Lecture 2 Code

Alex Hoagland

2024-02-06

# I like to include several additional notes in the header of my files here:  
#  
# Last modified: 10/4/2023  
#  
### PURPOSE:  
 # Lecture 2 code and output file  
#  
### NOTES:  
 # - uses the Tidyverse package and Dplyr  
 # - uses the NHANES package to load data from the US National Health and Nutrition Examination Survey (NHANES, 1999-2004).

## Starting with R: Some things to be aware of

* Where is RStudio “pointed at” – what’s the current directory?
  + We’ll talk more about project organization as we proceed through the course
* Using *projects* in R – keeping files altogether
* RStudio is meant to be an “all-in-one” tool; you can use this to code (duh), but also to make and format output including figures, tables, documents (like this one), slides, and even apps!
* This makes it a convenient one-stop-shop for everything you do in a research project!

## R Markdown: Introduction

R Markdown documents are the files used to create HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

You can embed code (and its output) into these files. When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

You can embed an R code chunk like this:

# Any commands you want to have happen go here  
# summary(mydata) # note: what does R look like? What's a comment?

Let’s test this in copilot. Can I create a chunk that calculates the mean of a variable? What about windsorizing the mean?

## Packages in R

R is a collaborative open-source language, which means people write and share packages that do all sorts of helpful things. We’ll make use of some of the most common packages for econometric reserach throughout this course.

# The packages we will need for this file  
# install.packages('tidyverse') # if you need to install the package, uncomment this line.  
# install.packages('NHANES')  
  
library(tidyverse) # load the installed package for each new session of R

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.4.0 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(NHANES) # where we'll get our data extract  
?NHANES # documentation for a package or function looks like this! This one gives us a list of all variables we could want

## starting httpd help server ...

## done

## Loading Data

Let’s load some data from the NHANES and save it as our own object.

mydata <- NHANES # gives us a random smaple of 10,000 observations  
 # note from documentation: simple random sample of the American population  
View(mydata) # Let's look at the structure

## Summarizing and Visualizing Data

For most of our data visualization needs, we will use ggplot. Note that data visualization is extremely important! Captivating visualizations make a great paper stand out, and bad ones can sink even the best paper.GGplot is extremely versatile! I recommend this book if you are looking to master ggplot: <https://tinyurl.com/4k4wj8px>

Let’s start with two continuous variables: age and BMI.

# Let's calculate the mean, median, and main quantiles  
summary(mydata$Age) # What do we see?

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 17.00 36.00 36.74 54.00 80.00

summary(mydata$BMI) # Note the NAs, what does this mean?

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 12.88 21.58 25.98 26.66 30.89 81.25 366

# Why are these useful? An example  
summary(mydata$DaysMentHlthBad) # Another example (this is discrete -- what do we get from the difference between the mean and the median?)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.000 0.000 0.000 4.127 4.000 30.000 2466

# Suppose you want to calculate a specific quantile  
quantile(mydata$Age) # standard

## 0% 25% 50% 75% 100%   
## 0 17 36 54 80

quantile(mydata$Age, probs=c(0, .01, .04, .33, .75, 1)) # fancy

## 0% 1% 4% 33% 75% 100%   
## 0 0 2 24 54 80

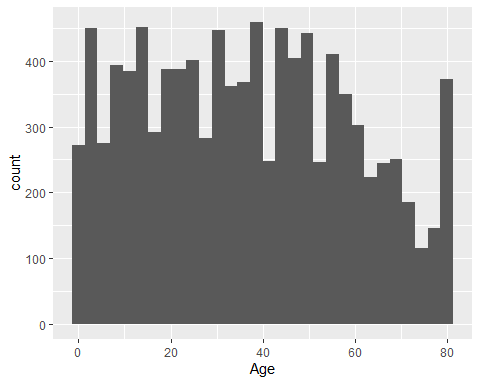
quantile(mydata$Age, probs=seq(0, 1, by=.1)) # uniform

## 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%   
## 0 7 14 21 29 36 43 50 58 68 80

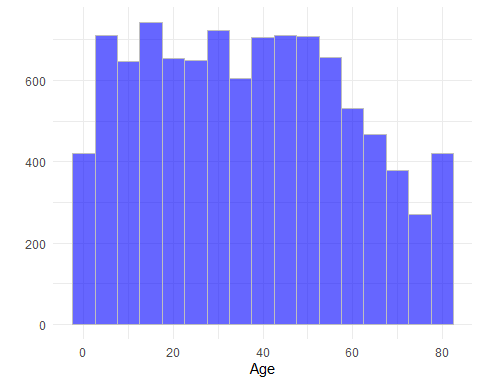
Now let’s start visualizing the data.

# first, a histogram of our two variables  
ggplot(data=mydata, aes(x=Age)) + geom\_histogram() # the simplest version. How can we clean it up? Let's do this live.

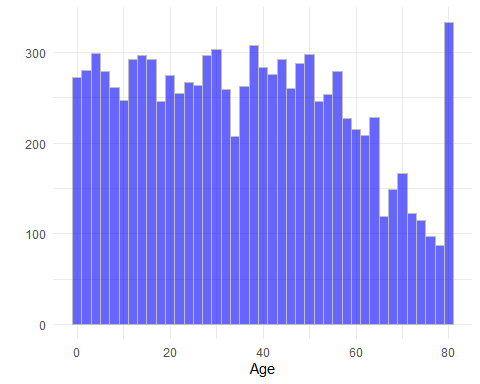
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



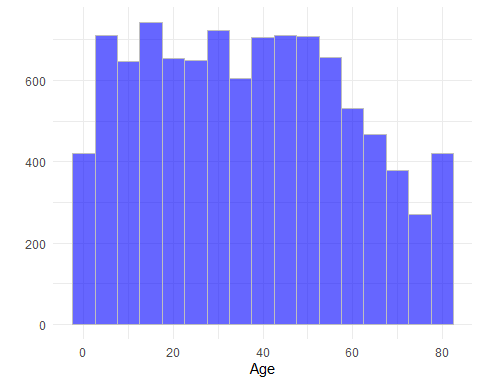
# A final version  
# includes: bin width, color, labels, theme  
ggplot(data=mydata, aes(x=Age)) + geom\_histogram(binwidth = 5, color="gray", fill="blue", alpha=.6) + labs(y = "") + theme\_minimal()



# A note on bin widths: what do you notice about these? What is similar/different?  
ggplot(data=mydata, aes(x=Age)) + geom\_histogram(binwidth = 2, color="gray", fill="blue", alpha=.6) + labs(y = "") + theme\_minimal()

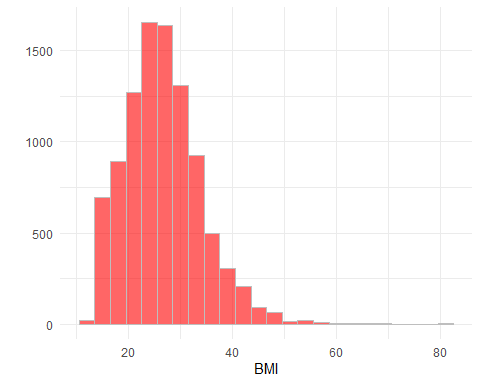


ggplot(data=mydata, aes(x=Age)) + geom\_histogram(binwidth = 5, color="gray", fill="blue", alpha=.6) + labs(y = "") + theme\_minimal()



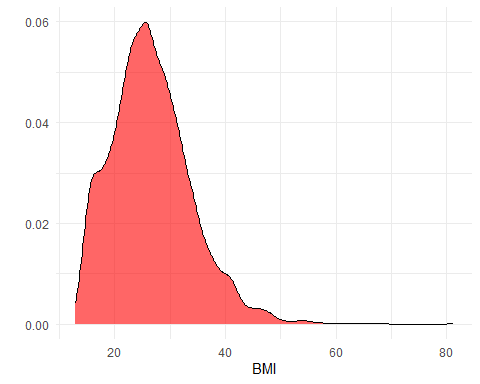
# Now let's look at BMI  
ggplot(data=mydata, aes(x=BMI)) + geom\_histogram(binwidth = 3, color="gray", fill="red", alpha=.6) + labs(y = "") + theme\_minimal()

## Warning: Removed 366 rows containing non-finite values (`stat\_bin()`).



# with enough data, can we ignore bin widths altogether?  
ggplot(data=mydata, aes(x=BMI)) + geom\_density(color="black", fill="red", alpha=.6) + labs(y = "") + theme\_minimal()

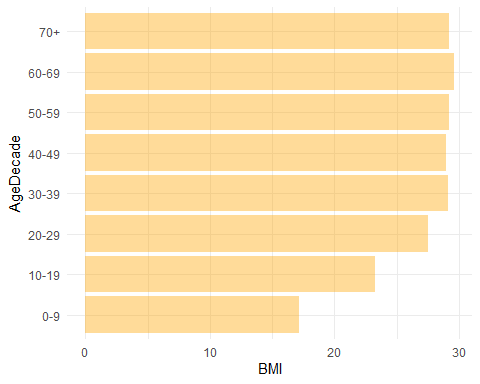
## Warning: Removed 366 rows containing non-finite values (`stat\_density()`).



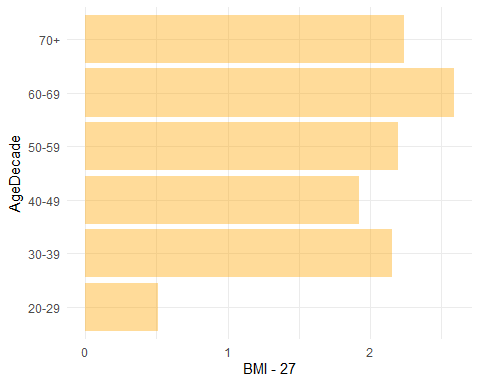
### Stratifying data

Now let’s look at these across groups. We’ll start with BMI across age bins.

# This gives us an introduction to pipe notation!  
mydata %>% group\_by(AgeDecade) %>% drop\_na(AgeDecade) %>%  
 summarize(BMI = mean(BMI, na.rm=T)) %>%  
 ggplot(aes(x=BMI,y=AgeDecade)) + geom\_bar(position="dodge",  
 stat="identity",  
 fill="orange", alpha=.4) +  
 theme\_minimal()



# What about honest visualization?  
mydata %>% group\_by(AgeDecade) %>% filter(Age >= 20) %>%  
 drop\_na(AgeDecade) %>%  
 summarize(BMI = mean(BMI, na.rm=T)) %>%  
 ggplot(aes(x=BMI-27,y=AgeDecade)) + geom\_bar(position="dodge",  
 stat="identity",  
 fill="orange", alpha=.4) +  
 theme\_minimal()



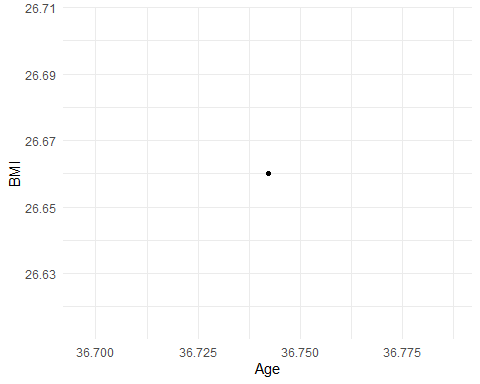
# What does this plot suggest? Is it fair?  
 # Note: data hack thanks to Stack Overflow! https://stackoverflow.com/questions/35324892/ggplot2-setting-geom-bar-baseline-to-1-instead-of-zero

## Covariance and Correlation

We can also define our own functions. This will become more important as the things we want to do in a single code file become more and more complex. Additionally, many econometricians like to code up their own estimators to really feel confident that they understand what is happening “under the hood” (we won’t do much of this here.)

Starting with a scatterplot:

#How useful is this?  
# ggplot(mydata, aes(x=Age,y=BMI))+geom\_point(color = 'green', alpha = .8, size = 2)+theme\_minimal()  
#  
# #Bubble chart  
# ggplot(mydata, aes(x=Age,y=BMI,size=PhysActive,color=PhysActive))+geom\_point(alpha = .8)+theme\_minimal()  
  
#Binscatter (in a function)  
binscatter <- function(nbins) { #note default choice  
 # this will create 25 quantiles using x and assign the observations in each quantile to a separate bin  
 mydata <- mydata %>% mutate(bin = ntile(Age, n=nbins))  
 mydata %>% group\_by(bin) %>%  
 summarise(xmean = mean(Age,na.rm=T), ymean = mean(BMI,na.rm=T)) %>%  
 #find the x and y mean of each bin  
 ggplot(aes(x=xmean, y=ymean)) +  
 geom\_point() + theme\_minimal() +  
 labs(x = 'Age', y = 'BMI')  
}  
  
binscatter(1) # what happens if I play around? Can you add other arguments?



### Covariance and Correlation

cov(mydata$BMI, mydata$Age,use="pairwise.complete.obs") # Base covariance -- but 65 what?

## [1] 65.58192

cov(mydata$BMI\*100, mydata$Age,use="pairwise.complete.obs")

## [1] 6558.192

cor(mydata$BMI, mydata$Age,use="pairwise.complete.obs") # This makes more sense!

## [1] 0.4078759

# What about relationships between multiple variables? Wait for regression, DAGs, etc.!

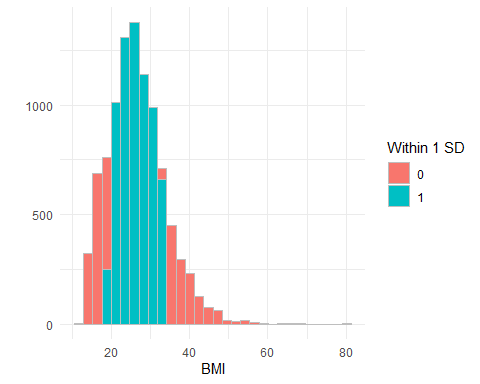
## Variance and Errors

Now let’s look at quantifying uncertainty

# This adds SDs to our histogram of BMI to show that we capture most of our data within a standard deviation  
mymean <- mean(mydata$BMI, na.rm = T)  
mysd <- sd(mydata$BMI, na.rm = T) # What does this mean?  
  
mydata <- mydata %>% drop\_na(BMI) %>%  
 mutate(myfill = ifelse(abs(BMI-mymean)<=mysd, 1, 0))  
mydata$myfill <- factor(mydata$myfill)  
ggplot(mydata, aes(x=BMI, fill = myfill)) +  
 geom\_histogram(width = 1, color='gray') +  
 labs(y = "", fill = "Within 1 SD") + theme\_minimal()

## Warning: The `width` argument of `stat\_bin()` is deprecated as of ggplot2 2.1.0.  
## ℹ Please use `geom\_bar()` instead.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## What do we do at the end of an RMarkdown file?

First, make sure to cite all the packages you used – it’s just as important to give credit to coding work as it is to other types of previous scholarly work.

### Package Citations

Let’s knit this file and save it!