Big or Not So Big?

Data and Computing Fundamentals

In March 2014, the Centers for Medicare and Medicaid Services released data on treatments provided by more than 880,000 health-care providers. (CMS News Release and New York Times article.

The download site for the data, here, provides the data in ZIP archive format (but **don't** download it). The ZIP file is 414 MB. When unzipped, the resulting text file is 1.7 GB. There are 9,153,273 rows. Is this large? How can you approach such data?

There are different definitions of "big".

1. Used as a term of art, "big data" is generally meant to refer to data that is too large to fit on any one computer. There are special techniques for working on distributed data, e.g. HADOOP, that are beyond the scope of this course.
2. Data is "big" when it takes too long, i.e. longer than you have available, to process in useful ways.
3. Data is "big" when it takes too long to install on the computer you have available in the time that you have available.

By the standards of (1) and (2), the CMS data is not big. However, by the standards of (3), the data are big. During the class period it's impractical to download the data from the CMS.

To simplify things, a very small subset of the data — 100 rows — is available this way:

```
load( url( "http://tinyurl.com/m4o4n2b/DCF/CMS/CMSProceduresSmall.rda" ) )
```

Bring the data into R, then look at it within RStudio with

```
View(CMSProceduresSmall)
```

The codebook is available here.

Looking at the data in `View()`, identify some redundancies between rows — the same information stored in multiple places. In particular, think about whether the data table could be divided into two or more data tables that would not have redundant entries. Also, think about what variables would connect those tables so that they could be joined, if necessary, to reproduce the original.

** DISCUSS THIS WITH YOUR NEIGHBORS **

Dividing data in this way is known in database circles as *normalization*.

Getting familiar

When you encounter a new data set, take some time to get familiar with it. This includes reading through the codebook, and looking at individual variables.

Read in `Providers`, a file that describing the providers in the CMS data.

```
load( url( "http://tinyurl.com/m4o4n2b/DCF/CMS/Providers.rda" ) )
```

This has more than 880,000 rows, one for each of the health-care providers in the full CMS data.[1] This data table is small enough that you can `View()` it.

[1] There are presumably many more than 880,000 health-care providers in the US, but not all of them are connected to the Medicare system.

1. The `provider_type` variable has 90 different labels. Look at these arranged in descending order by the number of health-care. Which are the most common types? Do you see any obvious anomolies?

2. Look at `nppes_provider_gender` in the same way. Any obvious anomolies?

   Focus on the cases for which no gender is given. You can pull these out in this way:

   ```
   NoGender <- Providers %>%
     filter( nppes_provider_gender=="")
   ```

   Look at the distribution of `provider_type` for the `NoGender` cases. What do you conclude?

3. The `nppes_credentials` variable describes the type of health-care provider? How many different levels are there? Look at the counts in each level in descending order. What patterns do you see? Outline in words a strategy to correct the mistakes?