Lecture 3 Code

Alex Hoagland

2025-01-20

Here is my header text, where I load my packages and look at documentation for them:

# I like to include several additional notes in the header of my files here:  
#  
# Last modified: 1/20/2025  
#  
### PURPOSE:  
 # Lecture 3 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).  
  
library(tidyverse) # load the installed package for each new session of R

## Warning: package 'tidyverse' was built under R version 4.2.3

## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tibble' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.2

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'purrr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.2

## Warning: package 'forcats' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(NHANES) # where we'll get our data extract

## Warning: package 'NHANES' was built under R version 4.2.3

?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  
 # What does each row (observation) represent? What kinds of variables do we have? How would we describe this dataset in words?

Now how would we describe this dataset in numbers?

## Simple tables and figures

Remember that tables are generally easier to work with outside of knitting, but you might want some simple code to generate the numbers

# Let's summarize days of poor mental health by education   
levels(mydata$Education) # What does this tell us?

## [1] "8th Grade" "9 - 11th Grade" "High School" "Some College"   
## [5] "College Grad"

tabledata <- mydata %>% group\_by(Education) %>%   
 summarize(mean\_days\_mental\_health = mean(DaysMentHlthBad, na.rm=T),  
 median\_days\_mental\_health = median(DaysMentHlthBad, na.rm=T),  
 n = n()) # Note that this came from copilot!   
kableExtra::kable(tabledata) # Now you can format this in Word! This is what I recommend for the next assignment.

## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output  
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'

Education

mean\_days\_mental\_health

median\_days\_mental\_health

n

8th Grade

4.659401

0

451

9 - 11th Grade

5.611538

0

888

High School

4.204529

0

1517

Some College

4.644358

0

2267

College Grad

3.126516

0

2098

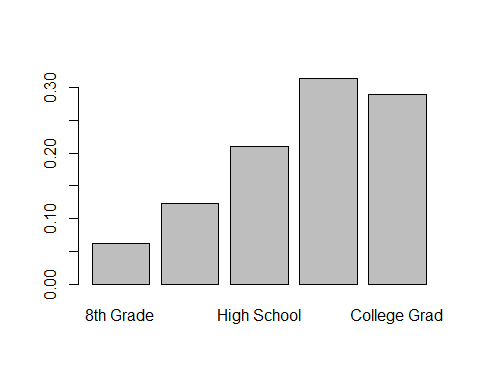
NA

3.545624

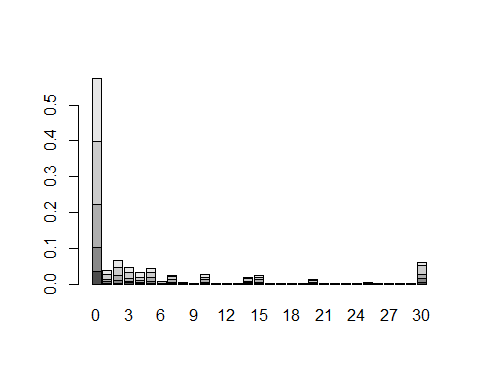
1

2779

# simple histogram  
barplot(prop.table(table(mydata$Education))) # What does this tell us?



barplot(prop.table(table(mydata$Education,mydata$DaysMentHlthBad))) # What does this tell us?

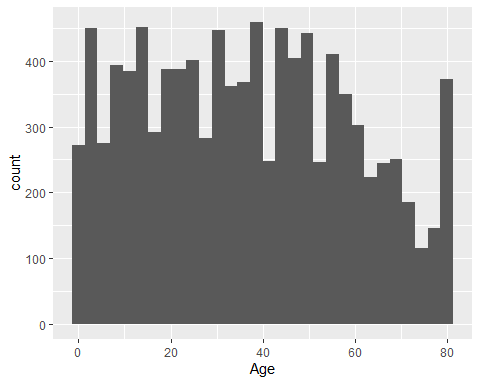


## Histograms

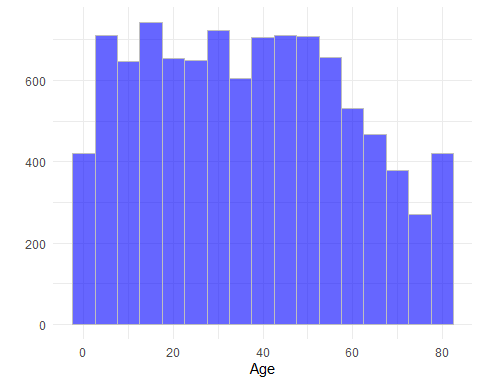
Now let’s start visualizing the data.

# 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.

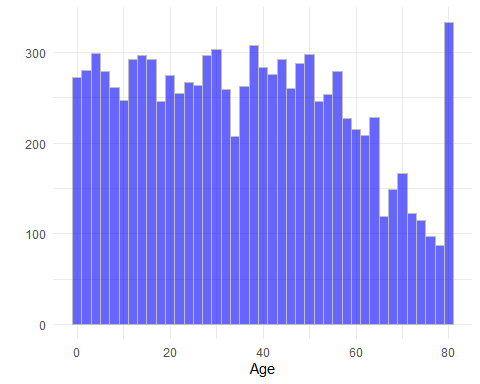
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



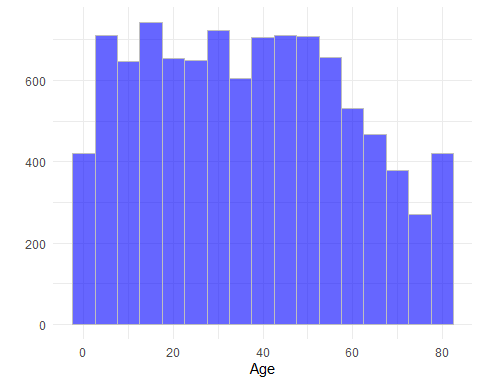
# A final version  
# includes: bin width, color, labels, theme  
ggplot(data=mydata, aes(x=Age)) + geom\_histogram(binwidth = 5, color="gray", fill="blue", alpha=.6) + labs(y = "") + theme\_minimal()



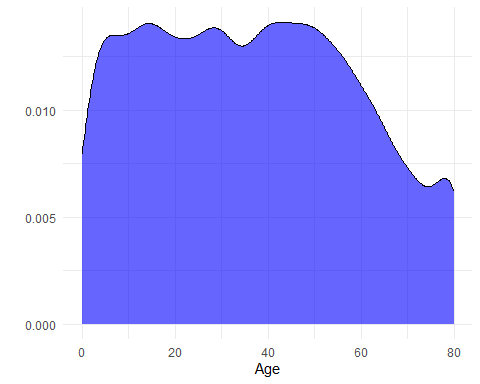
# A note on bin widths: what do you notice about these? What is similar/different?  
ggplot(data=mydata, aes(x=Age)) + geom\_histogram(binwidth = 2, color="gray", fill="blue", alpha=.6) + labs(y = "") + theme\_minimal()



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

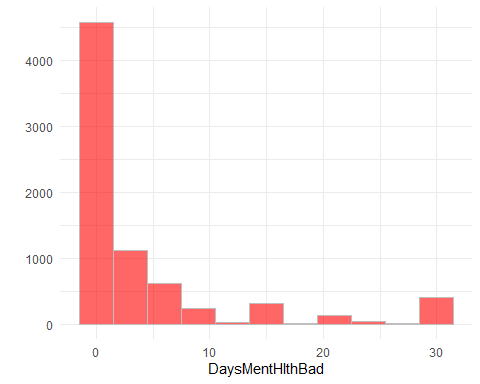


# with enough data, can we ignore bin widths altogether?  
ggplot(data=mydata, aes(x=Age)) + geom\_density(color="black", fill="blue", alpha=.6) + labs(y = "") + theme\_minimal()



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

## Warning: Removed 2466 rows containing non-finite values (`stat\_bin()`).



## Measures of central tendency

The histogram provides a nice overview, but we need some numbers to summarize lots of data in a simple way.

What if we try to summarize this Days of Poor Mental Health variable?

mean(mydata$DaysMentHlthBad) # uh oh!

## [1] NA

mean(mydata$DaysMentHlthBad, na.rm=T) # what did we do here?

## [1] 4.126493

median(mydata$DaysMentHlthBad, na.rm=T) # what do we conclude?

## [1] 0

quantile(mydata$DaysMentHlthBad,   
 probs = c(0.1, 0.25, 0.5, 0.75, 0.9),   
 na.rm=T) # what does this tell us?

## 10% 25% 50% 75% 90%   
## 0 0 0 4 15

# What if we want to zero in on a particular group?   
subset <- mydata %>% filter(DaysMentHlthBad >= 1) # What did we do?   
quantile(subset$DaysMentHlthBad,   
 probs = c(0.1, 0.25, 0.5, 0.75, 0.9),   
 na.rm=T) # what does this tell us?

## 10% 25% 50% 75% 90%   
## 1 2 5 14 30

# A (bit of a) shortcut  
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$DaysMentHlthBad) # Note the NAs, what does this mean?

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

Now what about variation?

sd(mydata$DaysMentHlthBad, na.rm=T) # How do we measure this?

## [1] 7.832971

# This tells us something about where the "useful" bit of data comes in  
var(mydata$DaysMentHlthBad, na.rm=T) # what is this in relation to SD?

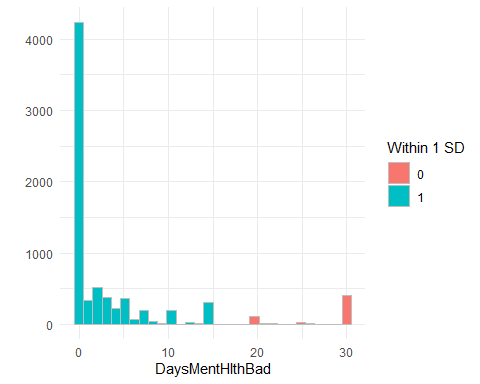
## [1] 61.35543

sd(subset$DaysMentHlthBad, na.rm=T) # Why is this larger than before?

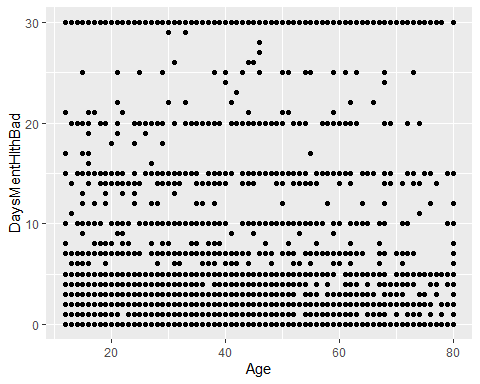
## [1] 9.497794

# This adds SDs to our histogram to show that we capture most of our data within a standard deviation  
mymean <- mean(mydata$DaysMentHlthBad, na.rm = T)  
mysd <- sd(mydata$DaysMentHlthBad, na.rm = T) # What does this mean?  
  
mydata <- mydata %>% drop\_na(DaysMentHlthBad) %>%  
 mutate(myfill = ifelse(abs(DaysMentHlthBad-mymean)<=mysd\*2, 1, 0))  
mydata$myfill <- factor(mydata$myfill)  
ggplot(mydata, aes(x=DaysMentHlthBad, fill = myfill)) +  
 geom\_histogram(width = 1, color='gray') +  
 labs(y = "", fill = "Within 1 SD") + theme\_minimal()

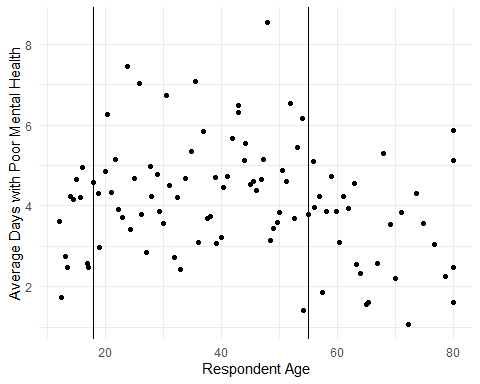
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## 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 days of poor mental health. We'll do a simple scatterplot and then a binned scatterplot:   
  
  
```r  
# Simple scatterplot between age and DaysMentHlthBad  
ggplot(data=mydata, aes(x=Age, y=DaysMentHlthBad)) + geom\_point() # What does this tell us?



# Now let's create a binned scatterplot   
mydata %>%  
 mutate(bin = ntile(Age, 100)) %>%   
 # What should we play around with to make this sensible?   
 group\_by(bin) %>%   
 summarize(xmean = mean(Age, na.rm=T),   
 ymean = mean(DaysMentHlthBad, na.rm=T)) %>%   
 ggplot(aes(x = xmean, y = ymean)) +  
 geom\_point() + theme\_minimal() +   
 geom\_vline(xintercept = 18) + geom\_vline(xintercept = 55) +   
 labs(x = "Respondent Age", y = "Average Days with Poor Mental Health")



### Covariance and Correlation

cov(mydata$DaysMentHlthBad, mydata$Age,use="pairwise.complete.obs") # Base covariance -- but in units of what?

## [1] -4.848859

cov(mydata$DaysMentHlthBad/7, mydata$Age,use="pairwise.complete.obs")

## [1] -0.6926941

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

## [1] -0.03200316

cor(mydata$DaysMentHlthBad/7, mydata$Age,use="pairwise.complete.obs") # This makes more sense!

## [1] -0.03200316

# What makes this number useful for you? What would you think about changing to make this number more useful, given our results above?