Problem Set 6, Fall 2021

Ben Karabinus

```
knitr::opts_chunk$set(echo = TRUE)

# Load necessary libraries

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.
3.1 ──

## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.4     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.3     ✓ stringr 1.4.0
## ✓ readr   2.0.1     ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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,PPEDU
CAT,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 di
splayed 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
```

A) How many people reported not using social media?

Your answer here:

1257

B) 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.

```
# create new column using dplyr
pew2 <- pew2 %>%
  mutate(SNSFREQ_recoded = ifelse(SNSUSE == 2 & is.na(SNSFREQ),
                                  6 , SNSFREQ))
```

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.

```
# original column
table(pew2$SNSFREQ, exclude = NULL)
```

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

```
# transformed column
table(pew2$SNSFREQ_recoded, exclude = NULL)

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

C) 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

```
# create new data set
pew.start <- dplyr::select(pew2, 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.

```
# print the number of rows in pew.start
nrow(pew.start)

## [1] 4024
```

A) 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.

```
# check the LIFE Variable
attributes(pew.start$LIFE)
```

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

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

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

```
# check the SNSUSE variable
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

# check the SNSFRQ_recoded variable
#attributes(pew.start$SNSFREQ)
table(pew.start$SNSFREQ_recoded, exclude = NULL)

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

# check the PPREG4 Variable
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

# check the PPPWORK variable
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

# check the PPINCIMP variable
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  1
9  20
##  66  31  40  91  77 104 144 183 179 167 258 321 378 285 319 486 226 253 16
0 125
##  21
## 131
```

```
# check the PPGENDER variable
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
```

```
# Check the PPETHM variable
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
```

```
# check the IDEO variable
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
```

```
# check the PPEDUCAT variable
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 
```

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

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

```r
# check the KNOWLEDGE variable
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
```

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

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

```r
# check the ENJOY variable
attributes(pew.start$ENJOY)
```

```
## $label
## [1] "How much would you say you enjoy following news about science compare
d 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

# check the AGE variable
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  3
6  37 
##  43  26  38  29  38  36  38  59  62  78  83  80  63  39  54  60  46  64  7
7  60 
##  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  5
6  57 
##  60  57  74  55  51  55  66  45  80  72  69  60  72  71  68  82 101 109  9
2  93 
##  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  7
6  77 
## 100 127  80  81  90  73  74  64  75  78  64  71  67  53  56  62  50  35  3
1  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:

B) The LIFE variable? Your answer here: 18
C) The SNSUSE variable? Your answer here: 12
D) The SNSFREQ_recoded variable? Your answer here: 18
E) The PPREG4 variable? Your answer here: 0

F) The PPWORK variable? Your answer here: 0
G) The PPINCIMP variable? Your answer here: 0
H) The PPGENDER variable? Your answer here: 0
I) The PPETHM variable? Your answer here: 0
J) The IDEO variable? Your answer here: 116
K) The PPEDUCAT variable? Your answer here: 0
L) The KNOWLEDGE variable? Your answer here: 13
M) The ENJOY variable? Your answer here: 46
N) The AGE variable? Your answer here: 0

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

```
# set -1 and -2  equal to "NA"
pew.start[pew.start == -1| pew.start == -2] <- NA
```

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

```
# drop rows with "NA"
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.

```
# count rows in pew.start
nrow(pew.start)

## [1] 3836
```

O) 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.

Re-code the LIFE variable such that "Worse today" is equal to one and "Better today"/"About the same" are equal to 0.

```
# re-code the life variable
pew.start$worse <- ifelse(pew.start$LIFE == 2, 1, 0)
```

To confirm that you recoded the LIFE variable correctly, display a table showing the counts of both the original and the binarized LIFE variables.

```
# display the frequencies of the original and recoded outcome
table(pew.start$LIFE, exclude = NULL)

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

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

##
##    0    1
## 2033 1803
```

A) Per your table of the original LIFE variable, how many people responded "worse today" to the this question?

Your answer here: 1803

B) Per your table of the original LIFE variable, how many people responded something other than "worse today" to the this question?

Your answer here: 2033

C) Per your 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 your recoding)

Your answer here (yes/no): Yes

Finally, set all variables to the correct type: - Continuous: AGE, PPINCIMP - Categorical: all others

```
# re-code variables
pew.start$worse <- as.factor(pew.start$worse)
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)
```

```r
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.

```r
# check data structure
str(pew.start)
```

```
## tibble [3,836 × 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 tod
ay 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 Faceboo
k, 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" "Mi
dwest" ...
##  $ 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-His
panic" "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" "Conser
vative" "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 ne
ws 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 6
8 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               : Factor w/ 2 levels "0","1": 2 2 1 2 2 1 2 1 2 2 .
..
##  $ 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

```
# create new dataframe for analysis
pew3 <- dplyr::select(pew.start, worse, age, income, reg4_factor, work_factor
,              gender_factor, eth_factor, ideo_factor, edu_factor, know_factor
,
          enjoy_factor, snsuse_factor, snsfreqrecode_factor)
# print data structure
str(pew3)

## tibble [3,836 × 13] (S3: tbl_df/tbl/data.frame)
##  $ worse               : Factor w/ 2 levels "0","1": 2 2 1 2 2 1 2 1 2 2 .
..
##  $ 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.

```
# save the number of rows in pew3
n <- nrow(pew3)
```

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 following seed.

```
# set seed (reproducable)
set.seed(123456)
```

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*.1
5)))
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
```

A) 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

B) 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 and 1

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

```
# create random sample of int 0-2, poroportions 70, 15, 15
tvt2 <- sample(rep(0:2,c(round(n*.15),round(n*.15),n-2*round(n*.15))))
# subset rows by sample int 0-2 for train validate test
dat.train<-pew3[tvt2==2,]
dat.valid<-pew3[tvt2==1,]
dat.test<-pew3[tvt2==0,]
```

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

Your answer here: 2686

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

Your answer here: 575

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

Your answer here: 575

Question 5 - 5 points

For this problem set, you'll develop 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. Use the data in the training set for this. Display each step of the forward selection by including "trace=1" as an argument in your step() function.

```
# create the null model
model.null <-glm(worse~1,data=dat.train,family="binomial")
# create formula for forward model selection (scope)
fwd_fmla <- as.formula(str_c("worse ~ ",
            str_c(names(dat.train)[2:(ncol(dat.train))], collapse = "+")))
# sanity check
fwd_fmla

## worse ~ age + income + reg4_factor + work_factor + gender_factor +
##     eth_factor + ideo_factor + edu_factor + know_factor + enjoy_factor +
##     snsuse_factor + snsfreqrecode_factor

# create the model using automated forward selection
life.fwd <- step(model.null, scope = fwd_fmla, direction = "forward", trace =
1)

## Start:  AIC=3719.48
## worse ~ 1
##
##                          Df Deviance    AIC
## + income                  1   3660.1 3664.1
## + edu_factor              3   3662.9 3670.9
## + know_factor             3   3693.8 3701.8
## + ideo_factor             4   3692.4 3702.4
## + enjoy_factor            3   3699.7 3707.7
## + gender_factor           1   3704.9 3708.9
## + snsuse_factor           1   3712.2 3716.2
## <none>                        3717.5 3719.5
## + reg4_factor             3   3711.6 3719.6
## + snsfreqrecode_factor    5   3709.4 3721.4
## + age                     1   3717.5 3721.5
## + work_factor             6   3707.7 3721.7
## + eth_factor              4   3712.9 3722.9
##
## Step:  AIC=3664.1
## worse ~ income
##
##                          Df Deviance    AIC
## + edu_factor              3   3632.9 3642.9
## + ideo_factor             4   3633.3 3645.3
## + know_factor             3   3646.3 3656.3
## + gender_factor           1   3651.4 3657.4
## + enjoy_factor          3   3649.7 3659.7
## + snsuse_factor         1   3655.2 3661.2
## + eth_factor            4   3649.4 3661.4
## + reg4_factor           3   3651.6 3661.6
## <none>                      3660.1 3664.1
## + age                   1   3660.0 3666.0
## + snsfreqrecode_factor  5   3652.8 3666.8
## + work_factor           6   3658.4 3674.4
##
## Step:  AIC=3642.88
## worse ~ income + edu_factor
##
##                        Df Deviance    AIC
## + ideo_factor           4   3609.7 3627.7
## + gender_factor         1   3624.4 3636.4
## + eth_factor            4   3620.7 3638.7
## + know_factor           3   3623.9 3639.9
## + snsuse_factor         1   3628.0 3640.0
## + reg4_factor           3   3624.3 3640.3
## + enjoy_factor          3   3624.9 3640.9
## <none>                      3632.9 3642.9
## + age                   1   3632.8 3644.8
## + snsfreqrecode_factor  5   3625.7 3645.7
## + work_factor           6   3631.7 3653.7
##
## Step:  AIC=3627.67
## worse ~ income + edu_factor + ideo_factor
##
##                        Df Deviance    AIC
## + gender_factor         1   3598.5 3618.5
## + snsuse_factor         1   3603.8 3623.8
## + reg4_factor           3   3599.8 3623.8
## + know_factor           3   3600.1 3624.1
## + eth_factor            4   3599.6 3625.6
## + enjoy_factor          3   3602.4 3626.4
## <none>                      3609.7 3627.7
## + snsfreqrecode_factor  5   3601.4 3629.4
## + age                   1   3609.6 3629.6
## + work_factor           6   3608.2 3638.2
##
## Step:  AIC=3618.46
## worse ~ income + edu_factor + ideo_factor + gender_factor
##
##                        Df Deviance    AIC
## + reg4_factor           3   3588.7 3614.7
## + eth_factor            4   3588.3 3616.3
## + snsuse_factor         1   3594.7 3616.7
## + know_factor           3   3591.1 3617.1
## <none>                      3598.5 3618.5
## + enjoy_factor          3   3592.7 3618.7
## + age                    1   3598.5 3620.5
## + snsfreqrecode_factor  5   3592.4 3622.4
## + work_factor           6   3596.6 3628.6
##
## Step:  AIC=3614.73
## worse ~ income + edu_factor + ideo_factor + gender_factor + reg4_factor
##
##                        Df Deviance    AIC
## + snsuse_factor         1   3584.8 3612.8
## + know_factor           3   3582.0 3614.0
## + eth_factor            4   3580.2 3614.2
## <none>                      3588.7 3614.7
## + enjoy_factor          3   3582.8 3614.8
## + age                   1   3588.7 3616.7
## + snsfreqrecode_factor  5   3582.7 3618.7
## + work_factor           6   3587.0 3625.0
##
## Step:  AIC=3612.76
## worse ~ income + edu_factor + ideo_factor + gender_factor + reg4_factor +
##     snsuse_factor
##
##                        Df Deviance    AIC
## + know_factor           3   3577.7 3611.7
## + eth_factor            4   3576.2 3612.2
## <none>                      3584.8 3612.8
## + enjoy_factor          3   3578.9 3612.9
## + age                   1   3584.6 3614.6
## + snsfreqrecode_factor  4   3582.7 3618.7
## + work_factor           6   3583.2 3623.2
##
## Step:  AIC=3611.71
## worse ~ income + edu_factor + ideo_factor + gender_factor + reg4_factor +
##     snsuse_factor + know_factor
##
##                        Df Deviance    AIC
## + eth_factor            4   3568.5 3610.5
## <none>                      3577.7 3611.7
## + enjoy_factor          3   3573.1 3613.1
## + age                   1   3577.7 3613.7
## + snsfreqrecode_factor  4   3575.6 3617.6
## + work_factor           6   3576.0 3622.0
##
## Step:  AIC=3610.55
## worse ~ income + edu_factor + ideo_factor + gender_factor + reg4_factor +
##     snsuse_factor + know_factor + eth_factor
##
##                        Df Deviance    AIC
## <none>                      3568.5 3610.5
## + enjoy_factor          3   3563.9 3611.9
## + age                   1   3568.5 3612.5
## + snsfreqrecode_factor  4   3566.0 3616.0
## + work_factor           6   3566.8 3620.8
```

Now, save each of the models that appeared as steps (e.g., the one-predictor model, the two-predictor model, the three predictor model, all the way to the model where the forward selection terminates) in your forward selection as model objects. You will need these for the next question.

```
# create glm's
model.1 <- glm(worse~income,data=dat.train,family="binomial")

model.2 <- glm(worse~income+edu_factor,data=dat.train,family="binomial")

model.3 <- glm(worse~income+edu_factor+ideo_factor
               ,data=dat.train,family="binomial")

model.4 <- glm(worse~income+edu_factor+ideo_factor+gender_factor
               ,data=dat.train,family="binomial")

model.5 <- glm(worse~income+edu_factor+ideo_factor+gender_factor+reg4_factor
               ,data=dat.train,family="binomial")

model.6 <- glm(worse~income+edu_factor+ideo_factor+gender_factor+reg4_factor+
                 snsuse_factor
               ,data=dat.train,family="binomial")

model.7 <- glm(worse~income+edu_factor+ideo_factor+gender_factor+reg4_factor+
                 snsuse_factor+know_factor
               ,data=dat.train,family="binomial")

model.8 <- glm(worse~income+edu_factor+ideo_factor+gender_factor+reg4_factor+
                 snsuse_factor+know_factor+eth_factor
               ,data=dat.train,family="binomial")
```

Question 6 - 10 points

To test the candidate models, 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$sum(dat.thischd * log(pred.m) + (1 - dat.thischd)$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.

```
# create valid.dev to calculate model deviance
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))
}
```

Use this adapted function to compute the deviances of each of the candidate models when applied to the validation data set. You'll need to change the outcome variable's type to numeric for the function to work correctly.

```
# convert worse to numeric type for use with deviance function
dat.valid$worse <- as.numeric(as.character(dat.valid$worse))



# Your code for computing the validation-set deviances of each of the candida
te models.

dev.1 <- valid.dev(model.1, dat.this = dat.valid)
dev.2 <- valid.dev(model.2, dat.this = dat.valid)
dev.3 <- valid.dev(model.3, dat.this = dat.valid)
dev.4 <- valid.dev(model.4, dat.this = dat.valid)
dev.5 <- valid.dev(model.5, dat.this = dat.valid)
dev.6 <- valid.dev(model.6, dat.this = dat.valid)
dev.7 <- valid.dev(model.7, dat.this = dat.valid)
dev.8 <- valid.dev(model.8, dat.this = dat.valid)
```

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 deviance for the candidate models
dev.1

## [1] 784.7869

dev.2

## [1] 771.5897

dev.3

## [1] 767.713

dev.4

## [1] 772.3748

dev.5

## [1] 772.8025
```

```
dev.6

## [1] 772.4389

dev.7

## [1] 774.0455

dev.8

## [1] 775.0488
```

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

Your answer here: 784.7869

B) What is the validation deviance of the model with the most predictors (the last model fit in the forward selection)?

Your answer here: 775.0488

C) Which of the models out of all the candidate models had the lowest validation deviance?

Your answer here: model.3

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

Your answer here: model.3

Question 7 - 10 points

Now that you've chosen a candidate model based on its performance on the validation data set, you'll now test that model by computing the deviance of this model when applied to the test data set. You can re-use the valid.dev function for this.

```
# convert worse to numeric type for use with deviance function
dat.test$worse <- as.numeric(as.character(dat.test$worse))
# compute deviance
test.dev <- valid.dev(model.3, dat.test)
# print deviance
test.dev

## [1] 787.0983
```

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

Your answer here: 787.0983

To further examine the generalizability, construct a confusion matrix comparing the actual 0/1 values from the test set for the re-coded LIFE variable ("worse) and the predicted 0/1 values 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.

```
# convert worse back to factor
dat.test$worse <- as.factor(dat.test$worse)
# create prediction
prediction <- predict(model.2,dat.test,type="response")
# binarize the outcome
prediction <- as.factor(ifelse(prediction > 0.5, 1, 0))
# create the confusion matrix using table function
confusion.matrix <- table(dat.test$worse, prediction,dnn=c("Actual","Predicte
d"))
# print the confusion matrix
confusion.matrix

##        Predicted
## Actual   0   1
##       0 183 122
##       1 149 121
```

2) How many true positives did your model produce?

Your answer here: 121

3) How many true negatives did your model produce?

Your answer here: 183

4) How many false positives did your model produce?

Your answer here: 122

5) 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.

```r
# compute accuracy
# true positives and true negatives are on the diagonals
accuracy <- sum(diag(confusion.matrix))/sum(confusion.matrix)

# compute precision
# (true positive/(true positive + false positive))
precision <- confusion.matrix[2,2]/sum(confusion.matrix[,2])

# compute recall
# (true positive/(true positive + false negative)
recall <- confusion.matrix[2,2]/sum(confusion.matrix[2,])

# compute F1 score
F1 <- 2*((precision*recall)/(precision+recall))
```

6) What is the *accuracy* of this model?

Your answer here: 0.528695652173913

7) What is the *precision* of this model?

Your answer here: 0.497942386831276

8) What is the *recall* of this model?

Your answer here: 0.448148148148148

9) What is the *F1 score* of this model?

Your answer here: 0.471734892787524