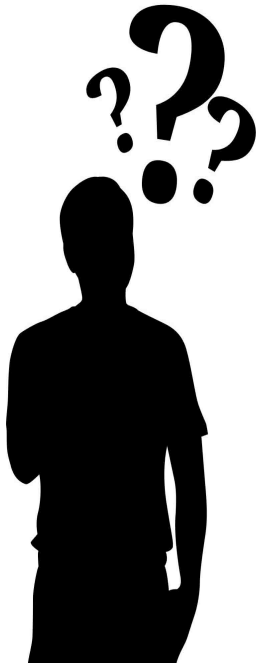
Introduction to Data Science

GETTING STARTED: EXPLORATORY DATA ANALYSIS

BRIAN D'ALESSANDRO

FIRST THINGS FIRST:
KNOW THE SHAPE OF YOUR DATA



2. What does it look like?

- Feature Types
- Missing values/outliers
- Distribution
- Covariance structure

A ROSE BY ANY OTHER NAME

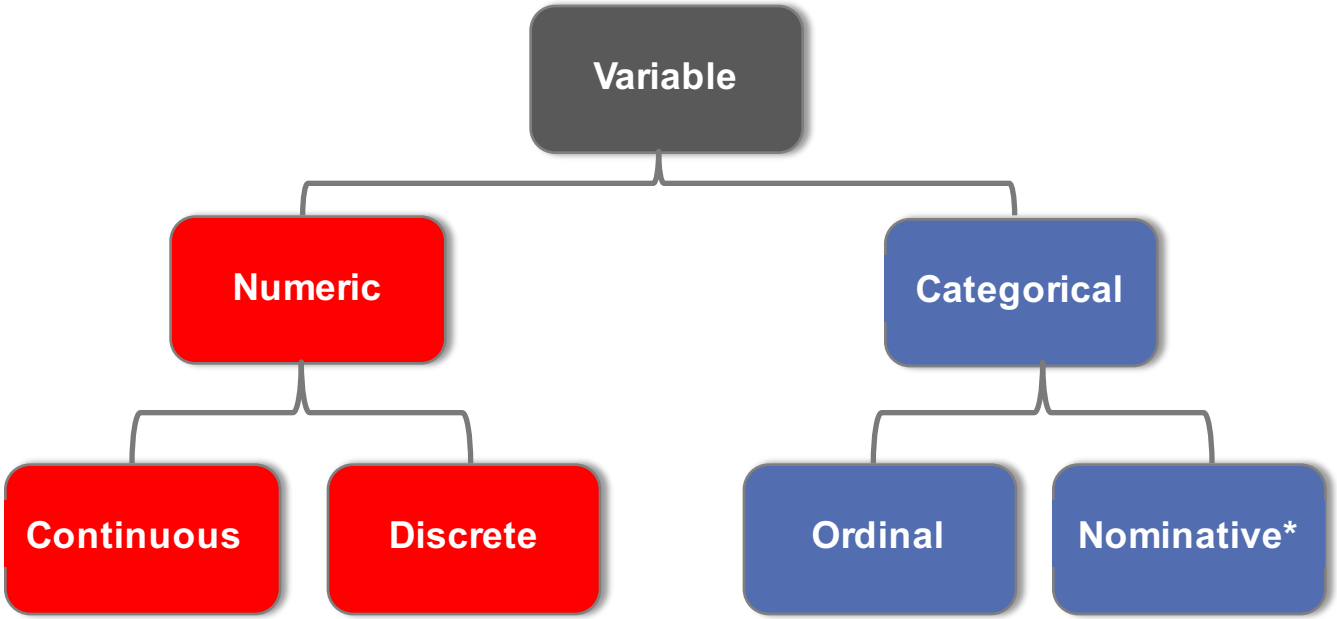
Given the multiple disciplinary origins of what is now Data Science, we often see several words used to mean the same concepts.

	Statistics	Machine Learning	Common Symbols
Thing you want to predict	Dependent Variable	Target Variable / Label	Y, f(X)
Things you use to predict	Independent Variable	Feature	X

NB: Admittedly, I often use these terms interchangeably.

TYPES OF FEATURES

Most variables fall under four basic categories



**Notably missing from this chart is text data, but I view text as a special type of unordered categorical variables*

IMPLICATIONS OF FEATURE TYPE

Modeling

- Every algorithm has requirements on what type of data can be used as an input. I.e., regression based methods require numeric or binary variables, while tree methods can accept any type*.

- Often times you can 'cheat' linear models by transforming the data to capture non-linear form (more on this in later lectures).

Analysis/Exploration

- Not all distributional statistics are defined on categorical data, but you can use category labels to compare statistics across category groupings.

**Note: this often varies by software implementation. It is best to consider software documentation as the authoritative guide*

STEP 1 IN KNOWING YOUR DATA: LOOK AT IT

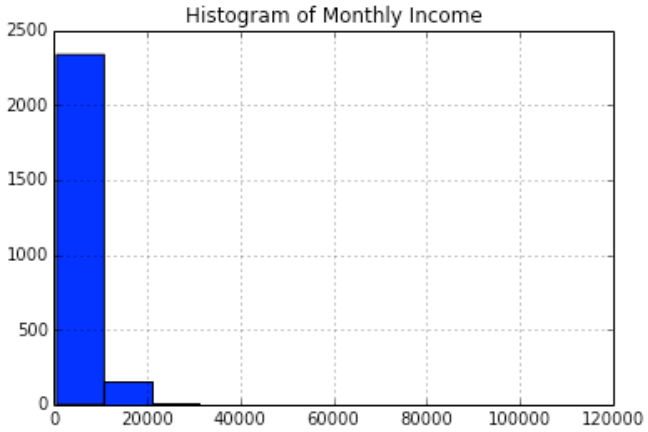
Simply looking at the first 10 lines (and get to know your Unix commands to do this) can reveal a lot of what you need to know

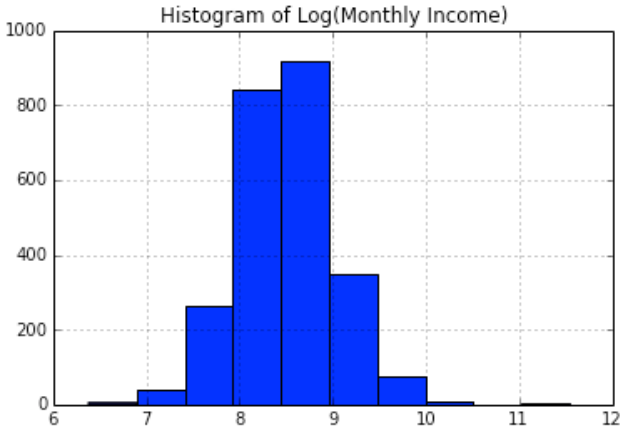
```
MacBook-Air-2:data brian$ head ads_dataset.txt
isbuyer buy_freq       visit_freq      buy_interval    sv_interval     expected_time_buy       expected_time_visit     last_buy        last_vis
t       multiple_buy   multiple_visit  uniq_urls       num_checkins    y_buy
        0       NaN     1       0       0       0       0       106     106     0       0       169     2130    0
        0       NaN     1       0       0       0       0       72      72      0       0       154     1100    0
        0       NaN     1       0       0       0       0       5       5       0       0       4       12      0
        0       NaN     1       0       0       0       0       6       6       0       0       150     539     0
        0       NaN     2       0       .5      0       -101.1493       101     101     0       1       103     362     0
        0       NaN     1       0       0       0       0       42      42      0       0       17      35      0
        0       NaN     1       0       0       0       0       42      42      0       0       42      110     0
        0       NaN     2       0       29.79167        0       -106.1883       121     121     0       1       101     401     0
        0       NaN     3       0       45.47917        0       -34.14473       64      64      0       1       100     298     0
```

EMPIRICAL DISTRIBUTIONS

Most data comes in as is and it doesn't always matter what you call its Distribution (in terms of known, parametric probability distributions)…

but its still useful to understand its shape. The histogram is a great tool for this.





EMPIRICAL DISTRIBUTIONS

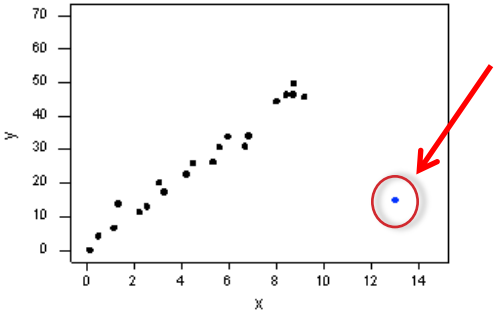
It is often very useful to know the distributional statistics of your data…

```
In [7]: loansData['Monthly.LogIncome'].describe()

Out[7]: count    2499.000000
        mean        8.501915
        std         0.523019
        min         6.377577
        25%         8.160518
        50%         8.517193
        75%         8.824678
        max        11.540054
        dtype: float64
```

DATA CLEANING

Beyond general data exploration and transformations, one usually wants some degree of data cleanup, looking for outliers and missing values.



Outliers are often the result of erroneous data collection or processing. Not only are they a good clue for data processing QA, but they can have severe influence on model estimates or summary statistics.

Missing values can indicate processing/collection errors. When present in the data (as either null, NA or “”), they can break a lot of modeling algorithms.

HANDLING OUTLIERS/NULLS

What should one do when faced with these issues?
The easiest answer in data science is *it depends…*

If missing/bad at random…

- If the occurrence is rare, delete observations (most extreme)
- Can also impute/replace with average or median for that feature.

Otherwise…

- Impute with some constant (usually the average or median), create a dummy variable to indicate missing/bad
- Exploit multi-collinearity, i.e., use a model to estimate E[Missing Val|X].

No matter your preferred technique, this becomes a testable design choice. I.e,

1. Identify multiple competing methods for imputation and/or outlier cleaning
2. Train a model for each method (on the same training data)
3. Evaluate each method against the same out-of-sample validation data
4. Choose the best performing