Problem Set #1

Sven Svensson

# This is just to make sure you know in Word you have to load these.  
library(tidyverse)  
library(stevedata)

This example document will mimic what you will do in the first homework. It will make use of the ESS9GB data in {stevedata}. You can [read more about those data here](https://svmiller.com/stevedata/reference/ESS9GB.html). I also have [a blog post](https://svmiller.com/blog/2020/03/what-explains-british-attitudes-toward-immigration-a-pedagogical-example/) and [a grad-level lab script](https://eh6127.svmiller.com/lab-scripts/lab-4/) built around analyzing these data.

# I can't really practice the use of `read\_csv()` in the context of a guide like this.  
my\_data <- ESS9GB

Now, let’s see how many columns there are. There are a number of ways of doing this and the eagle-eye student will note that the default “tibble” format basically tells you this as well.

ncol(my\_data)  
#> [1] 19  
dim(my\_data) # rows, then columns  
#> [1] 1905 19

Let’s see how many rows there are.

nrow(my\_data)  
#> [1] 1905  
dim(my\_data) # rows, then columns  
#> [1] 1905 19

Let’s see what the mean and median are of age. Again, there are a number of ways of doing this. Note that the presence of any missing observations results in an error here, so you should get in the habit of explicitly telling R that you don’t care if there are missing values and that you just want the summary statistic of what’s available. That would be specifying an argument of na.rm = TRUE or na.rm = T in your mean() and median() function.

mean(my\_data$agea, na.rm=TRUE)  
#> [1] 53.67301  
median(my\_data$agea, na.rm=TRUE)  
#> [1] 55  
  
# notice the pipe operator.  
my\_data %>%  
 summarize(avgage = mean(agea, na.rm=T),  
 medage = median(agea, na.rm=T))  
#> # A tibble: 1 × 2  
#> avgage medage  
#> <dbl> <dbl>  
#> 1 53.7 55

Let’s do the same for the years of education variable.

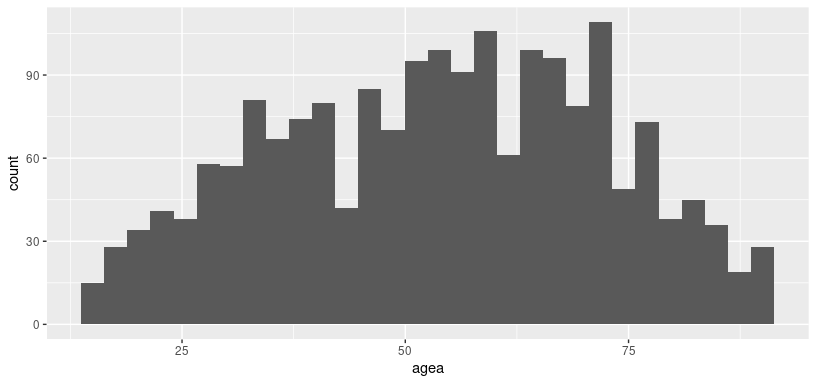
mean(my\_data$eduyrs, na.rm=TRUE)  
#> [1] 14.04913  
median(my\_data$eduyrs, na.rm=TRUE)  
#> [1] 13  
  
# notice the pipe operator.  
my\_data %>%  
 summarize(avgage = mean(eduyrs, na.rm=T),  
 medage = median(eduyrs, na.rm=T))  
#> # A tibble: 1 × 2  
#> avgage medage  
#> <dbl> <dbl>  
#> 1 14.0 13

Finally, let’s say isolate the respondents who live in London.

my\_data %>%  
 filter(region %in% c("London"))  
#> # A tibble: 114 × 19  
#> name essround edition idno cntry region brncntr stintrvw endintrvw   
#> <chr> <dbl> <chr> <dbl> <chr> <chr> <dbl> <date> <date>   
#> 1 ESS9e01\_2 9 1.2 238 GB London 1 2019-01-31 2019-01-31  
#> 2 ESS9e01\_2 9 1.2 325 GB London 1 2018-09-06 2018-09-06  
#> 3 ESS9e01\_2 9 1.2 849 GB London 1 2018-10-24 2018-10-24  
#> 4 ESS9e01\_2 9 1.2 1077 GB London 1 2019-01-03 2019-01-03  
#> 5 ESS9e01\_2 9 1.2 1677 GB London 1 2018-12-20 2018-12-20  
#> 6 ESS9e01\_2 9 1.2 1719 GB London 1 2018-09-17 2018-09-17  
#> 7 ESS9e01\_2 9 1.2 1729 GB London 1 2018-10-24 2018-10-24  
#> 8 ESS9e01\_2 9 1.2 2587 GB London 1 2018-09-26 2018-09-26  
#> 9 ESS9e01\_2 9 1.2 3064 GB London 1 2019-01-17 2019-01-17  
#> 10 ESS9e01\_2 9 1.2 3285 GB London 1 2018-10-13 2018-10-13  
#> # ℹ 104 more rows  
#> # ℹ 10 more variables: imbgeco <dbl>, imueclt <dbl>, imwbcnt <dbl>,  
#> # immigsent <dbl>, agea <dbl>, female <dbl>, eduyrs <dbl>, uempla <dbl>,  
#> # hinctnta <dbl>, lrscale <dbl>  
  
# Alternatively, if you have just one thing to find...  
  
my\_data %>%  
 filter(region == "London")  
#> # A tibble: 114 × 19  
#> name essround edition idno cntry region brncntr stintrvw endintrvw   
#> <chr> <dbl> <chr> <dbl> <chr> <chr> <dbl> <date> <date>   
#> 1 ESS9e01\_2 9 1.2 238 GB London 1 2019-01-31 2019-01-31  
#> 2 ESS9e01\_2 9 1.2 325 GB London 1 2018-09-06 2018-09-06  
#> 3 ESS9e01\_2 9 1.2 849 GB London 1 2018-10-24 2018-10-24  
#> 4 ESS9e01\_2 9 1.2 1077 GB London 1 2019-01-03 2019-01-03  
#> 5 ESS9e01\_2 9 1.2 1677 GB London 1 2018-12-20 2018-12-20  
#> 6 ESS9e01\_2 9 1.2 1719 GB London 1 2018-09-17 2018-09-17  
#> 7 ESS9e01\_2 9 1.2 1729 GB London 1 2018-10-24 2018-10-24  
#> 8 ESS9e01\_2 9 1.2 2587 GB London 1 2018-09-26 2018-09-26  
#> 9 ESS9e01\_2 9 1.2 3064 GB London 1 2019-01-17 2019-01-17  
#> 10 ESS9e01\_2 9 1.2 3285 GB London 1 2018-10-13 2018-10-13  
#> # ℹ 104 more rows  
#> # ℹ 10 more variables: imbgeco <dbl>, imueclt <dbl>, imwbcnt <dbl>,  
#> # immigsent <dbl>, agea <dbl>, female <dbl>, eduyrs <dbl>, uempla <dbl>,  
#> # hinctnta <dbl>, lrscale <dbl>  
  
# Alternatively, using subset() in base R.  
  
subset(my\_data, region %in% c("London"))  
#> # A tibble: 114 × 19  
#> name essround edition idno cntry region brncntr stintrvw endintrvw   
#> <chr> <dbl> <chr> <dbl> <chr> <chr> <dbl> <date> <date>   
#> 1 ESS9e01\_2 9 1.2 238 GB London 1 2019-01-31 2019-01-31  
#> 2 ESS9e01\_2 9 1.2 325 GB London 1 2018-09-06 2018-09-06  
#> 3 ESS9e01\_2 9 1.2 849 GB London 1 2018-10-24 2018-10-24  
#> 4 ESS9e01\_2 9 1.2 1077 GB London 1 2019-01-03 2019-01-03  
#> 5 ESS9e01\_2 9 1.2 1677 GB London 1 2018-12-20 2018-12-20  
#> 6 ESS9e01\_2 9 1.2 1719 GB London 1 2018-09-17 2018-09-17  
#> 7 ESS9e01\_2 9 1.2 1729 GB London 1 2018-10-24 2018-10-24  
#> 8 ESS9e01\_2 9 1.2 2587 GB London 1 2018-09-26 2018-09-26  
#> 9 ESS9e01\_2 9 1.2 3064 GB London 1 2019-01-17 2019-01-17  
#> 10 ESS9e01\_2 9 1.2 3285 GB London 1 2018-10-13 2018-10-13  
#> # ℹ 104 more rows  
#> # ℹ 10 more variables: imbgeco <dbl>, imueclt <dbl>, imwbcnt <dbl>,  
#> # immigsent <dbl>, agea <dbl>, female <dbl>, eduyrs <dbl>, uempla <dbl>,  
#> # hinctnta <dbl>, lrscale <dbl>  
  
# The benefit of the %in% approach is using it to isolate even more. Like this:  
  
subset(my\_data, region %in% c("London", "Scotland"))  
#> # A tibble: 283 × 19  
#> name essround edition idno cntry region brncntr stintrvw endintrvw   
#> <chr> <dbl> <chr> <dbl> <chr> <chr> <dbl> <date> <date>   
#> 1 ESS9e01\_2 9 1.2 238 GB London 1 2019-01-31 2019-01-31  
#> 2 ESS9e01\_2 9 1.2 325 GB London 1 2018-09-06 2018-09-06  
#> 3 ESS9e01\_2 9 1.2 512 GB Scotland 1 2018-10-11 2018-10-11  
#> 4 ESS9e01\_2 9 1.2 718 GB Scotland 1 2018-11-01 2018-11-01  
#> 5 ESS9e01\_2 9 1.2 849 GB London 1 2018-10-24 2018-10-24  
#> 6 ESS9e01\_2 9 1.2 1077 GB London 1 2019-01-03 2019-01-03  
#> 7 ESS9e01\_2 9 1.2 1395 GB Scotland 1 2019-01-22 2019-01-22  
#> 8 ESS9e01\_2 9 1.2 1432 GB Scotland 1 2018-09-10 2018-09-10  
#> 9 ESS9e01\_2 9 1.2 1677 GB London 1 2018-12-20 2018-12-20  
#> 10 ESS9e01\_2 9 1.2 1719 GB London 1 2018-09-17 2018-09-17  
#> # ℹ 273 more rows  
#> # ℹ 10 more variables: imbgeco <dbl>, imueclt <dbl>, imwbcnt <dbl>,  
#> # immigsent <dbl>, agea <dbl>, female <dbl>, eduyrs <dbl>, uempla <dbl>,  
#> # hinctnta <dbl>, lrscale <dbl>  
  
# It'd be good to know what the regions are, though...  
  
my\_data %>% distinct(region)  
#> # A tibble: 12 × 1  
#> region   
#> <chr>   
#> 1 West Midlands (England)   
#> 2 South West (England)   
#> 3 South East (England)   
#> 4 Northern Ireland   
#> 5 London   
#> 6 East of England   
#> 7 East Midlands (England)   
#> 8 North West (England)   
#> 9 Scotland   
#> 10 Yorkshire and the Humber  
#> 11 North East (England)   
#> 12 Wales

Here’s a plot. This is text!

ggplot(my\_data, aes(agea)) + geom\_histogram()



There’s still plenty more to do…