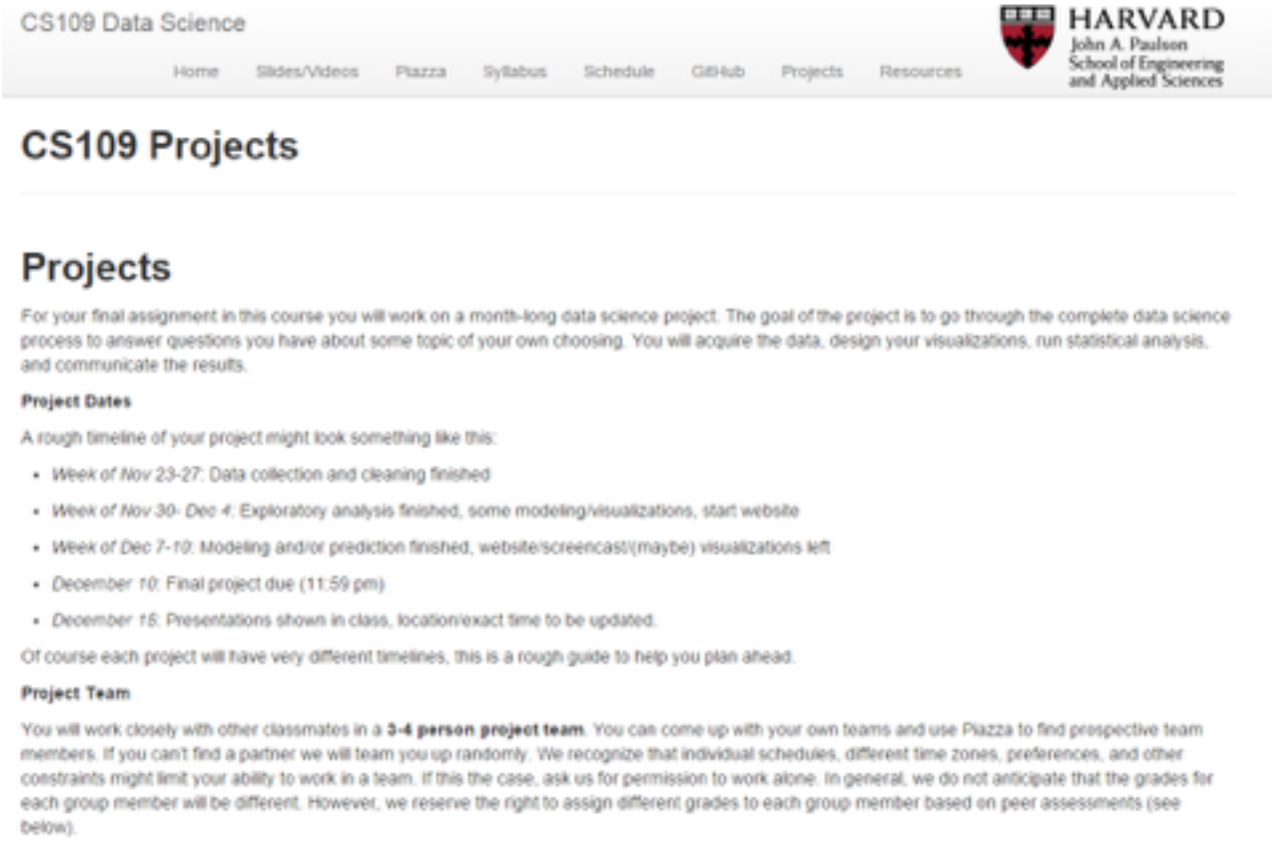
Data Science – CS109

Joe Blitzstein, Verena Kaynig-Fittkau, Hanspeter Pfister

Final Projects

http://cs109.github.io/2015/pages/projects.html



CS109 Projects

Projects

For your final assignment in this course you will work on a month-long data science project. The goal of the project is to go through the complete data science process to answer questions you have about some topic of your own choosing. You will acquire the data, design your visualizations, run statistical analysis, and communicate the results.

Project Dates

A rough timeline of your project might look something like this:

- *Week of Nov 23-27*: Data collection and cleaning finished
- *Week of Nov 30- Dec 4*: Exploratory analysis finished, some modeling/visualizations, start website
- *Week of Dec 7-10*: Modeling and/or prediction finished, website/screencast/(maybe) visualizations left
- *December 10*: Final project due (11:59 pm)
- *December 15*: Presentations shown in class, location/exact time to be updated.

Of course each project will have very different timelines, this is a rough guide to help you plan ahead.

Project Team

You will work closely with other classmates in a **3-4 person project team**. You can come up with your own teams and use Piazza to find prospective team members. If you can't find a partner we will team you up randomly. We recognize that individual schedules, different time zones, preferences, and other constraints might limit your ability to work in a team. If this the case, ask us for permission to work alone. In general, we do not anticipate that the grades for each group member will be different. However, we reserve the right to assign different grades to each group member based on peer assessments (see below).

Project Dates

- *Week of Nov 23-27*: Data collection and cleaning finished
- *Week of Nov 30- Dec 4*: Exploratory analysis finished, some modeling/visualizations, start website
- *Week of Dec 7-10*: Modeling and/or prediction finished, website/screencast/(maybe) visualizations left
- *Th, December 10*: Final project due (11:59 pm)
- *December 15*: Best projects shown in class (location TBA)

IPython Process Book

- important part of your project
- standalone document that **fully** describes your project in detail
- overview and motivation
- related Work
- initial Questions
- data: source, scraping method, cleanup, etc.
- exploratory data analysis
- final analysis
- Use visualizations!

Code – Zen (Picks)

- The Zen of Python, by Tim Peters
- >>> import this

- **Explicit is better than implicit**
- **Simple is better than complex**
- Flat is better than nested
- **Readability counts**
- Errors should never pass silently
- Now is better than never
- **If the implementation is hard to explain, it's a bad idea**

Website

- Use Github, Google Sites, Personal Sites
- Open to the public
- Links to Ipython notebook and data
- Visualizations
- Video

Video

- 2 min limit
- Use a good microphone
- Do not just scroll through your notebook!
- Use what you learned about story telling and presentations
- Do not put in too much
- Show demos! Show visualizations! Be engaging

Some Examples

- Predicting Hubway availability, L. Alexander, G. Goulet-Langlois, J. Wolff
  https://www.youtube.com/watch?v=2wK8jpNMjXI&feature=youtu.be
- AirBnB success, H. Husain, Y. Mao, L. Awad, X. Li
  https://www.youtube.com/watch?v=raGjUj5qArc
- The green canvas, A. Hosny, J. Huang , Y. Wang
  https://vimeo.com/114379373

Peer Assessment

- Preparation - were they prepared during team meetings?
- Contribution - did they contribute productively to the team discussion and work?
- Respect for others' ideas - did they encourage others to contribute their ideas?
- Flexibility - were they flexible when disagreements occurred?

- You will also assess your own performance

What we learned

A (cs109) Data Scientist Is...

- a programmer: Python
- a statistician: Statistical Learning Theory
- a computer scientist: Algorithms, Risk Minimization
- an engineer: AWS, Spark, installs, debugging, etc
- a story teller: essays, presentations, projects
- an artist: visualizations, how to show information

Some Key Principles

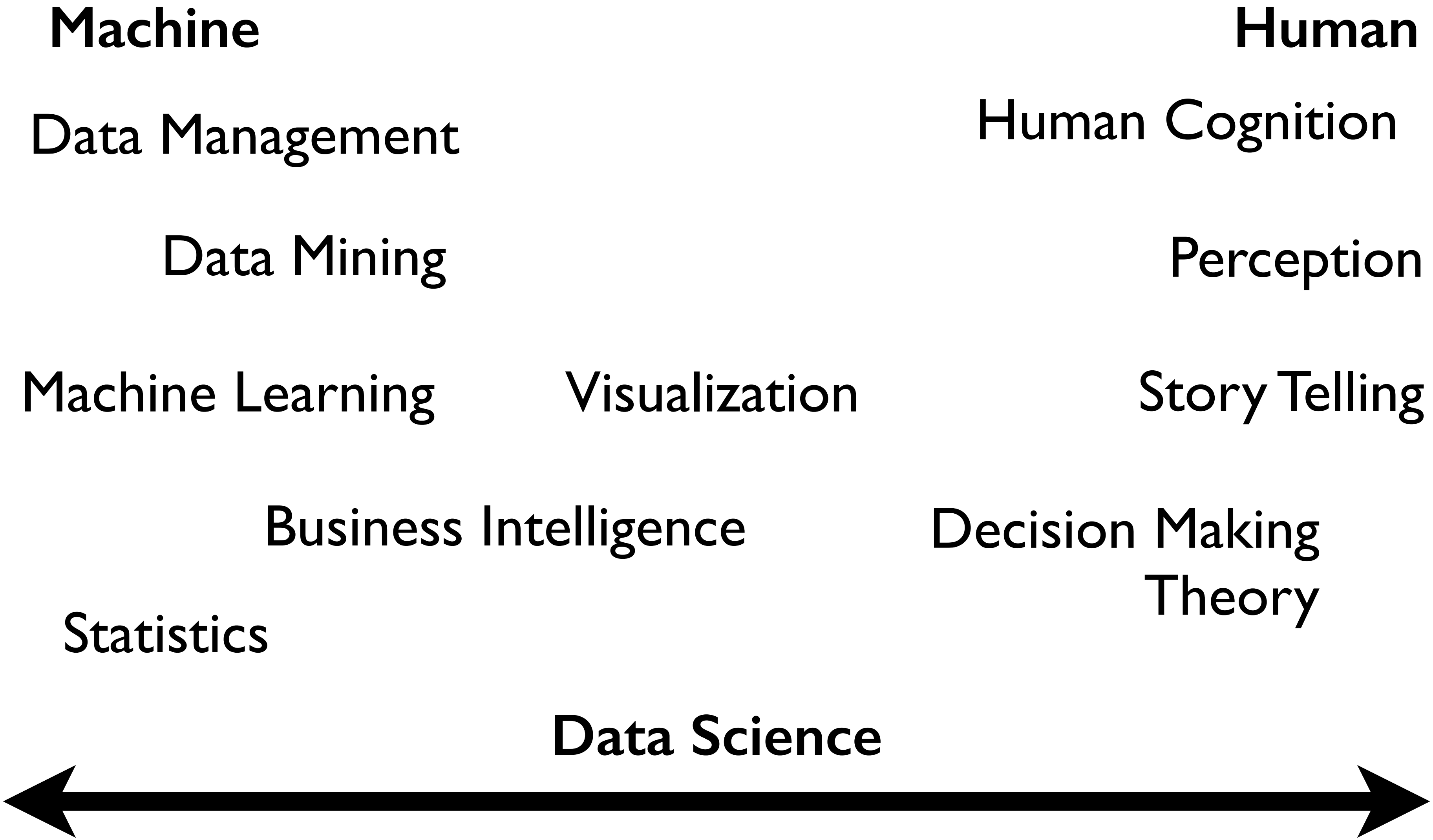
- *use many data sources* (the plural of anecdote is not data)
- *understand how the data were collected* (sampling is essential)
- *weight the data thoughtfully* (not all polls are equally good)
- *use statistical models* (not just hacking around in Excel)
- *understand correlations* (e.g., states that trend similarly)
- *think like a Bayesian, check like a frequentist* (reconciliation)
- *have good communication skills* (What does a 60% probability even mean? How can we visualize, validate, and understand the conclusions?)

What we did do

- Learning a model: EDA, play-viz, simplicity, complexity, features, overfitting, train-test, validation, cross-validation, regularization
- Statistics: Simple Distributions, LLN, CLT, Hypothesis tests, Multiple Comparisons, Bootstrap, Bayesian Stats, Hierarchical Models, regression-to-the-mean, regularization
- Machine Learning: (Empirical) Risk Minimization, Decision theory, Feature selection, Pipelines, Dimensionality and reduction (PCA), Regression, Classification, Clustering. Generative vs Discriminative vs Discriminant.
- Models: Perceptron, SVM, Logistic, LDA, NB, K-Means, LDA, Hierarchical Clustering, Bagging, Boosting, Stacking, RF.
- Engineering: Data Model - SQL, Dataframes, Scraping, Cleaning, Imputing, AWS, Containers, Struggling.
- Story Telling: Writing, Presenting, Visualization, Stories.

What we didn’t do (but could have)

- Optimization of a general loss: Gradient Descent, Stochastic Gradient Descent
- Inference on a ROC curve (Inference in General)
- Connect the Bayesian techniques to the frequentist
- More Ops: Big Data, SQL, AWS, Docker…



Machine

Human

Data Management

Human Cognition

Data Mining

Perception

Machine Learning

Visualization

Story Telling

Business Intelligence

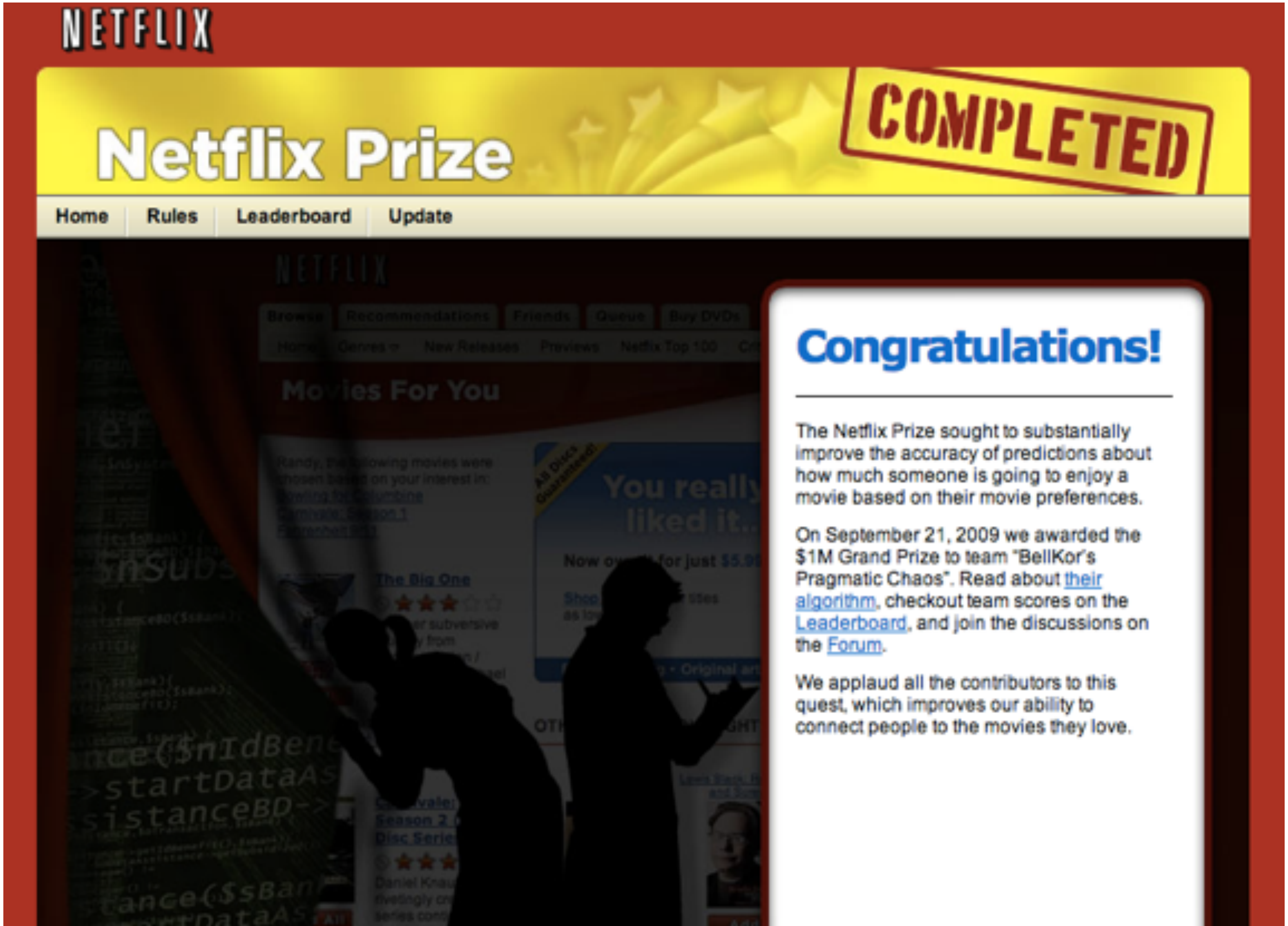
Decision Making Theory

Statistics

Data Science

Inspired by Daniel Keim, "Visual Analytics: Definition, Process, and Challenges"

Netflix Prize



Congratulations!

The Netflix Prize sought to substantially improve the accuracy of predictions about how much someone is going to enjoy a movie based on their movie preferences.

On September 21, 2009 we awarded the $1M Grand Prize to team "BellKor's Pragmatic Chaos". Read about their algorithm, checkout team scores on the Leaderboard, and join the discussions on the Forum.

We applaud all the contributors to this quest, which improves our ability to connect people to the movies they love.

Some Challenges

- *massive data* (500k users, 20k movies, 100m ratings)
- *curse of dimensionality* (very high-dimensional problem)
- *missing data* (99% of data missing; *not* missing at random)
- *extremely complicated set of factors that affect people's ratings of movies* (actors, directors, genre, ...)
- *need to avoid overfitting* (test data vs. training data)
- what model? local? global? ensemble?

Why do this?

"...10 years from now, each cancer patient is going to want to get a genomic analysis of their cancer and will expect customized therapy based on that information."

Director, The Cancer Genome Atlas (TCGA), Time Magazine, 6/13/11

"By 2018, the US could face a shortage of up to 190,000 workers with analytical skills"

McKinsey Global Institute

"The sexy job in the next 10 years will be ~~statisticians.~~" *Data Scientists?*

Hal Varian, Prof. Emeritus UC Berkeley
Chief Economist, Google

External Courses

- https://www.coursera.org/learn/machine-learning/ (log in to access videos): Andrew Ng
- https://www.coursera.org/course/machlearning : Pedro Domingos (also his book, The Master Algorithm)
- Learning From Data (more mathy, with book): https://work.caltech.edu/telecourse.html
- Harvard Computefest
- Stanford Statistical Learning MOOC: http://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/

Quick Reads

- A Few Useful Things to Know about Machine Learning: https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf
- Bayesian Reasoning and Machine Learning Chapter13 (http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=Brml.Online)
- Murphy Chapter 1: https://www.cs.ubc.ca/~murphyk/MLbook/

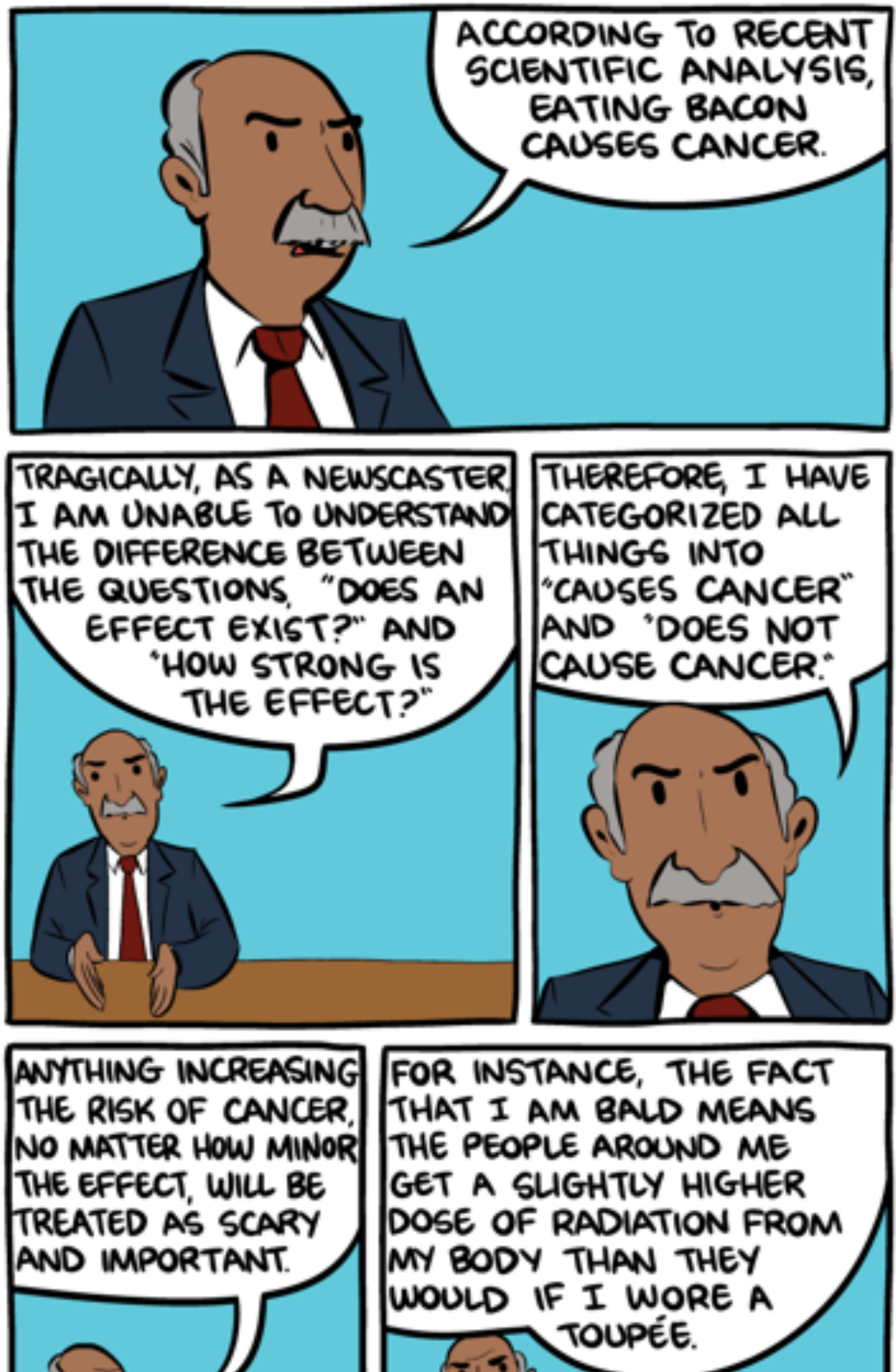
Useful free books

- http://statweb.stanford.edu/~tibs/ElemStatLearn/
- http://www-bcf.usc.edu/~gareth/ISL/
- http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=Brml.Online
- http://www.inference.phy.cam.ac.uk/itila/book.html
- https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers (also dead tree now)
- http://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/

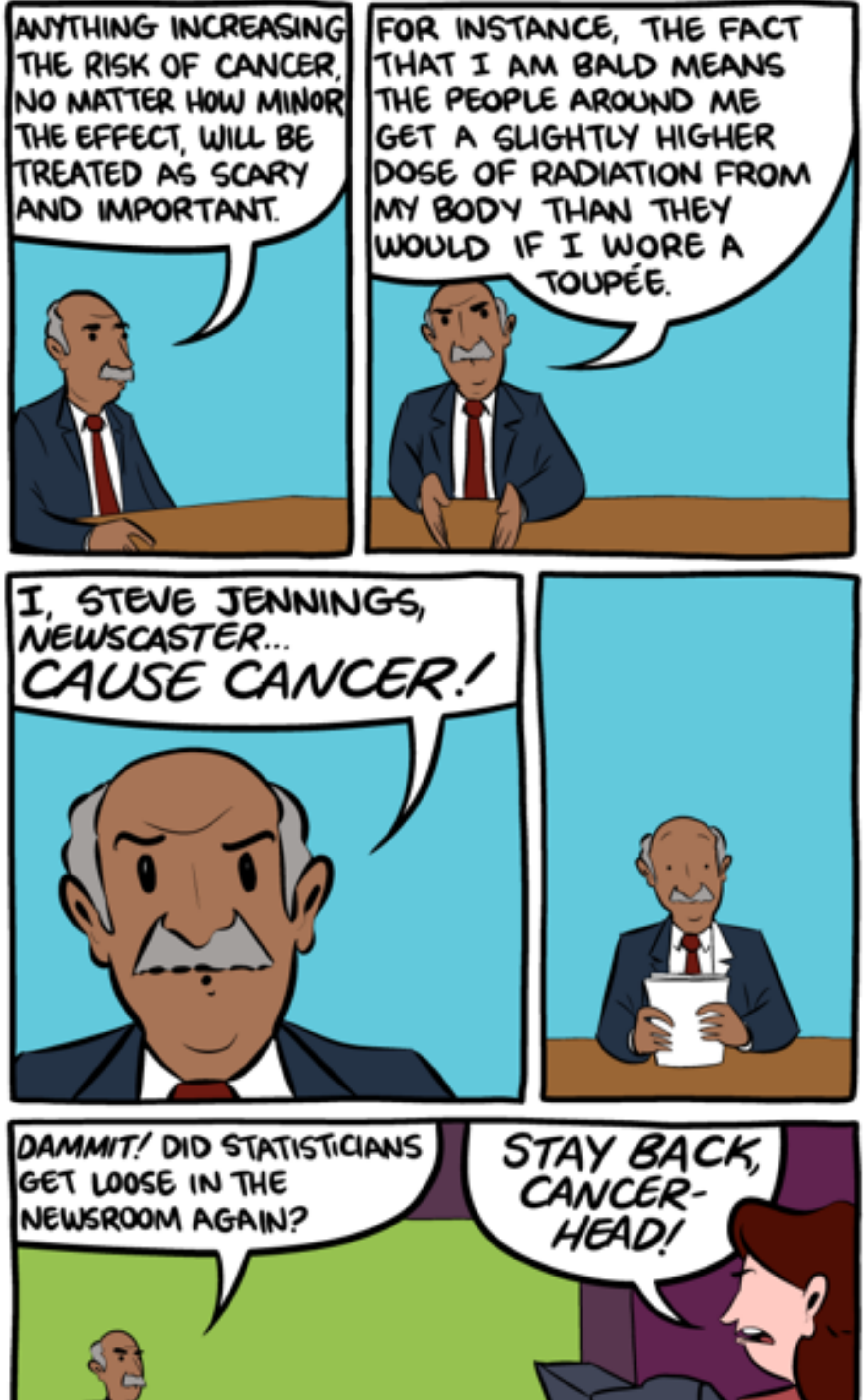
Some other resources

- Non free: Very recent but very relevant: https://github.com/rasbt/python-machine-learning-book (also read Sebastian's Blog)
- simplystatistics.org
- The YHat Blog and the RODEO IDE.
- Chris Fonnesbeck's BIOS 8366 at Vanderbilt: http://stronginference.com/Bios8366/
- Follow Liz Cherney, Andrej Karpathy, Yhat, Chris Albon, John Myles White, Cam Davidson Pilon and their networks.

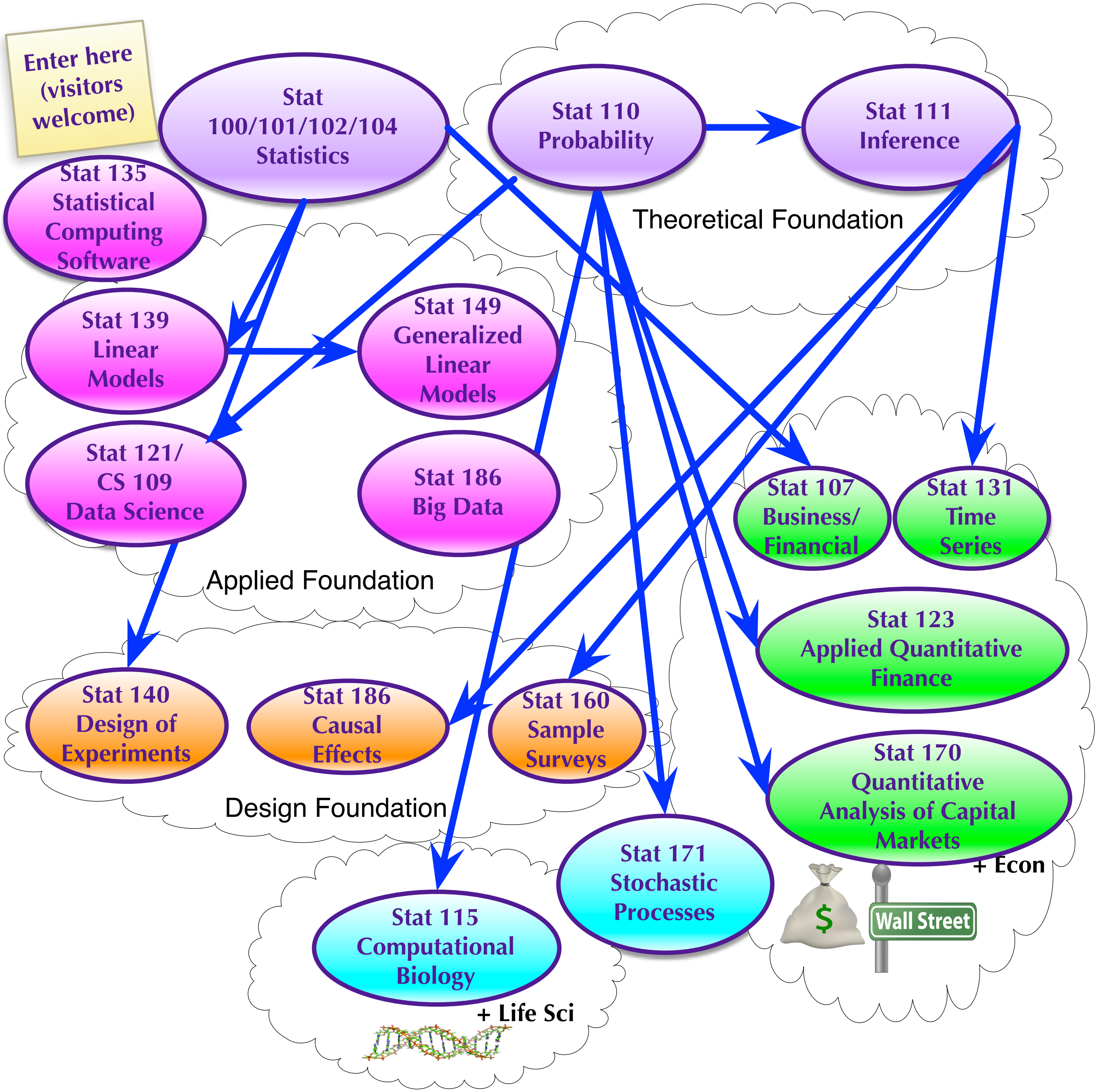
Next Step: Stats Courses


ACCORDING TO RECENT SCIENTIFIC ANALYSIS, EATING BACON CAUSES CANCER.
TRAGICALLY, AS A NEWSCASTER I AM UNABLE TO UNDERSTAND THE DIFFERENCE BETWEEN THE QUESTIONS, "DOES AN EFFECT EXIST?" AND "HOW STRONG IS THE EFFECT?"
THEREFORE, I HAVE CATEGORIZED ALL THINGS INTO "CAUSES CANCER" AND "DOES NOT CAUSE CANCER."
ANYTHING INCREASING THE RISK OF CANCER, NO MATTER HOW MINOR THE EFFECT, WILL BE TREATED AS SCARY AND IMPORTANT.
FOR INSTANCE, THE FACT THAT I AM BALD MEANS THE PEOPLE AROUND ME GET A SLIGHTLY HIGHER DOSE OF RADIATION FROM MY BODY THAN THEY WOULD IF I WORE A TOUPÉE.

http://smbc-comics.com/index.php?id=3930


ANYTHING INCREASING THE RISK OF CANCER, NO MATTER HOW MINOR THE EFFECT, WILL BE TREATED AS SCARY AND IMPORTANT.
FOR INSTANCE, THE FACT THAT I AM BALD MEANS THE PEOPLE AROUND ME GET A SLIGHTLY HIGHER DOSE OF RADIATION FROM MY BODY THAN THEY WOULD IF I WORE A TOUPÉE.
I, STEVE JENNINGS, NEWSCASTER... CAUSE CANCER!
DAMMIT! DID STATISTICIANS GET LOOSE IN THE NEWSROOM AGAIN?
STAY BACK, CANCER-HEAD!

http://smbc-comics.com/index.php?id=3930



Enter here (visitors welcome)

Stat 100/101/102/104 Statistics

Stat 110 Probability

Stat 111 Inference

Theoretical Foundation

Stat 135 Statistical Computing Software

Stat 139 Linear Models

Stat 149 Generalized Linear Models

Stat 121/ CS 109 Data Science

Stat 186 Big Data

Applied Foundation

Stat 107 Business/ Financial

Stat 131 Time Series

Stat 123 Applied Quantitative Finance

Stat 170 Quantitative Analysis of Capital Markets

+ Econon

Stat 140 Design of Experiments

Stat 186 Causal Effects

Stat 160 Sample Surveys

Design Foundation

Stat 171 Stochastic Processes

Stat 115 Computational Biology

+ Life Sci

Next Step: CS / AM Courses

https://sites.google.com/a/g.harvard.edu/seasthreeyearcourseplan/

Data Wrangling

- Python
- Pandas
- Spark
- Map Reduce

CS124 -Data Structures and Algorithms

Spring / Prof. Michael Mitzenmacher

- Fundamentals
- Graph Algorithms
- Greedy Algorithms
- Dynamic Programming
- Divide and Conquer
- Hashing
- Linear Programming
- Randomized Algorithms
- NP-completeness review
- Novel approaches to NP-complete problems

CS 165 – Data Systems

Fall / Prof. Stratos Idreos

- **Expected learning outcomes**
- Become familiar with the history and evolution of data systems design over the past 4-5 decades.
- Understanding the basic tradeoffs in designing and implementing modern data systems.
- Being able to design a new data system given a data-driven scenario and to built a prototype.
- Being able to understand which data system is a good fit given the needs of an application.
- Advanced C programming and debugging skills.

CS207- Systems Development for Computational Science

Spring / Dr. Rahul Dave

- Apply basic software development tool-chains, including source-code control, testing frameworks, and documentation tools, to the process of designing and implementing large software systems;
- Apply design principles to the decomposition of software into re-usable components, and to the production of those components;
- Know how to approach an existing piece of software for maintenance, extension, and modification;
- Design, develop, and deploy a set of software components to produce a scalable, reliable, and reproducible experimental system for scientific investigation;
- Use a variety of approaches to software development team organization, and select techniques that are appropriate in different circumstances.

CS205 – Computing Foundations for Computational Science

Fall / Dr. Ray Jones

- Apply basic computer science concepts such as modularity, abstraction, and encapsulation to scientific problems
- Recognize and recall computer architectures, algorithms, and data structures that are relevant to computational science
- Apply concepts of parallel programming and “parallel thinking” to computational science
- Analyze and visualize large scientific data and implement data-intensive computations on cluster and cloud infrastructures
- Use open-source tools for large- and fine-grain parallel computations, cloud computing, and visualization

CS262 - Introduction to Distributed Computing

Spring / Prof. Jim Waldo

- design and implementation of large systems
- running on multiple computers connected by a network.
- investigate the fundamental characteristics of distributed systems
- investigate how to build systems that exploit those fundamental characteristics.

Visualization and Storytelling

- EDA
- Effective visualization
- Interactive visualization
- Story telling

CS171-Visualizations

Spring / Prof. Hanspeter Pfister

www.cs171.org

- **After successful completion of this course, you will be able to…**
- Critically evaluate visualizations and suggest improvements and refinements
- Apply a structured process to design interactive visualizations
- Iteratively generate visual encoding ideas using sketching and prototyping
- Use principles of human perception and cognition in visualization design
- Know various visualization methods to judge and create new designs
- Create web-based interactive visualizations using JavaScript and D3
- Work constructively as a member of a team to plan and carry out a complex project

Machine Learning

- Supervised learning:
  - K-nearest neighbors
  - SVM
  - Decision Trees
  - Random Forest
  - Boosting
- Unsupervised learning:
  - Dimensionality reduction
  - Clustering

CS181 – Machine Learning

Spring / Prof. Finale Doshi-Velez

- *Introduction and Course Overview*
- *Clustering with K-Means and K-Medoids*
- *Hierarchical Clustering*
- *Principal Component Analysis*
- *Supervised Learning*
- *Linear Regression*
- *Model Selection*
- *Linear Classification*
- *Classification and CV Review*
- *Probabilistic Classification*
- *Neural Networks*
- *Regression and Classification Trees*
- *Max-Margin Classification*
- *Support Vector Machines*
- *Markov Decision Processes*
- *Reinforcement Learning*
- *Partially-Observable Markov Decision Processes*
- *Expectation Maximization*
- *Hidden Markov Models*

CS182 – Artificial Intelligence

Fall / Prof. Sasha Rush

- **After successful completion of this course, you will be able to…**
- choose the appropriate representation for an AI problem or domain model, and construct domain models in that representation
- choose the appropriate algorithm for reasoning within an AI problem domain
- implement and debug core AI algorithms in a clean and structured manner
- design and analyze the performance of an AI system or component
- describe AI algorithms and representations and explain their performance, in writing and orally
- critically read papers on AI systems

CS187 – Computational Linguistics

Fall / Prof. Stuart Shieber

Watson is the world Jeopardy champion. Siri responds accurately to "Should I bring an umbrella tomorrow?". How do they work? This course provides an introduction to the field of computational linguistics, the study of human language using the tools and techniques of computer science, with applications to a variety of natural-language-processing problems such as those deployed in Watson and Siri, and covers pertinent ideas from linguistics, logic programming, and statistical modeling. The course will include an experimental practicum component covering skills in technical writing and editing that should be of general use as well.

CS283 – Computer Vision

Fall / Prof. Todd Zickler

The goal of computer vision is to create artificial vision systems that can reliably extract information from images. The field is at an exciting stage of development, spurred by the explosion of online imagery, the emergence of new visual sensors, and the growth of processing and storage capabilities. After decades of hard work, we are finally starting see some stunning applications. I argue that the time for vision is now.

Addressing the vision problem requires an understanding of image formation (radiometry, optics, projective and differential geometry) as well as a firm grasp of mathematical and computational tools for image analysis. So this course, which is designed to be a first course in computer vision, covers a bit of both. It also blends theory and practice.

AM207 - Monte Carlo Methods for Inference and Data Analysis

Spring / Dr. Verena Kaynig

- Introduction to basic Monte Carlo methods
- Bayes formalism and sampling
- Stochastic Optimization
- Dynamic systems
- Advanced sampling methods
- Graphical Models

CS105 – Privacy and Technology (GOV1430)

Fall / Prof. Jim Waldo

- Privacy Concepts
- Surveillance
- Tapping and tracing
- Data aggregation, analytics, and privacy
- Data-intensive Science
- Biometrics and DNA

IACS Harvard John A. Paulson School of Engineering and Applied Sciences

IACS Institute for Applied Computational Science

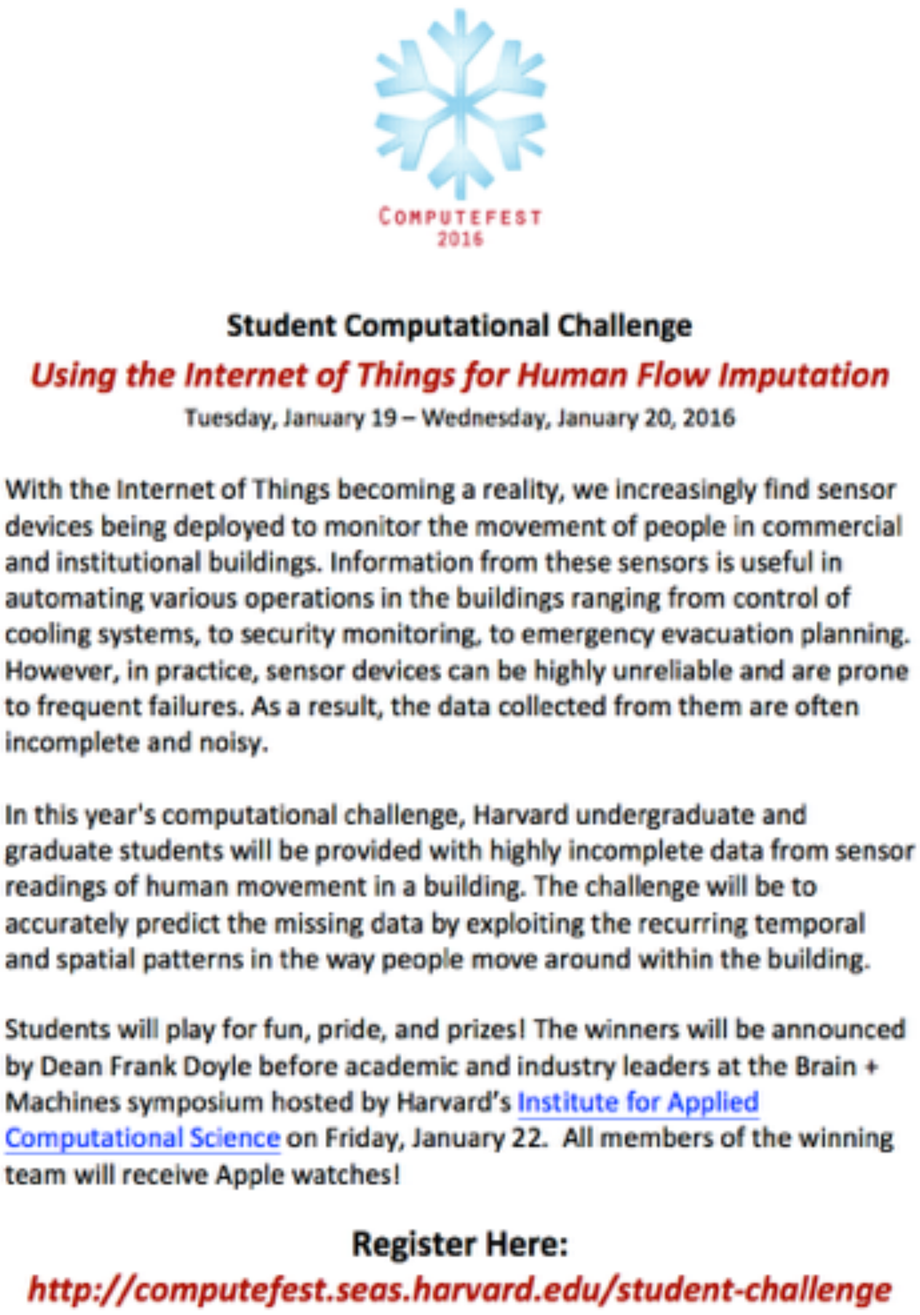
- Master of Science in CSE
  - 1 year, 8 courses
- Master of Engineering in CSE
  - 2 years, 8 courses, 1 thesis
- AB/SM in CSE option

- For PhD students:
  - secondary field in CSE
  - 4 courses

IACS Compute Fest

- Skill building workshops
  - Monday, January 11 - Friday, January 15
- Student computational challenge!
  - Tuesday, January 19- Thursday, January 21
- IACS Symposium: Brain & Machines
  - Friday, January 22

http://computefest.seas.harvard.edu



COMPUTEFEST
2016

Student Computational Challenge

Using the Internet of Things for Human Flow Imputation

Tuesday, January 19 – Wednesday, January 20, 2016

With the Internet of Things becoming a reality, we increasingly find sensor devices being deployed to monitor the movement of people in commercial and institutional buildings. Information from these sensors is useful in automating various operations in the buildings ranging from control of cooling systems, to security monitoring, to emergency evacuation planning. However, in practice, sensor devices can be highly unreliable and are prone to frequent failures. As a result, the data collected from them are often incomplete and noisy.

In this year's computational challenge, Harvard undergraduate and graduate students will be provided with highly incomplete data from sensor readings of human movement in a building. The challenge will be to accurately predict the missing data by exploiting the recurring temporal and spatial patterns in the way people move around within the building.

Students will play for fun, pride, and prizes! The winners will be announced by Dean Frank Doyle before academic and industry leaders at the Brain + Machines symposium hosted by Harvard's Institute for Applied Computational Science on Friday, January 22. All members of the winning team will receive Apple watches!

Register Here:

http://computefest.seas.harvard.edu/student-challenge