Lab 10 - Merging Data

Johnathan Hsu

November 2, 2017

Using your own dataset (which may include more than one table) carry out the following data cleaning steps. Knit together the PDF document and commit both the Lab 10 RMD file and the PDF document to Git. Push the changes to GitHub so both documents are visible in your public GitHub repository.

1. For your poster project, do you have multiple tables you'd like to join together to create your complete dataset? If so, describe what each table represents.

I do! They are data_95_96 (1995-1996)... all the way up to data_2015. They are individual sentences from each year.

2. What is/are your primary key(s)? If you have more than one table in your data, what is/are your foreign key(s)? Do your primary key(s) and foreign key(s) have the same name? If not, what does this mean for the way you need to specify potential data merges?

They all have the sae name. I do not have to specify anything. The keys have the same name. The data has been merged and you can see the work in "scripts/merge_data.r".

3. If you do not need to merge tables to create your final dataset, create a new dataset from your original dataset with a `grouped_by()` summary of your choice. You will use this separate dataset to complete the following exercises.

If you are merging separate tables as part of your data manipulation process, are your keys of the same data type? If not, what are the differences? Figure out the appropriate coercion process(es) and carry out the steps below.

My keys are the same datatype after I swapped them out. If you look in merge_data.r I do a lot of the coersion and other necessary steps. Data is saved in num form so I could refer to the codebook as needed.

4. Perform each version of the mutating joins (don't forget to specify the `by` argument) and print the results to the console. Describe what each join did to your datasets and what the resulting data table looks like. For those joining two separate datasets, did any of these joins result in your desired final dataset? Why or why not?

I really don't need to join my datasets, I will practice though.

```
# tidyverse
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.3.2
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
## Warning: package 'tibble' was built under R version 3.3.2
```

```
## Warning: package 'tidyr' was built under R version 3.3.2
```

```
## Warning: package 'readr' was built under R version 3.3.2
```

```
## Warning: package 'purrr' was built under R version 3.3.2
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
# Spliting mtcars into to 2 parts
tibble_mtcars <- tbl_df(mtcars)
mtcars_1 <- tibble_mtcars[, 1:5]
mtcars_2 <- tibble_mtcars[, 6:10]
names <- rownames(mtcars)
mtcars_1 <- cbind(names, mtcars_1)
mtcars_2 <- cbind(names, mtcars_2)

# left_join
left_join(mtcars_1, mtcars_2, by = "names")
```

```
##                  names  mpg cyl  disp  hp drat    wt  qsec vs am gear
## 1            Mazda RX4 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4
## 2        Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4
## 3           Datsun 710 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4
## 4       Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3
## 5    Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3
## 6              Valiant 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3
## 7           Duster 360 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3
## 8            Merc 240D 24.4   4 146.7  62 3.69 3.190 20.00  1  0    4
## 9             Merc 230 22.8   4 140.8  95 3.92 3.150 22.90  1  0    4
## 10            Merc 280 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4
## 11           Merc 280C 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4
## 12          Merc 450SE 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3
## 13          Merc 450SL 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3
## 14         Merc 450SLC 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3
## 15  Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3
## 16 Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3
## 17   Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3
## 18            Fiat 128 32.4   4  78.7  66 4.08 2.200 19.47  1  1    4
## 19         Honda Civic 30.4   4  75.7  52 4.93 1.615 18.52  1  1    4
## 20      Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4
## 21       Toyota Corona 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3
## 22    Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3
## 23         AMC Javelin 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3
## 24          Camaro Z28 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3
## 25    Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3
## 26           Fiat X1-9 27.3   4  79.0  66 4.08 1.935 18.90  1  1    4
## 27       Porsche 914-2 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5
## 28        Lotus Europa 30.4   4  95.1 113 3.77 1.513 16.90  1  1    5
## 29      Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5
## 30        Ferrari Dino 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5
## 31       Maserati Bora 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5
## 32          Volvo 142E 21.4   4 121.0 109 4.11 2.780 18.60  1  1    4
```

```
# right_join
right_join(mtcars_1, mtcars_2, by = "names")
```

```
##                  names  mpg cyl  disp  hp drat    wt  qsec vs am gear
## 1            Mazda RX4 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4
## 2        Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4
## 3           Datsun 710 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4
## 4       Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3
## 5    Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3
## 6              Valiant 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3
## 7           Duster 360 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3
## 8            Merc 240D 24.4   4 146.7  62 3.69 3.190 20.00  1  0    4
## 9             Merc 230 22.8   4 140.8  95 3.92 3.150 22.90  1  0    4
## 10            Merc 280 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4
## 11           Merc 280C 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4
## 12          Merc 450SE 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3
## 13          Merc 450SL 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3
## 14         Merc 450SLC 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3
## 15  Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3
## 16 Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3
## 17   Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3
## 18            Fiat 128 32.4   4  78.7  66 4.08 2.200 19.47  1  1    4
## 19         Honda Civic 30.4   4  75.7  52 4.93 1.615 18.52  1  1    4
## 20      Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4
## 21       Toyota Corona 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3
## 22    Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3
## 23         AMC Javelin 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3
## 24          Camaro Z28 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3
## 25    Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3
## 26           Fiat X1-9 27.3   4  79.0  66 4.08 1.935 18.90  1  1    4
## 27       Porsche 914-2 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5
## 28        Lotus Europa 30.4   4  95.1 113 3.77 1.513 16.90  1  1    5
## 29      Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5
## 30        Ferrari Dino 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5
## 31       Maserati Bora 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5
## 32          Volvo 142E 21.4   4 121.0 109 4.11 2.780 18.60  1  1    4
```

```
# inner_join (simulated by taking out Datsun 710)
mtcars_3 <- mtcars_1 %>%
  filter(names != "Datsun 710")
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.2
```

```
inner_join(mtcars_3, mtcars_2)
```

```
## Joining, by = "names"
```

```
##                  names  mpg cyl  disp  hp drat    wt  qsec vs am gear
## 1            Mazda RX4 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4
## 2        Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4
## 3       Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3
## 4    Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3
## 5              Valiant 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3
## 6           Duster 360 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3
## 7            Merc 240D 24.4   4 146.7  62 3.69 3.190 20.00  1  0    4
## 8             Merc 230 22.8   4 140.8  95 3.92 3.150 22.90  1  0    4
## 9             Merc 280 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4
## 10           Merc 280C 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4
## 11          Merc 450SE 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3
```

```
## 12          Merc 450SL 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3
## 13         Merc 450SLC 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3
## 14  Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3
## 15 Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3
## 16   Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3
## 17            Fiat 128 32.4   4  78.7  66 4.08 2.200 19.47  1  1    4
## 18         Honda Civic 30.4   4  75.7  52 4.93 1.615 18.52  1  1    4
## 19      Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4
## 20       Toyota Corona 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3
## 21    Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3
## 22         AMC Javelin 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3
## 23          Camaro Z28 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3
## 24    Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3
## 25           Fiat X1-9 27.3   4  79.0  66 4.08 1.935 18.90  1  1    4
## 26       Porsche 914-2 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5
## 27        Lotus Europa 30.4   4  95.1 113 3.77 1.513 16.90  1  1    5
## 28      Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5
## 29        Ferrari Dino 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5
## 30       Maserati Bora 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5
## 31          Volvo 142E 21.4   4 121.0 109 4.11 2.780 18.60  1  1    4
```

```
# full_join - we see that the missing variables for Datsun is last.
full_join(mtcars_3, mtcars_2)
```

```
## Joining, by = "names"
```

```
##                  names  mpg cyl  disp  hp drat    wt  qsec vs am gear
## 1            Mazda RX4 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4
## 2        Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4
## 3       Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3
## 4    Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3
## 5              Valiant 18.1   6 225.0 105 2.76 3.460 20.22  1  0    3
## 6           Duster 360 14.3   8 360.0 245 3.21 3.570 15.84  0  0    3
## 7            Merc 240D 24.4   4 146.7  62 3.69 3.190 20.00  1  0    4
## 8             Merc 230 22.8   4 140.8  95 3.92 3.150 22.90  1  0    4
## 9             Merc 280 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4
## 10           Merc 280C 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4
## 11          Merc 450SE 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3
## 12          Merc 450SL 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3
## 13         Merc 450SLC 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3
## 14  Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3
## 15 Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3
## 16   Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3
## 17            Fiat 128 32.4   4  78.7  66 4.08 2.200 19.47  1  1    4
## 18         Honda Civic 30.4   4  75.7  52 4.93 1.615 18.52  1  1    4
## 19      Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4
## 20       Toyota Corona 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3
## 21    Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3
## 22         AMC Javelin 15.2   8 304.0 150 3.15 3.435 17.30  0  0    3
## 23          Camaro Z28 13.3   8 350.0 245 3.73 3.840 15.41  0  0    3
## 24    Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3
## 25           Fiat X1-9 27.3   4  79.0  66 4.08 1.935 18.90  1  1    4
## 26       Porsche 914-2 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5
## 27        Lotus Europa 30.4   4  95.1 113 3.77 1.513 16.90  1  1    5
## 28      Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5
```

```
## 29        Ferrari Dino 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5
## 30       Maserati Bora 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5
## 31          Volvo 142E 21.4   4 121.0 109 4.11 2.780 18.60  1  1    4
## 32          Datsun 710   NA  NA    NA  NA   NA 2.320 18.61  1  1    4
```

5. Do the same thing with the filtering joins. What was the result? Give an example of a case in which a `semi_join()` or an `anti_join()` might be used with your primary dataset.

```
# we see that semi_join excludes Datsun 710
semi_join(mtcars_3, mtcars_2, "names")
```

```
##                  names  mpg cyl  disp  hp drat
## 1            Mazda RX4 21.0   6 160.0 110 3.90
## 2        Mazda RX4 Wag 21.0   6 160.0 110 3.90
## 3       Hornet 4 Drive 21.4   6 258.0 110 3.08
## 4    Hornet Sportabout 18.7   8 360.0 175 3.15
## 5              Valiant 18.1   6 225.0 105 2.76
## 6           Duster 360 14.3   8 360.0 245 3.21
## 7            Merc 240D 24.4   4 146.7  62 3.69
## 8             Merc 230 22.8   4 140.8  95 3.92
## 9             Merc 280 19.2   6 167.6 123 3.92
## 10           Merc 280C 17.8   6 167.6 123 3.92
## 11          Merc 450SE 16.4   8 275.8 180 3.07
## 12          Merc 450SL 17.3   8 275.8 180 3.07
## 13         Merc 450SLC 15.2   8 275.8 180 3.07
## 14  Cadillac Fleetwood 10.4   8 472.0 205 2.93
## 15 Lincoln Continental 10.4   8 460.0 215 3.00
## 16   Chrysler Imperial 14.7   8 440.0 230 3.23
## 17            Fiat 128 32.4   4  78.7  66 4.08
## 18         Honda Civic 30.4   4  75.7  52 4.93
## 19      Toyota Corolla 33.9   4  71.1  65 4.22
## 20       Toyota Corona 21.5   4 120.1  97 3.70
## 21    Dodge Challenger 15.5   8 318.0 150 2.76
## 22         AMC Javelin 15.2   8 304.0 150 3.15
## 23          Camaro Z28 13.3   8 350.0 245 3.73
## 24    Pontiac Firebird 19.2   8 400.0 175 3.08
## 25           Fiat X1-9 27.3   4  79.0  66 4.08
## 26       Porsche 914-2 26.0   4 120.3  91 4.43
## 27        Lotus Europa 30.4   4  95.1 113 3.77
## 28      Ford Pantera L 15.8   8 351.0 264 4.22
## 29        Ferrari Dino 19.7   6 145.0 175 3.62
## 30       Maserati Bora 15.0   8 301.0 335 3.54
## 31          Volvo 142E 21.4   4 121.0 109 4.11
```

```
# we see that Datsun is the exception
anti_join(mtcars_2, mtcars_3, "names")
```

```
##        names   wt  qsec vs am gear
## 1 Datsun 710 2.32 18.61  1  1    4
```

6. What happens when you apply the set operations joins to your tables? Are these functions useful for you for this project? Explain why or why not. If not, give an example in which one of them might be usefully applied to your data.

```
mtcars_frankenstein <- bind_cols(mtcars_1, mtcars_2)
```

Binding rows will be useful for my project, but since I'm working with mtcars, I will be binding them by

columns.

7. If you have any reason to compare tables, apply `setequal()` below. What were the results?

Not applicable.

8. What is the purpose of binding data and why might you need to take extra precaution when carrying out this specific form of data merging? If your data requires any binding, carry out the steps below and describe what was accomplished by your merge.

I binded the data so we can match everything by year

```
# importing clean data
setwd("/Users/hsujohnathan/monitoring-federal-criminal-sentences")
data_95_96 <- read.csv("clean_data/collapsed_data/95-96.csv")
data_96_97 <- read.csv("clean_data/collapsed_data/96-97.csv")
data_97_98 <- read.csv("clean_data/collapsed_data/97-98.csv")
data_1999 <- read.csv("clean_data/collapsed_data/1999.csv")
data_2000 <- read.csv("clean_data/collapsed_data/2000.csv")
data_2001 <- read.csv("clean_data/collapsed_data/2001.csv")
data_2002 <- read.csv("clean_data/collapsed_data/2002.csv")
data_2003 <- read.csv("clean_data/collapsed_data/2003.csv")
data_2004 <- read.csv("clean_data/collapsed_data/2004.csv")
data_2005 <- read.csv("clean_data/collapsed_data/2005.csv")
data_2006 <- read.csv("clean_data/collapsed_data/2006.csv")
data_2007 <- read.csv("clean_data/collapsed_data/2007.csv")
data_2008 <- read.csv("clean_data/collapsed_data/2008.csv")
data_2009 <- read.csv("clean_data/collapsed_data/2009.csv")
data_2010 <- read.csv("clean_data/collapsed_data/2010.csv")
data_2011 <- read.csv("clean_data/collapsed_data/2011.csv")
data_2012 <- read.csv("clean_data/collapsed_data/2012.csv")
data_2013 <- read.csv("clean_data/collapsed_data/2013.csv")
data_2014 <- read.csv("clean_data/collapsed_data/2014.csv")
data_2015 <- read.csv("clean_data/collapsed_data/2015.csv")

data_bind <- bind_rows(data_95_96, data_96_97, data_97_98, data_1999,
                       data_2000, data_2001, data_2002, data_2003,
                       data_2004, data_2005, data_2006, data_2007,
                       data_2008, data_2009, data_2010, data_2011,
                       data_2012, data_2013, data_2014, data_2015)
```

9. Do you need to merge multiple tables together using the same type of merge? If so, utilize the `reduce()` function from the `purrr` package to carry out the appropriate merge below.

Not applicable

10. Are there any other steps you need to carry out to further clean, transform, or merge your data into one, final, tidy dataset? If so, describe what they are and carry them out below.

No.

```
tbl_df(head(data_bind))
```

```
## # A tibble: 6 x 13
##   MONRACE  YEAR MONSEX EDUCATN   AGE STATMIN STATMAX DISPOSIT TOTPRISN
## *   <int> <dbl>  <int>   <int> <int>   <int>   <dbl>    <int>    <int>
```

```
## 1       2  1996      1       1    20     120     540        0      120
## 2       2  1996      0       1    54       0     252        0       15
## 3       2  1996      0       2    21       0     240        0       87
## 4       1  1996      0       1    25      60     480        0       36
## 5       1  1996      0       1    35       0      36        0       24
## 6       1  1996      1       1    21       0      60        0       42
## # ... with 4 more variables: XFOLSOR <int>, XCRHISSR <int>,
## #   DISTRICT <int>, MONCIRC <int>
```