Lecture 9 problem set

INSERT YOUR NAME HERE

INSERT DATE HERE

Contents

Grade: /20

Required reading and instructions

Required reading before next class

- Work through material from lecture 9 that we don't get to in class
  - [REQUIRED] slides from section 4 "Tidying data", particularly 4.2 "gathering"
  - [OPTIONAL] slides from section 5 "Missing data"
- [OPTIONAL] GW chapter 12 (tidy data)
  - Lecture 9 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

General Problem Set instructions

In this homework, you will specify `pdf_document` as the output format. You must have LaTeX installed in order to create pdf documents.

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: "spreading" (i.e., going 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 "spread" 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: "gathering" (i.e., going 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 --------------------------------- tidyverse 1.3.0 --
#> v ggplot2 3.3.2     v purrr   0.3.4
#> v tibble  3.0.3     v dplyr   1.0.2
#> v tidyr   1.1.1     v stringr 1.4.0
#> v readr   1.3.1     v forcats 0.5.0
#> -- Conflicts ------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
library(haven)
library(labelled)
```

Part I: Conceptual questions

- According to Wickham, what is the difference between "data structure" and "data concepts" (he uses the term "data semantics")
  /1
    - Data structure refers to the the physical layout of the data (e.g., what the rows and columns in a dataset actually represent)
    - Data concepts – which were introduced by Wickham (2014) – refer to how the data should be structured
- According to Wickham:
  /1

  - what is an "observation"?
    ANSWER: An observation contains the values for all attributes measured on the same unit (like a person, or a day)...across attributes"
  - give an example of an observation?
    ANSWER: Imagine a dataset consisting of demographic/socioeconomic data about 6th graders (e.g., age, address, parental education). An observation would contain the value of all attributes for one 6th grader
  - What is the difference between an "observation" and a "row"?
    ANSWER: A row refers to the physical layout of a dataset (e.g., one row consisting of cells in that row) but there are no rules about the kind of information contained in the row; by contrast an observation contains the values of all attributes for a particular observational unit (e.g., person, organization-year)
  - Under what condition is an observation the same thing as a row?
    ANSWER: When data is tidy (satisfies all three conditions of tidy data)
- According to Wickham:
  /1
  - what is a "variable?"
    ANSWER: "A variable contains all values that measure the same underlying attribute (like height, temperature, duration) across units"
  - give an example of a variable
    ANSWER: Height, weight, or age for all students in a dataset that contains demographic data on 6th graders; note that a variable could be represented by two columns, but in a tidy dataset, each variable must be contained within one column
  - what is the difference between a "variable" and a "column"
    ANSWER: A variable contains the values of an attribute for all observational units in a dataset; by contrast a column just refers to physical structure of the data and there are no rules about what kind of information belongs in a column
  - Under what condition is a variable the same thing as a column
    ANSWER: When data is tidy
- According to Wickham:
  /1
  - what is a "value"?
    ANSWER: "A single element within some data structure (e.g., vector, list), usually a number or a character string."
  - give an example of a value
    ANSWER: The value of the variable height for one person in a dataset where each observation represents a person
  - what is the difference between a "cell" and a value?
    ANSWER: A cell is just the contents of the intersection of one row and one column; by contrast, a value represents the value of one attribute for one observational unit
  - Under what condition is a value the same thing as a cell?
    When data is tidy
- What is the difference between the terms "unit of analysis" [an "ozan" term; not necessarily used outside this class] and "observational level" [A Wickham term]
  /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?
  /0.5

  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 spreading

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 contains 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 spreading 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 spread 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 spread 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 spread this one second
- Questions related to spreading `agegroup1_obs`
- Questions related to spreading `levstudy1_obs`
- Questions related to spreading `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/e

#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 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)
#> Rows: 85,129
#> Columns: 8
#> $ fullname <chr> "Amridge University", "Amridge University", "Amridge Unive...
#> $ unitid   <dbl> 100690, 100690, 100690, 100690, 100690, 100690, 100690, 10...
#> $ agegroup <dbl+lbl>  1,  2,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13,  1,  2...
#> $ levstudy <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, ...
#> $ efage09  <dbl> 597, 57, 7, 16, 34, 540, 88, 97, 110, 158, 78, 9, 294, 46,...
#> $ stabbr   <chr> "AL", "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, 2, 2, ...
#> $ locale   <dbl+lbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12...

#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)
#> Rows: 7,019
#> Columns: 7
#> $ fullname <chr> "Amridge University", "Amridge University", "Amridge Unive...
#> $ unitid   <dbl> 100690, 100690, 100690, 100724, 100724, 100724, 100751, 10...
#> $ levstudy <dbl+lbl> 1, 2, 5, 1, 2, 5, 1, 2, 5, 1, 2, 1, 2, 5, 1, 2, 5, 1, ...
#> $ efage09  <dbl> 597, 294, 303, 5318, 4727, 591, 37663, 32563, 5100, 1769, ...
#> $ stabbr   <chr> "AL", "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, 4, ...
#> $ locale   <dbl+lbl> 12, 12, 12, 12, 12, 12, 13, 13, 13, 32, 32, 12, 12, 12...

#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)
```

```
#> Rows: 36,703
#> Columns: 7
#> $ fullname <chr> "Amridge University", "Amridge University", "Amridge Unive...
#> $ unitid   <dbl> 100690, 100690, 100690, 100690, 100690, 100690, 100690, 10...
#> $ 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, 44...
#> $ stabbr   <chr> "AL", "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, 1, ...
#> $ locale   <dbl+lbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12...
```

Questions related to spreading the dataset `agegroup1_obs`

/0.5

- 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"   "locale"
str(agegroup1_obs)
#> tibble [7,019 x 7] (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"
#>  $ sector  : dbl+lbl [1:7019] 2, 2, 2, 1, 1, 1, 1, 1, 1, 4, 4, 1, 1, 1, 1, 1, 1, 4,...
#>    ..@ label       : chr "Sector of institution"
#>    ..@ format.stata: chr "%43.0g"
#>    ..@ labels      : Named num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
#>    .. ..- attr(*, "names")= chr [1:11] "0. Administrative Unit" "1. Public, 4-year or above" "2. Pri
#>  $ 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"
#>
#> $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 Univers~ 100690 1. All Stu~     597 AL     2. Private not-~ 12. City~
#>  2 Amridge Univers~ 100690 2. Undergr~     294 AL     2. Private not-~ 12. City~
#>  3 Amridge Univers~ 100690 5. Graduate     303 AL     2. Private not-~ 12. City~
#>  4 Alabama State U~ 100724 1. All Stu~    5318 AL     1. Public, 4-ye~ 12. City~
#>  5 Alabama State U~ 100724 2. Undergr~    4727 AL     1. Public, 4-ye~ 12. City~
#>  6 Alabama State U~ 100724 5. Graduate     591 AL     1. Public, 4-ye~ 12. City~
#>  7 The University ~ 100751 1. All Stu~   37663 AL     1. Public, 4-ye~ 13. City~
#>  8 The University ~ 100751 2. Undergr~   32563 AL     1. Public, 4-ye~ 13. City~
#>  9 The University ~ 100751 5. Graduate    5100 AL     1. Public, 4-ye~ 13. City~
#> 10 Central Alabama~ 100760 1. All Stu~    1769 AL     4. Public, 2-ye~ 32. Town~
```

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     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()` regrouping output by 'unitid' (override with `.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
/0.5

```
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
#> `summarise()` ungrouping output (override with `.groups` argument)
#> # 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
  /2
  - YOUR ANSWER 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 `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

/1

```
agegroup1_obs %>% head(n=5)
#> # A tibble: 5 x 7
#>   fullname      unitid      levstudy efage09 stabbr           sector      locale
#>   <chr>          <dbl>     <dbl+lbl>   <dbl> <chr>         <dbl+lbl>   <dbl+lbl>
#> 1 Amridge Univ~ 100690 1 [1. All St~     597 AL     2 [2. Private n~ 12 [12. Ci~
#> 2 Amridge Univ~ 100690 2 [2. Underg~     294 AL     2 [2. Private n~ 12 [12. Ci~
#> 3 Amridge Univ~ 100690 5 [5. Gradua~     303 AL     2 [2. Private n~ 12 [12. Ci~
#> 4 Alabama Stat~ 100724 1 [1. All St~    5318 AL     1 [1. Public, 4~ 12 [12. Ci~
#> 5 Alabama Stat~ 100724 2 [2. Underg~    4727 AL     1 [1. Public, 4~ 12 [12. Ci~
agegroup1_obs_tidy <- agegroup1_obs %>% spread(key = levstudy, value = efage09)
agegroup1_obs_tidy %>% head(n=5)
#> # A tibble: 5 x 8
#>   fullname        unitid stabbr            sector      locale   `1`   `2`   `5`
#>   <chr>            <dbl> <chr>          <dbl+lbl>   <dbl+lbl> <dbl> <dbl> <dbl>
#> 1 Amridge Univer~ 100690 AL     2 [2. Private not~ 12 [12. Ci~   597   294   303
#> 2 Alabama State ~ 100724 AL     1 [1. Public, 4-y~ 12 [12. Ci~  5318  4727   591
#> 3 The University~ 100751 AL     1 [1. Public, 4-y~ 13 [13. Ci~ 37663 32563  5100
#> 4 Central Alabam~ 100760 AL     4 [4. Public, 2-y~ 32 [32. To~  1769  1769    NA
#> 5 Auburn Univers~ 100830 AL     1 [1. Public, 4-y~ 12 [12. Ci~  4878  4273   605
```

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

/0.5

```
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
#> `summarise()` ungrouping output (override with `.groups` argument)
#> # 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:

/1.5

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

```
agegroup1_obs_tidy_v2 <- agegroup1_obs %>%
  mutate(level=recode(as.integer(levstudy),
    `1` = "all",
    `2` = "ug",
    `5` = "grad")
  ) %>% select(-levstudy) %>% # drop variable lstudy
  spread(key = level, value = efage09)
```

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

/0.5

```
head(agegroup1_obs_tidy_v2)
#> # A tibble: 6 x 8
#>   fullname        unitid stabbr             sector      locale   all  grad    ug
#>   <chr>            <dbl> <chr>           <dbl+lbl>   <dbl+lbl> <dbl> <dbl> <dbl>
#> 1 Amridge Univer~ 100690 AL     2 [2. Private not~ 12 [12. Ci~   597   303   294
#> 2 Alabama State ~ 100724 AL     1 [1. Public, 4-y~ 12 [12. Ci~  5318   591  4727
#> 3 The University~ 100751 AL     1 [1. Public, 4-y~ 13 [13. Ci~ 37663  5100 32563
#> 4 Central Alabam~ 100760 AL     4 [4. Public, 2-y~ 32 [32. To~  1769    NA  1769
#> 5 Auburn Univers~ 100830 AL     1 [1. Public, 4-y~ 12 [12. Ci~  4878   605  4273
#> 6 Auburn Univers~ 100858 AL     1 [1. Public, 4-y~ 13 [13. Ci~ 28290  5632 22658
```

YOUR ANSWER HERE: more intuitive to have variable names that are not numbers

Questions related to spreading the dataset levstudy1_obs

/0.5

- 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"   "sector"   "locale"
str(levstudy1_obs)
#> tibble [36,703 x 7] (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"
```

```
#>  $ sector  : dbl+lbl [1:36703] 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1...
#>    ..@ label       : chr "Sector of institution"
#>    ..@ format.stata: chr "%43.0g"
#>    ..@ labels      : Named num [1:11] 0 1 2 3 4 5 6 7 8 9 ...
#>    .. ..- attr(*, "names")= chr [1:11] "0. Administrative Unit" "1. Public, 4-year or above" "2. Pri
#>  $ locale  : dbl+lbl [1:36703] 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"
#>
#> $sector
#> [1] "Sector of institution"
#>
#> $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 7
#>    fullname     unitid agegroup       efage09 stabbr sector            locale
#>    <chr>         <dbl> <fct>            <dbl> <chr>  <fct>             <fct>
#>  1 Amridge Uni~ 100690 1. All age ca~     597 AL     2. Private not-f~ 12. City~
#>  2 Amridge Uni~ 100690 2. Age under ~      57 AL     2. Private not-f~ 12. City~
#>  3 Amridge Uni~ 100690 4. Age 18-19         7 AL     2. Private not-f~ 12. City~
#>  4 Amridge Uni~ 100690 5. Age 20-21        16 AL     2. Private not-f~ 12. City~
#>  5 Amridge Uni~ 100690 6. Age 22-24        34 AL     2. Private not-f~ 12. City~
#>  6 Amridge Uni~ 100690 7. Age 25 and~     540 AL     2. Private not-f~ 12. City~
#>  7 Amridge Uni~ 100690 8. Age 25-29        88 AL     2. Private not-f~ 12. City~
#>  8 Amridge Uni~ 100690 9. Age 30-34        97 AL     2. Private not-f~ 12. City~
#>  9 Amridge Uni~ 100690 10. Age 35-39      110 AL     2. Private not-f~ 12. City~
#> 10 Amridge Uni~ 100690 11. Age 40-49      158 AL     2. Private not-f~ 12. City~
```

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
  /0.5

```
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()` regrouping output by 'unitid' (override with `.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
/0.5

```
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
#> `summarise()` ungrouping output (override with `.groups` argument)
#> # 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
```

/0.5

- 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

/1.5

```
levstudy1_obs %>% head(n=5)
#> # A tibble: 5 x 7
#>   fullname    unitid       agegroup efage09 stabbr            sector      locale
#>   <chr>        <dbl>      <dbl+lbl>   <dbl> <chr>          <dbl+lbl>   <dbl+lbl>
#> 1 Amridge Un~ 100690 1 [1. All age~     597 AL     2 [2. Private no~ 12 [12. Ci~
#> 2 Amridge Un~ 100690 2 [2. Age und~      57 AL     2 [2. Private no~ 12 [12. Ci~
#> 3 Amridge Un~ 100690 4 [4. Age 18-~       7 AL     2 [2. Private no~ 12 [12. Ci~
#> 4 Amridge Un~ 100690 5 [5. Age 20-~      16 AL     2 [2. Private no~ 12 [12. Ci~
#> 5 Amridge Un~ 100690 6 [6. Age 22-~      34 AL     2 [2. Private no~ 12 [12. Ci~
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) %>%
  spread(key = age, value = efage09)

levstudy1_obs_tidy %>% head(n=5)
#> # A tibble: 5 x 19
#>   fullname unitid stabbr  sector   locale age_18_19 age_20_21 age_22_24
#>   <chr>     <dbl> <chr>  <dbl+l> <dbl+lb>     <dbl>     <dbl>     <dbl>
#> 1 Amridge~ 100690 AL     2 [2. ~ 12 [12.~         7        16        34
#> 2 Alabama~ 100724 AL     1 [1. ~ 12 [12.~      1750      1463      1191
#> 3 The Uni~ 100751 AL     1 [1. ~ 13 [13.~     13415     11741      5492
#> 4 Central~ 100760 AL     4 [4. ~ 32 [32.~       612       379       177
#> 5 Auburn ~ 100830 AL     1 [1. ~ 12 [12.~      1150      1157      1093
#> # ... with 11 more variables: age_25_29 <dbl>, age_25_plus <dbl>,
#> #   `age_30-34` <dbl>, `age_35-39` <dbl>, age_40_49 <dbl>, age_50_64 <dbl>,
#> #   age_65_plus <dbl>, age_all <dbl>, age_lt18 <dbl>, age_lt25 <dbl>,
#> #   age_unknown <dbl>
```

Confirm that the new object `levstudy1_obs_tidy` contains one observation for each value of unitid
/0.5

```
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
#> `summarise()` ungrouping output (override with `.groups` argument)
#> # A tibble: 1 x 2
#>   n_per_group     n
#>         <int> <int>
#> 1           1  2944
```

Questions related to spreading the dataset all_obs

Investigate data frame `all_obs` if you want, but not required to show code
/0.5

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

```
all_obs %>% group_by(unitid,agegroup,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()` regrouping output by 'unitid', 'agegroup' (override with `.groups` argument)
#> # A tibble: 1 x 2
#>   n_per_group     n
#>         <int> <int>
#> 1           1 85129
```

- Why is the data frame `all_obs` not tidy?
  /0.5
  - YOUR ANSWER HERE: the data frame has one row per college-agegroup-levstudy; these rows do not meet the requirements of being observations because an observation contains all values for some unit (e.g., a college)
- What changes need to be made to `all_obs` to make it tidy?
  - YOUR ANSWER HERE: each combination of the variables agegroup and levstudy should be converted from a row into a variable of its own
- 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

```
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 not-~ 12 [12. Cit~ age_all_l~
#>  2 Amridge Un~ 100690      57 AL     2 [2. Private not-~ 12 [12. Cit~ age_lt25_~
#>  3 Amridge Un~ 100690       7 AL     2 [2. Private not-~ 12 [12. Cit~ age_18_19~
#>  4 Amridge Un~ 100690      16 AL     2 [2. Private not-~ 12 [12. Cit~ age_20_21~
#>  5 Amridge Un~ 100690      34 AL     2 [2. Private not-~ 12 [12. Cit~ age_22_24~
#>  6 Amridge Un~ 100690     540 AL     2 [2. Private not-~ 12 [12. Cit~ age_25_pl~
#>  7 Amridge Un~ 100690      88 AL     2 [2. Private not-~ 12 [12. Cit~ age_25_29~
#>  8 Amridge Un~ 100690      97 AL     2 [2. Private not-~ 12 [12. Cit~ age_30-34~
#>  9 Amridge Un~ 100690     110 AL     2 [2. Private not-~ 12 [12. Cit~ age_35-39~
#> 10 Amridge Un~ 100690     158 AL     2 [2. Private not-~ 12 [12. Cit~ age_40_49~
#> 11 Amridge Un~ 100690      78 AL     2 [2. Private not-~ 12 [12. Cit~ age_50_64~
#> 12 Amridge Un~ 100690       9 AL     2 [2. Private not-~ 12 [12. Cit~ age_65_pl~
#> 13 Amridge Un~ 100690     294 AL     2 [2. Private not-~ 12 [12. Cit~ age_all_l~
#> 14 Amridge Un~ 100690      46 AL     2 [2. Private not-~ 12 [12. Cit~ age_lt25_~
#> 15 Amridge Un~ 100690       7 AL     2 [2. Private not-~ 12 [12. Cit~ age_18_19~
#> 16 Amridge Un~ 100690      15 AL     2 [2. Private not-~ 12 [12. Cit~ age_20_21~
#> 17 Amridge Un~ 100690      24 AL     2 [2. Private not-~ 12 [12. Cit~ age_22_24~
#> 18 Amridge Un~ 100690     248 AL     2 [2. Private not-~ 12 [12. Cit~ age_25_pl~
#> 19 Amridge Un~ 100690      45 AL     2 [2. Private not-~ 12 [12. Cit~ age_25_29~
#> 20 Amridge Un~ 100690      47 AL     2 [2. Private not-~ 12 [12. Cit~ age_30-34~
```

Tidy the data frame `all_obs_temp` and create a new object `all_obs_tidy`; then print a few observations /1

```
all_obs_tidy <- all_obs_temp %>%
  spread(key=age_lev, value=efage09)

all_obs_tidy %>% head(n=20)
#> # A tibble: 20 x 47
#>    fullname unitid stabbr  sector   locale age_18_19_lev_a~ age_18_19_lev_g~
#>    <chr>     <dbl> <chr>  <dbl+l> <dbl+lb>            <dbl>            <dbl>
#>  1 Amridge~ 100690 AL     2 [2. ~ 12 [12.~                7               NA
#>  2 Alabama~ 100724 AL     1 [1. ~ 12 [12.~             1750                2
#>  3 The Uni~ 100751 AL     1 [1. ~ 13 [13.~            13415                4
#>  4 Central~ 100760 AL     4 [4. ~ 32 [32.~              612               NA
#>  5 Auburn ~ 100830 AL     1 [1. ~ 12 [12.~             1150               NA
#>  6 Auburn ~ 100858 AL     1 [1. ~ 13 [13.~             9240                1
#>  7 Chattah~ 101028 AL     4 [4. ~ 41 [41.~              420               NA
#>  8 Enterpr~ 101143 AL     4 [4. ~ 32 [32.~              548               NA
#>  9 James H~ 101161 AL     4 [4. ~ 32 [32.~             1627               NA
#> 10 Faulkne~ 101189 AL     2 [2. ~ 12 [12.~              432               NA
#> 11 Gadsden~ 101240 AL     4 [4. ~ 13 [13.~             1385               NA
#> 12 George ~ 101286 AL     4 [4. ~ 41 [41.~             1161               NA
#> 13 George ~ 101295 AL     4 [4. ~ 32 [32.~             1587               NA
#> 14 George ~ 101301 AL     4 [4. ~ 32 [32.~              451               NA
#> 15 Hunting~ 101435 AL     2 [2. ~ 12 [12.~              326               NA
#> 16 J F Dra~ 101462 AL     4 [4. ~ 12 [12.~              104               NA
#> 17 J F Ing~ 101471 AL     4 [4. ~ 21 [21.~                3               NA
#> 18 Jackson~ 101480 AL     1 [1. ~ 13 [13.~             2132               NA
#> 19 Jeffers~ 101499 AL     4 [4. ~ 32 [32.~              274               NA
```

```
#> 20 Jeffers~ 101505 AL     4 [4. ~ 12 [12.~              2233                NA
#> # ... with 40 more variables: age_18_19_lev_ug <dbl>, age_20_21_lev_all <dbl>,
#> #   age_20_21_lev_grad <dbl>, age_20_21_lev_ug <dbl>, age_22_24_lev_all <dbl>,
#> #   age_22_24_lev_grad <dbl>, age_22_24_lev_ug <dbl>, age_25_29_lev_all <dbl>,
#> #   age_25_29_lev_grad <dbl>, age_25_29_lev_ug <dbl>,
#> #   age_25_plus_lev_all <dbl>, age_25_plus_lev_grad <dbl>,
#> #   age_25_plus_lev_ug <dbl>, `age_30-34_lev_all` <dbl>,
#> #   `age_30-34_lev_grad` <dbl>, `age_30-34_lev_ug` <dbl>,
#> #   `age_35-39_lev_all` <dbl>, `age_35-39_lev_grad` <dbl>,
#> #   `age_35-39_lev_ug` <dbl>, age_40_49_lev_all <dbl>,
#> #   age_40_49_lev_grad <dbl>, age_40_49_lev_ug <dbl>, age_50_64_lev_all <dbl>,
#> #   age_50_64_lev_grad <dbl>, age_50_64_lev_ug <dbl>,
#> #   age_65_plus_lev_all <dbl>, age_65_plus_lev_grad <dbl>,
#> #   age_65_plus_lev_ug <dbl>, age_all_lev_all <dbl>, age_all_lev_grad <dbl>,
#> #   age_all_lev_ug <dbl>, age_lt18_lev_all <dbl>, age_lt18_lev_grad <dbl>,
#> #   age_lt18_lev_ug <dbl>, age_lt25_lev_all <dbl>, age_lt25_lev_grad <dbl>,
#> #   age_lt25_lev_ug <dbl>, age_unknown_lev_all <dbl>,
#> #   age_unknown_lev_grad <dbl>, age_unknown_lev_ug <dbl>
```

- Confirm that the new object `all_obs_tidy` contains one observation for each value of unitid **/0.5**

```
all_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
#> `summarise()` ungrouping output (override with `.groups` argument)
#> # A tibble: 1 x 2
#>   n_per_group     n
#>         <int> <int>
#> 1           1  2944
```

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 tot_fall_2011
#>    <chr>   <chr>         <chr>         <chr>         <chr>         <chr>
#>  1 Alabam~ 48194.400000~ 57757         47492         49363.240000~ 47722.669999~
#>  2 Alaska~ 7880.3999999~ 7912          8083.1000000~ 8170.6399999~ 8087.8700000~
#>  3 Arizon~ 44438.400000~ 51376         51947.230000~ 50030.619999~ 50800.150000~
#>  4 Arkans~ 31947.400000~ 32997         37240         34272.800000~ 33982.959999~
#>  5 Califo~ 298021.40000~ 309222        316298.58000~ 260806.29999~ 268688.92999~
#>  6 Colora~ 41983.400000~ 45841         49060.32      48542.990000~ 48077.760000~
#>  7 Connec~ 41044.400000~ 39687         43592.829999~ 42951.389999~ 43804.810000~
#>  8 Delawa~ 7469.3999999~ 7998          8639.5799999~ 8933          8587.2099999~
#>  9 Distri~ 4949.3999999~ 5481          5854          5925.3299999~ 6278.0600000~
#> 10 Florid~ 132030.39999~ 158962        183827        175609.28999~ 175006.30000~
#> # ... with 41 more rows
```

/0.5

- 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: "Gather" year columns or reshape from wide to long

Tidy the data frame `table208_30` and create a new object `table208_30_tidy`:
/1.5

- 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

```
names(table208_30)
#> [1] "state"         "tot_fall_2000" "tot_fall_2005" "tot_fall_2009"
#> [5] "tot_fall_2010" "tot_fall_2011"
names(table208_30)<- c("state","2000","2005","2009","2010", "2011")
names(table208_30)
#> [1] "state" "2000"  "2005"  "2009"  "2010"  "2011"

table208_30_tidy <- table208_30 %>%
  gather(`2000`,`2005`,`2009`,`2010`, `2011`,  key = year, value = total_teachers)

#sort data (optional)
table208_30_tidy<- table208_30_tidy%>%
  arrange(state,year)

#examine data
head(table208_30_tidy, n=20)
#> # A tibble: 20 x 3
#>    state                          year  total_teachers
#>    <chr>                          <chr> <chr>
#>  1 Alabama ....................... 2000  48194.400000000001
#>  2 Alabama ....................... 2005  57757
#>  3 Alabama ....................... 2009  47492
#>  4 Alabama ....................... 2010  49363.240000000005
#>  5 Alabama ....................... 2011  47722.669999999998
#>  6 Alaska .......................  2000  7880.3999999999996
#>  7 Alaska .......................  2005  7912
#>  8 Alaska .......................  2009  8083.1000000000004
#>  9 Alaska .......................  2010  8170.6399999999994
#> 10 Alaska .......................  2011  8087.8700000000008
#> 11 Arizona ......................  2000  44438.400000000001
#> 12 Arizona ......................  2005  51376
#> 13 Arizona ......................  2009  51947.230000000003
#> 14 Arizona ......................  2010  50030.619999999995
#> 15 Arizona ......................  2011  50800.150000000001
#> 16 Arkansas .....................  2000  31947.400000000001
#> 17 Arkansas .....................  2005  32997
#> 18 Arkansas .....................  2009  37240
#> 19 Arkansas .....................  2010  34272.800000000003
```

```
#> 20 Arkansas ......................  2011  33982.959999999999
```

Bonus Question:

/4

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

```
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()

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
Remember to use this naming convention "lastname_firstname_ps6"