test

Daneal O'Habib

7/23/2019

Load Packages

```
library(haven)
library(tidyverse)
library(janitor)
```

Converting Titles to CSV

The first step is the convert all of the paper titles into csv files. This is done by running the python file called convert_to_csv.

This script accepts one required argument (–input) that specifies the path to folder with the raw paper titles. You can specify the output directory by optional argument –output. If you don't specify the output directory, a directory called "paper_titles_processed" will be created and will contain all the CSV files. This is what I did for this analysis - I created a new folder called "paper_titles_processed".

Example code to run the file: python convert_to_csv.py –input paper_titles

Joining Titles

I joined all of the csv files into one large dataframe that contains all of the paper titles. This was done using the R script called "join_titles_R". The outcome of this script is that it creates a data set called "joined_titles". We will be working with this later on in this file.

Import Data

Importing the data provided. There are three data sets you provided me with:

(1) "wordcount_pubmed18n_with_journal_pubtype",
(2) "pubmed18n_journalid_journalcategory",
(3) "pubmed18n_names_temp", and
(4) is the joined paper titles that I created called "joined_titles".

```
# import word count data (1)

pubmed18n_word_count_journal_pub_type <-
read_dta("paper_characteristics/wordcount_pubmed18n_with_journal_pubtype.dta")

# import journal id/category type data (2)

pubmed18n_journal_id_category <-
  read_dta("paper_characteristics/pubmed18n_journalid_journalcategory.dta")
```

```
#import paper names (3)

pubmed18n_names_temp <- read_dta("paper_characteristics/pubmed18n_names_temp.dta")

# import joined titles (4)

joined_titles <- read_csv("joined_titles.csv")
```

Data Processing

Word Count

I am processing the data called "pubmed18n_word_count_journal_pub_type". This is important because it has a variable - "is_journal_article2" - that identifies whether something is a journal article.

I took the following steps to processing the data. First, I filtered out everything that isn't a journal article. Second, I also removed all variables I thought weren't needed for the analysis. Third, I filtered for the years under study (1946 - 2012 for biomedicine journals). I created a new data set called "journal_filter". I am planning on joining this to the data that contains the paper titles.

```
# filtering out everything that isn't a journal article
# I also removed all variables that I thought weren't needed for this analysis
# processed the "pubmed18n_word_count_journal_pub_type" data set and
# created a new data set called "journal_filter"

journal_filter <- pubmed18n_word_count_journal_pub_type %>%
  filter(is_journal_article2 == 1,
         year >= 1946 & year <= 2012) %>%
  select(pmid, year)

# viewing output for new data set

journal_filter
```

```
## # A tibble: 16,115,778 x 2
##        pmid  year
##       <dbl> <dbl>
##  1 12255545  1946
##  2 12278355  1946
##  3 12305597  1946
##  4 12332284  1946
##  5 16016712  1946
##  6 16016713  1946
##  7 16016714  1946
##  8 16016715  1946
##  9 16016716  1946
## 10 16016718  1946
## # ... with 16,115,768 more rows
```

Journal Category

Now I am processing the data with journal categories - i.e, "pubmed18n_journal_id_category". I created a new data set called "journal_id_category", and renamed the variables using snake_case (just a personal

preference). I also removed the nlmid column because I want to use pmid as the unique identifier for the journals.

```
# processing data set with journal id/category type data
# renaming columns for coherence (just a preferred stylistic convention)
# removing nlmid column

journal_id_category <- pubmed18n_journal_id_category %>%
  rename("journal_category_id" = "journalcategoryid",
         "journal_category" = "journalcategory") %>%
  select(-nlmid)

# viewing data set I just created

journal_id_category
```

```
## # A tibble: 35,648,368 x 3
##     pmid journal_category journal_category_id
##    <dbl> <chr>                          <dbl>
##  1     1 ""                                NA
##  2     2 Biochemistry                       6
##  3     2 Biophysics                        62
##  4     3 Biophysics                        62
##  5     3 Biochemistry                       6
##  6     4 Biophysics                        62
##  7     4 Biochemistry                       6
##  8     5 Biophysics                        62
##  9     5 Biochemistry                       6
## 10     6 Biophysics                        62
## # ... with 35,648,358 more rows
```

I'm seeing that some papers can be categorized into more than one journal category. For example, the output above shows that pmid 3 falls is categorized by biophysics and biochemistry.

In the email you sent me, you said you wanted this analysis done for "biomedicine" journals. I explored the journal categories to see if there were any journals of biomedicine.

```
# grouping by journal category id and journal category and counting
# arrange in descending order

journal_category_count <- journal_id_category %>%
  group_by(journal_category_id, journal_category) %>%
  count() %>%
  arrange(-n)

# viewing

journal_category_count
```

```
## # A tibble: 126 x 3
## # Groups:   journal_category_id, journal_category [126]
##    journal_category_id journal_category        n
##                  <dbl> <chr>               <int>
```

```
##  1                  NA ""                 6695410
##  2                   1 Medicine           2253261
##  3                   6 Biochemistry       1329957
##  4                   2 Neurology           996553
##  5                   5 General Surgery     902660
##  6                   4 Neoplasms           859609
##  7                  60 Science             802708
##  8                  23 Chemistry           782302
##  9                   7 Molecular Biology   679257
## 10                  10 Pharmacology        672473
## # ... with 116 more rows
```

I viewed the complete output in R and didn't see any "biomedicine" journal

Another way to check:

```
# using a string detect function to see if there is any match
# for "biomedicine" in the "journal_category" column.

str_detect(journal_category_count$journal_category, regex("biomedicine", ignore_case = TRUE))
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [34] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [100] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [111] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [122] FALSE FALSE FALSE FALSE FALSE
```

I confirmed there is no journal category called "biomedicine". I looked at all of the journals that are listed, and it seems like every journal category is already indirectly related to biomedicine (e.g., medicine, biochemistry). So I assume that this is just a broad catagory. Further, I'll assume that you have already applied the biomedicine filter in this data, and that every unique pmid in the journal category data corresponds to a biomedicine journal.

```
# saving output if you want to see the full csv for journal counts

# write_csv(journal_category_count, "journal_category_count.csv")

# OR, just view it in R studio

# journal_category_count %>% View()
```

I'm not sure how to deal with the missing values. They could either be a journal related to biomedicine, and in that case we should leave them because they correspond to a unique journal. Or, they are some other field and I should filter them out. If the latter is true, and I easily change this in a later step. For now, I will leave the missing values.

I plan on joining everything together by a unique pmid, so I will create a dataframe with the unique pmid for all biomedicine journals. Since some journals fall under more then one category, I will just take the first category that appears. This dataframe is called "biomed_pmid_distint".

```
biomed_pmid_distint <-  journal_id_category %>%
  distinct(pmid, .keep_all = TRUE)
```

Names

I computed the team size by counting the distinct surnames for each pmid. Again, I am filtering for the years under study. This is a huge file, and it takes a long time to run. So only run it if you want to replicate the results. I am counting the number of distinct surnames for each pmid. This will give me the team size for each paper.

```
# This is a huge file, and it takes a long time to run.
# So only run it if you want to replicate the results.
# I am counting the number of distinct surnames for each pmid
# This will give me the team size for each paper

# grouping by pmid and year.
# the summarise funtion counts the number of distint last name.
# this count gives up the number of authers for each paper
# filter for the years under study
# (the paper notes the use biomedicine journals from 1946 - 2012)

number_authors <- pubmed18n_names_temp %>%
  group_by(pmid, year) %>%
  summarise(number_authors = n_distinct(lastname)) %>%
  filter(year >= 1946 & year <= 2012)

# view data

number_authors
```

```
## # A tibble: 21,759,495 x 3
## # Groups:   pmid [21,759,494]
##     pmid  year number_authors
##    <dbl> <dbl>          <int>
##  1     1  1975              4
##  2     2  1975              2
##  3     3  1975              2
##  4     4  1975              3
##  5     5  1975              2
##  6     6  1975              3
##  7     7  1975              2
##  8     8  1975              2
##  9     9  1975              2
## 10    10  1975              3
## # ... with 21,759,485 more rows
```

Quick summary statistics for the team size (measured by distinct surnames).

```r
summary(number_authors$number_authors)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    1.00    3.00    3.49    5.00 4634.00
```

Paper Titles

Filtering for the years under study.

```r
# just applying a filter - year 1946 - 2012 (as noted in the paper)
joined_titles <- joined_titles %>%
  filter(year >= 1946 & year <= 2012)
```

Joining Data

I want to join all of the data together. First, I take the data that contains all journal - "journal_filter" - and join it to the data that contains all distinct biomed articles - "biomed_pmid_distint".

I will join them by the unique pmid assigned to each paper and create a new dataframe called "biomed_journals".

```r
# joining everything that I know is a journal to everything that I know is a biomedical article

biomed_journals <- inner_join(journal_filter, biomed_pmid_distint, by = c("pmid"))

# view output

biomed_journals
```

```
## # A tibble: 16,115,778 x 4
##        pmid  year journal_category journal_category_id
##       <dbl> <dbl> <chr>                           <dbl>
##  1 12255545  1946 ""                                 NA
##  2 12278355  1946 ""                                 NA
##  3 12305597  1946 ""                                 NA
##  4 12332284  1946 ""                                 NA
##  5 16016712  1946 ""                                 NA
##  6 16016713  1946 ""                                 NA
##  7 16016714  1946 ""                                 NA
##  8 16016715  1946 ""                                 NA
##  9 16016716  1946 ""                                 NA
## 10 16016718  1946 ""                                 NA
## # ... with 16,115,768 more rows
```

I want to join all biomed journals to all of the paper titles. I will use both pmid and year as the unique key.

```r
# joining all biomed journals - dataframe called "biomed_journals" - to our
# dataframe of joined paper titles - this new data set

biomed_titles <- inner_join(biomed_journals, joined_titles, by = c("pmid", "year"))
```

```r
# view data

biomed_titles
```

```
## # A tibble: 16,395,482 x 5
##       pmid  year journal_category journal_category~ title
##      <dbl> <dbl> <chr>                         <dbl> <chr>
##  1  1.23e7  1946 ""                               NA The nutrition of expec~
##  2  1.23e7  1946 ""                               NA Ritual mutilation amon~
##  3  1.23e7  1946 ""                               NA Vitamin-C test for ovu~
##  4  1.23e7  1946 ""                               NA The clinical use of or~
##  5  1.60e7  1946 ""                               NA Editorial.
##  6  1.60e7  1946 ""                               NA The Army Medical Libra~
##  7  1.60e7  1946 ""                               NA Building a New Nursing~
##  8  1.60e7  1946 ""                               NA The Value of Exhibit M~
##  9  1.60e7  1946 ""                               NA Volunteers in Hospital~
## 10  1.60e7  1946 ""                               NA STANDARDS FOR MEDICAL ~
## # ... with 16,395,472 more rows
```

I am joining the dataframe created above to the dataframe that tells us team size for each paper (computed by the distinct surnames for each pmid).

```r
# joining dataframe that contain biomed journals + titles - "biomed_journal_titles" -
# to the data set that contains the number of authors - "number_authors"

biomed_titles_team <- inner_join(biomed_titles, number_authors, by = c("pmid", "year"))

# view

biomed_titles_team
```

```
## # A tibble: 16,112,065 x 6
##       pmid  year journal_category journal_categor~ title     number_authors
##      <dbl> <dbl> <chr>                       <dbl> <chr>              <int>
##  1  1.23e7  1946 ""                             NA The nutr~              1
##  2  1.23e7  1946 ""                             NA Ritual m~              1
##  3  1.23e7  1946 ""                             NA Vitamin-~              1
##  4  1.23e7  1946 ""                             NA The clin~              1
##  5  1.60e7  1946 ""                             NA The Army~              1
##  6  1.60e7  1946 ""                             NA Building~              1
##  7  1.60e7  1946 ""                             NA The Valu~              1
##  8  1.60e7  1946 ""                             NA Voluntee~              1
##  9  1.60e7  1946 ""                             NA The Effe~              1
## 10  1.60e7  1946 ""                             NA Yellow F~              1
## # ... with 16,112,055 more rows
```

This is raw version of our final dataframe.

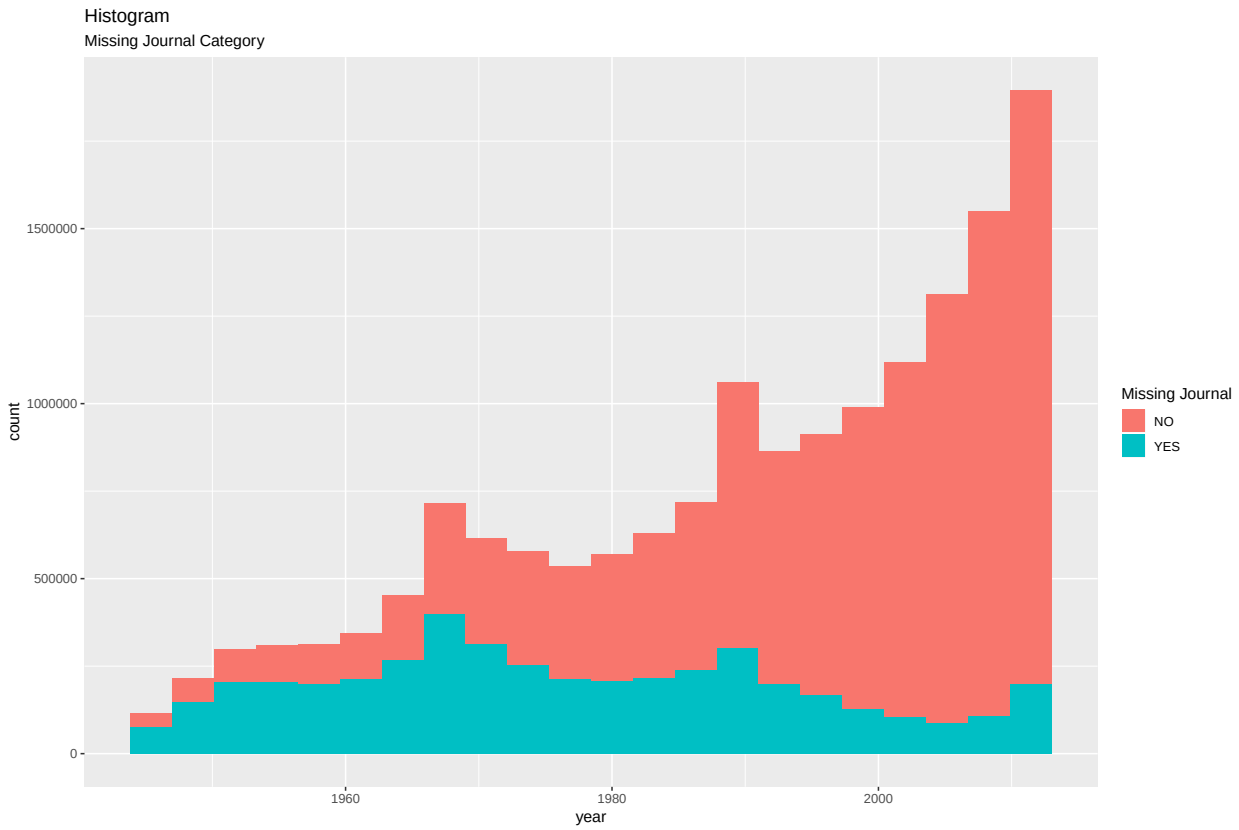
Data Quality

The purpose of this section is to check for any data quality issues in the data we created in the previous step.

Missing Journals and Years

Histogram of years under study. I want to see if there is a trend for missing journal category over time. Do we have more data for more contemporary biomedicine journals?

```
biomed_titles_team %>%
  mutate(missing_category = ifelse(journal_category == "", "YES", "NO")) %>%
  ggplot(aes(x = year, fill = missing_category)) + geom_histogram(alpha = 0.8) + stat_bin(bins = 22) +
  labs(title = "Histogram",
       subtitle = "Missing Journal Category",
       fill = "Missing Journal")
```



If everything provided is a biomedical journal then this shouldn't be a problem. The trend for the number of years under study looks reasonable as well.

Duplicates

I check for any duplicates in the data. Group by the unique identifiers and count each occurance.

```
# counting the number of unique titles
# we should only have n = 1 so I filtered only for n > 1
# the column n counts the number of duplicates for each pmid
duplicate_data <- biomed_titles_team %>%
  group_by(pmid, year, title) %>%
  count(sort = TRUE) %>%
  filter(n > 1)
```

```r
# viewing
# variable "n" just counts the number of duplicates for that pmid
# arranged in descending order

duplicate_data
```

```
## # A tibble: 270,797 x 4
## # Groups:   pmid, year, title [270,797]
##        pmid  year title                                                    n
##       <dbl> <dbl> <chr>                                                <int>
##  1 20029666  2009 Public preparedness guidance for a severe influenz~    24
##  2 22699293  2012 Performance of HbA1c as an early diagnostic indica~    12
##  3 19120261  2008 The Environmental Determinants of Diabetes in the ~    11
##  4 20703919  2010 Achieving standardized medication data in clinical~    11
##  5 21029290  2011 The Environmental Determinants of Diabetes in the ~    11
##  6 21419878  2011 Enrollment experiences in a pediatric longitudinal~    11
##  7 21527903  2011 Country-specific birth weight and length in type 1~    11
##  8 21564455  2011 The Environmental Determinants of Diabetes in the ~    11
##  9 21972409  2011 Reduced prevalence of diabetic ketoacidosis at dia~    11
## 10 22058606  2011 Food composition database harmonization for betwee~    11
## # ... with 270,787 more rows
```

Quick summary of the duplicates variable

```r
# summary just to see the distribution of duplicates
summary(duplicate_data$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   2.024   2.000  24.000
```

We have some problems in the data as some titles are duplicates. I also ran this code on the unprocess titles data - before I joined/filtered everything - and got a similar result. This shouldn't bias the result too much, but it is still more precise to remove all of the duplicates.

Final Processing Step

I remove all of the duplicates.

```r
# keeping all distinct data by pmid
distinct_data <- biomed_titles_team %>%
  distinct(pmid, .keep_all = TRUE)

# final processing step
# don't need the journal categoy

data_process <- distinct_data %>%
  select(-journal_category, -journal_category_id)
```

Here is the final dataframe - one unique pmid, year and title. The team size is called "number_authors" and is computed by counting the distinct surnames for each pmid.

```r
#view
data_process
```

```
## # A tibble: 15,834,578 x 4
##        pmid  year title                                        number_authors
##       <dbl> <dbl> <chr>                                                <int>
##  1 12255545  1946 The nutrition of expectant and nursing mo~              1
##  2 12278355  1946 Ritual mutilation among primitive peoples.              1
##  3 12305597  1946 Vitamin-C test for ovulation.                           1
##  4 12332284  1946 The clinical use of oral basal temperatur~              1
##  5 16016713  1946 The Army Medical Library: In Retrospect a~              1
##  6 16016714  1946 Building a New Nursing School Library.                  1
##  7 16016715  1946 The Value of Exhibit Material to the Prof~              1
##  8 16016716  1946 Volunteers in Hospital Libraries.                       1
##  9 16016719  1946 The Effect of the War Upon Medical Librar~              1
## 10 16016720  1946 Yellow Fever in New York City.                          1
## # ... with 15,834,568 more rows
```

The paper said they had something like 19 million observations so it is worth noting that I only have approx 15 million to work with. I'm not sure what I did wrong when processing the data, but I outined all of my steps.

Output

Save the data frame we just created - called "data_process" - to our working directory.

```r
write_csv(data_process, "data_process.csv")
```