Module 8 Problem Set

INSERT YOUR NAME HERE

Contents

Instructions **1**
Overview 1

Load library and data **1**

Part I: Conceptual questions **2**

Part II: Questions about reshaping long to wide **2**
Description of the data 2
Overview of the reshaping long to wide tasks 2
Load data and create three new data frames 3
Questions related to reshaping the dataset `agegroup1_obs` from long to wide 5
Questions related to reshaping the dataset levstudy1_obs from long to wide 10

Part III: Questions about reshaping wide to long **14**

Instructions

Overview

This problem set has three parts.

1. I'll ask you some definition/conceptual questions about the concepts introduced in lecture
2. Tidying untidy data: reshaping from long to wide
    - e.g., dataset has one row for each combination of university ID and enrollment age group, but you want a dataset with one row per university ID and one enrollment variable for each age group
    - for these questions we'll use fall enrollment data from the Integrated Postsecondary Data System (IPEDS), specifically the fall enrollment sub-survey that focuses on enrollment by age group
3. Tidying untidy data: reshaping from wide to long
    - for these questions we'll use data from the NCES digest of education statistics that contains data about the total number of teachers in each state

Load library and data

```
library(tidyr)
library(tidyverse)
#> -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
#> v ggplot2 3.3.6     v dplyr   1.0.9
#> v tibble  3.1.8     v stringr 1.4.0
#> v readr   2.1.2     v forcats 0.5.1
#> v purrr   0.3.4
```

```
#> -- Conflicts ---------------------------------------- tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
library(haven)
library(labelled)
```

Part I: Conceptual questions

- What is the difference between the terms "unit of analysis" [our term; not necessarily used outside this class] and "observational level" [A Wickham term]?
  - ANSWER: Wickham defines "observational level" as what each observation should represent in a tidy dataset (i.e., it is a data concept), whereas I define "unit of analysis" as what each row in the data actually represents (i.e., refers to data structure).
- What are the three rules of tidy data?
  - ANSWER: 1) Each variable must have its own column; 2) Each observation must have its own row; 3) Each value must have its own cell.

Part II: Questions about reshaping long to wide

Description of the data

For these questions, we'll be using data from the Fall Enrollment survey component of the Integrated Postsecondary Education Data System (IPEDS)

- Specifically, we'll be using data from the survey sub-component that focuses on enrollment by age-group.
- The dataset we'll be using data from Fall 2016 (i.e., Fall of the 2016-17 academic year)
- Here is a link to a data dictionary (an excel file) for the enrollment by age dataset: LINK
- In the dataset you load below:
  - I've dropped a few of the variables from the raw enrollment by age data
  - I've added a few variables from the "institutional characteristics" survey (e.g., institution name, state, sector) that should be pretty self explanatory if you examine the variable labels and/or value labels
- the variable `unitid` is the ID variable for each college/university
- the dataset has one observation for each combination of the variables unitid-efbage-lstudy; in other words the unit of analysis is university per age group per level of study

Overview of the reshaping long to wide tasks

- We will load the data frame via read_dta using the hyperlink and assign it the name `age_f16_allvars_allobs`
- Then, we'll create two different data frame objects based on the data frame `age_f16_allvars_allobs`
  - A dataframe `agegroup1_obs` that has fewer variables than `age_f16_allvars_allobs` and keeps observations where age-group equals `1` (1. All age categories total)
    - this data frame has the simplest structure; we'll reshape this one first
  - A dataframe `levstudy1_obs` that has fewer variables than `age_f16_allvars_allobs` and keeps observations where "level of study" equals `1` (1. All Students total)
    - we'll reshape this one second
- Questions related to reshaping `agegroup1_obs`
- Questions related to reshaping `levstudy1_obs`

Load data and create three new data frames

- Load IPEDS data that contains fall enrollment by age

NOTE: IN THIS QUESTION, WE GIVE YOU THE ANSWERS; ALL YOU HAVE TO DO IS RUN THE BELOW CODE CHUNK

```
rm(list = ls()) # remove all objects
#getwd()
#list.files("../../../documents/rclass/data/ipeds/ef/age") # list files in directory w/ NLS data

#Read Stata data into R using read_data() function from haven package
age_f16_allvars_allobs <- read_dta(file="https://github.com/ksalazar3/HED696c_Rclass/raw/master/data/ip

#rename a couple variables
age_f16_allvars_allobs <- age_f16_allvars_allobs %>% rename(agegroup=efbage, levstudy=lstudy)

#list variables and variable labels
names(age_f16_allvars_allobs)
#>  [1] "unitid"       "agegroup"     "levstudy"     "efage01"      "efage02"
#>  [6] "efage03"      "efage04"      "efage05"      "efage06"      "efage07"
#> [11] "efage08"      "efage09"      "fullname"     "stabbr"       "sector"
#> [16] "iclevel"      "control"      "hloffer"      "locale"       "merge_age_ic"
age_f16_allvars_allobs %>% var_label()
#> $unitid
#> [1] "Unique identification number of the institution"
#>
#> $agegroup
#> [1] "Age category"
#>
#> $levstudy
#> [1] "Level of student"
#>
#> $efage01
#> [1] "Full time men"
#>
#> $efage02
#> [1] "Full time women"
#>
#> $efage03
#> [1] "Part time men"
#>
#> $efage04
#> [1] "Part time women"
#>
#> $efage05
#> [1] "Full time total"
#>
#> $efage06
#> [1] "Part time total"
#>
#> $efage07
#> [1] "Total men"
#>
#> $efage08
```

```
#> [1] "Total women"
#>
#> $efage09
#> [1] "Grand total"
#>
#> $fullname
#> [1] "Institution (entity) name"
#>
#> $stabbr
#> [1] "State abbreviation"
#>
#> $sector
#> [1] "Sector of institution"
#>
#> $iclevel
#> [1] "Level of institution"
#>
#> $control
#> [1] "Control of institution"
#>
#> $hloffer
#> [1] "Highest level of offering"
#>
#> $locale
#> [1] "Degree of urbanization (Urban-centric locale)"
#>
#> $merge_age_ic
#> NULL
```

- Create two new data frames based on `age_f16_allvars_allobs`

NOTE: IN THIS QUESTION, WE GIVE YOU THE ANSWERS; ALL YOU HAVE TO DO IS RUN THE BELOW CODE CHUNK

```
#Create dataframe that keeps observations where age-group equals `1` (1. All age categories total)
agegroup1_obs <- age_f16_allvars_allobs %>%
  select(fullname,unitid,agegroup,levstudy,efage09,stabbr,locale) %>%
  filter(agegroup==1) %>%
  select(-agegroup)

glimpse(agegroup1_obs)
#> Rows: 7,019
#> Columns: 6
#> $ fullname <chr> "Amridge University", "Amridge University", "Amridge Universi~
#> $ unitid   <dbl> 100690, 100690, 100690, 100724, 100724, 100724, 100751, 10075~
#> $ levstudy <dbl+lbl> 1, 2, 5, 1, 2, 5, 1, 2, 5, 1, 2, 1, 2, 5, 1, 2, 5, 1, 2, ~
#> $ efage09  <dbl> 597, 294, 303, 5318, 4727, 591, 37663, 32563, 5100, 1769, 176~
#> $ stabbr   <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "~
#> $ locale   <dbl+lbl> 12, 12, 12, 12, 12, 12, 13, 13, 13, 32, 32, 12, 12, 12, 1~

#Create dataframe keeps observations where "level of study" equals `1` (1. All Students total)
levstudy1_obs <- age_f16_allvars_allobs %>%
  select(fullname,unitid,agegroup,levstudy,efage09,stabbr,locale) %>%
  filter(levstudy==1) %>%
```

```
  select(-levstudy)

glimpse(levstudy1_obs)
#> Rows: 36,703
#> Columns: 6
#> $ fullname <chr> "Amridge University", "Amridge University", "Amridge Universi~
#> $ unitid   <dbl> 100690, 100690, 100690, 100690, 100690, 100690, 100690, 10069~
#> $ agegroup <dbl+lbl>  1,  2,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13,  1,  2,  ~
#> $ efage09  <dbl> 597, 57, 7, 16, 34, 540, 88, 97, 110, 158, 78, 9, 5318, 4464,~
#> $ stabbr   <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "~
#> $ locale   <dbl+lbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 1~
```

Questions related to reshaping the dataset agegroup1_obs from long to wide

- Run whatever investigations seem helpful to you to get to know the data (e.g., list variable names, list variable variable labels, list variable values, tabulations). You may decide to comment out some of these investigations before you knit and submit the problem set so that your pdf doesn't get too long.

```
#basic investigations of dataset
names(agegroup1_obs)
#> [1] "fullname" "unitid"   "levstudy" "efage09"  "stabbr"   "locale"
str(agegroup1_obs)
#> tibble [7,019 x 6] (S3: tbl_df/tbl/data.frame)
#>  $ fullname: chr [1:7019] "Amridge University" "Amridge University" "Amridge University" "Alabama St
#>   ..- attr(*, "label")= chr "Institution (entity) name"
#>   ..- attr(*, "format.stata")= chr "%91s"
#>  $ unitid  : num [1:7019] 100690 100690 100690 100724 100724 ...
#>   ..- attr(*, "label")= chr "Unique identification number of the institution"
#>   ..- attr(*, "format.stata")= chr "%12.0g"
#>  $ levstudy: dbl+lbl [1:7019] 1, 2, 5, 1, 2, 5, 1, 2, 5, 1, 2, 1, 2, 5, 1, 2, 5, 1,...
#>    ..@ label       : chr "Level of student"
#>    ..@ format.stata: chr "%21.0g"
#>    ..@ labels      : Named num [1:3] 1 2 5
#>    .. ..- attr(*, "names")= chr [1:3] "1. All Students total" "2. Undergraduate" "5. Graduate"
#>  $ efage09 : num [1:7019] 597 294 303 5318 4727 ...
#>   ..- attr(*, "label")= chr "Grand total"
#>   ..- attr(*, "format.stata")= chr "%12.0g"
#>  $ stabbr  : chr [1:7019] "AL" "AL" "AL" "AL" ...
#>   ..- attr(*, "label")= chr "State abbreviation"
#>   ..- attr(*, "format.stata")= chr "%9s"
#>  $ locale  : dbl+lbl [1:7019] 12, 12, 12, 12, 12, 12, 13, 13, 13, 32, 32, 12, 12, 1...
#>    ..@ label       : chr "Degree of urbanization (Urban-centric locale)"
#>    ..@ format.stata: chr "%19.0g"
#>    ..@ labels      : Named num [1:13] -3 11 12 13 21 22 23 31 32 33 ...
#>    .. ..- attr(*, "names")= chr [1:13] "-3. {Not available}" "11. City: Large" "12. City: Midsize" "
#>  - attr(*, "label")= chr "dct_ef2016b"
agegroup1_obs %>% var_label()
#> $fullname
#> [1] "Institution (entity) name"
#>
#> $unitid
#> [1] "Unique identification number of the institution"
#>
#> $levstudy
```

```
#> [1] "Level of student"
#>
#> $efage09
#> [1] "Grand total"
#>
#> $stabbr
#> [1] "State abbreviation"
#>
#> $locale
#> [1] "Degree of urbanization (Urban-centric locale)"
```

Sort and print a few obs

```
#sort
agegroup1_obs <- agegroup1_obs %>% arrange(unitid,levstudy)

#print a few obs
agegroup1_obs %>% head(n=10) %>% as_factor
#> # A tibble: 10 x 6
#>    fullname                          unitid levstudy       efage09 stabbr locale
#>    <chr>                              <dbl> <fct>            <dbl> <chr>  <fct>
#>  1 Amridge University                100690 1. All Studen~     597 AL     12. C~
#>  2 Amridge University                100690 2. Undergradu~     294 AL     12. C~
#>  3 Amridge University                100690 5. Graduate        303 AL     12. C~
#>  4 Alabama State University          100724 1. All Studen~    5318 AL     12. C~
#>  5 Alabama State University          100724 2. Undergradu~    4727 AL     12. C~
#>  6 Alabama State University          100724 5. Graduate        591 AL     12. C~
#>  7 The University of Alabama         100751 1. All Studen~   37663 AL     13. C~
#>  8 The University of Alabama         100751 2. Undergradu~   32563 AL     13. C~
#>  9 The University of Alabama         100751 5. Graduate       5100 AL     13. C~
#> 10 Central Alabama Community College 100760 1. All Studen~    1769 AL     32. T~
```

Run some frequencies

```
#frequency of level of study variable
agegroup1_obs %>% select(levstudy) %>% val_labels()
#> $levstudy
#> 1. All Students total      2. Undergraduate           5. Graduate
#>                     1                     2                     5
agegroup1_obs %>% count(levstudy) %>% as_factor
#> # A tibble: 3 x 2
#>   levstudy                 n
#>   <fct>                <int>
#> 1 1. All Students total  2944
#> 2 2. Undergraduate       2844
#> 3 5. Graduate            1231

#frequency of state variable
agegroup1_obs %>% select(stabbr) %>% val_labels()
#> $stabbr
#> NULL
agegroup1_obs %>% count(stabbr) %>% as_factor
#> # A tibble: 57 x 2
#>    stabbr     n
#>    <chr>  <int>
```

```
#>  1 AK         9
#>  2 AL       116
#>  3 AR        69
#>  4 AS         2
#>  5 AZ       135
#>  6 CA       699
#>  7 CO       127
#>  8 CT       117
#>  9 DC        33
#> 10 DE        20
#> # ... with 47 more rows
#> # i Use `print(n = ...)` to see more rows

#frequency of locale variable
agegroup1_obs %>% select(locale) %>% val_labels()
#> $locale
#> -3. {Not available}     11. City: Large   12. City: Midsize     13. City: Small
#>                  -3                   11                   12                   13
#>   21. Suburb: Large 22. Suburb: Midsize   23. Suburb: Small    31. Town: Fringe
#>                  21                   22                   23                   31
#>   32. Town: Distant    33. Town: Remote   41. Rural: Fringe  42. Rural: Distant
#>                  32                   33                   41                   42
#>   43. Rural: Remote
#>                  43
agegroup1_obs %>% count(locale) %>% as_factor
#> # A tibble: 13 x 2
#>    locale                 n
#>    <fct>              <int>
#>  1 -3. {Not available}     4
#>  2 11. City: Large      1621
#>  3 12. City: Midsize     841
#>  4 13. City: Small       926
#>  5 21. Suburb: Large    1596
#>  6 22. Suburb: Midsize   206
#>  7 23. Suburb: Small     143
#>  8 31. Town: Fringe      165
#>  9 32. Town: Distant     530
#> 10 33. Town: Remote      436
#> 11 41. Rural: Fringe     403
#> 12 42. Rural: Distant    110
#> 13 43. Rural: Remote      38
```

- Run the following code, which confirms that there is one row per each combination of unitid-levstudy

NOTE: IN THIS QUESTION, WE GIVE YOU THE ANSWERS; BUT TRY TO UNDERSTAND WHAT EACH PART OF THE CODE IS DOING

```
agegroup1_obs %>% group_by(unitid,levstudy) %>% # group by vars
  summarise(n_per_group=n()) %>% # create a measure of number of observations per group
  ungroup %>% # ungroup (otherwise frequency table [next step] created) separately for each group
  count(n_per_group) # frequency of number of observations per group
#> `summarise()` has grouped output by 'unitid'. You can override using the
#> `.groups` argument.
#> # A tibble: 1 x 2
#>   n_per_group     n
```

```
#>         <int> <int>
#> 1           1  7019
```

Using code from previous question as a guide, confirm that the object `agegroup1_obs` has more than one observation for each value of unitid

```
agegroup1_obs %>% group_by(unitid) %>% # group by vars
  summarise(n_per_group=n()) %>% # create a measure of number of observations per group
  ungroup %>% # ungroup (otherwise frequency table [next step] created) separately for each group
  count(n_per_group) # frequency of number of observations per group
#> # A tibble: 2 x 2
#>   n_per_group     n
#>         <int> <int>
#> 1           2  1813
#> 2           3  1131
```

- Diagnose whether the data frame `agegroup1_obs` meets each of the three criteria for tidy data
  - YOUR ANSWERS HERE:
    - Each variable must have its own column: false; the values of the column levstudy should each be variables with their own column
    - Each observation must have its own row: false; there should be one row per college/university, but this data frame has one row per college-levstudy
    - Each value must have its own cell: true
- What changes need to be made to `agegroup1_obs` to make it tidy?
  - YOUR ANSWER HERE: convert the values of the variable levstudy into their own variables; each variable will contain enrollment for that level of study
- With respect to "reshaping long to wide" to tidy a dataset, define the "names_from" parameter.
  - YOUR ANSWER HERE: the column name(s) in the untidy dataset whose values will become variable names in the tidy data
- What should the "names_from" column be in the data frame `agegroup1_obs`?
  - YOUR ANSWER HERE: names_from column should be `levstudy`
- With respect to "reshaping long to wide" to tidy a dataset, define the "values_from" paramenter.
  - YOUR ANSWER HERE: the column name(s) in the untidy dataset that contains the values for the new variables that will be created in the tidy dataset
- What should the "values_from" column be in the data frame `agegroup1_obs`?
  - YOUR ANSWER HERE: values_from column should be `efage09`

Tidy the data frame `agegroup1_obs` and create a new object `agegroup1_obs_tidy`, then print a few observations

```
agegroup1_obs %>% head(n=5)
#> # A tibble: 5 x 6
#>   fullname                unitid               levstudy efage09 stabbr   locale
#>   <chr>                    <dbl>              <dbl+lbl>   <dbl> <chr>  <dbl+lb>
#> 1 Amridge University      100690 1 [1. All Students to~     597 AL     12 [12.~
#> 2 Amridge University      100690 2 [2. Undergraduate]       294 AL     12 [12.~
#> 3 Amridge University      100690 5 [5. Graduate]            303 AL     12 [12.~
#> 4 Alabama State University 100724 1 [1. All Students to~    5318 AL     12 [12.~
#> 5 Alabama State University 100724 2 [2. Undergraduate]     4727 AL     12 [12.~

agegroup1_obs_tidy <- agegroup1_obs %>%
  pivot_wider(names_from = levstudy, values_from = efage09)

agegroup1_obs_tidy %>% head(n=5)
#> # A tibble: 5 x 7
```

```
#>   fullname                          unitid stabbr       locale   `1`   `2`   `5`
#>   <chr>                              <dbl> <chr>     <dbl+lbl> <dbl> <dbl> <dbl>
#> 1 Amridge University                100690 AL     12 [12. Cit~   597   294   303
#> 2 Alabama State University          100724 AL     12 [12. Cit~  5318  4727   591
#> 3 The University of Alabama         100751 AL     13 [13. Cit~ 37663 32563  5100
#> 4 Central Alabama Community College 100760 AL     32 [32. Tow~  1769  1769    NA
#> 5 Auburn University at Montgomery   100830 AL     12 [12. Cit~  4878  4273   605
```

Confirm that the new object `agegroup1_obs_tidy` contains one observation for each value of unitid

```
agegroup1_obs_tidy %>% group_by(unitid) %>% # group by vars
  summarise(n_per_group=n()) %>% # create a measure of number of observations per group
  ungroup %>% # ungroup (otherwise frequency table [next step] created) separately for each group
  count(n_per_group) # frequency of number of observations per group
#> # A tibble: 1 x 2
#>   n_per_group     n
#>         <int> <int>
#> 1           1  2944
```

Create a new object `agegroup1_obs_tidy_v2` from the object `agegroup1_obs` by performing the following steps in one line of code with multiple pipes:

- Create a variable `level` that is a character version of the variable 'levstudy'
- Drop the original variable `levstudy`
- Tidy the dataset

```
attributes(agegroup1_obs$levstudy)
#> $label
#> [1] "Level of student"
#>
#> $format.stata
#> [1] "%21.0g"
#>
#> $labels
#> 1. All Students total      2. Undergraduate           5. Graduate
#>                     1                     2                     5
#>
#> $class
#> [1] "haven_labelled" "vctrs_vctr"     "double"

agegroup1_obs_tidy_v2 <- agegroup1_obs %>%
  mutate(level= recode(as.integer(levstudy),
                       `1`= "all",
                       `2`= "ug",
                       `5`= "grad")) %>%
  select(-levstudy) %>%
  pivot_wider(names_from = level, values_from = efage09)
```

Print a few observations of `agegroup1_obs_tidy_v2`; Why is this data frame preferable over `agegroup1_obs_tidy`?

```
- YOUR ANSWER HERE: more intuitive to have variable names that describe the data within that column rath
```

```
agegroup1_obs_tidy_v2 %>% head(n=5)
#> # A tibble: 5 x 7
#>   fullname                          unitid stabbr       locale   all    ug  grad
#>   <chr>                              <dbl> <chr>     <dbl+lbl> <dbl> <dbl> <dbl>
```

```
#> 1 Amridge University                100690 AL     12 [12. Cit~   597   294   303
#> 2 Alabama State University          100724 AL     12 [12. Cit~  5318  4727   591
#> 3 The University of Alabama         100751 AL     13 [13. Cit~ 37663 32563  5100
#> 4 Central Alabama Community College 100760 AL     32 [32. Tow~  1769  1769    NA
#> 5 Auburn University at Montgomery   100830 AL     12 [12. Cit~  4878  4273   605
```

Questions related to reshaping the dataset levstudy1_obs from long to wide

- Run whatever investigations seem helpful to you to get to know the data frame `levstudy1_obs` (e.g., list variable names, list variable variable labels, list variable values, tabulations). You may decide to comment out some of these investigations before you knit and submit the problem set so that your pdf doesn't get too long.

```
#basic investigations of dataset
names(levstudy1_obs)
#> [1] "fullname" "unitid"   "agegroup" "efage09"  "stabbr"   "locale"
str(levstudy1_obs)
#> tibble [36,703 x 6] (S3: tbl_df/tbl/data.frame)
#>  $ fullname: chr [1:36703] "Amridge University" "Amridge University" "Amridge University" "Amridge U
#>   ..- attr(*, "label")= chr "Institution (entity) name"
#>   ..- attr(*, "format.stata")= chr "%91s"
#>  $ unitid  : num [1:36703] 100690 100690 100690 100690 100690 ...
#>   ..- attr(*, "label")= chr "Unique identification number of the institution"
#>   ..- attr(*, "format.stata")= chr "%12.0g"
#>  $ agegroup: dbl+lbl [1:36703]  1,  2,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13,  1, ...
#>    ..@ label       : chr "Age category"
#>    ..@ format.stata: chr "%27.0g"
#>    ..@ labels      : Named num [1:14] 1 2 3 4 5 6 7 8 9 10 ...
#>    .. ..- attr(*, "names")= chr [1:14] "1. All age categories total" "2. Age under 25 total" "3. Age
#>  $ efage09 : num [1:36703] 597 57 7 16 34 540 88 97 110 158 ...
#>   ..- attr(*, "label")= chr "Grand total"
#>   ..- attr(*, "format.stata")= chr "%12.0g"
#>  $ stabbr  : chr [1:36703] "AL" "AL" "AL" "AL" ...
#>   ..- attr(*, "label")= chr "State abbreviation"
#>   ..- attr(*, "format.stata")= chr "%9s"
#>  $ locale  : dbl+lbl [1:36703] 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, ...
#>    ..@ label       : chr "Degree of urbanization (Urban-centric locale)"
#>    ..@ format.stata: chr "%19.0g"
#>    ..@ labels      : Named num [1:13] -3 11 12 13 21 22 23 31 32 33 ...
#>    .. ..- attr(*, "names")= chr [1:13] "-3. {Not available}" "11. City: Large" "12. City: Midsize" "
#>  - attr(*, "label")= chr "dct_ef2016b"
levstudy1_obs %>% var_label()
#> $fullname
#> [1] "Institution (entity) name"
#>
#> $unitid
#> [1] "Unique identification number of the institution"
#>
#> $agegroup
#> [1] "Age category"
#>
#> $efage09
#> [1] "Grand total"
#>
```

```
#> $stabbr
#> [1] "State abbreviation"
#>
#> $locale
#> [1] "Degree of urbanization (Urban-centric locale)"
```

Sort and print a few obs

```
#sort
levstudy1_obs <- levstudy1_obs %>% arrange(unitid,agegroup)

#print a few obs
levstudy1_obs %>% head(n=10) %>% as_factor
#> # A tibble: 10 x 6
#>    fullname           unitid agegroup                    efage09 stabbr locale
#>    <chr>               <dbl> <fct>                         <dbl> <chr>  <fct>
#>  1 Amridge University 100690 1. All age categories total     597 AL     12. Cit~
#>  2 Amridge University 100690 2. Age under 25 total           57 AL     12. Cit~
#>  3 Amridge University 100690 4. Age 18-19                     7 AL     12. Cit~
#>  4 Amridge University 100690 5. Age 20-21                    16 AL     12. Cit~
#>  5 Amridge University 100690 6. Age 22-24                    34 AL     12. Cit~
#>  6 Amridge University 100690 7. Age 25 and over total       540 AL     12. Cit~
#>  7 Amridge University 100690 8. Age 25-29                    88 AL     12. Cit~
#>  8 Amridge University 100690 9. Age 30-34                    97 AL     12. Cit~
#>  9 Amridge University 100690 10. Age 35-39                  110 AL     12. Cit~
#> 10 Amridge University 100690 11. Age 40-49                  158 AL     12. Cit~
```

Run some frequencies

```
#frequency of level of study variable
levstudy1_obs %>% select(agegroup) %>% val_labels()
#> $agegroup
#> 1. All age categories total       2. Age under 25 total
#>                           1                           2
#>             3. Age under 18               4. Age 18-19
#>                           3                           4
#>                5. Age 20-21               6. Age 22-24
#>                           5                           6
#>    7. Age 25 and over total               8. Age 25-29
#>                           7                           8
#>                9. Age 30-34              10. Age 35-39
#>                           9                          10
#>               11. Age 40-49              12. Age 50-64
#>                          11                          12
#>         13. Age 65 and over             14. Age unknown
#>                          13                          14
levstudy1_obs %>% count(agegroup) %>% as_factor
#> # A tibble: 14 x 2
#>    agegroup                       n
#>    <fct>                      <int>
#>  1 1. All age categories total  2944
#>  2 2. Age under 25 total        2936
#>  3 3. Age under 18              2232
#>  4 4. Age 18-19                 2758
#>  5 5. Age 20-21                 2873
```

```
#>  6 6. Age 22-24                 2929
#>  7 7. Age 25 and over total     2936
#>  8 8. Age 25-29                 2931
#>  9 9. Age 30-34                 2905
#> 10 10. Age 35-39                2870
#> 11 11. Age 40-49                2862
#> 12 12. Age 50-64                2732
#> 13 13. Age 65 and over          1962
#> 14 14. Age unknown               833
```

- Confirm that there is one row per each combination of unitid-agegroup

```
levstudy1_obs %>% group_by(unitid,agegroup) %>% # group by vars
  summarise(n_per_group=n()) %>% # create a measure of number of observations per group
  ungroup %>% # ungroup (otherwise frequency table [next step] created) separately for each group
  count(n_per_group) # frequency of number of observations per group
#> `summarise()` has grouped output by 'unitid'. You can override using the
#> `.groups` argument.
#> # A tibble: 1 x 2
#>   n_per_group     n
#>         <int> <int>
#> 1           1 36703
```

Using code from previous question as a guide, confirm that the object `levstudy1_obs` has more than observation for each value of unitid

```
levstudy1_obs %>% group_by(unitid) %>% # group by vars
  summarise(n_per_group=n()) %>% # create a measure of number of observations per group
  ungroup %>% # ungroup (otherwise frequency table [next step] created) separately for each group
  count(n_per_group) # frequency of number of observations per group
#> # A tibble: 11 x 2
#>    n_per_group     n
#>          <int> <int>
#>  1           3     1
#>  2           4     4
#>  3           6     8
#>  4           7     6
#>  5           8    22
#>  6           9    62
#>  7          10   156
#>  8          11   371
#>  9          12   469
#> 10          13  1239
#> 11          14   606
```

- Why is the data frame `levstudy1_obs` not tidy?
  - YOUR ANSWER HERE: the data frame has one row per college-agegroup; these rows do not meet the requirements of being observations because an observation contains all values for some unit.
- What changes need to be made to `levstudy1_obs` to make it tidy?
  - YOUR ANSWER HERE: convert the values of the variable agegroup into their own variables; each variable will contain enrollment for that age group

Tidy the data frame `levstudy1_obs` and create a new object `levstudy1_obs_tidy` (it is up to you whether you want to create character version of the variable `agegroup` prior to tidying) then print a few observations

```
levstudy1_obs %>% head(n=5)
#> # A tibble: 5 x 6
#>   fullname           unitid                    agegroup efage09 stabbr   locale
#>   <chr>               <dbl>                   <dbl+lbl>   <dbl> <chr>  <dbl+lb>
#> 1 Amridge University 100690 1 [1. All age categories to~     597 AL     12 [12.~
#> 2 Amridge University 100690 2 [2. Age under 25 total]         57 AL     12 [12.~
#> 3 Amridge University 100690 4 [4. Age 18-19]                   7 AL     12 [12.~
#> 4 Amridge University 100690 5 [5. Age 20-21]                  16 AL     12 [12.~
#> 5 Amridge University 100690 6 [6. Age 22-24]                  34 AL     12 [12.~
levstudy1_obs %>% count(agegroup) %>% as_factor()
#> # A tibble: 14 x 2
#>    agegroup                        n
#>    <fct>                       <int>
#>  1 1. All age categories total  2944
#>  2 2. Age under 25 total        2936
#>  3 3. Age under 18              2232
#>  4 4. Age 18-19                 2758
#>  5 5. Age 20-21                 2873
#>  6 6. Age 22-24                 2929
#>  7 7. Age 25 and over total     2936
#>  8 8. Age 25-29                 2931
#>  9 9. Age 30-34                 2905
#> 10 10. Age 35-39                2870
#> 11 11. Age 40-49                2862
#> 12 12. Age 50-64                2732
#> 13 13. Age 65 and over          1962
#> 14 14. Age unknown               833

levstudy1_obs_tidy <- levstudy1_obs %>%
  mutate(age = recode(as.integer(agegroup),
    `1`="age_all",
    `2`="age_lt25",
    `3`="age_lt18",
    `4`="age_18_19",
    `5`="age_20_21",
    `6`="age_22_24",
    `7`="age_25_plus",
    `8`="age_25_29",
    `9`="age_30-34",
    `10`="age_35-39",
    `11`="age_40_49",
    `12`="age_50_64",
    `13`="age_65_plus",
    `14`="age_unknown")
  ) %>% select(-agegroup) %>%
  pivot_wider(names_from = age, values_from = efage09)

levstudy1_obs_tidy %>% head(n=5)
#> # A tibble: 5 x 18
#>   fulln~1 unitid stabbr   locale age_all age_l~2 age_1~3 age_2~4 age_2~5 age_2~6
#>   <chr>    <dbl> <chr>  <dbl+lb>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
#> 1 Amridg~ 100690 AL     12 [12.~     597      57       7      16      34     540
#> 2 Alabam~ 100724 AL     12 [12.~    5318    4464    1750    1463    1191     854
```

```
#> 3 The Un~ 100751 AL     13 [13.~   37663   31594   13415   11741    5492    6065
#> 4 Centra~ 100760 AL     32 [32.~    1769    1380     612     379     177     389
#> 5 Auburn~ 100830 AL     12 [12.~    4878    3440    1150    1157    1093    1438
#> # ... with 8 more variables: age_25_29 <dbl>, `age_30-34` <dbl>,
#> #   `age_35-39` <dbl>, age_40_49 <dbl>, age_50_64 <dbl>, age_65_plus <dbl>,
#> #   age_lt18 <dbl>, age_unknown <dbl>, and abbreviated variable names
#> #   1: fullname, 2: age_lt25, 3: age_18_19, 4: age_20_21, 5: age_22_24,
#> #   6: age_25_plus
#> # i Use `colnames()` to see all variable names
```

Confirm that the new object `levstudy1_obs_tidy` contains one observation for each value of unitid

```
levstudy1_obs_tidy %>% group_by(unitid) %>% # group by vars
  summarise(n_per_group=n()) %>% # create a measure of number of observations per group
  ungroup %>% # ungroup (otherwise frequency table [next step] created) separately for each group
  count(n_per_group) # frequency of number of observations per group
#> # A tibble: 1 x 2
#>   n_per_group     n
#>         <int> <int>
#> 1           1  2944
```

Part III: Questions about reshaping wide to long

Here, we load a table from NCES digest of education statistics that contains data about the total number of teachers in each state for particular years.

```
load(url("https://github.com/ksalazar3/HED696C_Rclass/raw/master/data/nces_digest/nces_digest_table_208

#covert character variables for teacher totals to integers
table208_30[2:6] <- data.frame(lapply(table208_30[2:6],as.integer))

table208_30
#> # A tibble: 51 x 6
#>    state                          tot_fall~1 tot_f~2 tot_f~3 tot_f~4 tot_f~5
#>    <chr>                               <int>   <int>   <int>   <int>   <int>
#>  1 Alabama ........................    48194   57757   47492   49363   47722
#>  2 Alaska .........................     7880    7912    8083    8170    8087
#>  3 Arizona ........................    44438   51376   51947   50030   50800
#>  4 Arkansas .......................    31947   32997   37240   34272   33982
#>  5 California .....................   298021  309222  316298  260806  268688
#>  6 Colorado .......................    41983   45841   49060   48542   48077
#>  7 Connecticut ....................    41044   39687   43592   42951   43804
#>  8 Delaware .......................     7469    7998    8639    8933    8587
#>  9 District of Columbia ...........     4949    5481    5854    5925    6278
#> 10 Florida ........................   132030  158962  183827  175609  175006
#> # ... with 41 more rows, and abbreviated variable names 1: tot_fall_2000,
#> #   2: tot_fall_2005, 3: tot_fall_2009, 4: tot_fall_2010, 5: tot_fall_2011
#> # i Use `print(n = ...)` to see more rows
```

- Why is the data frame `table208_30` not tidy?
  - YOUR ANSWER HERE: Some of the column names (tot_fall_2000...) are not names of variables, but values of a variable, which results in a single variable (e.g., total fall enrollment) being spread across multiple columns.
- What changes need to be made to `table208_30` to make it tidy?

- YOUR ANSWER HERE: Create `year` column or reshape from wide to long

Tidy the data frame `table208_30` and create a new object `table208_30_tidy`:

- hint: use the `cols = starts_with()` and `names_prefix=()` options for `pivot_longer()`
- after you tidy the data, print a few observations

```
table208_30_tidy<- table208_30 %>%
  pivot_longer(
    cols = starts_with("tot_fall_"),
    names_to = "year",
    names_prefix = ("tot_fall_"),
    values_to = "tot_tchrs"
  )

#examine data
head(table208_30_tidy, n=20)
#> # A tibble: 20 x 3
#>    state                          year  tot_tchrs
#>    <chr>                          <chr>     <int>
#>  1 Alabama ....................... 2000      48194
#>  2 Alabama ....................... 2005      57757
#>  3 Alabama ....................... 2009      47492
#>  4 Alabama ....................... 2010      49363
#>  5 Alabama ....................... 2011      47722
#>  6 Alaska ........................  2000       7880
#>  7 Alaska ........................  2005       7912
#>  8 Alaska ........................  2009       8083
#>  9 Alaska ........................  2010       8170
#> 10 Alaska ........................  2011       8087
#> 11 Arizona .......................  2000      44438
#> 12 Arizona .......................  2005      51376
#> 13 Arizona .......................  2009      51947
#> 14 Arizona .......................  2010      50030
#> 15 Arizona .......................  2011      50800
#> 16 Arkansas ......................  2000      31947
#> 17 Arkansas ......................  2005      32997
#> 18 Arkansas ......................  2009      37240
#> 19 Arkansas ......................  2010      34272
#> 20 Arkansas ......................  2011      33982
```

Once finished, knit to (pdf) and upload both .Rmd and pdf files to D2L