1980
## 5    1974
## 6    1958
```

```
##check dimensions
dim(anes2)
```

```
## [1] 1200    9
```

Now, we're going to work on getting the variables into more usable form.

Task three: Using the codebook for the study and the appropriate R command, create a new factor variable-racenew- that collapses the racial groups into the following categories and labels them appropriately:

- 1 = white
- 2 = black
- 3 = hispanic
- 4 = other (includes all non-hispanic categories)

Save the results into a new data.frame, *anes3*

We could do this in a for loop but a short trick is to recognize that race in the anes codebook is 1 = white, 2 = black, 3 = hispanic, 4, 5, 6, 7, 8 are all other categories. So we can use subsetting to classify anyone in category four or above as belong in a fourth category, and then use the factor command to label the categories appropriately

```
##recode so that the first 3 categories keep
##their same value but 4 up are collapsed into other
table(anes2$race)
```

```
##
##   1   2   3   4   5   6   7   8
## 875 135 113  23   7  30  15   2
```

```
anes3 <- anes
anes3$racenew <- anes2$race
anes3$racenew[anes2$race >= 4] <- 4
table(anes3$racenew)
```

```
##
```

```
##   1   2   3   4
## 875 135 113  77
```

```
# dpylr solution
library(magrittr)
library(dplyr)
anes3.TD <- anes2 %>%
  mutate(racenew = ifelse(race >= 4, 4, anes2$race))
table(anes3.TD$racenew)
```

```
##
##   1   2   3   4
## 875 135 113  77
```

```
##see that it's numeric- change into factor type
anes3$racenew <- factor(anes3$racenew,
                        levels = c(1, 2, 3, 4),
                        labels = c("white", "black",
                                   "hispanic", "other"))
```

Now we're going to clean some of the other variables.

Task four: Use the codebook and the appropriate R command in either base R or dplyr, create labels for the levels of the following variables. When creating these labels, it's up to you whether you want to create new variables or just save over the existing names

- gender
- educ
- freetrade

Save the results in *anes3*

```
#dplyr
anes3 <- mutate(anes2,
                gender = factor(gender,
                                levels = c(1, 2),
                                labels= c("male",
                                          "female")),
                educ = factor(educ,
                              levels = c(1, 2, 3, 4, 5, 6),
                              labels = c("nohs",
                                         "hsgrad",
                                         "somecollege",
                                         "2year",
                                         "4year",
                                         "postgrad")),
                freetrade = factor(freetrade,
                              levels = c(1, 2, 3, 4, 5, 6, 7),
                              labels = c("favorgreat",
                                         "favormod",
                                         "favorlittle",
                                         "neither",
                                         "opposelittle",
                                         "opposemod",
                                         "opposegreat")))

#note: one group had the good example of instead of writting out the
##full levels vector for education and gender, can do: levels = 1:7
```

Step two: logical statements and filtering

Task one: To make sure the continuous variables are usable, use the "summary" command to view the range of the following heat thermometer variables we will be examining (make sure to use indexing so it only summarizes those three variables):

- fttrump
- ftsanders
- fthrc

The feeling thermometers should go from 0-100, but what is the range of these three variables?

```
summary(anes3[, c("fttrump",
                  "ftsanders",
                  "fthrc")])
```

```
##     fttrump         ftsanders          fthrc
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  2.00   1st Qu.: 19.00   1st Qu.:  3.00
##  Median : 30.00   Median : 51.00   Median : 44.00
##  Mean   : 40.78   Mean   : 56.73   Mean   : 43.79
##  3rd Qu.: 72.00   3rd Qu.: 82.00   3rd Qu.: 76.00
##  Max.   :998.00   Max.   :998.00   Max.   :998.00
```

Task two: Use the codebook to:

1. For each of the variables, recode values above 100 to NA
2. Exclude from the data observations that have NA for at least one of the heat thermometer variables - there should be 1188 observations remaining

```
##set to NA
anes3$fttrump[anes3$fttrump > 100] <- NA
anes3$ftsanders[anes3$ftsanders > 100] <- NA
anes3$fthrc[anes3$fthrc > 100] <- NA

# dplyr solutions
anes3.TD2 <- anes3.TD %>%
  mutate(fttrump2 = replace(fttrump, fttrump > 100, NA))

anes3.TD2$fttrump.Replaced <- replace(anes3.TD$fttrump, anes3.TD$fttrump > 100, NA)

##omit from data
anes3 <- anes3[!is.na(anes3$fttrump) &
                   !is.na(anes3$ftsanders) &
                 !is.na(anes3$fthrc), ]
nrow(anes3)
```

```
## [1] 1188
```

```
##note that using or with is.na doesn't
##exclude anyone
test <- anes3[!is.na(anes3$fttrump) |
                   !is.na(anes3$ftsanders) |
                 !is.na(anes3$fthrc), ]
```

```
nrow(test)
```

```
## [1] 1188
```

You want to see if there's a meaningful enough subset of respondents who oppose free trade to look at whether they're more likely to support Trump or Sanders than Clinton.

Task three: Using either dplyr or base R, find and print the number and percentage of respondents who *either* oppose free trade a little, a moderate amount, or a lot

```
##base R
##number
sum(table(anes3$freetrade)[c("opposelittle",
                             "opposemod",
                             "opposegreat")])
```

```
## [1] 322
```

```
##percentage
sum(table(anes3$freetrade)[c("opposelittle",
                             "opposemod",
                             "opposegreat")])/nrow(anes3)
```

```
## [1] 0.2710438
```

```
##dplyr
anes3 %>%
  filter(freetrade == "opposelittle" |
        freetrade == "opposemod" |
        freetrade == "opposegreat")%>%
  summarise(noppose = n(),
            percentoppose = noppose/nrow(anes3))
```

```
##   noppose percentoppose
## 1     322     0.2710438
```

Question one: How many and what % of respondents oppose free trade at least a little bit?

Answer: 322- 27%

Now you want to plot the distribution of feelings about Trump versus Sanders versus Clinton among those who oppose making free trade deals with other countries

Task four: To make this plotting easier by avoiding logical statements inside the plotting function, create a dataframe, *opposers*, composed of just those who oppose free trade at least a little (i.e. oppose a little, moderate, or greatly)

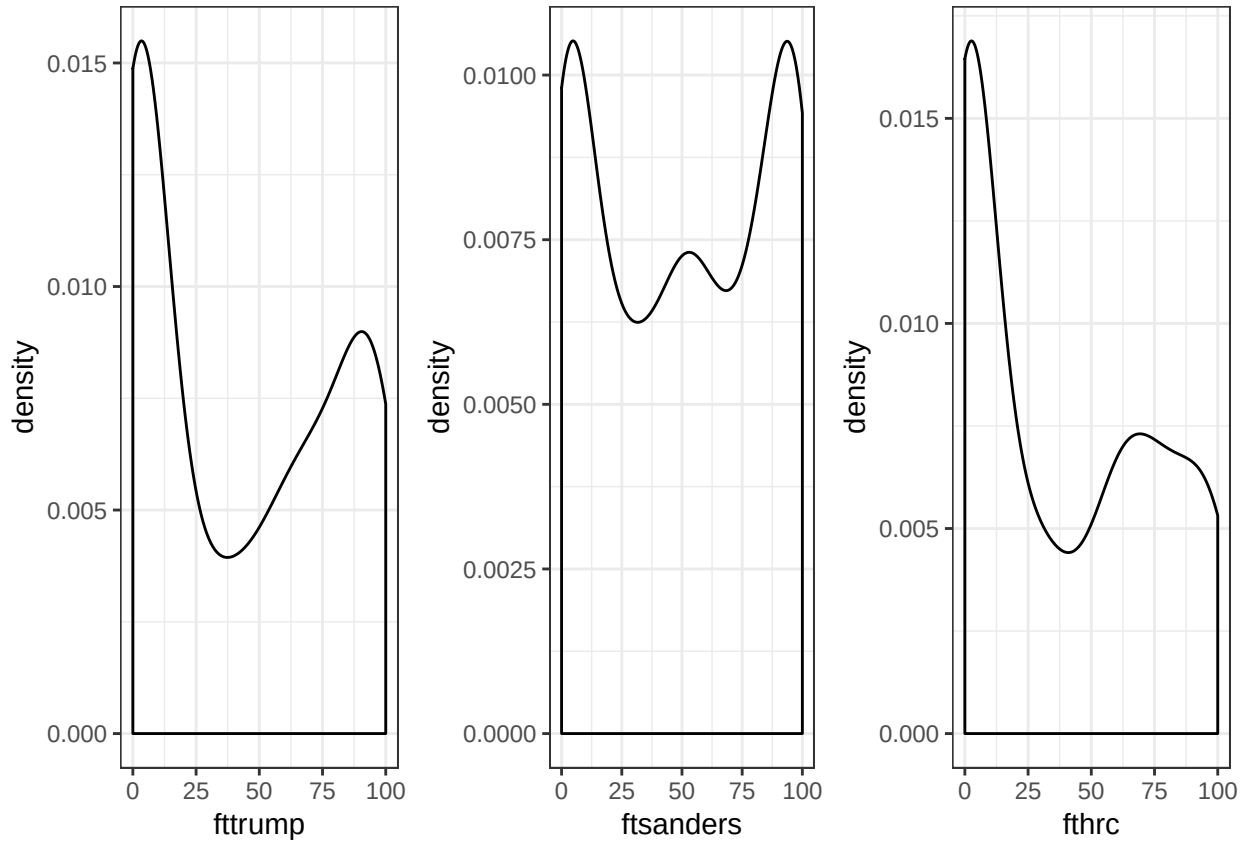
```
##dplyr
opposers <- anes3 %>%
            filter(freetrade == "opposelittle" |
                             freetrade == "opposemod" |
                             freetrade == "opposegreat")

##base R
opposers <- anes3[anes3$freetrade == "opposelittle" |
                  anes3$freetrade == "opposemod" |
                  anes3$freetrade == "opposegreat", ]
```

Task five: Now, using *ggplot2* and *grid.arrange* from the package *gridExtra* (use the help file to find out the syntax), create the following 3 plots side by side (so a 3 column grid). No need to add titles and axis labels

yet, but you can for extra credit:

- Density of feelings about Trump among opposers
- Density of feelings about Sanders among opposers
- Density of feelings about Hillary among opposers

```
library(ggplot2)
library(gridExtra)
trumpfeel <- ggplot(opposers, aes(x = fttrump)) +
  geom_density() +
  theme_bw()
sandersfeel <- ggplot(opposers, aes(x = ftsanders)) +
  geom_density() +
  theme_bw()
hilfeel <- ggplot(opposers, aes(x = fthrc)) +
  geom_density()  +
  theme_bw()
grid.arrange(trumpfeel, sandersfeel, hilfeel, ncol = 3)
```



Keeping in mind that 0 = dislike and 100 = like, what does the graph show?

Answer: Among those opposed to free trade, for Sanders and Trump, there seems to be those who strongly like or dislike; for hillary, seems more skewed towards dislike

We'll learn how to improve upon these graphics during the graphics session of the camp, so for now, we'll leave them as is.

Step three: look at education as a confounder

You notice a trend but are wondering if it stems from an omitted variable- a person's education level, where persons with lower educational attainment might both be more likely to support trump or sanders and also more likely to oppose free trade because of the sectors in which they work

Task one: Going back to the *anes3* data, do the following:

1. Create a numeric version of the freetrade variable called freetradenum- this will allow you to take the mean
2. use dplyr to find the mean views about trade (higher = more opposition) for each educational category, using *na.rm* in the *mean* command to exclude NA's from this calculation (the *mean* command won't run if you have missing values), and then to rank the education groups from most opposed to least opposed.

```
#create a numeric version
anes3$freetradenum <- as.numeric(anes3$freetrade)
summary(anes3$educ)
```

```
##        nohs      hsgrad somecollege       2year       4year    postgrad
##         101         406         255         105         200         121
```

```
##dplyr
anes3 %>%
  group_by(educ) %>%
  summarise(ftmean = mean(freetradenum)) %>%
  arrange((desc(ftmean)))
```

```
## # A tibble: 6 x 2
##   educ        ftmean
##   <fct>        <dbl>
## 1 2year         4.07
## 2 nohs          4.06
## 3 somecollege   4.00
## 4 hsgrad        3.93
## 5 postgrad      3.72
## 6 4year         3.64
```

```
##as an aside base R
tapply(anes3$freetradenum, anes3$educ, mean, na.rm = TRUE)
```

```
##        nohs      hsgrad somecollege       2year       4year    postgrad
##    4.059406    3.928571    4.003922    4.066667    3.640000    3.719008
```

Question one: Which educational groups are most opposed to free trade?

Answer: Those with no high school and those with a 2-year college degree

Task two: Now, use dplyr to find the mean thermometer rating for trump, sanders, and clinton by education group, creating three summary measures using the *summarise* command: *trumpheat*, *sandersheat*, *hilheat*. Display the data so that rows are education categories and the columns are the candidates.

```
mean(anes3$fttrump)
```

```
## [1] 38.29545
```

```
anes3 %>%
  group_by(educ) %>%
  summarise(trumpheat = mean(fttrump),
```

```
            sandersheat = mean(ftsanders),
            hilheat = mean(fthrc))
```

```
## # A tibble: 6 x 4
##   educ        trumpheat sandersheat hilheat
##   <fct>           <dbl>       <dbl>   <dbl>
## 1 nohs             42.6        44.2    38.2
## 2 hsgrad           41.8        46.7    45.1
## 3 somecollege      35.3        56.4    41.2
## 4 2year            37.8        56.8    47.3
## 5 4year            34.8        49.2    41.7
## 6 postgrad         35.7        52.0    42.2
```

Task three: You're also interested in descriptively exploring what role gender might play in these views– find the mean degree of opposition to free trade for each education and gender category (e.g., males with 2 year, females with 2 year, males with some college, etc.) using dplyr

Then, order from most opposed (highest score) to least opposed (lowest score)

```
anes3 %>%
  group_by(educ, gender) %>%
  summarise(freetradeview = mean(freetradenum)) %>%
  arrange(rank(desc(freetradeview)))
```

```
## # A tibble: 12 x 3
## # Groups:   educ [6]
##    educ        gender freetradeview
##    <fct>       <fct>          <dbl>
##  1 2year       male            4.21
##  2 postgrad    female          4.14
##  3 nohs        male            4.12
##  4 somecollege male            4.06
##  5 hsgrad      female          4.02
##  6 nohs        female          4.02
##  7 somecollege female          3.95
##  8 2year       female          3.95
##  9 hsgrad      male            3.83
## 10 4year       female          3.68
## 11 4year       male            3.59
## 12 postgrad    male            3.33
```

What patterns, if any, do you notice?

Answer: difficult to see clear pattern- males might be more opposed in general than females

Task four: You're confused about why postgrad females seem one of the most opposed to free trade– use subsetting in base R or dplyr to find out how many males versus females with a postgrad education were surveyed

```
##dplyr
anes3 %>%
  filter(educ == "postgrad") %>%
  summarise(nfem = sum(gender == "female"),
            nmale = sum(gender == "male"))
```

```
##   nfem nmale
## 1   58    63
```

```
##base R
nrow(anes3[anes3$educ == "postgrad" &
           anes3$gender == "female", ])
```

```
## [1] 58
```

```
nrow(anes3[anes3$educ == "postgrad" &
           anes3$gender == "male", ])
```

```
## [1] 63
```

Step four: for loops for sampling

You're still perplexed by the somewhat unexpected finding that postgrad females seem to oppose free trade so much. Given that the sample is somewhat small, you want to make sure that there is not one observation that is unduly influencing the results. One way to do this is to use a loop to iteratively remove each observation, take the mean of the freetrade variable in the rest of the sample with that observation removed, and then plot those means.

Note: Of course, a simpler way would just be to plot the values of that variable and look for outliers, but the structure of this loop format can be adapted to do more advanced analysis other than taking the mean, like running a regression model and extracting the coefficient of interest. For simple functions like means, it is pretty clear how an outlier input can contribute to a skewed output, but for more complicated functions like regressions (which you'll be doing a lot of), it is harder to see how one outlier input is affecting the entire model.

Substep one: Subset the data to include only females with a postgrad education and save it as a dataframe called *pgfem*

```
pgfem <- anes3 %>%
  filter(educ == "postgrad" & gender == "female")
```

Substep two: to make sure the "meat" part of the for loop is correct, practice running with one iteration

Remove the first row of the *pgfem* dataframe and save it as a new object called *pgfem.minus1*. Then take the mean of the opposition to free trade variable in *pgfem.minus1* and save it as an object called *ftmean*.

Answer:

```
pgfem.minus1 <- pgfem[-1, ]
ftmean <- mean(pgfem.minus1$freetradenum)
```

Question one: What type of object is ftmean?

Answer: numeric

Substep three: then, use that general code to generate a for loop

Run a for loop that iterates through the data, removes an observation, takes the mean, and stores the mean value in a vector called *mean.vec*.

```r
# create mean.vec container
mean.vec <- c()
pgfem <- anes3 %>%
  filter(educ == "postgrad" & gender == "female")
#loop
for(i in 1:nrow(pgfem)){
  pgfem.minus1 <- pgfem[-i, ]
  ftmean <- mean(pgfem.minus1$freetradenum)
  mean.vec[i] <- ftmean
}
```

Substep four: Plot

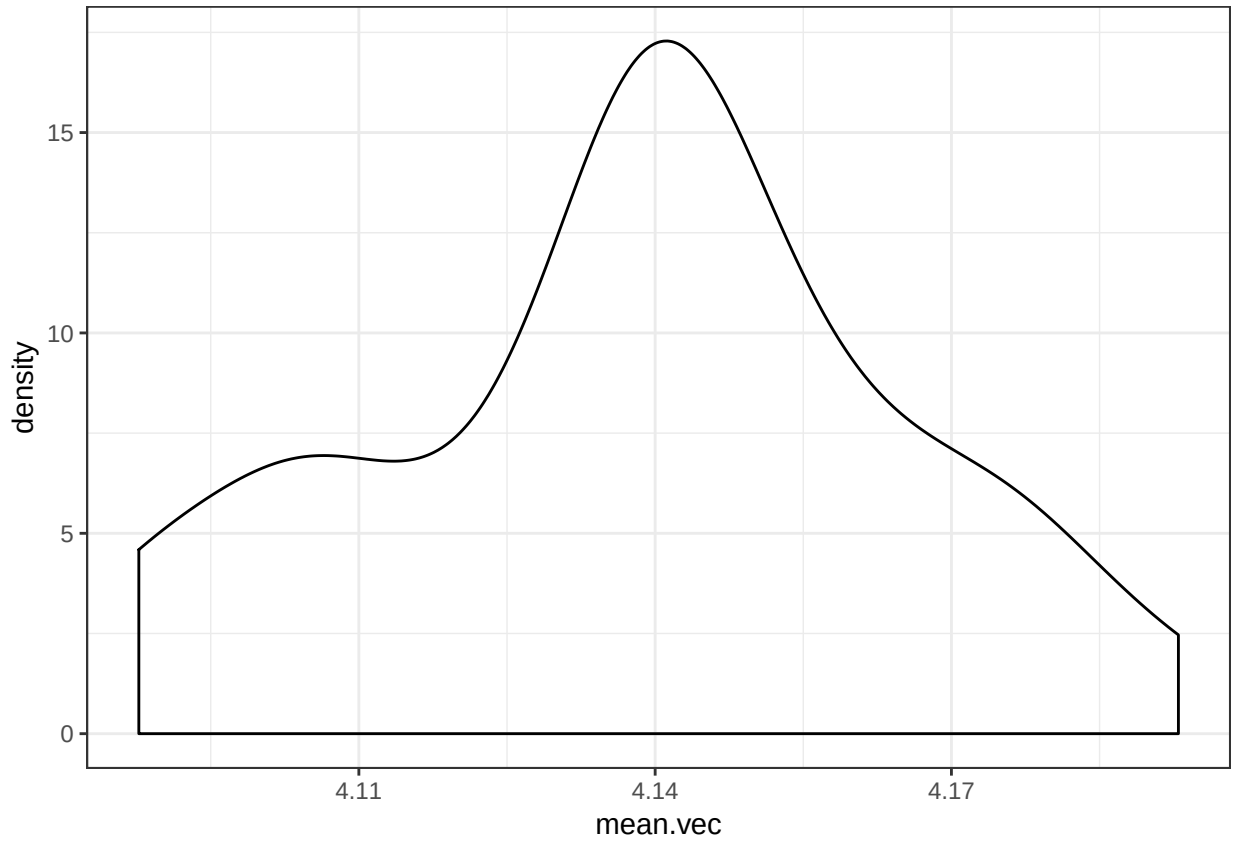
Use ggplot2 to do a density plot of the means. *Hint: before inputting, you'll have to turn the vector into something that adheres to ggplot's plotting requirements such as a data.frame*

Answer:

```r
ftdf <- as.data.frame(mean.vec)
head(mean.vec)
```

```
## [1] 4.140351 4.157895 4.140351 4.105263 4.105263 4.105263
```

```r
ggplot(ftdf, aes(x = mean.vec)) +
  geom_density() +
  theme_bw()
```



What was the mean value of the *freetradenum* variable in the entire *pgfem* data? To your plot, add a red vertical dashed line that intersects the x-axis at this value.

```
ggplot(ftdf, aes(x = mean.vec)) +
  geom_density() +
  theme_bw() +
  geom_vline(xintercept = mean(pgfem$freetradenum),
             color = "red",
             linetype = "dashed")
```