Lecture 6 problem set

INSERT YOUR NAME HERE

November 9, 2018

Contents

Required reading and instructions

Required reading before next class

- Work through slides from lecture 6 that we don't get to in class
  - [REQUIRED] slides from section 5 "Missing data"
- [REQUIRED] R Pivot Blog
  - https://tidyr.tidyverse.org/dev/articles/pivot.html
- [OPTIONAL] GW chapter 12 (tidy data)
  - Lecture 6 covers this material pretty closely, so read chapter if you can, but I get it if you don't have time
- [OPTIONAL] Wickham, H. (2014). Tidy Data. *Journal of Statistical Software*, 59(10), 1-23. doi: 10.18637/jss.v059.i10
  - This is the journal article that introduced the data concepts covered in GW chapter 12 and created the packages related to tidying data

Mid-quarter evaluation

- Please take 10 minutes to complete the anonymous mid-quarter evaluation Here

---

Overview

This problem set has three parts.

1. I'll ask you some definitional/conceptual questions about the concepts introduced in lecture
2. Tidying untidy data: reshaping from long to wide
   - this will be the longest part of the problem set because it is very common that data we find "in the wild" needs to be reshaped long to wide before it is tidy
     - 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
   - This section will be short because it is less common that datasets need to be "gathered" before they are tidy

Load library and data

```
#install.packages("tidyverse") #uncomment if you haven't installed these packaged
#install.packages("haven")
#install.packages("labelled")
library(tidyverse)
#> -- Attaching packages ------------------------------------------------------------------------------
#> v ggplot2 3.2.1          v purrr   0.3.2
#> v tibble  2.1.3          v dplyr   0.8.3
#> v tidyr   1.0.0.9000     v stringr 1.4.0
#> v readr   1.3.1          v forcats 0.4.0
#> -- 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: **/0.5**

Wickham defines "observational level" as what each observation should represent in a tidy dataset (i.e., it is a data concept), whereas Ozan defines "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?

  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

Overview of the reshaping long to wide tasks

- Load the data frame and assign it the name `age_f16_allvars_allobs`
- Create three different data frame objects based on the data frame `age_f16_allvars_allobs`
  - A dataframe `all_obs` that has fewer variables than `age_f16_allvars_allobs` but the same number of observations
    - ∗ this data frame has the most complex structure; we'll reshape this one last
  - 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 simplist 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)
    - ∗ this data frame has the second simplist structure; we'll reshape this one second
- Questions related to reshaping `agegroup1_obs`
- Questions related to reshaping `levstudy1_obs`
- Questions related to reshaping `all_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/ozanj/rclass/raw/master/data/ipeds/ef/age/ef

#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"
#>  [5] "efage02"      "efage03"      "efage04"      "efage05"
#>  [9] "efage06"      "efage07"      "efage08"      "efage09"
#> [13] "fullname"     "stabbr"       "sector"       "iclevel"
#> [17] "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 three 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 has fewer variables than `age_f16_allvars_allobs` but the same number of observa
all_obs <- age_f16_allvars_allobs %>%
  select(fullname,unitid,agegroup,levstudy,efage09,stabbr,sector,locale)

glimpse(all_obs)
#> Observations: 85,129
#> Variables: 8
#> $ fullname <chr> "Amridge University", "Amridge University", "Amridge ...
#> $ unitid   <dbl> 100690, 100690, 100690, 100690, 100690, 100690, 10069...
#> $ agegroup <dbl+lbl> 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1, 2, 4, ...
#> $ levstudy <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2...
#> $ efage09  <dbl> 597, 57, 7, 16, 34, 540, 88, 97, 110, 158, 78, 9, 294...
#> $ stabbr   <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL",...
#> $ sector   <dbl+lbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2...
#> $ locale   <dbl+lbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 1...

#Create dataframe that keeps observations where age-group equals `1` (1. All age categories total)
agegroup1_obs <- all_obs %>%
  filter(agegroup==1) %>% select(-agegroup)

glimpse(agegroup1_obs)
#> Observations: 7,019
#> Variables: 7
#> $ fullname <chr> "Amridge University", "Amridge University", "Amridge ...
#> $ unitid   <dbl> 100690, 100690, 100690, 100724, 100724, 100724, 10075...
#> $ levstudy <dbl+lbl> 1, 2, 5, 1, 2, 5, 1, 2, 5, 1, 2, 1, 2, 5, 1, 2, 5...
#> $ efage09  <dbl> 597, 294, 303, 5318, 4727, 591, 37663, 32563, 5100, 1...
#> $ stabbr   <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL",...
#> $ sector   <dbl+lbl> 2, 2, 2, 1, 1, 1, 1, 1, 1, 4, 4, 1, 1, 1, 1, 1, 1...
#> $ locale   <dbl+lbl> 12, 12, 12, 12, 12, 12, 13, 13, 13, 32, 32, 12, 1...

#Create dataframe keeps observations where "level of study" equals `1` (1. All Students total)
levstudy1_obs <- all_obs %>%
  filter(levstudy==1) %>% select(-levstudy)

glimpse(levstudy1_obs)
#> Observations: 36,703
#> Variables: 7
#> $ fullname <chr> "Amridge University", "Amridge University", "Amridge ...
#> $ unitid   <dbl> 100690, 100690, 100690, 100690, 100690, 100690, 10069...
#> $ agegroup <dbl+lbl> 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1, 2, 3, ...
#> $ efage09  <dbl> 597, 57, 7, 16, 34, 540, 88, 97, 110, 158, 78, 9, 531...
#> $ stabbr   <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL",...
#> $ sector   <dbl+lbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1...
#> $ locale   <dbl+lbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 1...
```

Questions related to spreading the dataset `agegroup1_obs`

- 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"   "sector"
#> [7] "locale"
str(agegroup1_obs)
#> Classes 'tbl_df', 'tbl' and 'data.frame':    7019 obs. of  7 variables:
#>  $ fullname: chr  "Amridge University" "Amridge University" "Amridge University" "Alabama State Univ
#>   ..- attr(*, "label")= chr "Institution (entity) name"
#>   ..- attr(*, "format.stata")= chr "%91s"
#>  $ unitid  : num  100690 100690 100690 100724 100724 ...
#>   ..- attr(*, "label")= chr "Unique identification number of the institution"
#>   ..- attr(*, "format.stata")= chr "%12.0g"
#>  $ levstudy: 'haven_labelled' num  1 2 5 1 2 5 1 2 5 1 ...
#>   ..- attr(*, "label")= chr "Level of student"
#>   ..- attr(*, "labels")= Named num  1 2 5
#>   .. ..- attr(*, "names")= chr  "1. All Students total" "2. Undergraduate" "5. Graduate"
#>  $ efage09 : num  597 294 303 5318 4727 ...
#>   ..- attr(*, "label")= chr "Grand total"
#>   ..- attr(*, "format.stata")= chr "%12.0g"
#>  $ stabbr  : chr  "AL" "AL" "AL" "AL" ...
#>   ..- attr(*, "label")= chr "State abbreviation"
#>   ..- attr(*, "format.stata")= chr "%9s"
#>  $ sector  : 'haven_labelled' num  2 2 2 1 1 1 1 1 1 4 ...
#>   ..- attr(*, "label")= chr "Sector of institution"
#>   ..- attr(*, "labels")= Named num  0 1 2 3 4 5 6 7 8 9 ...
#>   .. ..- attr(*, "names")= chr  "0. Administrative Unit" "1. Public, 4-year or above" "2. Private no
#>  $ locale  : 'haven_labelled' num  12 12 12 12 12 12 13 13 13 32 ...
#>   ..- attr(*, "label")= chr "Degree of urbanization (Urban-centric locale)"
#>   ..- attr(*, "labels")= Named num  -3 11 12 13 21 22 23 31 32 33 ...
#>   .. ..- attr(*, "names")= chr  "-3. {Not available}" "11. City: Large" "12. City: Midsize" "13. Cit
#>  - 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"
#> 
#> $sector
#> [1] "Sector of institution"
```

```
#>
#> $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 7
#>    fullname       unitid levstudy    efage09 stabbr sector         locale
#>    <chr>           <dbl> <fct>         <dbl> <chr>  <fct>          <fct>
#>  1 Amridge Unive~ 100690 1. All Stu~     597 AL     2. Private no~ 12. Cit~
#>  2 Amridge Unive~ 100690 2. Undergr~     294 AL     2. Private no~ 12. Cit~
#>  3 Amridge Unive~ 100690 5. Graduate     303 AL     2. Private no~ 12. Cit~
#>  4 Alabama State~ 100724 1. All Stu~    5318 AL     1. Public, 4-~ 12. Cit~
#>  5 Alabama State~ 100724 2. Undergr~    4727 AL     1. Public, 4-~ 12. Cit~
#>  6 Alabama State~ 100724 5. Graduate     591 AL     1. Public, 4-~ 12. Cit~
#>  7 The Universit~ 100751 1. All Stu~   37663 AL     1. Public, 4-~ 13. Cit~
#>  8 The Universit~ 100751 2. Undergr~   32563 AL     1. Public, 4-~ 13. Cit~
#>  9 The Universit~ 100751 5. Graduate    5100 AL     1. Public, 4-~ 13. Cit~
#> 10 Central Alaba~ 100760 1. All Stu~    1769 AL     4. Public, 2-~ 32. Tow~
```

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 sector variable
agegroup1_obs %>% select(sector) %>% val_labels()
#> $sector
#>                       0. Administrative Unit
#>                                            0
#>                   1. Public, 4-year or above
#>                                            1
#>   2. Private not-for-profit, 4-year or above
#>                                            2
#>       3. Private for-profit, 4-year or above
#>                                            3
#>                            4. Public, 2-year
#>                                            4
#>            5. Private not-for-profit, 2-year
#>                                            5
#>               6. Private for-profit, 2-year 
#>                                           6 
#>                  7. Public, less-than 2-year 
#>                                           7 
#> 8. Private not-for-profit, less-than 2-year 
#>                                           8 
#>     9. Private for-profit, less-than 2-year 
#>                                           9 
#>             99. Sector unknown (not active) 
#>                                          99 
agegroup1_obs %>% count(sector) %>% as_factor
#> # A tibble: 9 x 2
#>   sector                                          n
#>   <fct>                                       <int>
#> 1 1. Public, 4-year or above                   1701
#> 2 2. Private not-for-profit, 4-year or above   2082
#> 3 3. Private for-profit, 4-year or above        608
#> 4 4. Public, 2-year                            1370
#> 5 5. Private not-for-profit, 2-year              96
#> 6 6. Private for-profit, 2-year                 430
#> 7 7. Public, less-than 2-year                    80
#> 8 8. Private not-for-profit, less-than 2-year    30
#> 9 9. Private for-profit, less-than 2-year       622

#frequency of locale variable
agegroup1_obs %>% select(locale) %>% val_labels()
#> $locale
#> -3. {Not available}     11. City: Large   12. City: Midsize 
#>                  -3                  11                  12 
#>     13. City: Small   21. Suburb: Large 22. Suburb: Midsize 
#>                  13                  21                  22 
#>   23. Suburb: Small    31. Town: Fringe   32. Town: Distant 
#>                  23                  31                  32 
#>    33. Town: Remote   41. Rural: Fringe  42. Rural: Distant 
#>                  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
#> # 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 ANSWER HERE:
    - Each variable must have its own column: false; the values of the column levstudy should each be
    - Each observation must have its own row: false; there should be one row per college/university,
    - Each value must have its own cell: true
```

- what changes need to be made to `age_all` 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 "spreading" to tidy a dataset, define the concept "key column"
    - YOUR ANSWER HERE: Column name in the untidy data whose values will become variable names in the tidy data
- What should the key column be in the data frame `agegroup1_obs`?
    - YOUR ANSWER HERE: key column should be `levstudy`
- With respect to "spreading" to tidy a dataset, define the concept "value column"
    - YOUR ANSWER HERE: Column name in untidy data that contains values for the new variables that will be created in the tidy data
- what should the value column be in the data frame `agegroup1_obs`?
    - YOUR ANSWER HERE: value column should be `efage09`

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

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

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

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

```
- YOUR ANSWER HERE:
```

Questions related to spreading the dataset levstudy1_obs

- 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.

Sort and print a few obs

Run some frequencies

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

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

- Why is the data frame `levstudy1_obs` not tidy?
    - YOUR ANSWER HERE:
- What changes need to be made to `levstudy1_obs` to make it tidy?
    - YOUR ANSWER HERE:

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

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

Questions related to spreading the dataset all_obs

Investigate data frame `all_obs` if you want, but not required to show code

- Confirm that there is one row per each combination of unitid-agegroup-levstudy
- Why is the data frame `all_obs` not tidy?
    - YOUR ANSWER HERE:
- What changes need to be made to `all_obs` to make it tidy?
    - YOUR ANSWER HERE:
- The `spread()` function can only have a single key variable. we have two key variables: `agegroup` and `level`. Run the below code, which creates character versions of these two variables and then uses the `unit()` function to combine these two variables into a single variable. This code will create a new object all_obs_temp.

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

```r
all_obs_temp <- all_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"),
  level=recode(as.integer(levstudy),
    `1` = "lev_all",
    `2` = "lev_ug",
    `5` = "lev_grad")
  ) %>% unite("age_lev", age, level) %>%
  select(-levstudy,-agegroup)

all_obs_temp %>% head(n=20)
#> # A tibble: 20 x 7
#>    fullname    unitid efage09 stabbr           sector      locale age_lev
#>    <chr>        <dbl>   <dbl> <chr>         <dbl+lbl>   <dbl+lbl> <chr>
#>  1 Amridge Un~ 100690     597 AL     2 [2. Private n~ 12 [12. Ci~ age_all_~
#>  2 Amridge Un~ 100690      57 AL     2 [2. Private n~ 12 [12. Ci~ age_lt25~
#>  3 Amridge Un~ 100690       7 AL     2 [2. Private n~ 12 [12. Ci~ age_18_1~
#>  4 Amridge Un~ 100690      16 AL     2 [2. Private n~ 12 [12. Ci~ age_20_2~
#>  5 Amridge Un~ 100690      34 AL     2 [2. Private n~ 12 [12. Ci~ age_22_2~
#>  6 Amridge Un~ 100690     540 AL     2 [2. Private n~ 12 [12. Ci~ age_25_p~
#>  7 Amridge Un~ 100690      88 AL     2 [2. Private n~ 12 [12. Ci~ age_25_2~
#>  8 Amridge Un~ 100690      97 AL     2 [2. Private n~ 12 [12. Ci~ age_30-3~
#>  9 Amridge Un~ 100690     110 AL     2 [2. Private n~ 12 [12. Ci~ age_35-3~
#> 10 Amridge Un~ 100690     158 AL     2 [2. Private n~ 12 [12. Ci~ age_40_4~
#> 11 Amridge Un~ 100690      78 AL     2 [2. Private n~ 12 [12. Ci~ age_50_6~
#> 12 Amridge Un~ 100690       9 AL     2 [2. Private n~ 12 [12. Ci~ age_65_p~
#> 13 Amridge Un~ 100690     294 AL     2 [2. Private n~ 12 [12. Ci~ age_all_~
#> 14 Amridge Un~ 100690      46 AL     2 [2. Private n~ 12 [12. Ci~ age_lt25~
#> 15 Amridge Un~ 100690       7 AL     2 [2. Private n~ 12 [12. Ci~ age_18_1~
#> 16 Amridge Un~ 100690      15 AL     2 [2. Private n~ 12 [12. Ci~ age_20_2~
#> 17 Amridge Un~ 100690      24 AL     2 [2. Private n~ 12 [12. Ci~ age_22_2~
#> 18 Amridge Un~ 100690     248 AL     2 [2. Private n~ 12 [12. Ci~ age_25_p~
#> 19 Amridge Un~ 100690      45 AL     2 [2. Private n~ 12 [12. Ci~ age_25_2~
#> 20 Amridge Un~ 100690      47 AL     2 [2. Private n~ 12 [12. Ci~ age_30-3~
```

Tidy the data frame `all_obs_temp` and create a new object `all_obs_tidy`; then print a few observations

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

Part III: Questions about gathering

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/ozanj/rclass/raw/master/data/nces_digest/nces_digest_table_208_30.RData"))
table208_30
#> # A tibble: 51 x 6
#>    state tot_fall_2000 tot_fall_2005 tot_fall_2009 tot_fall_2010
#>    <chr> <chr>         <chr>         <chr>         <chr>
#>  1 Alab~ 48194.400000~ 57757         47492         49363.240000~
#>  2 Alas~ 7880.3999999~ 7912          8083.1000000~ 8170.6399999~
#>  3 Ariz~ 44438.400000~ 51376         51947.230000~ 50030.619999~
#>  4 Arka~ 31947.400000~ 32997         37240         34272.800000~
#>  5 Cali~ 298021.40000~ 309222        316298.58000~ 260806.29999~
#>  6 Colo~ 41983.400000~ 45841         49060.32      48542.990000~
#>  7 Conn~ 41044.400000~ 39687         43592.829999~ 42951.389999~
#>  8 Dela~ 7469.3999999~ 7998          8639.5799999~ 8933
#>  9 Dist~ 4949.3999999~ 5481          5854          5925.3299999~
#> 10 Flor~ 132030.39999~ 158962        183827        175609.28999~
#> # ... with 41 more rows, and 1 more variable: tot_fall_2011 <chr>
```

- Why is the data frame `table208_30` not tidy?
  - YOUR ANSWER HERE:
- What changes need to be made to `table208_30` to make it tidy?
  - YOUR ANSWER HERE:

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

- Recommended but optional: prior to gathering, rename the **names** columns (i.e., the set of columns that represent values, not variables in your untidy data). Specifically, rename these variables to remove characters prior to gathering (e.g., rename "tot_fall_2000" -> "2000"). See the end of section 4.2.1 for an example of how to do this.
- after you tidy the data, print a few observations

Bonus Question:

Run this code below to create the data frame `allobs_v1` and examine its contents

```
names(age_f16_allvars_allobs)
#>  [1] "unitid"       "agegroup"     "levstudy"     "efage01"
#>  [5] "efage02"      "efage03"      "efage04"      "efage05"
#>  [9] "efage06"      "efage07"      "efage08"      "efage09"
#> [13] "fullname"     "stabbr"       "sector"       "iclevel"
#> [17] "control"      "hloffer"      "locale"       "merge_age_ic"
#age_f16_allvars_allobs %>% var_label()

allobs_v1 <- age_f16_allvars_allobs %>%
  select(1:9, 13:19)
names(allobs_v1)
#>  [1] "unitid"   "agegroup" "levstudy" "efage01"  "efage02"  "efage03"
#>  [7] "efage04"  "efage05"  "efage06"  "fullname" "stabbr"   "sector"
#> [13] "iclevel"  "control"  "hloffer"  "locale"
allobs_v1
#> # A tibble: 85,129 x 16
#>    unitid agegroup levstudy efage01 efage02 efage03 efage04 efage05 efage06
#>     <dbl> <dbl+lb> <dbl+lb>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
#>  1 100690  1 [1. ~ 1 [1. A~      89     127     144     237     216     381
#>  2 100690  2 [2. ~ 1 [1. A~       9      14      12      22      23      34
#>  3 100690  4 [4. ~ 1 [1. A~       1       2       1       3       3       4
#>  4 100690  5 [5. ~ 1 [1. A~       3       6       5       2       9       7
#>  5 100690  6 [6. ~ 1 [1. A~       5       6       6      17      11      23
#>  6 100690  7 [7. ~ 1 [1. A~      80     113     132     215     193     347
#>  7 100690  8 [8. ~ 1 [1. A~      12      26      16      34      38      50
#>  8 100690  9 [9. ~ 1 [1. A~      22      20      19      36      42      55
#>  9 100690 10 [10.~ 1 [1. A~      15      20      23      52      35      75
#> 10 100690 11 [11.~ 1 [1. A~      22      33      46      57      55     103
#> # ... with 85,119 more rows, and 7 more variables: fullname <chr>,
#> #   stabbr <chr>, sector <dbl+lbl>, iclevel <dbl+lbl>, control <dbl+lbl>,
#> #   hloffer <dbl+lbl>, locale <dbl+lbl>
```

Your task in this bonus question is to make the untidy data frame `allobs_v1` tidy. note that `allobs_v1` contains multiple enrollment variables (in addition to the variables `efbage` and `lstudy` which were in the previous data frames we tidied.

The end of Section 4.3 "Tidying data: spreading" of Lecture 6 states that the `spread()` function is not designed to create tidy datasets when there are multiple **value** variables. Therefore, in order to spread to create a tidy dataset from an untidy dataset that has multiple **value** variables, we would need to incorporate additional/alternative programming skills **not taught** in class. And that is why this is a bonus question.

Your end result should be a "tidy" version of `allobs_tidy`.
Hint: Google "How to spread mulitple value columns in R"

Once finished, knit to (pdf) and upload both .Rmd and pdf files to class website under the week 6 tab
Remeber to use this naming convention "lastname_firstname_ps6"