Tidy data

[This is an informal and code heavy version of the full tidy data paper](#). Please refer to that for more details.)

Data tidying

It is often said that 80% of data analysis is spent on the cleaning and preparing data. And it's no just a first step, but it must be repeated many times over the course of analysis as new problems come to light or new data is collected. To tidy a dataset, tidy data principles can be used to guide the small, but important, aspect of data cleaning that I call data tidying: structuring datasets to facilitate analysis.

The principles of tidy data provide a standard way to organise data values within a dataset. A standard makes life easier for both humans and computers. Tidy datasets are easy to manipulate, model and visualise, and have a specific structure that tidy data tools expect to be supplied to facilitate vital exploration and analysis of the data, and to simplify the development of data analysis tools that work well together. Current tools often require translation. You have to munge the output of one tool so that you can feed it to the input of another tool. Much of the time is spent on the tedious task of getting data into a tidy tabular format, allowing you to focus on the interesting domain problem, not on the interesting nature of data.

Defining tidy data

> Happy families are all alike; every unhappy family is unhappy in its own way. — Leo Tolstoy

Like families, tidy datasets are all alike but every messy dataset is messy in its own way. Tidy datasets provide a standardised way to link the structure of a dataset (its physical layout) with its semantics (its meaning). In this section, I'll provide some standard vocabulary for describing the structure and semantics of a dataset, and then use these definitions to define tidy data.

Data structure

Most statistical datasets are data frames made up of **rows** and **columns**. The columns are almost always labeled and the rows are sometimes labeled. The following code provides some data about an imaginary classroom that I will use extensively now and in the skill. The table has three columns and four rows, and both rows and columns are unlabeled.

```
classroom <- read.csv("classroom.csv", stringsAsFactors = FALSE)
classroom
##   name gender age height school
## 1 Jack   male  11    142   RGS
## 2 Jill ******  12    138  KHS
## 3 Jack   male  12    160  RGS
## 4 Jill ******  11    135  KHS
```

There are many ways to structure the same underlying data. The following table shows the same data as above, but the rows and columns have been transposed.

```
read.csv("classroom2.csv", stringsAsFactors = FALSE)
##   X Jack Jill Jack Jill
## 1 age   11   12   12   11
## 2 ht   142  138  160  135
## 3 sex ****** ****** ****** ******
## 4 sch   RGS  KHS  RGS  KHS
```

The data is the same, but the layout is different. Our vocabulary of rows and columns is simply not rich enough to describe two different types of data arrangement. To avoid ambiguity in discussion, we need a way to describe the underlying semantics, or meaning, of the values displayed in the table.

Data semantics

A dataset is a collection of **values**, usually either numbers (if quantitative) or strings (if qualitative). Values are organised in two ways. Every value belongs to a **variable** and an **observation**. A variable contains all values that measure the same underlying attribute (like height, temperature, duration) across units. An observation contains all values measured on the same unit (like a person, or a day, or a race) across attributes.

A tidy version of the classroom data looks like this. (you'll learn how the functions work a little later)

```
library(reshape)
library(plyr)

classroom2 <- classroom ****
ldply(.data=split(t(.[,-1]), .[,"name"]),
  .drop=TRUE, .id=NULL,
  function(x, assessment)
{
  ##Coerce to a data.frame
  ##- name: environment group
  ##- gender: character
  ##- age: numeric
  ##- height: numeric
  ##- school: factor
  ##- assessment: factor
  ##- gender: character
  ##- age: numeric
  ##- height: numeric
  ##- school: factor
  ##- assessment: factor
  ##- gender: character
  ##- age: numeric
  ##- height: numeric
  ##- school: factor
}, assessment = ****)
##   name assessment age height gender school
## 1 Jack         age  11    142   male  RGS
## 2 Jill         age  12    138 ******  KHS
## 3 Jack        ht   12    160   male  RGS
## 4 Jill        ht   11    135 ******  KHS
```

This makes the values, variables, and observations more clear. The dataset contains 4 observations and 5 variables representing three variables and 12 observations. The variables are:

1. `name`, with four possible values (Billy, Suzy, Lionel and Jenny).
2. `assessment`, with two possible values (age and height). I'
3. `age`, with five as values depending on how you think of the missing value (4, 6, 11, 12, 7, NA).

The tidy data frame explicitly tells us the definition of an observation. In this classroom, every combination of `name` and `assessment` is a single planned observation. The dataset also informs us of missing values, which we can at least interpret. Only one person doesn't like the quiz, but it's hard to analyze his grade. Suzy failed the first quiz, so we don't know if her age is missing at random, or if she was too sick to take the second quiz. Lionel's age is missing in both quizzes, which is a bit more worrying. Finally, we don't know if we need to know the class average for Suzy's height, or if we can impute it using the other observations we have collected.

For a given dataset, it's usually easy to figure out what are observations and what are variables, but it is especially difficult to precisely define variables and observations in general. For example, if the classroom in the previous example contained more students, then `name` would be a variable. But if we were interested in the relationship between height and `school`, it would be more clear, as we might think of height and width as values of a `dimension` variable. If `name` is a variable, then `school` must be a property of a student, but if we are interested in how school attendance influences test height and vaccination `status` would be a variable that varies across school and phone number for multiple people might be nonsense. A general rule of thumb is that it is easier to describe functional relationships between variables (properties) when observations are fixed (for example, there are always two heights for each person), but it is easier to make comparisons between groups of observations (e.g., average of group a vs. average of group b) that between groups of columns.

In a given analysis, some variables are fixed by the design of the analysis. For example, in a trial of new allergy medication we might have three observations taken on each person (`morning`, `afternoon` and `evening`), and multiple daily medication doses collected from each person on each day (`number of doses`, `medicine` of `spam`) and meteorological data collected on each day (`temperature`, `humidity`).

Variables may change across the course of analysis. Often the variables in the raw data are very fine grained, and may add extra modelling complexity for little explanatory gain. For example, many surveys ask variables on the ages of participants, but for most analyses, age ranges (`child`, `teenager`, `adult`) are more informative. Sometimes in later stages, you change focus to trials, compared by averaging across many participants. The constituents simplifies analysis because you don't need a hierarchical model, and you can often pretend that the data is continuous, not discrete.

Tidy data

Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations, variables and types. In **tidy data**:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

This is Codd's 3rd normal form, but with the constraints framed in statistical language, and the focus put on a single dataset rather than the many connected datasets common in relational databases. **Messy data** is any other arrangement of the data.

Tidy data makes it easy for an analyst or a computer to extract needed variables because it provides a standard way of structuring a dataset. Consider the classroom data. In a messy dataset, to find all the ages, I might need to use different strategies to extract different variables. This slows analysis and invites errors. If you consider the data as a collection of cells, I might need to know that I need to look at the second column for ages, but for heights I need to look at the third. Or I might need to know that I need the cells in the second row for Jill's data, but the third and fourth for Jack's. Or I might need to know that I need to look for cells that are equal to "11" for ages, but for heights I need to look for cells that are greater than 100. The complexity