Problem Set 6, Winter 2022

Krista Miller

knitr::opts\_chunk$set(echo = TRUE)  
  
# Load any packages, if any, that you use as part of your answers here  
# For example:   
  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.0 v dplyr 1.0.5  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

#library(lmtest) # For lrtest()  
#library(ggplot2)  
#library("dplyr")

CONTEXT: Pew Research Center data

The data in “pew\_data.RData” comes from the Pew Research Center, an organization that conducts nationally-representative public opinion polls on a variety of political and social topics. Dr. Durso constructed this data set from the 2017 Pew Research Center Science and NewsSurvey, downloaded from <https://www.journalism.org/datasets/2018/> on 4/16/2019.

There are 224 variables in this data set, but only a subset will be used in this problem set. For this problem set, the outcome of interest will be the LIFE variable, which was presented to respondents like so:

“In general, would you say life in America today is better, worse or about the same as it was 50 years ago for people like you?”

Possible responses included:

1 = Better today

2 = Worse today

3 = About the same as it was 50 years ago

-1 = Refused

You will use the Pew data set again for these questions, but the set of variables will be different than those used in Problem Set 5. The data for this question will be stored in a new data set called “pew2”. You will need to have your directory set to where the data set is on your computer, so be sure to do that before running the code chunk below.

load("pew\_data.RData")  
pew2<-dplyr::select(dat,AGE,PPREG4,PPWORK,PPINCIMP,PPGENDER,PPETHM,IDEO,PPEDUCAT,LIFE, KNOWLEDGE,ENJOY,SNSUSE,SNSFREQ)

## Question 1 - 5 points

Like in Problem Set 5, you will conduct a complete case analysis. Missing values in R are denoted “NA”; however, not all NAs are created equal!

Two of the new variables relate to use of social media. SNSUSE asks if the participant uses social media, and SNSFREQ asks how frequently the participant uses social media. Many of the NAs in this data set come from people who responded that they did not use social media; that is, although the responses are denoted NA, these responses are not truly missing. Therefore, such respondents should be included in the complete case analysis.

Examine the output produced by the following chunk and answer the questions.

attributes(pew2$SNSUSE)

## $label  
## [1] "Do you ever use social media (such as Facebook, Twitter, or Snapchat)?"  
##   
## $format.spss  
## [1] "F4.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Refused Yes No   
## -1 1 2

table(pew2$SNSUSE, exclude = NULL) # Exclude argument allows for NAs to be displayed and counted

##   
## -1 1 2   
## 12 2755 1257

attributes(pew2$SNSFREQ)

## $label  
## [1] "And do you use social media…"  
##   
## $format.spss  
## [1] "F4.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Refused Several times a day About once a day A few times a week   
## -1 1 2 3   
## Every few weeks Less often   
## 4 5

table(pew2$SNSFREQ, exclude = NULL)

##   
## -1 1 2 3 4 5 <NA>   
## 6 1425 650 420 137 117 1269

1. How many people reported not using social media?

Your answer here: 1257

1. How many people had responses recorded as NAs for the SNSFREQ variable?

Your answer here: 1269

Now that you’ve examined the variables, recode all NAs in SNSFREQ to 6 if the participant responded “no” to the SNSUSE variable.

pew2<- pew2 %>%  
 mutate(SNSFREQ\_recoded = case\_when ((is.na(SNSFREQ)& SNSUSE==2 ~ 6),   
 SNSFREQ == -1 ~ -1,  
 SNSFREQ == 1 ~ 1,  
 SNSFREQ == 2 ~ 2,  
 SNSFREQ ==3 ~ 3,  
 SNSFREQ ==4 ~ 4,  
 SNSFREQ == 5 ~ 5))

To verify that you recoded SNSFREQ properly, display a table showing the counts of the responses to SNSFREQ. Be sure that NAs are included in the count and that they are shown in your knitted document (that’s what exclude = NULL does). Once you’ve done this, answer the question below.

table(pew2$SNSFREQ\_recoded, exclude = NULL)

##   
## -1 1 2 3 4 5 6 <NA>   
## 6 1425 650 420 137 117 1257 12

1. Does the number of 6’s in the recoded SNSFREQ variable match the number of people who reported not using social media?

Your answer here (yes or no): yes

Hint: if the answer to this question isn’t “yes”, go back and re-do the steps.

## Question 2 - 10 points

Be sure that you have completed Question 1 before starting this question, and then do the following steps *in order*:

Before you start this process, first save only the following variables to a new data set, pew.start:

LIFE SNSUSE SNSFREQ\_recoded PPREG4 PPWORK PPINCIMP PPGENDER PPETHM IDEO PPEDUCAT KNOWLEDGE ENJOY AGE

pew.start <- pew2[, c("LIFE", "SNSUSE", "SNSFREQ\_recoded", "PPREG4", "PPWORK", "PPINCIMP", "PPGENDER", "PPETHM", "IDEO", "PPEDUCAT", "KNOWLEDGE", "ENJOY", "AGE")]

First, count the number of observations (i.e., rows) in your data set (pew.start). Once you’ve done so, answer the question below this code chunk.

count(pew.start)

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 4024

1. How many rows are currently present in your data set (pew.start)?

Your answer here: 4024

Next, we need to identify missing values in our data set (pew.start). Before writing any code to drop these variables, it helps to manually inspect your data to see what values should be considered missing. The attributes() and table() functions are useful for this, and examples of their use are shown in the previous question. Along with NAs, also consider labels such as “Not asked” and “Refused” as missing. Once you’ve done so, answer the three questions below this code chunk.

# Your code for variable examination here - use all the space you need!  
attributes(pew.start$LIFE)

## $label  
## [1] "In general, would you say life in America today is better, worse or about the same as it was 50 years ago for people like you?"  
##   
## $format.spss  
## [1] "F4.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Refused Better today   
## -1 1   
## Worse today About the same as it was 50 years ago   
## 2 3

table(pew.start$LIFE, exclude = NULL)

##   
## -1 1 2 3   
## 18 1596 1900 510

attributes(pew.start$SNSUSE)

## $label  
## [1] "Do you ever use social media (such as Facebook, Twitter, or Snapchat)?"  
##   
## $format.spss  
## [1] "F4.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Refused Yes No   
## -1 1 2

table(pew.start$SNSUSE, exclude = NULL)

##   
## -1 1 2   
## 12 2755 1257

attributes(pew.start$SNSFREQ\_recoded)

## NULL

table(pew.start$SNSFREQ\_recoded, exclude = NULL)

##   
## -1 1 2 3 4 5 6 <NA>   
## 6 1425 650 420 137 117 1257 12

attributes(pew.start$PPREG4)

## $label  
## [1] "Region 4 - Based on State of Residence"  
##   
## $format.spss  
## [1] "F2.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Not asked REFUSED Northeast Midwest South West   
## -2 -1 1 2 3 4

table(pew.start$PPREG4, exclude = NULL)

##   
## 1 2 3 4   
## 738 879 1485 922

attributes(pew.start$PPWORK)

## $label  
## [1] "Current Employment Status"  
##   
## $format.spss  
## [1] "F2.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Not asked   
## -2   
## REFUSED   
## -1   
## Working - as a paid employee   
## 1   
## Working - self-employed   
## 2   
## Not working - on temporary layoff from a job   
## 3   
## Not working - looking for work   
## 4   
## Not working - retired   
## 5   
## Not working - disabled   
## 6   
## Not working - other   
## 7

table(pew.start$PPWORK, exclude = NULL)

##   
## 1 2 3 4 5 6 7   
## 2204 333 13 198 841 157 278

attributes(pew.start$PPINCIMP)

## $label  
## [1] "Household Income"  
##   
## $format.spss  
## [1] "F2.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Not asked REFUSED Less than $5,000   
## -2 -1 1   
## $5,000 to $7,499 $7,500 to $9,999 $10,000 to $12,499   
## 2 3 4   
## $12,500 to $14,999 $15,000 to $19,999 $20,000 to $24,999   
## 5 6 7   
## $25,000 to $29,999 $30,000 to $34,999 $35,000 to $39,999   
## 8 9 10   
## $40,000 to $49,999 $50,000 to $59,999 $60,000 to $74,999   
## 11 12 13   
## $75,000 to $84,999 $85,000 to $99,999 $100,000 to $124,999   
## 14 15 16   
## $125,000 to $149,999 $150,000 to $174,999 $175,000 to $199,999   
## 17 18 19   
## $200,000 to $249,999 $250,000 or more   
## 20 21

table(pew.start$PPINCIMP, exclude = NULL)

##   
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 66 31 40 91 77 104 144 183 179 167 258 321 378 285 319 486 226 253 160 125   
## 21   
## 131

attributes(pew.start$PPGENDER)

## $label  
## [1] "Gender"  
##   
## $format.spss  
## [1] "F2.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Not asked REFUSED Male Female   
## -2 -1 1 2

table(pew.start$PPGENDER, exclude = NULL)

##   
## 1 2   
## 1993 2031

attributes(pew.start$PPETHM)

## $label  
## [1] "Race / Ethnicity"  
##   
## $format.spss  
## [1] "F2.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Not asked REFUSED White, Non-Hispanic   
## -2 -1 1   
## Black, Non-Hispanic Other, Non-Hispanic Hispanic   
## 2 3 4   
## 2+ Races, Non-Hispanic   
## 5

table(pew.start$PPETHM, exclude = NULL)

##   
## 1 2 3 4 5   
## 2862 392 166 447 157

attributes(pew.start$IDEO)

## $label  
## [1] "In general, would you describe your political views as..."  
##   
## $format.spss  
## [1] "F4.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Refused Very conservative Conservative Moderate   
## -1 1 2 3   
## Liberal Very liberal   
## 4 5

table(pew.start$IDEO, exclude = NULL)

##   
## -1 1 2 3 4 5   
## 116 314 1095 1624 616 259

attributes(pew.start$PPEDUCAT)

## $label  
## [1] "Education (Categorical)"  
##   
## $format.spss  
## [1] "F2.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Not asked REFUSED   
## -2 -1   
## Less than high school High school   
## 1 2   
## Some college Bachelor's degree or higher   
## 3 4

table(pew.start$PPEDUCAT, exclude = NULL)

##   
## 1 2 3 4   
## 303 1130 1147 1444

attributes(pew.start$KNOWLEDGE)

## $label  
## [1] "How much would you say you know about science?"  
##   
## $format.spss  
## [1] "F4.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Refused A lot Some Not much Nothing at all   
## -1 1 2 3 4

table(pew.start$KNOWLEDGE, exclude = NULL)

##   
## -1 1 2 3 4   
## 13 411 2236 1174 190

attributes(pew.start$ENJOY)

## $label  
## [1] "How much would you say you enjoy following news about science compared with other kinds of news?"  
##   
## $format.spss  
## [1] "F4.0"  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## Refused A lot more than other news   
## -1 1   
## More than other news Less than other news   
## 2 3   
## A lot less than other news   
## 4

table(pew.start$ENJOY, exclude = NULL)

##   
## -1 1 2 3 4   
## 46 325 1775 1379 499

attributes(pew.start$AGE)

## $label  
## [1] "Age"  
##   
## $format.spss  
## [1] "F1.0"  
##   
## $display\_width  
## [1] 10  
##   
## $class  
## [1] "labelled"  
##   
## $labels  
## 90 years or older   
## 90

table(pew.start$AGE, exclude = NULL)

##   
## 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37   
## 43 26 38 29 38 36 38 59 62 78 83 80 63 39 54 60 46 64 77 60   
## 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57   
## 60 57 74 55 51 55 66 45 80 72 69 60 72 71 68 82 101 109 92 93   
## 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77   
## 100 127 80 81 90 73 74 64 75 78 64 71 67 53 56 62 50 35 31 25   
## 78 79 80 81 82 83 84 85 86 87 88 89 90   
## 27 24 27 23 12 14 6 13 7 1 3 3 3

How many missing values (NAs, “not asked”, or “refused”) are present for the following variables:

1. The LIFE variable? Your answer here: 18
2. The SNSUSE variable? Your answer here: 12
3. The SNSFREQ\_recoded variable? Your answer here: 18
4. The PPREG4 variable? Your answer here: 0
5. The PPWORK variable? Your answer here: 0
6. The PPINCIMP variable? Your answer here: 0
7. The PPGENDER variable? Your answer here: 0
8. The PPETHM variable? Your answer here: 0
9. The IDEO variable? Your answer here: 116
10. The PPEDUCAT variable? Your answer here: 0
11. The KNOWLEDGE variable? Your answer here: 13
12. The ENJOY variable? Your answer here: 46
13. The AGE variable? Your answer here: 0

Now that you know what values should be counted as missing, set these responses equal to “NA”.

# Your code for setting all responses that are considered missing to NA here  
  
pew.start[pew.start == -1]= NA

Once you’ve set everything that’s missing equal to NA, drop all rows that contain at least one NA.

# Your code for dropping all observations with at least one NA here  
pew.start<- pew.start %>% drop\_na()

Finally, count the number of rows again and answer the question below the code chunk. This is the final sample size for your complete cases analysis.

# Your code here to count the number of rows  
  
count(pew.start)

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 3836

1. How many rows are now present in your data set?

Your answer here: 3836

One more thing: We recoded the SNSFREQ variable to have a value of 6 if SNSUSE was missing. We could do one of two things at this point. The first thing we could do (and what we will do in this case) is leave it as it is. Because we will treat SNSFREQ as a categorical variable, the value of 6 becomes just a label for a category (i.e., just another dummy vector), which we can conceptualize of as a “never” category for social media use frequency. This wouldn’t work if it were a numeric variable; in such a case, we would want change the number placeholder back to NA to ensure that the arbitrary value isn’t used as part of estimating a coefficient for a numeric variable.

## Question 3 - 5 points

Be sure that you have completed all parts of Question 2 and have the results of all prior code chunks in memory before starting this question.

Later in this problem set, you will be computing validation and test deviances for logistic regression models. To ensure that your outcome is the proper data type for this, use the following code to recode the LIFE variable such that “Worse today” is equal to one and “Better today”/“About the same” are equal to 0.

# Run this; no additional changes needed  
  
pew.start$worse[pew.start$LIFE==1] <- 0

## Warning: Unknown or uninitialised column: `worse`.

pew.start$worse[pew.start$LIFE==3] <- 0  
pew.start$worse[pew.start$LIFE==2] <- 1

To confirm that LIFE variable was recoded correctly, examine the table showing the counts of both the original and the binarized LIFE (“worse”) variables.

# Code to display tables  
  
table(pew.start$LIFE)

##   
## 1 2 3   
## 1550 1803 483

table(pew.start$worse)

##   
## 0 1   
## 2033 1803

attributes(pew.start$LIFE)

## $label  
## [1] "In general, would you say life in America today is better, worse or about the same as it was 50 years ago for people like you?"  
##   
## $format.spss  
## [1] "F4.0"  
##   
## $labels  
## Refused Better today   
## -1 1   
## Worse today About the same as it was 50 years ago   
## 2 3   
##   
## $class  
## [1] "labelled"

1. Per the table of the original LIFE variable, how many people responded “worse today” to the this question?

Your answer here: 1803

1. Per the table of the original LIFE variable, how many people responded something other than “worse today” to the this question?

Your answer here: 2033

1. Per the table of your recoded variable (“worse”), does the number of ones in this variable match the number of people who responded “worse today” in the original LIFE variable? (Hint: if no, check in with me about it)

Your answer here (yes/no): yes

Finally, set all variables *EXCEPT pew.start$worse* (it’s already the correct type) to the correct type: - Continuous: AGE, PPINCIMP - Categorical: all others

# Per the correction, Do NOT change anything about pew.start$worse. Leave it as numeric.   
  
pew.start$age <- as.numeric(pew.start$AGE)  
pew.start$income <- as.numeric(pew.start$PPINCIMP)  
pew.start$reg4\_factor <- as.factor(pew.start$PPREG4)   
pew.start$work\_factor <- as.factor(pew.start$PPWORK)   
pew.start$gender\_factor <- as.factor(pew.start$PPGENDER)   
pew.start$eth\_factor <- as.factor(pew.start$PPETHM)   
pew.start$ideo\_factor <- as.factor(pew.start$IDEO)   
pew.start$edu\_factor <- as.factor(pew.start$PPEDUCAT)   
pew.start$know\_factor <- as.factor(pew.start$KNOWLEDGE)   
pew.start$enjoy\_factor <- as.factor(pew.start$ENJOY)   
pew.start$snsuse\_factor <- as.factor(pew.start$SNSUSE)   
pew.start$snsfreqrecode\_factor <- as.factor(pew.start$SNSFREQ\_recoded)

Check that these were typed correctly by using the str function.

str(pew.start)

## tibble [3,836 x 26] (S3: tbl\_df/tbl/data.frame)  
## $ LIFE : 'labelled' num [1:3836] 2 2 1 2 2 1 2 1 2 2 ...  
## ..- attr(\*, "label")= chr "In general, would you say life in America today is better, worse or about the same as it was 50 years ago for people like you?"  
## ..- attr(\*, "format.spss")= chr "F4.0"  
## ..- attr(\*, "labels")= Named num [1:4] -1 1 2 3  
## .. ..- attr(\*, "names")= chr [1:4] "Refused" "Better today" "Worse today" "About the same as it was 50 years ago"  
## $ SNSUSE : 'labelled' num [1:3836] 1 2 1 2 1 1 1 1 2 1 ...  
## ..- attr(\*, "label")= chr "Do you ever use social media (such as Facebook, Twitter, or Snapchat)?"  
## ..- attr(\*, "format.spss")= chr "F4.0"  
## ..- attr(\*, "labels")= Named num [1:3] -1 1 2  
## .. ..- attr(\*, "names")= chr [1:3] "Refused" "Yes" "No"  
## $ SNSFREQ\_recoded : num [1:3836] 1 6 2 6 1 1 2 1 6 1 ...  
## $ PPREG4 : 'labelled' num [1:3836] 4 1 3 3 4 3 1 3 1 1 ...  
## ..- attr(\*, "label")= chr "Region 4 - Based on State of Residence"  
## ..- attr(\*, "format.spss")= chr "F2.0"  
## ..- attr(\*, "labels")= Named num [1:6] -2 -1 1 2 3 4  
## .. ..- attr(\*, "names")= chr [1:6] "Not asked" "REFUSED" "Northeast" "Midwest" ...  
## $ PPWORK : 'labelled' num [1:3836] 1 1 5 1 2 4 1 2 5 1 ...  
## ..- attr(\*, "label")= chr "Current Employment Status"  
## ..- attr(\*, "format.spss")= chr "F2.0"  
## ..- attr(\*, "labels")= Named num [1:9] -2 -1 1 2 3 4 5 6 7  
## .. ..- attr(\*, "names")= chr [1:9] "Not asked" "REFUSED" "Working - as a paid employee" "Working - self-employed" ...  
## $ PPINCIMP : 'labelled' num [1:3836] 16 19 12 12 21 18 19 16 7 10 ...  
## ..- attr(\*, "label")= chr "Household Income"  
## ..- attr(\*, "format.spss")= chr "F2.0"  
## ..- attr(\*, "labels")= Named num [1:23] -2 -1 1 2 3 4 5 6 7 8 ...  
## .. ..- attr(\*, "names")= chr [1:23] "Not asked" "REFUSED" "Less than $5,000" "$5,000 to $7,499" ...  
## $ PPGENDER : 'labelled' num [1:3836] 1 2 1 1 1 1 2 2 2 2 ...  
## ..- attr(\*, "label")= chr "Gender"  
## ..- attr(\*, "format.spss")= chr "F2.0"  
## ..- attr(\*, "labels")= Named num [1:4] -2 -1 1 2  
## .. ..- attr(\*, "names")= chr [1:4] "Not asked" "REFUSED" "Male" "Female"  
## $ PPETHM : 'labelled' num [1:3836] 1 2 4 4 1 5 1 5 1 1 ...  
## ..- attr(\*, "label")= chr "Race / Ethnicity"  
## ..- attr(\*, "format.spss")= chr "F2.0"  
## ..- attr(\*, "labels")= Named num [1:7] -2 -1 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:7] "Not asked" "REFUSED" "White, Non-Hispanic" "Black, Non-Hispanic" ...  
## $ IDEO : 'labelled' num [1:3836] 1 3 2 3 2 3 2 3 3 2 ...  
## ..- attr(\*, "label")= chr "In general, would you describe your political views as..."  
## ..- attr(\*, "format.spss")= chr "F4.0"  
## ..- attr(\*, "labels")= Named num [1:6] -1 1 2 3 4 5  
## .. ..- attr(\*, "names")= chr [1:6] "Refused" "Very conservative" "Conservative" "Moderate" ...  
## $ PPEDUCAT : 'labelled' num [1:3836] 4 4 2 1 3 3 4 4 2 4 ...  
## ..- attr(\*, "label")= chr "Education (Categorical)"  
## ..- attr(\*, "format.spss")= chr "F2.0"  
## ..- attr(\*, "labels")= Named num [1:6] -2 -1 1 2 3 4  
## .. ..- attr(\*, "names")= chr [1:6] "Not asked" "REFUSED" "Less than high school" "High school" ...  
## $ KNOWLEDGE : 'labelled' num [1:3836] 2 3 2 3 1 3 2 2 3 1 ...  
## ..- attr(\*, "label")= chr "How much would you say you know about science?"  
## ..- attr(\*, "format.spss")= chr "F4.0"  
## ..- attr(\*, "labels")= Named num [1:5] -1 1 2 3 4  
## .. ..- attr(\*, "names")= chr [1:5] "Refused" "A lot" "Some" "Not much" ...  
## $ ENJOY : 'labelled' num [1:3836] 2 4 1 3 1 2 3 1 3 1 ...  
## ..- attr(\*, "label")= chr "How much would you say you enjoy following news about science compared with other kinds of news?"  
## ..- attr(\*, "format.spss")= chr "F4.0"  
## ..- attr(\*, "labels")= Named num [1:5] -1 1 2 3 4  
## .. ..- attr(\*, "names")= chr [1:5] "Refused" "A lot more than other news" "More than other news" "Less than other news" ...  
## $ AGE : 'labelled' num [1:3836] 64 32 58 46 34 23 26 29 68 26 ...  
## ..- attr(\*, "label")= chr "Age"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 10  
## ..- attr(\*, "labels")= Named num 90  
## .. ..- attr(\*, "names")= chr "90 years or older"  
## $ worse : num [1:3836] 1 1 0 1 1 0 1 0 1 1 ...  
## $ age : num [1:3836] 64 32 58 46 34 23 26 29 68 26 ...  
## $ income : num [1:3836] 16 19 12 12 21 18 19 16 7 10 ...  
## $ reg4\_factor : Factor w/ 4 levels "1","2","3","4": 4 1 3 3 4 3 1 3 1 1 ...  
## $ work\_factor : Factor w/ 7 levels "1","2","3","4",..: 1 1 5 1 2 4 1 2 5 1 ...  
## $ gender\_factor : Factor w/ 2 levels "1","2": 1 2 1 1 1 1 2 2 2 2 ...  
## $ eth\_factor : Factor w/ 5 levels "1","2","3","4",..: 1 2 4 4 1 5 1 5 1 1 ...  
## $ ideo\_factor : Factor w/ 5 levels "1","2","3","4",..: 1 3 2 3 2 3 2 3 3 2 ...  
## $ edu\_factor : Factor w/ 4 levels "1","2","3","4": 4 4 2 1 3 3 4 4 2 4 ...  
## $ know\_factor : Factor w/ 4 levels "1","2","3","4": 2 3 2 3 1 3 2 2 3 1 ...  
## $ enjoy\_factor : Factor w/ 4 levels "1","2","3","4": 2 4 1 3 1 2 3 1 3 1 ...  
## $ snsuse\_factor : Factor w/ 2 levels "1","2": 1 2 1 2 1 1 1 1 2 1 ...  
## $ snsfreqrecode\_factor: Factor w/ 6 levels "1","2","3","4",..: 1 6 2 6 1 1 2 1 6 1 ...

## Question 4 - 5 points

The first step of the train-validate-test process is to split the data into training, validation, and test sets. To make this easier, first create a new data set that contains only the variables that will be used in the analysis:

worse age income reg4\_factor  
work\_factor gender\_factor  
eth\_factor ideo\_factor edu\_factor know\_factor  
enjoy\_factor snsuse\_factor snsfreqrecode\_factor

pew3 <- pew.start[, c("worse", "age", "income", "reg4\_factor", "work\_factor", "gender\_factor", "eth\_factor", "ideo\_factor", "edu\_factor", "know\_factor", "enjoy\_factor", "snsuse\_factor", "snsfreqrecode\_factor")]  
   
str(pew3)

## tibble [3,836 x 13] (S3: tbl\_df/tbl/data.frame)  
## $ worse : num [1:3836] 1 1 0 1 1 0 1 0 1 1 ...  
## $ age : num [1:3836] 64 32 58 46 34 23 26 29 68 26 ...  
## $ income : num [1:3836] 16 19 12 12 21 18 19 16 7 10 ...  
## $ reg4\_factor : Factor w/ 4 levels "1","2","3","4": 4 1 3 3 4 3 1 3 1 1 ...  
## $ work\_factor : Factor w/ 7 levels "1","2","3","4",..: 1 1 5 1 2 4 1 2 5 1 ...  
## $ gender\_factor : Factor w/ 2 levels "1","2": 1 2 1 1 1 1 2 2 2 2 ...  
## $ eth\_factor : Factor w/ 5 levels "1","2","3","4",..: 1 2 4 4 1 5 1 5 1 1 ...  
## $ ideo\_factor : Factor w/ 5 levels "1","2","3","4",..: 1 3 2 3 2 3 2 3 3 2 ...  
## $ edu\_factor : Factor w/ 4 levels "1","2","3","4": 4 4 2 1 3 3 4 4 2 4 ...  
## $ know\_factor : Factor w/ 4 levels "1","2","3","4": 2 3 2 3 1 3 2 2 3 1 ...  
## $ enjoy\_factor : Factor w/ 4 levels "1","2","3","4": 2 4 1 3 1 2 3 1 3 1 ...  
## $ snsuse\_factor : Factor w/ 2 levels "1","2": 1 2 1 2 1 1 1 1 2 1 ...  
## $ snsfreqrecode\_factor: Factor w/ 6 levels "1","2","3","4",..: 1 6 2 6 1 1 2 1 6 1 ...

Saving the number of rows in this new data set will be useful, so run the following code chunk to do so.

n <- nrow(pew3)

In the async material, the following line of code was provided to help create the split:

tvt2 <- sample(rep(0:2,c(round(n*.2),round(n*.2),n-2*round(n*.2))),n)

To help you understand what’s going on here before you use it, have a look at what’s produced by what’s in the inner rep() function by running the code chunk below.

Sixty.twenty.twenty <- rep(0:2,c(round(n\*.2),round(n\*.2),n-2\*round(n\*.2)))  
table(Sixty.twenty.twenty)

## Sixty.twenty.twenty  
## 0 1 2   
## 767 767 2302

Seventy.fifteen.fifteen <- rep(0:2,c(round(n\*.15),round(n\*.15),n-2\*round(n\*.15)))  
table(Seventy.fifteen.fifteen)

## Seventy.fifteen.fifteen  
## 0 1 2   
## 575 575 2686

Eighty.ten.ten <- rep(0:2,c(round(n\*.10),round(n\*.10),n-2\*round(n\*.10)))  
table(Eighty.ten.ten)

## Eighty.ten.ten  
## 0 1 2   
## 384 384 3068

Ninety.five.five <- rep(0:2,c(round(n\*.05),round(n\*.05),n-2\*round(n\*.05)))  
table(Ninety.five.five)

## Ninety.five.five  
## 0 1 2   
## 192 192 3452

1. Which value/s in these tables (0, 1, or 2) correspond to the portion of sample that will be assigned to the training set?

Your answer here: 2

1. Which value/s in these tables (0, 1, or 2) correspond to the portion of sample that will be assigned to the validation and test sets, respectively?

Your answer here: 0, 1

Split your data set into training, validation, and test sets. Use the following proportions: 70% training, 15% validation, and 15% test.

When splitting data into training/validation/test data sets, it’s good practice to set a random seed to create a split that’s reproducible (i.e., recoverable later). For this question, use the seed provided. To ensure that your answers match, be sure to run the set.seed() line immediately before your completed tvt2 line.

set.seed(123456)   
  
tvt2 <- rep(0:2,c(round(n\*.15),round(n\*.15),n-2\*round(n\*.15)))  
  
dat.train<-pew3[tvt2==2,]   
dat.valid<-pew3[tvt2==1,]   
dat.test<-pew3[tvt2==0,]   
  
nrow(dat.train)

## [1] 2686

nrow(dat.valid)

## [1] 575

nrow(dat.test  
 )

## [1] 575

1. How many rows are in the dat.train data set?

Your answer here: 2686

1. How many rows are in the dat.valid data set?

Your answer here: 575

1. How many rows are in the dat.test data set?

Your answer here: 575

## Question 5 - 5 points

For this problem set, you’ll generate a set of candidate models to test by using forward selection to fit a series of logistic regression models using the binarization of LIFE variable (“worse”) as the outcome and all other variables in the pew3 data set as potential predictors.

Step 1: Conduct a forward selection using the training data set. Use “worse” as the outcome and all other variables as the potential predictors. Be sure trace=1 is included in the step() function.

# Code for your forward selection here  
glm.null<- glm(worse~1, data= dat.train)  
all <- glm(worse ~ ., data=dat.train)  
  
# Code for your forward selection goes here  
  
forward.model <- step(glm.null, direction='forward', scope=formula(all), family= "binomial", trace=1)

## Start: AIC=3891.66  
## worse ~ 1  
##   
## Df Deviance AIC  
## + edu\_factor 3 652.42 3831.5  
## + income 1 656.62 3844.8  
## + ideo\_factor 4 661.14 3869.2  
## + enjoy\_factor 3 661.86 3870.1  
## + know\_factor 3 661.88 3870.2  
## + gender\_factor 1 664.53 3876.9  
## + snsuse\_factor 1 666.54 3885.1  
## <none> 668.68 3891.7  
## + snsfreqrecode\_factor 5 666.40 3892.5  
## + eth\_factor 4 667.06 3893.1  
## + age 1 668.60 3893.3  
## + work\_factor 6 666.58 3895.2  
## + reg4\_factor 3 668.40 3896.5  
##   
## Step: AIC=3831.51  
## worse ~ edu\_factor  
##   
## Df Deviance AIC  
## + income 1 647.53 3813.3  
## + ideo\_factor 4 646.77 3816.2  
## + gender\_factor 1 648.64 3817.9  
## + enjoy\_factor 3 648.13 3819.8  
## + snsuse\_factor 1 650.11 3824.0  
## + know\_factor 3 649.73 3826.4  
## + eth\_factor 4 650.11 3830.0  
## + snsfreqrecode\_factor 5 649.95 3831.3  
## <none> 652.42 3831.5  
## + age 1 652.36 3833.3  
## + reg4\_factor 3 652.03 3835.9  
## + work\_factor 6 651.32 3839.0  
##   
## Step: AIC=3813.33  
## worse ~ edu\_factor + income  
##   
## Df Deviance AIC  
## + ideo\_factor 4 641.69 3797.0  
## + gender\_factor 1 644.46 3802.6  
## + enjoy\_factor 3 643.63 3803.1  
## + snsuse\_factor 1 645.47 3806.8  
## + eth\_factor 4 644.18 3807.4  
## + know\_factor 3 645.21 3809.7  
## <none> 647.53 3813.3  
## + snsfreqrecode\_factor 5 645.31 3814.1  
## + age 1 647.51 3815.2  
## + reg4\_factor 3 646.99 3817.1  
## + work\_factor 6 646.76 3822.1  
##   
## Step: AIC=3796.98  
## worse ~ edu\_factor + income + ideo\_factor  
##   
## Df Deviance AIC  
## + gender\_factor 1 638.02 3783.6  
## + enjoy\_factor 3 638.14 3788.1  
## + snsuse\_factor 1 639.34 3789.1  
## + know\_factor 3 639.35 3793.2  
## + eth\_factor 4 639.10 3794.1  
## + snsfreqrecode\_factor 5 639.19 3796.5  
## <none> 641.69 3797.0  
## + age 1 641.57 3798.5  
## + reg4\_factor 3 640.98 3800.0  
## + work\_factor 6 640.65 3804.6  
##   
## Step: AIC=3783.58  
## worse ~ edu\_factor + income + ideo\_factor + gender\_factor  
##   
## Df Deviance AIC  
## + enjoy\_factor 3 634.96 3776.6  
## + snsuse\_factor 1 636.51 3779.2  
## + eth\_factor 4 635.52 3781.0  
## + know\_factor 3 636.42 3782.8  
## <none> 638.02 3783.6  
## + age 1 637.92 3785.2  
## + snsfreqrecode\_factor 5 636.30 3786.3  
## + reg4\_factor 3 637.32 3786.6  
## + work\_factor 6 636.37 3788.6  
##   
## Step: AIC=3776.65  
## worse ~ edu\_factor + income + ideo\_factor + gender\_factor + enjoy\_factor  
##   
## Df Deviance AIC  
## + snsuse\_factor 1 633.35 3771.8  
## + eth\_factor 4 631.97 3772.0  
## <none> 634.96 3776.6  
## + age 1 634.78 3777.9  
## + know\_factor 3 634.08 3778.9  
## + snsfreqrecode\_factor 5 633.15 3779.0  
## + reg4\_factor 3 634.23 3779.6  
## + work\_factor 6 633.37 3781.9  
##   
## Step: AIC=3771.84  
## worse ~ edu\_factor + income + ideo\_factor + gender\_factor + enjoy\_factor +   
## snsuse\_factor  
##   
## Df Deviance AIC  
## + eth\_factor 4 630.43 3767.4  
## <none> 633.35 3771.8  
## + age 1 633.33 3773.8  
## + know\_factor 3 632.46 3774.1  
## + reg4\_factor 3 632.63 3774.8  
## + work\_factor 6 631.88 3777.6  
## + snsfreqrecode\_factor 4 633.15 3779.0  
##   
## Step: AIC=3767.43  
## worse ~ edu\_factor + income + ideo\_factor + gender\_factor + enjoy\_factor +   
## snsuse\_factor + eth\_factor  
##   
## Df Deviance AIC  
## <none> 630.43 3767.4  
## + know\_factor 3 629.35 3768.8  
## + age 1 630.35 3769.1  
## + reg4\_factor 3 630.02 3771.7  
## + work\_factor 6 628.76 3772.3  
## + snsfreqrecode\_factor 4 630.15 3774.2

Each “step” of the forward selection process will be used as a candidate model. For example, if your forward selection process terminated with a five-predictor model, you’ll use the one-predictor model, the two-predictor model, the three-predictor model, the four-predictor model, and the five-predictor model as the set of candidate models to test against the validation data set.

Step 2: Save each of the candidate models from your forward selection as model objects. You will need these model objects for the next step in the process

# Use this naming convention to save the model objects for as many additional models that were a step in the forward selection process. For example, if you ended up with seven predictors, make new lines for model.6 and model.7  
   
model.1 <- glm(worse ~ edu\_factor, data = dat.train, family = "binomial")  
   
model.2 <- glm(worse ~ edu\_factor + income, data = dat.train, family = "binomial")  
   
model.3 <- glm(worse ~ edu\_factor + income + ideo\_factor, data = dat.train, family = "binomial")  
   
model.4 <- glm(worse ~ edu\_factor + income + ideo\_factor + gender\_factor,data = dat.train, family = "binomial")  
   
model.5 <- glm(worse ~ edu\_factor + income + ideo\_factor + gender\_factor + enjoy\_factor, data = dat.train, family = "binomial")  
  
model.6 <- glm(worse ~ edu\_factor + income + ideo\_factor + gender\_factor + enjoy\_factor + snsuse\_factor, data = dat.train, family = "binomial")  
  
model.7 <- glm(worse ~ edu\_factor + income + ideo\_factor + gender\_factor + enjoy\_factor + snsuse\_factor +eth\_factor, data = dat.train, family = "binomial")

## Question 6 - 10 points

To test the candidate models against the validation and test sets, you’ll use the model deviances. There is a provided function that is good for this in the async material in 5.2.1 (backward\_train\_validate\_test\_5\_2\_1, lines 112-116). For your convenience, here is the function that was given to you:

valid.dev<-function(m.pred, dat.this){ pred.m<-predict(m.pred,dat.this, type=“response”) -2*sum(dat.thischd)*log(1-pred.m)) }

This function needs to be adapted to this data set. Specifically, you need to change two things. Copy and paste this function into the code chunk below and make the two changes that will make this function usable for this data set.

# Copy and paste the valid.dev function here and make the two necessary changes  
  
valid.dev<-function(m.pred, dat.this){  
 pred.m<-predict(m.pred,dat.this, type="response")  
-2\*sum(dat.this$worse\*log(pred.m)+(1-dat.this$worse)\*log(1-pred.m))  
}  
  
# Be sure to run this after you've made your changes so you have the user-defined function in memory for the next code chunk

You’re now ready for Step 3 of this process: Use the adapted function to compute the deviances of each of the candidate models when applied to the validation data set. *Because the outcome (worse) is already numeric, no change to the outcome variable is necessary*

# Your code for computing the validation-set deviances of each of the candidate models.   
  
dev.1 <- valid.dev(model.1, dat.valid)  
dev.2 <- valid.dev(model.2, dat.valid)  
dev.3 <- valid.dev(model.3, dat.valid)  
dev.4 <- valid.dev(model.4, dat.valid)  
dev.5 <- valid.dev(model.5, dat.valid)  
dev.6 <- valid.dev(model.6, dat.valid)  
dev.7 <- valid.dev(model.7, dat.valid)  
  
# Add as many of these as model objects you created in Question 5 (e.g., if you had seven models in the previous question, add dev.6 and dev.7)

Once you’ve computed the validation deviances, display the validation-set deviances for each model (that is, show the value of the deviance for each of the models). Make sure that these are visible in your knitted document. After doing this, answer the following four questions.

print(c("1-predictor model validation deviance",dev.1))

## [1] "1-predictor model validation deviance"  
## [2] "782.803056298385"

print(c("2-predictor model validation deviance",dev.2))

## [1] "2-predictor model validation deviance"  
## [2] "776.044802074403"

print(c("3-predictor model validation deviance",dev.3))

## [1] "3-predictor model validation deviance"  
## [2] "776.685969897544"

print(c("4-predictor model validation deviance",dev.4))

## [1] "4-predictor model validation deviance"  
## [2] "777.936245302335"

print(c("5-predictor model validation deviance",dev.5))

## [1] "5-predictor model validation deviance"  
## [2] "778.066883790594"

print(c("6-predictor model validation deviance",dev.6))

## [1] "6-predictor model validation deviance"  
## [2] "782.822340715666"

print(c("7-predictor model validation deviance",dev.7))

## [1] "7-predictor model validation deviance"  
## [2] "783.936338410039"

# Add more print statements if you had more than five validation deviances (e.g., print(c("6-predictor model validation deviance",dev.6)))

1. What is the validation deviance of the single-predictor model (intercept + first chosen predictor in the forward selection)?

Your answer here: 782.80

1. What is the validation deviance of the model with the most predictors (i.e., the model chosen by forward selection)?

Your answer here: 783.93

1. Which of the models out of all the candidate models had the lowest validation deviance?

Your answer here: model.2

1. Based on the validation deviances you computed, which model do you choose based on the results you obtained?

Your answer here: model.2

## Question 7 - 10 points

Now that you’ve chosen a candidate model based on its performance on the validation data set, you’ll now do the final step in the process: test that model by computing the deviance of this model when applied to the test data set.

Use the adapted deviance function to compute the deviances of the chosen model when applied to the test set.

test.dev <- valid.dev(model.2, dat.test)  
  
test.dev

## [1] 785.1264

1. What is the deviance of the chosen model when applied to the test set?

Your answer here: 785.12

To further examine how well the model performed when applied to the test data set, construct a confusion matrix comparing the actual 0/1 values of “worse” from the test set vs the predicted 0/1 values of “worse” when generated by the chosen model when applied to the test set. For this question, do so manually (i.e., using the table() function) and not by using a package to do it for you. Construct your confusion matrix such that the rows and columns are labeled; that is, it should be clear what the rows and columns represent without reading your code. Once you’ve done that, answer the four questions below.

# Code for your confusion matrix here  
probs.logreg <- predict(model.2, dat.test, type="response")  
preds.logreg <- probs.logreg >= 0.5  
  
confusion.matrix <- table(Actual = dat.test$worse, Predicted = preds.logreg)  
confusion.matrix

## Predicted  
## Actual FALSE TRUE  
## 0 188 118  
## 1 149 120

# Be sure that your confusion matrix is visible in your knitted document!

1. How many true positives did your model produce?

Your answer here: 120

1. How many true negatives did your model produce?

Your answer here: 188

1. How many false positives did your model produce?

Your answer here:118

1. How many false negatives did your model produce?

Your answer here:149

Now that you’ve constructed your confusion matrix, use it to compute the four indices of model fit that we dicussed.

# Code to compute accuracy  
accuracy <- sum(diag(confusion.matrix))/sum(confusion.matrix)  
accuracy

## [1] 0.5356522

# Code to compute precision  
precision<- confusion.matrix[2,2]/sum(confusion.matrix[,2])  
precision

## [1] 0.5042017

# Code to compute recall  
recall<- confusion.matrix[2,2]/sum(confusion.matrix[2,])  
recall

## [1] 0.4460967

# Code to compute F1 score  
F1 = 2\*(precision\*recall)/(precision+recall)  
F1

## [1] 0.4733728

1. What is the *accuracy* of this model?

Your answer here: 0.535

1. What is the *precision* of this model?

Your answer here: 0.504

1. What is the *recall* of this model?

Your answer here: 0.446

1. What is the *F1 score* of this model?

Your answer here: 0.473