EPID674 Epidemiologic Data Analysis using R

Data Management with R

Kelly Bakulski

Last compiled on February 02, 2021

Table of Contents

## Chapter 3, Data Management with R

# Install new packages

# Install packages. In general, do this only once.  
# We have this completed for you on Cloud  
options(repos="https://cran.rstudio.com" )  
install.packages("foreign")  
install.packages("Hmisc")  
install.packages("compareGroups")  
install.packages("car")  
install.packages("epiDisplay")  
  
# To avoid installing every time: change set up in curly brackets to eval=FALSE

# Set up: Query the current R environment and load relevant packages

search() # list the packages, environments, or data frames

## [1] ".GlobalEnv" "package:stats" "package:graphics"   
## [4] "package:grDevices" "package:utils" "package:datasets"   
## [7] "package:methods" "Autoloads" "package:base"

ls() # list the objects

## character(0)

##### Load these packages for the current session  
library(foreign)  
library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

library(epiDisplay)

## Loading required package: MASS

## Loading required package: nnet

##   
## Attaching package: 'epiDisplay'

## The following object is masked from 'package:ggplot2':  
##   
## alpha

## The following object is masked from 'package:lattice':  
##   
## dotplot

library(compareGroups)  
library(car)

## Loading required package: carData

sessionInfo() # record the versions of packages used in the current session

## R version 4.0.3 (2020-10-10)  
## Platform: x86\_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 16.04.7 LTS  
##   
## Matrix products: default  
## BLAS: /usr/lib/atlas-base/atlas/libblas.so.3.0  
## LAPACK: /usr/lib/atlas-base/atlas/liblapack.so.3.0  
##   
## 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   
##   
## other attached packages:  
## [1] car\_3.0-10 carData\_3.0-4 compareGroups\_4.4.6  
## [4] epiDisplay\_3.5.0.1 nnet\_7.3-14 MASS\_7.3-53   
## [7] Hmisc\_4.4-2 ggplot2\_3.3.3 Formula\_1.2-4   
## [10] survival\_3.2-7 lattice\_0.20-41 foreign\_0.8-81   
##   
## loaded via a namespace (and not attached):  
## [1] httr\_1.4.2 tidyr\_1.1.2 viridisLite\_0.3.0   
## [4] HardyWeinberg\_1.7.1 splines\_4.0.3 assertthat\_0.2.1   
## [7] latticeExtra\_0.6-29 cellranger\_1.1.0 yaml\_2.2.1   
## [10] gdtools\_0.2.3 pillar\_1.4.7 backports\_1.2.1   
## [13] glue\_1.4.2 uuid\_0.1-4 chron\_2.3-56   
## [16] digest\_0.6.27 RColorBrewer\_1.1-2 checkmate\_2.0.0   
## [19] rvest\_0.3.6 colorspace\_2.0-0 htmltools\_0.5.1   
## [22] Matrix\_1.2-18 pkgconfig\_2.0.3 broom\_0.7.3   
## [25] haven\_2.3.1 purrr\_0.3.4 webshot\_0.5.2   
## [28] scales\_1.1.1 openxlsx\_4.2.3 jpeg\_0.1-8.1   
## [31] rio\_0.5.16 officer\_0.3.16 htmlTable\_2.1.0   
## [34] tibble\_3.0.5 generics\_0.1.0 ellipsis\_0.3.1   
## [37] withr\_2.4.0 readxl\_1.3.1 magrittr\_2.0.1   
## [40] crayon\_1.3.4 evaluate\_0.14 mice\_3.12.0   
## [43] forcats\_0.5.0 xml2\_1.3.2 truncnorm\_1.0-8   
## [46] tools\_4.0.3 data.table\_1.13.6 hms\_1.0.0   
## [49] lifecycle\_0.2.0 stringr\_1.4.0 flextable\_0.6.1   
## [52] munsell\_0.5.0 cluster\_2.1.0 writexl\_1.3.1   
## [55] zip\_2.1.1 kableExtra\_1.3.1 compiler\_4.0.3   
## [58] systemfonts\_0.3.2 rlang\_0.4.10 grid\_4.0.3   
## [61] rstudioapi\_0.13 htmlwidgets\_1.5.3 Rsolnp\_1.16   
## [64] base64enc\_0.1-3 rmarkdown\_2.6 gtable\_0.3.0   
## [67] abind\_1.4-5 curl\_4.3 DBI\_1.1.1   
## [70] R6\_2.5.0 gridExtra\_2.3 knitr\_1.30   
## [73] dplyr\_1.0.3 stringi\_1.5.3 parallel\_4.0.3   
## [76] Rcpp\_1.0.5 vctrs\_0.3.6 rpart\_4.1-15   
## [79] png\_0.1-7 tidyselect\_1.1.0 xfun\_0.20

# Specify file directories

directory <- "/cloud/project" #Class option when coding on RStudio Cloud  
  
# Three alternate options to customize and specify source data directory if programming on a personal computer (only need to do the one that works for you)  
# directory<-"M:/EPID674/Data/" # Alternate option 1: location of the parent dataset in afs space of school computer  
#directory <- "/Volumes/GoogleDrive/My Drive/Teaching/EPID674/2020\_fall/EPID674\_Week3\_Class/" # Alternate option 2: location of data on my Macbook  
# directory<-"C:/Users/bakulski/Documents/EPID674/Data/" #Alternate option 3: location of data on a personal PC

# Load data, remake useful variables

# Check the file path  
file.path(directory, "nhanes3.rda")

## [1] "/cloud/project/nhanes3.rda"

# Load the saved R data  
load(file.path(directory, "nhanes3.rda"))  
  
  
# Remake a few variables from last class if they are no longer in your environment  
sex1 <- factor(nhanes$sex, levels = c(1, 2), labels = c("male", "female"))  
AGE5b <- cut(nhanes$age, quantile(nhanes$age, c(0, .2, .4, .6, .8, 1)), include.lowest = T) # quintiles  
AGE5c <- cut(nhanes$age, breaks = c(19, 40, 50, 60, 70, 90))  
age5c <- unclass(AGE5c)

# Assign names to objects, bind columns to dataframe

# Assign the subject IDs as the rownames  
rownames(nhanes)[1:10]

## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"

nhanes$seqn[1:10]

## [1] 3 10 51 55 63 70 71 84 90 97

rownames(nhanes) <- nhanes$seqn  
rownames(nhanes)[1:10]

## [1] "3" "10" "51" "55" "63" "70" "71" "84" "90" "97"

### Bind vectors/matrices/data  
nhanes.a <- cbind(nhanes, sex1, AGE5b, age5c) # bind by column (merge)  
head(nhanes.a)

## strata seqn race sex age urban region pir psu wt\_mh sbp dbp bmi  
## 3 44 3 1 1 21 1 4 0.641 1 1735.14 120 67 25.5  
## 10 6 10 1 1 35 1 1 5.406 1 27769.56 130 82 29.4  
## 51 28 51 1 1 44 2 3 1.121 1 2730.56 130 86 44.4  
## 55 44 55 1 1 48 1 4 0.622 2 804.03 155 91 37.5  
## 63 48 63 1 2 66 1 2 NaN 2 4965.67 137 73 23.6  
## 70 9 70 2 1 63 1 2 0.702 1 2629.39 137 68 31.1  
## hematoc bpb chol trig scalc creat calc sodium potass educ smk packyrs  
## 3 43.80 5.0 268 174 9.5 1.1 70033 460 0.3099999 2 1 0.0  
## 10 46.70 2.0 225 109 9.4 1.2 70385 38591 0.5103998 3 1 0.0  
## 51 49.80 6.0 162 89 9.6 1.2 40040 3050 0.5599999 1 3 10.0  
## 55 47.70 15.5 212 479 10.6 1.0 10003 1321 0.8099999 1 3 2.0  
## 63 40.60 11.3 202 96 9.3 1.0 40021 5295 0.6499996 1 1 0.0  
## 70 37.85 7.2 186 300 8.8 1.3 80005 831 0.7099996 1 3 17.5  
## diag\_ca diag\_dm diag\_ht alc phyact med\_ht htn d\_total pmon\_int pmon\_mec  
## 3 0 0 1 1 1 0 1 0 203 203  
## 10 0 0 1 1 1 0 1 0 196 196  
## 51 0 0 1 NaN 1 0 1 0 216 215  
## 55 0 0 1 1 0 0 1 1 133 132  
## 63 0 1 0 0 1 0 0 0 185 184  
## 70 0 0 0 1 0 0 0 1 144 143  
## d\_cancer d\_cvd sex1 AGE5b age5c  
## 3 0 0 male [20,30] 1  
## 10 0 0 male (30,40] 1  
## 51 0 0 male (40,54] 2  
## 55 0 1 male (40,54] 2  
## 63 0 0 female (54,69] 4  
## 70 1 0 male (54,69] 4

# Set up a simulated dataset with unequal rows and columns, attempt to bind

nhanes.left <- nhanes[, 1:20]  
nhanes.right <- nhanes[, c(2, 21:37)]  
names(nhanes.left)

## [1] "strata" "seqn" "race" "sex" "age" "urban" "region"   
## [8] "pir" "psu" "wt\_mh" "sbp" "dbp" "bmi" "hematoc"  
## [15] "bpb" "chol" "trig" "scalc" "creat" "calc"

names(nhanes.right)

## [1] "seqn" "sodium" "potass" "educ" "smk" "packyrs"   
## [7] "diag\_ca" "diag\_dm" "diag\_ht" "alc" "phyact" "med\_ht"   
## [13] "htn" "d\_total" "pmon\_int" "pmon\_mec" "d\_cancer" "d\_cvd"

nhanes.right <- nhanes.right[-c(1:30), ]  
dim(nhanes.left)

## [1] 5074 20

dim(nhanes.right)

## [1] 5044 18

# nhanes.b<-cbind(nhanes.left, nhanes.right)  
# does not work b/c of different number of rows, must comment out for knitting  
  
## let's also shuffle nhanes3.right  
nhanes.right1 <- nhanes.right[sample(nrow(nhanes.right)), ]  
nhanes.right[1:10, 1:3]

## seqn sodium potass  
## 307 307 991 0.1400000  
## 313 313 261 0.1999999  
## 321 321 3705 0.2800000  
## 331 331 3482 0.5599999  
## 335 335 6320 0.6799998  
## 346 346 1010 0.6399999  
## 350 350 3171 0.6499996  
## 353 353 2514 0.3899999  
## 357 357 282 0.5200000  
## 358 358 12225 0.6200995

nhanes.right1[1:10, 1:3]

## seqn sodium potass  
## 45579 45579 953 7.799997e-01  
## 43864 43864 1041 3.699999e-01  
## 13493 13493 1999 7.199998e-01  
## 18846 18846 1154 2.399999e-01  
## 16566 16566 12724 3.299999e-01  
## 16707 16707 8568 1.502999e-01  
## 36489 36489 0 8.800000e+04  
## 18865 18865 37885 5.603995e-01  
## 36329 36329 25 8.500000e+00  
## 20384 20384 5769 4.499998e-01

# Use the merge() function to combine datasets, explore function options

nhanes.both <- merge(nhanes.left, nhanes.right1, by = "seqn")  
names(nhanes.both)

## [1] "seqn" "strata" "race" "sex" "age" "urban"   
## [7] "region" "pir" "psu" "wt\_mh" "sbp" "dbp"   
## [13] "bmi" "hematoc" "bpb" "chol" "trig" "scalc"   
## [19] "creat" "calc" "sodium" "potass" "educ" "smk"   
## [25] "packyrs" "diag\_ca" "diag\_dm" "diag\_ht" "alc" "phyact"   
## [31] "med\_ht" "htn" "d\_total" "pmon\_int" "pmon\_mec" "d\_cancer"  
## [37] "d\_cvd"

nhanes[1:40, c(1:3, 21:22)]

## strata seqn race sodium potass  
## 3 44 3 1 460 0.30999994  
## 10 6 10 1 38591 0.51039982  
## 51 28 51 1 3050 0.55999994  
## 55 44 55 1 1321 0.80999994  
## 63 48 63 1 5295 0.64999962  
## 70 9 70 2 831 0.70999956  
## 71 43 71 1 182 5.50000000  
## 84 14 84 2 77 0.50000000  
## 90 26 90 1 2868 0.56999969  
## 97 39 97 1 665 0.05999997  
## 112 19 112 2 2357 0.65009975  
## 115 31 115 1 1523 0.10999995  
## 116 30 116 1 697 0.45999980  
## 120 32 120 2 13818 0.08999997  
## 122 16 122 2 560 0.80999994  
## 124 33 124 1 5607 0.57019997  
## 125 4 125 1 9294 0.81009960  
## 126 40 126 1 5914 0.91999960  
## 139 17 139 2 591 0.94999981  
## 145 23 145 1 17596 0.36999989  
## 164 3 164 2 11909 0.88999987  
## 169 7 169 1 5665 0.19019997  
## 209 48 209 1 745 0.27999997  
## 211 34 211 1 3322 6.10099792  
## 215 5 215 2 1362 0.02000000  
## 218 27 218 1 18436 0.16999996  
## 221 13 221 1 1785 0.66999960  
## 262 10 262 1 5513 0.51999998  
## 275 18 275 1 8207 0.87999964  
## 303 39 303 1 87 0.66999960  
## 307 22 307 2 991 0.13999999  
## 313 28 313 1 261 0.19999993  
## 321 29 321 1 3705 0.27999997  
## 331 40 331 1 3482 0.55999994  
## 335 44 335 2 6320 0.67999983  
## 346 25 346 1 1010 0.63999987  
## 350 23 350 1 3171 0.64999962  
## 353 26 353 1 2514 0.38999987  
## 357 11 357 1 282 0.51999998  
## 358 20 358 2 12225 0.62009954

nhanes.both[1:40, c(1:3, 21:22)]

## seqn strata race sodium potass  
## 1 307 22 2 991 1.400000e-01  
## 2 313 28 1 261 1.999999e-01  
## 3 321 29 1 3705 2.800000e-01  
## 4 331 40 1 3482 5.599999e-01  
## 5 335 44 2 6320 6.799998e-01  
## 6 346 25 1 1010 6.399999e-01  
## 7 350 23 1 3171 6.499996e-01  
## 8 353 26 1 2514 3.899999e-01  
## 9 357 11 1 282 5.200000e-01  
## 10 358 20 2 12225 6.200995e-01  
## 11 359 19 1 1497 1.199999e-01  
## 12 366 8 1 47533 2.800000e-01  
## 13 371 43 1 21594 2.399999e-01  
## 14 380 1 1 52670 5.900998e-01  
## 15 388 4 2 980 5.299997e-01  
## 16 395 15 2 6214 5.400000e-01  
## 17 400 46 1 543 8.799996e-01  
## 18 404 2 1 1133 5.200000e-01  
## 19 422 27 1 20009 7.500000e-01  
## 20 449 31 1 600 8.699999e-01  
## 21 459 43 2 611 7.900000e-01  
## 22 460 21 2 1282 8.399997e-01  
## 23 465 43 1 2126 7.199998e-01  
## 24 482 7 1 1657 9.999998e-03  
## 25 488 39 1 19441 6.599998e-01  
## 26 493 33 1 0 8.230000e+04  
## 27 494 24 1 9226 2.999999e-02  
## 28 499 40 1 2425 6.299996e-01  
## 29 507 22 1 6 7.470000e+04  
## 30 522 49 1 44322 6.099997e-01  
## 31 526 46 1 824 2.800000e-01  
## 32 528 17 1 6018 9.502997e-01  
## 33 529 34 1 2550 3.299999e-01  
## 34 532 14 2 0 8.420000e+04  
## 35 537 26 1 3186 1.799999e-01  
## 36 559 39 1 314 2.600000e-01  
## 37 578 10 1 23979 5.101995e-01  
## 38 589 6 1 22673 4.403000e-01  
## 39 592 9 2 1299 2.199999e-01  
## 40 610 3 1 33575 6.499996e-01

dim(nhanes)

## [1] 5074 37

dim(nhanes.both)

## [1] 5044 37

nhanes.both1 <- merge(nhanes.left, nhanes.right1, by = "seqn", all.x = T)  
dim(nhanes.both1)

## [1] 5074 37

nhanes.both1[1:40, c(1:3, 21:22)]

## seqn strata race sodium potass  
## 1 3 44 1 NA NA  
## 2 10 6 1 NA NA  
## 3 51 28 1 NA NA  
## 4 55 44 1 NA NA  
## 5 63 48 1 NA NA  
## 6 70 9 2 NA NA  
## 7 71 43 1 NA NA  
## 8 84 14 2 NA NA  
## 9 90 26 1 NA NA  
## 10 97 39 1 NA NA  
## 11 112 19 2 NA NA  
## 12 115 31 1 NA NA  
## 13 116 30 1 NA NA  
## 14 120 32 2 NA NA  
## 15 122 16 2 NA NA  
## 16 124 33 1 NA NA  
## 17 125 4 1 NA NA  
## 18 126 40 1 NA NA  
## 19 139 17 2 NA NA  
## 20 145 23 1 NA NA  
## 21 164 3 2 NA NA  
## 22 169 7 1 NA NA  
## 23 209 48 1 NA NA  
## 24 211 34 1 NA NA  
## 25 215 5 2 NA NA  
## 26 218 27 1 NA NA  
## 27 221 13 1 NA NA  
## 28 262 10 1 NA NA  
## 29 275 18 1 NA NA  
## 30 303 39 1 NA NA  
## 31 307 22 2 991 0.1400000  
## 32 313 28 1 261 0.1999999  
## 33 321 29 1 3705 0.2800000  
## 34 331 40 1 3482 0.5599999  
## 35 335 44 2 6320 0.6799998  
## 36 346 25 1 1010 0.6399999  
## 37 350 23 1 3171 0.6499996  
## 38 353 26 1 2514 0.3899999  
## 39 357 11 1 282 0.5200000  
## 40 358 20 2 12225 0.6200995

nhanes.both2 <- merge(nhanes.right1, nhanes.left, by = "seqn", all.x = T)  
dim(nhanes.both2)

## [1] 5044 37

nhanes.both2[1:40, c(1:3, 21:22)]

## seqn sodium potass sex age  
## 1 307 991 1.400000e-01 2 76  
## 2 313 261 1.999999e-01 1 20  
## 3 321 3705 2.800000e-01 1 30  
## 4 331 3482 5.599999e-01 2 78  
## 5 335 6320 6.799998e-01 2 43  
## 6 346 1010 6.399999e-01 2 66  
## 7 350 3171 6.499996e-01 1 89  
## 8 353 2514 3.899999e-01 1 47  
## 9 357 282 5.200000e-01 2 46  
## 10 358 12225 6.200995e-01 2 22  
## 11 359 1497 1.199999e-01 1 73  
## 12 366 47533 2.800000e-01 2 42  
## 13 371 21594 2.399999e-01 1 77  
## 14 380 52670 5.900998e-01 1 36  
## 15 388 980 5.299997e-01 2 39  
## 16 395 6214 5.400000e-01 1 38  
## 17 400 543 8.799996e-01 2 30  
## 18 404 1133 5.200000e-01 2 81  
## 19 422 20009 7.500000e-01 1 53  
## 20 449 600 8.699999e-01 1 26  
## 21 459 611 7.900000e-01 1 70  
## 22 460 1282 8.399997e-01 1 22  
## 23 465 2126 7.199998e-01 2 32  
## 24 482 1657 9.999998e-03 2 87  
## 25 488 19441 6.599998e-01 2 61  
## 26 493 0 8.230000e+04 1 81  
## 27 494 9226 2.999999e-02 1 70  
## 28 499 2425 6.299996e-01 2 32  
## 29 507 6 7.470000e+04 2 47  
## 30 522 44322 6.099997e-01 2 32  
## 31 526 824 2.800000e-01 1 72  
## 32 528 6018 9.502997e-01 2 30  
## 33 529 2550 3.299999e-01 2 24  
## 34 532 0 8.420000e+04 2 35  
## 35 537 3186 1.799999e-01 1 26  
## 36 559 314 2.600000e-01 2 27  
## 37 578 23979 5.101995e-01 2 61  
## 38 589 22673 4.403000e-01 1 64  
## 39 592 1299 2.199999e-01 2 23  
## 40 610 33575 6.499996e-01 2 48

nhanes.both3 <- merge(nhanes.right1, nhanes.left, by = "seqn", all.y = T)  
dim(nhanes.both3)

## [1] 5074 37

nhanes.both3[1:40, c(1:3, 21:22)]

## seqn sodium potass sex age  
## 1 3 NA NA 1 21  
## 2 10 NA NA 1 35  
## 3 51 NA NA 1 44  
## 4 55 NA NA 1 48  
## 5 63 NA NA 2 66  
## 6 70 NA NA 1 63  
## 7 71 NA NA 2 37  
## 8 84 NA NA 2 23  
## 9 90 NA NA 1 28  
## 10 97 NA NA 2 86  
## 11 112 NA NA 2 21  
## 12 115 NA NA 2 45  
## 13 116 NA NA 1 84  
## 14 120 NA NA 1 28  
## 15 122 NA NA 1 69  
## 16 124 NA NA 2 63  
## 17 125 NA NA 1 31  
## 18 126 NA NA 2 25  
## 19 139 NA NA 1 72  
## 20 145 NA NA 1 55  
## 21 164 NA NA 2 49  
## 22 169 NA NA 2 42  
## 23 209 NA NA 1 85  
## 24 211 NA NA 1 32  
## 25 215 NA NA 1 32  
## 26 218 NA NA 1 62  
## 27 221 NA NA 2 48  
## 28 262 NA NA 2 83  
## 29 275 NA NA 1 24  
## 30 303 NA NA 1 60  
## 31 307 991 0.1400000 2 76  
## 32 313 261 0.1999999 1 20  
## 33 321 3705 0.2800000 1 30  
## 34 331 3482 0.5599999 2 78  
## 35 335 6320 0.6799998 2 43  
## 36 346 1010 0.6399999 2 66  
## 37 350 3171 0.6499996 1 89  
## 38 353 2514 0.3899999 1 47  
## 39 357 282 0.5200000 2 46  
## 40 358 12225 0.6200995 2 22

# Ordering data

nhanes[1:10, c("seqn", "age", "sex")]

## seqn age sex  
## 3 3 21 1  
## 10 10 35 1  
## 51 51 44 1  
## 55 55 48 1  
## 63 63 66 2  
## 70 70 63 1  
## 71 71 37 2  
## 84 84 23 2  
## 90 90 28 1  
## 97 97 86 2

nhanes.b <- nhanes[order(nhanes$age), ]  
nhanes.b[1:10, c("seqn", "age", "sex")]

## seqn age sex  
## 313 313 20 1  
## 878 878 20 2  
## 1594 1594 20 1  
## 3144 3144 20 1  
## 3344 3344 20 2  
## 3424 3424 20 2  
## 3591 3591 20 1  
## 4026 4026 20 1  
## 4297 4297 20 2  
## 4340 4340 20 1

nhanes.c <- nhanes[order(nhanes$age, nhanes$sex), ] # sort based on 2 variables  
nhanes.c[1:10, c("seqn", "age", "sex")]

## seqn age sex  
## 313 313 20 1  
## 1594 1594 20 1  
## 3144 3144 20 1  
## 3591 3591 20 1  
## 4026 4026 20 1  
## 4340 4340 20 1  
## 4559 4559 20 1  
## 5552 5552 20 1  
## 5585 5585 20 1  
## 5836 5836 20 1

# Create a new variable in a dataset using calculations

## Create a new variable in a dataset using calculations  
nhanes$map <- nhanes$dbp + (nhanes$sbp - nhanes$dbp) / 3  
summary(nhanes$map)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 52.67 83.33 90.67 91.77 99.00 155.00

nhanes.b$map <- with(nhanes.b, dbp + (sbp - dbp) / 3)  
summary(nhanes.b$map)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 52.67 83.33 90.67 91.77 99.00 155.00

nhanes.c <- transform(nhanes.c, map = dbp + (sbp - dbp) / 3)  
summary(nhanes.c$map)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 52.67 83.33 90.67 91.77 99.00 155.00

# Recode a variable or replace a value within a variable

# Replace  
summary(nhanes$age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 20.00 32.00 46.00 48.74 65.00 90.00

length(nhanes$age[nhanes$age==90]) #What do you expect?

## [1] 35

nhanes$age1 <- replace(nhanes$age, nhanes$age == 90, NA)  
summary(nhanes$age1) #What do you get?

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 20.00 32.00 45.00 48.46 64.00 89.00 35

#Does it match your expectations?  
  
# Recode  
# the recode function in the library 'car'  
table(nhanes$sex)

##   
## 1 2   
## 2335 2739

nhanes[c(1:20), ]

## strata seqn race sex age urban region pir psu wt\_mh sbp dbp bmi  
## 3 44 3 1 1 21 1 4 0.641 1 1735.14 120 67 25.5  
## 10 6 10 1 1 35 1 1 5.406 1 27769.56 130 82 29.4  
## 51 28 51 1 1 44 2 3 1.121 1 2730.56 130 86 44.4  
## 55 44 55 1 1 48 1 4 0.622 2 804.03 155 91 37.5  
## 63 48 63 1 2 66 1 2 NaN 2 4965.67 137 73 23.6  
## 70 9 70 2 1 63 1 2 0.702 1 2629.39 137 68 31.1  
## 71 43 71 1 2 37 1 4 NaN 1 1147.32 128 70 28.0  
## 84 14 84 2 2 23 2 3 3.340 1 2069.97 109 63 29.4  
## 90 26 90 1 1 28 2 4 0.764 1 3898.74 120 82 27.9  
## 97 39 97 1 2 86 1 3 0.761 1 583.88 143 85 26.4  
## 112 19 112 2 2 21 2 3 0.327 1 8948.72 113 75 17.9  
## 115 31 115 1 2 45 1 4 5.063 1 4809.06 103 67 24.2  
## 116 30 116 1 1 84 2 4 NaN 1 2487.07 151 46 25.5  
## 120 32 120 2 1 28 2 3 2.279 1 8256.11 112 69 23.7  
## 122 16 122 2 1 69 2 3 1.434 1 1433.31 130 65 24.0  
## 124 33 124 1 2 63 1 4 5.460 1 17775.59 119 72 23.6  
## 125 4 125 1 1 31 1 1 3.358 1 35722.39 127 76 26.0  
## 126 40 126 1 2 25 1 4 2.459 2 16999.13 84 53 18.9  
## 139 17 139 2 1 72 2 3 NaN 1 1758.20 128 86 32.0  
## 145 23 145 1 1 55 2 3 2.563 1 13427.95 137 94 25.9  
## hematoc bpb chol trig scalc creat calc sodium potass educ smk packyrs  
## 3 43.80 5.0 268 174 9.5 1.1 70033 460 0.30999994 2 1 0.0  
## 10 46.70 2.0 225 109 9.4 1.2 70385 38591 0.51039982 3 1 0.0  
## 51 49.80 6.0 162 89 9.6 1.2 40040 3050 0.55999994 1 3 10.0  
## 55 47.70 15.5 212 479 10.6 1.0 10003 1321 0.80999994 1 3 2.0  
## 63 40.60 11.3 202 96 9.3 1.0 40021 5295 0.64999962 1 1 0.0  
## 70 37.85 7.2 186 300 8.8 1.3 80005 831 0.70999956 1 3 17.5  
## 71 42.85 8.5 212 155 9.9 0.8 90024 182 5.50000000 1 3 5.0  
## 84 33.45 2.8 157 88 8.8 1.1 40035 77 0.50000000 2 1 0.0  
## 90 49.55 5.4 182 96 9.0 1.0 40048 2868 0.56999969 1 3 NaN  
## 97 NaN 4.1 194 89 9.0 1.2 30001 665 0.05999997 1 1 0.0  
## 112 38.85 1.1 221 76 9.9 0.9 20026 2357 0.65009975 2 1 0.0  
## 115 42.20 2.3 206 87 8.6 1.0 20017 1523 0.10999995 2 3 23.0  
## 116 40.10 4.7 132 173 8.2 1.4 20042 697 0.45999980 1 2 188.0  
## 120 50.00 4.7 192 98 9.7 1.2 20028 13818 0.08999997 3 3 NaN  
## 122 41.35 8.1 249 249 9.5 1.3 90004 560 0.80999994 1 2 94.0  
## 124 38.75 1.1 205 201 8.5 1.1 30058 5607 0.57019997 3 1 0.0  
## 125 46.80 5.6 226 165 9.4 1.0 40538 9294 0.81009960 2 1 0.0  
## 126 42.85 0.7 177 56 9.2 0.9 50327 5914 0.91999960 3 1 0.0  
## 139 38.75 2.8 228 124 9.5 1.1 10025 591 0.94999981 1 3 27.0  
## 145 49.95 8.6 216 162 9.1 1.0 30166 17596 0.36999989 1 2 32.0  
## diag\_ca diag\_dm diag\_ht alc phyact med\_ht htn d\_total pmon\_int pmon\_mec  
## 3 0 0 1 1 1 0 1 0 203 203  
## 10 0 0 1 1 1 0 1 0 196 196  
## 51 0 0 1 NaN 1 0 1 0 216 215  
## 55 0 0 1 1 0 0 1 1 133 132  
## 63 0 1 0 0 1 0 0 0 185 184  
## 70 0 0 0 1 0 0 0 1 144 143  
## 71 0 0 0 0 0 0 0 0 202 202  
## 84 0 0 0 0 1 0 0 0 207 206  
## 90 0 0 0 0 0 0 0 0 188 187  
## 97 0 0 1 0 0 0 1 1 116 115  
## 112 0 1 0 0 1 0 0 0 184 184  
## 115 0 0 0 0 1 0 0 0 214 213  
## 116 1 0 0 0 1 0 1 1 33 32  
## 120 0 0 0 1 1 0 0 0 217 217  
## 122 0 0 0 0 0 0 0 1 30 29  
## 124 1 0 0 1 1 0 0 0 187 186  
## 125 0 0 1 1 1 0 1 0 208 207  
## 126 0 0 0 1 1 0 0 0 190 190  
## 139 0 1 0 0 1 0 0 1 99 98  
## 145 0 0 0 0 1 0 1 0 193 193  
## d\_cancer d\_cvd map age1  
## 3 0 0 84.66667 21  
## 10 0 0 98.00000 35  
## 51 0 0 100.66667 44  
## 55 0 1 112.33333 48  
## 63 0 0 94.33333 66  
## 70 1 0 91.00000 63  
## 71 0 0 89.33333 37  
## 84 0 0 78.33333 23  
## 90 0 0 94.66667 28  
## 97 0 1 104.33333 86  
## 112 0 0 87.66667 21  
## 115 0 0 79.00000 45  
## 116 1 0 81.00000 84  
## 120 0 0 83.33333 28  
## 122 0 0 86.66667 69  
## 124 0 0 87.66667 63  
## 125 0 0 93.00000 31  
## 126 0 0 63.33333 25  
## 139 0 0 100.00000 72  
## 145 0 0 108.33333 55

# convert sex=2 to 0  
nhanes$male <- recode(nhanes$sex, "2=0; else=nhanes$sex")  
nhanes[c(1:20), c("sex", "male")]

## sex male  
## 3 1 1  
## 10 1 1  
## 51 1 1  
## 55 1 1  
## 63 2 0  
## 70 1 1  
## 71 2 0  
## 84 2 0  
## 90 1 1  
## 97 2 0  
## 112 2 0  
## 115 2 0  
## 116 1 1  
## 120 1 1  
## 122 1 1  
## 124 2 0  
## 125 1 1  
## 126 2 0  
## 139 1 1  
## 145 1 1

# Recode 65+ as 1, other as 2  
length(nhanes$age[nhanes$age>=65 & nhanes$age<=90]) #What do you expect?

## [1] 1293

nhanes$age2 <- recode(nhanes$age, "65:90=1; else=2")  
nhanes[c(1:20), c("age", "age2")]

## age age2  
## 3 21 2  
## 10 35 2  
## 51 44 2  
## 55 48 2  
## 63 66 1  
## 70 63 2  
## 71 37 2  
## 84 23 2  
## 90 28 2  
## 97 86 1  
## 112 21 2  
## 115 45 2  
## 116 84 1  
## 120 28 2  
## 122 69 1  
## 124 63 2  
## 125 31 2  
## 126 25 2  
## 139 72 1  
## 145 55 2

table(nhanes$age2) #What do you get?

##   
## 1 2   
## 1293 3781

### Note the function 'recode' in epiDisplay is different than in the car package.  
### It is similar to the replace() function  
### use package::function to overcome function masking between packages

# Do Exercise 3A

# Check function default settings for handling missing values

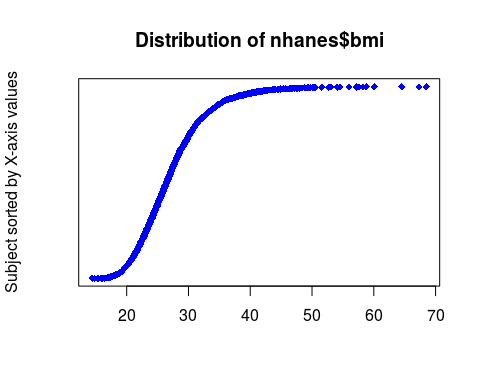
### How do functions handle missing values?  
cor(nhanes$age, nhanes$bmi)

## [1] NA

summary(nhanes$bmi)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 14.40 23.20 26.40 27.24 30.12 68.50 10

summ(nhanes$bmi)



## obs. mean median s.d. min. max.   
## 5064 27.244 26.4 5.821 14.4 68.5

# What options can you use to overcome the default missing data setting?  
cor(nhanes$age, nhanes$bmi, use = "complete")

## [1] 0.0382773

mean(nhanes$bmi)

## [1] NaN

mean(nhanes$bmi, na.rm = T)

## [1] 27.24423

mean(na.omit(nhanes$bmi))

## [1] 27.24423

# Use the na.omit() function to drop missing observations, make a new dataset

nhanes.d <- na.omit(nhanes[, c(  
 "age", "bpb", "sbp", "dbp", "bmi", "packyrs",  
 "hematoc", "educ", "smk", "alc", "med\_ht", "diag\_dm"  
)])  
nhanes.e <- na.omit(nhanes)  
dim(nhanes.d)

## [1] 4580 12

dim(nhanes.e)

## [1] 4031 41

cor(nhanes.d$age, nhanes.d$bmi)

## [1] 0.04185875

cor(nhanes.e$age, nhanes.e$bmi)

## [1] 0.05750064

# Use a logical vector and square brackets to index non-missing observations

OK <- !is.na(nhanes$age) & !is.na(nhanes$bmi)  
cor(nhanes$age[OK], nhanes$bmi[OK])

## [1] 0.0382773

cor(nhanes$age[OK == 1], nhanes$bmi[OK == 1])

## [1] 0.0382773

# Strategies to make repeated tasks in your coding more efficient

# Use the apply family of functions for more efficient coding

apply(nhanes, 2, mean)

## strata seqn race sex age urban   
## 2.326271e+01 2.708617e+04 1.283800e+00 1.539811e+00 4.874359e+01 1.516752e+00   
## region pir psu wt\_mh sbp dbp   
## 2.748325e+00 NaN 1.511037e+00 1.041630e+04 1.262637e+02 7.451892e+01   
## bmi hematoc bpb chol trig scalc   
## NaN NaN 3.958041e+00 NaN NaN NaN   
## creat calc sodium potass educ smk   
## NaN 5.085216e+04 8.121295e+03 8.546532e+02 NaN 1.750296e+00   
## packyrs diag\_ca diag\_dm diag\_ht alc phyact   
## NaN 7.449744e-02 NaN NaN NaN 7.703981e-01   
## med\_ht htn d\_total pmon\_int pmon\_mec d\_cancer   
## NaN 3.821443e-01 NaN NaN NaN NaN   
## d\_cvd map age1 male age2   
## NaN 9.176718e+01 NA 4.601892e-01 1.745171e+00

tapply(nhanes$bmi, age5c, mean)

## 1 2 3 4 5   
## NaN 28.22687 28.55475 NaN NaN

tapply(nhanes$bmi, age5c, summary)

## $`1`  
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 15.90 22.30 25.40 26.61 29.57 68.50 2   
##   
## $`2`  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 17.00 24.00 27.20 28.23 31.30 64.50   
##   
## $`3`  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 15.30 24.70 27.95 28.55 31.38 52.90   
##   
## $`4`  
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 15.30 24.70 27.70 28.12 31.10 47.20 2   
##   
## $`5`  
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 14.40 22.90 26.10 26.25 28.80 50.40 6

## similar grouped outputs can be obtained using aggregate()  
aggregate(nhanes$bmi, by = list(Sex = nhanes$sex, Race = nhanes$race), mean, na.rm = T) # stratified by sex and race

## Sex Race mean.nhanes$bmi  
## 1 1 1 26.84863  
## 2 2 1 27.04982  
## 3 1 2 26.50016  
## 4 2 2 29.10862

aggregate(nhanes$race, by = list(nhanes$sex), table, na.rm = T) # stratified by sex and race

## Group.1 table.nhanes$race.1 table.nhanes$race.2  
## 1 1 1711 624  
## 2 2 1923 816

aggregate(nhanes[, c("sbp", "dbp")], by = list(Sex = nhanes$sex, Race = nhanes$race), mean) # for multiple columns at once

## Sex Race sbp dbp  
## 1 1 1 128.6429 76.42081  
## 2 2 1 124.7275 71.72335  
## 3 1 2 128.4135 78.66987  
## 4 2 2 123.2512 73.94485

codebook(nhanes.a) # returns summary stat by variable characteristics

##   
##   
##   
## strata :   
## obs. mean median s.d. min. max.   
## 5074 23.263 22 13.486 1 49   
##   
## ==================   
## seqn :   
## obs. mean median s.d. min. max.   
## 5074 27086.17 32365.5 17087.064 3 53594   
##   
## ==================   
## race :   
## obs. mean median s.d. min. max.   
## 5074 1.284 1 0.451 1 2   
##   
## ==================   
## sex :   
## obs. mean median s.d. min. max.   
## 5074 1.54 2 0.498 1 2   
##   
## ==================   
## age :   
## obs. mean median s.d. min. max.   
## 5074 48.744 46 19.297 20 90   
##   
## ==================   
## urban :   
## obs. mean median s.d. min. max.   
## 5074 1.517 2 0.5 1 2   
##   
## ==================   
## region :   
## obs. mean median s.d. min. max.   
## 5074 2.748 3 0.954 1 4   
##   
## ==================   
## pir :   
## obs. mean median s.d. min. max.   
## 4587 2.461 2.011 1.787 0 11.889  
##   
## ==================   
## psu :   
## obs. mean median s.d. min. max.   
## 5074 1.511 2 0.5 1 2   
##   
## ==================   
## wt\_mh :   
## obs. mean median s.d. min. max.   
## 5074 10416.302 4495.92 13014.001 225.93 139744.91  
##   
## ==================   
## sbp :   
## obs. mean median s.d. min. max.   
## 5074 126.264 122 19.958 80 237   
##   
## ==================   
## dbp :   
## obs. mean median s.d. min. max.   
## 5074 74.519 74 10.569 32 134   
##   
## ==================   
## bmi :   
## obs. mean median s.d. min. max.   
## 5064 27.244 26.4 5.821 14.4 68.5   
##   
## ==================   
## hematoc :   
## obs. mean median s.d. min. max.   
## 5013 41.426 41.45 4.173 19.1 57.35   
##   
## ==================   
## bpb :   
## obs. mean median s.d. min. max.   
## 5074 3.958 3.2 3.228 0.7 48.1   
##   
## ==================   
## chol :   
## obs. mean median s.d. min. max.   
## 5016 207.21 204 45.503 83 676   
##   
## ==================   
## trig :   
## obs. mean median s.d. min. max.   
## 5006 147.213 115 122.706 22 3616   
##   
## ==================   
## scalc :   
## obs. mean median s.d. min. max.   
## 4942 9.271 9.3 0.434 7.3 13.9   
##   
## ==================   
## creat :   
## obs. mean median s.d. min. max.   
## 4942 1.09 1 0.405 0.4 13.9   
##   
## ==================   
## calc :   
## obs. mean median s.d. min. max.   
## 5074 50852.161 50049 25961.734 6 90842   
##   
## ==================   
## sodium :   
## obs. mean median s.d. min. max.   
## 5074 8121.295 3155 13206.104 0 99712   
##   
## ==================   
## potass :   
## obs. mean median s.d. min. max.   
## 5074 854.653 0.55 8176.14 0.01 97800   
##   
## ==================   
## educ :   
## obs. mean median s.d. min. max.   
## 5041 1.728 2 0.678 1 3   
##   
## ==================   
## smk :   
## obs. mean median s.d. min. max.   
## 5074 1.75 1 0.83 1 3   
##   
## ==================   
## packyrs :   
## obs. mean median s.d. min. max.   
## 4856 9.579 0 22.023 0 456   
##   
## ==================   
## diag\_ca :   
## obs. mean median s.d. min. max.   
## 5074 0.074 0 0.263 0 1   
##   
## ==================   
## diag\_dm :   
## obs. mean median s.d. min. max.   
## 5065 0.091 0 0.287 0 1   
##   
## ==================   
## diag\_ht :   
## obs. mean median s.d. min. max.   
## 5035 0.281 0 0.45 0 1   
##   
## ==================   
## alc :   
## obs. mean median s.d. min. max.   
## 4942 0.443 0 0.497 0 1   
##   
## ==================   
## phyact :   
## obs. mean median s.d. min. max.   
## 5074 0.77 1 0.421 0 1   
##   
## ==================   
## med\_ht :   
## obs. mean median s.d. min. max.   
## 5029 0.167 0 0.373 0 1   
##   
## ==================   
## htn :   
## obs. mean median s.d. min. max.   
## 5074 0.382 0 0.486 0 1   
##   
## ==================   
## d\_total :   
## obs. mean median s.d. min. max.   
## 5069 0.252 0 0.434 0 1   
##   
## ==================   
## pmon\_int :   
## obs. mean median s.d. min. max.   
## 5069 159.605 171 49.493 1 218   
##   
## ==================   
## pmon\_mec :   
## obs. mean median s.d. min. max.   
## 5069 158.7 170 49.481 0 217   
##   
## ==================   
## d\_cancer :   
## obs. mean median s.d. min. max.   
## 5069 0.05 0 0.219 0 1   
##   
## ==================   
## d\_cvd :   
## obs. mean median s.d. min. max.   
## 5069 0.114 0 0.317 0 1   
##   
## ==================   
## sex1 :   
## Frequency Percent  
## male 2335 46  
## female 2739 54  
##   
## ==================   
## AGE5b :   
## Frequency Percent  
## [20,30] 1130 22.3  
## (30,40] 962 19.0  
## (40,54] 1002 19.7  
## (54,69] 1030 20.3  
## (69,90] 950 18.7  
##   
## ==================   
## age5c :   
## obs. mean median s.d. min. max.   
## 5074 2.516 2 1.55 1 5   
##   
## ==================

summ(nhanes)

##   
## No. of observations = 5074  
##   
## Var. name obs. mean median s.d. min. max.   
## 1 strata 5074 23.26 22 13.49 1 49   
## 2 seqn 5074 27086.17 32365.5 17087.06 3 53594   
## 3 race 5074 1.28 1 0.45 1 2   
## 4 sex 5074 1.54 2 0.5 1 2   
## 5 age 5074 48.74 46 19.3 20 90   
## 6 urban 5074 1.52 2 0.5 1 2   
## 7 region 5074 2.75 3 0.95 1 4   
## 8 pir 4587 2.46 2.01 1.79 0 11.89   
## 9 psu 5074 1.51 2 0.5 1 2   
## 10 wt\_mh 5074 10416.3 4495.92 13014 225.93 139744.91  
## 11 sbp 5074 126.26 122 19.96 80 237   
## 12 dbp 5074 74.52 74 10.57 32 134   
## 13 bmi 5064 27.24 26.4 5.82 14.4 68.5   
## 14 hematoc 5013 41.43 41.45 4.17 19.1 57.35   
## 15 bpb 5074 3.96 3.2 3.23 0.7 48.1   
## 16 chol 5016 207.21 204 45.5 83 676   
## 17 trig 5006 147.21 115 122.71 22 3616   
## 18 scalc 4942 9.27 9.3 0.43 7.3 13.9   
## 19 creat 4942 1.09 1 0.4 0.4 13.9   
## 20 calc 5074 50852.16 50049 25961.73 6 90842   
## 21 sodium 5074 8121.3 3155 13206.1 0 99712   
## 22 potass 5074 854.65 0.55 8176.14 0.01 97800   
## 23 educ 5041 1.73 2 0.68 1 3   
## 24 smk 5074 1.75 1 0.83 1 3   
## 25 packyrs 4856 9.58 0 22.02 0 456   
## 26 diag\_ca 5074 0.07 0 0.26 0 1   
## 27 diag\_dm 5065 0.09 0 0.29 0 1   
## 28 diag\_ht 5035 0.28 0 0.45 0 1   
## 29 alc 4942 0.44 0 0.5 0 1   
## 30 phyact 5074 0.77 1 0.42 0 1   
## 31 med\_ht 5029 0.17 0 0.37 0 1   
## 32 htn 5074 0.38 0 0.49 0 1   
## 33 d\_total 5069 0.25 0 0.43 0 1   
## 34 pmon\_int 5069 159.6 171 49.49 1 218   
## 35 pmon\_mec 5069 158.7 170 49.48 0 217   
## 36 d\_cancer 5069 0.05 0 0.22 0 1   
## 37 d\_cvd 5069 0.11 0 0.32 0 1   
## 38 map 5074 91.77 90.67 11.89 52.67 155   
## 39 age1 5039 48.46 45 19.05 20 89   
## 40 male 5074 0.46 0 0.5 0 1   
## 41 age2 5074 1.75 2 0.44 1 2

# Try compiling results with a for loop.

# First prep the output dataset  
out <- data.frame(matrix(nrow = ncol(nhanes), ncol = 3))  
colnames(out) <- c("variable", "mean", "sd")  
out[, 1] <- colnames(nhanes)  
# Then initiate the loop  
for (i in 1:ncol(nhanes)) {  
 out[i, 2] <- mean(as.numeric(nhanes[, i]), na.rm = T)  
 out[i, 3] <- sd(as.numeric(nhanes[, i]), na.rm = T)  
}  
out

## variable mean sd  
## 1 strata 2.326271e+01 1.348595e+01  
## 2 seqn 2.708617e+04 1.708706e+04  
## 3 race 1.283800e+00 4.508853e-01  
## 4 sex 1.539811e+00 4.984617e-01  
## 5 age 4.874359e+01 1.929721e+01  
## 6 urban 1.516752e+00 4.997685e-01  
## 7 region 2.748325e+00 9.542678e-01  
## 8 pir 2.461101e+00 1.787247e+00  
## 9 psu 1.511037e+00 4.999274e-01  
## 10 wt\_mh 1.041630e+04 1.301400e+04  
## 11 sbp 1.262637e+02 1.995809e+01  
## 12 dbp 7.451892e+01 1.056872e+01  
## 13 bmi 2.724423e+01 5.821274e+00  
## 14 hematoc 4.142620e+01 4.172914e+00  
## 15 bpb 3.958041e+00 3.227670e+00  
## 16 chol 2.072103e+02 4.550287e+01  
## 17 trig 1.472131e+02 1.227059e+02  
## 18 scalc 9.270862e+00 4.344049e-01  
## 19 creat 1.089923e+00 4.045527e-01  
## 20 calc 5.085216e+04 2.596173e+04  
## 21 sodium 8.121295e+03 1.320610e+04  
## 22 potass 8.546532e+02 8.176140e+03  
## 23 educ 1.727832e+00 6.782601e-01  
## 24 smk 1.750296e+00 8.299804e-01  
## 25 packyrs 9.579335e+00 2.202303e+01  
## 26 diag\_ca 7.449744e-02 2.626046e-01  
## 27 diag\_dm 9.062192e-02 2.870990e-01  
## 28 diag\_ht 2.814300e-01 4.497414e-01  
## 29 alc 4.429381e-01 4.967835e-01  
## 30 phyact 7.703981e-01 4.206183e-01  
## 31 med\_ht 1.674289e-01 3.733955e-01  
## 32 htn 3.821443e-01 4.859594e-01  
## 33 d\_total 2.515289e-01 4.339346e-01  
## 34 pmon\_int 1.596049e+02 4.949260e+01  
## 35 pmon\_mec 1.586999e+02 4.948088e+01  
## 36 d\_cancer 5.030578e-02 2.185967e-01  
## 37 d\_cvd 1.136319e-01 3.173949e-01  
## 38 map 9.176718e+01 1.189351e+01  
## 39 age1 4.845704e+01 1.905420e+01  
## 40 male 4.601892e-01 4.984617e-01  
## 41 age2 1.745171e+00 4.358077e-01

# Generate descriptive statistics for each group

sapply(nhanes[nhanes$age >= 65, ], mean, na.rm = T) # for a subset age>=65

## strata seqn race sex age urban   
## 2.100155e+01 2.681042e+04 1.177881e+00 1.515081e+00 7.504099e+01 1.607889e+00   
## region pir psu wt\_mh sbp dbp   
## 2.654292e+00 2.461121e+00 1.511988e+00 6.852687e+03 1.426226e+02 7.352746e+01   
## bmi hematoc bpb chol trig scalc   
## 2.675984e+01 4.119926e+01 4.672235e+00 2.226495e+02 1.602522e+02 9.306789e+00   
## creat calc sodium potass educ smk   
## 1.227077e+00 5.064514e+04 5.312997e+03 1.518058e+03 1.575117e+00 1.624903e+00   
## packyrs diag\_ca diag\_dm diag\_ht alc phyact   
## 1.701887e+01 1.840681e-01 1.742835e-01 4.657854e-01 2.638332e-01 6.867749e-01   
## med\_ht htn d\_total pmon\_int pmon\_mec d\_cancer   
## 3.644860e-01 6.813612e-01 6.867749e-01 1.162606e+02 1.153983e+02 1.167827e-01   
## d\_cvd map age1 male age2   
## 3.364269e-01 9.655916e+01 7.462480e+01 4.849188e-01 1.000000e+00

tapply(nhanes$bmi, age5c, mean, na.rm = T) # stratified by age

## 1 2 3 4 5   
## 26.61368 28.22687 28.55475 28.11983 26.25337

# Try compiling/exporting results by group with compareGroups  
createTable(compareGroups(nhanes)) # for all variables

##   
## --------Summary descriptives table ---------  
##   
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
## [ALL] N   
## N=5074   
## ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯   
## strata 23.3 (13.5) 5074   
## seqn 27086 (17087) 5074   
## race 1.28 (0.45) 5074   
## sex 1.54 (0.50) 5074   
## age 48.7 (19.3) 5074   
## urban 1.52 (0.50) 5074   
## region 2.75 (0.95) 5074   
## pir 2.46 (1.79) 4587   
## psu 1.51 (0.50) 5074   
## wt\_mh 10416 (13014) 5074   
## sbp 126 (20.0) 5074   
## dbp 74.5 (10.6) 5074   
## bmi 27.2 (5.82) 5064   
## hematoc 41.4 (4.17) 5013   
## bpb 3.96 (3.23) 5074   
## chol 207 (45.5) 5016   
## trig 147 (123) 5006   
## scalc 9.27 (0.43) 4942   
## creat 1.09 (0.40) 4942   
## calc 50852 (25962) 5074   
## sodium 8121 (13206) 5074   
## potass 855 (8176) 5074   
## educ 1.73 (0.68) 5041   
## smk 1.75 (0.83) 5074   
## packyrs 9.58 (22.0) 4856   
## diag\_ca 0.07 (0.26) 5074   
## diag\_dm 0.09 (0.29) 5065   
## diag\_ht 0.28 (0.45) 5035   
## alc 0.44 (0.50) 4942   
## phyact 0.77 (0.42) 5074   
## med\_ht 0.17 (0.37) 5029   
## htn 0.38 (0.49) 5074   
## d\_total 0.25 (0.43) 5069   
## pmon\_int 160 (49.5) 5069   
## pmon\_mec 159 (49.5) 5069   
## d\_cancer 0.05 (0.22) 5069   
## d\_cvd 0.11 (0.32) 5069   
## map 91.8 (11.9) 5074   
## age1 48.5 (19.1) 5039   
## male 0.46 (0.50) 5074   
## age2 1.75 (0.44) 5074   
## ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯

bivariate.table<-createTable(compareGroups(sex1 ~ . - strata - seqn, data = nhanes.a)) # for all variables, stratified by sex

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

bivariate.table

##   
## --------Summary descriptives table by 'sex1'---------  
##   
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
## male female p.overall   
## N=2335 N=2739   
## ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯   
## race 1.27 (0.44) 1.30 (0.46) 0.015   
## sex 1.00 (0.00) 2.00 (0.00) .   
## age 49.4 (19.4) 48.1 (19.2) 0.017   
## urban 1.51 (0.50) 1.52 (0.50) 0.321   
## region 2.76 (0.95) 2.74 (0.96) 0.381   
## pir 2.67 (1.86) 2.28 (1.70) <0.001   
## psu 1.49 (0.50) 1.53 (0.50) 0.023   
## wt\_mh 10749 (14039) 10133 (12067) 0.097   
## sbp 129 (17.7) 124 (21.5) <0.001   
## dbp 77.0 (10.4) 72.4 (10.2) <0.001   
## bmi 26.8 (4.82) 27.7 (6.53) <0.001   
## hematoc 44.1 (3.42) 39.1 (3.29) 0.000   
## bpb 5.05 (3.68) 3.02 (2.43) <0.001   
## chol 204 (41.8) 210 (48.3) <0.001   
## trig 158 (142) 138 (102) <0.001   
## scalc 9.31 (0.41) 9.23 (0.45) <0.001   
## creat 1.21 (0.42) 0.98 (0.35) <0.001   
## calc 49684 (26171) 51848 (25745) 0.003   
## sodium 8252 (13771) 8010 (12706) 0.518   
## potass 1241 (10098) 525 (6058) 0.003   
## educ 1.72 (0.71) 1.73 (0.65) 0.529   
## smk 1.94 (0.81) 1.59 (0.81) <0.001   
## packyrs 15.1 (28.3) 5.00 (13.3) <0.001   
## diag\_ca 0.07 (0.26) 0.07 (0.26) 0.919   
## diag\_dm 0.08 (0.27) 0.10 (0.30) 0.007   
## diag\_ht 0.25 (0.43) 0.31 (0.46) <0.001   
## alc 0.60 (0.49) 0.31 (0.46) <0.001   
## phyact 0.82 (0.38) 0.72 (0.45) <0.001   
## med\_ht 0.15 (0.36) 0.18 (0.39) 0.001   
## htn 0.38 (0.48) 0.39 (0.49) 0.512   
## d\_total 0.29 (0.45) 0.22 (0.41) <0.001   
## pmon\_int 156 (53.6) 163 (45.5) <0.001   
## pmon\_mec 155 (53.6) 162 (45.5) <0.001   
## d\_cancer 0.06 (0.24) 0.04 (0.19) <0.001   
## d\_cvd 0.13 (0.33) 0.10 (0.30) 0.002   
## AGE5b: 0.006   
## [20,30] 512 (21.9%) 618 (22.6%)   
## (30,40] 401 (17.2%) 561 (20.5%)   
## (40,54] 456 (19.5%) 546 (19.9%)   
## (54,69] 512 (21.9%) 518 (18.9%)   
## (69,90] 454 (19.4%) 496 (18.1%)   
## age5c 2.58 (1.55) 2.46 (1.55) 0.008   
## ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯

# export2word(bivariate.table, file=paste0(directory,"bivariate\_table.docx"))

# Date and Time

# Input variable formats, convert to desired format

# create some dates to use in this exercise  
bdays <- c("11/2/1959", "1/1/1970")  
bdays

## [1] "11/2/1959" "1/1/1970"

# convert to Julian dates  
bdays.julian <- as.Date(bdays, format = "%m/%d/%Y")  
bdays.julian

## [1] "1959-11-02" "1970-01-01"

as.numeric(bdays.julian)

## [1] -3713 0

bdays.am <- format(bdays.julian, "%b %d, %Y")  
bdays.am

## [1] "Nov 02, 1959" "Jan 01, 1970"

# day of the week  
format(bdays.julian, "%a-%d%b%y")

## [1] "Mon-02Nov59" "Thu-01Jan70"

weekdays(bdays.julian)

## [1] "Monday" "Thursday"

# Calculate age as of today’s date

date.today <- Sys.Date() # Sys.time or Sys.Date: Current time/Date  
date.today

## [1] "2021-02-02"

agecomp <- (date.today - bdays.julian) / 365.25  
agecomp

## Time differences in days  
## [1] 61.25394 51.08830

agecomp2 <- trunc(as.numeric(agecomp))  
  
  
# create data frame  
bd <- data.frame(Birthday = bdays, Standard = bdays.julian, Julian = as.numeric(bdays.julian), Age = agecomp2)  
bd

## Birthday Standard Julian Age  
## 1 11/2/1959 1959-11-02 -3713 61  
## 2 1/1/1970 1970-01-01 0 51

# Paste function

# combine (paste) two or more variables that are parts of date  
day1 <- c("12", "13", "14")  
month1 <- c("07", "08", "12")  
year1 <- c("1980", "2000", "2010")  
paste(day1, month1, year1)

## [1] "12 07 1980" "13 08 2000" "14 12 2010"

as.Date(paste(day1, month1, year1), "%d %m %Y")

## [1] "1980-07-12" "2000-08-13" "2010-12-14"

# Do Exercise 3B