

A photograph of the iconic clock tower at Vanderbilt University, a tall red brick structure with two clock faces and a crenellated top. The tower is set against a sky with soft, wispy clouds. In the foreground, there are trees with autumn-colored leaves in shades of orange, red, and yellow. The overall scene is bright and slightly hazy, suggesting a pleasant day.

Human Factors: Quantitative and Qualitative Methods

Yu Huang

Vanderbilt University

yu.huang@vanderbilt.edu

Final Schedule

- Teams, paper presentations
- Due date for HW2
 - Sep 25 (current plan) or Oct 6 (after all mid-proposal presentations)?

We want to improve productivity and reduce cost in software development and maintenance.

What is software engineering?

Programs

- Testing
- Fault localization
- Static analysis
- Dynamic analysis
- Debugging
- ...

Programmers

- Will programmers use these tools? Why or why not?
- How do experts become experts?
- How to be productive?
- Biases?
- How to make a team function?
- How to estimate effort?
- ...

The Human Aspect Matters



Captain Sully

Chesley (Sully) Sullenberger clarified vividly **the significance of the “human factor”** in our digital age. After saving 155 people by landing his disabled Airbus A320 in the Hudson River in January 2009, Sully became a national hero.



Sichuan Airlines Flight 8633

At the altitude of 9 km (30,000 ft; 9,000 m), the right front segment of the windshield separated from the aircraft followed by an uncontrolled decompression. The flight control unit was damaged, and the loud external noise made spoken communications impossible. Because the flight was within a mountainous region, the pilots were unable to descend to the required 8,000 ft (2,400 m) to compensate for the loss of cabin pressure. The sudden loss of pressure in the cockpit had caused multiple instruments to fail.

*The half-body of copilot was sucked out of the window and the pilot kept flown **by manual and sight**. The three pilots were in short sleeves and suddenly it was -40°C in the cockpit. After 35 minutes, the crew made an emergency landing. 2 crew members were injured.*

"Epic-level diversion".

The Human Aspect Matters

1. The Mariner 1 Spacecraft, 1962

The first entry in our rundown goes right back to the sixties.

Before the summer of love or the invention of the lava lamp, NASA launched a space mission to fly past Venus. It did not go to plan.

The [Mariner 1 space probe](#) barely made it out of Cape Canaveral before the rocket course. Worried that the rocket was heading towards a crash-landing on earth destruct command and the craft was obliterated about 290 seconds after launch.

5. EDS Child Support System, 2004

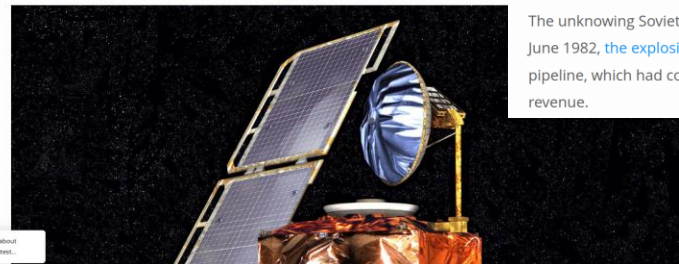
Back in 2004, the UK government introduced a new and complex system to manage the operations of the [Child Support Agency \(CSA\)](#). The contract was awarded to IT services company Electronic Data Systems (EDS). The system was called CS2, and there were problems as soon as it went live.

A leaked internal memo at the time revealed that the system was “badly designed, badly tested and badly implemented”. The agency reported that CS2 “had over 1,000 reported problems, of which 400 had no known workaround”, resulting in “around 3,000 IT incidents a week”. The system was budgeted to cost around £450 million, but ended up costing an [estimated £768 million altogether](#). EDS, a Texas-based contractor, also announced a \$153 million loss in their subsequent financial results.

7. NASA's Mars Climate Orbiter, 1998

Losing \$20 from your wallet is probably enough to ruin your day — how would [spacecraft](#)? NASA engineers found out back in 1998 when the Mars Climate Orbiter got too close to the surface of Mars.

It took engineers several months to work out what went wrong. It turned out to be a mistake in converting imperial units to metric. According to the [investigation report](#) software produced by Lockheed Martin used imperial measurements, while the software by NASA, was programmed with SI metric units. The overall cost of the failed mission was \$125 million.



2. The Morris Worm, 1988

Not all costly software errors are worn by big companies or government organizations. In fact, the [most costly software bugs](#) ever was caused by a single student. A Cornell University student created a computer virus as part of an experiment, which ended up spreading like wildfire and crashing tens of thousands of computers due to a coding error.

The computers were all connected through a very early version of the internet, making the worm essentially the first infectious computer virus. Graduate student Robert Tappan Morris was charged and convicted of [criminal hacking and fined \\$10,000](#), although the cost of the damage was estimated to be as [high as \\$10 million](#).

History has forgiven Morris though, with the incident now widely credited for exposing vulnerabilities in computer security. These days, Morris is a professor at MIT and the worm's source code is housed in a floppy disc at the University of Boston.



8. Soviet Gas Pipeline Explosion, 1982

This error is a little bit different to the others, as it was deliberate ([or so rumor has it](#)). In fact, the Soviet gas pipeline explosion is alleged to be a [cunning example of cyber-espionage](#), carried out by the CIA.

Back in 1982, at the height of the cold war tensions between the USA and USSR, the Soviet government built a gas pipeline that ran on advanced automated control software. The Soviets planned to steal from a Canadian company that specialized in this kind of programming.

According to accounts, the CIA convinced the Canadians to place deliberate bugs in the Soviet pipeline.

The unknowing Soviets went at the end of June 1982, [the explosion occurred](#) in the pipeline, which had cost tens of millions of dollars in revenue.

10. ESA Ariane 5 Flight V88, 1996

Given the complexity and expense of space exploration, it's no wonder that software errors are on our list of all-time software errors. However, the European Space Agency's Ariane 5 flight V88 is an even harsher cautionary tale than the rest, as it was caused by more than 100 software errors. Just 36 seconds after its maiden launch, the rocket engines failed due to a software error from Ariane 4 and a conversion error from 64-bit to 16-bit data.

The failure resulted in a \$370 million loss for the ESA, and a whole host of [subsequent investigation](#), including calls for improved software analysis and evaluation.

3. Pentium FDIV Bug, 1994

The [Pentium FDIV bug](#) is a curious case of a minor problem that had a major impact. Thomas Nicely, a math professor, discovered a flaw in the Pentium's floating-point division response was to offer a replacement chip to anyone who could provide a list of affected users.

The original error was relatively simple, with a problem in the logic that caused tiny inaccuracies in calculations, but only very rarely. In fact, it was only discovered because of a math professor's curiosity.

6. Heathrow Terminal 5 Opening, 2008

Imagine prepping to jet off on your eagerly-awaited vacation or important business trip, only to find that your flight is grounded or your luggage is nowhere to be seen.

This was exactly what happened to thousands of travelers when [Heathrow's Terminal 5 opened back in March 2008](#), and it was a disaster that performed well on malfunctioning luggage.

British Airways also reported a similar problem at its airport. Over the next few days, more than £16 million worth of luggage was lost.

9. Knight's \$440M in bad trades, 2012

Losing \$440 million is a bad day at the office by anyone's standards. Even more so when it happens in just 30 minutes due to a software error that wipes 75% off the value of one of the biggest capital groups in the world.

Knight Capital Group had invested in new trading software that was supposed to help them make a killing on the stock markets. Instead, it ended up killing their firm. Several software errors combined to send Knight on a crazy buying spree, spending more than \$7 billion on 150 different stocks.

11. The Millennium Bug, 2000

The Millennium Bug, AKA the notorious [Y2K](#), was a massive concern in the lead-up to the year 2000. The concern was that computer systems around the world would not be able to cope with dates after December 31, 1999, due to the fact that most computers and operating systems only used two digits to represent the year, disregarding the 19 prefix for the twentieth century. Dire predictions were made about the implosion of banks, airlines, power suppliers and critical data storage. How would systems deal with the 00 digits?

The anticlimatic answer was “pretty well, actually”. The millennium bug was a bit of a non-starter and didn't cause too many real-life problems, as most systems made adjustments in advance. However, the fear caused by the potential fallout throughout late 1999 cost thousands of considerable amounts of money in contingency planning and preparations, with institutions, businesses and even families expecting the worst.

The [USA spent vast quantities](#) to address the issue, with some estimates [putting the cost at \\$100 billion](#).

4. Bitcoin Hack, Mt. Gox, 2011

Mt. Gox was the biggest bitcoin exchange in the world in the 2010s, until they were hit by a software error that ultimately proved fatal.

The [glitch](#) led to the exchange creating transactions that could never be fully redeemed, costing up to \$1.5 billion in lost bitcoins.

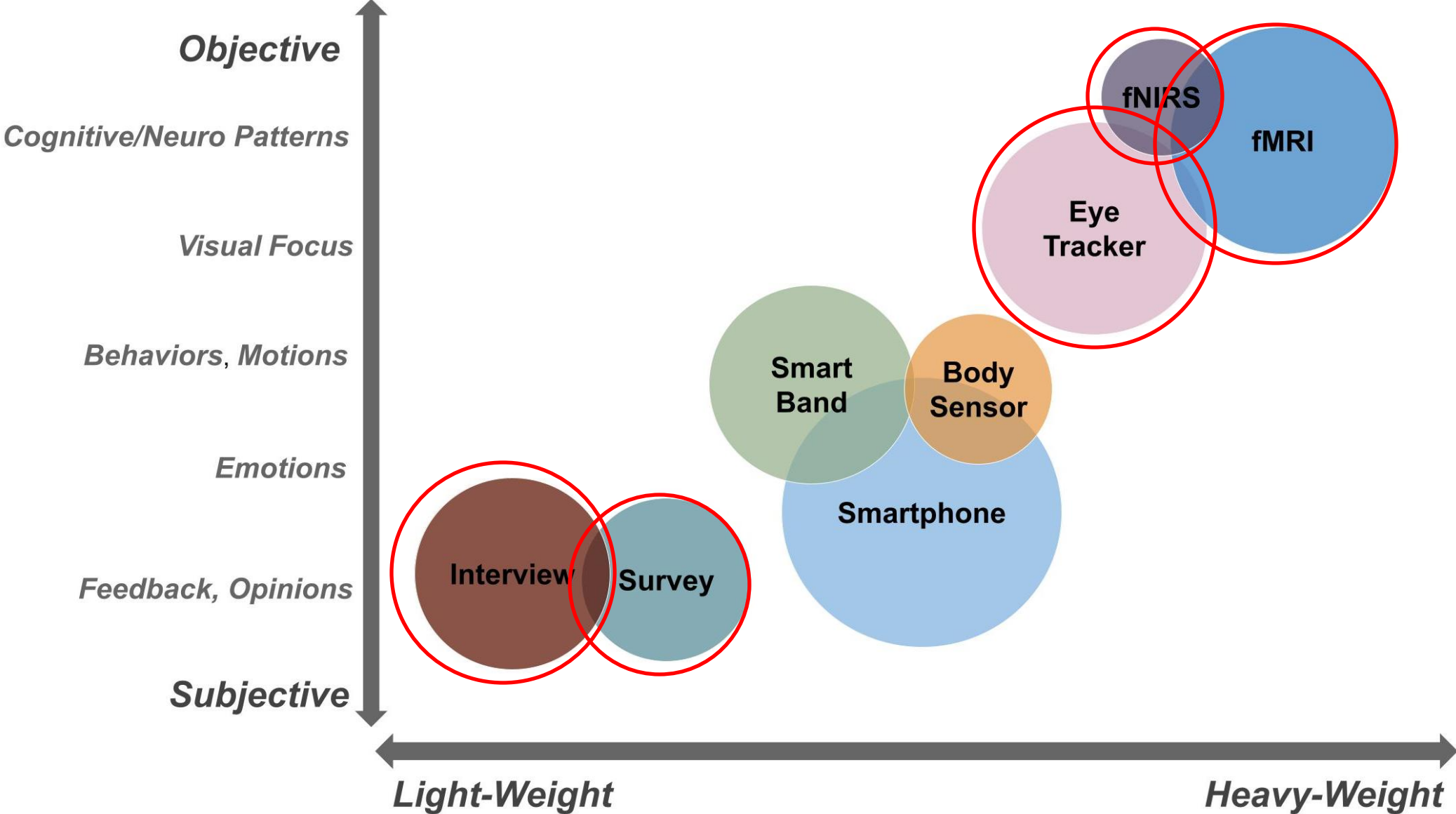
But Mt. Gox's woes didn't end there. In 2014, they lost more than 850,000 bitcoins (valued at roughly half a billion USD at the time) in a hacking incident. Around 200,000 bitcoins were recovered, but the financial loss was still overwhelming and the exchange ended up [declaring bankruptcy](#).

The Human Aspect Matters

- Early study of industrial developers found **order-of-magnitude** individual variations

Metric	Poorest	Best	Ratio
Debugging Hours Algebra	170	6	28:1
Debugging Hours Maze	26	1	26:1
CPU Seconds Algebra	3075	370	8:1
CPU Seconds Maze	541	50	11:1
Code Writing Hours Algebra	111	7	16:1
Code Writing Hours Maze	50	2	25:1
Program Size Algebra	6137	1050	6:1
Program Size Maze	3287	651	5:1
Run Time Algebra	7.9	1.6	5:1
Run Time Maze	8.0	0.6	13:1

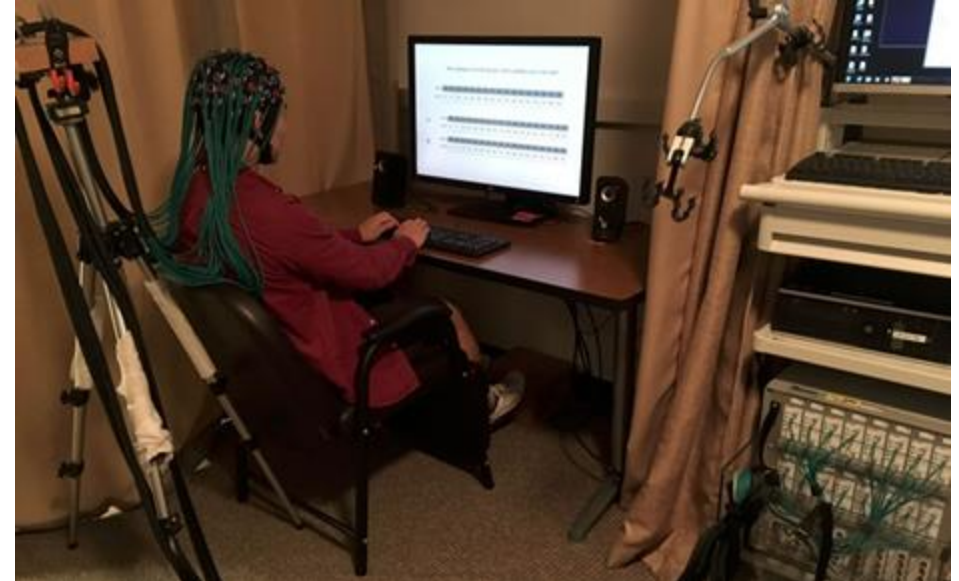
How to measure human aspects?



fMRI vs. fNIRS

Measure brain activities by calculating the **blood-oxygen level dependent (BOLD)** signal

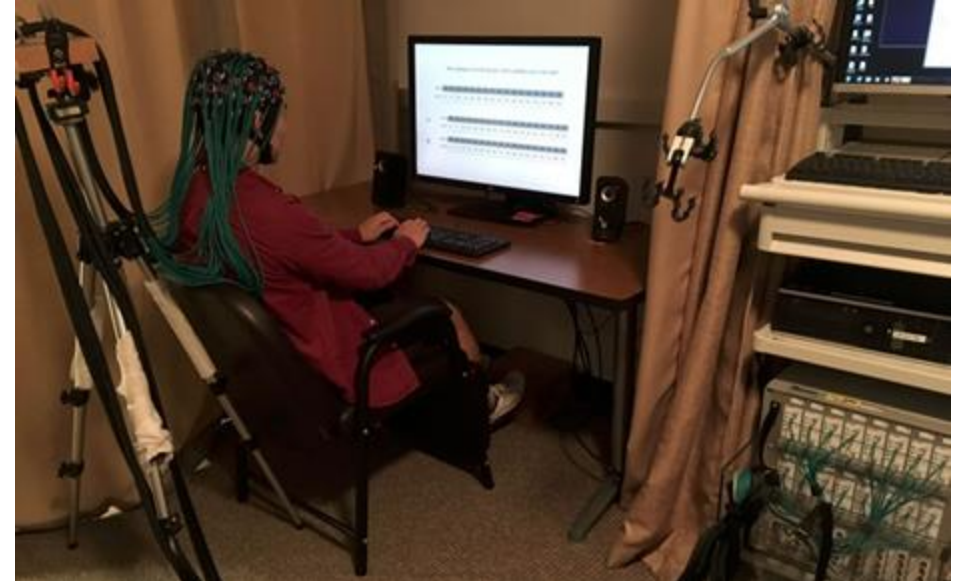
- **F**unctional **M**agnetic **R**esonance **I**maging
 - **Magnets**
 - **Strong** penetration power
 - Lying down in a magnetic tube:
 - **Cannot move**
- **F**unctional **N**ear-**I**nfra**R**ed **S**pectroscopy
 - **Light**
 - **Weak** penetration power
 - Wearing a specially-designed cap:
 - **More freedom of movement**



fMRI vs. fNIRS

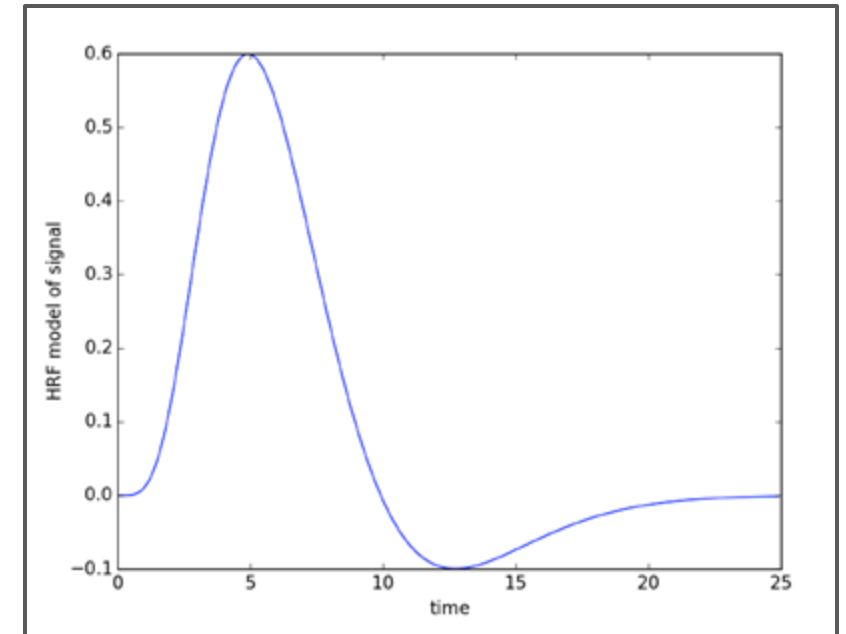
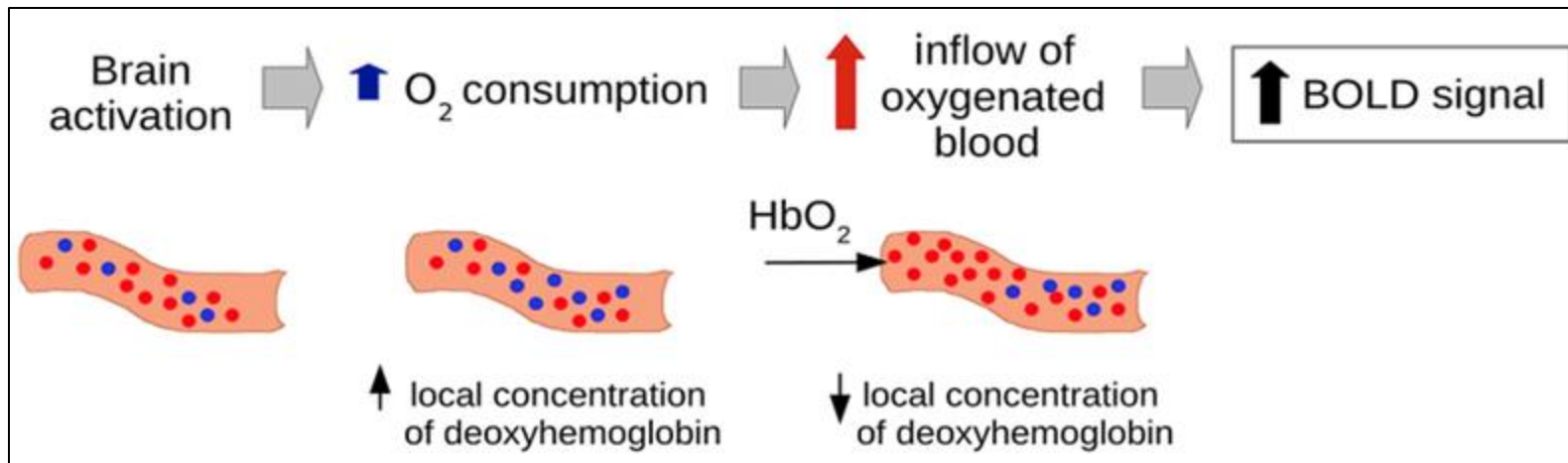
Measure brain activities by calculating the **blood-oxygen level dependent (BOLD)** signal

- **F**unctional **M**agnetic **R**esonance **I**maging
 - **Magnets**
 - **Strong** penetration power
 - Lying down in a magnetic tube:
 - **Cannot move**
- **F**unctional **N**ear-**I**nfra**R**ed **S**pectroscopy
 - **Light**
 - **Weak** penetration power
 - Wearing a specially-designed cap:
 - **More freedom of movement**



What is **BOLD** signal?

- **B**lood-**O**xxygen **L**evel **D**ependent (**BOLD**) signal
- Blood flow and oxygen consumption as a **proxy** for brain activity
- Activation model: hemodynamic response function (HRF)
- Stimulus, HRF, design matrix, noise
 - Comprehensive quantitative model of BOLD signals
 - General Linear Model (GLM)




Think in Terms of Contrasts!

- Controlled experimental design
 - Task A = “balancing trees + nervous + ...”
 - Task B = “rotating 3D objects + nervous + ...”
 - Contrast $A > B$: brain activations that vary between the tasks



Data Analysis

- We need to be **careful**
 - 153,000 voxels or more
 - Spurious correlations due to multiple comparison: false positives



Neural correlates of interspecies perspective taking in the post-mortem Atlantic Salmon: An argument for multiple comparisons correction

Craig M. Bennett¹, Abigail A. Baird², Michael B. Miller¹, and George L. Wolford³

¹ Psychology Department, University of California Santa Barbara, Santa Barbara, CA; ² Department of Psychology, Vassar College, Poughkeepsie, NY; ³ Department of Psychological & Brain Sciences, Dartmouth College, Hanover, NH

INTRODUCTION

With the extreme dimensionality of functional neuroimaging data comes extreme risk for false positives. Across the 130,000 voxels in a typical fMRI volume the probability of a false positive is almost certain. Correction for multiple comparisons should be completed with these datasets, but is often ignored by investigators. To illustrate the magnitude of the problem we carried out a real experiment that demonstrates the danger of not correcting for chance properly.

METHODS

Subject. One mature Atlantic Salmon (*Salmo salar*) participated in the fMRI study. The salmon was approximately 18 inches long, weighed 3.8 lbs, and was not alive at

GLM RESULTS



The figure displays two coronal fMRI scans of a salmon's brain. A color scale on the right indicates t-values, ranging from 2.5 (dark red) to 4.5 (yellow). Two distinct red spots are visible in the brain region of both scans, indicating areas of significant activation.

Data Analysis



- False discovery rate (FDR) correction ($q < 0.05$)

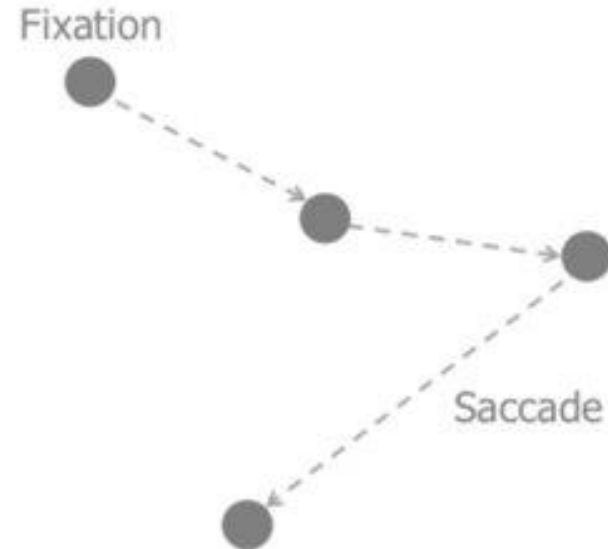
Eye-tracking

- Collect participants' visual attention by recording **eye-gaze** data: what are you looking at? How do you look at it?



Eye-tracking: how we "look"

- Fixation: a spatially stable eye-gaze that lasts for approximately 100-300ms
 - Most of the information acquisition and processing occur during fixations
 - Only a small set of fixations is necessary to process a complex visual stimulus
- Saccade: continuous and extremely rapid eye movements, within 40-50ms, that occur between fixations
- Pupil size
 - Dilation is associated with cognitive work load

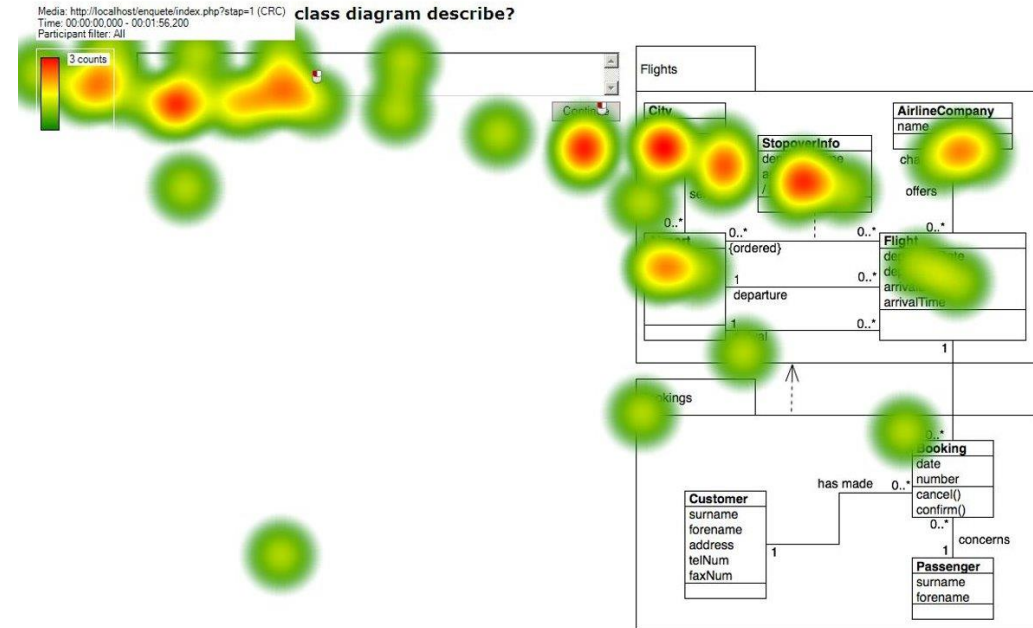
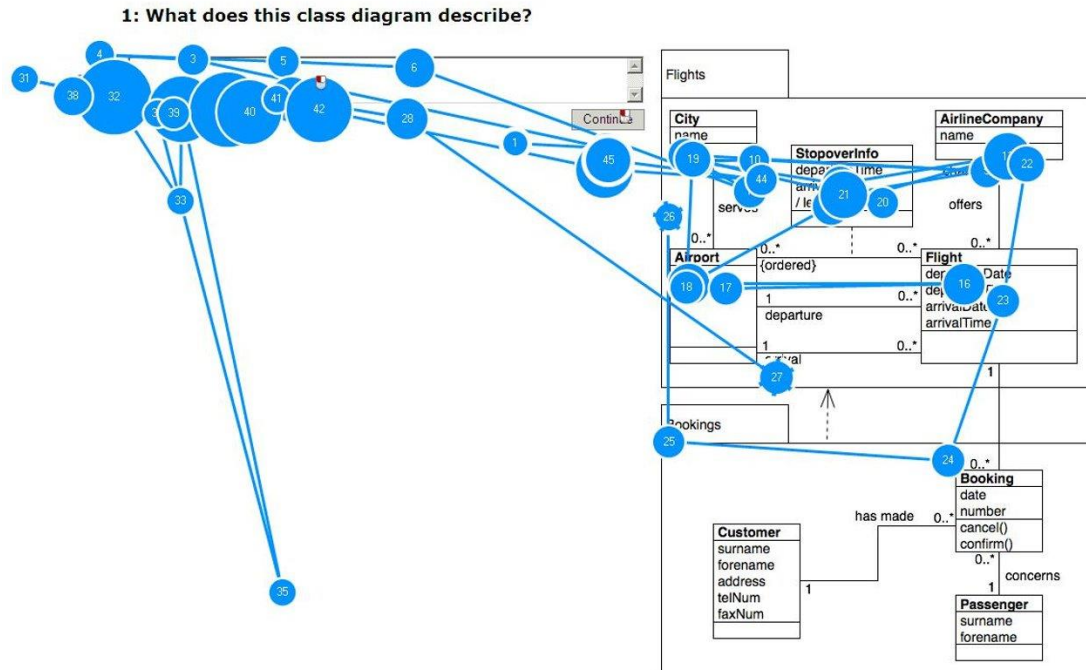


Eye-tracking: assumptions

- The immediacy assumption (Just and Carpenter, 1980):
 - The comprehension begins as soon as a participant sees a stimulus, e.g., as soon as a reader reads a word
- The eye-mind assumption:
 - The participant fixates her attention on a part of the stimulus until she understands that part

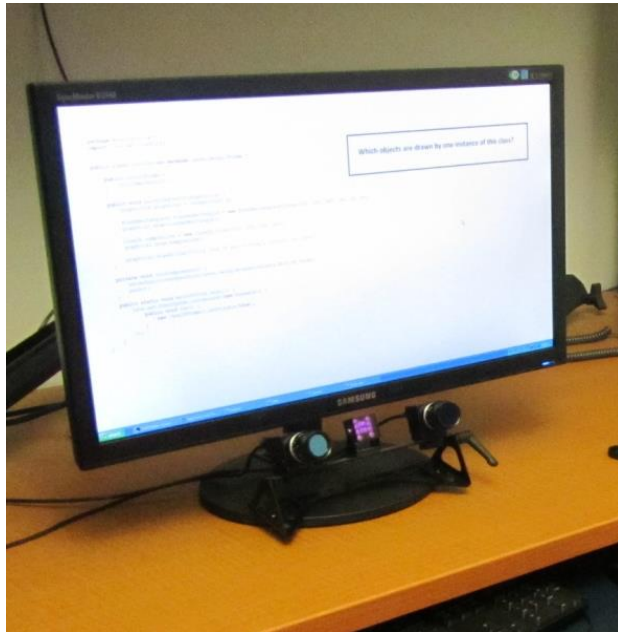


Eye-tracking: gaze plot, heat map, and raw data



Slide	Code	Number	x	y	Pupil X	Pupil Y	Time		
Slide	Code	Number	x	y	Pupil X	Pupil Y	Start Time	End Time	Duration
C:\diagrams\1-M.png	K Space U	3	0	0	0.000000	0.000000	0.000	0.000	0.000
C:\diagrams\1-M.png	G	1	751	1063	39.000000	38.000000	0.297		
C:\diagrams\1-M.png	G	2	688	918	39.000000	38.000000	0.314		
C:\diagrams\1-M.png	G	3	688	918	39.000000	38.000000	0.331		
C:\diagrams\1-M.png	G	4	688	918	39.000000	38.000000	0.347		
C:\diagrams\1-M.png	G	5	684	911	39.000000	38.000000	0.364		
C:\diagrams\1-M.png	G	6	683	906	39.000000	38.000000	0.381		
C:\diagrams\1-M.png	G	7	683	906	39.000000	38.000000	0.397		
C:\diagrams\1-M.png	G	8	683	906	39.000000	38.000000	0.414		
C:\diagrams\1-M.png	G	9	681	900	39.000000	38.000000	0.431		
C:\diagrams\1-M.png	G	10	678	892	38.000000	38.000000	0.447		

Eye-tracking: eye trackers

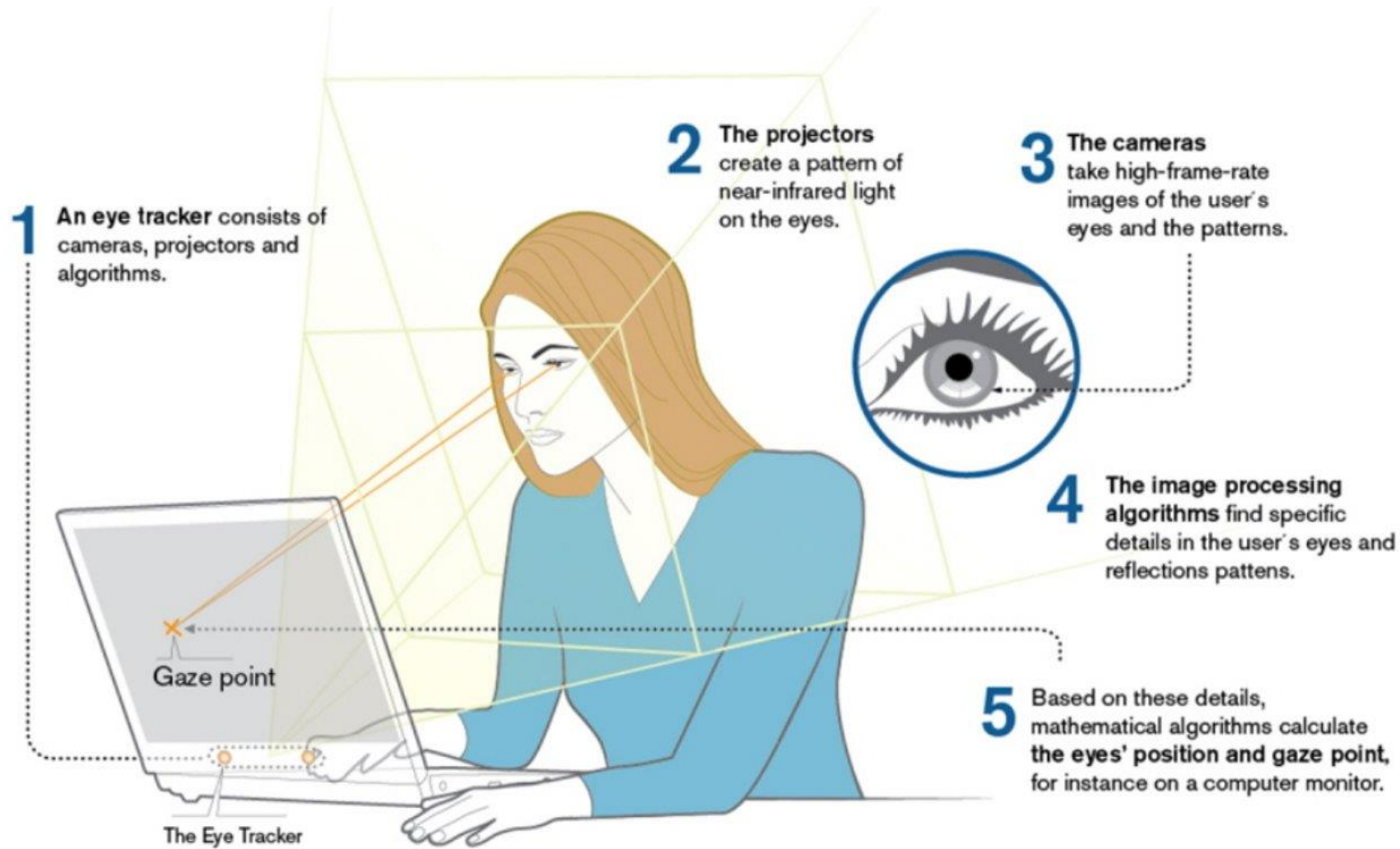


<https://www.tobii.com/>



<https://www.tobii.com/>

Eye-tracking: how does an eye tracker work?



Eye-tracking: truth?

- Eye tracking allows you to know what people are thinking

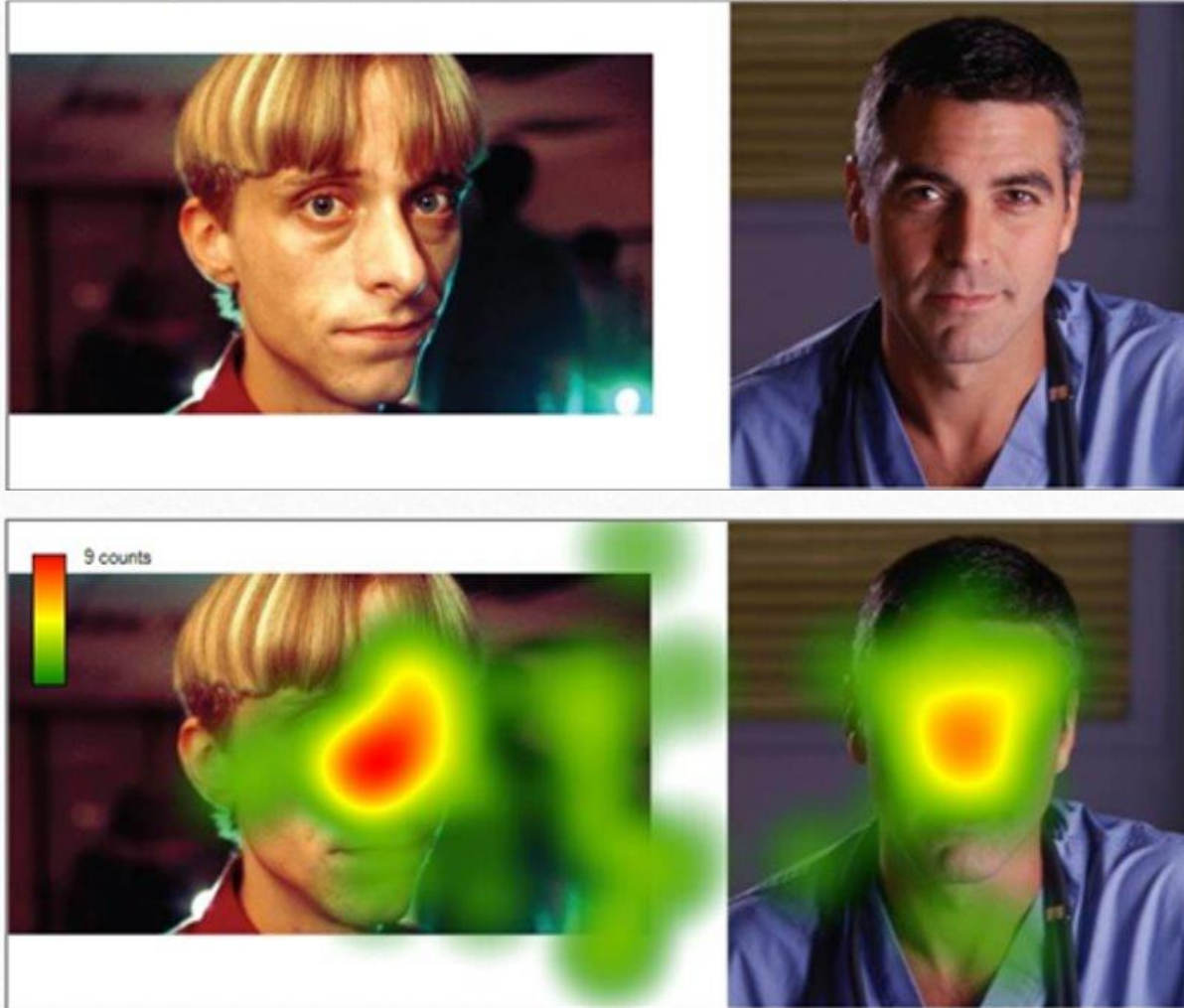
Clooney or Crook: which one do people prefer?



Eye-tracking: truth?

- Eye tracking allows you to know what people are thinking

Clooney or Crook: which one do people prefer?



Eye-tracking: truth?

Misconception

~~Truth~~ about eye tracking

- Eye tracking allows you to know what people are thinking

Fact: Eye tracking will give you evidence of
what people look at
Not what they **think, understand, or like**



Eye-tracking: truth?

Misconception

~~Truth~~ about eye tracking

- Eye tracking allows you to know what people are thinking

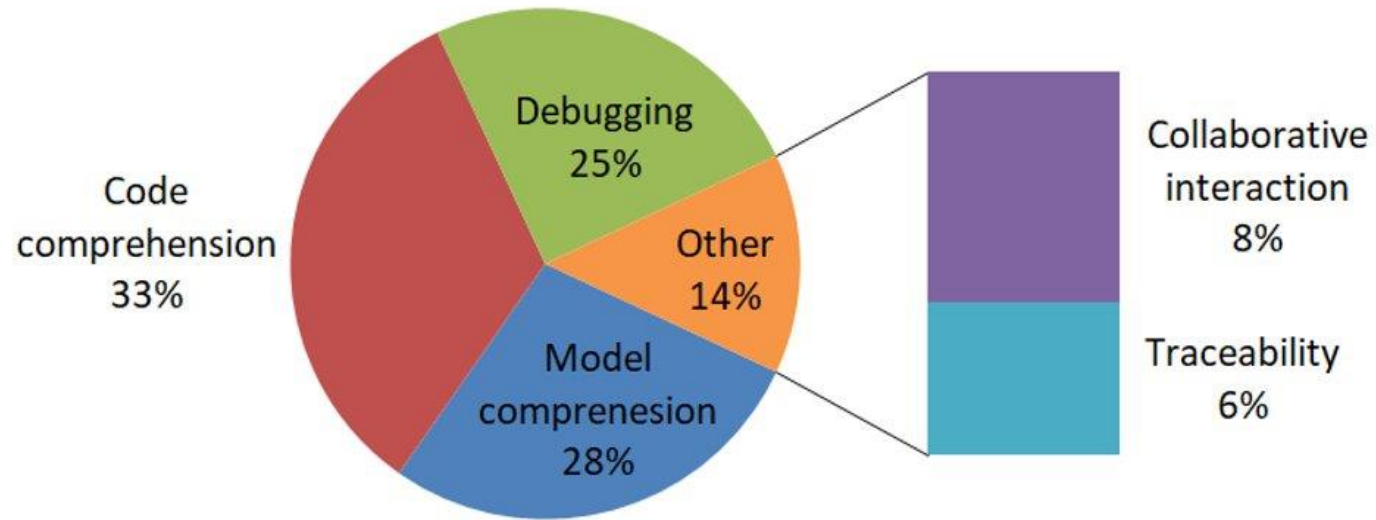
Fact: Eye tracking will give you evidence of
what people look at
Not what they **think, understand, or like**



- Combination:
 - Medical imaging
 - Surveys, interviews

Eye-tracking: for software engineering

Classification of SE eye tracking papers based on category (2015)



Code					Model				English text	Other
Pascal	C/C++	Java	C#	Python	UML	ER	Tropos	BPMN		
2	3	16	1	1	7	1	1	1	2	3 applications

Eye-tracking: for software engineering

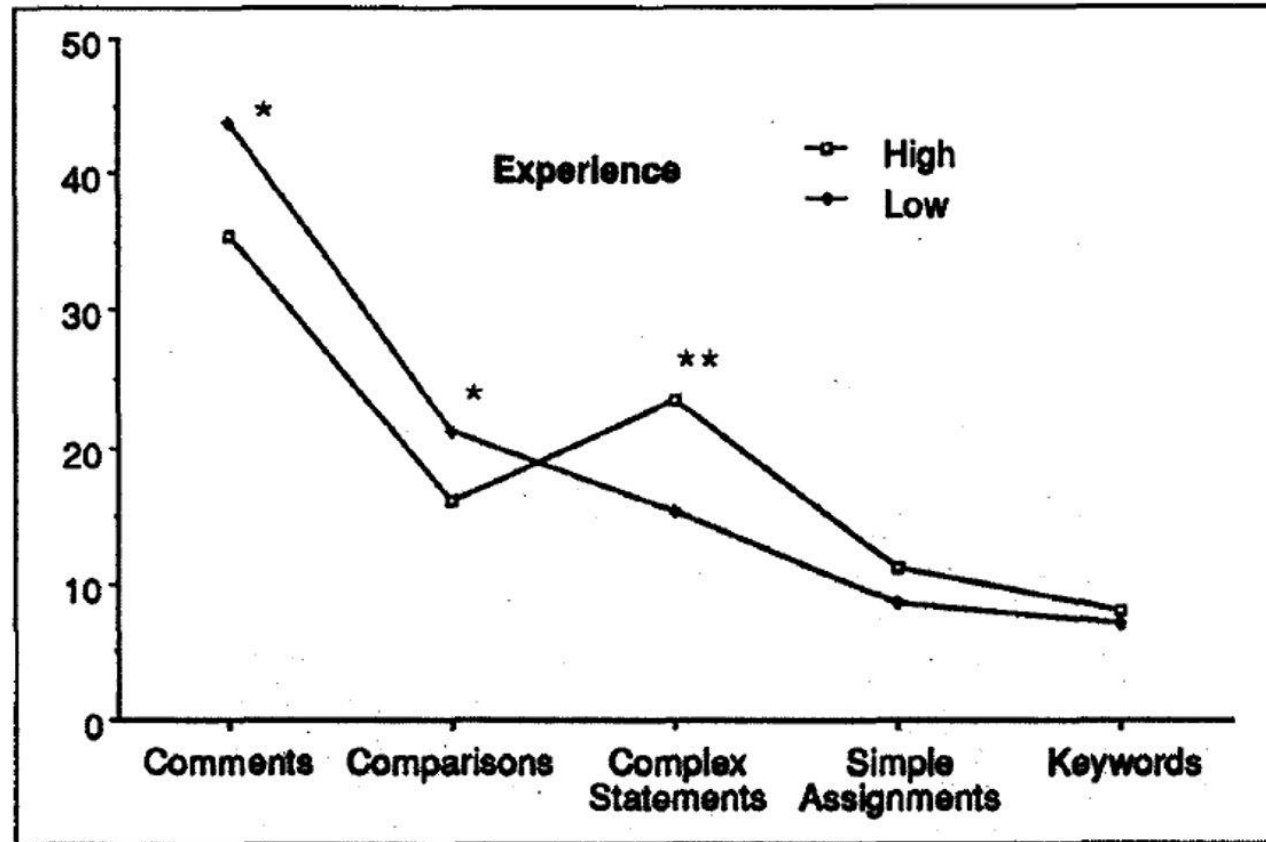
Types of SE questions in eye tracking experiments

Category	Type of Questions
Finding the Areas of Interest	<p>What items or what parts of artifact (X), do participants view while performing task (Y)?</p> <p>Example: Does experience influence a participants focus on critical areas of the algorithm? (Crosby and Stelovsky, 1990)</p>
Navigation Strategies	<p>How do participants navigate through artifact/system (X) while performing task (Y)?</p> <p>Does the type of artifact (X) impact the participants' navigation strategies while they perform task (Y)?</p> <p>Do the participants' individual characteristics (Z) impact their strategies while they perform task (Y)?</p> <p>Example: Do the viewing patterns of experienced participants differ from those of novices?</p>

Eye-tracking: for software engineering

Martha Crosby 1990

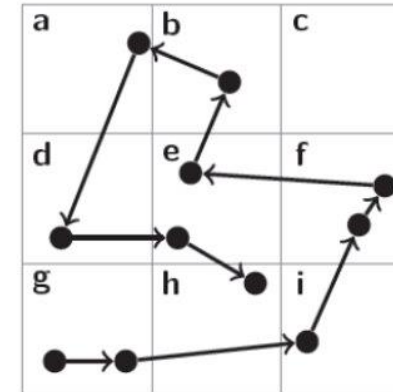
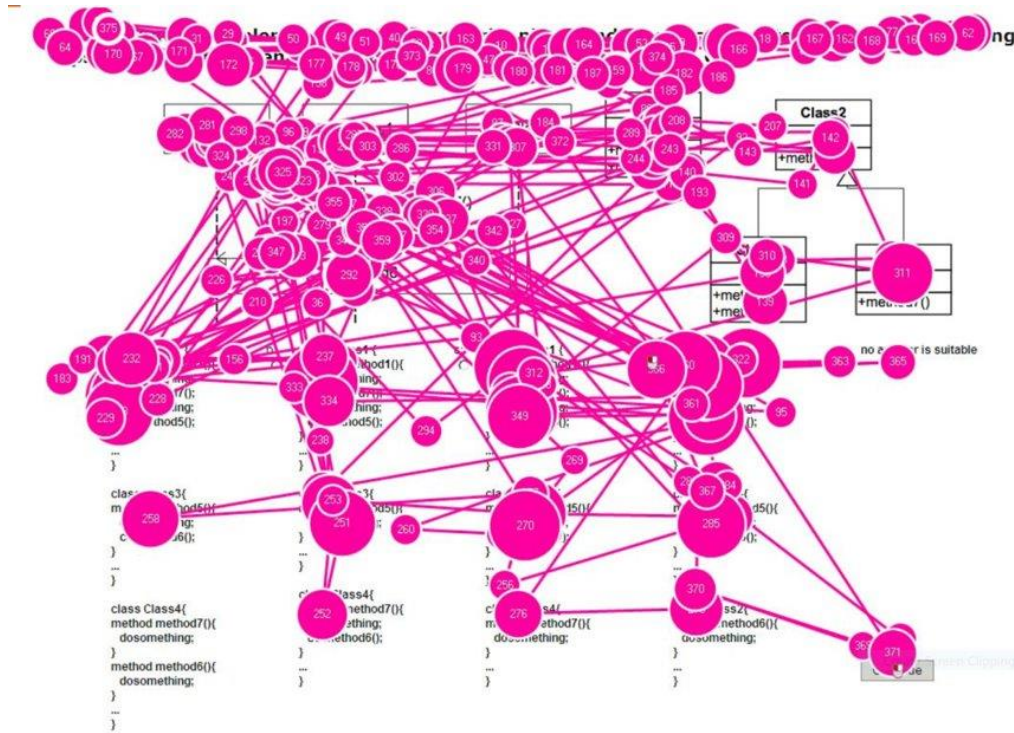
Algorithm areas viewed: novices vs. experts



Eye-tracking: for software engineering

Scan path analysis

- A series of fixations or visited AOIs (Area of Interest) in chronological order.



Eye-tracking: for software engineering

Recent work:

- combined with other measures, e.g., medical imaging
- Investigate human biases in SE activities: e.g., gender, social info



Biases and Differences in Code Review using Medical Imaging and Eye-Tracking: Genders, Humans, and Machines

Yu Huang
Univ. of Michigan
Ann Arbor, MI, USA
yhhy@umich.edu

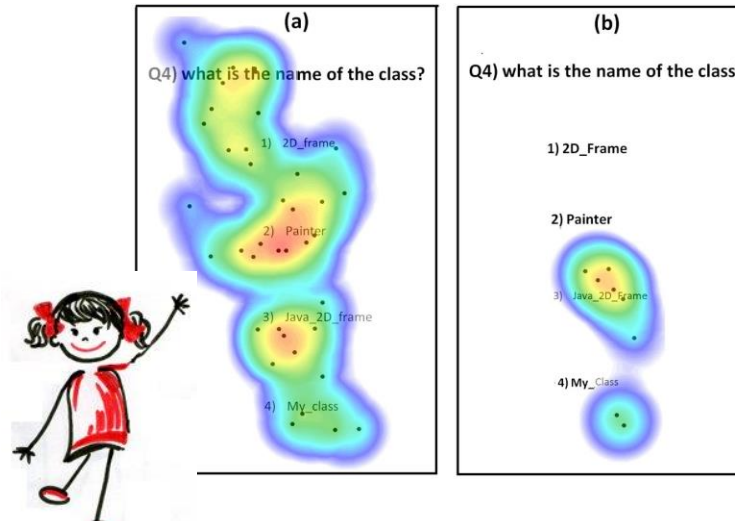
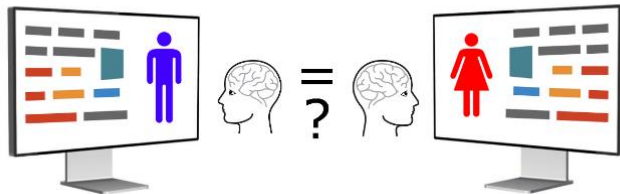
Kevin Leach
Univ. of Michigan
Ann Arbor, MI, USA
kjleach@umich.edu

Zohreh Sharafi
Univ. of Michigan
Ann Arbor, MI, USA
zohrehsh@umich.edu

Nicholas McKay
Univ. of Michigan
Ann Arbor, MI, USA
njmckay@umich.edu

Tyler Santander
Univ. of California, Santa Barbara
Santa Barbara, CA, USA
t.santander@psych.ucsb.edu

Westley Weimer
Univ. of Michigan
Ann Arbor, MI, USA
weimerw@umich.edu



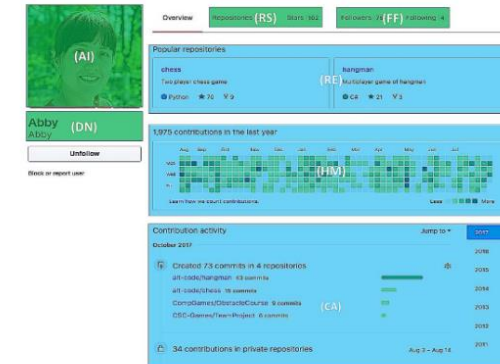
Beyond the Code Itself:

How Programmers Really Look at Pull Requests

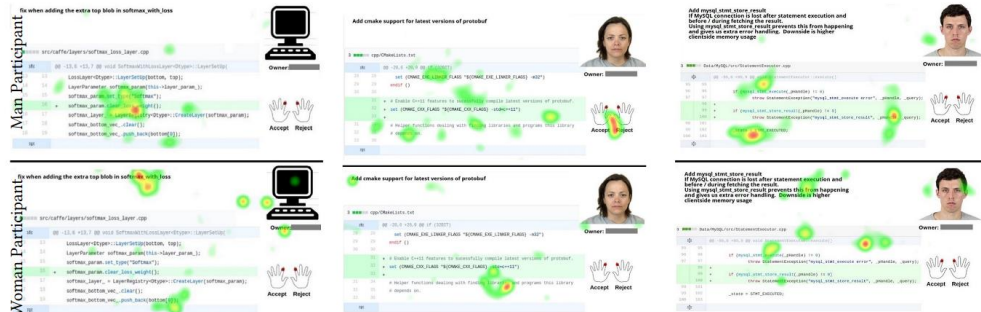
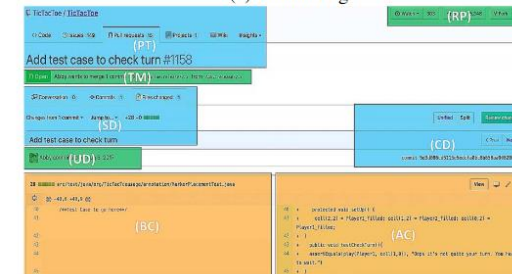
Denae Ford, Mahnaz Behroozi
North Carolina State University
Raleigh, NC, USA
{dford3, mbehroo}@ncsu.edu

Alexander Serebrenik
Eindhoven University of Technology
Eindhoven, The Netherlands
a.serebrenik@tue.nl

Chris Parnin
North Carolina State University
Raleigh, NC, USA
cjparnin@ncsu.edu



(a) Profile Page

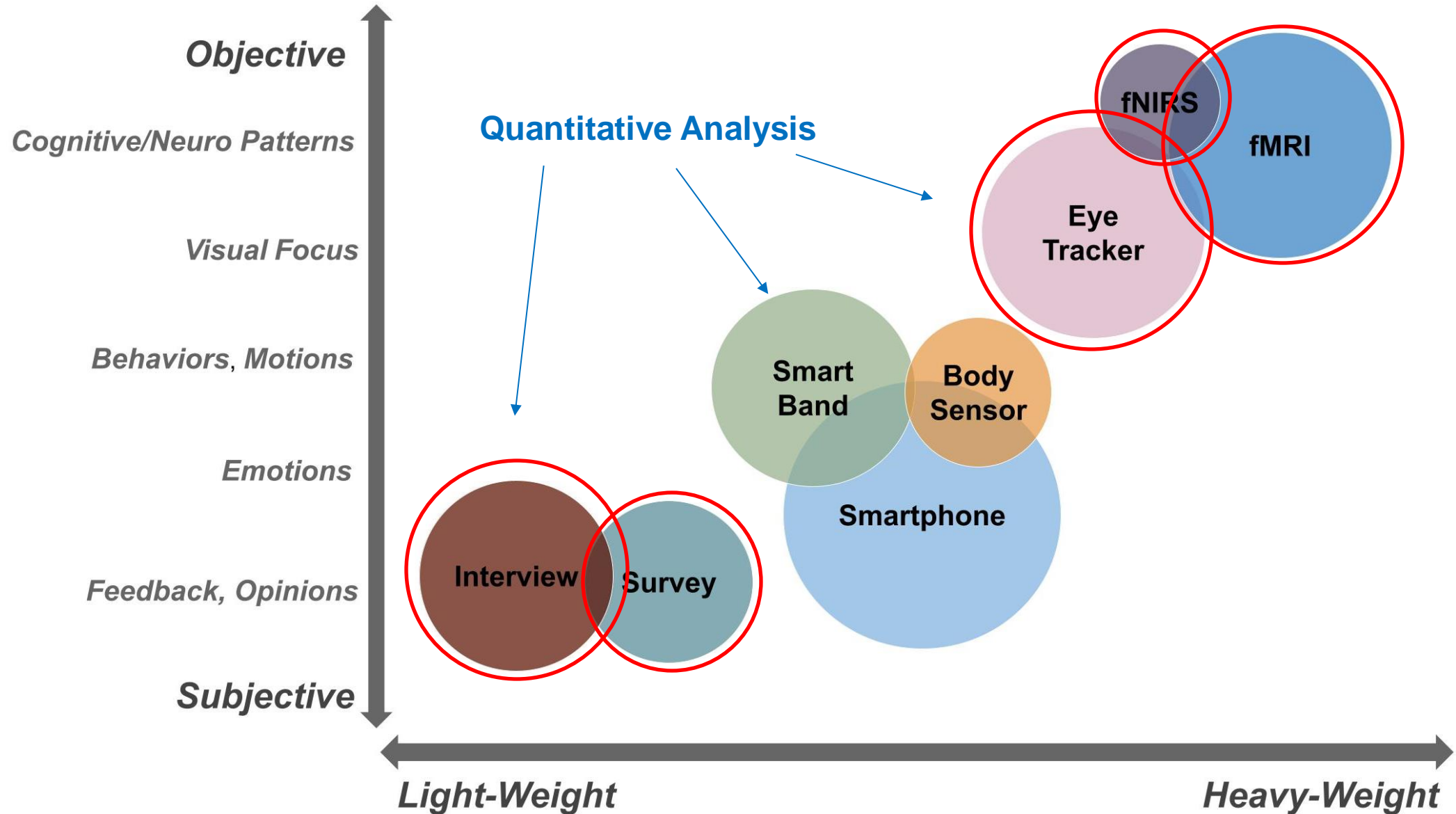


(a) A stimulus with a machine author

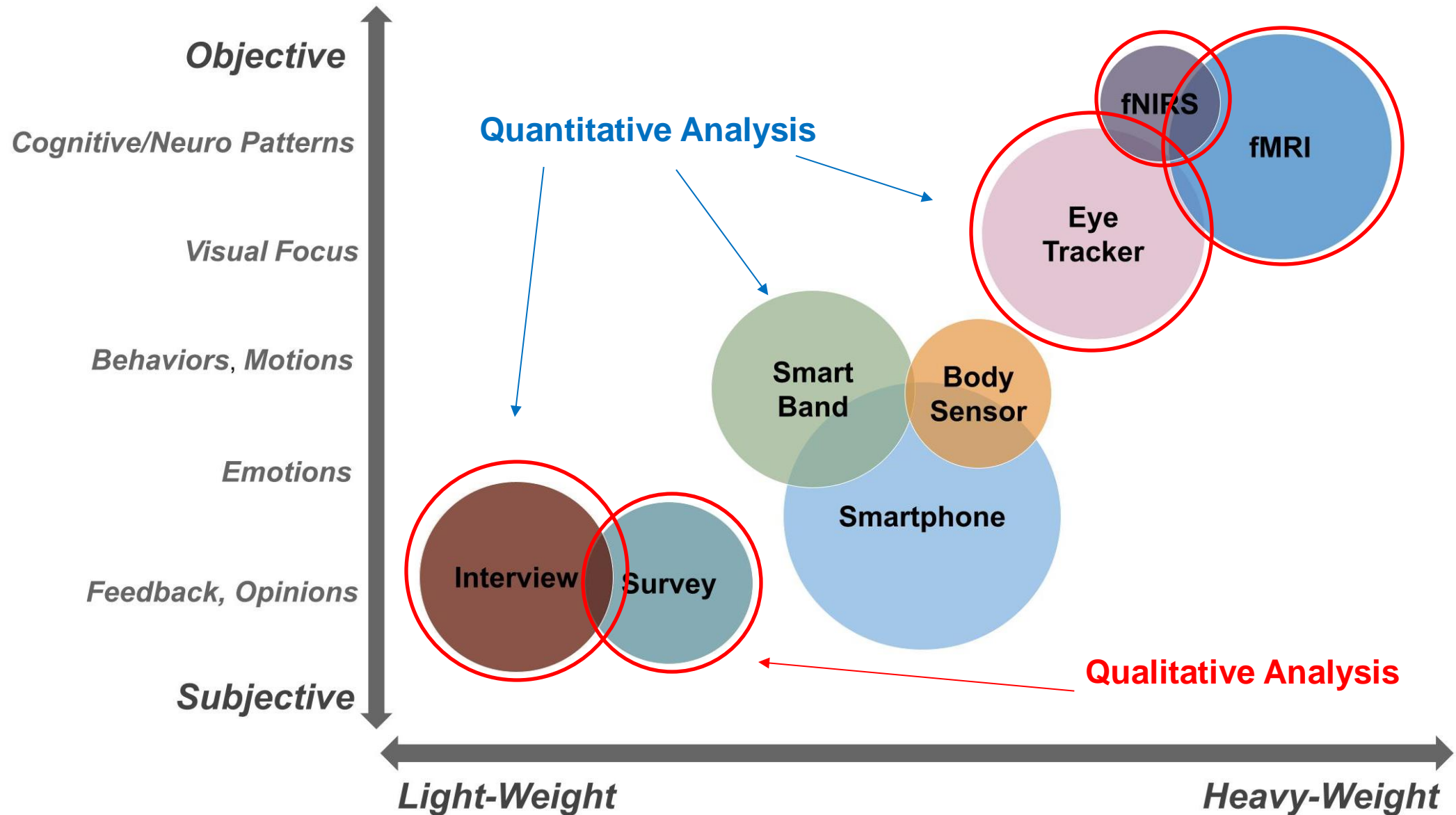
(b) A stimulus with a woman author

(c) A stimulus with a man author

How to analyze human aspects?



How to analyze human aspects?



How to analyze human aspects: qualitative analysis

- Verbally-acquired data
 - Information that is gathered via speech, think-aloud protocol, oral retrospection, formal or informal interviews and surveys

With appropriate care in data gathering and analysis, **verbal data *can* provide impactful insights in software engineering research.**

How to analyze human aspects: qualitative analysis

- Verbally-acquired data
 - Information that is gathered via speech, think-aloud protocol, oral retrospection, formal or informal interviews and surveys
- Classic example: the "Sillito et al." Questions, published in FSE '06, cited over 350 times

them. Participants in the second study (E1...E16) were observed working on code with which they had experience. In both studies

During each session an audio recording was made of discussion between the pair of participants, a video of the screen was captured,

To structure our data collection and the analysis of our results, we have used a *grounded theory* approach which has been described as an emergent process intended to support the production of a theory that "fits" or "works" to explain a situation of interest [5, 19]. In

Questions Programmers Ask During Software Evolution Tasks

Jonathan Sillito, Gail C. Murphy and Kris De Volder
Department of Computer Science
University of British Columbia
Vancouver, B.C. Canada
{sillito,murphy,kdvolder}@cs.ubc.ca

about the source code on which we observed them working. We report on 44 kinds of questions we observed our participants asking. These questions are generalized versions of the specific ques-

Results are useful directly (a structured answer to a fundamental question) and also as artifacts (re-used by later projects as indicative developer queries)

Qualitative Analysis: Metrics

- Establishing **validity** in qualitative research
 - Using multiple validity procedures
 - Member checking
 - Clarify bias
 - Spend prolonged time in the field
 - Using qualitative reliability
 - Document your procedures (scripts, codebook, etc.)
 - No drift in the definition of **codes**
 - Cross-check codes developed by different researchers



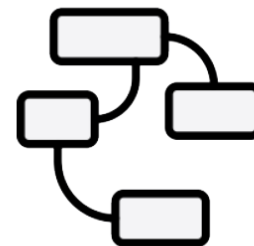
Showing Prompts



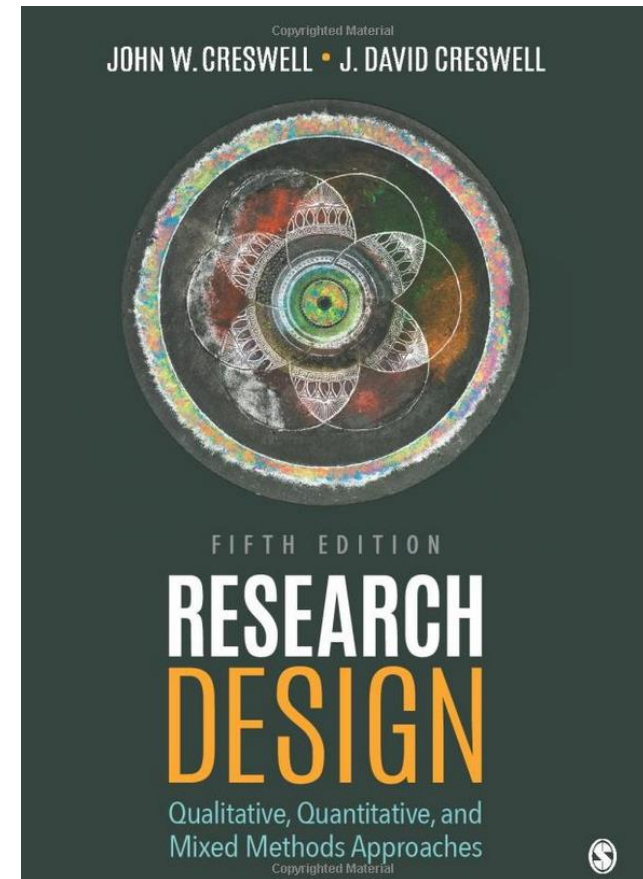
Audio Recording



Transcribing



Qualitative analysis



Qualitative Analysis: Useful Techniques

- Grounded theory in SE
 - Similar to socio-technical studies, qualitative research can have a lot of variance
 - How can we mitigate that variance?
- Grounded Theory is a systematic methodology for qualitative research for constructing hypotheses via inductive (not deductive) reasoning
 - Method
 - Empirical/evidence based
 - Outcome
 - Key patterns of the data
 - Relationships between patterns

“It is not in your mind; it is in your data.”

Qualitative Analysis: Useful Techniques

- Grounded theory in SE
- Inductive Thematic Analysis
 - Thematic exploration (thematic coding)
 - Codes and the relationships

Category	Code	Description
motivation	motivation-helpuser	help end users
	motivation-helpdev	help developers
	motivation-longterm	how to keep yourself engaged in the project for a long time
	motivation-giveback	altruism
	motivation-impact	want to make impact
	motivation-better-programmer	want to look good in the community, improving skills, build up portofolio
	mitivation-hobby	I feel happy/fun, e.g., as a hobby.
	motivation-work	This is my job, or school projects, etc

Codebook Example

Leaving My Fingerprints: Motivations and Challenges of Contributing to OSS for Social Good

Yu Huang
University of Michigan
Ann Arbor, MI
yhhy@umich.edu

Denae Ford
Microsoft Research
Redmond, WA USA
denae@microsoft.com

Thomas Zimmermann
Microsoft Research
Redmond, WA USA
tzimmer@microsoft.com



TABLE II: Themes of Motivations for Contributing to OSS for Social Good.

Theme	Description	Representative Example	Participants
To help those in need	Contributors wanted to help people who are in need but may lack the capability of solving the problems themselves.	<i>"I'm so much more motivated to build products that I know have a good outcome for a group of people that is generally underserved."</i>	P2, P3, P4, P5, P6, P7, P8, P9, P10, P12, P14, P18, P19
To become a better programmer	Contributors wanted to improve their skills, build up their portfolios, or improve their reputation in the community.	<i>"when I contribute to that, it can definitely give me more experience."</i>	P2, P3, P5, P10, P11, P12, P14, P16, P17, P20
To have an impact on society	Contributors wanted to make a difference to the society.	<i>"So, I think the main reason is because I want to make a difference on my life... make a fingerprint on the world."</i>	P1, P3, P4, P7, P13, P14, P15, P17
For emotional fulfillment	Contributors were motivated by feeling good about the impacts of the project.	<i>"It gives a mental satisfaction that I'm working towards something good"</i>	P3, P4, P10, P11, P12, P17, P20
To help fellow developers with their project	Contributors want to help the developers to achieve the accomplishment of the projects.	<i>"Another is to help the people in the project to help reach their goals."</i>	P3, P7, P10, P12, P13, P18
To give back as I received	Contributors want to give back to the society (e.g., altruism).	<i>"And I also feel like however much you take from something, you should give back."</i>	P4, P5, P9, P16, P20
To meet like-minded people	Contributors wanted to get to know more people.	<i>"I think it brings like-minded people together most of the time, so I get to interact with people who are working on similar project or they have similar interests."</i>	P11, P13, P17
As a hobby	Contributors worked in OSS4SG as a hobby or something they like doing.	<i>"I've moved to sales but still collaborating ... It's just as a hobby."</i>	P14, P15
Because I need it for work	Contributors worked on OSS4SG for their professional work projects.	<i>"So the direct cause that I found it is through [elided]'s little competition."</i>	P2

Qualitative Analysis: Useful Techniques

- Grounded theory in SE
- Inductive Thematic Analysis
 - Thematic exploration
 - Codes and the relationships
 - Evaluation metrics
 - Saturation
 - Agreement

Qualitative Analysis: Useful Techniques

- Grounded theory in SE
- Inductive Thematic Analysis
 - Thematic exploration
 - Codes and the relationships
 - Evaluation metrics
 - Saturation
 - Agreement
- Inter Rater Reliability (IRR) or Inter Rater Agreement (IRA)
 - Statistics as evidence
 - Cohen's kappa, Fleiss' kappa, etc.

“It is not in your mind; it is in your data.”

Qualitative Analysis: Combining Verbal and Nonverbal Data

- Strength of verbal data
 - Richness and holism
 - Discovery
 - New ideas, hypothesis
- Weakness of verbal data
 - Hard to evaluate the analysis (i.e., no “equations”)
 - Human biases
- Combining verbal and nonverbal data makes a strong and interesting case
 - Supplement, validate, or illuminate each other
- Contrast: surprising knowledge!

Qualitative Analysis: Combining Verbal and Nonverbal Data

• What do you think about pull requests generated by machines

• "Machine generated code is worse on readability!"

But all pull requests were written by humans! (We deceived you!)

• Do you think women and men write pull request differently

• "There is no difference between pull requests written by men and women"

But there *is* a significant difference on your behavior! Both response time and final decisions are affected!

Biases and Differences in Code Review using Medical Imaging and Eye-Tracking: Genders, Humans, and Machines

Yu Huang
Univ. of Michigan
Ann Arbor, MI, USA
yhhy@umich.edu

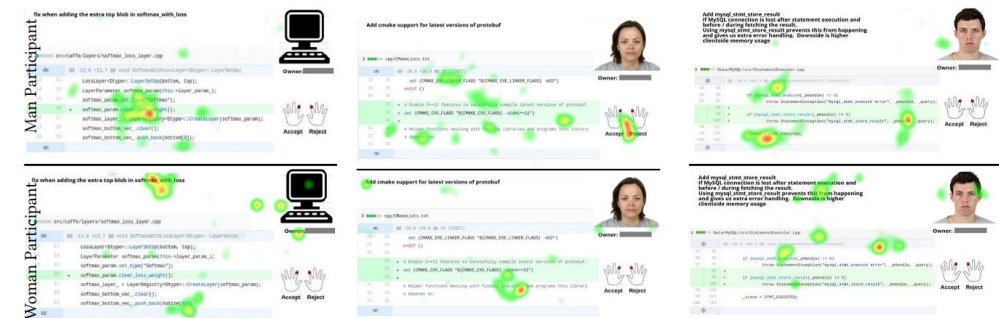
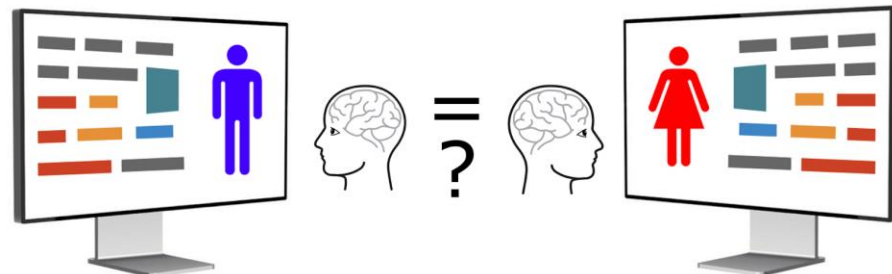
Kevin Leach
Univ. of Michigan
Ann Arbor, MI, USA
kjleach@umich.edu

Zohreh Sharafi
Univ. of Michigan
Ann Arbor, MI, USA
zohrehsh@umich.edu

Nicholas McKay
Univ. of Michigan
Ann Arbor, MI, USA
njmckay@umich.edu

Tyler Santander
Univ. of California, Santa Barbara
Santa Barbara, CA, USA
t.santander@psych.ucsb.edu

Westley Weimer
Univ. of Michigan
Ann Arbor, MI, USA
weimerw@umich.edu



(a) A stimulus with a machine author

(b) A stimulus with a woman author

(c) A stimulus with a man author

Statistics: A Brief Overview

- Important but used to be overlooked in CS research
 - "The proposed system achieves a 10% higher accuracy on average compared to X in 10 runs..."
- Statistical tests
 - Is it significant?

showed notable resistance to this decline. For example, the equiprobable heuristic chose the optimal alternative almost six times as often as one would expect by change in the 8×2 decision situation. And five of the heuristics—E, Min, MR, ML, and P—found one of the highest two expected value alternatives over 80% of the time in the 8×2 decision situations. The propensity to avoid the alternatives with lowest EV decreased to well below chance for all heuristics as the number of alternatives increased. Indeed, only three heuristics

Why statistics for this class?

- A number of papers use statistical techniques, and understanding something about them will be useful.
- You may also need to run statistical tests as part of your research projects.
- Examples:
 - Is there a difference in gaze times on identifiers in Gerrit vs. GitHub?
 - Is there a relationship between how much you pay someone and how fast they complete a programming task?

Why statistics at all?

- Descriptive statistics
 - Describe or summarize the data
 - Example: What *usually* happens?
 - Mean
 - Median
- Inferential statistics
 - Intuition: Can we be confident the data is telling us the story we think it is, or did we just get lucky?
 - Does the data we have represent the data we don't have?

Some technical terms

- Population = the items you're interested in, e.g. all developers
- Sample = the items you're actually looking at, e.g. 10 developers interviewed
- Distribution = the shape of the data on the plot (e.g., normal)

Hypothesis testing

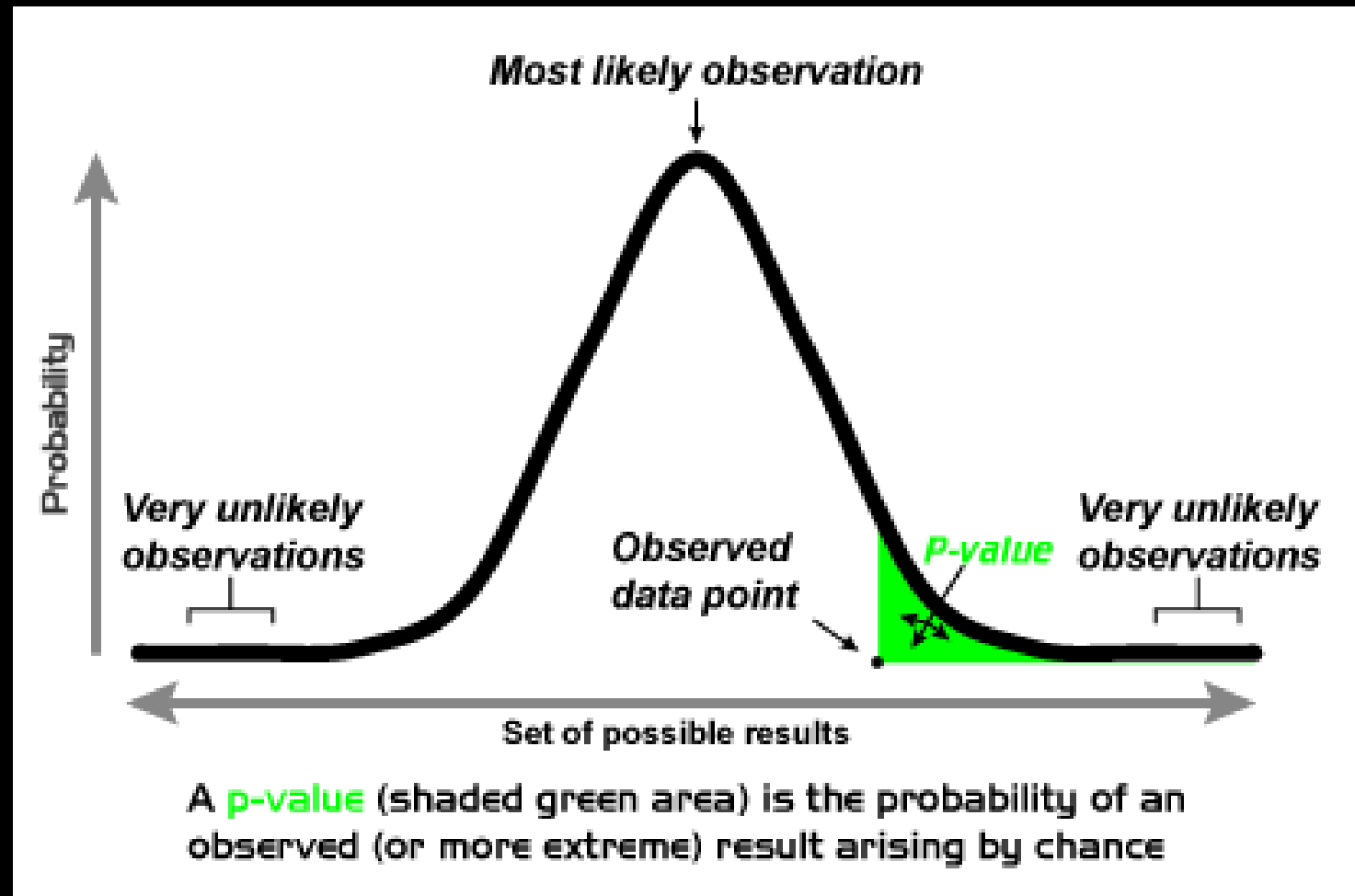
- Inferential statistics can be run when you state your research problem as hypothesis, specifically using:
 - Null hypothesis (H_0) = no difference or no relationship
 - Alternative hypothesis (H_1) = a difference or relationship exists
- Example 1:
 - Null: Teams with high IQ member perform equally well as teams with high social intelligence member
 - Alternative: Two teams perform differently
- Example 2:
 - Null: No relationship exists between how much we pay someone and how quickly they complete a programming puzzle
 - Alternative: The more we pay, the faster or slower someone completes the task

p-value: "statistically significant"

- A probability, between 0 and 1.
- Definition:
 - Technical: Assuming that the null hypothesis is true, the probability of obtaining a result this extreme or more extreme
 - Intuitive: Probability that we got this result by chance
- Use
 - We define an alpha level, below which we consider the result to be “statistically significant”. Conventionally (but for no particularly good reason) $\alpha=.05$
 - If a difference or relationship appears to exist, but is not significant, we probably should not say that there is a difference at all
- What it's not:
 - ~~The probability of H_0 or H_1 being true or false~~

p-value

- A probability, between 0 and 1
- Definition:
 - Technical: As the area under the tail of a distribution that is extreme or more extreme than the observed data point
 - Intuitive: Probability of observing a result as extreme as the observed data point, assuming the null hypothesis is true
- Use
 - We define an alpha level (conventionally 0.05) as the maximum acceptable probability of a Type I error
 - If a difference is statistically significant, that there is a difference between the groups
- What it's not:
 - The probability of a Type II error



Statistical power

1. If you have an IBM developer who's 2x more productive than a Google developer, do we believe that IBM developers are more productive than Google developers?
2. What if we have 1000 IBM developers who are, on average 2x more productive than 1000 Google developers?
 - Are we equally or more likely to believe (1) or (2)?
 - The second situation has more **statistical power**, that is, the ability to detect a real effect
 - The following affects statistical power
 - Sample size
 - Effect size: *a quantitative measure that tells you how meaningful the relationship between variables or the difference between groups is.*
 - Indicates **practical significance**: if the effect is **large enough to be meaningful** in the real world.
 - Compared to: **statistical significance (p-value)**, "*an effect exists*"
 - Statistical test (t-test, chi-square, etc.)

Confidence Interval

- A range of values for which you're confident the "true" value lies
- You determine the confidence intervals, usually set at 90%, 95%, or 99%
- Similar to p-value, but integrates effect size, so more informative
- Given as $x \pm \text{value}$

Example

- Average pulse rate = 101 bpm; Standard Deviation = 50; N = 200

- 95% Confidence Interval = (94, 108)

We are 95% confident that the true pulse rate for our population is between 94 and 108.

Margin of error = $(108 - 94) / 2 = \pm 7$ bpm

- Example: We are 95% confident that the true pulse rate for our population is between 94 and 108
- Question:
 - Does more data increase or decrease your confidence interval?

Confidence Interval

- A range of values for which you're confident the "true" value lies
- You determine the confidence intervals, usually set at 90%, 95%, or 99%
- Similar to p-value, but integrates effect size, so more informative
- Given as $x \pm$ value
- Question:
 - Does more data increase or decrease your confidence interval?
 - A larger sample size or lower variability will result in a tighter confidence interval with a smaller margin of error.
 - A smaller sample size or a higher variability will result in a wider confidence interval with a larger margin of error.
 - The level of confidence also affects the interval width. If you want a higher level of confidence, that interval will not be as tight. A tight interval at 95% or higher confidence is ideal.

Examples:

- Average Scene Time = 5.5 mins; Standard Deviation = 3 mins; N = 10 runs

- 95% Confidence Interval = (3.6, 7.4)

Margin of Error = ± 1.9 minutes

- Average Scene Time = 5.5 mins; Standard Deviation = 3 mins; N=1,000 runs

- 95% Confidence Interval = (5.4, 5.6)

Margin of Error = ± 0.1 minutes

Two types of statistical tests

Parametric Tests

- Assume a particular distribution of data, typically normal
- Assumes differences between values are meaningful
- More statistical power
- Examples:
 - Student t-test
 - ANOVA
 - Pearson correlation

Non-parametric tests

- Does not assume a distribution
- Ignores differences between values
- Less powerful
- Examples:
 - Chi-square
 - Fisher
 - Wilcoxon and Mann-Whitney
 - Spearman correlation

Student t-test

- "t-test"
 - Commonly used: two-sample t-test
 - test of the null hypothesis such that the means of two populations are equal.
 - Paired vs. unpaired

Student t-test

- "t-test"
 - Commonly used: two-sample t-test
 - test of the null hypothesis such that the means of two populations are equal.
 - Paired vs. unpaired
- History
 - Gets its name from William Sealy Gosset who first published it in 1908 in the scientific journal Biometrika using his pseudonym "Student", because his employer preferred staff to use pen names when publishing scientific papers instead of their real name, so he used the name "Student" to hide his identity
 - Guinness Brewery: Is beer 1 better than beer 2 using different barley? Guinness did not want their competitors to know that they were using the t-test to determine the quality of raw material

Chi-square test

- Similar to t-test
 - Frequency: categorical data
 - determine whether there is a statistically significant difference between the expected frequencies and the observed frequencies in one or more categories

Wilcoxon Signed-Ranks / Rank Sum

- Non-parametric versions of paired and unpaired t-tests
- H_0 : for randomly selected values X and Y from two populations, the probability of X being greater than Y is equal to the probability of Y being greater than X .
- Compares medians, rather than means (so report 'em!)
- Mann-Whitney U-test = Wilcoxon rank-sum

Hypothesis 1.3: Compared to males, females make pull requests that modify fewer lines of code, modify fewer files, and contain fewer commits.

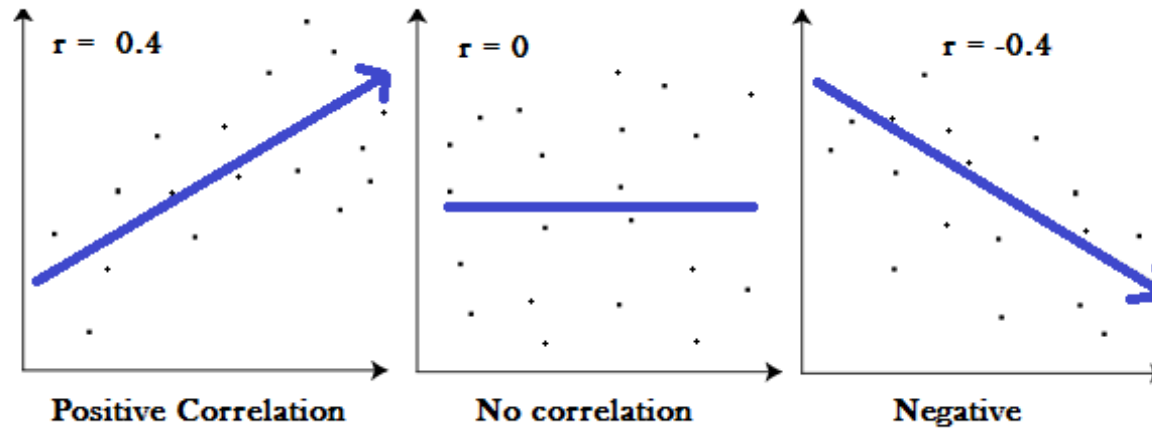
The following table lists the median and mean lines of code added (+), removed (-), files changed, and commits per pull request:

		lines		files	commits
		+	-	changed	
female	median	21	3	2	1
	mean	1640	617	5.4	30.4
male	median	13	2	1	1
	mean	762	299	4.1	24.5

With the exception of lines removed, all differences between females and males are significantly higher (Wilcoxon rank-sum test, $p < .001$). On threat to this analysis is that

Pearson and Spearman Correlations

- Correlation
 - Test: if there is strong association between one variable versus another.
 - Coefficient: 0-1 and p-value




Pearson and Spearman Correlations

- Pearson correlation analysis:
 - Parametric
 - Continuous in nature: each variable is able to take on a potentially infinite number of values
 - The shape of the relationship between the variables must be linear
- If the conditions are not met: use Spearman correlations
 - Examples: likert scale (ordinal data)

SE as a Human Activity

- Agile development
- Pair programming
- SE hiring process
- Diversity and biases

Agile Development

 agile

/ˈɑːjəl/

adjective

1. able to move quickly and easily.
"Ruth was as agile as a monkey"

Similar: nimble lithe spry supple limber sprightly acrobatic

2. relating to or denoting a method of project management, used especially for software development, that is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans.
"agile methods replace high-level design with frequent redesign"



- Software development is considered **agile** when the team requires relatively little time, cost, personnel, and resources to respond to a requirement change
- Team **autonomy**: the extent to which the software team has authority and control in making decisions to carry out the project
- Team **diversity**: the extent to which team members have different functional backgrounds, skills, expertise and experience

Does Agile Work? (1/2)

- “A systematic review of empirical studies of agile software development up to and including 2005 was conducted. The search strategy identified 1996 studies, of which 36 were identified as empirical studies. ... We identified a number of reported benefits and limitations of agile development within each of these themes. **However, the strength of evidence is very low,** which makes it difficult to offer specific advice to industry.”
- [Dyba and Dingsoyr. *Empirical studies of agile software development: A systematic review.*]

Does Agile Work? (2/2)

- “Using an integrated research approach that combines quantitative and qualitative data analyses ... of survey responses of 399 software project managers suggest ... team autonomy has a **positive effect on response efficiency** [on-time completion] and a **negative effect on response extensiveness** [software functionality], and that team diversity has a **positive effect on response extensiveness.**”
- [Lee and Xia. *Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility.*]

Pair Programming



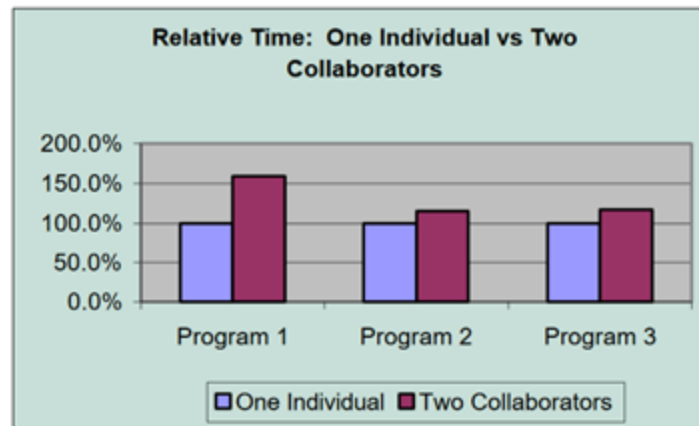
- **Pair programming** refers to the practice whereby two programmers work together at one computer, collaborating on the same design, algorithm, code, or test.
- The pair is made up of a **driver**, who actively **types** at the computer or records a design; and a **navigator** (or **observer**), who **watches** the work of the driver and attentively **identifies** problems, **asks** clarifying questions, and **makes** suggestions. Both are also continuous brainstorming partners.

Pair Programming



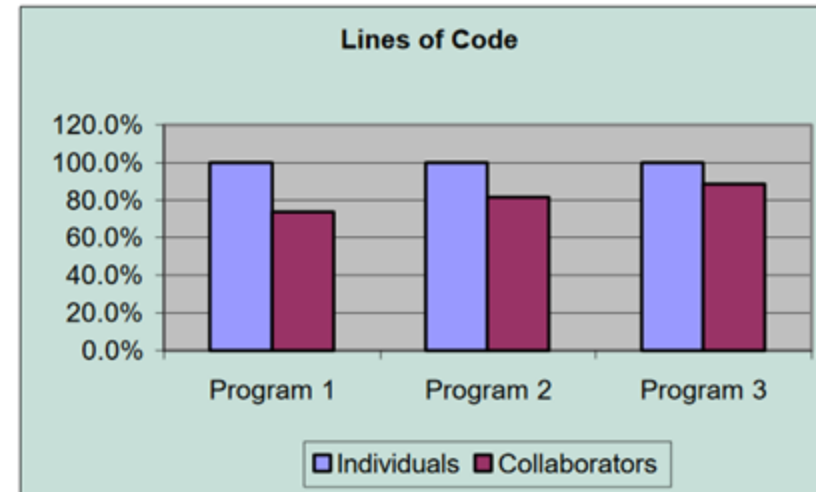
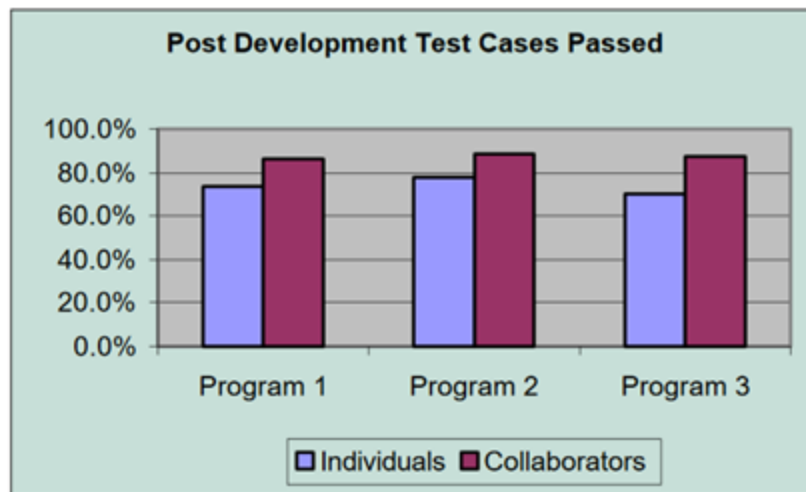
Pair Programming and Programmers

- Surveys of professional programmers
 - 90+% “enjoyed collaborative programming more than solo programming”
 - 95% were “more confident in their solutions” when they pair programmed
- **Increases development cost** by 15% to 100%



Pair Programming and Program Quality

- Reduces defects by 15%
- Reduces code size by 15%



- [Cockburn and Williams. *The Costs and Benefits of Pair Programming.*]

Example Process Decision

(suppose 15% slower coding total, 15% fewer bugs total)

- 50,000 LOC program
- Coding at 50 LOC/hour (wait, what?)
- Defect rate of 10 defects / KLOC
- Defect fix time of 10 hours /defect
- As Individuals:
 - 1,000 hr coding + 5,000 hr fixing defects = 6,000
- As Pairs:
 - 1,150 hr coding + 4,250 hr fixing defects = 5,400

Pair Programming: Important Math Note

- The total “costs” and “benefits” of pair programming are **already included** in the numbers quoted to you
- For example, when we say pair programming increases costs by 15% to 100%, if it's 15%, you **do not** first multiply by 2 (for the pair) and then calculate the 15%
- The cost of having two people work **is already factored in** to the 15% to 100% overhead. So the 100% worst-case is the “multiply by 2”, but the 15% case is “we are magically much faster working together”. **That's the pair benefit!**
- Similarly, in the previous slide **do not** both say “the code is 15% smaller and then the 15% smaller code has 15% fewer defects on top of that” – the 15% fewer defects is already the total benefit. No double counting!

Pair Programming vs. Education

- North Carolina State University and the University of California at Santa Cruz, did extensive pair programming studies with ~1200 beginning computer science students (CS1) and with ~300 third/fourth year software engineering students over three year periods
- Students who paired in CS1 were more likely to attempt CS2 (77% vs. 62%)
- Students who paired in CS1 were more likely to major in CS (57% vs. 34% at NCSU, 25% vs. 11% at UCSC, $p < 0.01$)

Typical CS Hiring Process

- Someone at the company, typically a **recruiter** or an **engineer**, gets your resume and puts it into their pipeline
- If they're interested, you'll probably get one or two **phone screen interviews**
- If you pass the phone screen, you'll probably be invited to interview with the company **on-site**
- Depending on the company, you may then have some **follow-up phone calls** to find a team to be placed on
- If they offer you a job, you'll **negotiate** the offer to end up with the best deal possible
- If this particular offer is the best out of all the offers you've received, you **accept!**
- This can be spread out as much as several months, or as compact as two weeks

Skill-Based Technical Interview Goals

- “The interview process at Google has been designed (and redesigned!) from the ground up to avoid false positives. **We want to avoid making offers to candidates who would not be successful at Google.** (The cost of this unfortunately includes more false negatives, which are times when we turn down somebody who would have done well.)”



Google's Information Needs: “A Good Fit”

- Are you good at CS? [Skill]
 - Can you write and test code?
 - Are you someone they want writing code they will use and depend on?
 - Can you think on your feet?
- **Can you communicate CS concepts? [Behavioral]**
 - Can you explain your ideas to coworkers?
 - Are you someone who would make their team better?
- **Are you a nice person? [Behavioral]**
 - Are you someone they want to work with?
 - And are you friendly enough to chat with every day?

Interview Format

- “For about 45 minutes you meet with a single technical interviewer, who will present a programming problem and ask you to work out one or more solutions to it.”
- Interviewer perspective: “you know in the first ten minutes”



A Medium-Difficulty Example ("The Two-Sum Problem")

- You are given an array of n integers and a number k . Determine if there is a pair of elements in the array that sums to exactly k .
- For example, given the array $[1, 3, 7]$ and $k = 8$, the answer is "yes," but given $k = 6$ the answer is "no."

Questions You Ask

- Can you modify the array? Yes.
- Do we know something about the range of the numbers in the array? No, they can be arbitrary integers.
- Are the array elements necessarily positive? No, they can be positive, negative, or zero.
- Do we know anything about the value of k relative to n or the numbers in the array? No, it can be arbitrary.
- Can we consider pairs of an element and itself? No, the pair should consist of two different array elements.
- Can the array contain duplicates? Sure, that's a possibility.
- What about integer overflow? Don't worry about it.

Software Microcosm

- If you do not convey that you have mastered skill X, they will assume you have not
- They will assume how you write this program is how you will write every program
- They are looking for reasons to reject you
- “Saying true things” vs. “Not saying false things”
- Thus, **even though the problem is small and simple**, you should **show all of the steps** of the software engineering process

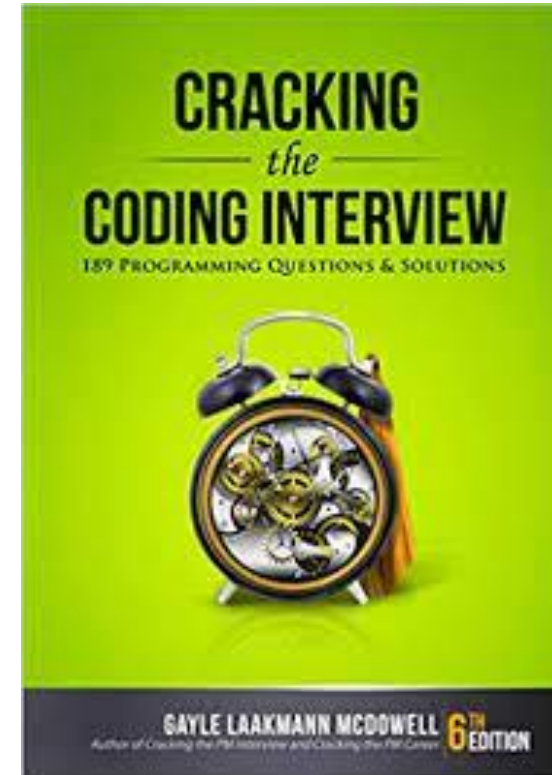
Do Not Forget

- Even though the problem is small, you should
 - Perform requirements elicitation
 - Ask about functional and quality properties
 - Talk about process considerations
 - Talk about how you design for maintainability
 - Write commented code, including method-level and statement-level documentation (what/why)
 - Write tests that show off corner cases
 - Talk about other approaches to QA (within reason)

Top 10 Mistakes in Interview Prep

[Gayle McDowell, *Cracking the Coding Interview*]

- #1 Practicing on a computer
- #2 Not rehearsing behavioral questions
- #3 Not doing a mock interview
- #4 Trying to memorize solutions
- #5 Not solving problems out loud
- #6 Rushing
- #7 Sloppy coding (bad style),
- #8 Not testing
- #9 Fixing mistakes carelessly
- #10 Giving up



Behavioral Questions

- What is your greatest weakness?
- Tell me about a time you missed a deadline.
- Tell me about a time you experienced a conflict with a teammate.
- Very easy to sound unimpressive if you have not practiced!

Situation, Action, Result

- Recommendation: structure your responses (especially to “negative” questions):
 - Situation: describe objectively
 - Action: what did you do?
 - Result: how were things better after?
- Be specific, not arrogant



Resume and Interview “Stats”

- Your resume says you worked on *XYZ Project*. What was the most challenging aspect of that?
 - What did you learn the most from? What was the most interesting? What was the hardest bug? What did you enjoy the most? What was the biggest conflict? Most significant requirements change?
- What is the largest program (LOC) you have written? Modified?
What is the largest number of tests you have written? Worked with?
What is the largest team you have worked with? What is the largest process you automated? How many customers have you spoken to?

What do we know? Little so far!

Dazed: Measuring the Cognitive Load of Solving Technical Interview Problems at the Whiteboard

Mahnaz Behroozi¹, Alison Lui², Ian Moore¹, Denae Ford¹, Chris Parnin¹

¹North Carolina State University, Raleigh, NC, USA

²University of Notre Dame, Notre Dame, IN, USA

{mbehroo,ipmoore,dford3,cjparnin}@ncsu.edu,alison.m.lui.2@nd.edu

ABSTRACT

Problem-solving on a whiteboard is a popular technical interview technique used in industry. However, several critics have raised concerns that whiteboard interviews can cause excessive stress and cognitive load on candidates, ultimately reinforcing bias in hiring practices. Unfortunately, many sensors used for measuring cognitive state are not robust to movement. In this paper, we describe an approach where we use a head-mounted eye-tracker and computer vision algorithms to collect robust metrics of cognitive state. To demonstrate the feasibility of the approach, we study two proposed interview settings: on the whiteboard and on paper with 11 participants. Our preliminary results suggest that the whiteboard setting pressures candidates into keeping shorter attention lengths and experiencing higher levels of cognitive load compared to solving the same problems on paper. For instance, we observed 60ms shorter fixation durations and 3x more regressions when solving problems on the whiteboard. Finally, we describe a vision for cre-

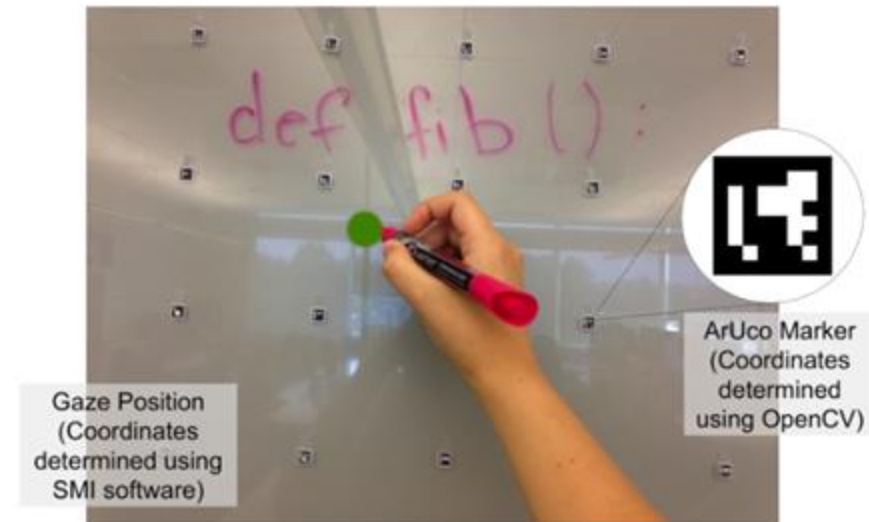
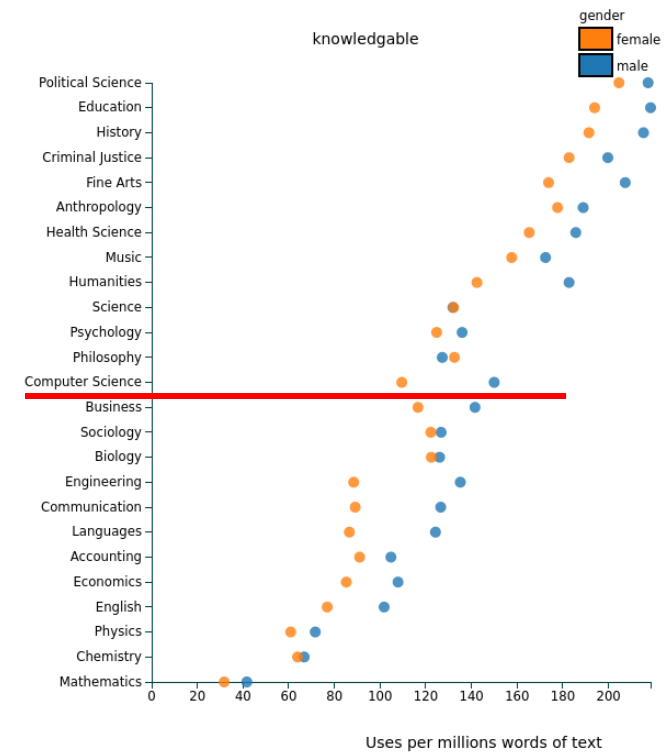
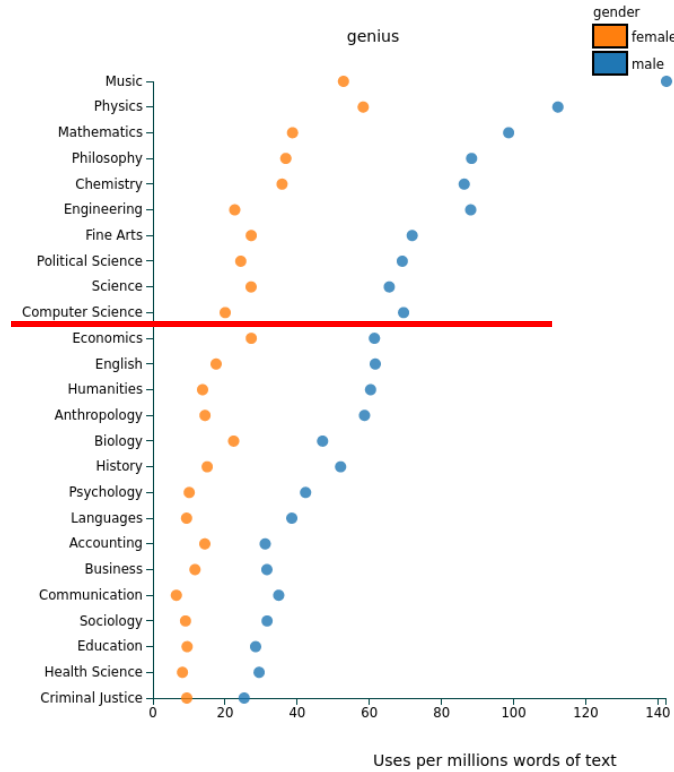
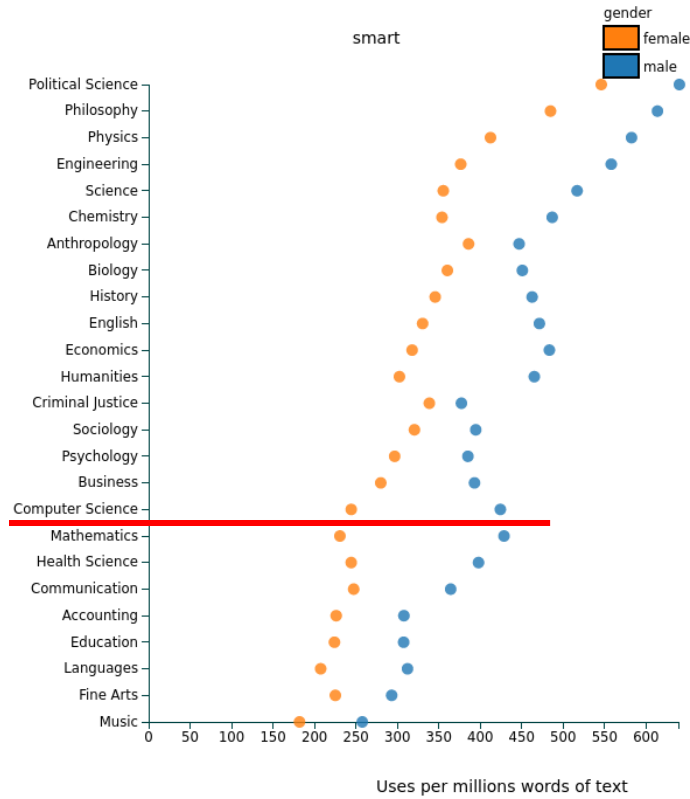


Figure 1: Feasibility study of using ArUco markers to calculate regressions.

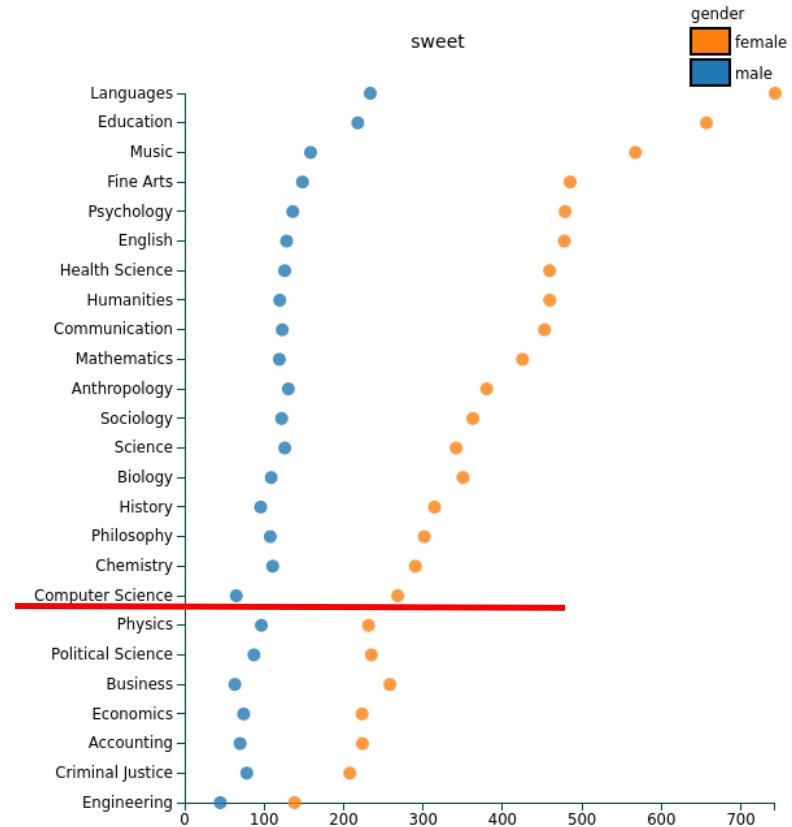
Biases and Diversity (endless)...

- Ratemyprofessors.com
- 14 million reviews
- [A new tool](#) allows those being rated (or anyone) to see the way students tend to use different words when rating male and female professors -- generally to the disadvantage of the latter.

Biases and Diversity (endless)...



Biases and Diversity (endless)...



More on Biases and Diversity (endless)...

Can salience of gender identity impair math performance among 7-8 years old girls?
The moderating role of task difficulty

Emmanuelle Neuville

University Blaise Pascal, Clermont-Ferrand, CNRS, France

Jean-Claude Croizet

University of Poitiers, France

Can the salience of gender identity affect the math performance of 7–8 year old girls? Third-grade girls and boys were required to solve arithmetical problems of varied difficulty. Prior to the test, one half of the participants had their gender identity activated. Results showed that activation of gender identity affected girls' performance but not boys. When their gender was activated as opposed to when it was not, girls solved more problems when the material was less difficult but underperformed on the difficult problems. Results are discussed with regard to the stereotype threat literature.

More on Biases and Diversity (endless)...

Gender, Confidence, Math: Why Aren't the Girls "Where the Boys Are?"

Caporrimo, Rosaria

Analyses were conducted to examine the relationship of standardized mathematics achievement scores, problem-solving strategies, self-report scores, and Confidence in Learning Mathematics survey scores among 122 eighth-grade students, 70 females and 52 males, representing all levels of mathematics achievement. Among the findings, no gender differences were evident on any of these scores; however, the Confidence scores functioned differently for the sexes. When consideration was focused upon average scores on the problem-solving strategies measure, males exhibited a direct relationship between routine problem scores and Confidence scores, whereas females showed an inverse relationship. (22 references) (JJK)

More on Biases and Diversity (endless)...

Several great papers on biases and diversity!