# Structure of a Data Analysis

Roger D. Peng, Associate Professor of Biostatistics

May 18, 2016

# Steps in a data analysis

- Define the question
- Define the ideal data set
- Determine what data you can access
- Obtain the data
- Clean the data
- Exploratory data analysis
- Statistical prediction/modeling
- Interpret results
- Challenge results
- Synthesize/write up results
- Create reproducible code

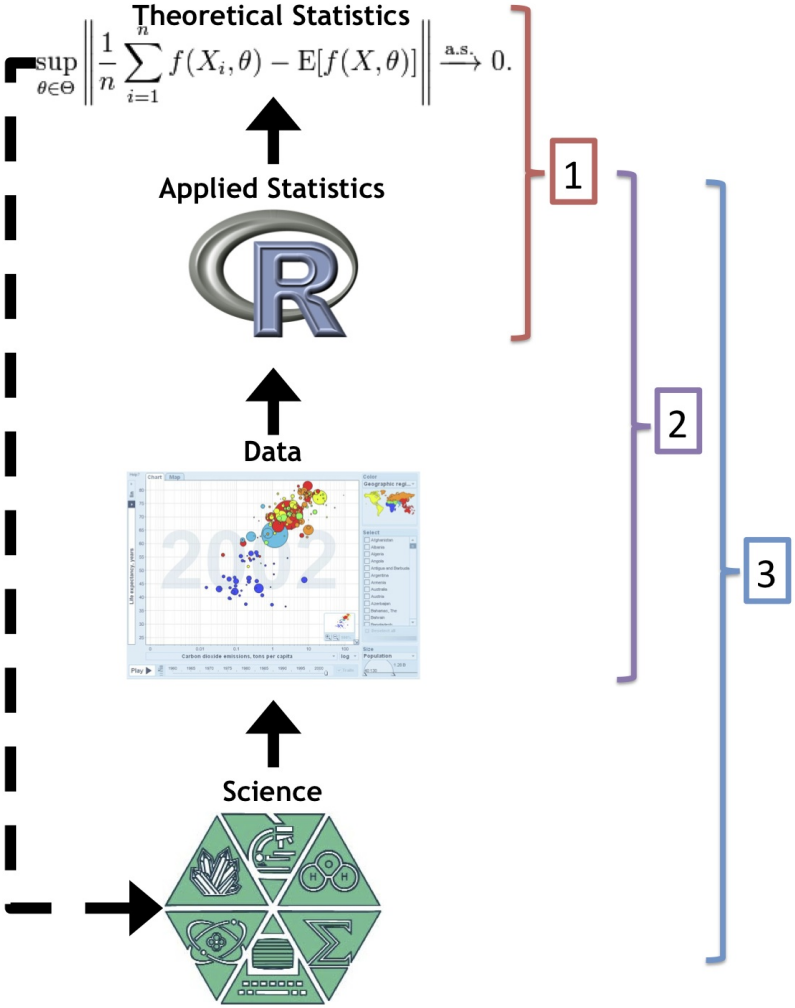
# Steps in a data analysis

- Define the question
- Define the ideal data set
- Determine what data you can access
- Obtain the data
- Clean the data
- Exploratory data analysis
- Statistical prediction/modeling
- Interpret results
- Challenge results
- Synthesize/write up results
- Create reproducible code

# The key challenge in data analysis

"Ask yourselves, what problem have you solved, ever, that was worth solving, where you knew all of the given information in advance? Where you didn't have a surplus of information and have to filter it out, or you had insufficient information and have to go find some?"

# Defining a question




1. Statistical methods development
2. Danger zone!!!

# An example

**Start with a general question**

Can I automatically detect emails that are SPAM that are not?

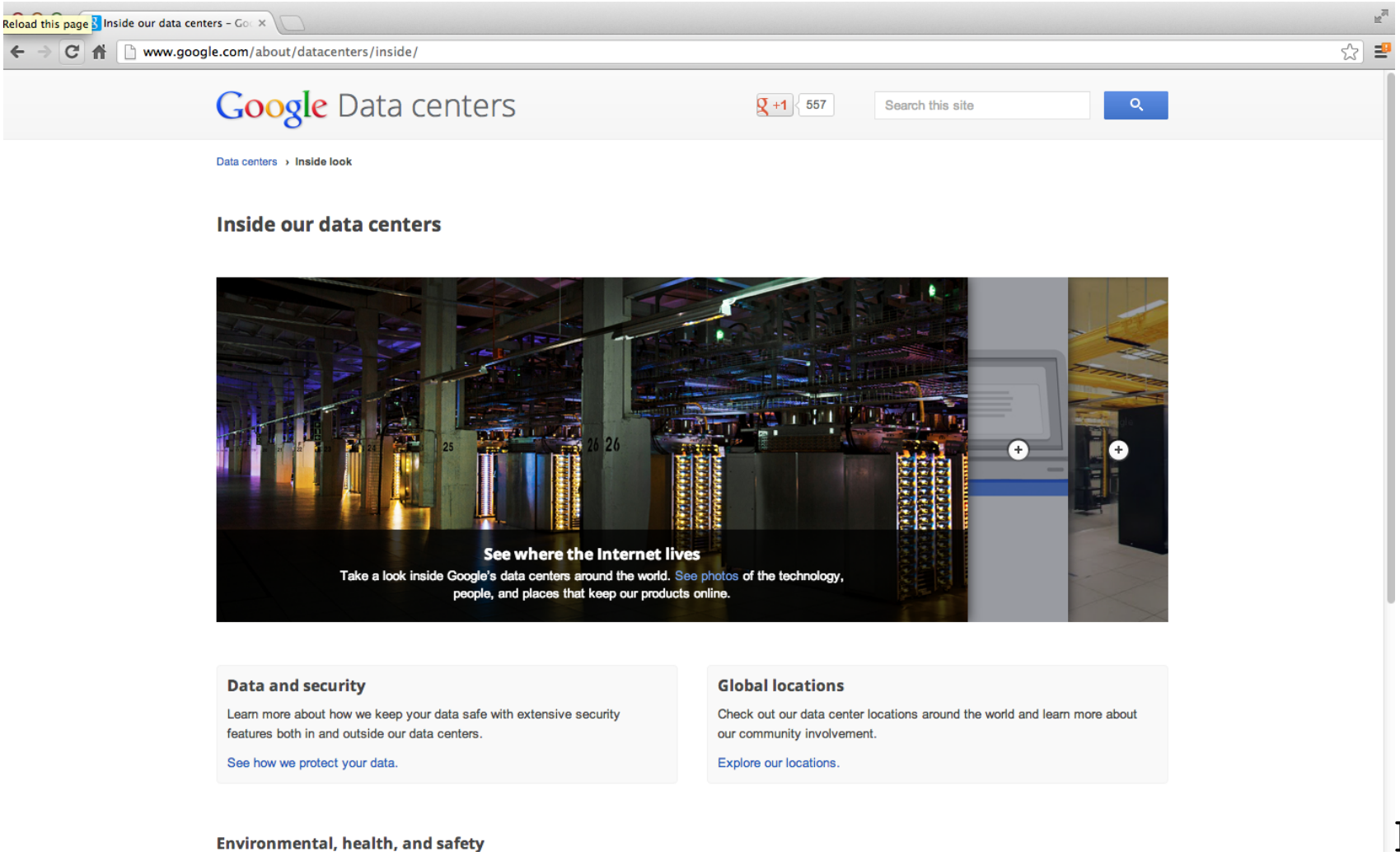
**Make it concrete**

Can I use quantitative characteristics of the emails to classify them as SPAM/HAM?

# Define the ideal data set

- The data set may depend on your goal
- Descriptive - a whole population
- Exploratory - a random sample with many variables measured
- Inferential - the right population, randomly sampled
- Predictive - a training and test data set from the same population
- Causal - data from a randomized study
- Mechanistic - data about all components of the system

# Our example

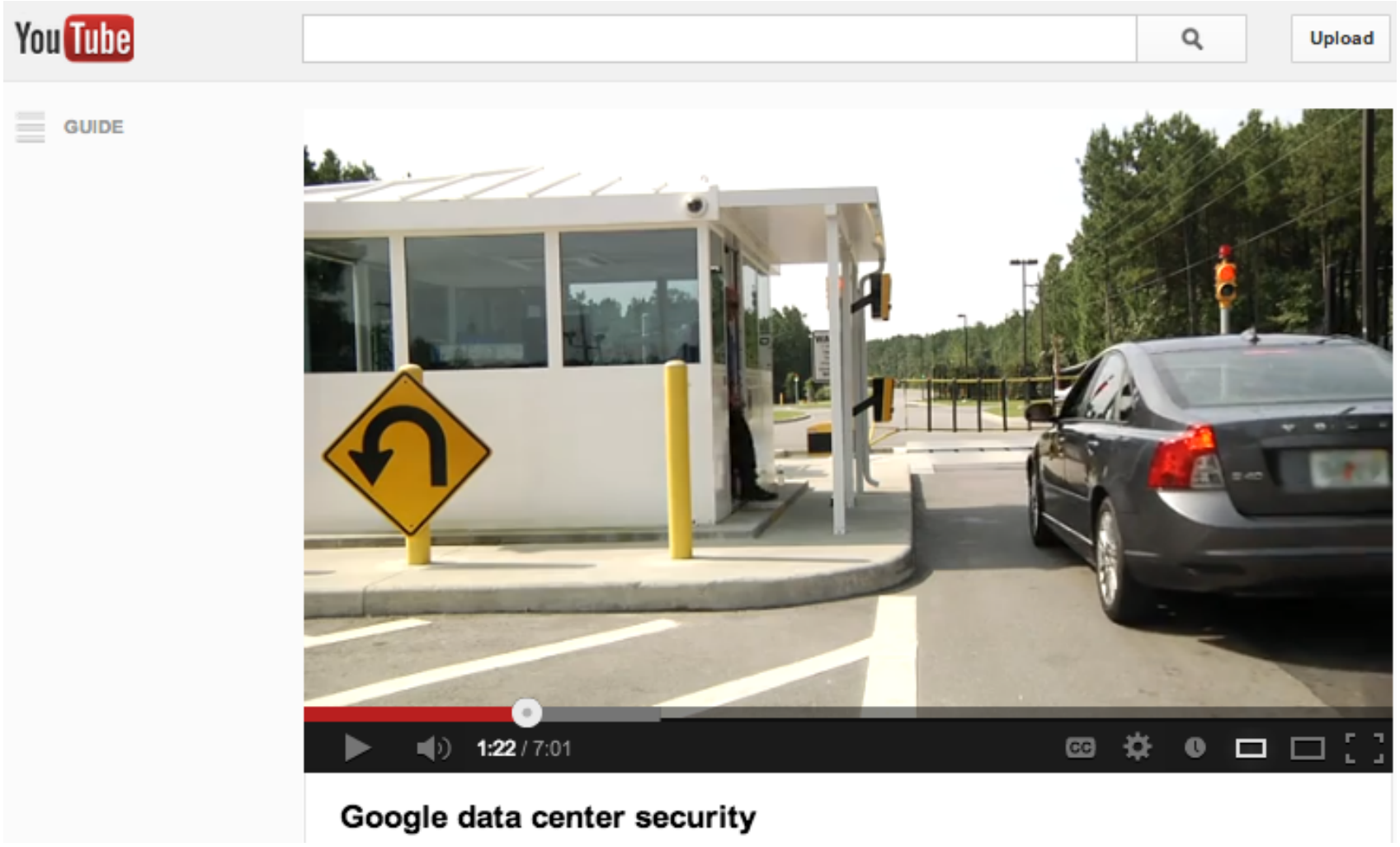



http://www.google.com/about/datacenters/inside/

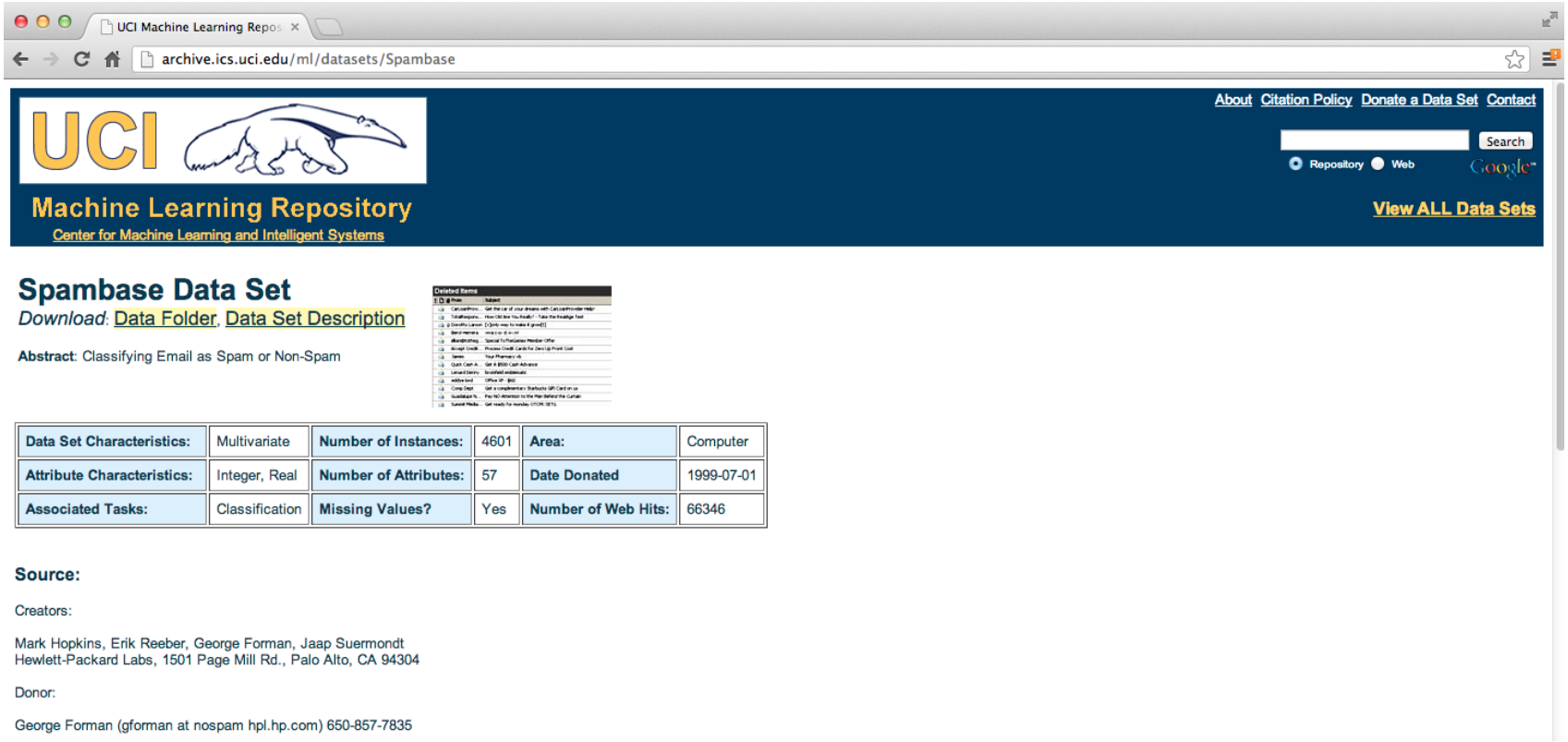
# Determine what data you can access

- Sometimes you can find data free on the web
- Other times you may need to buy the data
- Be sure to respect the terms of use
- If the data don't exist, you may need to generate it yourself

# Back to our example



# A possible solution

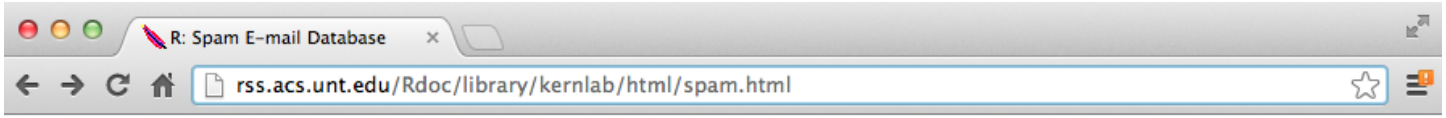



`http://archive.ics.uci.edu/ml/datasets/Spambase`

# Obtain the data

- Try to obtain the raw data
- Be sure to reference the source
- Polite emails go a long way
- If you will load the data from an internet source, record the url and time accessed

# Our data set




spam {kernlab} R Documentation

**Spam E-mail Database**

**Description**

A data set collected at Hewlett-Packard Labs, that classifies 4601 e-mails as spam or non-spam. In addition to this class label there are 57 variables indicating the frequency of certain words and characters in the e-mail.

**Usage**

```
data(spam)
```

**Format**

A data frame with 4601 observations and 58 variables.

The first 48 variables contain the frequency of the variable name (e.g., business) in the e-mail. If the variable name starts with num (e.g., num650) the it indicates the frequency of the corresponding number (e.g., 650). The variables 49-54 indicate the frequency of the characters `;', `(', `[', `!', `$', and `#'. The variables 55-57 contain the average, longest and total run-length of captial letters. Variable 58 indicates the type of the mail and is either `"nonspam"` or `"spam"`, i.e. unsolicited commercial e-mail.

**Details**

The data set contains 2788 e-mails classified as `"nonspam"` and 1813 classified as `"spam"`.

The ``spam'' concept is diverse: advertisements for products/web sites, make money fast schemes, chain letters, pornography... This collection of spam e-mails came from the collectors' postmaster and individuals who had filed spam. The collection of non-spam e-mails came from filed work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter. One would either have to blind such non-spam indicators or get a very wide collection of non-spam to generate a general purpose spam filter.

**Source**

- Creators: Mark Hopkins, Erik Reeber, George Forman, Jaap Suermondt at Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304
- Donor: George Forman (gforman at nospam hpl.hp.com) 650-857-7835

These data have been taken from the UCI Repository Of Machine Learning Databases at http://www.ics.uci.edu/~mlearn/MLRepository.html

**References**

T. Hastie, R. Tibshirani, J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

`http://search.r-project.org/library/kernlab/html/spam.html`

# Clean the data

- Raw data often needs to be processed
- If it is pre-processed, make sure you understand how
- Understand the source of the data (census, sample, convenience sample, etc.)
- May need reformating, subsampling - record these steps
- **Determine if the data are good enough** - if not, quit or change data

# Our cleaned data set

```
# If it isn't installed, install the kernlab package with
library(kernlab)
data(spam)
str(spam[, 1:5])
```

```
## 'data.frame':    4601 obs. of  5 variables:
##  $ make    : num  0 0.21 0.06 0 0 0 0 0 0.15 0.06 ...
##  $ address: num  0.64 0.28 0 0 0 0 0 0 0 0.12 ...
##  $ all     : num  0.64 0.5 0.71 0 0 0 0 0 0.46 0.77 ...
##  $ num3d   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ our     : num  0.32 0.14 1.23 0.63 0.63 1.85 1.92 1.88
```

http://search.r-project.org/library/kernlab/html/spam.html