Data Wrangling and Tidying Unit Summary

Yogindra Raghav

10/26/2018

```
library(tidyverse)
```

Problem 1: Tidy Data

The following data set will be used to investigate whether Boy scout experience lowers the delinquency rate. The numeric values represent the count of boys in each category. The column `Social` stands for "Socioeconomic status".

```
##   Social Boyscout Delinquent.Y Delinquent.N
## 1   High      Yes           18           13
## 2   High       No           22           14
## 3    Low      Yes           16           18
## 4    Low       No           18           13
```

a. The dataset is not tidy. Why is that?

The data set is wide and does not contain one observation per row.

b. Tranform the data set into narrow format.

```
df %>% gather("Delinquent", "cases", 3:4)
```

```
##   Social Boyscout   Delinquent cases
## 1   High      Yes Delinquent.Y    18
## 2   High       No Delinquent.Y    22
## 3    Low      Yes Delinquent.Y    16
## 4    Low       No Delinquent.Y    18
## 5   High      Yes Delinquent.N    13
## 6   High       No Delinquent.N    14
## 7    Low      Yes Delinquent.N    18
## 8    Low       No Delinquent.N    13
```

Problem 2: Data wrangling – One Table

Load the `nycflights13` package. In the `flights` data set, find the top 5 carriers (airlines) which have the worst average arrival delays. You should only consider the positive arrival delays. Make sure you only select variables that are necessary solve this problem. Otherwise, the table will be too wide.

```
library(nycflights13)
```

```
## Warning: package 'nycflights13' was built under R version 3.4.4
```

```
flights %>% filter(arr_delay>-1) %>% select(carrier, arr_delay) %>% group_by(carrier) %>% summaris
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
## # A tibble: 5 x 2
##   carrier average_arrival_delay
##   <chr>                   <dbl>
## 1 OO                       60.6
```

```
## 2 YV                        49.9
## 3 9E                        47.6
## 4 EV                        46.7
## 5 F9                        45.5
```

Problem 3: Data wrangling – One Table

Table 1 contains information about TB cases in three different counties in 1999 and 2000.

```
table1
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

a. Calculate the proportion of TB in each country in each year.

```
table1 %>% mutate(prop_TB = cases/population)
```

```
## # A tibble: 6 x 5
##   country      year  cases population   prop_TB
##   <chr>       <int>  <int>      <int>     <dbl>
## 1 Afghanistan  1999    745   19987071 0.0000373
## 2 Afghanistan  2000   2666   20595360 0.000129
## 3 Brazil       1999  37737  172006362 0.000219
## 4 Brazil       2000  80488  174504898 0.000461
## 5 China        1999 212258 1272915272 0.000167
## 6 China        2000 213766 1280428583 0.000167
```

b. Expand the table from part a to a wide table, namely list 1999 and 2000 proportion of TB in separate columns. Note that you should only keep country, year and proportion before you spread the table.

```
table1 %>% mutate(prop_TB = cases/population) %>% spread(year, prop_TB)
```

```
## # A tibble: 6 x 5
##   country      cases population     `1999`     `2000`
##   <chr>        <int>      <int>      <dbl>      <dbl>
## 1 Afghanistan    745   19987071  0.0000373 NA
## 2 Afghanistan   2666   20595360 NA          0.000129
## 3 Brazil       37737  172006362  0.000219  NA
## 4 Brazil       80488  174504898 NA          0.000461
## 5 China       212258 1272915272  0.000167  NA
## 6 China       213766 1280428583 NA          0.000167
```

c. Create a new variable which is the percentage change in TB proportion from 1999 to 2000. Note that 1999 and 2000 are non-syntactic names (because they don't start with a letter) so we have to surround them in backticks in your R code. For example, `` `1999` ``.

Problem 4: Data wrangling – Two Tables

Suppose that we have two rectangular arrays of data, labeled `students` and `houses`. [In R terminology, they are `data.frames`.] `students` contains information about individual students (e.g. her student id, name, date of birth, class year, campus house, etc.). Each row in students contains data about one student. `houses` contains data about houses (e.g. house name, capacity, street address, etc.). Each row in houses contains data about one house. Suppose further that we want to generate a student address book. The address book will consist of two columns of data: the first column will contain the student's name; and the second will be the address where she lives.

a. Describe, in words, a data management operation that you could perform in order to achieve this. Be as specific as you can about what the operation will do and how it must be specified, but note that you do not have to write or reference R code!

Left_join() the 'students' data set with the 'houses' data set. You join it with the house name. Once you join the tables by the house name, use select() to select for the student name and address of the house.

b. It is important that every student appears in the address book, regardless of whether she lives on campus. Would a `inner_join()`, `left_join()`, `right_join()` or `full_join()` be most appropriate? Explain why.

Assuming that you give the "students" table first to R, then a left_join() would be most appropriate since it will guarantee that all entries in "students" are found in the joined table even if they cannot be matched to an entry in the "houses" table.

c. Suppose now that only students from the Class of 2014 are to be included in the address book. What additional data management operation could you perform to achieve this? Again, be specific, but there is no need to write code.

Perform a left_join() as before and then filter() for "Class of 2014". Once that is done you can select() for the name and address.