Introduction

Introduction to Data Science Lecture 1

Kyle M. Lang

Department of Methodology & Statistics
Tilburg University

Block 4 2020

Outline

1. Overview of this course
2. What is data science?
3. Data science workflow
4. Data science novelties
5. Statistical modeling

Course Outline

Course Road-map

This course will be broken into two parts:

- We will spend the first three weeks on *supervised learning* methods.
- Around the beginning of May, we'll begin discussing *unsupervised learning* methods.

Course Structure

- For the first three weeks:
  - Two plenary lecture sessions per week
    - These lectures will be streamed via Zoom.
    - Hopefully, the live-stream will be recorded and posted to Canvas for students with time-zone difficulties.
  - Two lab meetings per week
    - I will pre-record a demonstration lecture and post the video to Canvas.
    - During the scheduled lab session, I will be available for questions via Zoom.
- For the final four weeks:
  - One plenary lecture session per week
  - One lab meeting per week
  - Kim will flesh-out the details at a later date.

Grading & Evaluation

- You will complete two group assignments.
  - One on supervised learning
  - One on unsupervised learning
- The course is rounded-off with a written final exam.
- Your course grade will be a weighted average of the grades you receive for the group assignments and your exam grade.
  - The assignments will contribute 40% to your grade.
  - The exam will contributed 60% to your grade.
  - The three grades *can* compensate one another.

Defining Data Science

What is data science?

At the very least, “data science” is a buzzword.

- It’s also a job
  - You can certainly be hired as a data scientist.

What is data science?

At the very least, “data science” is a buzzword.

- It’s also a job
  - You can certainly be hired as a data scientist.

In a strict sense, “data science” is almost certainly a misnomer.

- Data science is not the *science* of data.

What is data science?

A mixture of skills and a merger of disciplines:

- Statistics
- Computer science
- Mathematics
- Programming
- Data processing
- Data visualization
- Communication
- Substantive expertise

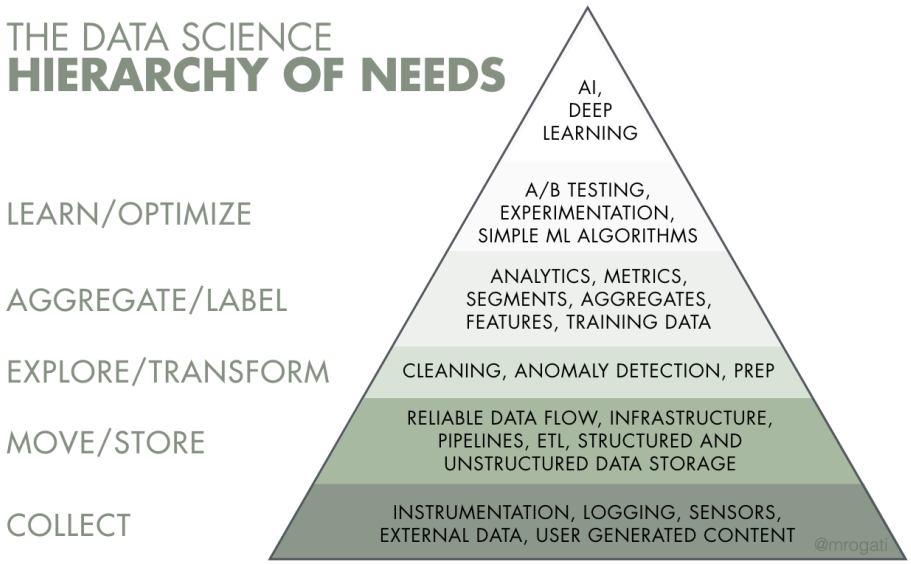
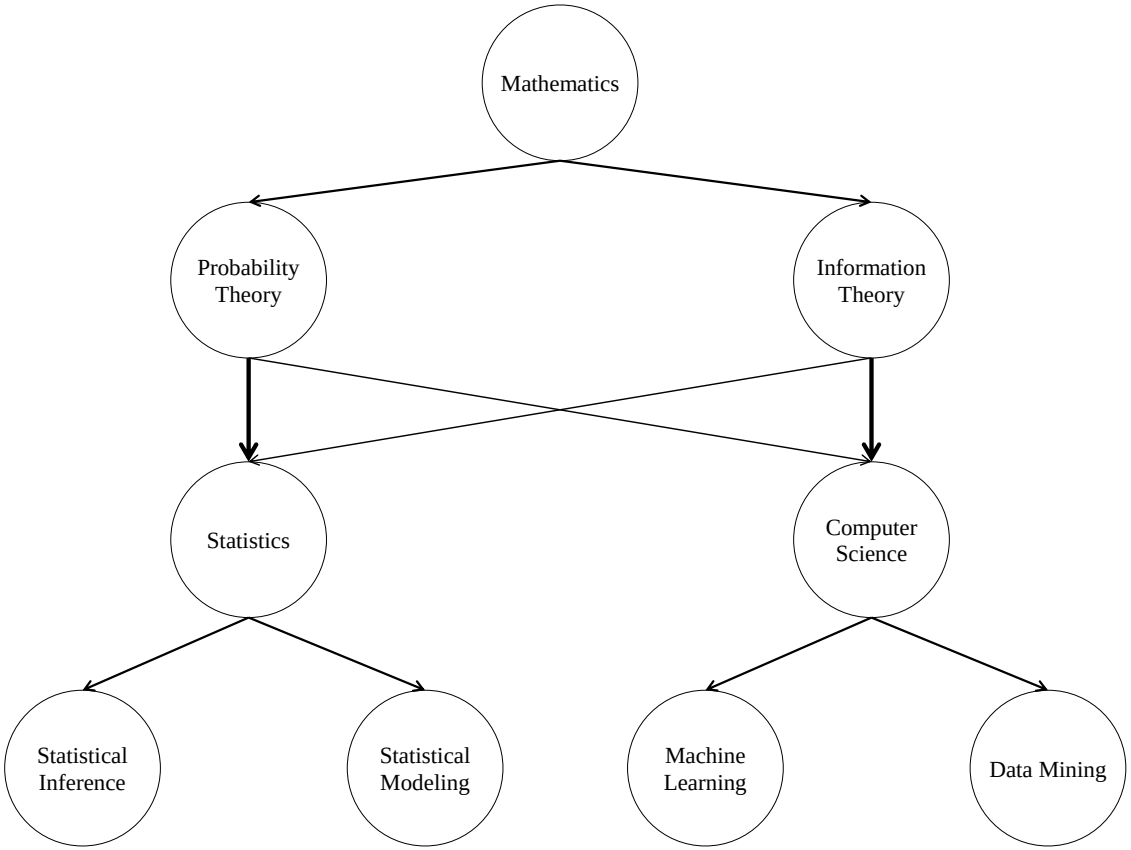


Figure source: `https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007`

Aside: Confusing Nomenclature

When you start studying this stuff, you will encounter a dizzying array of terms that all seem to describe the same things. Here's why:



What is data science?

A focus on practical problem solving

- Data analysis should create value.
  - We're trying to extract knowledge from data.
  - Start with a question and use data to answer it.
  - Don't start with data and generate answerable questions.
- Use appropriately complex methods.
  - Don't waste resources on complex analyses when simpler analyses will solve your problem equally well.
  - Don't settle for bad answers just because good answers will require complex/difficult analyses.
- Don't ask if you *can*; ask if you *should*.
  - Why are you doing a particular analysis?
  - All analytic decisions should be justified.

What is data science?

A strong focus on pragmatism and skepticism

- Don't be tied to a "pet method".
- Embrace exploratory methods.
  - Don't overgeneralize exploratory findings.
- Treat neither data nor theory as sacred.
  - Don't sanctify theory in the face of (definitively) contradictory data.
  - Don't blithely let data overrule well-supported theory.
- Trust no one.
  - Not data, other people, or yourself
  - Check and double check
- Don't assume what can be tested.
- When in doubt, err on the side of conservative inference.
- Document everything!

What is data science?

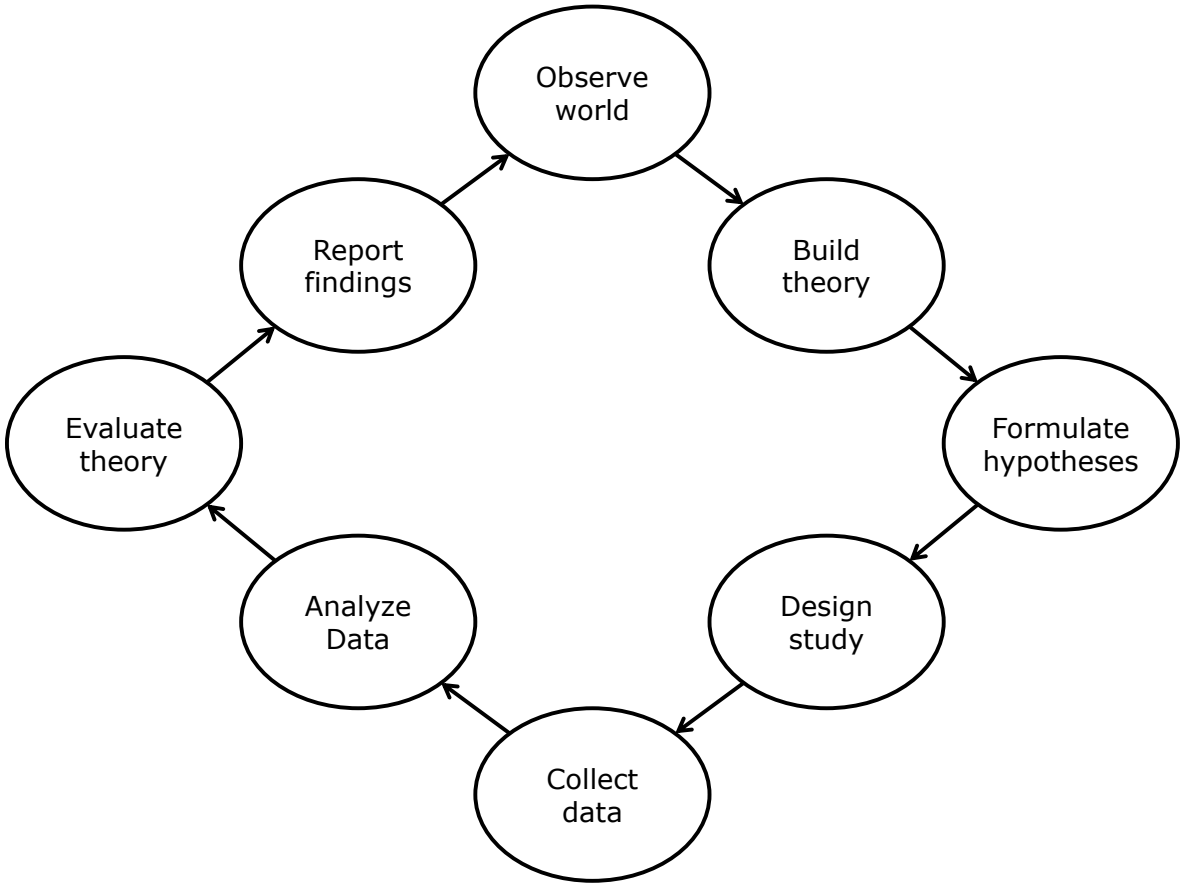
A fast-paced, curious, open-minded attitude

- Iterate quickly, fail quickly
- Never stop learning.
  - Learn and use new methods
  - Always remain open to new ideas/approaches.
- Don't be afraid to tackle new problems.
  - Generalize and extend what you know.
  - Don't stagnate.
- Show an appropriate degree of humility.
  - You don't know everything.
    - Embrace and correct your ignorance.
  - Ask questions.
    - Communicate. Don't just talk.

Data Science Workflow

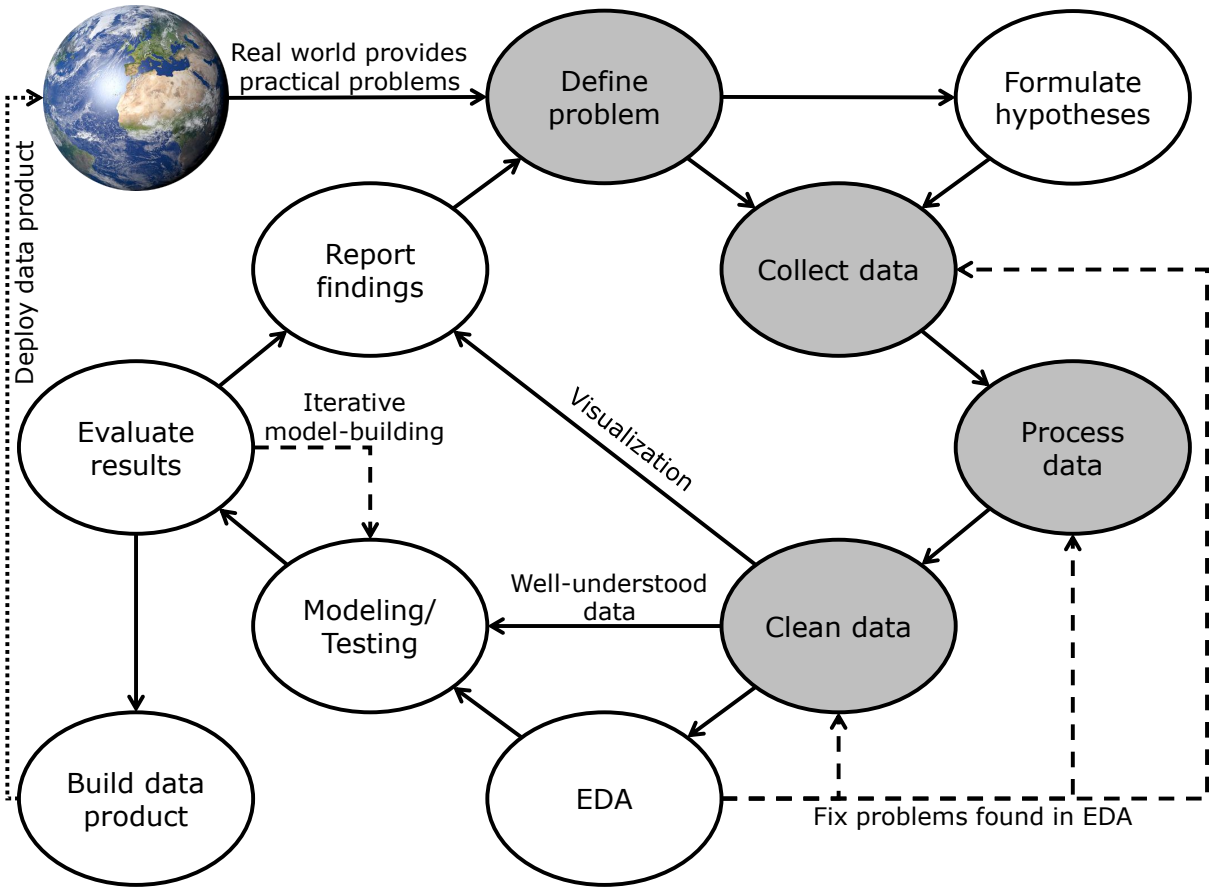
Research Cycle

The following is a representation of the *Research Cycle* used for empirical research in most of the sciences.



Data Science Cycle

The *Data Science Cycle* represented here was adapted from O'Neil and Schutt (2014).



Data Science Novelties

Novel Data Structures

In the social and behavioral sciences, we are accustomed to analyzing small, rectangular datasets.

- Rows represent observational units.
- Columns represent variables.

Data science applications deal with much more diverse forms of data.

- Relational databases
- Data streams
- Web logs
- Sensor data
- Image data
- Unstructured text

These datasets are often much larger and less structured that those traditionally analyzed in the social and behavioral sciences.

Parallel Processing/Distributed Computing

When dealing with large amounts of (distributed) data, we should move the data as little as possible.

- We can analyze distributed data *in situ* without moving them to a central computer.
  - Distributed computing

When executing long-running jobs, we should try to split the calculations into smaller pieces that can be executed simultaneously.

- Parallel processing

Parallel processing comes in two flavors:

- Embarrassingly Parallel
- Multi-threading

Parallel Processing Technologies

We can distribute embarrassingly parallel jobs directly.

- No real need for clever task partitioning

We must break multi-threaded jobs into independent subtasks.

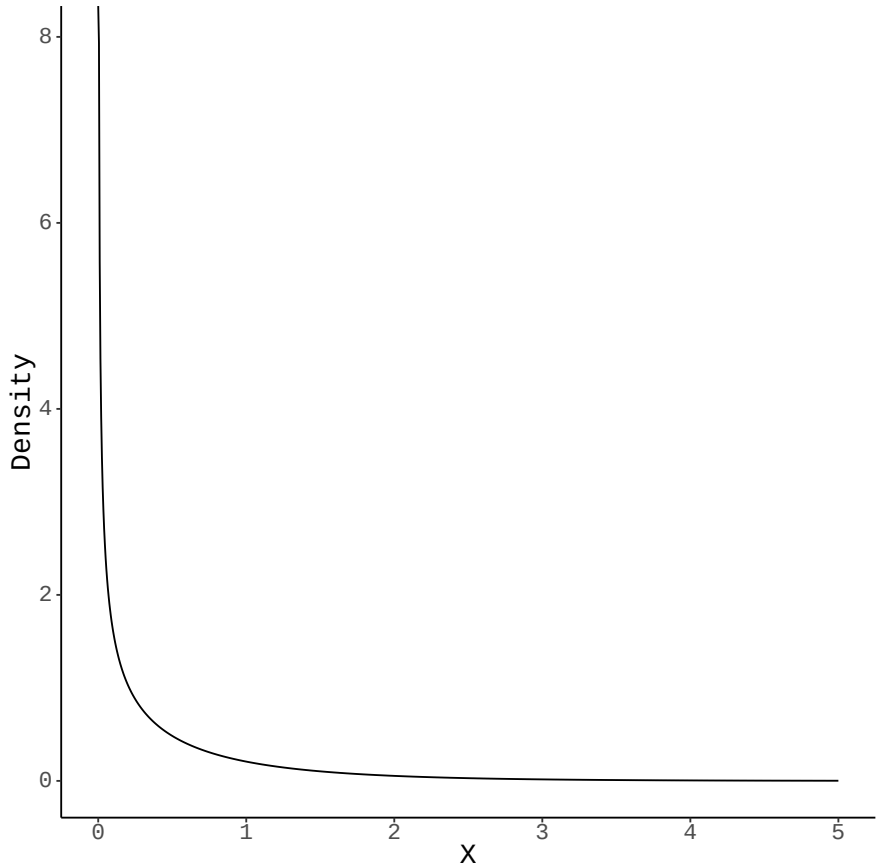
- Several technologies can facilitate multi-threading.
  - Small-scale:
    - Message Passing Interface (MPI)
    - Open Multi Processing (OpenMP)
  - Large-scale:
    - Google's MapReduce algorithm
    - Apache Hadoop
    - Apache Spark

Some software can use parallel processing behind-the-scenes.

Parallel Processing Example

Run a Monte Carlo simulation to test the central limit theorem.

- Population model:
  - $x_p \sim \Gamma(0.5, 1.0)$
- Parameters:
  - $P \in \{1, 2, \ldots, 50\}$
  - $N \in \{5, 10, \ldots, 100\}$
- Mean score for the nth row:
  - $\bar{x}_n = P^{-1} \sum_{p=1}^{P} x_{np}$
- Outcome:
  - KS statistic testing if $\bar{x}_n$ is normally distributed



Parallel Processing Example

First, we'll define a function to run one replication of the simulation:

```
## Run one replication of the simulation:
doRep <- function(rp, conds) {
    res <- rep(NA, nrow(conds))
    for(i in 1 : nrow(conds)) { # Loop through conditions
        ## Compute a mean score from p variables:
        x <- rowMeans(
            replicate(conds[i, "p"],
                      rgamma(conds[i, "n"], 0.5, 1)
                      )
        )

        ## Calculate the KS statistic:
        res[i] <- ks.test(x, "pnorm", mean(x), sd(x))$stat
    }
    cbind(conds, res) # Return the results
}
```

Parallel Processing Example

Then, we prepare the environment:

```
library(parallel) # We'll need this for parallel processing
library(lattice)  # We'll use this for plotting

## Define simulation conditions:
nVec  <- seq(5, 100, 5)
pVec  <- 1 : 50
conds <- expand.grid(n = nVec, p = pVec)

## How many replications?
nReps <- 500
```

Parallel Processing Example

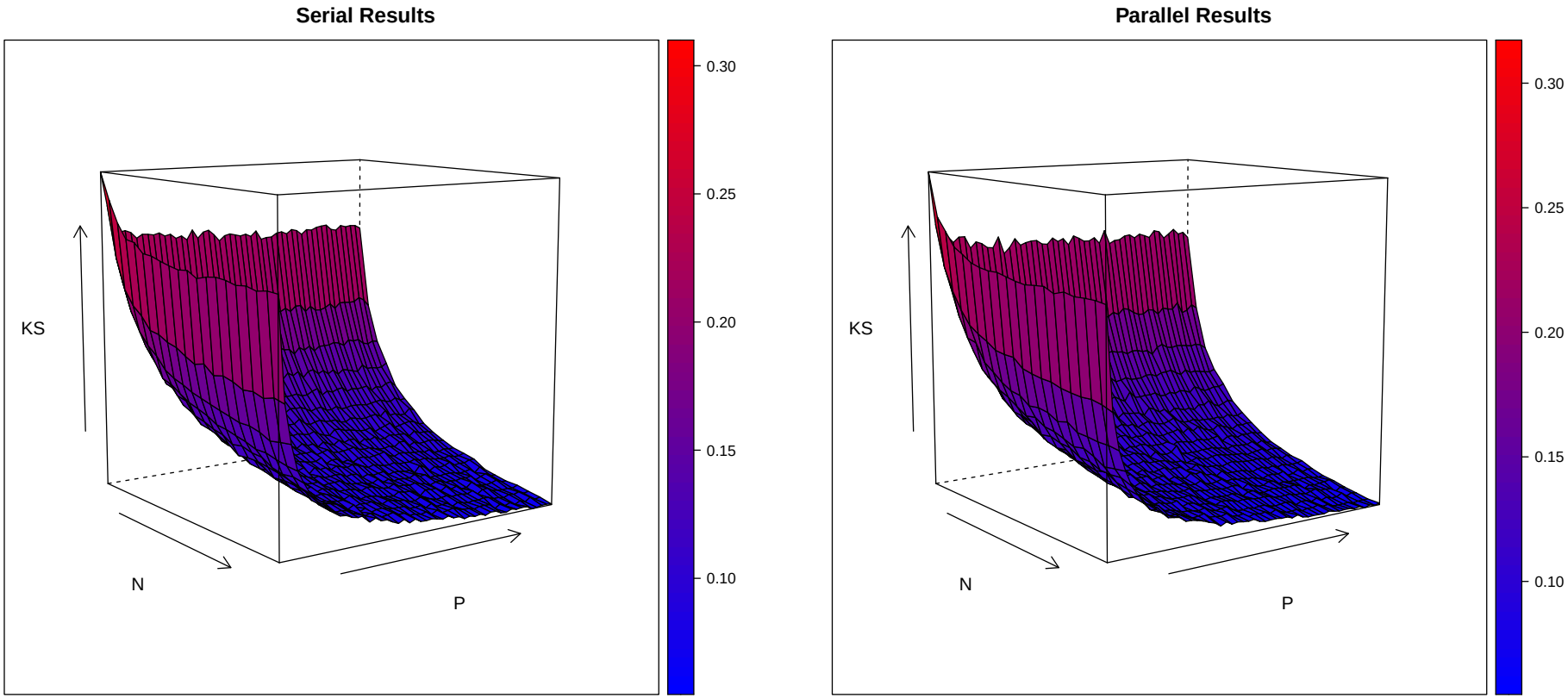
We'll run the simulation in two ways:

```
# Run 'nReps' replications in a loop:
t1 <- system.time(
{
    out1 <- list()
    for(rp in 1 : nReps)
        out1[[rp]] <- doRep(rp, conds = conds)
}
)

## Run 'nReps' replications in parallel using mclapply():
t2 <- system.time(
    out2 <- mclapply(X          = 1 : nReps,
                     FUN        = doRep,
                     conds      = conds,
                     mc.cores = 2)
)
```

Parallel Processing Example

We can visualize the results of our simulation:



Parallel Processing Example

Finally, we'll compare the computational speed of the parallel and serial approaches:

```
## Serial version:
t1

##    user  system elapsed
## 557.867   0.004 558.187

## Parallel version:
t2

##    user  system elapsed
## 271.906   0.128 276.540
```

Running the program in parallel substantially speeds computation.

- The parallel version is 2.02 times faster than the serial version.

Computationally Efficient Algorithms

In general, there will be many ways to estimate a given model.

- When dealing with large data structures, choosing a computationally efficient approach is important.

There will usually be a trade-off between memory efficiency and computational efficiency.

- We can compute faster by storing the result of initial calculations, but doing so entails higher memory usage.

Certain data structures should be analyzed with specialized computational techniques.

- Data streams → online learning, batch processing
- Distributed data → distributed computing

Online Learning Example

Suppose we want to estimate the mean of X.

- Traditional approach:

$$\bar{X} = N^{-1} \sum_{n=1}^{N} x_n$$

Maybe, we don't want to keep all of X in memory.

- Online updating:

$$\bar{X}_n = \frac{(n-1)\bar{X}_{n-1} + x_n}{n}$$

```
## Generate some data:
x <- runif(10000)

## Traditional estimation:
m0 <- sum(x) / length(x)

## Online updating:
m1 <- x[1]
for(n in 2 : length(x))
    m1 <- ((n - 1) * m1 + x[n]) / n

## Compare results:
m0; m1

## [1] 0.4995201
## [1] 0.4995201
```

Statistical Modeling

Statistical Reasoning

Statistics and data science are used to answer questions about hypothetical populations.

- Do men have higher job satisfaction than women?
- Can I predict your voting behavior?
- Can I detect groups of people who share similar attitudes towards climate change?

To answer these questions, we need to use *statistical reasoning*.

- The foundation of all good statistical analyses is a deliberate, careful, and thorough consideration of uncertainty.

Statistical Reasoning

If I measure a mean satisfaction rating for men of 5.6 and a mean satisfaction rating for women of 5.1, does that imply higher job satisfaction for men?

- Maybe...
- If the satisfaction ratings are highly variable, with respect to the size of the mean difference, we may not care much about the observed mean difference.
- The *observed* mean difference may not represent a *true* mean difference in the population.

The purpose of statistics is to systematize the way that we account for uncertainty when making data-based decisions.

Statistical Modeling

To implement this "statistical reasoning," we could use two different approaches: *statistical testing* or *statistical modeling*.

- In experimental contexts, real-world "messiness" is controlled through random assignment, and statistical testing is a sufficient method of knowledge generation.
- Apart from A/B testing, data scientists rarely have the luxury of being able to conduct experiments.
- Data scientists work with messy observational data and often don't have questions that lend themselves to straight-forward testing.

Data scientists need *statistical modeling*.

Statistical Modeling

Modelers attempt to build a mathematical representation of the (interesting aspects) of a data distribution.

- The model succinctly describes whatever system is being analyzed.
- Beginning with a model ensures that we are learning the important features of a distribution.
- The modeling approach is especially important in messy data science applications.

Two Modeling Traditions

Breiman (2001) defines two cultures of statistical modeling:

- Data models
- Algorithmic models

Two Modeling Traditions

Breiman (2001) defines two cultures of statistical modeling:

- Data models
- Algorithmic models

Data scientists use both types of models.

- Both types of model have strengths and weaknesses.
  - Data models tend to support a priori hypothesis testing more easily.
  - Data models also tend to provide more interpretable results.
  - Algorithmic models can't be beat for pure power.

Two Modeling Traditions

Breiman (2001) defines two cultures of statistical modeling:

- Data models
- Algorithmic models

Data scientists use both types of models.

- Both types of model have strengths and weaknesses.
  - Data models tend to support a priori hypothesis testing more easily.
  - Data models also tend to provide more interpretable results.
  - Algorithmic models can't be beat for pure power.

- Algorithmic models are currently preferred in cutting edge prediction/classification applications.

Two Modeling Traditions

Breiman (2001) defines two cultures of statistical modeling:

- Data models
- Algorithmic models

Data scientists use both types of models.

- Both types of model have strengths and weaknesses.
  - Data models tend to support a priori hypothesis testing more easily.
  - Data models also tend to provide more interpretable results.
  - Algorithmic models can't be beat for pure power.
- Algorithmic models are currently preferred in cutting edge prediction/classification applications.
- Many models can be viewed as data models or algorithmic models, depending on how they're used.

Characteristics of Models

Data models share several core features:

- Data models are built from probability distributions.
  - Data models are modular.
- Data models encode our hypothesized understanding of the system we're exploring.
  - Data models are constructed in a "top-down", theory-driven way.

Characteristics of Models

Data models share several core features:

- Data models are built from probability distributions.
  - Data models are modular.
- Data models encode our hypothesized understanding of the system we're exploring.
  - Data models are constructed in a "top-down", theory-driven way.

Algorithmic models are distinct from data models in several ways:

- Algorithmic models do not have to be built from probability distributions.
  - Often, they are based on a set of decision rules (i.e., an algorithm).
- Algorithmic models begin with an objective (i.e., a problem to solve) and seek the optimal solution, given the data.
  - They are built in a "bottom-up", data-driven way.

Data Modeling Example

Suppose we believe the following:

1. BMI is positively associated with disease progression in diabetic patients after controlling for age and average blood pressure.
2. After controlling for age and average blood pressure, the effect of BMI on disease progression is different for men and women.

We can represent these beliefs with a moderated regression model:

$$Y_{prog} = \beta_0 + \beta_1 X_{BMI} + \beta_2 X_{sex} + \beta_3 X_{age} + \beta_4 X_{BP} + \beta_5 X_{BMI} X_{sex} + \varepsilon$$

Data Modeling Example

We can use R to fit our model to some patient data:

```
library(rockchalk)

##
## Attaching package:  'rockchalk'
## The following object is masked from 'package:plyr':
##
##     summarize

## Load the data:
dataDir <- "../data/"
dDat    <- readRDS(paste0(dataDir, "diabetes.rds"))

## Fit the regression model:
fit <- lm(progress ~ bmi * sex + age + bp, data = dDat)
```

Data Modeling Example

```
partSummary(fit, -c(1, 2))

## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -174.7986    27.0004  -6.474 2.58e-10
## bmi            7.2106     0.8922   8.082 6.34e-15
## sexmale      -90.1718    35.1134  -2.568   0.0106
## age            0.1691     0.2322   0.728   0.4670
## bp             1.4032     0.2385   5.884 7.97e-09
## bmi:sexmale    3.0257     1.3090   2.311   0.0213
##
## Residual standard error: 59.68 on 436 degrees of freedom
## Multiple R-squared:  0.4075,Adjusted R-squared:  0.4007
## F-statistic: 59.98 on 5 and 436 DF,  p-value: < 2.2e-16
```

Data Modeling Example

We can do a simple slopes analysis to test the group-specific effects of BMI on disease progression:

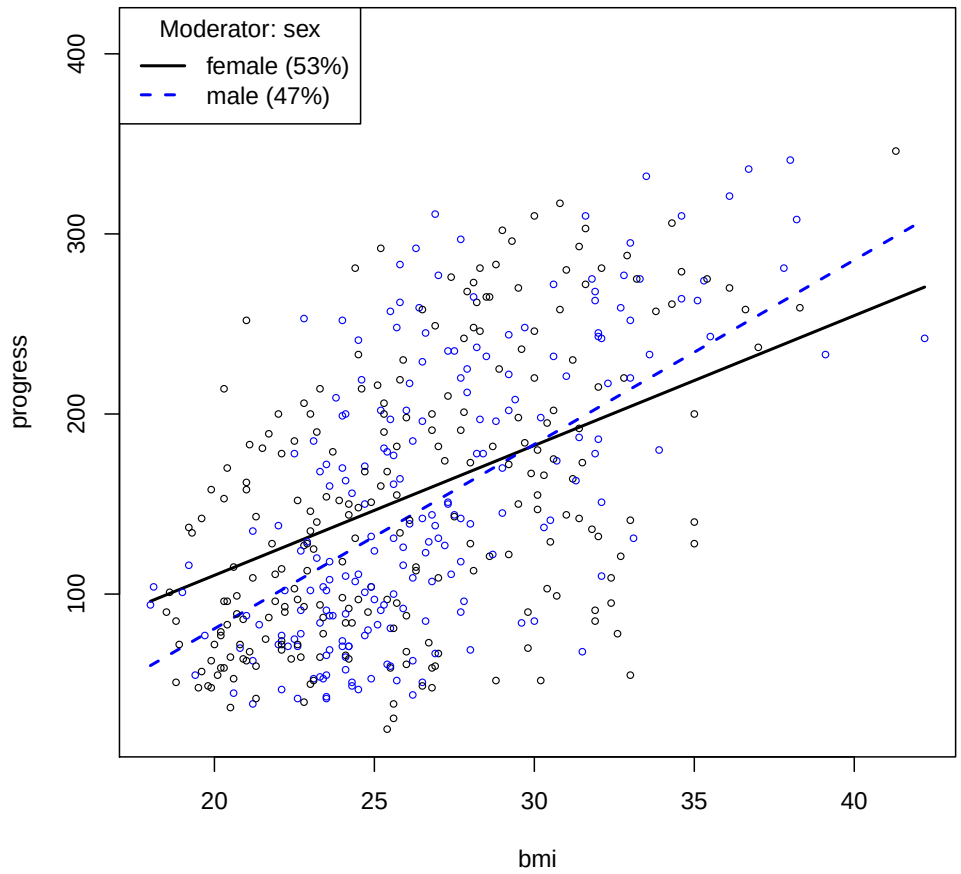
```
psOut <- plotSlopes(fit, plotx = "bmi", modx = "sex")
tsOut <- testSlopes(psOut)
```

```
tsOut$hypotests[ , -1]

##             slope Std. Error  t value     Pr(>|t|)
## female  7.210575  0.8921929 8.081856 6.335264e-15
## male   10.236323  1.0328739 9.910525 5.137409e-21
```

Data Modeling Example

We can also visualize the simple slopes:



Algorithmic Modeling Example

Suppose we want to find the best predictors of disease progression among the variables contained in our dataset:

- Age
- BMI
- Blood Pressure
- Blood Glucose
- Sex
- Total Cholesterol
- LDL Cholesterol
- HDL Cholesterol
- Triglycerides
- Lamorigine

We could try *best-subset selection*.

- Fit a series of regression models wherein disease progression is predicted by all possible subsets of X variables.
- Choose the set of X variables that minimizes the prediction error.

Algorithmic Modeling Example

```
library(leaps)

## Save the predictor variables' names:
xNames <- grep(pattern = "progress",
               x       = colnames(dDat),
               invert  = TRUE,
               value   = TRUE)

## Train the models:
fit <- regsubsets(x     = progress ~ .,
                  data  = dDat,
                  nvmax = ncol(dDat) - 1)

## Summarize the results:
sum <- summary(fit)
```

Algorithmic Modeling Example

```
sum$outmat

##              age bmi bp  tc  ldl hdl tch ltg glu sexmale
## 1   ( 1 )  " "  "*" " " " " " " " " " " " " " " " "  " "
## 2   ( 1 )  " "  "*" " " " " " " " " " " " " " "*" " " " "
## 3   ( 1 )  " "  "*" "*" " " " " " " " " " " " "*" " " " "
## 4   ( 1 )  " "  "*" "*" "*" " " " " " " " " "*" " " " "
## 5   ( 1 )  " "  "*" "*" " " " " " " "*" " " " "*" " " "*"
## 6   ( 1 )  " "  "*" "*" "*" "*" " " " " " " "*" " " "*"
## 7   ( 1 )  " "  "*" "*" "*" "*" " " " " "*" "*" " " "*"
## 8   ( 1 )  " "  "*" "*" "*" "*" " " " " "*" "*" "*" "*"
## 9   ( 1 )  " "  "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
## 10   ( 1 ) "*"  "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

Algorithmic Modeling Example

```
## Variables selected by BIC:
xNames[with(sum, which[which.min(bic), -1])]

## [1] "bmi" "bp"  "hdl" "ltg" "sex"

## Variables selected by Adjusted R^2:
xNames[with(sum, which[which.max(adjr2), -1])]

## [1] "bmi" "bp"  "tc"  "ldl" "tch" "ltg" "glu" "sex"

## Variables selected by Mallow's Cp:
xNames[with(sum, which[which.min(cp), -1])]

## [1] "bmi" "bp"  "tc"  "ldl" "ltg" "sex"
```

Algorithmic Modeling Example

The results seem to be highly sensitive to the error measure. What should we do?

Algorithmic Modeling Example

The results seem to be highly sensitive to the error measure. What should we do?

- We could pick our favorite error measure and use its results.
- We could throw our hands up in defeat and quit.
- We could look at the results and pick the answer we like best.
  - The previous two suggestions are sub-optimal, but this one is actually unethical. Don't do this!

Algorithmic Modeling Example

The results seem to be highly sensitive to the error measure. What should we do?

- We could pick our favorite error measure and use its results.
- We could throw our hands up in defeat and quit.
- We could look at the results and pick the answer we like best.
  - The previous two suggestions are sub-optimal, but this one is actually unethical. Don't do this!

If we think like a data scientist and get creative, we don't need to settle for these ambiguous results.

- We could implement a more robust method of calculating prediction error like *K-fold cross validation*.
- We can use resampling methods to quantify uncertainty in the variable selection process.

Algorithmic Modeling Example

```
bic <- r2 <- cp <- matrix(NA, 100, ncol(dDat) - 1)
for(rp in 1 : 100) {
    ## Resample the data:
    tmp <- dDat[sample(1 : nrow(dDat), nrow(dDat), TRUE), ]

    ## Train the models:
    fit <- regsubsets(x      = progress ~ .,
                      data   = tmp,
                      nvmax  = ncol(tmp) - 1)
    sum <- summary(fit)

    ## Save the optimal selections:
    bic[rp, ] <- with(sum, which[which.min(bic), -1])
    r2[rp, ]  <- with(sum, which[which.max(adjr2), -1])
    cp[rp, ]  <- with(sum, which[which.min(cp), -1])
}
```

Algorithmic Modeling Example

```
colMeans(bic)

##      age      bmi       bp       tc      ldl      hdl      tch
##     0.02     1.00     0.99     0.59     0.21     0.42     0.22
##      ltg      glu sexmale
##     1.00     0.10     0.87

colMeans(r2)

##      age      bmi       bp       tc      ldl      hdl      tch
##     0.28     1.00     1.00     0.94     0.71     0.32     0.57
##      ltg      glu sexmale
##     1.00     0.55     1.00

colMeans(cp)

##      age      bmi       bp       tc      ldl      hdl      tch
##     0.10     1.00     0.99     0.91     0.57     0.26     0.45
##      ltg      glu sexmale
##     1.00     0.37     1.00
```

Algorithmic Modeling Example

```
## Find the best subset via majority vote:
votes <- colMeans(rbind(bic, r2, cp)); round(votes, 3)

##     age     bmi      bp      tc     ldl     hdl     tch
##   0.133   1.000   0.993   0.813   0.497   0.333   0.413
##     ltg     glu sexmale
##   1.000   0.340   0.957

preds <- xNames[votes > 0.5]; preds

## [1] "bmi" "bp"  "tc"  "ltg" "sex"

## Fit the winning model to the original data:
form <- paste0("progress ~ ",
               paste(preds, collapse = " + ")
               )
fit  <- lm(form, data = dDat)
```

Algorithmic Modeling Example

```
partSummary(fit, -c(1, 2))

## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -335.11146   25.68289 -13.048  < 2e-16
## bmi            6.47376    0.68565   9.442  < 2e-16
## bp             1.05016    0.21789   4.820 1.99e-06
## tc            -0.29836    0.08833  -3.378 0.000796
## ltg           60.36010    6.49158   9.298  < 2e-16
## sexmale      -14.14306    5.40833  -2.615 0.009231
##
## Residual standard error: 54.83 on 436 degrees of freedom
## Multiple R-squared:  0.4999,Adjusted R-squared:  0.4941
## F-statistic: 87.15 on 5 and 436 DF,  p-value: < 2.2e-16
```

Data Science for the Social and Behavioral Sciences

Social and behavioral scientists specialize in the types of problems for which a data science approach is most beneficial.

- Social systems are messy, noisy, and chaotic.
- Social systems are usually complex systems.
- Social scientific constructs tend to be difficult to measure.
- Human behavior produces a lot of data.
- Many layers of uncertainty open the door for a host of poor/unethical research practices.

References

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231.

O'Neil, C., & Schutt, R. (2014). *Doing data science: Straight talk from the frontline*. Sebastopol, CA: O'Reilly Media, Inc.