EPID674 Epidemiologic Data Analysis using R

Wrangling and Exploring Data with R

Kelly Bakulski, Lauren Middleton

Last compiled on October 25, 2022

## Session 2, Wrangling and Exploring Data with R

### Identify base packages and install new packages

sessionInfo() # check what packages are already loaded by default

R version 4.2.1 (2022-06-23)  
Platform: x86\_64-pc-linux-gnu (64-bit)  
Running under: Ubuntu 20.04.5 LTS  
  
Matrix products: default  
BLAS: /usr/lib/x86\_64-linux-gnu/atlas/libblas.so.3.10.3  
LAPACK: /usr/lib/x86\_64-linux-gnu/atlas/liblapack.so.3.10.3  
  
locale:  
 [1] LC\_CTYPE=C.UTF-8 LC\_NUMERIC=C LC\_TIME=C.UTF-8   
 [4] LC\_COLLATE=C.UTF-8 LC\_MONETARY=C.UTF-8 LC\_MESSAGES=C.UTF-8   
 [7] LC\_PAPER=C.UTF-8 LC\_NAME=C LC\_ADDRESS=C   
[10] LC\_TELEPHONE=C LC\_MEASUREMENT=C.UTF-8 LC\_IDENTIFICATION=C   
  
attached base packages:  
[1] stats graphics grDevices utils datasets methods base   
  
loaded via a namespace (and not attached):  
 [1] compiler\_4.2.1 magrittr\_2.0.3 fastmap\_1.1.0 cli\_3.4.1   
 [5] tools\_4.2.1 htmltools\_0.5.3 rstudioapi\_0.14 yaml\_2.3.6   
 [9] stringi\_1.7.8 rmarkdown\_2.17 knitr\_1.40 stringr\_1.4.1   
[13] xfun\_0.33 digest\_0.6.30 jsonlite\_1.8.2 rlang\_1.0.6   
[17] evaluate\_0.17

### Install new packages

# The operator #| is called the hashpipe. This is how we specify options for the entire code chunk.   
# In this case with the option eval: false, we're telling R not to actually evaluate this code chunk.  
  
# Install packages. Do this only once.  
# Note, we already installed packages to this workspace for the class. If working on your personal computer, will need to run this code  
install.packages("tidyverse")  
install.packages("here")  
install.packages("nhanesA")  
install.packages("sjlabelled")  
  
# To actually install the packages on your personal computer: change option in with the hashpipe #| operator to eval: true

## Load packages

# Load packages. Load relevant packages every time you start a new R session and at the top of every .Rmd file  
library(tidyverse)

── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
✔ ggplot2 3.3.6 ✔ purrr 0.3.5   
✔ tibble 3.1.8 ✔ dplyr 1.0.10  
✔ tidyr 1.2.1 ✔ stringr 1.4.1   
✔ readr 2.1.3 ✔ forcats 0.5.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()

library(here)

here() starts at /cloud/project

library(nhanesA)  
library(sjlabelled)

Attaching package: 'sjlabelled'  
  
The following object is masked from 'package:forcats':  
  
 as\_factor  
  
The following object is masked from 'package:dplyr':  
  
 as\_label  
  
The following object is masked from 'package:ggplot2':  
  
 as\_label

## Specify file directories

here() # Orient yourself to the default file path format on your computer

[1] "/cloud/project"

#Expect to be when coding on RStudio Cloud  
#"/cloud/project"

## Data cleaning plan:

1. Load demographics, complete blood counts, and chemical datasets
2. Check datasets
3. Keep only the useful variables
4. Join datasets together
5. Create new numeric variables from numeric variables
6. Create categorical variables from numeric variables
7. Update object types and set reference levels for factor variables
8. Save cleaned dataset

## Import datasets from 2017-2018 NHANES

* Datasets can be found at: **https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?BeginYear=2017**
* Data are saved in multiple datasets that we will need to combine, starting with demographics
* NHANES variables have labels as well as variable names

# Download NHANES demographics data and remove the labels  
demog <- nhanes("DEMO\_J")  
# Check the column names (variable names)  
colnames(demog)

[1] "SEQN" "SDDSRVYR" "RIDSTATR" "RIAGENDR" "RIDAGEYR" "RIDAGEMN"  
 [7] "RIDRETH1" "RIDRETH3" "RIDEXMON" "RIDEXAGM" "DMQMILIZ" "DMQADFC"   
[13] "DMDBORN4" "DMDCITZN" "DMDYRSUS" "DMDEDUC3" "DMDEDUC2" "DMDMARTL"  
[19] "RIDEXPRG" "SIALANG" "SIAPROXY" "SIAINTRP" "FIALANG" "FIAPROXY"  
[25] "FIAINTRP" "MIALANG" "MIAPROXY" "MIAINTRP" "AIALANGA" "DMDHHSIZ"  
[31] "DMDFMSIZ" "DMDHHSZA" "DMDHHSZB" "DMDHHSZE" "DMDHRGND" "DMDHRAGZ"  
[37] "DMDHREDZ" "DMDHRMAZ" "DMDHSEDZ" "WTINT2YR" "WTMEC2YR" "SDMVPSU"   
[43] "SDMVSTRA" "INDHHIN2" "INDFMIN2" "INDFMPIR"

# Check the Demographics Data website for information about these variables:  
# https://wwwn.cdc.gov/nchs/nhanes/search/datapage.aspx?Component=Demographics&CycleBeginYear=2017  
# What does RIAGENDR mean?

## Explore the NHANES demographics dataset

# Explore the data set  
str(demog) # Get detailed information about each variable and the object overall

'data.frame': 9254 obs. of 46 variables:  
 $ SEQN : num 93703 93704 93705 93706 93707 ...  
 $ SDDSRVYR: num 10 10 10 10 10 10 10 10 10 10 ...  
 $ RIDSTATR: num 2 2 2 2 2 2 2 2 2 2 ...  
 $ RIAGENDR: num 2 1 2 1 1 2 2 2 1 1 ...  
 $ RIDAGEYR: num 2 2 66 18 13 66 75 0 56 18 ...  
 $ RIDAGEMN: num NA NA NA NA NA NA NA 11 NA NA ...  
 $ RIDRETH1: num 5 3 4 5 5 5 4 3 5 1 ...  
 $ RIDRETH3: num 6 3 4 6 7 6 4 3 6 1 ...  
 $ RIDEXMON: num 2 1 2 2 2 2 1 2 2 2 ...  
 $ RIDEXAGM: num 27 33 NA 222 158 NA NA 13 NA 227 ...  
 $ DMQMILIZ: num NA NA 2 2 NA 2 2 NA 2 2 ...  
 $ DMQADFC : num NA NA NA NA NA NA NA NA NA NA ...  
 $ DMDBORN4: num 1 1 1 1 1 2 1 1 2 2 ...  
 $ DMDCITZN: num 1 1 1 1 1 1 1 1 1 2 ...  
 $ DMDYRSUS: num NA NA NA NA NA 7 NA NA 6 5 ...  
 $ DMDEDUC3: num NA NA NA 15 6 NA NA NA NA 12 ...  
 $ DMDEDUC2: num NA NA 2 NA NA 1 4 NA 5 NA ...  
 $ DMDMARTL: num NA NA 3 NA NA 1 2 NA 1 NA ...  
 $ RIDEXPRG: num NA NA NA NA NA NA NA NA NA NA ...  
 $ SIALANG : num 1 1 1 1 1 1 1 1 1 1 ...  
 $ SIAPROXY: num 1 1 2 2 1 2 2 1 2 2 ...  
 $ SIAINTRP: num 2 2 2 2 2 1 2 2 2 2 ...  
 $ FIALANG : num 1 1 1 NA 1 1 1 1 1 2 ...  
 $ FIAPROXY: num 2 2 2 NA 2 2 2 2 2 2 ...  
 $ FIAINTRP: num 2 2 2 NA 2 2 2 2 2 2 ...  
 $ MIALANG : num NA NA 1 1 1 1 NA NA 1 1 ...  
 $ MIAPROXY: num NA NA 2 2 2 2 NA NA 2 2 ...  
 $ MIAINTRP: num NA NA 2 2 2 1 NA NA 2 2 ...  
 $ AIALANGA: num NA NA 1 1 1 3 NA NA 1 1 ...  
 $ DMDHHSIZ: num 5 4 1 5 7 2 1 3 3 4 ...  
 $ DMDFMSIZ: num 5 4 1 5 7 2 1 3 3 4 ...  
 $ DMDHHSZA: num 3 2 0 0 0 0 0 1 0 0 ...  
 $ DMDHHSZB: num 0 0 0 0 3 0 0 0 0 2 ...  
 $ DMDHHSZE: num 0 0 1 1 0 2 1 0 0 0 ...  
 $ DMDHRGND: num 1 1 2 1 1 1 2 1 1 2 ...  
 $ DMDHRAGZ: num 2 2 4 4 3 4 4 2 3 3 ...  
 $ DMDHREDZ: num 3 3 1 3 2 1 2 3 3 1 ...  
 $ DMDHRMAZ: num 1 1 2 1 1 1 2 1 1 2 ...  
 $ DMDHSEDZ: num 3 2 NA 2 3 1 NA 3 3 NA ...  
 $ WTINT2YR: num 9246 37339 8615 8549 6769 ...  
 $ WTMEC2YR: num 8540 42567 8338 8723 7065 ...  
 $ SDMVPSU : num 2 1 2 2 1 2 1 1 2 2 ...  
 $ SDMVSTRA: num 145 143 145 134 138 138 136 134 134 147 ...  
 $ INDHHIN2: num 15 15 3 NA 10 6 2 15 15 4 ...  
 $ INDFMIN2: num 15 15 3 NA 10 6 2 15 15 4 ...  
 $ INDFMPIR: num 5 5 0.82 NA 1.88 1.63 0.41 4.9 5 0.76 ...

dim(demog) # What are the dimensions?

[1] 9254 46

colnames(demog) # What are the column names?

[1] "SEQN" "SDDSRVYR" "RIDSTATR" "RIAGENDR" "RIDAGEYR" "RIDAGEMN"  
 [7] "RIDRETH1" "RIDRETH3" "RIDEXMON" "RIDEXAGM" "DMQMILIZ" "DMQADFC"   
[13] "DMDBORN4" "DMDCITZN" "DMDYRSUS" "DMDEDUC3" "DMDEDUC2" "DMDMARTL"  
[19] "RIDEXPRG" "SIALANG" "SIAPROXY" "SIAINTRP" "FIALANG" "FIAPROXY"  
[25] "FIAINTRP" "MIALANG" "MIAPROXY" "MIAINTRP" "AIALANGA" "DMDHHSIZ"  
[31] "DMDFMSIZ" "DMDHHSZA" "DMDHHSZB" "DMDHHSZE" "DMDHRGND" "DMDHRAGZ"  
[37] "DMDHREDZ" "DMDHRMAZ" "DMDHSEDZ" "WTINT2YR" "WTMEC2YR" "SDMVPSU"   
[43] "SDMVSTRA" "INDHHIN2" "INDFMIN2" "INDFMPIR"

rownames(demog)[1:20] # What are the row names?

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15"  
[16] "16" "17" "18" "19" "20"

head(demog) # What do the first 6 rows look like?

SEQN SDDSRVYR RIDSTATR RIAGENDR RIDAGEYR RIDAGEMN RIDRETH1 RIDRETH3 RIDEXMON  
1 93703 10 2 2 2 NA 5 6 2  
2 93704 10 2 1 2 NA 3 3 1  
3 93705 10 2 2 66 NA 4 4 2  
4 93706 10 2 1 18 NA 5 6 2  
5 93707 10 2 1 13 NA 5 7 2  
6 93708 10 2 2 66 NA 5 6 2  
 RIDEXAGM DMQMILIZ DMQADFC DMDBORN4 DMDCITZN DMDYRSUS DMDEDUC3 DMDEDUC2  
1 27 NA NA 1 1 NA NA NA  
2 33 NA NA 1 1 NA NA NA  
3 NA 2 NA 1 1 NA NA 2  
4 222 2 NA 1 1 NA 15 NA  
5 158 NA NA 1 1 NA 6 NA  
6 NA 2 NA 2 1 7 NA 1  
 DMDMARTL RIDEXPRG SIALANG SIAPROXY SIAINTRP FIALANG FIAPROXY FIAINTRP MIALANG  
1 NA NA 1 1 2 1 2 2 NA  
2 NA NA 1 1 2 1 2 2 NA  
3 3 NA 1 2 2 1 2 2 1  
4 NA NA 1 2 2 NA NA NA 1  
5 NA NA 1 1 2 1 2 2 1  
6 1 NA 1 2 1 1 2 2 1  
 MIAPROXY MIAINTRP AIALANGA DMDHHSIZ DMDFMSIZ DMDHHSZA DMDHHSZB DMDHHSZE  
1 NA NA NA 5 5 3 0 0  
2 NA NA NA 4 4 2 0 0  
3 2 2 1 1 1 0 0 1  
4 2 2 1 5 5 0 0 1  
5 2 2 1 7 7 0 3 0  
6 2 1 3 2 2 0 0 2  
 DMDHRGND DMDHRAGZ DMDHREDZ DMDHRMAZ DMDHSEDZ WTINT2YR WTMEC2YR SDMVPSU  
1 1 2 3 1 3 9246.492 8539.731 2  
2 1 2 3 1 2 37338.768 42566.615 1  
3 2 4 1 2 NA 8614.571 8338.420 2  
4 1 4 3 1 2 8548.633 8723.440 2  
5 1 3 2 1 3 6769.345 7064.610 1  
6 1 4 1 1 1 13329.451 14372.489 2  
 SDMVSTRA INDHHIN2 INDFMIN2 INDFMPIR  
1 145 15 15 5.00  
2 143 15 15 5.00  
3 145 3 3 0.82  
4 134 NA NA NA  
5 138 10 10 1.88  
6 138 6 6 1.63

demog[1:10, ] # What do the first 10 rows look like?

SEQN SDDSRVYR RIDSTATR RIAGENDR RIDAGEYR RIDAGEMN RIDRETH1 RIDRETH3  
1 93703 10 2 2 2 NA 5 6  
2 93704 10 2 1 2 NA 3 3  
3 93705 10 2 2 66 NA 4 4  
4 93706 10 2 1 18 NA 5 6  
5 93707 10 2 1 13 NA 5 7  
6 93708 10 2 2 66 NA 5 6  
7 93709 10 2 2 75 NA 4 4  
8 93710 10 2 2 0 11 3 3  
9 93711 10 2 1 56 NA 5 6  
10 93712 10 2 1 18 NA 1 1  
 RIDEXMON RIDEXAGM DMQMILIZ DMQADFC DMDBORN4 DMDCITZN DMDYRSUS DMDEDUC3  
1 2 27 NA NA 1 1 NA NA  
2 1 33 NA NA 1 1 NA NA  
3 2 NA 2 NA 1 1 NA NA  
4 2 222 2 NA 1 1 NA 15  
5 2 158 NA NA 1 1 NA 6  
6 2 NA 2 NA 2 1 7 NA  
7 1 NA 2 NA 1 1 NA NA  
8 2 13 NA NA 1 1 NA NA  
9 2 NA 2 NA 2 1 6 NA  
10 2 227 2 NA 2 2 5 12  
 DMDEDUC2 DMDMARTL RIDEXPRG SIALANG SIAPROXY SIAINTRP FIALANG FIAPROXY  
1 NA NA NA 1 1 2 1 2  
2 NA NA NA 1 1 2 1 2  
3 2 3 NA 1 2 2 1 2  
4 NA NA NA 1 2 2 NA NA  
5 NA NA NA 1 1 2 1 2  
6 1 1 NA 1 2 1 1 2  
7 4 2 NA 1 2 2 1 2  
8 NA NA NA 1 1 2 1 2  
9 5 1 NA 1 2 2 1 2  
10 NA NA NA 1 2 2 2 2  
 FIAINTRP MIALANG MIAPROXY MIAINTRP AIALANGA DMDHHSIZ DMDFMSIZ DMDHHSZA  
1 2 NA NA NA NA 5 5 3  
2 2 NA NA NA NA 4 4 2  
3 2 1 2 2 1 1 1 0  
4 NA 1 2 2 1 5 5 0  
5 2 1 2 2 1 7 7 0  
6 2 1 2 1 3 2 2 0  
7 2 NA NA NA NA 1 1 0  
8 2 NA NA NA NA 3 3 1  
9 2 1 2 2 1 3 3 0  
10 2 1 2 2 1 4 4 0  
 DMDHHSZB DMDHHSZE DMDHRGND DMDHRAGZ DMDHREDZ DMDHRMAZ DMDHSEDZ WTINT2YR  
1 0 0 1 2 3 1 3 9246.492  
2 0 0 1 2 3 1 2 37338.768  
3 0 1 2 4 1 2 NA 8614.571  
4 0 1 1 4 3 1 2 8548.633  
5 3 0 1 3 2 1 3 6769.345  
6 0 2 1 4 1 1 1 13329.451  
7 0 1 2 4 2 2 NA 12043.388  
8 0 0 1 2 3 1 3 16418.298  
9 0 0 1 3 3 1 3 11178.260  
10 2 0 2 3 1 2 NA 29040.497  
 WTMEC2YR SDMVPSU SDMVSTRA INDHHIN2 INDFMIN2 INDFMPIR  
1 8539.731 2 145 15 15 5.00  
2 42566.615 1 143 15 15 5.00  
3 8338.420 2 145 3 3 0.82  
4 8723.440 2 134 NA NA NA  
5 7064.610 1 138 10 10 1.88  
6 14372.489 2 138 6 6 1.63  
7 12277.557 1 136 2 2 0.41  
8 16848.020 1 134 15 15 4.90  
9 12390.920 2 134 15 15 5.00  
10 30336.654 2 147 4 4 0.76

# Explore individual variables in the NHANES dataset

# Explore the variables, starting with RIDAGEYR  
length(demog$RIDAGEYR) # How many observations are in this variable?

[1] 9254

str(demog$RIDAGEYR) # What type of object is the variable?

num [1:9254] 2 2 66 18 13 66 75 0 56 18 ...

demog$RIDAGEYR[1:10] # Are there any duplicates in the first few values?

[1] 2 2 66 18 13 66 75 0 56 18

# Check your understanding!

When you start with a new dataset, be sure to explore every variable. As an example, let’s explore another variable: INDFMPIR \* Make a new R code chunk \* How long is the INDFMPIR variable? \* What type of object is the variable? (for example: character, numeric, etc.) \* What are the first few values of the variable?

## Let’s load additional datasets, beyond demographics. This includes blood cell counts, metal levels, and cotinine levels

# Download Complete Blood Cell Laboratory Data, check the column names  
cbc <- nhanes("CBC\_J")  
colnames(cbc)

[1] "SEQN" "LBXWBCSI" "LBXLYPCT" "LBXMOPCT" "LBXNEPCT" "LBXEOPCT"  
 [7] "LBXBAPCT" "LBDLYMNO" "LBDMONO" "LBDNENO" "LBDEONO" "LBDBANO"   
[13] "LBXRBCSI" "LBXHGB" "LBXHCT" "LBXMCVSI" "LBXMCHSI" "LBXMC"   
[19] "LBXRDW" "LBXPLTSI" "LBXMPSI" "LBXNRBC"

# Download Iron Data, check column names  
fe <- nhanes("FETIB\_J")  
colnames(fe)

[1] "SEQN" "LBXIRN" "LBDIRNSI" "LBXUIB" "LBDUIBLC" "LBDUIBSI" "LBDTIB"   
[8] "LBDTIBSI" "LBDPCT"

# Download Arsenic Data, check column names  
as <- nhanes("UTAS\_J")  
colnames(as)

[1] "SEQN" "WTSA2YR" "URXUAS" "URDUASLC"

# Download Cadmium and Lead Data, check column names  
cd\_pb <- nhanes("PBCD\_J")  
colnames(cd\_pb)

[1] "SEQN" "LBXBPB" "LBDBPBSI" "LBDBPBLC" "LBXBCD" "LBDBCDSI"  
 [7] "LBDBCDLC" "LBXTHG" "LBDTHGSI" "LBDTHGLC" "LBXBSE" "LBDBSESI"  
[13] "LBDBSELC" "LBXBMN" "LBDBMNSI" "LBDBMNLC"

# Download Cotinine Data, check column names  
cotinine <- nhanes("COT\_J")  
colnames(cotinine)

[1] "SEQN" "LBXCOT" "LBDCOTLC" "LBXHCT" "LBDHCTLC"

# What variable is the same between the datasets?

## Select columns of datasets to keep only variables that we will use in the class

# select() is a useful tidyverse function that keeps columns in a dataframe  
# We only want to keep the necessary variables to avoid creating a huge dataset  
  
############ Demographics dataset ############  
dim(demog)

[1] 9254 46

#9254 46  
demog\_select <- demog %>%  
 select(SEQN, #participant identifier  
 RIAGENDR, #sex  
 RIDAGEYR, #age in years  
 RIDRETH1, #race in number code  
 INDFMPIR, #poverty-income ratio  
 DMDEDUC3, #education categories for age 6-19  
 DMDEDUC2, #education categories for age 20+  
 #these two are for later (survey weighting)  
 SDMVSTRA, #Strata: based on census region, metropolitan area, or population demographics  
 SDMVPSU #Primary Sampling Unit: 30 per nhanes cycle - mostly single counties, selected from strata  
 ) %>%  
 rename(RIASEX = RIAGENDR) #rename RIAGENDR to be RIASEX because NHANES asked about sex, not gender  
dim(demog\_select)

[1] 9254 9

#9254 9 - number of rows does not change  
  
############ Complete blood count dataset ############  
dim(cbc)

[1] 8366 22

#8366 22  
cbc\_select <- cbc %>%  
 select(SEQN,  
 LBXRBCSI,  
 LBXWBCSI,  
 LBDLYMNO,  
 LBDNENO)  
dim(cbc\_select)

[1] 8366 5

#8366 2  
  
############ Iron dataset ############  
dim(fe)

[1] 6401 9

#6401 9  
fe\_select <- fe %>%  
 select(SEQN,  
 LBXIRN)  
dim(fe\_select)

[1] 6401 2

#6401 2  
  
############ Arsenic dataset ############  
dim(as)

[1] 2979 4

#2979 4  
as\_select <- as %>%  
 select(SEQN,  
 URXUAS)  
dim(as\_select)

[1] 2979 2

#2979 2  
  
############ Cadmium/Lead dataset ############  
dim(cd\_pb)

[1] 8366 16

#8366 16  
cd\_pb\_select <- cd\_pb %>%  
 select(SEQN,  
 LBXBCD,  
 LBXBPB)  
dim(cd\_pb\_select)

[1] 8366 3

#8366 3  
  
############ Cotinine dataset ############  
dim(cotinine)

[1] 7936 5

#7936 5  
cotinine\_select <- cotinine %>%  
 select(SEQN,  
 LBXCOT)  
dim(cotinine\_select)

[1] 7936 2

#7936 2

## One dataset at a time, join all datasets with the demographics dataset

# Merge the demographics and complete blood count datasets  
demog\_cbc <- left\_join(demog\_select, cbc\_select, by = "SEQN")  
# Check the merge - How many participants and variables do you expect?  
str(demog\_cbc)

'data.frame': 9254 obs. of 13 variables:  
 $ SEQN : num 93703 93704 93705 93706 93707 ...  
 $ RIASEX : num 2 1 2 1 1 2 2 2 1 1 ...  
 $ RIDAGEYR: num 2 2 66 18 13 66 75 0 56 18 ...  
 $ RIDRETH1: num 5 3 4 5 5 5 4 3 5 1 ...  
 $ INDFMPIR: num 5 5 0.82 NA 1.88 1.63 0.41 4.9 5 0.76 ...  
 $ DMDEDUC3: num NA NA NA 15 6 NA NA NA NA 12 ...  
 $ DMDEDUC2: num NA NA 2 NA NA 1 4 NA 5 NA ...  
 $ SDMVSTRA: num 145 143 145 134 138 138 136 134 134 147 ...  
 $ SDMVPSU : num 2 1 2 2 1 2 1 1 2 2 ...  
 $ LBXRBCSI: num NA 4.25 5.48 5.24 5.02 4.59 5.13 NA 4.65 5.12 ...  
 $ LBXWBCSI: num NA 7.4 8.6 6.1 11.2 6 7.2 NA 5 7.1 ...  
 $ LBDLYMNO: num NA 3.5 3.4 1.5 4.2 1.9 1.9 NA 1.8 2.2 ...  
 $ LBDNENO : num NA 3.2 4.2 3.7 6.1 3.6 4.8 NA 2.7 4.2 ...

# Merge the previous dataset and iron dataset  
demog\_cbc\_fe <- left\_join(demog\_cbc, fe\_select, by = "SEQN")  
# Check the merge - How many participants and variables do you expect?  
str(demog\_cbc\_fe)

'data.frame': 9254 obs. of 14 variables:  
 $ SEQN : num 93703 93704 93705 93706 93707 ...  
 $ RIASEX : num 2 1 2 1 1 2 2 2 1 1 ...  
 $ RIDAGEYR: num 2 2 66 18 13 66 75 0 56 18 ...  
 $ RIDRETH1: num 5 3 4 5 5 5 4 3 5 1 ...  
 $ INDFMPIR: num 5 5 0.82 NA 1.88 1.63 0.41 4.9 5 0.76 ...  
 $ DMDEDUC3: num NA NA NA 15 6 NA NA NA NA 12 ...  
 $ DMDEDUC2: num NA NA 2 NA NA 1 4 NA 5 NA ...  
 $ SDMVSTRA: num 145 143 145 134 138 138 136 134 134 147 ...  
 $ SDMVPSU : num 2 1 2 2 1 2 1 1 2 2 ...  
 $ LBXRBCSI: num NA 4.25 5.48 5.24 5.02 4.59 5.13 NA 4.65 5.12 ...  
 $ LBXWBCSI: num NA 7.4 8.6 6.1 11.2 6 7.2 NA 5 7.1 ...  
 $ LBDLYMNO: num NA 3.5 3.4 1.5 4.2 1.9 1.9 NA 1.8 2.2 ...  
 $ LBDNENO : num NA 3.2 4.2 3.7 6.1 3.6 4.8 NA 2.7 4.2 ...  
 $ LBXIRN : num NA NA 92 164 91 90 63 NA 56 225 ...

# Merge the previous dataset and arsenic dataset  
demog\_cbc\_fe\_as <- left\_join(demog\_cbc\_fe, as\_select, by = "SEQN")  
# Check the merge - How many participants and variables do you expect?  
str(demog\_cbc\_fe\_as)

'data.frame': 9254 obs. of 15 variables:  
 $ SEQN : num 93703 93704 93705 93706 93707 ...  
 $ RIASEX : num 2 1 2 1 1 2 2 2 1 1 ...  
 $ RIDAGEYR: num 2 2 66 18 13 66 75 0 56 18 ...  
 $ RIDRETH1: num 5 3 4 5 5 5 4 3 5 1 ...  
 $ INDFMPIR: num 5 5 0.82 NA 1.88 1.63 0.41 4.9 5 0.76 ...  
 $ DMDEDUC3: num NA NA NA 15 6 NA NA NA NA 12 ...  
 $ DMDEDUC2: num NA NA 2 NA NA 1 4 NA 5 NA ...  
 $ SDMVSTRA: num 145 143 145 134 138 138 136 134 134 147 ...  
 $ SDMVPSU : num 2 1 2 2 1 2 1 1 2 2 ...  
 $ LBXRBCSI: num NA 4.25 5.48 5.24 5.02 4.59 5.13 NA 4.65 5.12 ...  
 $ LBXWBCSI: num NA 7.4 8.6 6.1 11.2 6 7.2 NA 5 7.1 ...  
 $ LBDLYMNO: num NA 3.5 3.4 1.5 4.2 1.9 1.9 NA 1.8 2.2 ...  
 $ LBDNENO : num NA 3.2 4.2 3.7 6.1 3.6 4.8 NA 2.7 4.2 ...  
 $ LBXIRN : num NA NA 92 164 91 90 63 NA 56 225 ...  
 $ URXUAS : num NA NA NA NA 5.09 ...

# Merge the previous dataset and cadmium/lead dataset  
demog\_cbc\_fe\_as\_cdpb <- left\_join(demog\_cbc\_fe\_as, cd\_pb\_select, by = "SEQN")  
# Check the merge - How many participants and variables do you expect?  
str(demog\_cbc\_fe\_as\_cdpb)

'data.frame': 9254 obs. of 17 variables:  
 $ SEQN : num 93703 93704 93705 93706 93707 ...  
 $ RIASEX : num 2 1 2 1 1 2 2 2 1 1 ...  
 $ RIDAGEYR: num 2 2 66 18 13 66 75 0 56 18 ...  
 $ RIDRETH1: num 5 3 4 5 5 5 4 3 5 1 ...  
 $ INDFMPIR: num 5 5 0.82 NA 1.88 1.63 0.41 4.9 5 0.76 ...  
 $ DMDEDUC3: num NA NA NA 15 6 NA NA NA NA 12 ...  
 $ DMDEDUC2: num NA NA 2 NA NA 1 4 NA 5 NA ...  
 $ SDMVSTRA: num 145 143 145 134 138 138 136 134 134 147 ...  
 $ SDMVPSU : num 2 1 2 2 1 2 1 1 2 2 ...  
 $ LBXRBCSI: num NA 4.25 5.48 5.24 5.02 4.59 5.13 NA 4.65 5.12 ...  
 $ LBXWBCSI: num NA 7.4 8.6 6.1 11.2 6 7.2 NA 5 7.1 ...  
 $ LBDLYMNO: num NA 3.5 3.4 1.5 4.2 1.9 1.9 NA 1.8 2.2 ...  
 $ LBDNENO : num NA 3.2 4.2 3.7 6.1 3.6 4.8 NA 2.7 4.2 ...  
 $ LBXIRN : num NA NA 92 164 91 90 63 NA 56 225 ...  
 $ URXUAS : num NA NA NA NA 5.09 ...  
 $ LBXBCD : num NA 0.07 0.24 0.21 0.14 0.73 1.08 NA 0.38 0.26 ...  
 $ LBXBPB : num NA NA 2.98 0.74 0.39 1.53 1.31 NA 2.15 0.27 ...

# Merge the previous dataset and cotinine dataset  
# Note, the variable label for SEQN in the cotinine dataset from NHANES has a different capitalization pattern, so R recognizes this as a different variable from the SEQN variable in all of the other datasets. Let's strip the label from that variable and then do the merge.  
demog\_cbc\_fe\_as\_cdpb <- remove\_label(demog\_cbc\_fe\_as\_cdpb, SEQN)  
cotinine\_select <- remove\_label(cotinine\_select, SEQN)  
nhanes <- left\_join(demog\_cbc\_fe\_as\_cdpb, cotinine\_select, by = "SEQN")  
# Check the merge - How many participants and variables do you expect?  
str(nhanes)

'data.frame': 9254 obs. of 18 variables:  
 $ SEQN : num 93703 93704 93705 93706 93707 ...  
 $ RIASEX : num 2 1 2 1 1 2 2 2 1 1 ...  
 $ RIDAGEYR: num 2 2 66 18 13 66 75 0 56 18 ...  
 $ RIDRETH1: num 5 3 4 5 5 5 4 3 5 1 ...  
 $ INDFMPIR: num 5 5 0.82 NA 1.88 1.63 0.41 4.9 5 0.76 ...  
 $ DMDEDUC3: num NA NA NA 15 6 NA NA NA NA 12 ...  
 $ DMDEDUC2: num NA NA 2 NA NA 1 4 NA 5 NA ...  
 $ SDMVSTRA: num 145 143 145 134 138 138 136 134 134 147 ...  
 $ SDMVPSU : num 2 1 2 2 1 2 1 1 2 2 ...  
 $ LBXRBCSI: num NA 4.25 5.48 5.24 5.02 4.59 5.13 NA 4.65 5.12 ...  
 $ LBXWBCSI: num NA 7.4 8.6 6.1 11.2 6 7.2 NA 5 7.1 ...  
 $ LBDLYMNO: num NA 3.5 3.4 1.5 4.2 1.9 1.9 NA 1.8 2.2 ...  
 $ LBDNENO : num NA 3.2 4.2 3.7 6.1 3.6 4.8 NA 2.7 4.2 ...  
 $ LBXIRN : num NA NA 92 164 91 90 63 NA 56 225 ...  
 $ URXUAS : num NA NA NA NA 5.09 ...  
 $ LBXBCD : num NA 0.07 0.24 0.21 0.14 0.73 1.08 NA 0.38 0.26 ...  
 $ LBXBPB : num NA NA 2.98 0.74 0.39 1.53 1.31 NA 2.15 0.27 ...  
 $ LBXCOT : num NA NA 0.028 0.138 0.555 0.011 54.3 NA 0.057 13.4 ...

# This is our joined dataset!

## Create a new continuous variable from other continuous variables

# Create a neutrophil:lymphocyte ratio variable (NLR)  
nhanes <- nhanes %>%  
 mutate(nlr = LBDNENO / LBDLYMNO)   
  
  
# View the new variable to look at the changes  
nhanes %>%  
 select(SEQN,  
 LBDNENO,  
 LBDLYMNO,  
 nlr) %>%  
 head()

SEQN LBDNENO LBDLYMNO nlr  
1 93703 NA NA NA  
2 93704 3.2 3.5 0.9142857  
3 93705 4.2 3.4 1.2352941  
4 93706 3.7 1.5 2.4666667  
5 93707 6.1 4.2 1.4523810  
6 93708 3.6 1.9 1.8947368

## Create a categorical variable from a continuous variable

# How many participants do you expect in each of the iron status groups?  
sum(nhanes$LBXIRN < 60, na.rm = TRUE)

[1] 1376

sum(nhanes$LBXIRN >= 60 & nhanes$LBXIRN <= 170, na.rm = TRUE)

[1] 4404

sum(nhanes$LBXIRN > 170, na.rm = TRUE)

[1] 142

# Create a categorical variable of iron status based on serum concentrations  
nhanes <- nhanes %>%  
 mutate(iron\_status = case\_when(LBXIRN < 60 ~ "Deficient",  
 LBXIRN > 170 ~ "Excessive",  
 LBXIRN >= 60 & LBXIRN <= 170 ~ "Normal")) %>%  
 mutate(iron\_status = relevel(factor(iron\_status),  
 ref = "Deficient"))  
  
table(nhanes$iron\_status) # How many participants did you get in each group?

Deficient Excessive Normal   
 1376 142 4404

# Check the structure of the dataset to see that iron\_status is after LBXIRN and it is a factor with reference = "Deficient"  
str(nhanes$iron\_status)

Factor w/ 3 levels "Deficient","Excessive",..: NA NA 3 3 3 3 3 NA 1 2 ...

## Use the cut function to slice a numeric variable into groups

# Make groups of equal range  
nhanes <- nhanes %>%  
 mutate(age\_groups = cut\_interval(RIDAGEYR, n = 5))   
  
table(nhanes$age\_groups) # How many participants did you get in each group?

[0,16] (16,32] (32,48] (48,64] (64,80]   
 3250 1519 1349 1636 1500

# Other options:  
  
# Five groups with approximately equal numbers of participants (cut\_number())  
nhanes <- nhanes %>%  
 mutate(cut\_groups = cut\_number(RIDAGEYR, n = 5))  
  
table(nhanes$cut\_groups) # How many participants did you get in each group?

[0,8] (8,20] (20,42] (42,62] (62,80]   
 1904 1857 1868 1884 1741

# Check your understanding!

Serum cotinine is a continuous variable in our dataset. Let’s use it to create a two-level categorical variable for nonsmoking versus smoking. Nonsmoking status can be defined as serum cotinine values less than or equal to 10 ng/mL (Pirkle et al., 1996). \* Make a new R code chunk \* Make a new factor variable for smoking status \* The variable should have the level “smoking” for cotinine greater than 10 \* The variable should have the level “nonsmoking” for cotinine less than or equal to 10 \* How many observations are in the nonsmoking category? In the smoking category?

## Convert sex and race/ethnicity variables from numeric to factors

# Update sex variable  
# Check initial counts and data type  
table(nhanes$RIASEX)

1 2   
4557 4697

str(nhanes$RIASEX)

num [1:9254] 2 1 2 1 1 2 2 2 1 1 ...

# Update sex from numbers to characters  
nhanes <- nhanes %>%  
 mutate(sex = case\_when(RIASEX == 1 ~ "Male",  
 RIASEX == 2 ~ "Female")) %>%  
 mutate(sex = factor(sex)) %>% #change the character variables into factor variables  
 mutate(sex = relevel(sex, #set the reference levels  
 ref = "Male"))  
  
# Check the count and data type of the new variable  
table(nhanes$sex)

Male Female   
 4557 4697

str(nhanes$sex)

Factor w/ 2 levels "Male","Female": 2 1 2 1 1 2 2 2 1 1 ...

# Do the counts still match?  
  
  
# Update race/ethnicity variable  
# Check the initial counts and data type  
table(nhanes$RIDRETH1)

1 2 3 4 5   
1367 820 3150 2115 1802

str(nhanes$RIDRETH1)

num [1:9254] 5 3 4 5 5 5 4 3 5 1 ...

# Update race/ethnicity from numbers to characters  
nhanes <- nhanes %>%  
 mutate(race\_eth = case\_when(RIDRETH1 == 1 ~ "Mexican American",  
 RIDRETH1 == 2 ~ "Other Hispanic",  
 RIDRETH1 == 3 ~ "Non-Hispanic White",  
 RIDRETH1 == 4 ~ "Non-Hispanic Black",  
 RIDRETH1 == 5 ~ "Other Race")) %>%  
 mutate(race\_eth = factor(race\_eth)) %>% #change the character variables into factor variables  
 mutate(race\_eth = relevel(race\_eth, #set the reference levels  
 ref = "Non-Hispanic White"))   
  
# Check the count and data type of the new variable  
table(nhanes$race\_eth)

Non-Hispanic White Mexican American Non-Hispanic Black Other Hispanic   
 3150 1367 2115 820   
 Other Race   
 1802

str(nhanes$race\_eth)

Factor w/ 5 levels "Non-Hispanic White",..: 5 1 3 5 5 5 3 1 5 2 ...

# Do the counts still match?

## Clean and convert the education variables from numeric to factors

* **https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DEMO\_J.htm**

# Education variable for youth   
# DMDEDUC3 - participants 6 - 19 years old  
# Check original counts of DMDEDUC3  
table(nhanes$RIDAGEYR < 20, useNA = "always") #3685 participants under 20

FALSE TRUE <NA>   
 5569 3685 0

table(nhanes$DMDEDUC3, useNA = "always") #values 0-15 and 66

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15   
 184 176 202 202 179 199 154 151 154 154 139 155 20 150 5 71   
 66 <NA>   
 11 6948

table(nhanes$DMDEDUC3 >= 0, useNA = "always") #2306 youth participants with education reported

TRUE <NA>   
2306 6948

# Update the numeric codes to categories  
nhanes <- nhanes %>%  
 mutate(education\_youth = case\_when(DMDEDUC3 >= 0 & DMDEDUC3 <= 5 ~ "Less than 5th grade",  
 DMDEDUC3 >= 6 & DMDEDUC3 <= 12 | DMDEDUC3 == 66 ~ "Less than high school",  
 DMDEDUC3 == 13 | DMDEDUC3 == 14 ~ "High school or GED",  
 DMDEDUC3 == 15 ~ "More than high school"))   
   
# Check the counts  
table(nhanes$education\_youth, useNA = "always")

High school or GED Less than 5th grade Less than high school   
 155 1142 938   
More than high school <NA>   
 71 6948

# Education variable for adults  
# DMDEDUC2 - participants 20+ years old  
# Check original counts of DMDEDUC2  
table(nhanes$RIDAGEYR >= 20, useNA = "always") #5569 participants 20+ years

FALSE TRUE <NA>   
 3685 5569 0

table(nhanes$DMDEDUC2, useNA = "always") #values 1-5, 7, 9

1 2 3 4 5 7 9 <NA>   
 479 638 1325 1778 1336 2 11 3685

table(nhanes$DMDEDUC2 >= 0, useNA = "always") #5569 adult participants with education data (13 with 7 or 9)

TRUE <NA>   
5569 3685

# Update the numeric codes to categories  
nhanes <- nhanes %>%  
 mutate(education\_adult = case\_when(DMDEDUC2 == 1 | DMDEDUC2 == 2 ~ "Less than high school",  
 DMDEDUC2 == 3 ~ "High school or GED",  
 DMDEDUC2 == 4 | DMDEDUC2 == 5 ~ "More than high school",  
 DMDEDUC2 == 7 | DMDEDUC2 == 9 ~ "Unknown")) %>%  
 mutate(education\_adult = na\_if(education\_adult, "Unknown")) #Make all the "Unknown" values missing  
  
# Check the counts  
table(nhanes$education\_adult, useNA = "always")

High school or GED Less than high school More than high school   
 1325 1117 3114   
 <NA>   
 3698

# Combine the two education variables  
nhanes <- nhanes %>%  
 mutate(education = coalesce(education\_youth,  
 education\_adult)) %>%  
 # Set the factor levels and reference for education, reorder the columns  
 mutate(education = relevel(factor(education,   
 levels = c("Less than 5th grade",  
 "Less than high school",  
 "High school or GED",  
 "More than high school")),  
 ref = "Less than high school"))  
  
# View the new columns  
nhanes %>%   
 select(education\_youth,  
 education\_adult,  
 education) %>%  
 head()

education\_youth education\_adult education  
1 <NA> <NA> <NA>  
2 <NA> <NA> <NA>  
3 <NA> Less than high school Less than high school  
4 More than high school <NA> More than high school  
5 Less than high school <NA> Less than high school  
6 <NA> Less than high school Less than high school

# Check the counts  
table(nhanes$education, useNA = "always")

Less than high school Less than 5th grade High school or GED   
 2055 1142 1480   
More than high school <NA>   
 3185 1392

# Note: setting a reference automatically makes the reference the first level

## Save dataset for the future

# Reorder the variable names for next time  
nhanes\_col <- c("SEQN", "RIASEX", "sex", "RIDAGEYR", "age\_groups", "RIDRETH1", "race\_eth", "INDFMPIR", "DMDEDUC3", "DMDEDUC2", "education\_youth", "education\_adult", "education", "SDMVSTRA", "SDMVPSU", "LBXRBCSI", "LBXWBCSI", "LBDLYMNO", "LBDNENO", "nlr", "LBXIRN", "iron\_status", "URXUAS", "LBXBCD", "LBXBPB", "LBXCOT")  
nhanes <- relocate(nhanes, all\_of(nhanes\_col))  
head(nhanes)

SEQN RIASEX sex RIDAGEYR age\_groups RIDRETH1 race\_eth INDFMPIR  
1 93703 2 Female 2 [0,16] 5 Other Race 5.00  
2 93704 1 Male 2 [0,16] 3 Non-Hispanic White 5.00  
3 93705 2 Female 66 (64,80] 4 Non-Hispanic Black 0.82  
4 93706 1 Male 18 (16,32] 5 Other Race NA  
5 93707 1 Male 13 [0,16] 5 Other Race 1.88  
6 93708 2 Female 66 (64,80] 5 Other Race 1.63  
 DMDEDUC3 DMDEDUC2 education\_youth education\_adult  
1 NA NA <NA> <NA>  
2 NA NA <NA> <NA>  
3 NA 2 <NA> Less than high school  
4 15 NA More than high school <NA>  
5 6 NA Less than high school <NA>  
6 NA 1 <NA> Less than high school  
 education SDMVSTRA SDMVPSU LBXRBCSI LBXWBCSI LBDLYMNO LBDNENO  
1 <NA> 145 2 NA NA NA NA  
2 <NA> 143 1 4.25 7.4 3.5 3.2  
3 Less than high school 145 2 5.48 8.6 3.4 4.2  
4 More than high school 134 2 5.24 6.1 1.5 3.7  
5 Less than high school 138 1 5.02 11.2 4.2 6.1  
6 Less than high school 138 2 4.59 6.0 1.9 3.6  
 nlr LBXIRN iron\_status URXUAS LBXBCD LBXBPB LBXCOT cut\_groups  
1 NA NA <NA> NA NA NA NA [0,8]  
2 0.9142857 NA <NA> NA 0.07 NA NA [0,8]  
3 1.2352941 92 Normal NA 0.24 2.98 0.028 (62,80]  
4 2.4666667 164 Normal NA 0.21 0.74 0.138 (8,20]  
5 1.4523810 91 Normal 5.09 0.14 0.39 0.555 (8,20]  
6 1.8947368 90 Normal 24.07 0.73 1.53 0.011 (62,80]

# Save dataset as a csv file  
write\_csv(nhanes, file = here("nhanes\_class\_dataset.csv"))  
# Save dataset as an R object  
save(nhanes, file = here("nhanes\_class\_dataset.rda"))  
  
# Import dataset back into R from the csv or R object  
# Only R object keeps the preset factor variables saved  
nhanes\_csv <- read\_csv(here(("nhanes\_class\_dataset.csv")))

Rows: 9254 Columns: 27  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (8): sex, age\_groups, race\_eth, education\_youth, education\_adult, educa...  
dbl (19): SEQN, RIASEX, RIDAGEYR, RIDRETH1, INDFMPIR, DMDEDUC3, DMDEDUC2, SD...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

load(here(("nhanes\_class\_dataset.rda")))

## Remember to save your R script!

## To exit R

# q()  
## if you close R, you will be asked to save your workspace image

## Click **Render** to run all of your code and generate a report