Module 6 problem set

INSERT YOUR NAME HERE

INSERT DATE

Contents

General instructions

The purpose of this problem set is to familiarize yourself with a new dataset, the National Longitudinal Study of 1972 (NLS-72). NLS is a nationally representative, longitudinal study of 12th graders in 1972 with follow-up surveys throughout their postsecondary years. You will be using the Postsecondary Education Transcript File of the NLS-72, which contains information on transcripts from NLS-72 senior cohort members who reported attending a postsecondary institution after high school.

For your next problem set [next week], you will use the NLS Postsecondary Education Transcript File to create college GPA variables.

Load library and data

You'll need to load the `tidyverse`, `haven` and `labelled` libraries in order to load and work with the NLS data. If these packages are not yet installed, then you must install before you load. Install in "console" rather than .Rmd file

- Generic syntax: `install.packages("package_name")`
- Install "haven": `install.packages("haven")`

Note: when we **load** package, name of package is not in quotes; but when we **install** package, name of package is in quotes:

- `install.packages("tidyverse")`
- `library(tidyverse)`

```
library(tidyverse)
library(haven)
library(labelled)
```

```
rm(list = ls()) # remove all objects

nls_crs<- read_dta(file="https://github.com/ksalazar3/HED696C_RClass/raw/master/data/nls72/nls72petscrs_
```

Step 1: Investigate Variables

1. Use `typeof`, `class`, `str`, and `attributes` functions to investigate the following variables: crsgrada, crsgradb, gradtype, crsecred.

```
#Base R
typeof(nls_crs$crsgrada)
class(nls_crs$crsgrada)
attributes(nls_crs$crsgrada)

typeof(nls_crs$crsgradb)
class(nls_crs$crsgradb)
attributes(nls_crs$crsgradb)

typeof(nls_crs$gradtype)
class(nls_crs$gradtype)
attributes(nls_crs$gradtype)

typeof(nls_crs$crsecred)
class(nls_crs$crsecred)
attributes(nls_crs$crsecred)


#Tidyverse
nls_crs %>% select(crsgrada) %>% typeof()
nls_crs %>% select(crsgrada) %>% class()
nls_crs %>% select(crsgrada) %>% attributes()

nls_crs %>% select(crsgradb) %>% typeof()
nls_crs %>% select(crsgradb) %>% class()
nls_crs %>% select(crsgradb) %>% attributes()

nls_crs %>% select(gradtype) %>% typeof()
nls_crs %>% select(gradtype) %>% class()
nls_crs %>% select(gradtype) %>% attributes()

nls_crs %>% select(crsecred) %>% typeof()
nls_crs %>% select(crsecred) %>% class()
nls_crs %>% select(crsecred) %>% attributes()
```

Step 2: Create New Variables

1. `crsgrada` is the variable for letter course grades. Create a factor version of the `crsgrada` variable. Hint: knowing what class the variable is currently and investigating the variable using `count()` will be helpful to creating the new factor version. Retain the new factor version variable in the nls_crs dataframe using the variable name `crsgrad_fac`. Check that this new variable is a factor class.

```
nls_crs %>% count(crsgrada)
```

```
## # A tibble: 25 x 2
##    crsgrada      n
##    <chr>     <int>
##  1 99        24814
##  2 A        113200
##  3 A+          523
##   4 A-          5221
##   5 AU           598
##   6 B         126003
##   7 B+          6639
##   8 B-          3813
##   9 C          89782
##  10 C+          4285
## # i 15 more rows
```

```
class(nls_crs$crsgrada)
```

```
## [1] "character"
```

```
nls_crs <- nls_crs %>%
  mutate(crsgrada_fac = factor(crsgrada))

#alternative code from Karina
#nls_crs$crsgrada_fac <- factor(nls_crs$crsgrada, levels = c("99", "A", "A-","A+", "AU", "B", "B-", "B+


nls_crs %>% count(crsgrada_fac)
```

```
## # A tibble: 25 x 2
##    crsgrada_fac      n
##    <fct>         <int>
##   1 99           24814
##   2 A           113200
##   3 A-            5221
##   4 A+             523
##   5 AU             598
##   6 B           126003
##   7 B-            3813
##   8 B+            6639
##   9 C            89782
##  10 C-            1841
## # i 15 more rows
```

```
class(nls_crs$crsgrada_fac)
```

```
## [1] "factor"
```

```
attributes(nls_crs$crsgrada_fac)
```

```
## $levels
##  [1] "99" "A"  "A-" "A+" "AU" "B"  "B-" "B+" "C"  "C-" "C+" "CR" "D"  "D-" "D+"
## [16] "E"  "F"  "I"  "NO" "P"  "S"  "U"  "W"  "WF" "WP"
##
## $class
## [1] "factor"
```

2. Create a numeric course grade version of the `crsgrada_fac` variable named `numgrade` with the following numeric values based on attribute levels from `crsgrada_fac` Hint: use `mutate()` and `recode()`. Retain this new `numgrade` variable.
    - A+= 4; A=4; A-=3.7; B+=3.3; B=3; B-=2.7; C+=2.3; C=2; C-=1.7; D+=1.3; D=1; D-=.7; F=0; E=0; WF=0
    - All other letter grades should have missing values for `numgrade`
    - When recoding to missing use `NA_real_` rather than `NA` due to `recode()` needing a double

type/numeric class value to recode and NA is a logical)

```
nls_crs <- nls_crs %>%
  mutate(numgrade =
    recode(crsgrada_fac,
      "A+" =4,
      "A" = 4,
      "A-" = 3.7,
      "B+" =3.3,
      "B" = 3,
      "B-" = 2.7,
      "C+" =2.3,
      "C" = 2,
      "C-" = 1.7,
      "D+" =1.3,
      "D" = 1,
      "D-" = 0.7,
      "F" = 0,
      "E" = 0,
      "WF" = 0,
      .default = NA_real_
    )
  )

nls_crs %>% count(numgrade)
```

```
## # A tibble: 13 x 2
##    numgrade      n
##       <dbl>  <int>
##  1       0    14838
##  2       0.7    286
##  3       1    22883
##  4       1.3    610
##  5       1.7   1841
##  6       2    89782
##  7       2.3   4285
##  8       2.7   3813
##  9       3   126003
## 10       3.3   6639
## 11       3.7   5221
## 12       4   113723
## 13      NA    94598
```

```
nls_crs %>% count(numgrade, crsgrada_fac)
```

```
## # A tibble: 25 x 3
##    numgrade crsgrada_fac      n
##       <dbl> <fct>         <int>
##  1       0   E              874
##  2       0   F            13170
##  3       0   WF             794
##  4       0.7 D-             286
##  5       1   D            22883
##  6       1.3 D+             610
##  7       1.7 C-            1841
##  8      2   C            89782
##  9      2.3 C+            4285
## 10      2.7 B-            3813
## # i 15 more rows
```

3. `gradtype` is a labelled class variable for the type of grade given for each course. Retrieve the variable label and value labels for `gradtype`. Get a count of `gradtype` showing the values and the value labels. Now, get another count by filtering for observations associated with "{MISSING}".

```
nls_crs %>% select(gradtype) %>% var_label()
```

```
## $gradtype
## [1] "TYPE OF GRADE"
```

```
nls_crs %>% select(gradtype) %>% val_labels()
```

```
## $gradtype
##    1. letter   2. numeric 9. {MISSING}
##            1            2            9
```

```
nls_crs %>% count(gradtype) %>% as_factor()
```

```
## # A tibble: 3 x 2
##   gradtype          n
##   <fct>         <int>
## 1 1. letter    459348
## 2 2. numeric    10517
## 3 9. {MISSING}  14657
```

```
nls_crs %>% filter(gradtype==9) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 14657
```

4. `crsgradb` is the variable for numerical course grades. There are several issues with this variable. First, missing observations for `crsgradb` are currently `999` and `999.999`. The variable also has values greater than 4 (problematic when the highest possible grade A+ = 4). Create and retain a new `crsgradb_v2` variable that replaces all values greater than 4 for `crsgradb` to `NA` (Hint: you can use the `mutate` and `if_else()` functions to either replace the value to NA or keep the current value of the variable based on whether the expression you specify evaluates to `TRUE` or `FALSE`. See below…

ANSWER PROVIDED FOR YOU

```
nls_crs %>% count(crsgradb)
#table(nls_crs$crsgradb)

nls_crs<- nls_crs %>%
  mutate(crsgradb_v2= ifelse(crsgradb>4, NA, crsgradb))
```

5. `crsecred` is the variable for how many total credits were possible for each course. Missing observations for `crsecred` are currently `999` and `999.999`. Using code similar to Question 5, create and retain a new `crsecred_v2` variable that replaces values of `999` and `999.999` to `NA`, whereas all other "non-missing" values stay the same as the original input variable.

```
nls_crs <- nls_crs%>%
  mutate(crsecredv2= ifelse(crsecred>=999, NA, crsecred))
```

```
# you can check the new var against the old var
nls_crs %>% count(crsecredv2, crsecred)
```

```
## # A tibble: 414 x 3
##    crsecredv2 crsecred     n
##         <dbl>    <dbl> <int>
##  1      0        0     11500
##  2      0.017    0.017     1
##  3      0.05     0.05      2
##  4      0.067    0.067    12
##  5      0.08     0.08      2
##  6      0.083    0.083     2
##  7      0.1      0.1      14
##  8      0.12     0.12      2
##  9      0.13     0.13      2
## 10      0.133    0.133    19
## # i 404 more rows
```

6. Create a "final" numerical grade variable named `numgrade_v2` that incorporates values from observations where `gradtype==1` (i.e., "type of grade" is "letter") and incorporates values from observations where `gradtype==2` (i.e., "type of grade" is "numeric"). For, observations where `gradtype` indicates letter grades were used and `crsecred_v2` is not missing, value of `numgrade_v2` should be the value of the variable `numgrade` which you created previously. For observations where `gradtype` indicates that numeric grades were used and `crsecred_v2` is not missing, value of `numgrade_v2` should be the value of the variable `crsgradb_v2` which you created previously.
   - Hint: use `mutate()` and `case_when()`.
   - Note: For, observations where `gradtype` indicates letter grades, values of numeric variable `numgrade` you previously created should be as follows:
     - A+= 4; A=4; A-=3.7; B+=3.3; B=3; B-=2.7; C+=2.3; C=2; C-=1.7; D+=1.3; D=1; D-=.7; F=0; E=0; WF=0
     - and `numgrade` should be missing for all observations that do not have these above values.

```
nls_crs <- nls_crs %>%
    mutate(
      numgrade_v2=case_when(
        gradtype==1 & (!is.na(crsecredv2)) ~ numgrade,
        gradtype==2 & (!is.na(crsecredv2)) ~ crsgradb_v2
      )
    )

# Just some quality checks....

# Method 1: Check the Mean of New Variable

  #Karina's Old School Way but still tidyverse
  nls_crs %>% select(numgrade_v2) %>%
    summarize_all(.fun = list(mean,sd), na.rm = TRUE)
```

```
## # A tibble: 1 x 2
##     fn1   fn2
##   <dbl> <dbl>
## 1  2.83  1.04
```

```
  #Tidyverse Way
  nls_crs %>%
    summarise(mean=mean(numgrade_v2, na.rm = TRUE)) #matched Brent's answer
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1  2.83
```

```
  #Base R
  mean(nls_crs$numgrade_v2, na.rm = TRUE)
```

```
## [1] 2.828731
```

```
# Method 2: Row by Row Checks
  nls_crs %>% filter(gradtype==2) %>%
    select(gradtype, crsecredv2, crsgrada_fac, crsgradb_v2, numgrade_v2) %>%
    print(n=10)
```

```
## # A tibble: 10,517 x 5
##    gradtype       crsecredv2 crsgrada_fac crsgradb_v2 numgrade_v2
##    <dbl+lbl>           <dbl> <fct>              <dbl>       <dbl>
##  1 2 [2. numeric]        3.1 99                   2.7         2.7
##  2 2 [2. numeric]        1.9 99                   2.7         2.7
##  3 2 [2. numeric]        2.5 99                   2           2
##  4 2 [2. numeric]        1   99                   2           2
##  5 2 [2. numeric]        2.5 99                   2           2
##  6 2 [2. numeric]        3.1 99                   3.3         3.3
##  7 2 [2. numeric]        3.4 99                   3.3         3.3
##  8 2 [2. numeric]        1   99                   2           2
##  9 2 [2. numeric]        4   99                   2.4         2.4
## 10 2 [2. numeric]        4   99                   2.7         2.7
## # i 10,507 more rows
```

```
  nls_crs %>% filter(gradtype==1) %>%
    select(gradtype, crsecredv2, crsgrada_fac, crsgradb_v2, numgrade_v2) %>%
    print(n=10)
```

```
## # A tibble: 459,348 x 5
##    gradtype      crsecredv2 crsgrada_fac crsgradb_v2 numgrade_v2
##    <dbl+lbl>          <dbl> <fct>              <dbl>       <dbl>
##  1 1 [1. letter]        0.5 B                      3           3
##  2 1 [1. letter]        1   C                      2           2
##  3 1 [1. letter]        0.5 A                      4           4
##  4 1 [1. letter]        0.7 A                      4           4
##  5 1 [1. letter]        1   B                      3           3
##  6 1 [1. letter]        0.7 C                      2           2
##  7 1 [1. letter]        1   B                      3           3
##  8 1 [1. letter]        1   B                      3           3
##  9 1 [1. letter]        0.7 A                      4           4
## 10 1 [1. letter]        0.7 C                      2           2
## # i 459,338 more rows
```

```
  nls_crs %>% filter(gradtype==9) %>%
    select(gradtype, crsecredv2, crsgrada_fac, crsgradb_v2, numgrade_v2) %>%
    print(n=10) #all obs with missing gradtype(==9) are missing in numgrade_v2 [correct!]
```

```
## # A tibble: 14,657 x 5
##    gradtype         crsecredv2 crsgrada_fac crsgradb_v2 numgrade_v2
##    <dbl+lbl>              <dbl> <fct>               <dbl>       <dbl>
##  1 9 [9. {MISSING}]        5    99                     NA          NA
##  2 9 [9. {MISSING}]        4    99                     NA          NA
##  3 9 [9. {MISSING}]        6    99                     NA          NA
##  4 9 [9. {MISSING}]        4    99                     NA          NA
##  5 9 [9. {MISSING}]        1    99                     NA          NA
##  6 9 [9. {MISSING}]       NA    99                     NA          NA
##  7 9 [9. {MISSING}]        3.25 99                     NA          NA
##  8 9 [9. {MISSING}]        3    99                     NA          NA
##  9 9 [9. {MISSING}]        3    99                     NA          NA
## 10 9 [9. {MISSING}]        2    99                     NA          NA
## # i 14,647 more rows
```

7. Use 'set_variable_labels' function to set the following variable labels to the new variables: 'numgrade', 'crsgradb_v2', 'crsecredv2' and 'numgrade_v2'.

- numgrade = "numeric grade version for crsgrada_fac"
- crsgradb_v2 = "crsgradb without values greater than 4"
- crsecredv2 = "recode missing values for crsecred"
- numgrade_v2 = "final numerical grade"

```
nls_crs <- nls_crs %>%
  set_variable_labels(numgrade = "numeric grade version for 'crsgrada_fac'",
                      crsgradb_v2 = "`crsgradb` without values greater than 4",
                      crsecredv2 = "recode missing values for `crsecred`",
                      numgrade_v2 = "final numerical grade")
```

8. First create a new variable named 'numgrade_v3', which equals to 1 if 'numgrade_v2' is greater than 3, and equals to 0 if 'numgrade_v2' is not greater than 3. Second use 'set_value_labels' function to add value labels ("greater than 3" and "not greater than 3") to this new variable. Third change the variable into a factor variable. Investigate the class of this variable in each step.

```
nls_crs <- nls_crs %>%
  mutate(numgrade_v3 = if_else(numgrade_v2>3,1,0))
class(nls_crs$numgrade_v3)
```

```
## [1] "numeric"
```

```
nls_crs <- nls_crs %>%
  set_value_labels(numgrade_v3 = c("greater than 3" = 1,
                                   "no greater than 3" = 0))
class(nls_crs$numgrade_v3)
```

```
## [1] "haven_labelled" "vctrs_vctr"     "double"
```

```
nls_crs <- nls_crs %>%
  mutate(numgrade_v3 = factor(numgrade_v3))
class(nls_crs$numgrade_v3)
```

```
## [1] "factor"
```