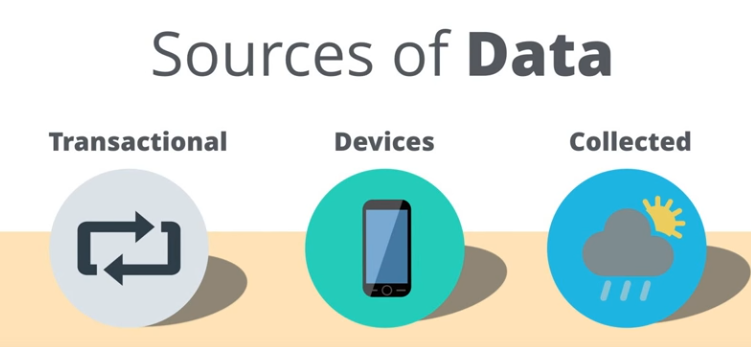
**Udacity-Creating an Analytical Dataset**

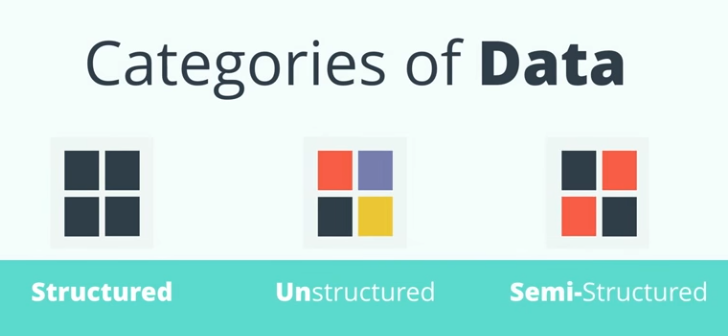
Jocelyn (Yuan) Li

February 2017

* Understanding Data
* Data Issues
* Data Formatting
* Data Blending

1. Structure of Data





**Three Types of Data Structure**

* **Structured Data**

Structured data are data with a high degree of organization. They are typically organized into columns and rows like in a spreadsheet. Sometimes columns are also called fields and rows are referred to as records and these terms may end up being used interchangeably throughout the course. Each column represents a variable and each row represents a record of data. Structured data is often stored in databases or files such as spreadsheets and it is usually easily accessible and most importantly, it’s easy to use.

* Column = Fields --> Variable
* Rows = Records
* **Unstructured Data (Resume, tweet, contract)**

Unstructured data can have no structure to it at all. Since the data isn’t organized into a typical columns and rows format, it can be time-consuming to work with as you have to pull what you want out of it. Some examples of this type of data are a resume, a tweet or a contract document.

* **Semi-structured Data**

Semi-structured data is data that has some structure to it but still requires some work to put it into a structured format of columns and rows. This could be a computer system log file that requires parsing and manipulating to put into a format that makes the data easier to analyze.

1. Sources of Data

File + web + database



1. Data Types

* String
* Numeric
* Data/Time
* Boolean
* Special Objects



**Data Types**

**Strings**

Strings are any **combination** of characters, alpha-numeric including symbols. Some examples can be an address field, a state code, an open-ended survey response or a product description.

**Numeric**

Numeric data are numbers which can be whole numbers such as integers or numbers with decimal places. Some examples are sales in dollars, population in a trade area around a store or the age of a person.

**Date/Time**

Date/time data can contain a specific date or a combination of both date and time. This data can be really handy for calculating the number of minutes between a caller reporting a problem and its resolution.

**Boolean (logical type)**

The Boolean type is sometimes also called a Logical type and is a conditional flag representing either true or false.

**Special Objects**

Special objects which can be objects such as images, maps, report objects, and sound files to name a few examples

1. Data Issues

* Dirty Data (Parsing)
* Extra Characters
* Missing Data (gaps about data)
* Outliers

1. Missing Data?

Why we care about missing data?

1. Some statistical algorithms won’t work.
2. Missing data can add bias to a model (overestimated or underestimated)

**Readings on Bias**

* General article: [**https://en.wikipedia.org/wiki/Sampling\_bias**](https://en.wikipedia.org/wiki/Sampling_bias)
* Detailed article: [**http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2917255/**](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2917255/)

1. How to deal with missing data?
2. Delete missing data
3. Impute data (provide the fake data)
4. Multiple Imputation
5. Full information maximum likelihood (limited to linear models)

How many data we are missing?

How the data is distributed?

Is the missing data numeric or categorical?

**Dealing with Missing Data**

We’ve taken a look at what missing data can look like, and we’ve also seen why it’s important to deal with. The end result that we are trying to get to is having data that creates a model based on data that eliminates as much bias as possible.

So now let’s talk through different ways that you can deal with the data that’s missing. Through these next steps, we’ll see that how we approach the missing data will have different effects on the resulting model process. There are several options an analyst can use to deal with missing data. The first option is to delete records with missing data. For example, let's start by identifying and removing missing data from the dataset.

**Deleting Missing Data**

Deleting missing data is often the default method because of it's simplicity. No decisions that need to be made that might confuse the data. You just get rid of records where there are missing values.

However, you should make sure that deleting missing data doesn't have adverse effects on your analysis. For example, if a particular demographic tended to leave a response blank in a survey, then removing records with blank entries will mean that a part of the population is underrepresented.

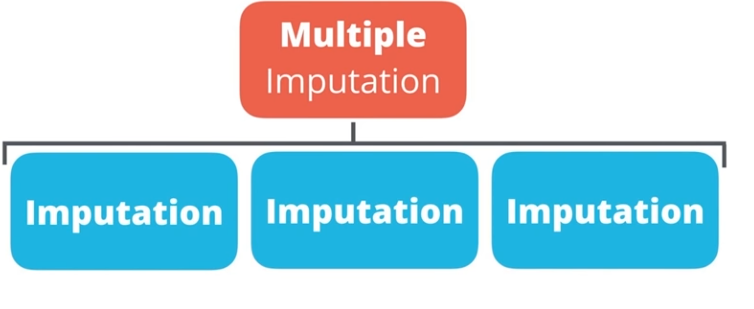
One of the downsides is that eliminating missing data reduces the size of the dataset. As you'll see in your exercise removing records with missing values eliminates half of the observations in the dataset. This can be especially problematic when collecting the data is expensive, such as paying people for a survey.

**Imputation**

Replace missing value with mean, median, or mode.

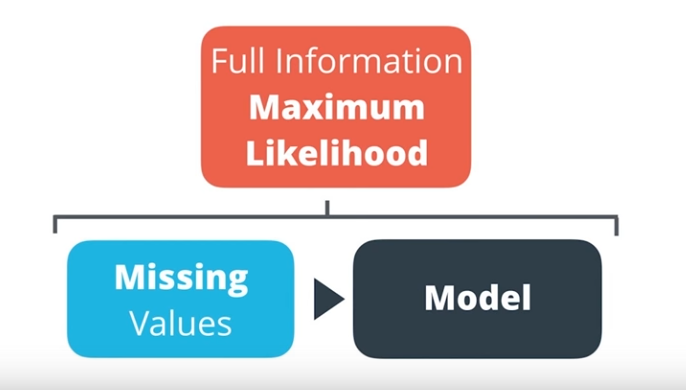
**Multiple Imputation**

[**Multiple imputation**](http://www.stefvanbuuren.nl/mi/MI.html): <http://www.stefvanbuuren.nl/mi/MI.html>



**Full information maximum likelihood**

**Maximum likelihood:** [**http://www.theanalysisfactor.com/missing-data-two-recommended-solutions/**](http://www.theanalysisfactor.com/missing-data-two-recommended-solutions/)



1. Outliers (Extreme data points)

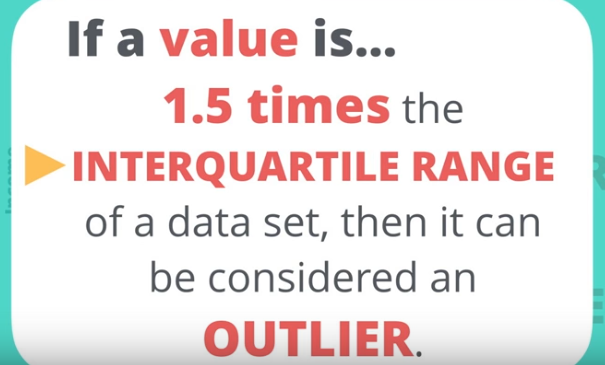
* Incorrect data (typo)
* Abnormal but correct data

Several ways to identify outliers.

* Box-and-WHISKER plot
* Violin Plot
* Use Z-scores (http://www.real-statistics.com/sampling-distributions/identifying-outliers-missing-data/)

In statistics, an outlier is an observation point that is distant from other observations.

Why do we care about outliers?



**From BOX-and-WHISKER plot**

* Calculate the Interquartile Range: IQR = Q3 - Q1
* Add 1.5 \* IQR to Q3 to get the upper fence: Upper Fence = Q3 + 1.5 \* IQR
* Subtract 1.5 \* IQR to Q1 to get the lower fence: Lower Fence = Q1 - 1.5 \* IQR

1. Data blending

Data blending

* Unions
* Joins
* Fuzzy Matching
* Spatial Matching

**Fuzzy matching** is a method that provides an improved ability to process word-based **matching** queries to find **matching** phrases or sentences from a database.