cleaning\_description

Lizette van Zyl

October 25, 2017

## Questions about Brady and Li Paper

1. The observational unit of this article was vehicle drivers whose deaths resulted within 1 hour of getting into an auto accident. The file that contains this information within the yearly zipped file is the "person" data file.
2. Li and Brady constrained their data to the following:

* only drivers that died within an hour of the accident
* only six states that performed toxicological tests on more than 80% of the fatally injured drivers: California, Hawaii, Illinois, New Hampshire, Rhode Island, and West Virginia
* only drivers that died between January 1, 1999 and December 31, 2010

The variables to use to limit the data to this subset are:

* st\_case and state: use state id number and name to limit the analysis to the six states mentioned: keep numbers 6, 15, 17, 33, 44, and 54
* inj\_sev: injury severity used to limit to deaths only: keep number 4
* lag\_hrs: how many hours after the accident the person died - use to limit deaths to those happening within 1 hour of the accident: keep numbers less than or equal to 1
* lag\_mins: how many minutes after the number of lag\_hrs a person died - use to help limit time of death to less than 1 hour after the accident: keep number 0
* per\_typ: only inlcude vehicle drivers in the study: keep number 1

To limit the years, files were only taken from the zip files for the 12 years included in the study.

1. The variables used to stratify data in the Brady and Li paper are:

* age: coded as numbers. For years 2008 and earlier, convert any 99 entries to NA. For years after 2008, 998 or 999 entries were both converted to NA
* alc\_res: measured alcohol blood level: need to convert any numbers greater than 94 to NA in the cleaning process. To get a correct alcohol level in the body, convert the number by 10, and then consider any value greater than 0.01 to mean the driver tested positive for alcohol. This will end up being a TRUE or FALSE in the test result column
* columns contianing "drugres": coded as numbers, each of which correspond to a specific drug category. An example of how to convert is this: for numbers 100 to 295, list as a narcotic
* sex: coded as 1 or 2 for male or female; need to convert 9 to NA in the cleaning process

Note that year does not have coding within the person file, but rather the year is part of the zip file name and can be input into the dataframe through the function.

## Extra explanations for each of the comments in the cleaning code:

1. A .csv file with raw FARS data for each year is saved in the “data-raw/yearly\_person\_data” directory as “person\_[year].csv”. Since the working directory is the project directory, the person\_file object gives the path to the raw data file for whatever year is entered in the year argument of the function. The raw data for a year is read into R with the read\_csv function, which reads in a comma separated file as a tibble in R. The function takes the file path (person\_file) as its first argument, and the raw data frame is saved as an object called df.
2. To reduce the issues that accompany upper/lower case matching for manipulation of the dataframe, change all the column names in the original dataframe to lowercase letters.
3. Reduce the number of columns (variables) that the data manipulation has to handle. Do this by selecting only the columns related to driver deaths within 1 hour of an accident in the 6 states that the analysis considers, as well as unique identification for the person, drug and alcohol test results, and timestamp information.
4. Limit severity of injury to "fatal injury" and limit the person type to those who were inside the vehicle (driving) only, excluding people that died as a result of being hit by a car/vehicle. After limiting the scope of the person and injury data, these two variables are removed from the dataset to simplify the later analysis, since there is only one "level" now for each of these, and there is no need to have the information repeated across all the other variables.
5. Pull the entries for unique\_id, st\_case, veh\_no, and per\_no into one column to create a unique id for every person included in the analysis. After this, mutate the data so that year is input into a newly created column called year. Since all this code is within one huge function, it will input the year that is input to the function at this step. Also paste together the year into the unique\_id column with a "\_" between the numbers entry to create a completely unique id for each person in the analysis.
6. Filter down the data so that only people killed in one of 6 states is included. According to their corresponding state number codes, the states included in this analysis are: California, Hawaii, Illinois, New Hampshire, Rhode Island, and West Virginia. These were states that performed toxicological testing on more than 80% of their fatally injured drivers. The state column was then removed, since none of the analysis was stratified by individual state in the Brady and Li study.
7. First mutate the sex column so that any entries with a value of 9 are actually NA values, but that if the entry is not equal to 9, just keep the original entry. After this, say that the entries in this column correspond to two levels, 1 and 2, and then label these two levels as "Male" and "Female" accordingly.
8. Replace values of greater than 94 in the alcohol results column with NA values, otherwise, divide the number in the alcohol results column by 10. After that, create a logical column that has a TRUE value if the alcohol results value is greater than 0.01. After that, delete the original alcohol results column.
9. If the entry to the lag\_mins column is equal to 99, replace this with a NA value, otherwise keep the original number in the column. Do this in the original column, do not create a new column.
10. For years 2008 and earlier, replace 99 or 999 with NA values in the lag\_hrs column, otherwise keep the original entry value for the lag\_hrs. For years after 2008, replace 999 values with NA values in the lag\_hrs column, otherwise keep the original entry value for the lag\_hrs.
11. Cut down the dataframe to only drivers that died within an hour of the accident by only keeping lag\_hrs that are either less than 1 or that are equal to 1 and have lag\_mins that are equal to 0. Once the dataset is reduced to this condition, remove the lag\_hrs and lag\_mins columns since they are no longer needed.
12. For years 2008 and earlier, change entries equal to 99 to NA values, and keep all other numbers as they originally appear in the age column. For years after 2008, change entries of 998 or 999 to NA values and keep all other numbers as they originally appear in the age column. Do this within the age column, instead of adding a new column for this step.
13. Create a new column that puts age into categories. Do this by creating breaks at numbers 0, 25, 45, 65, and 1000. The 1000 number in this case is just chosen as a number higher than any person could live so that anyone older than 65 would be included in this top age category. The cut function and the breaks listed create the age categories based on numbers in the age column. Labels are assigned to designate the ranges between the specified breaks. After this agecat column is created, remove the old age column from the dataset. Include.lowest = TRUE and right = FALSE means that the break values should be included in the categories of numbers, and that the interval should be closed on the left and open on the right. This means that age of (for example) 25 would included in the second grouping, not the first.
14. Take each of the individual "drugres"-including columns and place them into a single column ("drug\_number") with their entries listed in a column named "drug\_type\_raw". Test the drug\_type\_raw column for numbers within the ranges (eg 100 to 295 for the narcotics) and if the result is TRUE, label those as the label corresponding to that number interval in a new column named drug\_type. For the first interval, replace all other entries in this new column with NA. Then, as this process is repeated for the other number intervals and corresponding new drug labels, the NAs will be replaced with the new drug label names in the drug\_type column. When all labels are accounted for, change the character strings in the drug\_type column to factors. When this is complete, remove the drug\_number and drug\_type\_raw columns from the dataset.
15. If both the Alchol and drug\_type columns for an entry contain NA, remove those rows from the dataset. If only one of these columns has NA in it, the row remains in the dataset.
16. Create a new dataset called non\_missing\_drugs from the gathered\_df dataframe. Remove rows that have NA in the drug\_type column. Group this datframe by unique\_id and drug\_type for the ensuing manipulations. summarise this dataframe by creating a column, "has\_drug" that includes an entry for each unique id that tested positive for the drug type. Ungroup the uniqu\_id and drug\_type columns from this summarised table. Add a column called row\_num that lists numbers 1 through the number of entries (rows) in the summarised dataframe. Take the summarised table from long form into an easier form for reading by spreading the drug\_type column back into individual columns and filling in any blank spaces left by this in the has\_drug column with FALSE. Remove the row\_num column from the dataframe again.
17. Remove all the columns containing "drugres" from the df dataframe. Join the non\_missing\_drugs dataframe to the df dataframe, keeping all entries from both dataframes, by the unique\_id column. Remove the column "None" (pertaining to drugs) from this newly joined dataframe. Put all six drug categories into a single column labeled drug\_type and put the corresponding values into a column called positive\_for\_drug. Turn the entries in the newly created drug\_type column into factors. remove duplicate rows from the dataframe. Return this final dataframe from the year that was input into the function.
18. Map the clean\_yearly\_person\_file function to the numbers 1999 through 2010 and store this mapping to a dataframe called clean\_fars. the map\_df function from purrr binds the rows of the dataframes created for each of the years into a single dataframe. save this to the data folder within the working directory (where the R project is stored).