SML 201 – Week 1

John D. Storey

Spring 2016

Origins of Data Science

Statistics and Machine Learning

Statistics is the study of the collection, analysis, interpretation, presentation, and organization of data.

https://en.wikipedia.org/wiki/Statistics

Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Machine learning is closely related to and often overlaps with computational statistics; a discipline which also focuses in prediction-making through the use of computers.

https://en.wikipedia.org/wiki/Machine_learning

Data Science

Data Science is an interdisciplinary field about processes and systems to extract knowledge or insights from data in various forms, either structured or unstructured, which is a continuation of some of the data analysis fields such as statistics, data mining, and predictive analytics.

https://en.wikipedia.org/wiki/Data_science

What is Data Science?

- *Data Science* is a very new term
- No well-accepted definition
- Statistics, machine learning, and data science are all essentially about extracting knowledge or value from data
- DS deals with data in the following ways: collecting, storing, managing, wrangling, exploration, learning, discovery, communication, products

Some History

John Tukey

John Tukey pioneered a field called “exploratory data analysis” (EDA)

From **The Future of Data Analysis** (1962) *Annals of Mathematical Statistics* …

> *For a long time I have thought I was a statistician, interested in inferences from the particular to the general. But as I have watched mathematical statistics evolve, I have had cause to wonder and to doubt.*

John Tukey (cont'd)

All in all, I have come to feel that my central interest is in data analysis, which I take to include, among other things: procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data.

John Tukey (cont'd)

> *Data analysis is a larger and more varied field than inference, or incisive procedures, or allocation.*

IMO, Tukey saw the need for and initiated data science in 1962

David Donoho **seems to agree**

Jeff Wu

In November 1997, C.F. Jeff Wu gave the inaugural lecture entitled "Statistics = Data Science?". In this lecture, he characterized statistical work as a trilogy of data collection, data modeling and analysis, and decision making. In his conclusion, he initiated the modern, non-computer science, usage of the term "data science" and advocated that statistics be renamed data science and statisticians data scientists.

https://en.wikipedia.org/wiki/Data_science

William Cleveland

- In 2001, William Cleveland introduced data science as an independent discipline, extending the field of statistics to incorporate “advances in computing with data” in his article **Data Science: An Action Plan for Expanding the Technical Areas of the Field of Statistics** in *International Statistical Review*
- Cleveland establishes six technical areas which he believed to encompass the field of data science: multidisciplinary investigations, models and methods for data, computing with data, pedagogy, tool evaluation, and theory.

(The above is modified text from **Wikipedia**.)

Industry

- “In 2008, DJ Patil and Jeff Hammerbacher used the term ‘data scientist’ to define their jobs at LinkedIn and Facebook, respectively.” (**from Wikipedia**)
- The term “data scientist” is now often used to describe positions in industry that primarily involve data, whether it is statistics, machine learning, data curation, or other data-centric activities

Relevant Quotations

Nate Silver

“I think data scientist is a sexed-up term for a statistician.”

http://simplystatistics.org/2013/08/08/data-scientist-is-just-a-sexed-up-word-for-statistician/

Twitter

- “Data science is statistics on a Mac.”
- “A data scientist is a statistician who lives in San Francisco.”
- “A data scientist is someone who is better at statistics than any software engineer and better at software engineering than any statistician.”

http://datascopeanalytics.com/blog/what-is-a-data-scientist/

Hadley Wickham

“Recently, there has been much hand-wringing about the role of statistics in data science.

I think there are three main steps in a data science project: you collect data (and questions), analyze it (using visualization and models), then communicate the results."

http://bulletin.imstat.org/2014/09/data-science-how-is-it-different-to-statistics%E2%80%89/

Hadley Wickham (cont'd)

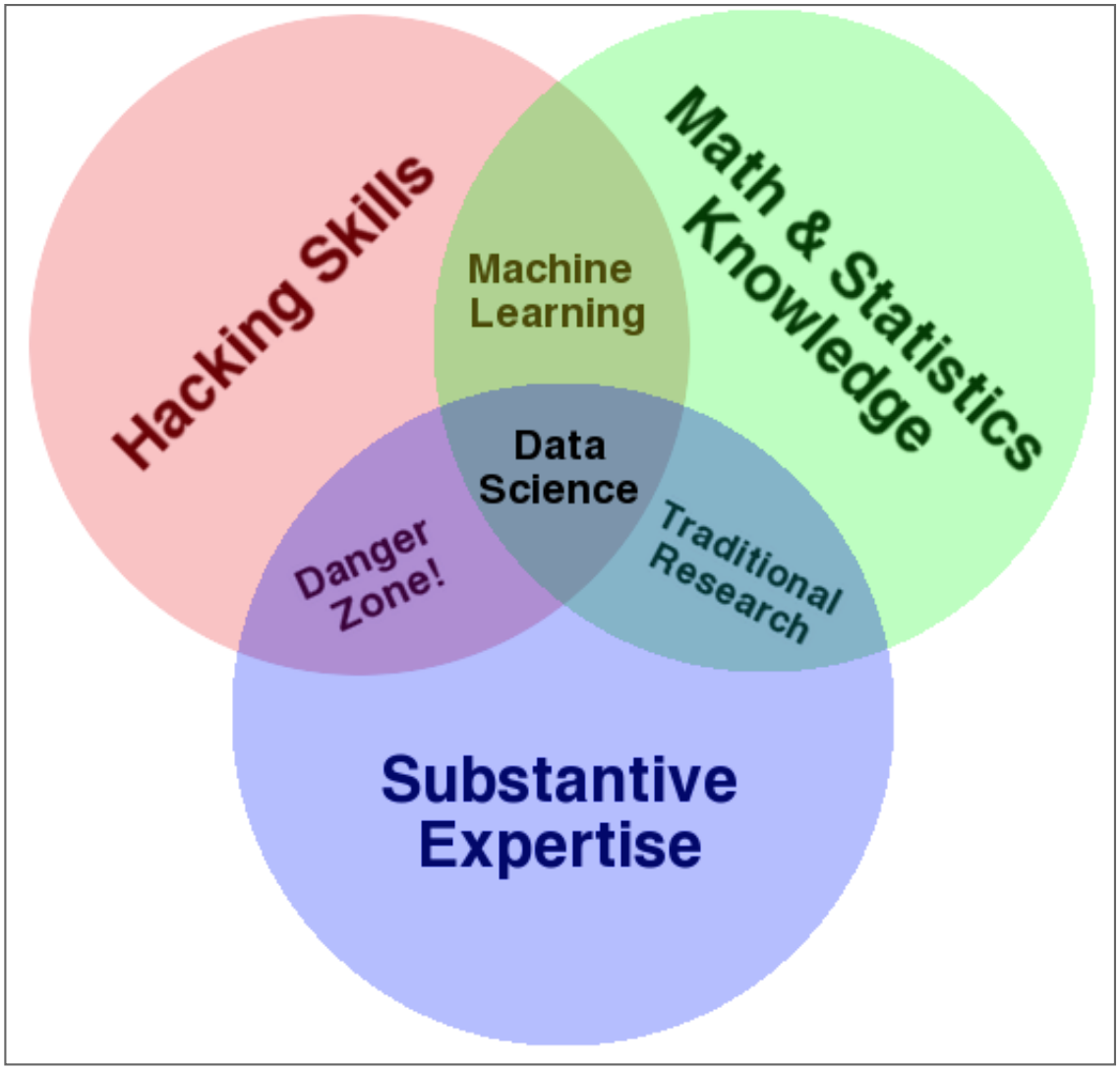
"Statistics is a part of data science, not the whole thing. Statistics research focuses on data collection and modelling, and there is little work on developing good questions, thinking about the shape of data, communicating results or building data products."

http://bulletin.imstat.org/2014/09/data-science-how-is-it-different-to-statistics%E2%80%89/

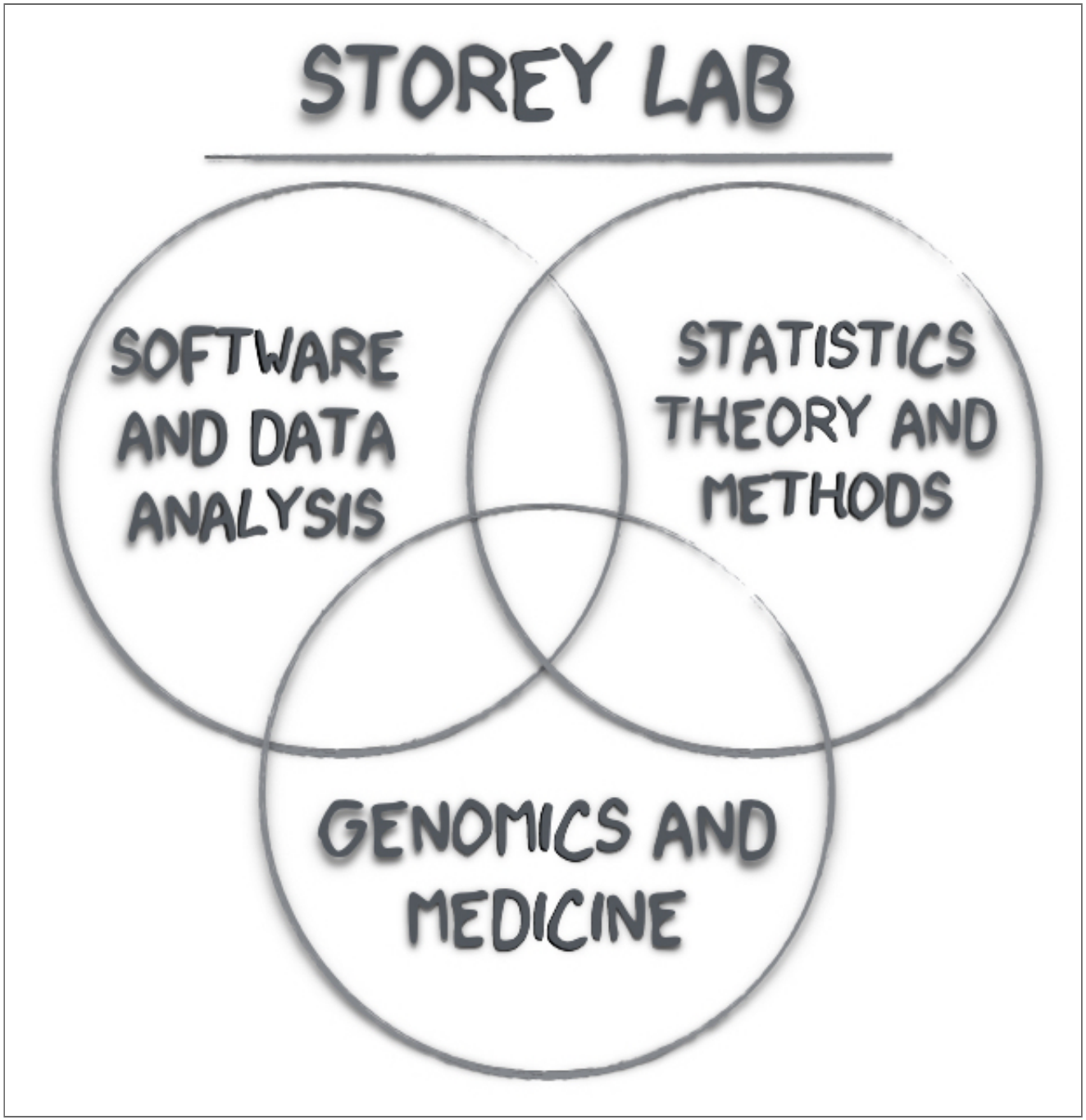
Jeff Leek

“The key word in Data Science is not Data, it is *Science*.”

http://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/



http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram


STOREY LAB
SOFTWARE AND DATA ANALYSIS
STATISTICS THEORY AND METHODS
GENOMICS AND MEDICINE

SML 201

What we cover

- Using R to do data science
- Data wrangling
- Data exploration and visualization
- Inference, modeling, and prediction
- Data products (briefly)

What we don’t cover

- Python
- Databases
- Big data computing (e.g., Hadoop, MapReduce, Spark)
- Mathematical theory

Other Princeton courses

- Fundamentals of statistics: ECO202, EEB355, POL345, WWS200, ORF245, PSY201, and SOC301
- Fundamentals of machine learning: SML302/COS424, ORF350
- SML 201 is complementary to these and can be taken before or after

Course Logistics

This course will be managed on:

- **Blackboard**
- **Piazza**
- **http://sml201.github.io**

All course notes will be made available on GitHub:

- **https://github.com/SML201**

The Art of Data Science

Data Analysis as Art

- Donald Knuth **said**, “Science is what we understand well enough to explain to a computer. Art is everything else we do.”
- Data analysis involves a lot of science – e.g., well-justified mathematical formulas that we can program a computer to calculate
- However, we cannot yet program a computer to carry out any data analysis from beginning to end

Data Analysis as Art

- In *The Art of Data Science*, the authors point out that data analysis is largely an art
- There is a gap between what is typically taught in a statistics course and what is required to carry out an excellent data analysis
- The books *The Art of Data Science* and *The Elements of Data Analytic Style* attempt to begin to address this gap

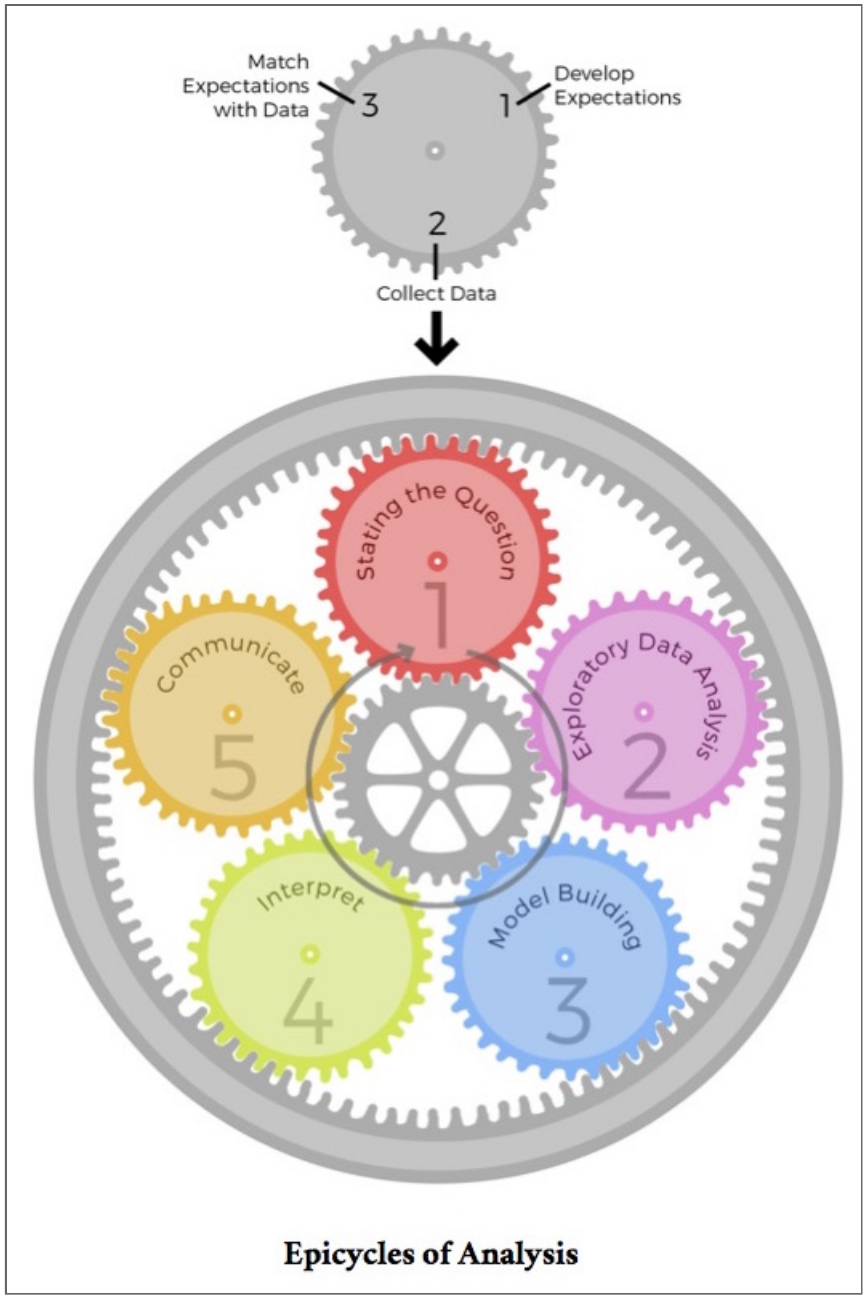
Examples

- Facebook's **Visualizing Friendships** (side note: **a discussion**)
- **Hans Rosling: Debunking third-world myths with the best stats you've ever seen**
- Flowing Data's **A Day in the Life of Americans**

Five Activities of Data Analysis

1. Stating and refining the question
2. Exploring the data
3. Building formal statistical models
4. Interpreting the results
5. Communicating the results

From *The Art of Data Science*.



Epicycles of Analysis

From *The Art of Data Science*.

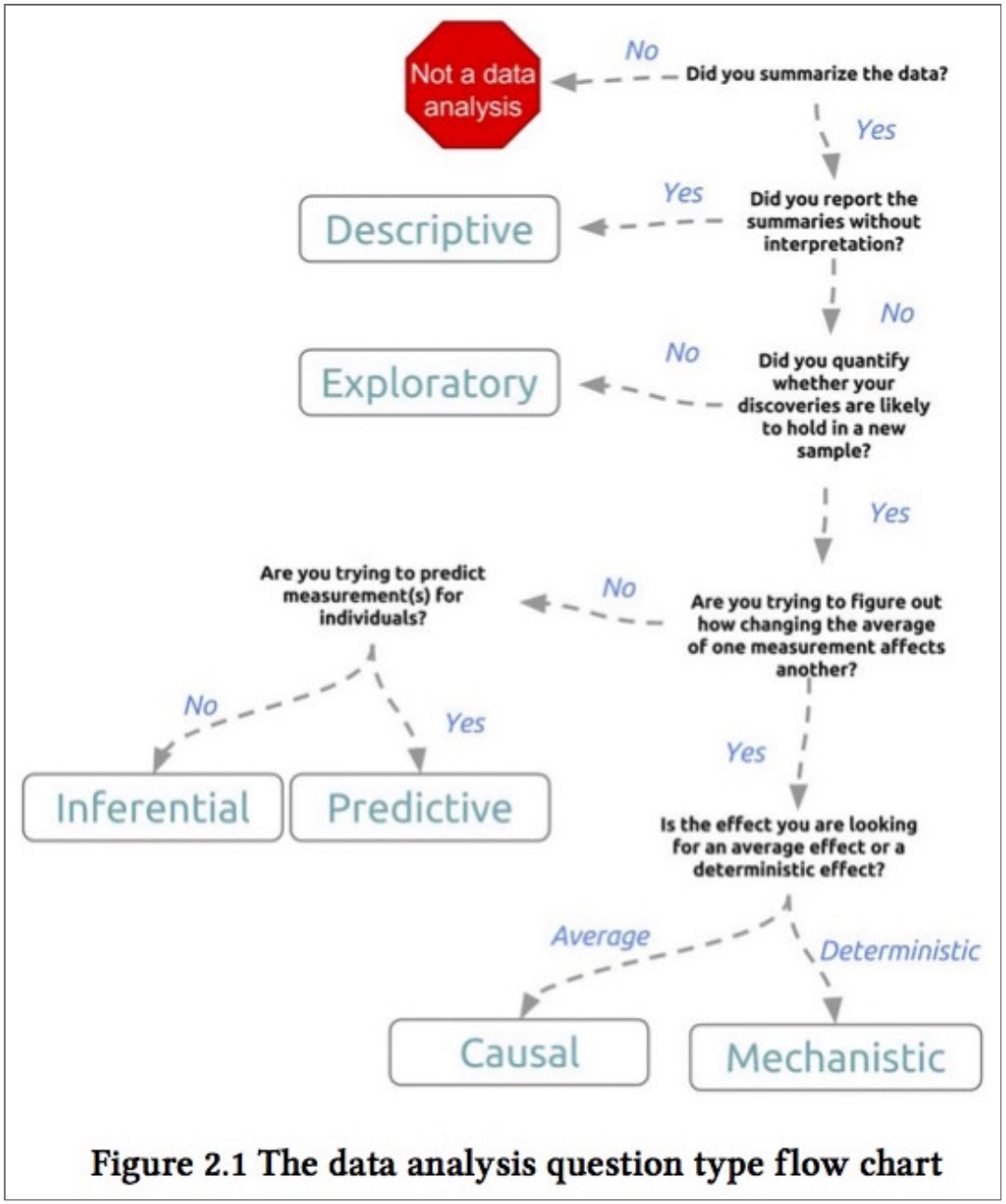


Figure 2.1 The data analysis question type flow chart

From *Elements of Data Analytic Style*.

Common Mistakes

- Correlation does not imply causation
- Overfitting
- $n = 1$ analysis (aka n of 1 analysis)
- Data dredging

Computing

R

- R is a programming language, a high-level “interpreted language”
- R is an interactive environment
- R is used for doing statistics and data science
- R is free and open-source
- R stays on the cutting-edge because of its ability to utilize independently developed “packages”
- R has some **peculiar featues** that experienced programmers should note (see **The R Inferno**)
- R has an **amazing community** of passionate users and developers

RStudio

- RStudio is an IDE (integrated development environment) for R
- It contains many useful features for using R
- We will use the free version of RStudio in this course

Getting Started in R

Calculator

Operations on numbers: `+ - * / ^`

```
> 2+1
[1] 3
```

```
> 6+3*4-2^3
[1] 10
```

```
> 6+(3*4)-(2^3)
[1] 10
```

Atomic Classes

There are five atomic classes (or modes) of objects in R:

1. character
2. complex
3. integer
4. logical
5. numeric (real number)

There is a sixth called “raw” that we will not discuss.

Assigning Values to Variables

```
> x <- "sml201" # character
> x <- 2+1i     # complex
> x <- 4L       # integer
> x <- TRUE     # logical
> x <- 3.14159  # numeric
```

Note: Anything typed after the # sign is not evaluated. The # sign allows you to add comments to your code.

More Ways to Assign Values

```
> x <- 1
> 1 -> x
> x = 1
```

In this class, we ask that you only use `x <- 1`.

Evaluation

When a complete expression is entered at the prompt, it is evaluated and the result of the evaluated expression is returned. The result may be auto-printed.

```
> x <- 1
> x+2
[1] 3
> print(x)
[1] 1
> print(x+2)
[1] 3
```

Functions

There are **many useful functions** included in R. “Packages” (covered later) can be loaded as libraries to provide additional functions. You can also write your own functions in any R session.

Here are some examples of built-in functions:

```
> x <- 2
> print(x)
[1] 2
> sqrt(x)
[1] 1.414214
> log(x)
[1] 0.6931472
> class(x)
[1] "numeric"
> is.vector(x)
[1] TRUE
```

Accessing Help in R

You can open the help file for any function by typing `?` with the functions name. Here is an example:

```
> ?sqrt
```

There's also a function `help.search` that can do general searches for help. You can learn about it by typing:

```
> ?help.search
```

It's also useful to use Google: for example, **"r help square root"**. The R help files are also on the web.

Variable Names

In the previous examples, we used `x` as our variable name. Do not use the following variable names, as they have special meanings in R:

```
c, q, s, t, C, D, F, I, T
```

When combining two words for a given variable, we recommend one of these options:

```
> my_variable <- 1
> myVariable <- 1
```

Variable names such as `my.variable` are problematic because of the special use of "." in R.

Vectors

The vector is the most basic object in R. You can create vectors in a number of ways.

```
> x <- c(1, 2, 3, 4, 5)
> x
[1] 1 2 3 4 5
>
> y <- 1:40
> y
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
[24] 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
>
> z <- seq(from=0, to=100, by=10)
> z
 [1]   0  10  20  30  40  50  60  70  80  90 100
> length(z)
[1] 11
```

Vectors

- Programmers: vectors are indexed starting at `1`, not `0`
- A vector can only contain elements of a single class:

```
> x <- "a"
> x[0]
character(0)
> x[1]
[1] "a"
>
> y <- 1:3
> z <- c(x, y, TRUE, FALSE)
> z
[1] "a"     "1"     "2"     "3"     "TRUE"  "FALSE"
```

Matrices

Like vectors, matrices are objects that can contain elements of only one class.

```
> m <- matrix(1:6, nrow=2, ncol=3)
> m
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
>
> m <- matrix(1:6, nrow=2, ncol=3, byrow=TRUE)
> m
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

Factors

In statistics, factors encode categorical data.

```
> paint <- factor(c("red", "white", "blue", "blue", "red",
+                   "red"))
> paint
[1] red   white blue  blue  red   red
Levels: blue red white
>
> table(paint)
paint
 blue   red white
    2     3     1
> unclass(paint)
[1] 2 3 1 1 2 2
attr(,"levels")
[1] "blue"  "red"   "white"
```

Lists

Lists allow you to hold different classes of objects in one variable.

```
> x <- list(1:3, "a", c(TRUE, FALSE))
> x
[[1]]
[1] 1 2 3

[[2]]
[1] "a"

[[3]]
[1]  TRUE FALSE
>
> # access any element of the list
> x[[2]]
[1] "a"
> x[[3]][2]
[1] FALSE
```

Lists with Names

The elements of a list can be given names.

```
> x <- list(counting=1:3, char="a", logic=c(TRUE, FALSE))
> x
$counting
[1] 1 2 3

$char
[1] "a"

$logic
[1]  TRUE FALSE
>
> # access any element of the list
> x$char
[1] "a"
> x$logic[2]
[1] FALSE
```

Missing Values

In data analysis and model fitting, we often have missing values. `NA` represents missing values and `NaN` means “not a number”, which is a special type of missing value.

```
> m <- matrix(nrow=3, ncol=3)
> m
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
[3,]   NA   NA   NA
> 0/1
[1] 0
> 1/0
[1] Inf
> 0/0
[1] NaN
```

NULL

`NULL` is a special type of reserved value in R.

```
> x <- vector(mode="list", length=3)
> x
[[1]]
NULL

[[2]]
NULL

[[3]]
NULL
```

Coercion

We saw earlier that when we mixed classes in a vector they were all coerced to be of type character:

```
> c("a", 1:3, TRUE, FALSE)
[1] "a"     "1"     "2"     "3"     "TRUE"  "FALSE"
```

You can directly apply coercion with functions `as.numeric()`, `as.character()`, `as.logical()`, etc.

This doesn't always work out well:

```
> x <- 1:3
> as.character(x)
[1] "1" "2" "3"
>
> y <- c("a", "b", "c")
> as.numeric(y)
Warning: NAs introduced by coercion
[1] NA NA NA
```

Data Frames

The data frame is one of the most important objects in R. Data sets very often come in tabular form of mixed classes, and data frames are constructed exactly for this.

Data frames are lists where each element has the same length.

Data Frames

```
> df <- data.frame(counting=1:3, char=c("a", "b", "c"),
+                  logic=c(TRUE, FALSE, TRUE))
> df
  counting char logic
1        1    a  TRUE
2        2    b FALSE
3        3    c  TRUE
```

Data Frames

```
> dim(df)
[1] 3 3
>
> names(df)
[1] "counting" "char"     "logic"
>
> attributes(df)
$names
[1] "counting" "char"     "logic"

$row.names
[1] 1 2 3

$class
[1] "data.frame"
```

Attributes

Attributes give information (or meta-data) about R objects. The previous slide shows `attributes(df)`, the attributes of the data frame `df`.

```
> x <- 1:3
> attributes(x) # no attributes for a standard vector
NULL
>
> m <- matrix(1:6, nrow=2, ncol=3)
> attributes(m)
$dim
[1] 2 3
>
> attributes(paint)
$levels
[1] "blue"  "red"   "white"

$class
[1] "factor"
```

Names

Names can be assigned to columns and rows of vectors, matrices, and data frames. This makes your code easier to write and read.

```
> names(x) <- c("Princeton", "Rutgers", "Penn")
> x
Princeton   Rutgers      Penn
        1         2         3
>
> colnames(m) <- c("NJ", "NY", "PA")
> rownames(m) <- c("East", "West")
> m
     NJ NY PA
East  1  3  5
West  2  4  6
> colnames(m)
[1] "NJ" "NY" "PA"
```

Accessing Names

Displaying or assigning names to these three types of objects does not have consistent syntax.

Object	Column Names	Row Names
vector	`names()`	N/A
data frame	`names()`	`row.names()`
matrix	`colnames()`	`rownames()`