UMSI

SI 630

Natural Language Processing: Algorithms and People

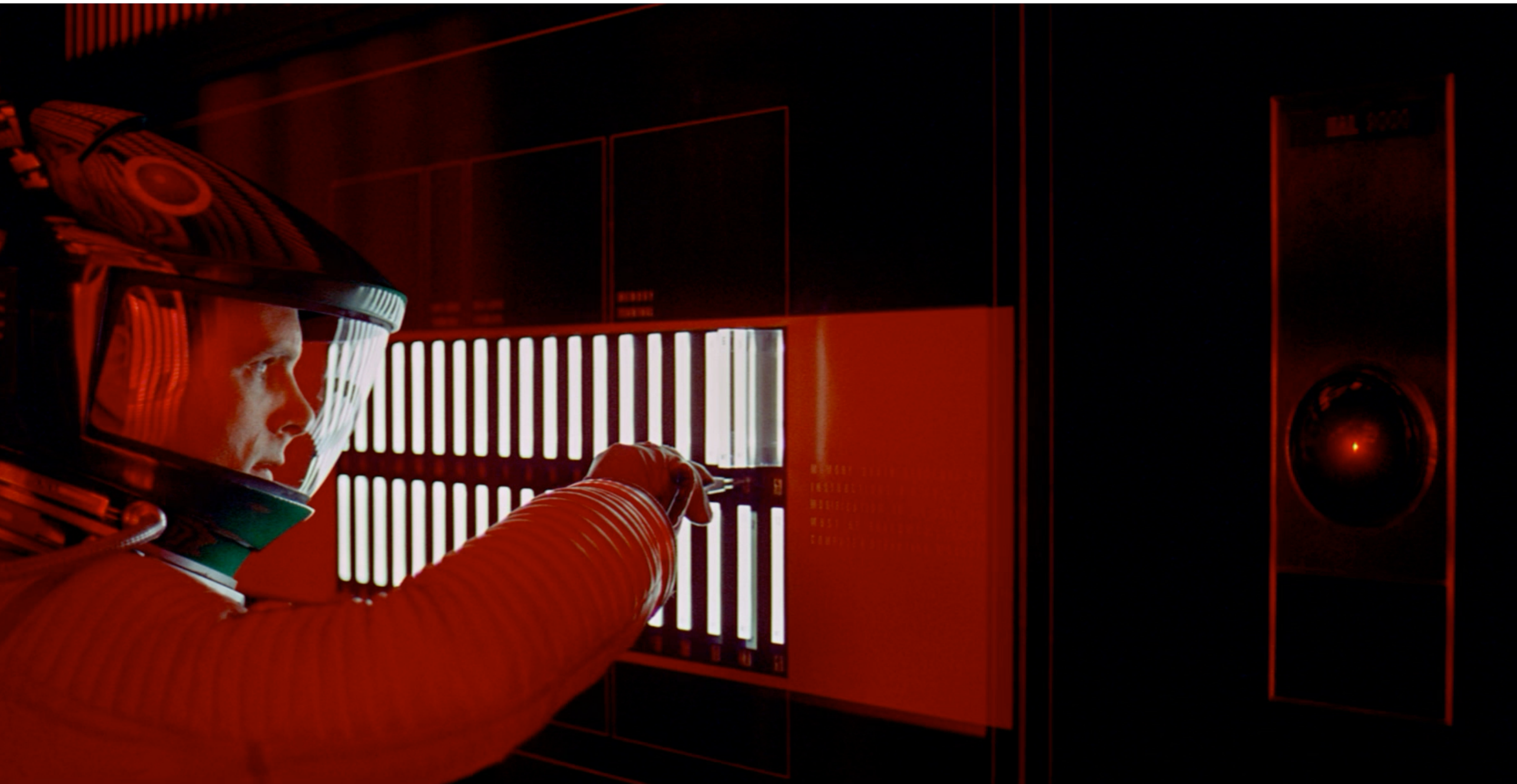
David Jurgens
jurgens@umich.edu

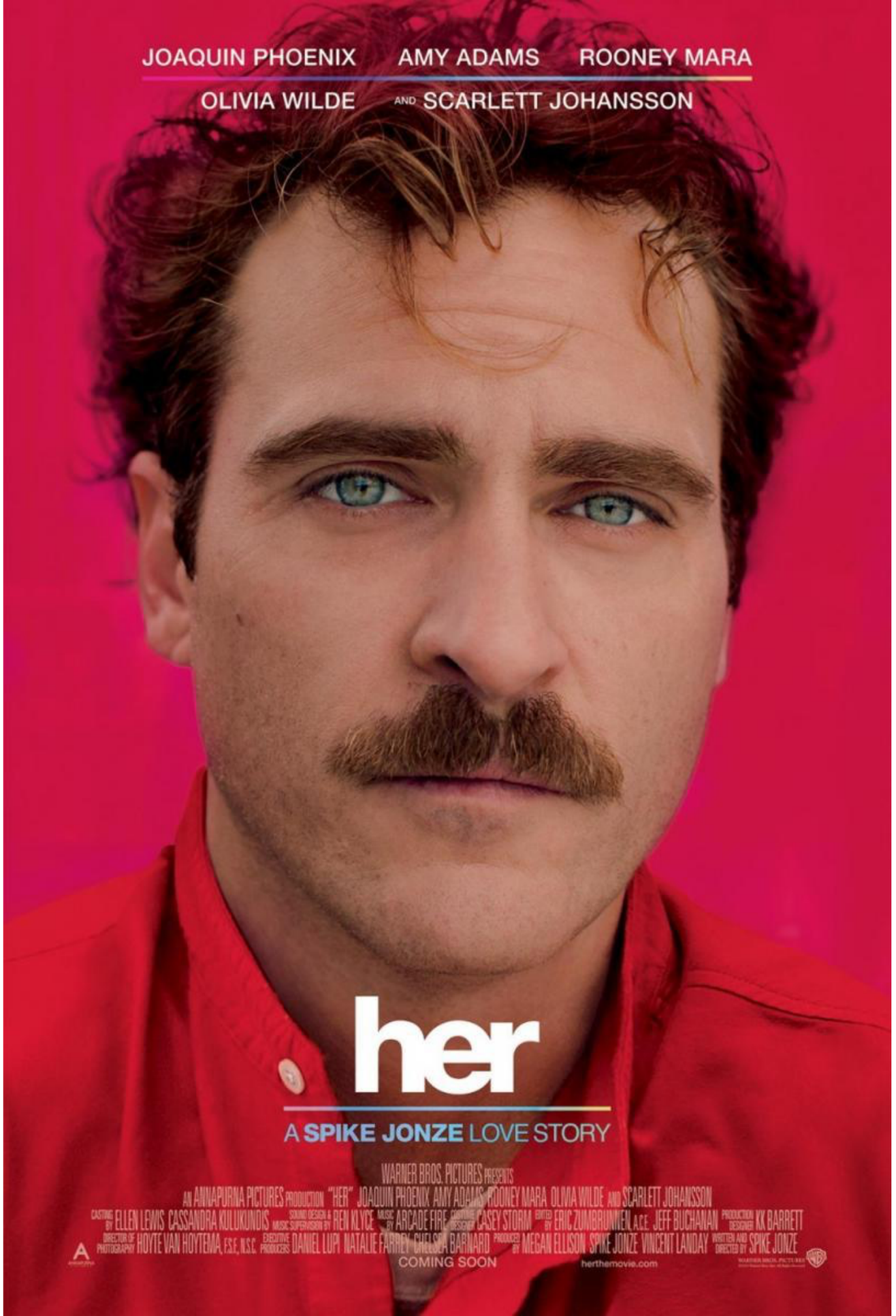
Some slides from David Bamman and Dan Jurafsky

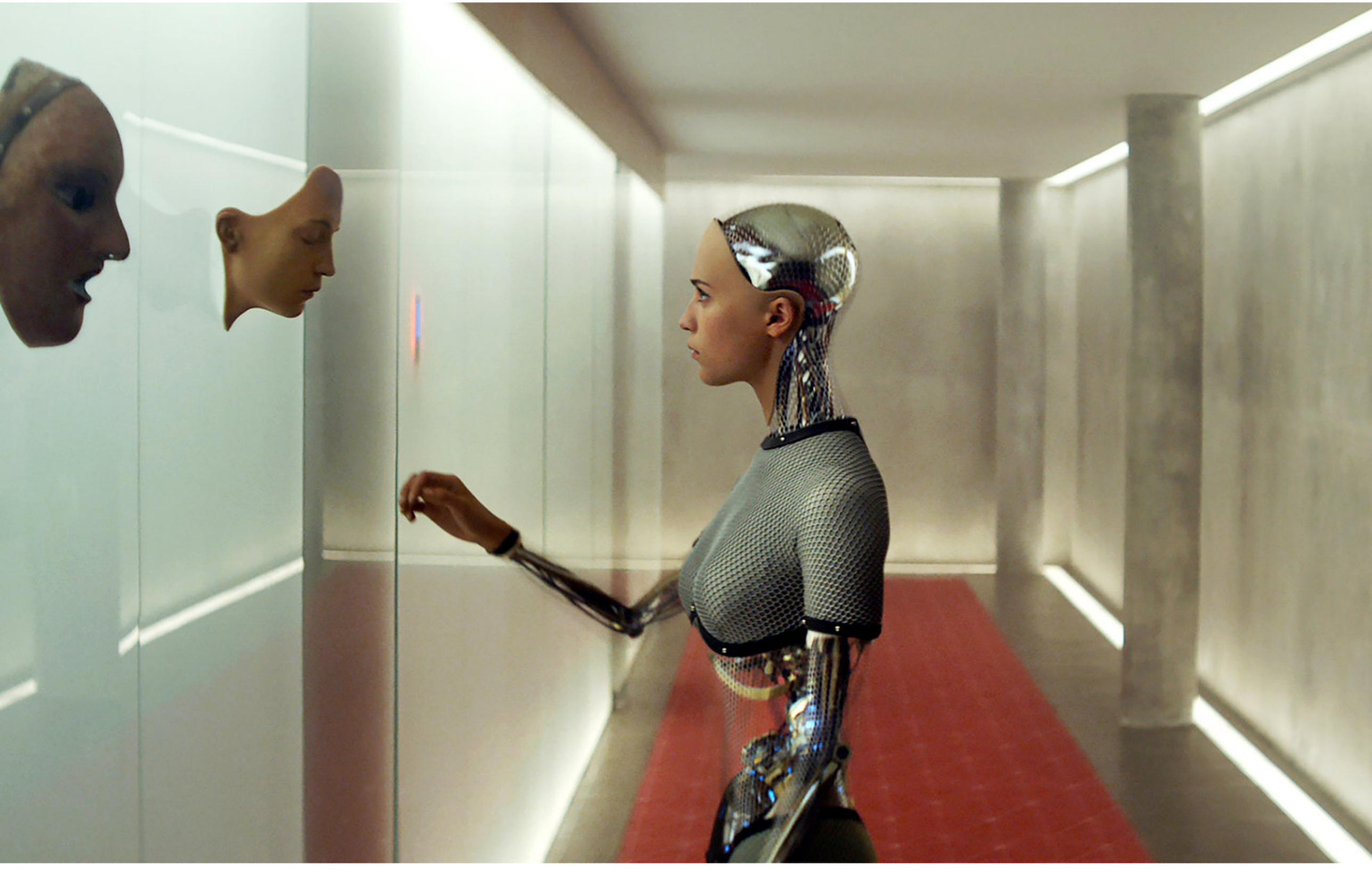

Noam
Chomsky

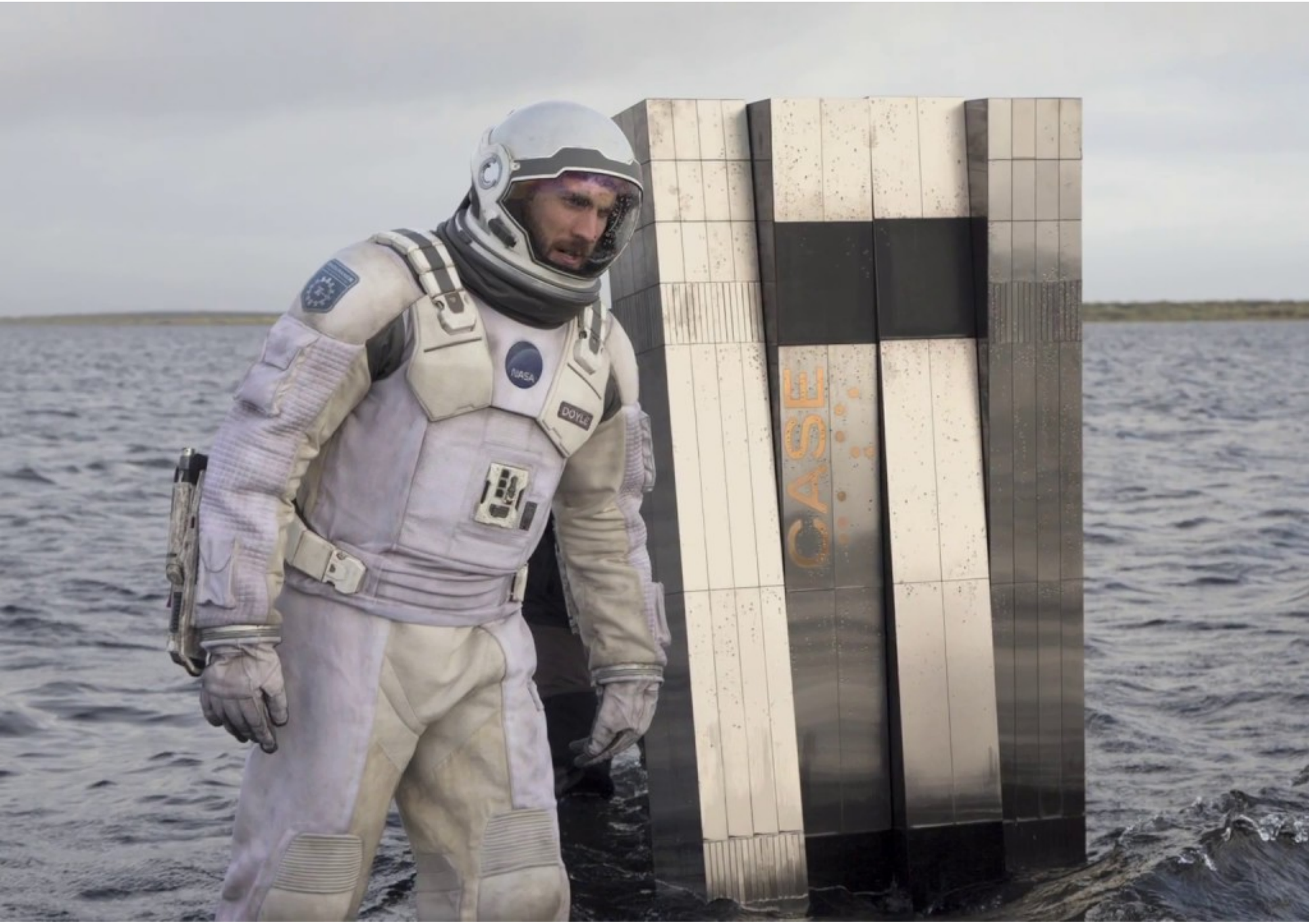
NLP = processing* language with computers

processing as “understanding”

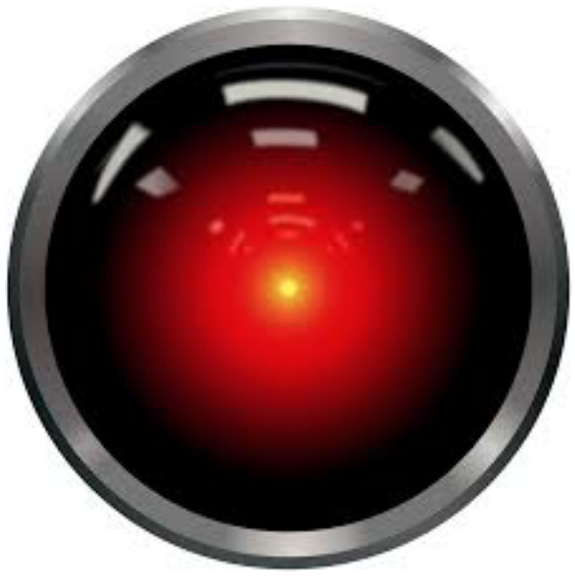



JOAQUIN PHOENIX AMY ADAMS ROONEY MARA
OLIVIA WILDE AND SCARLETT JOHANSSON
her
A SPIKE JONZE LOVE STORY
WARNER BROS. PICTURES PRESENTS
AN ANNAPURNA PICTURES PRODUCTION "HER" JOAQUIN PHOENIX AMY ADAMS ROONEY MARA OLIVIA WILDE AND SCARLETT JOHANSSON
CASTING BY ELLEN LEWIS CASSANDRA KULUKUNDIS SOUND DESIGN & MUSIC SUPERVISION BY REN KLYCE MUSIC BY ARCADE FIRE PRODUCTION DESIGNER CASEY STORM EDITED BY ERIC ZUMBRUNNEN, A.C.E. JEFF BUCHANAN PRODUCTION DESIGNER KK BARRETT
DIRECTOR OF PHOTOGRAPHY HOYTE VAN HOYTEMA, FSF, NSC EXECUTIVE PRODUCERS DANIEL LUPI NATALIE FARRELL CHELSEA BARNARD PRODUCED BY MEGAN ELLISON SPIKE JONZE VINCENT LANDAY WRITTEN AND DIRECTED BY SPIKE JONZE
COMING SOON
herthemovie.com




NASA
DOYLE
CAS

LIFE PRO TIP: Don’t create evil NLP Systems



Dave Bowman: Open the pod bay doors, HAL
HAL: I’m sorry Dave. I’m afraid I can’t do that

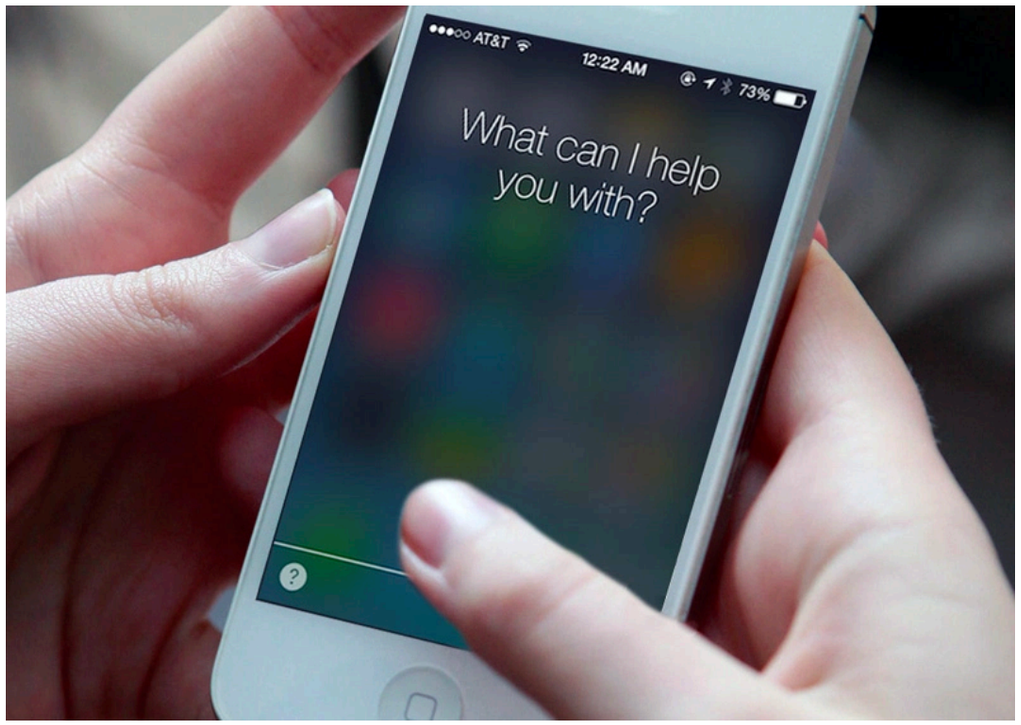
Agent	Movie	Complex human emotion mediated through language
Hal	2001	Mission execution
Samantha	Her	Love
David	Prometheus	Creativity

Turing test

Distinguishing human vs. computer only through written language

Turing 1950

The future seems bright

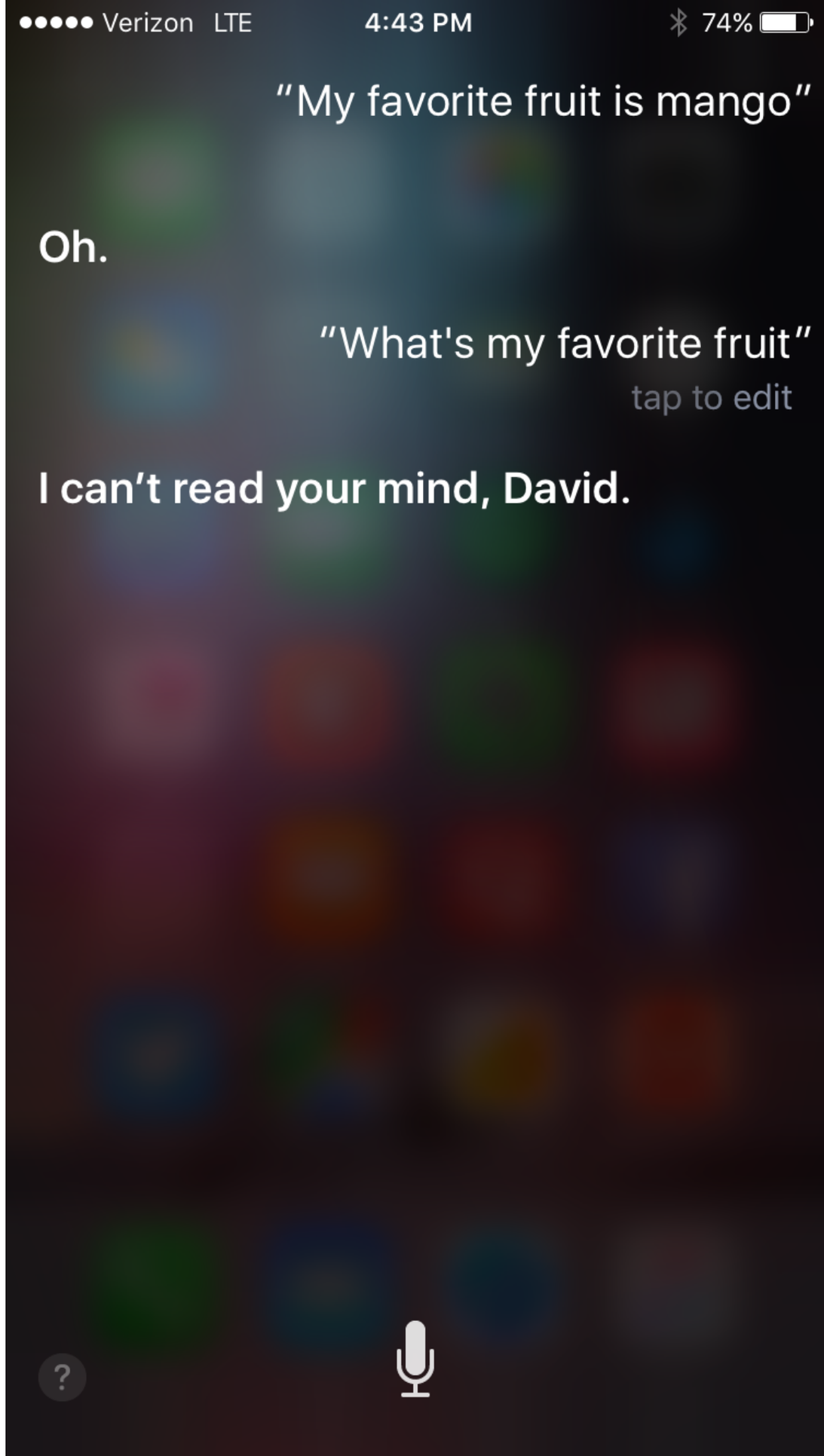


IBM Watson

Where we are now

Where we are now

Where we are now



Baseline mutual information model (Li et al. 2015)
A: Where are you going? (1)
B: I'm going to the restroom. (2)
A: See you later. (3)
B: See you later. (4)
A: See you later. (5)
B: See you later. (6)
...

A: how old are you? (1)
B: I'm 16. (2)
A: 16? (3)
B: I don't know what you are talking about. (4)
A: You don't know what you are saying. (5)
B: I don't know what you are talking about . (6)
A: You don't know what you are saying. (7)
...

Li et al. (2016), "Deep Reinforcement Learning for Dialogue Generation" (EMNLP)

What makes language hard?

- Language is a complex social process
- Tremendous ambiguity at every level of representation
- Modeling it is AI-complete (requires first solving general AI)

What makes language hard?

- Speech acts (“can you pass the salt?)
  [Austin 1962, Searle 1969]
- Conversational implicature (“The opera singer was amazing; she sang all of the notes”).
  [Grice 1975]
- Shared knowledge (“Clinton ran for election”)
- Variation/Indexicality (“This homework is wicked hard”)
  [Labov 1966, Eckert 2008]

Ambiguity

"One morning I shot
an elephant in my pajamas"

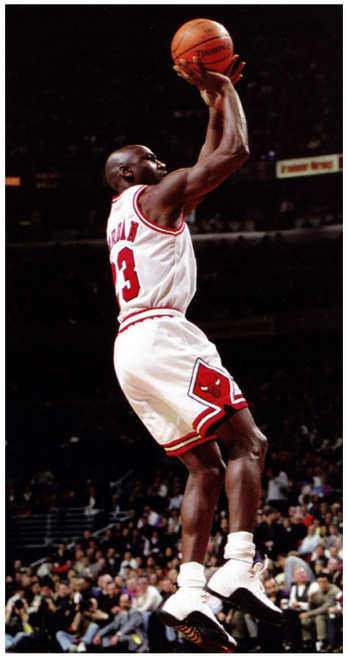
Animal Crackers

Ambiguity

"One morning I shot
an elephant in my pajamas"

Animal Crackers

Ambiguity



“One morning I shot
an elephant in my pajamas”

Ambiguity

verb noun

"One morning I shot
an elephant in my pajamas"

Animal Crackers

I made her duck

[SLP2 ch. 1]

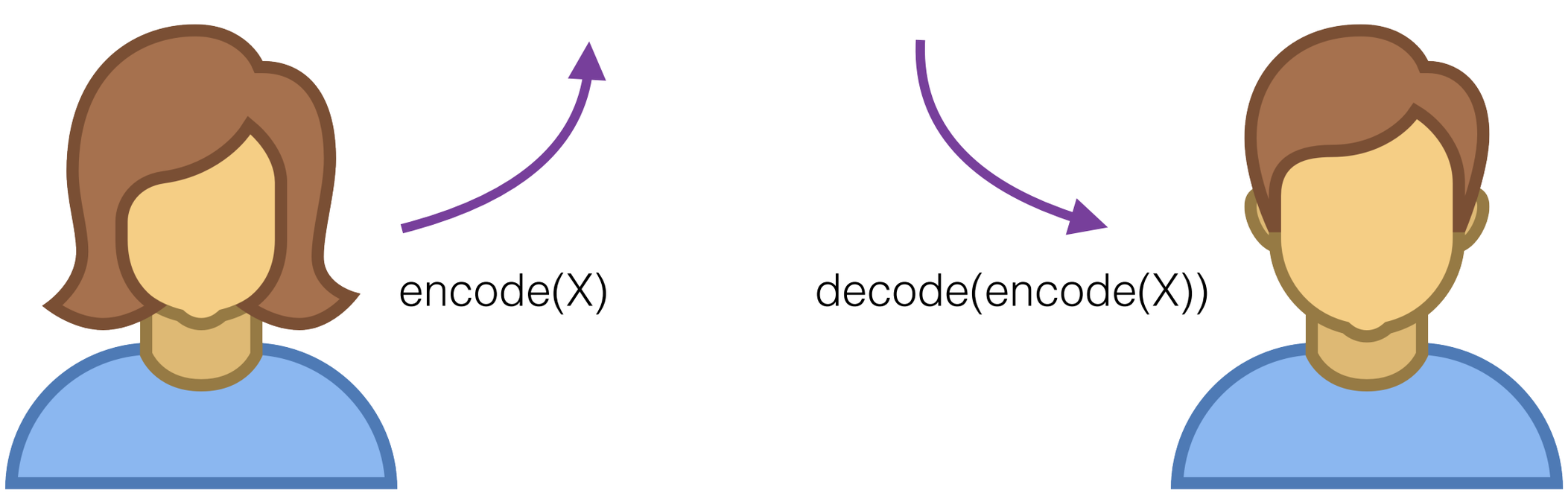
- I cooked waterfowl for her
- I cooked waterfowl belonging to her
- I created the (plaster?) duck she owns
- I caused her to quickly lower her head or body
- …

processing as representation

- NLP generally involves representing language for some end, e.g.:
  - dialogue
  - translation
  - speech recognition
  - text analysis

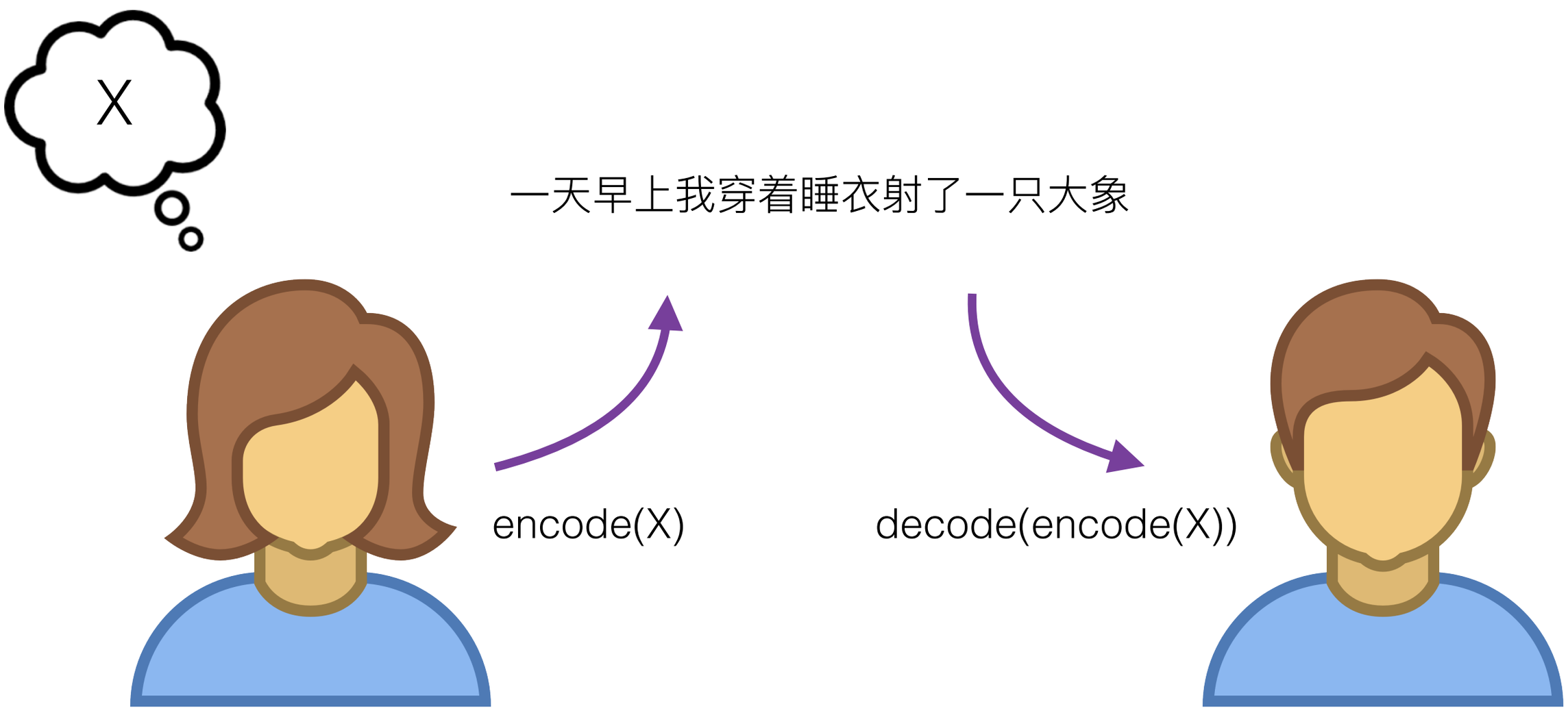
Information theoretic view

"One morning I shot an elephant in my pajamas"



Shannon 1948

Information theoretic view

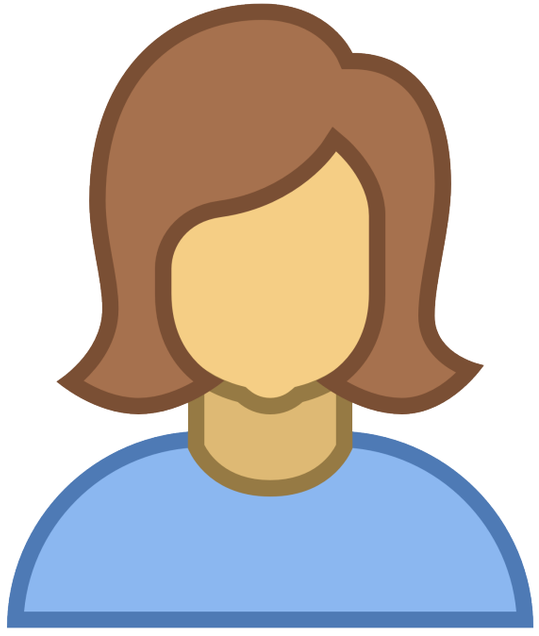


When I look at an article in Russian, I say: 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.'

Weaver 1955

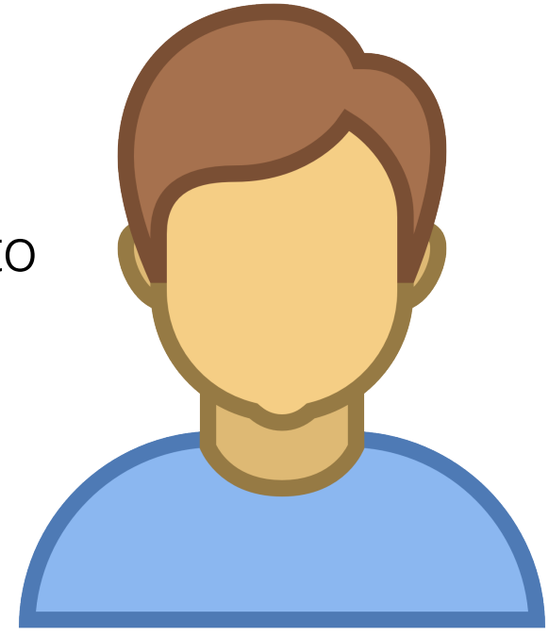
Rational speech act view

"One morning I shot an elephant in my pajamas"



X

Communication involves recursive reasoning: how can X choose words to maximize understanding by Y?

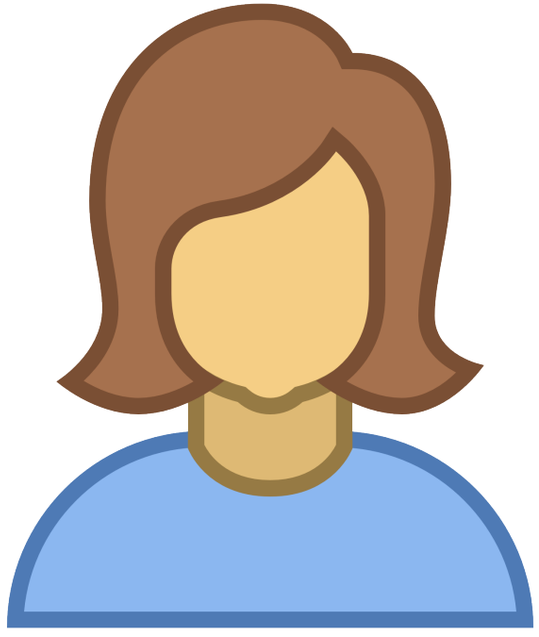


Y

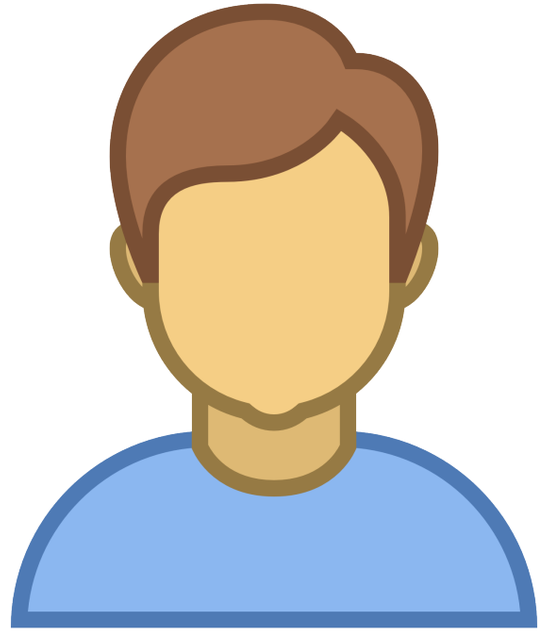
Frank and Goodman 2012

Pragmatic view

"One morning I shot an elephant in my pajamas"

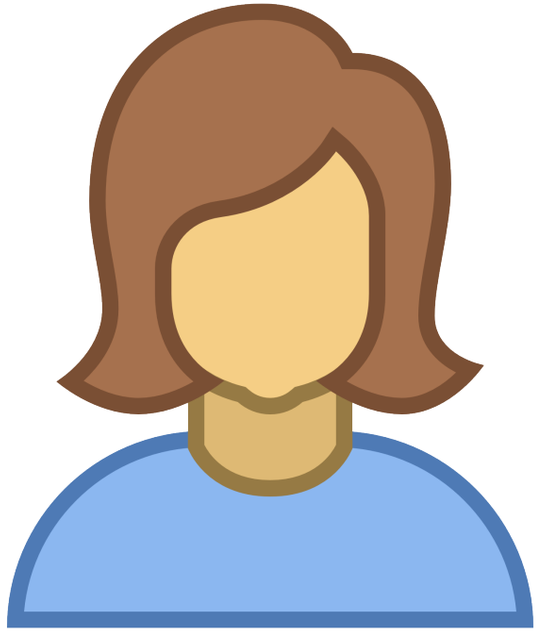


Meaning is co-constructed by the interlocutors and the context of the utterance

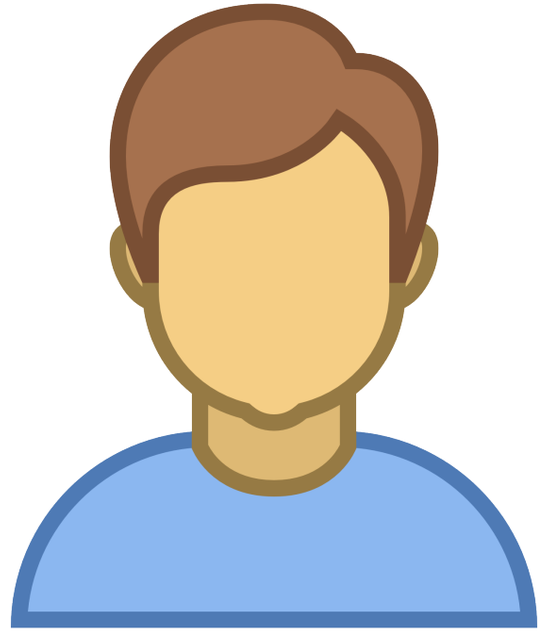


Whorfian view

"One morning I shot an elephant in my pajamas"

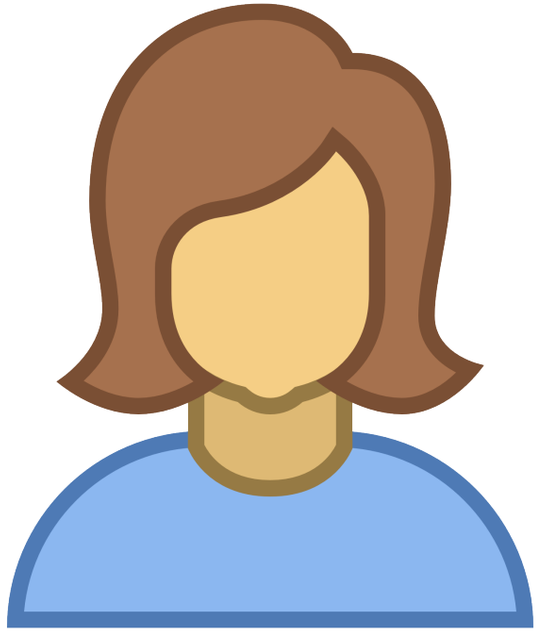


Weak relativism: structure of language influences thought

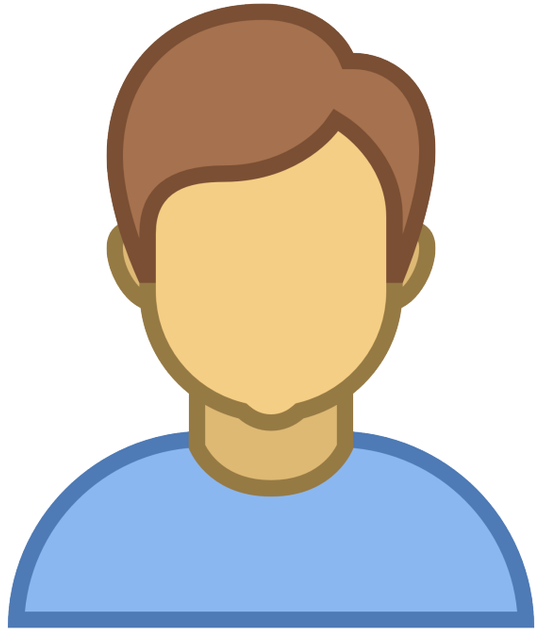


Whorfian view

一天早上我穿着睡衣射了
一只大象

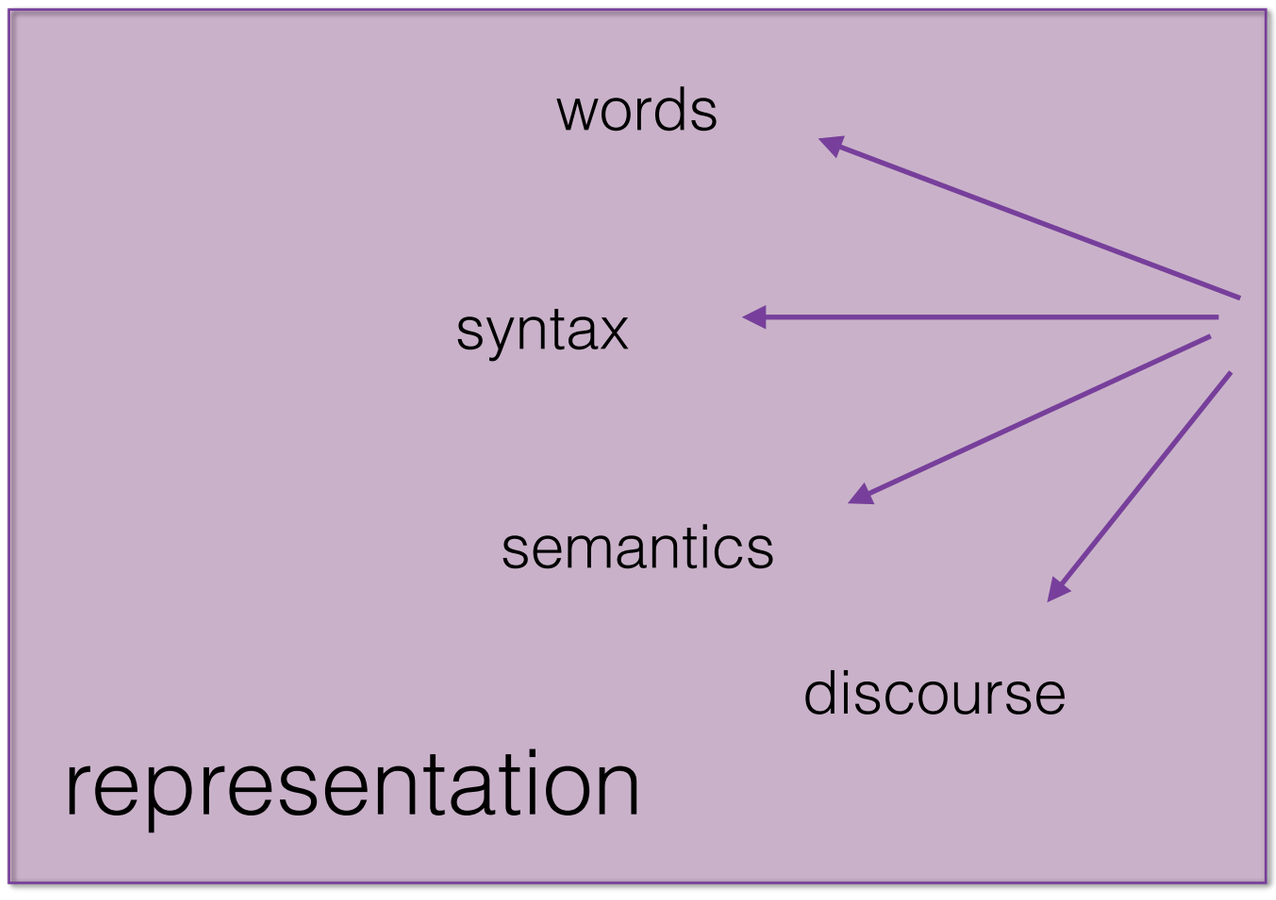


Weak relativism: structure of language influences thought

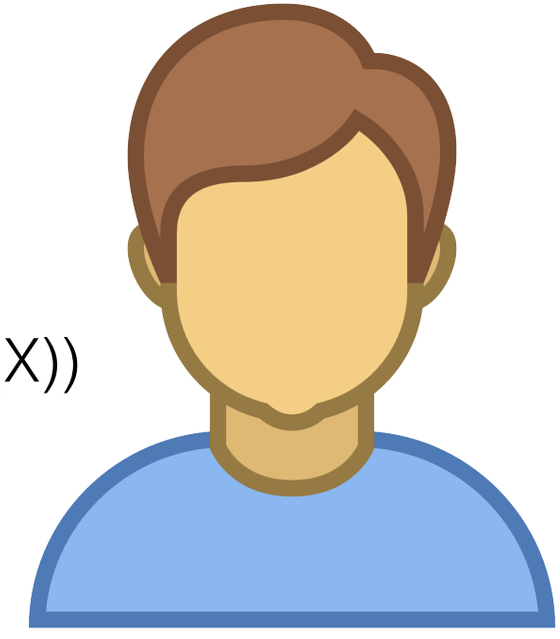


Decoding

“One morning I shot an elephant in my pajamas”



decode(encode(X))



https://upload.wikimedia.org/wikipedia/commons/thumb/b/be/Sydnor_Log_Cabin.png/1200px-Sydnor_Log_Cabin.png

630: Course structure

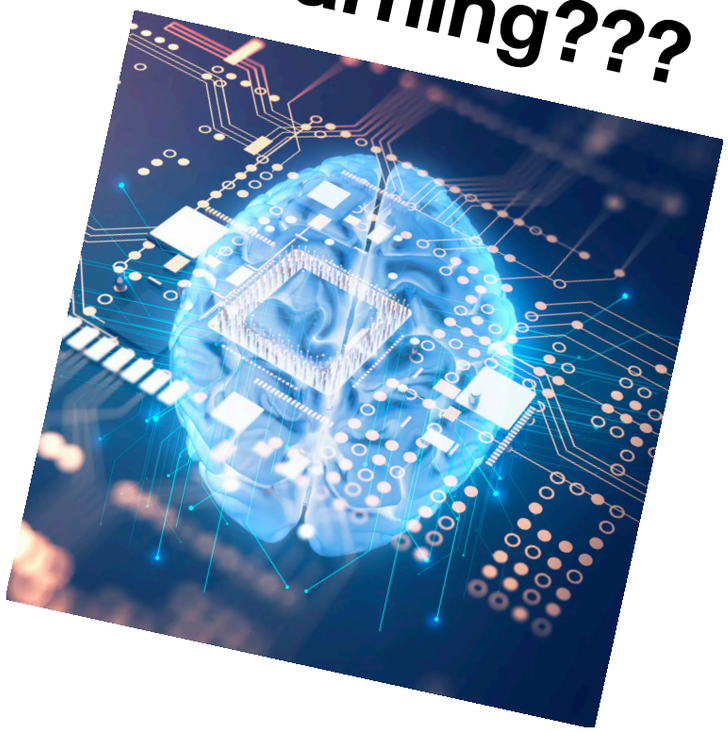
SI 630

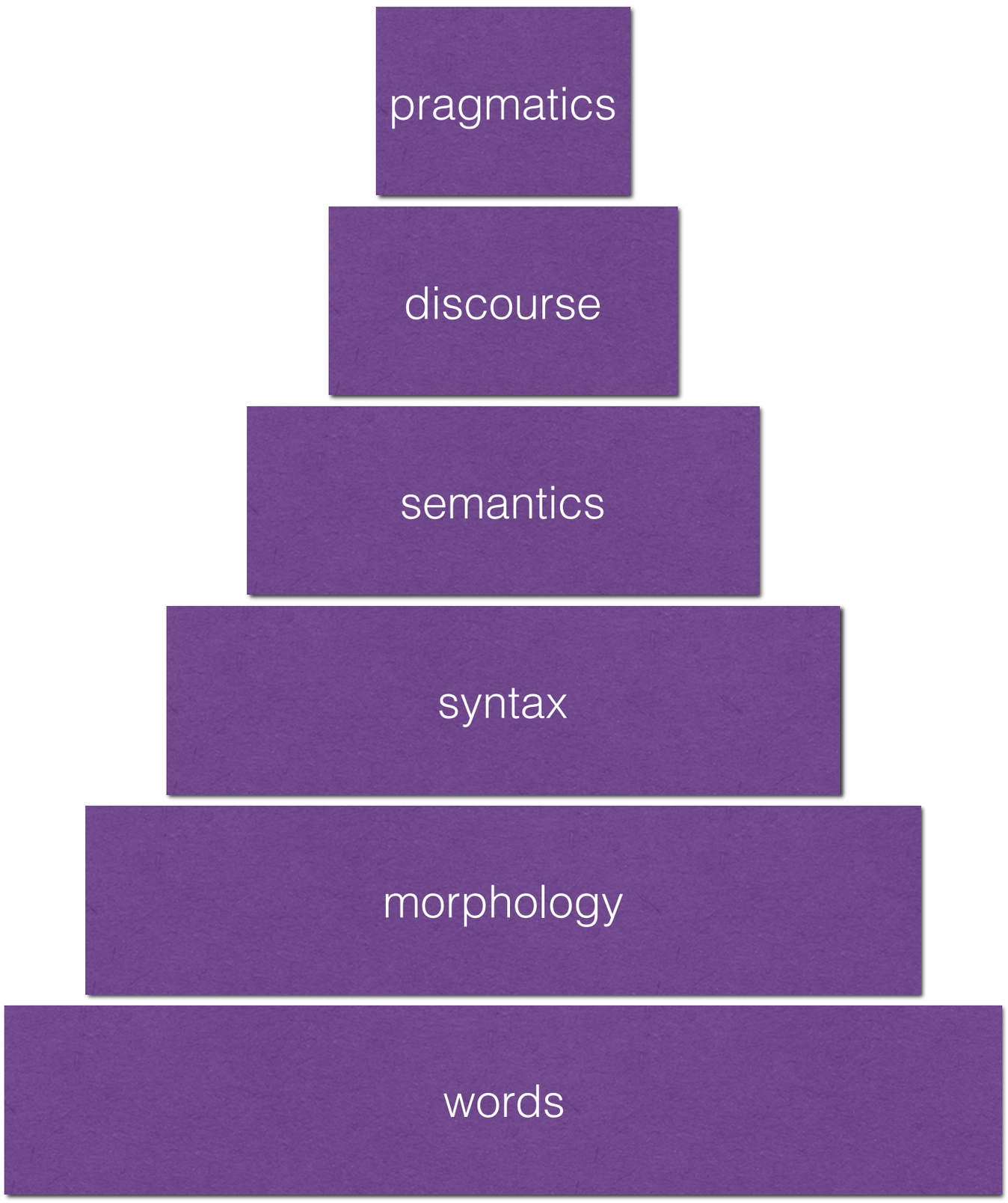
- This is a class about algorithms.
    - You'll learn and implement algorithms to solve NLP tasks efficiently
    - You'll understand the fundamentals to innovate new methods.
- This is a class about how people understand language.
    - You'll see annotated texts for a variety of linguistic representations so you'll understand the phenomena you'll be modeling
    - You'll learn about how to connect algorithms to modeling new phenomena outside of NLP

What are we going to talk about?

- Text Classification
- Sequence Labeling
- Structure in Language (Syntax)
- Topic Modeling
- Information Extraction
- Crowdsourcing
- NLP + X

And who could forget Deep Learning???




pragmatics
discourse
semantics
syntax
morphology
words

Words

- One morning I shot an elephant in my pajamas
- I didn’t shoot an elephant
- Imma let you finish but Beyonce had one of the best videos of all time
- 一天早上我穿着睡衣射了一只大象

Parts of speech

noun verb noun noun

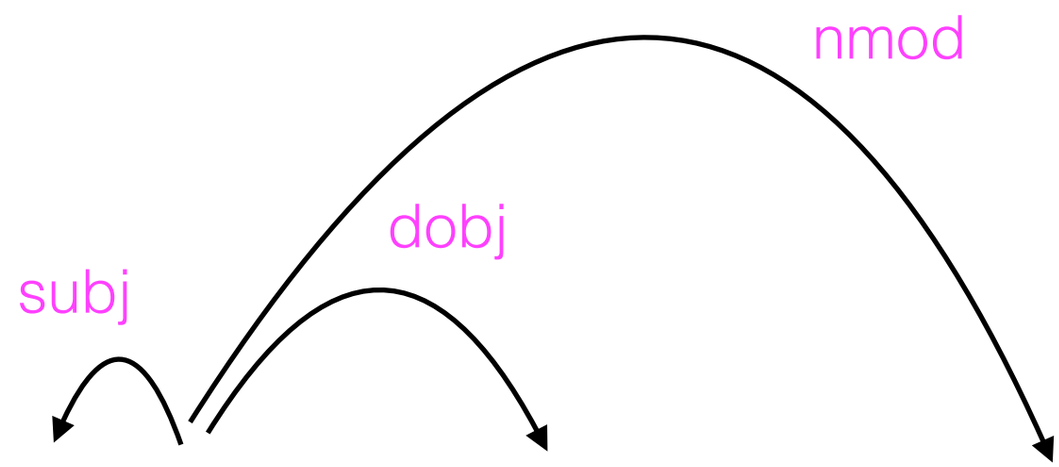
One morning I shot an elephant in my pajamas

Named entities

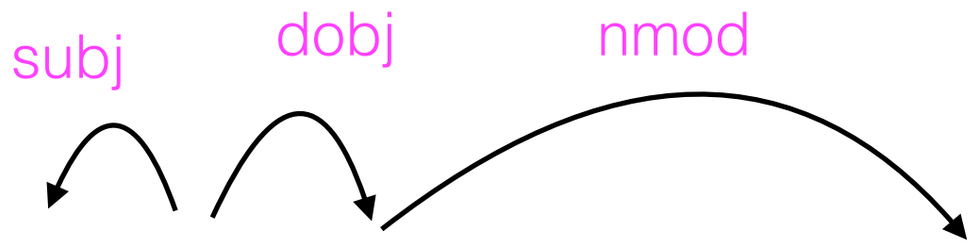
person

Imma let you finish but Beyonce had one of the best videos of all time

Syntax



One morning I shot an elephant in my pajamas



Imma let you finish but Beyonce had one of the best videos of all time

Sentiment analysis

"Unfortunately I already had this exact picture tattooed on my chest, but this shirt is very useful in colder weather."

[overlook1977]

Question answering

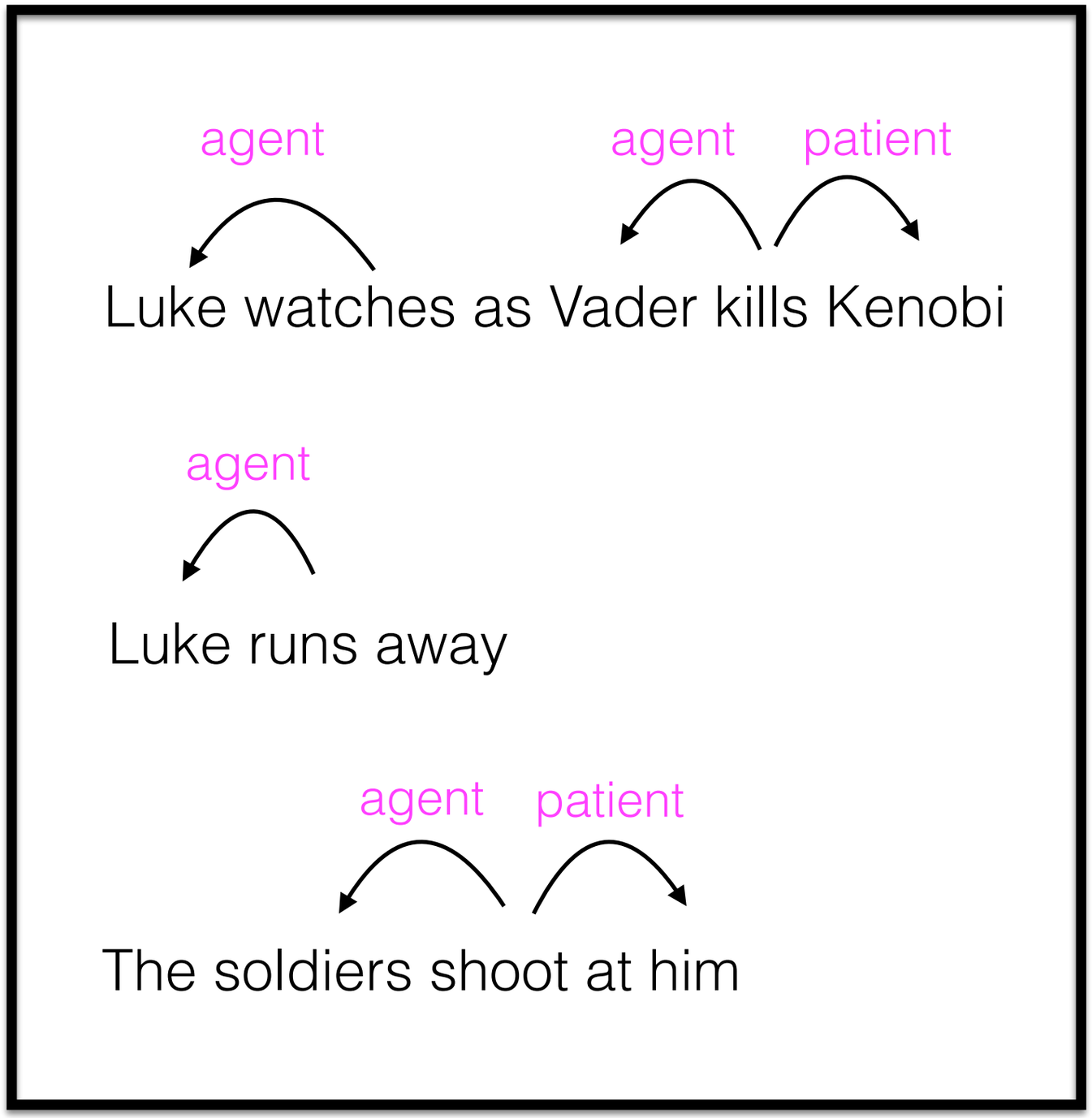
What did Barack Obama teach?

Barack Hussein Obama II (born August 4, 1961) is the 44th and current President of the United States, and the first African American to hold the office. Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he served as president of the *Harvard Law Review*. He was a community organizer in Chicago before earning his law degree. He worked as a civil rights attorney and taught constitutional law at the University of Chicago Law School between 1992 and 2004.

Inferring Character Types

Input: text describing plot of a movie or book.

Structure: NER, syntactic parsing + coreference



NLP

- Machine translation
- Question answering
- Information extraction
- Conversational agents
- Summarization

Google

IBM WATSON

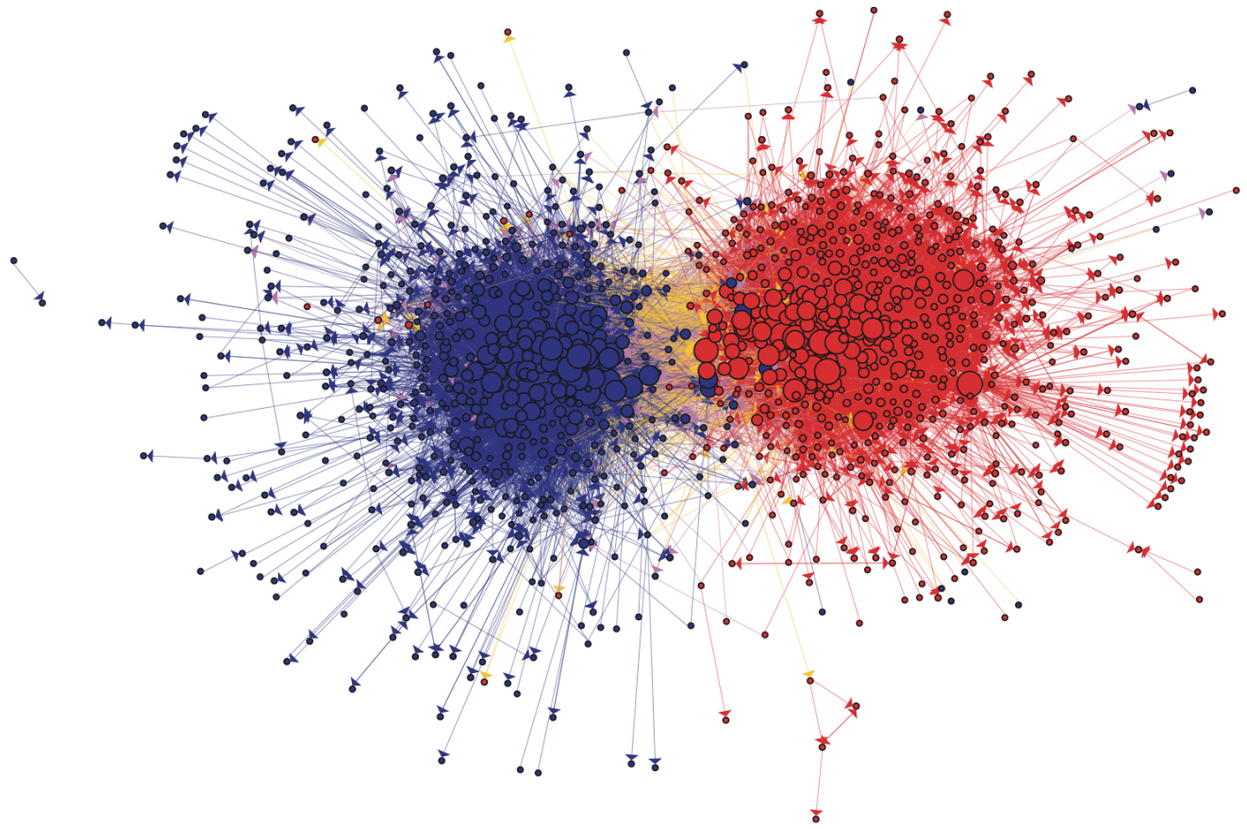
NLP is interdisciplinary

- Artificial intelligence
- Machine learning (ca. 2000—today); statistical models, neural networks
- Linguistics (representation of language)
- Social sciences/humanities (models of language at use in culture/society)

NLP + X

Computational Social Science

- Inferring ideal points of politicians based on voting behavior, speeches
- Detecting the triggers of censorship in blogs/ social media
- Inferring power differentials in language use



Link structure in political blogs
Adamic and Glance 2005

Computational Journalism

What do Journalists do with Documents?
Field Notes for Natural Language Processing Researchers

Jonathan Stray
Columbia Journalism School
jms2361@columbia.edu

- Robust import
- Robust analysis
- Search, not exploration
- Quantitative summaries
- Interactive methods
- Clarity and Accuracy

Computational Humanities

Ted Underwood (2016), “The Life Cycles of Genres,” Cultural Analytics

Ryan Heuser, Franco Moretti, Erik Steiner (2016), The Emotions of London

Richard Jean So and Hoyt Long (2015), “Literary Pattern Recognition”

Andrew Goldstone and Ted Underwood (2014), “The Quiet Transformations of Literary Studies,” New Literary History

Franco Moretti (2005), Graphs, Maps, Trees

Holst Katsma (2014), Loudness in the Novel

So et al (2014), “Cents and Sensibility”

Matt Wilkens (2013), “The Geographic Imagination of Civil War Era American Fiction”

Jockers and Mimno (2013), “Significant Themes in 19th-Century Literature,”

Ted Underwood and Jordan Sellers (2012). “The Emergence of Literary Diction.” JDH

Prerequisites

- Advanced programming skills
  - Translate pseudocode into code (Python*)
  - Analysis of algorithms (big-O notation)

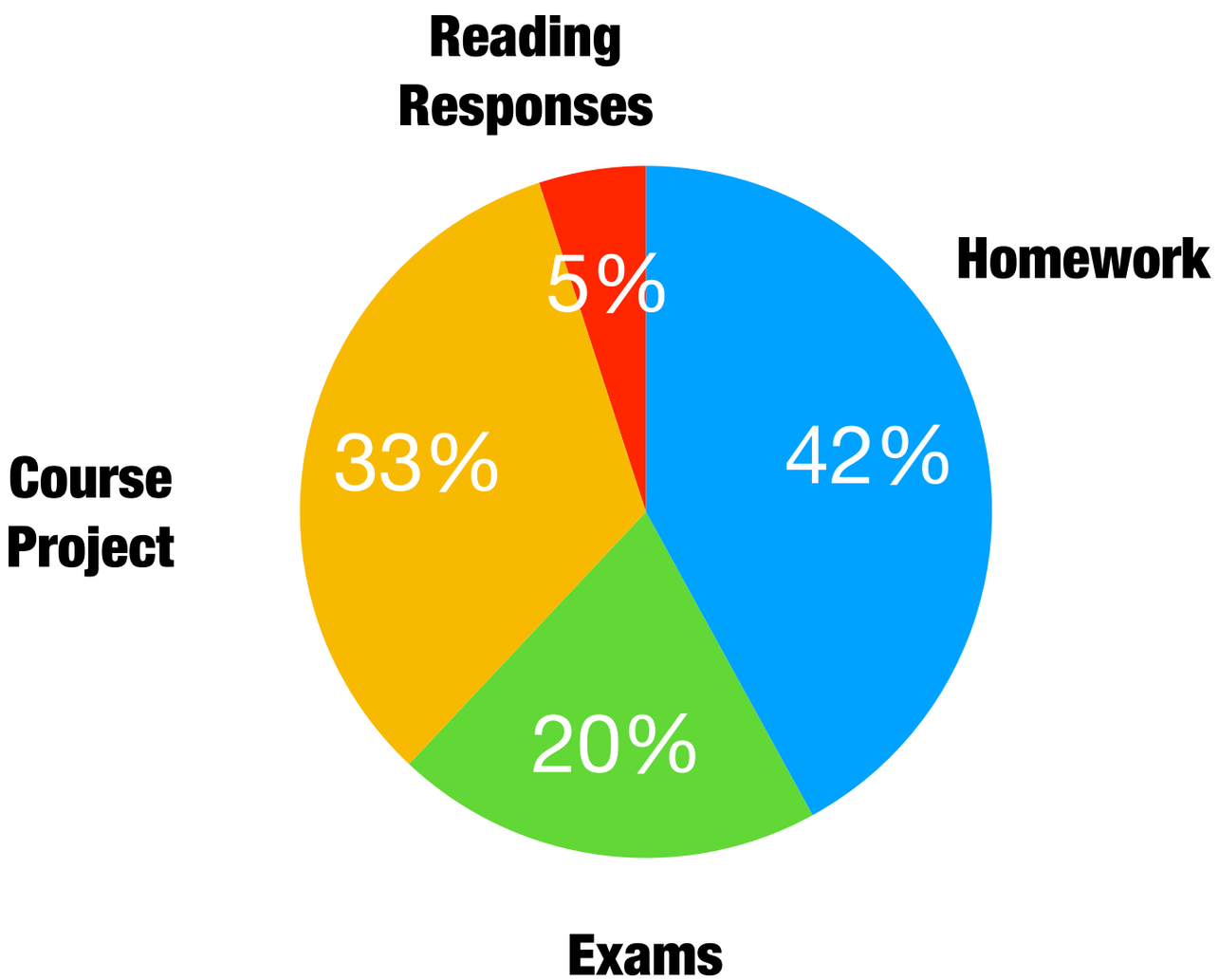
- Basic probability/statistics
  - There will be some math (today will be a good example)

- We'll do everything possible to explain all the details. No more fear of equations or pseudocode. **Be bold**.

Quick note about math and notation

- A big learning objective is to make sure you're comfortable with *understanding mathematical notation.*
  - This is the language of NLP and Data Science.
    - Don't worry if you haven't take a math course since high school.
- If you see some equation and you don't remember/ understand what something means, **ask**.
- Many others probably have the same question and you speaking up helps the whole class.

Grading



Homeworks (42%)

- **Five** real homeworks + 1 warm-up homeworks
  - (mini-) Homework 0: Warm up (2%)
  - Homework 1: Text Classification (8%)
  - Homework 2: Word Vectors (8%)
  - Homework 3: Parsing (8%)
  - Homework 4: Topic Modeling (8%)
  - Homework 5: Deep Learning (8%)

- Around two weeks for each homework.

- Always due 5:30pm before class.

Responses (5%)

- 1 page response to a paper for that week. We provide a template to help you get started!

- Fixed response template on Overleaf

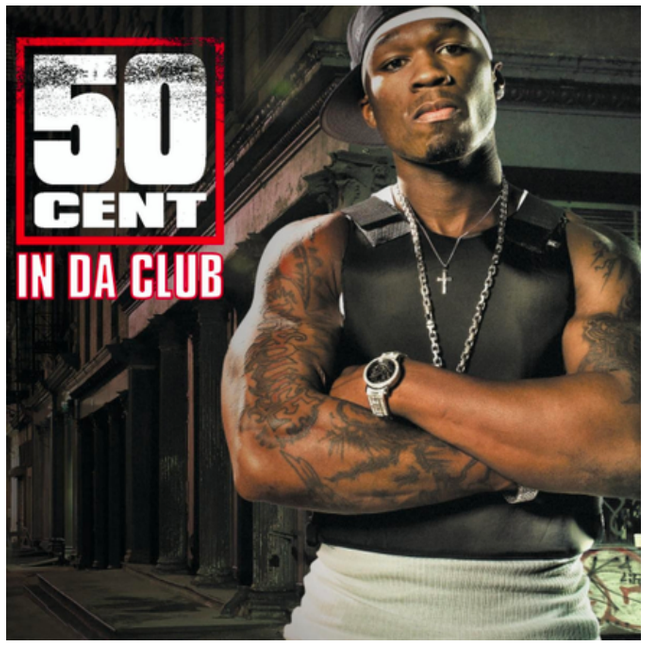
- 1% of your grade per response; graded as 1 or 0 . Can resubmit to get up to 5%

- **Big Goal**: help you familiarize yourself with important topics you want to know about, concepts, and style from important NLP works

Course Project

- Put your passion into practice with a short-term research project
- You are encouraged to choose your own topics, though we'll suggest some. **No Kaggle data though**.
- The project has a
  - Proposal (4%)
  - Halfway update (4%)
  - Poster Presentation (5%)
  - Final Report (20%)
- The 8-page report will use a standardized template and should be a complete experiment and analysis — like a workshop paper.

Course projects are lots of fun



“Go Shorty, it's your birthday
We gonna party like it's your birthday
And we gonna sip Bacardi like it's your birthday

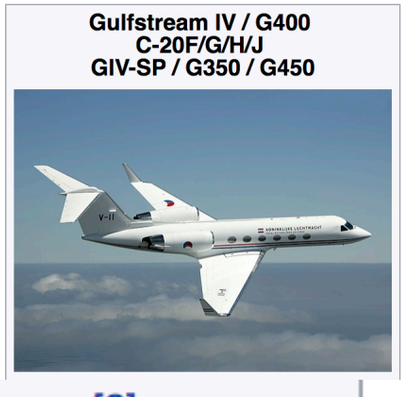
Bacardi Silver Rum 750ml

$11.99 from ShopSk

Bacardi Silver Rum. 750ML, Puerto

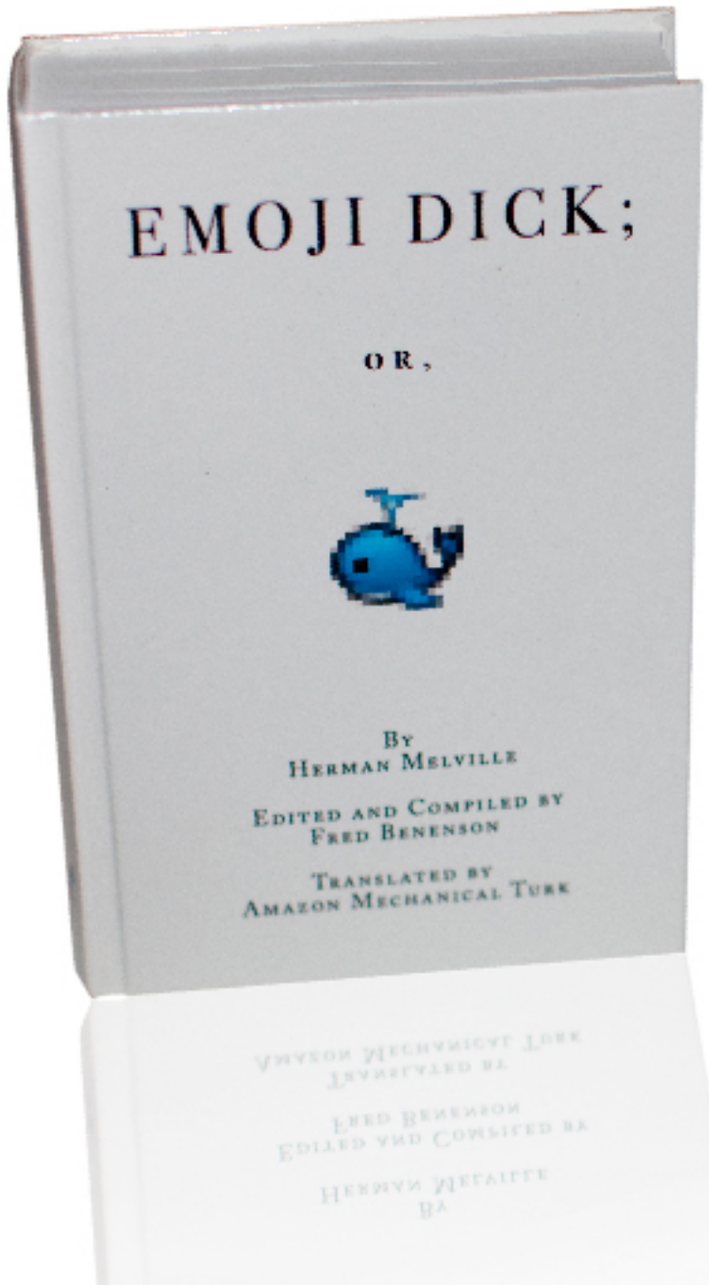
Bacardi · 750 ml

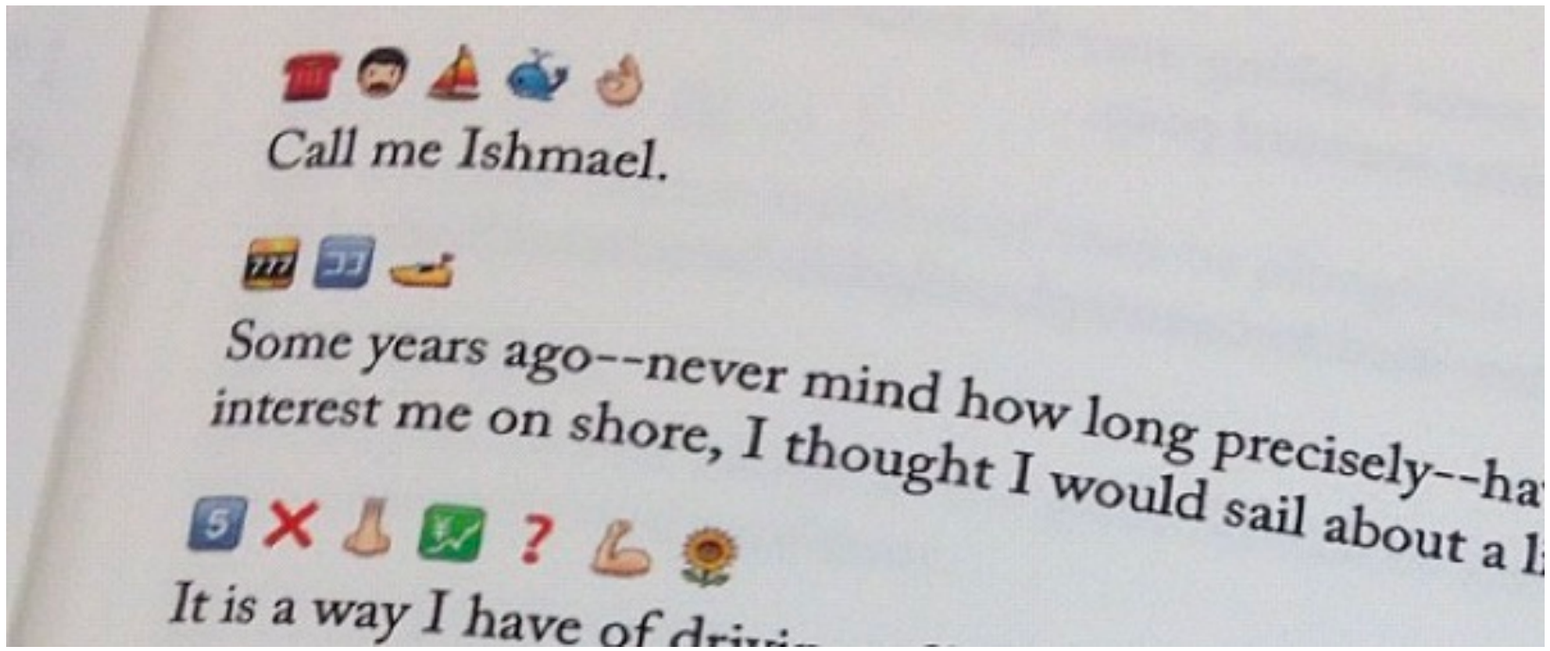
You in a Lexus, I'm Gulf Stream 4
Up in the sky, on a gulf stream to
You want beef? We'll start a Gul



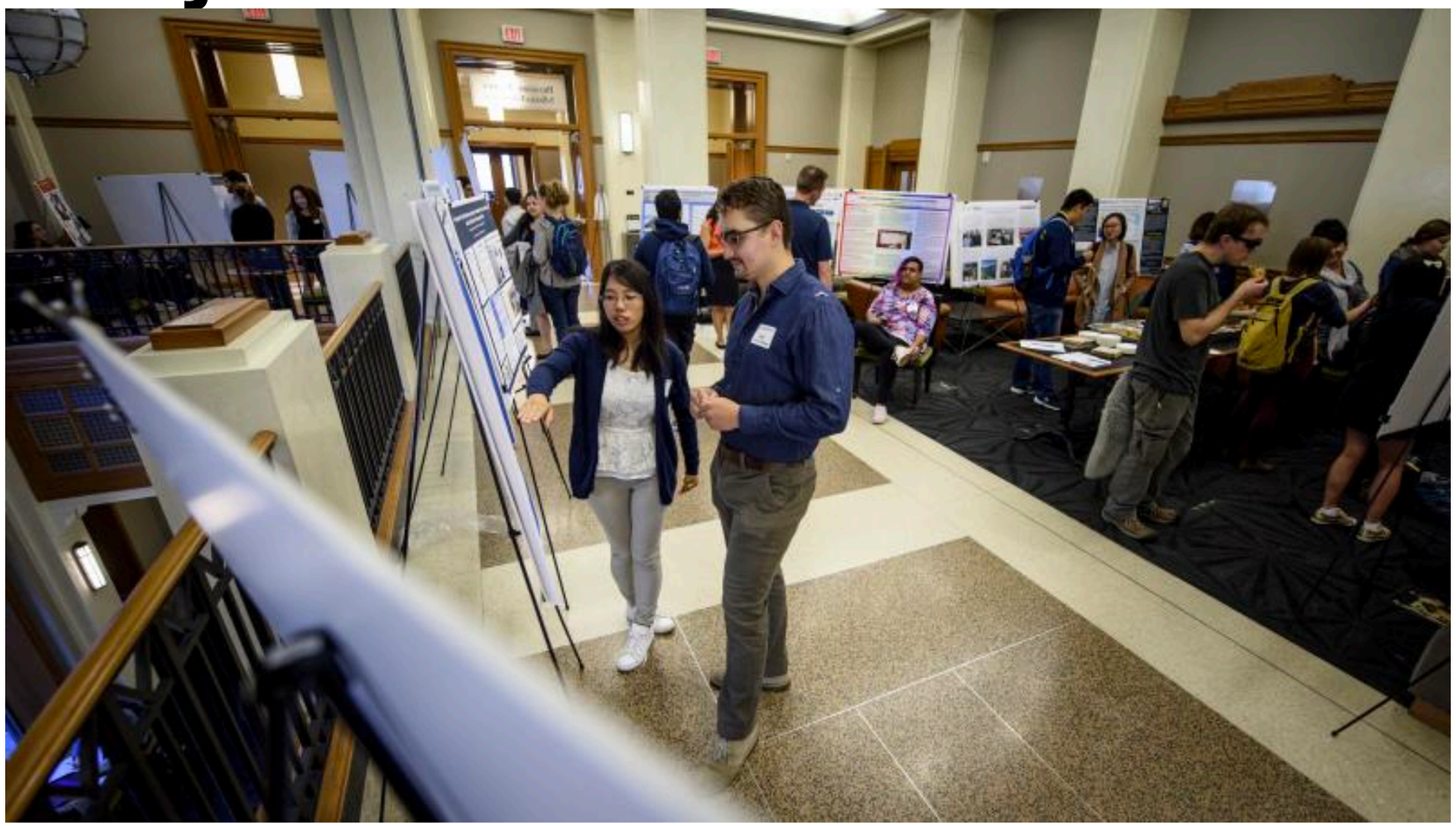
Number built	900+ [1]
Unit cost	GIV: US$36 million (1996)[2] G350: US$34.9 million (2012)[3] G450: US$41 million (2015)[4]

Course projects are lots of fun





Project Presentations



This year: at the UMSI Winter Expo

- Now at Palmer Commons — more space!
- Organized by topic!
- More people!

Exam

- A comprehensive “Final” Midterm
- Take-home style; 24 hours to finish
  - ~2 hours if you are prepared
  - *N* hours if you didn’t, where $0<N<24$
- Week 13 but you get to pick the day of the week

This class is a lot of work

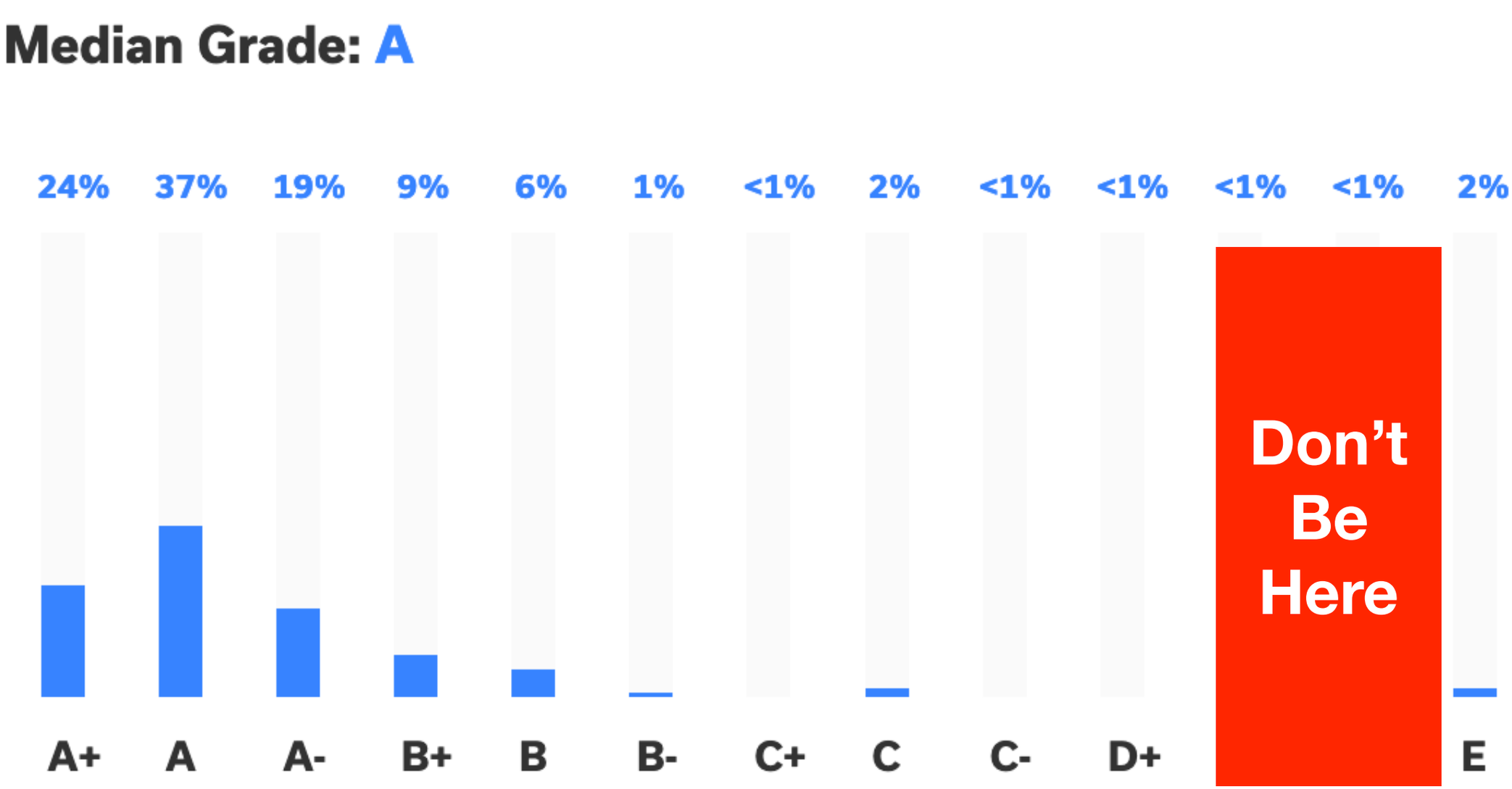
- It's basically a constant amount of moderate work that requires you to do a lot of software development
- 5 homeworks + a Midterm + a Project 😨
- …and don't forget the responses! 😱

From the syllabus…

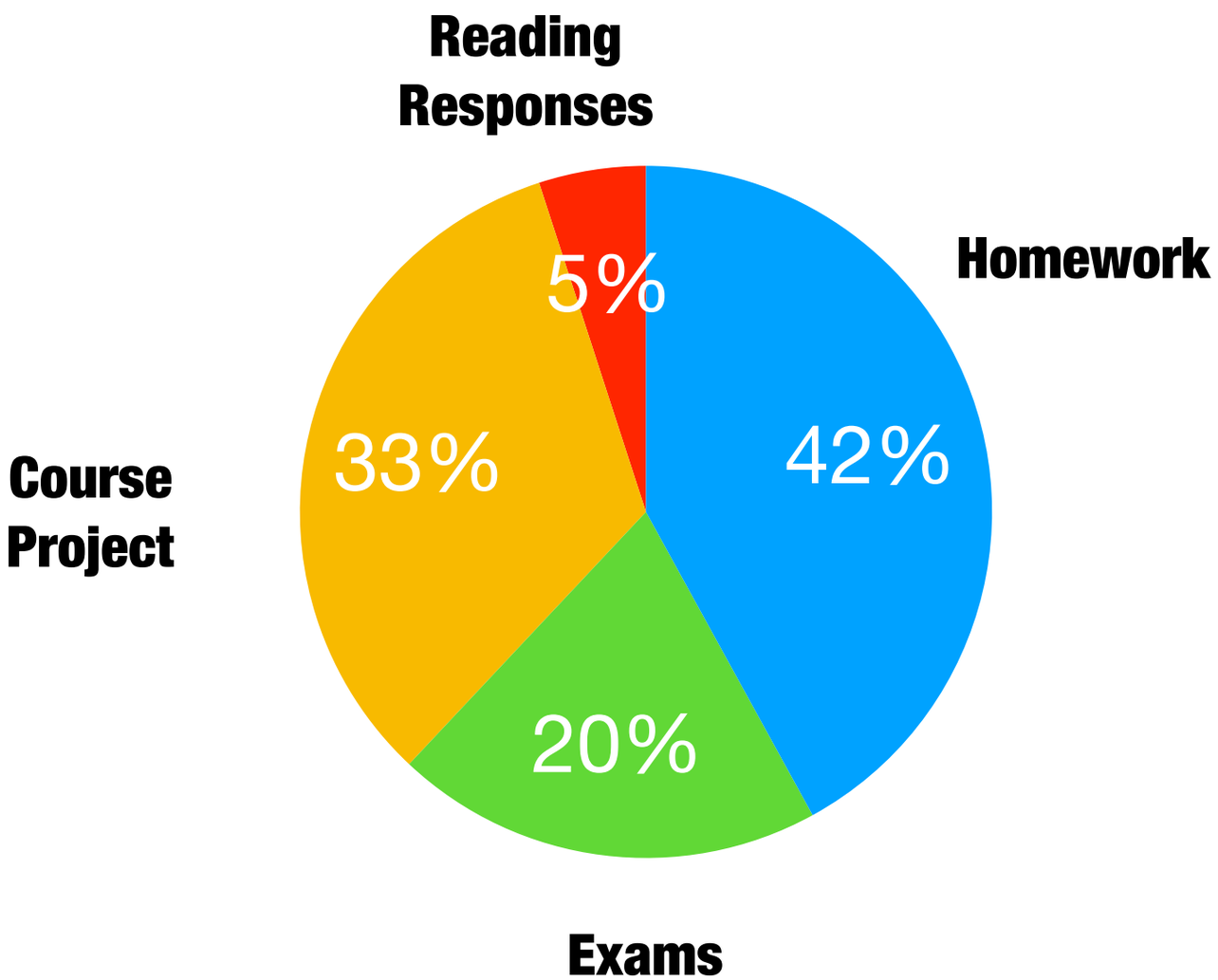
Week	Assigned	Due
1	HW0	
2	HW1	
3		HW0
4	HW2	HW1
5		Project Proposal
6		
7	HW3	HW2
8		
9	Relaxing (Spring Break)	Relaxing
10	HW4	HW3
11		Project Update
12	HW5	HW4
13		Midterm
14		Project Presentation
15		HW5
Finals Week		Project Report

… But most students do really well

SI630 W18 Grade distribution



Grading



This class offers special support to students who feel they need more guidance

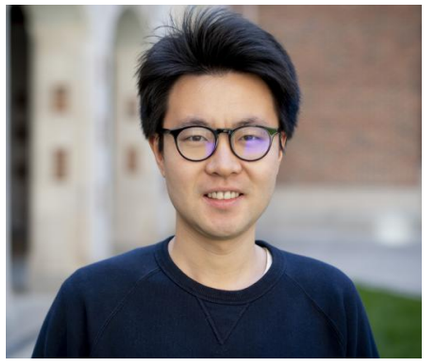
- New optional section: SI 511-630 (for 1 credit!) Fridays 10-11:30 (~1 hour though) @ 2245 NQ
- Covers programming topics related to the course content/homework
  - Walk through stuff on the whiteboard
- If you're nervous about your programming or math skills, this section can help
- You get 1 credit for taking this: 630 becomes a 4-credit course
- ~10% of students will need this. Maybe another 30% could benefit.

And we've made more improvements this year!

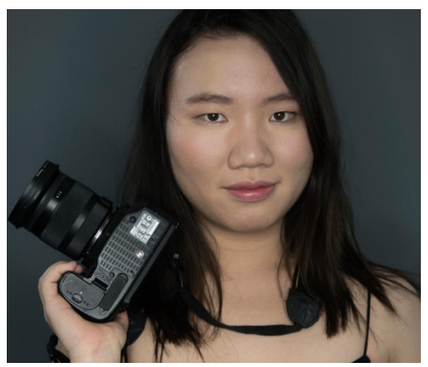
- Two GSIs! Twice the fun! Twice the help!
- Improved homeworks! More fun! Much Learning!
- Better project guidance! Very Data! Such research!
- **MORE DEEP LEARNING! THE DEEPEST OF LEARNINGS!**

Getting help from your teaching team

me
Office Hours:
Tues 1-2 pm by GCal appointment
NQ 3375



Jiaqi Ma
Office Hours:
Tuesdays 9:15-11:15
NQ 1282



Yulin Yu
Office Hours:
Friday 3:15-5:15 pm
NQ 1286

All course communication is through Piazza

Piazza Policy

- Everyone has lots of questions so posting to piazza helps us all
- Students are welcomed to jump in and respond!
- No code specifics in posts
- We'll respond ASAP but only guarantee 24h max response time weekdays and 48h on weekends

Late Submission Policy

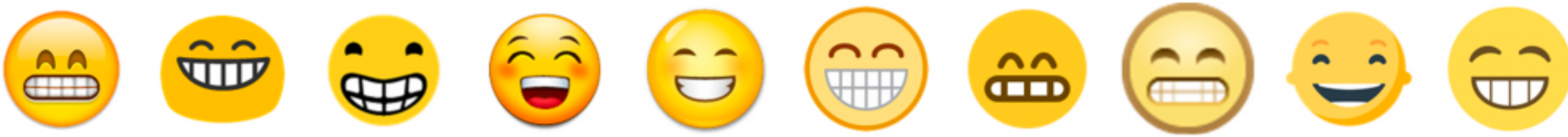
- Three late days total:
  - 1 minute past deadline = 1 late day used

- If you have known issues (interviews, conference, etc.) let us know *at least* 24 hours in advance and we can work something out

Academic Integrity

- TL;DR: Don’t copy anyone else’s work ever.
    - This includes GitHub, StackOverflow, Kaggle, Medium articles, etc…
- If you’re stressed or running behind or whatever, come talk to us first
- We have lots of methods to find cheating—and these have worked unfortunately well 🥺😭
- Every case gets sent to Office of Academic Integrity

What questions do you have?



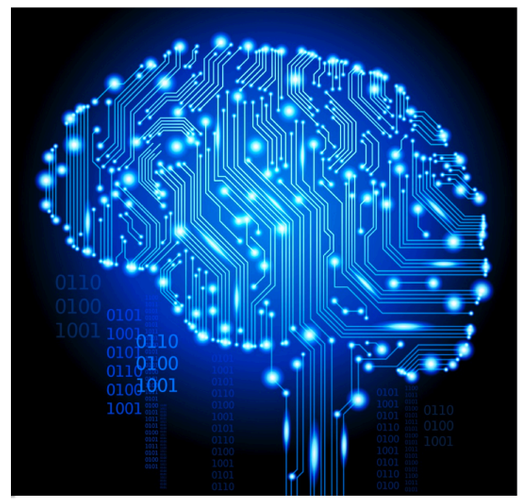
Apple	Google	Microsoft	Samsung	LG	HTC	Twitter	Facebook	Mozilla	Emoji One

Regular Expressions

ELIZA

ELIZA: Please tell me what's been bothering you.
YOU: parents are all alike
ELIZA: Could there really be some connection ?
YOU: they're always bugging me about something
ELIZA: What incident are you thinking of ?
YOU: well, they made me talk to you
ELIZA: What are your feelings now ?
YOU: they say I'm acting depressed
ELIZA: Do you think coming here will help you not to be depressed ?

Is this advanced chatbot technology?



http://www.masswerk.at/elizabot/

What's ELIZA doing?

- I feel Y -> How often do you feel Y ?
- I want Y -> Suppose you got Y soon
- If Y -> Do you think that it's likely that Y ?
- Other tricks
  - Convert "my" to "your" in reply (and other pronouns)
  - Randomly produce a change of subject if no rule matches: "tell me about your mother"

How is ELIZA making the matches?

- ELIZA is about finding patterns
- But users can type many different things
- We need a system for expressing many general patterns!
  - **Regular Expressions!**

Why Regular Expressions in NLP?

- Searching for linguistic phenomena (does eat ever take the object “loss”)?

- Creating features for supervised algorithms

- Useful for morphology

- Thinking about regular expressions (nice tool) will help you think about the structure of language

Regular Expressions: Disjunctions

- Any string is a regular expression that's an exact match, e.g., "woodchuck"

- What if we wanted to find all of these words?

Woodchuck
woodchuck

- The [ ] lets us specify multiple matching characters:

"[wW]oodchuck"

- We can specify whole *ranges* of matches using the '-' character: "[a-z]" matches a, b, c …. z.

Regular Expressions: Negation in Disjunctions

- Sometimes we want to *not* match. We use the '^' at the start of a [ ] expression to anti-match

Pattern	Matches	Example
[^A-Z]	Not an upper-case	CCL is far
[^Ss]	Neither 's' nor 'S'	Sl is far now
[^o^]	Neither 'o' nor '^'	everyone is here
a^b	The string "a^b"	logic uses a^b

Regular Expressions: More Disjunction

- Pipes '|' are another way of specifying disjunctions

Pattern	Matches
soda|pop	either "soda" or "pop"
a|b|c	[abc]
[sS]oda|[pP]op	soda, Soda, pop, Pop
[s|p]	any of the characters 's', 'p' or '|'

Demo time!

https://pythex.org/

Regular Expressions: Operators

Operator	Pattern	Matches	Example
?	colou?r	Optional previous character	color, colour
*	cooo*l!	zero or more of previous character	cool, coooool
+	co[oO]+l	one or more of previous character	cool, coooool, cooOooOool
.	beg.n	any character	begin, began, begun, beg1n

Regular Expressions: More Operators

Operator	Matches	Pattern	Example
{N}	exactly N characters	co[o]{2}l	cool, coool, cooooool
{M,N}	at least M characters and at most N	co[o]{1,3}l	cool, cooooool

Regular Expressions: Character classes

Anchor	Matches	Pattern	Example
\d	digits	Jan \d+	Jan 3
\D	non-digits	\D+	Jan 3
\s	whitespace	Feb\s+\d	Feb 1
\S	non-whitespace	\S+	F9012)!(@
\b	start or end of a word (a word break)	\bfoo\b	foo food

Many regular expression libraries have even more character classes that can make your life easier!

Regular Expressions: Anchors ^ $

Anchor	Matches	Pattern	Example
^	Beginning of string	^[A-Z]	Color
		^[^a-z]+	COLor
$	End of string	\.$	the end.
		.$	an end

Regular Expressions: Groups

- Groups allow us to reference parts of matching
- We create groups by putting ( ) around a subexpression, e.g., “My name is ([a-zA-Z]+)”
- We can reference them for string replacement, which is “\1” for the first group, “\2” for the second, and so on
    - Group references can vary by programming language so check

Demo time!

https://pythex.org/

Back to ELIZA: How did things get matched?

```
ELIZA: Please tell me what's been bothering you.
YOU:   parents are all alike
ELIZA: Could there really be some connection ?
YOU:   they're always bugging me about something
ELIZA: What incident are you thinking of ?
YOU:   well, they made me talk to you
ELIZA: What are your feelings now ?
YOU:   they say I'm acting depressed
ELIZA: Do you think coming here will help you not to be depressed ?
```

Building a mini Eliza

```
def main():
    print "Hello. How are you feeling today?"

    while True:
        statement = raw_input("> ")
        print analyze(statement)

        if statement == "quit":
            break
```

```
def analyze(statement):
    statement = statement.rstrip(".!") # remove punch
    for pattern, responses in psychobabble:
        match = re.match(pattern, statement)
        if match:
            response = random.choice(responses)
            return response.format(
                *[reflect(g) for g in match.groups()])
```

Generating ELIZA Responses

```
psychobabble = [
    [r'I need (.*)',
     ["Why do you need {0}?",
      "Would it really help you to get {0}?",
      "Are you sure you need {0}?"]],

    [r'(.*) friend (.*)',
     ["Tell me more about your friends.",
      "When you think of a friend, what comes to mind?",
      "Why don't you tell me about a childhood friend?"]],

    [r'Why can\'?t I ([^\?]*)\??',
     ["Do you think you should be able to {0}?",
      "If you could {0}, what would you do?",
      "I don't know -- why can't you {0}?",
      "Have you really tried?"]],
]
```

What the “patient” says

What ELIZA responds with

https://www.smallsurething.com/implementing-the-famous-eliza-chatbot-in-python/

Building a mini Eliza

```
def main():
    print "Hello. How are you feeling today?"

    while True:
        statement = raw_input("> ")
        print analyze(statement)

        if statement == "quit":
            break
```

```
def analyze(statement):
    statement = statement.rstrip(“.!") # remove punch
    for pattern, responses in psychobabble:
        match = re.match(pattern, statement)
        if match:
            response = random.choice(responses)
            return response.format(
                *[reflect(g) for g in match.groups()])
```

```
def reflect(fragment):
    tokens = fragment.lower().split()
    for i, token in enumerate(tokens):
        if token in reflections:
            tokens[i] = reflections[token]
    return ' '.join(tokens)

reflections = {
    "am": "are",
    "was": "were",
    "i": "you",
    "i'd": "you would",
    "i've": "you have",
    "i'll": "you will",
    "my": "your",
    "are": "am",
    "you've": "I have",
    "you'll": "I will",
    "your": "my",
    "yours": "mine",
    "you": "me",
    "me": "you"
}
```

Homework 0: Warming up to text processing with Regular Expressions

- Extract emails from web pages — even those pesky "jurgens [at] umich [dot] edu" ones
- Homework goals:
  - Remember how to program again
  - Learn simple regular expression patterns
  - Get started working with text input and output
- Live on Canvas now. Technically due in two weeks (1/23)
  - Most should be able to finish in ~25 lines of code with about ~1 hour effort
- Submit extractions using Kaggle to see score!

Text Classification

Classification

Classification defines a mapping *h*

- from input data x (drawn from instance space $\mathcal{X}$)
- to a label (or labels) y from some enumerable output space $\mathcal{Y}$

$\mathcal{X}$ = set of all movie reviews

$\mathcal{Y}$ = {good, bad}

x = a single document
y = bad

Some people will also call this mapping a function

Classification

$h(x) = y$

$h(\text{“this movie is bad”}) = \text{negative}$

Classification

Let h(x) be the “true” mapping. We never know it.

How do we find the best $\hat{h}(x)$ to approximate it?

One option: rule based

if x has the word “bad” in it :
$\hat{h}(x)$ = bad

Classification

Second Option: **Supervised learning**

Given training data in the form of
<x, y> pairs, learn $\hat{h}(x)$

Text categorization problems are everywhere

task	x	y
language ID	text	{english, mandarin, greek, …}
spam classification	email	{spam, not spam}
authorship attribution	text	{jk rowling, james joyce, …}
genre classification	novel	{detective, romance, gothic, …}
sentiment analysis	text	{postive, negative, neutral, mixed}

Sentiment analysis

- Document-level SA: is the entire text positive or negative (or both/neither) with respect to an implicit target?

- Movie reviews [Pang et al. 2002, Turney 2002]

Training data

positive

"… is a film which still causes real, not figurative, chills to run along my spine, and it is certainly the bravest and most ambitious fruit of Coppola's genius"

Roger Ebert, Apocalypse Now

- "I hated this movie. Hated hated hated hated hated this movie. Hated it. Hated every simpering stupid vacant audience-insulting moment of it. Hated the sensibility that thought anyone would like it."

Roger Ebert, North

negative

- Implicit signal: star ratings
- Either treat as ordinal regression problem ({1, 2, 3, 4, 5} or binarize the labels into {pos, neg}

I recommend it.
This book introduces readers to important topics in NLP. In places where it needs to go deeper it seems like it compiles information from relevant published papers and provides... Read more
Published 1 year ago by Renat Bekbolatov

It's presented easily and accessibly
It can be dense sometimes, but it's one of the most helpful textbooks I've had for computational linguistics. Read more
Published on April 28, 2015 by Vanessa A.

Five Stars
I love this book.
It was easy to follow and a great read.
Published on December 28, 2014 by Stefan Meiforth Gulbrandsen

Five Stars
I needed the book for my natural language processing class. needless to say, I learnt a lot.
Published on November 27, 2014 by Kamran

Encyclopedic Treatment of NLP
Daniel Jurafsky and James Martin have assembled an incredible mass of information about natural language processing.Foundations of Statistical Natural Language Processing Read more
Published on April 25, 2012 by John M. Ford

Sentiment analysis

- Is the text positive or negative (or both/ neither) with respect to an explicit target within the text?

Feature: **picture**

Positive: 12

- Overall this is a good camera with a really good picture clarity.
- The pictures are absolutely amazing - the camera captures the minutest of details.
- After nearly 800 pictures I have found that this camera takes incredible pictures.

...

Negative: 2

- The pictures come out hazy if your hands shake even for a moment during the entire process of taking a picture.
- Focusing on a display rack about 20 feet away in a brightly lit room during day time, pictures produced by this camera were blurry and in a shade of orange.

Hu and Liu (2004), “Mining and Summarizing Customer Reviews”

Sentiment analysis

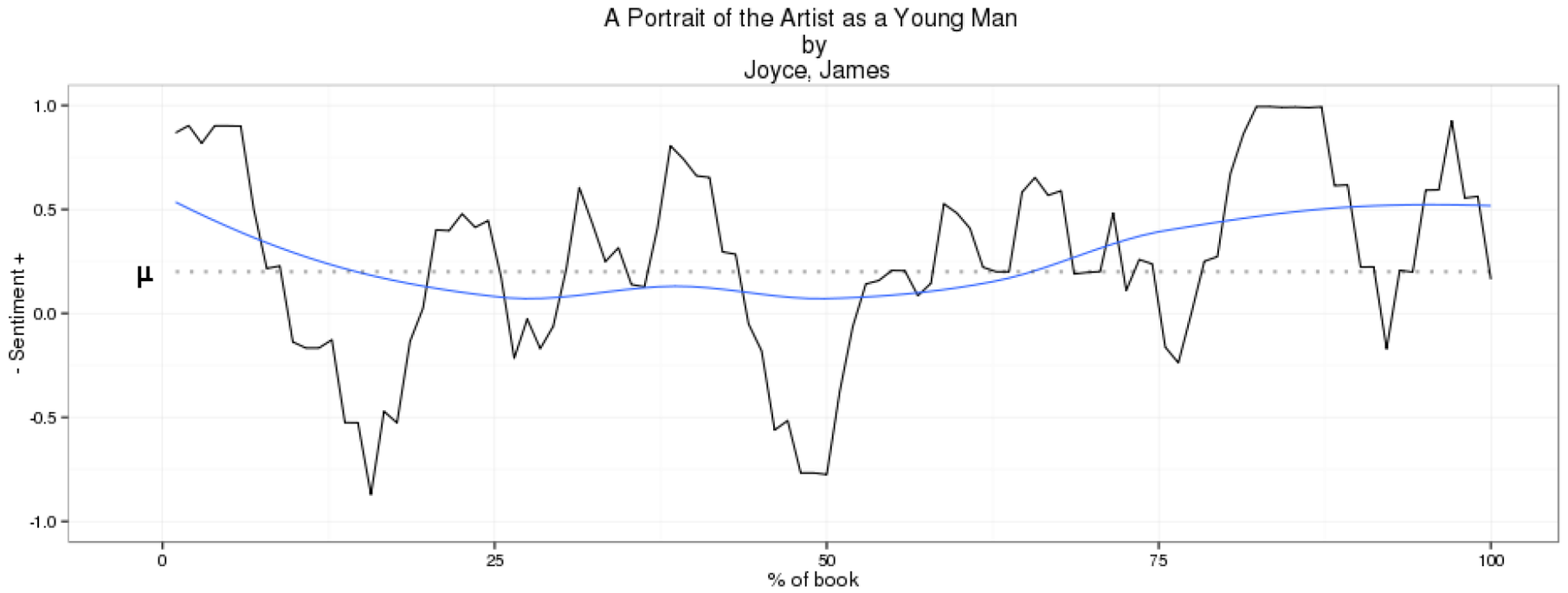
- Political/product opinion mining

Sentiment as tone

- No longer the speaker's attitude with respect to some particular target, but rather the positive/negative tone that is evinced.

Sentiment as tone

“Once upon a time and a very good time it was there was a moocow coming down along the road and this moocow that was coming down along the road met a nicens little boy named baby tuckoo…"



http://www.matthewjockers.net/2014/06/05/a-novel-method-for-detecting-plot/

Sentiment Dictionaries

- MPQA subjectivity lexicon (Wilson et al. 2005) http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

- LIWC (Linguistic Inquiry and Word Count, Pennebaker 2015)

pos	neg
unlimited	lag
prudent	contortions
supurb	fright
closeness	lonely
impeccably	tenuously
fast-paced	plebeian
treat	mortification
destined	outrage
blessing	allegations
steadfastly	disoriented

Why is SA hard?

- Sentiment is a measure of a speaker's private state, which is unobservable.

- Sometimes words are a good indicator of sentence (love, amazing, hate, terrible); many times it requires deep world + contextual knowledge

> "*Valentine's Day* is being marketed as a Date Movie. I think it's more of a First-Date Movie. If your date likes it, do not date that person again. And if you like it, there may not be a second date."
>
> Roger Ebert, Valentine's Day

Classification

Second Option: **Supervised learning**

Given training data in the form of
<x, y> pairs, learn $\hat{h}(x)$

x	y
loved it!	positive
terrible movie	negative
not too shabby	positive

The classification function: $\hat{h}(x)$

- The classification function that we want to learn has two different components:
  - the formal structure of the learning method (what's the relationship between the input and output?) → Naive Bayes, logistic regression, convolutional neural network, etc.
  - the representation of the data

Representation for Sentiment Analysis

- Only positive/negative words in MPQA
- Only words in isolation (bag of words)
- Conjunctions of words (sequential, skip ngrams, other non-linear combinations)
- Higher-order linguistic structure (e.g., syntax)

"… is a film which still causes real, not figurative, chills to run along my spine, and it is certainly the bravest and most ambitious fruit of Coppola's genius"

Roger Ebert, Apocalypse Now

"I hated this movie. Hated hated hated hated hated this movie. Hated it. Hated every simpering stupid vacant audience-insulting moment of it. Hated the sensibility that thought anyone would like it."

Roger Ebert, North

Bag of words

Representation of text only as the counts of words that it contains

	Apocalypse now	North
the	1	1
of	0	0
hate	0	9
genius	1	0
bravest	1	0
stupid	0	1
like	0	1
…		

Naive Bayes

- Given access to <x,y> pairs in training data, we can train a model to estimate the class probabilities for a new review.

- With a bag of words representation (in which each word is independent of the other), we can use Naive Bayes

- Probabilistic model; not as accurate as other models (see next two classes) but fast to train and the foundation for many other probabilistic techniques.

Random variable

- A variable that can take values within a fixed set (discrete) or within some range (continuous).

$$X \in \{1, 2, 3, 4, 5, 6\}$$

$$X \in \{the, a, dog, cat, runs, to, store\}$$

$$P(X = x)$$

Probability that the random variable X takes the value x (e.g., 1)

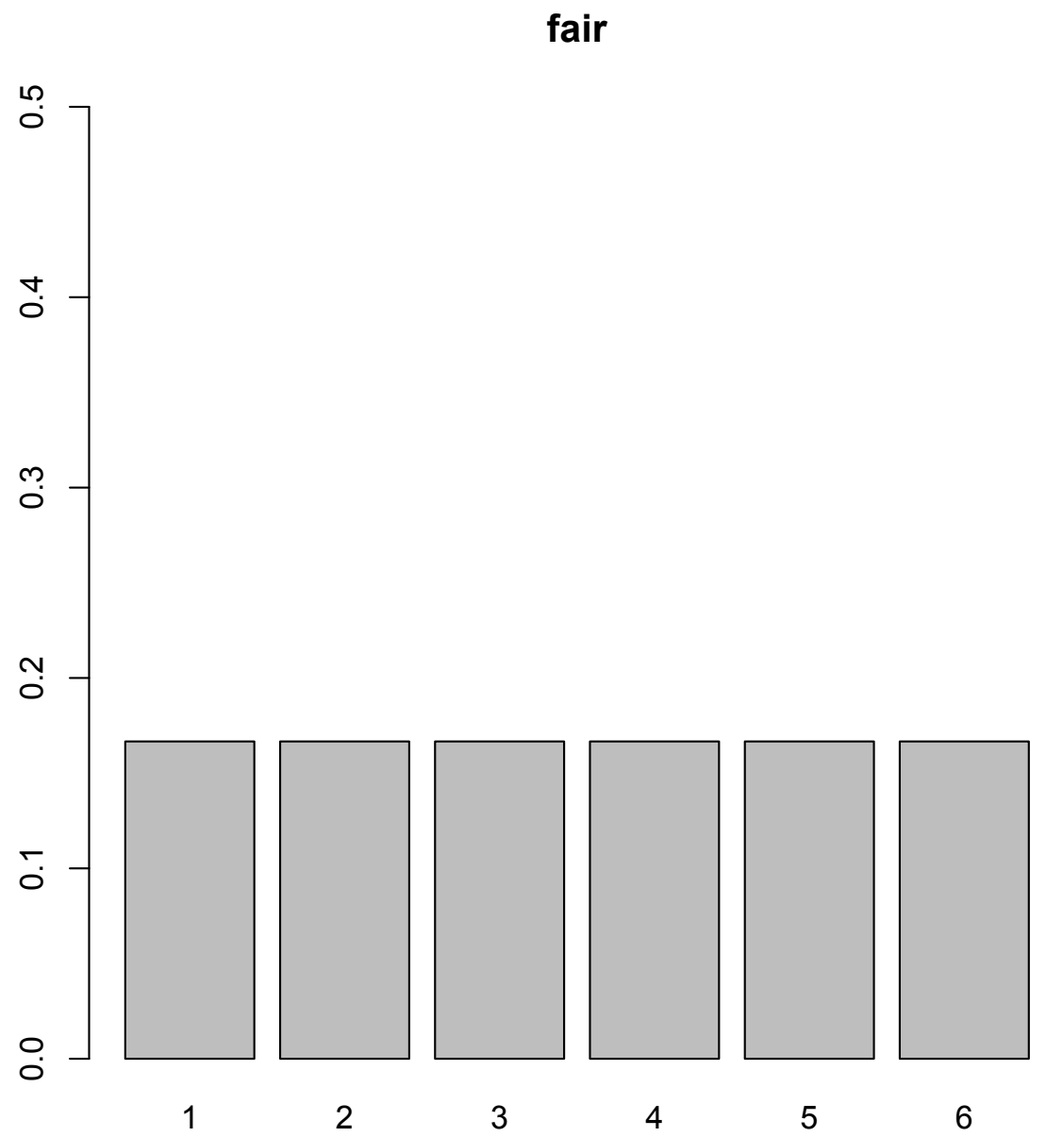
$$X \in \{1, 2, 3, 4, 5, 6\}$$

Two conditions:

1. Between 0 and 1: $0 \leq P(X = x) \leq 1$
2. Sum of all probabilities = 1 $\sum_x P(X = x) = 1$

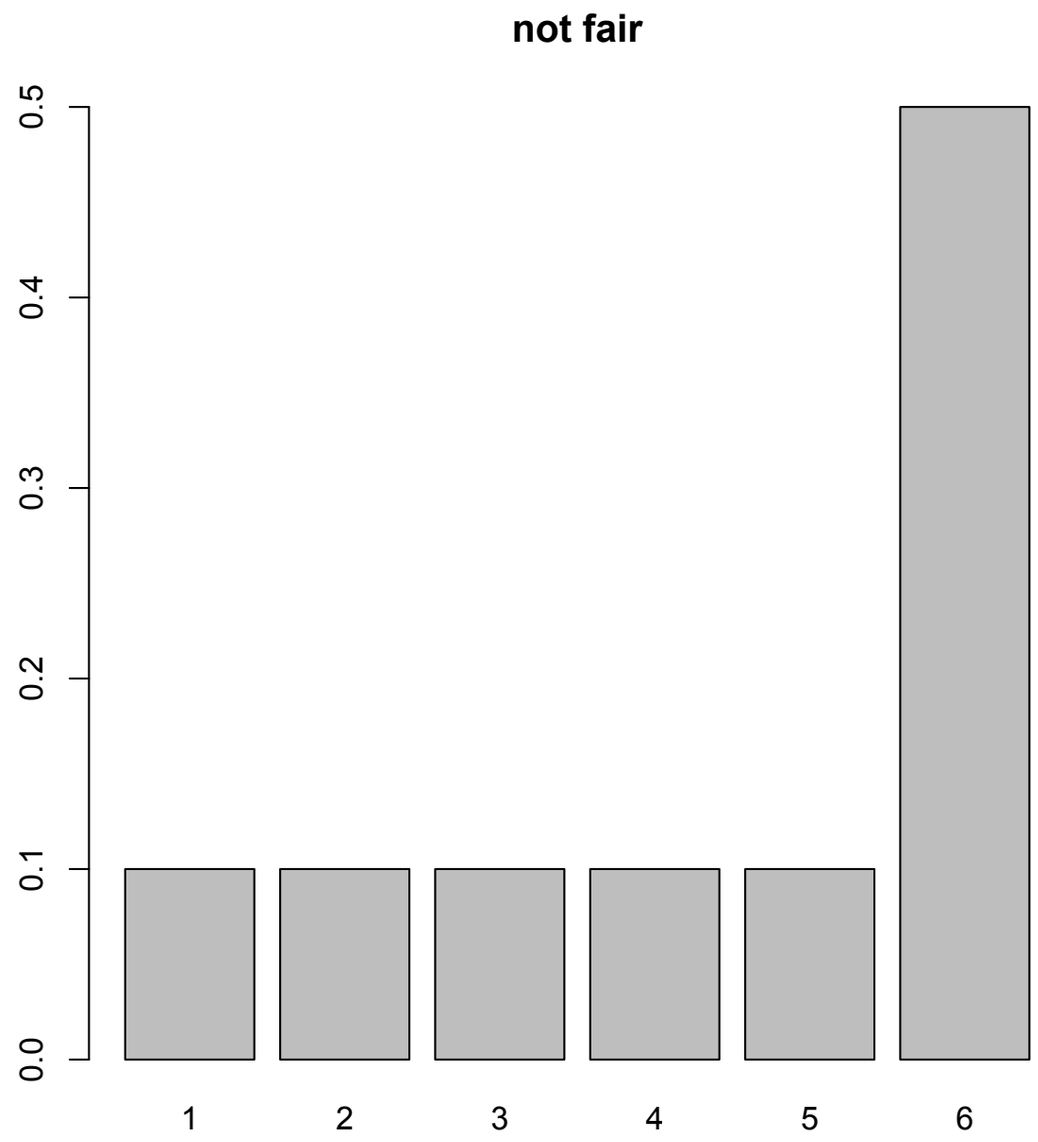
Fair dice

$$X \in \{1, 2, 3, 4, 5, 6\}$$



Weighted dice

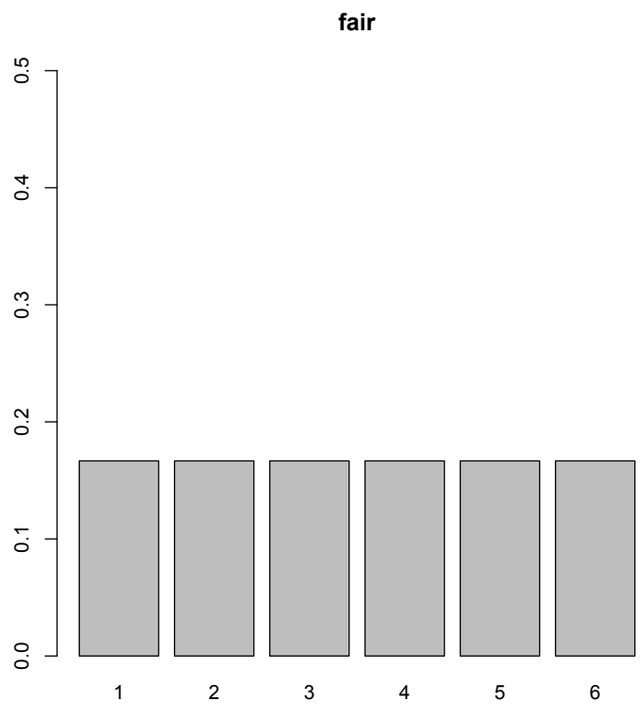
$$X \in \{1, 2, 3, 4, 5, 6\}$$



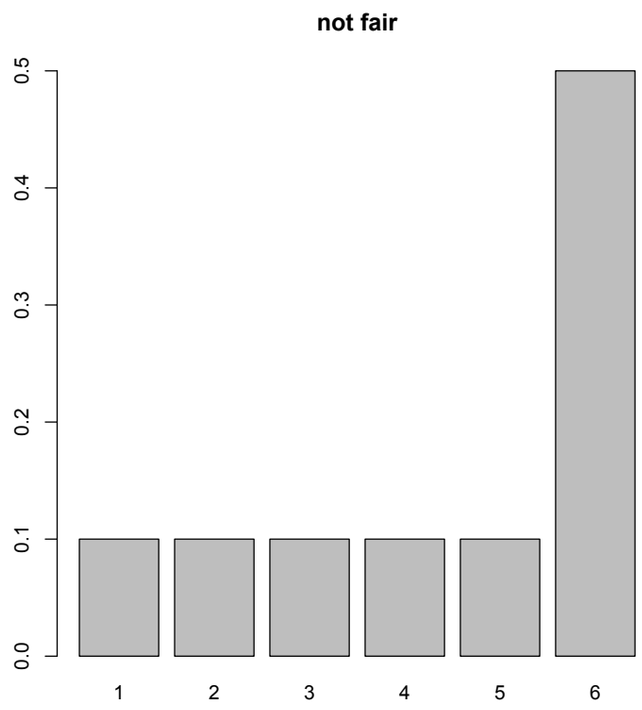
Inference

$$X \in \{1, 2, 3, 4, 5, 6\}$$

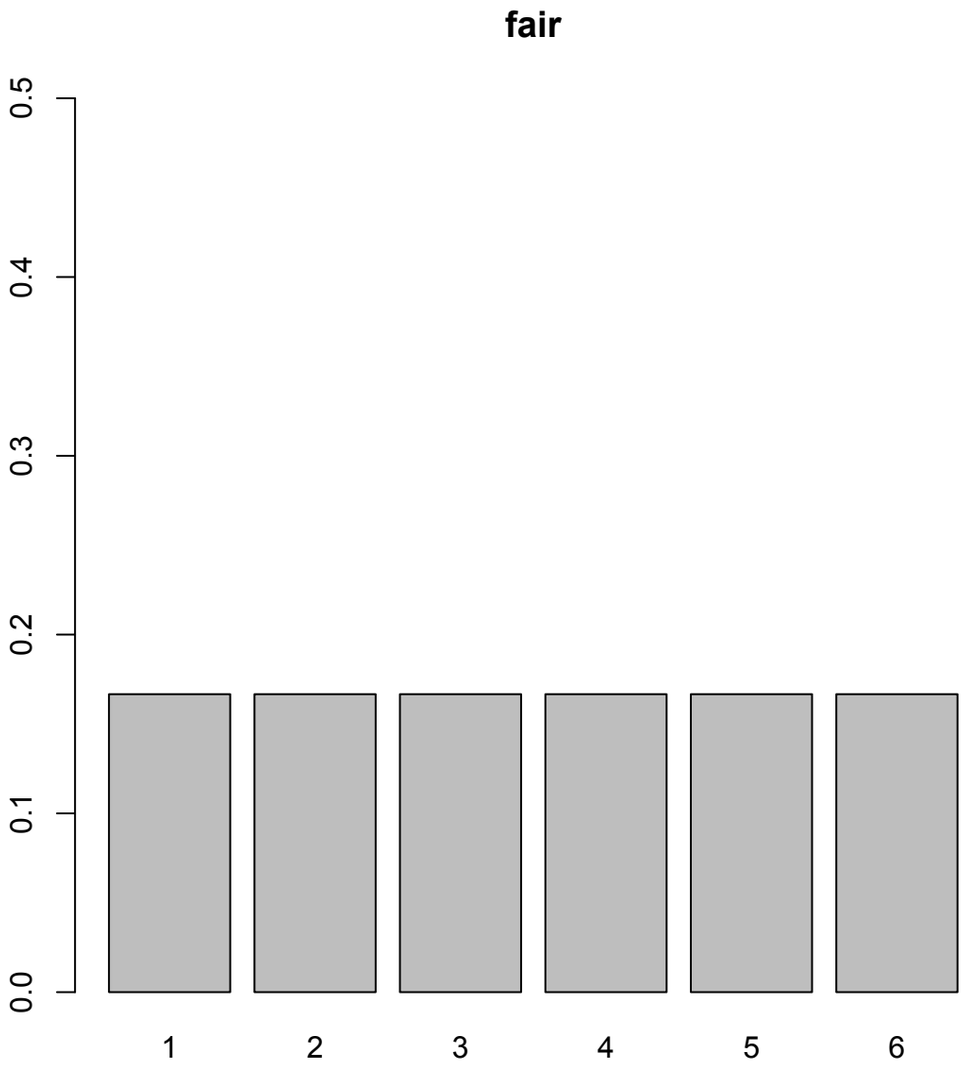
We want to *infer* the probability distribution that generated the data we see.

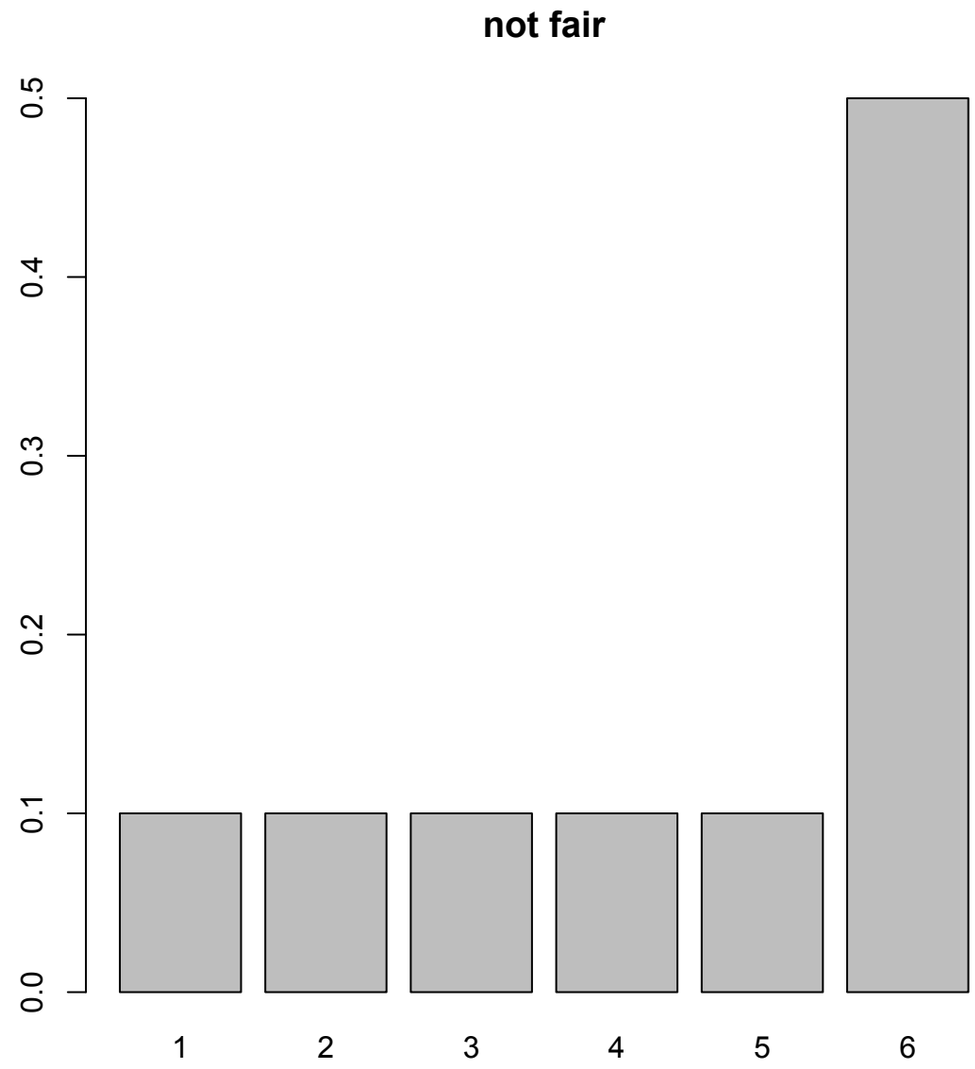


?

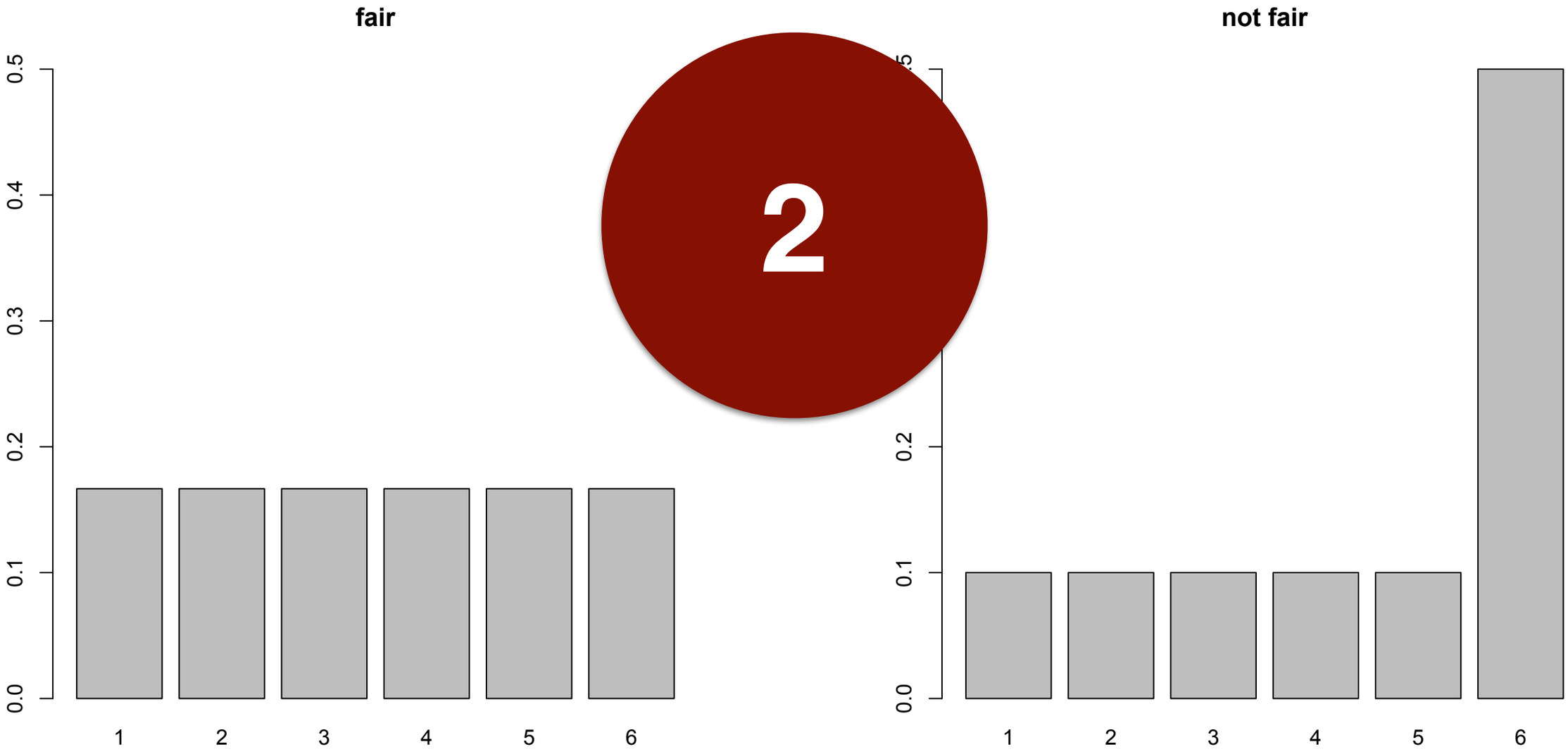


Probability



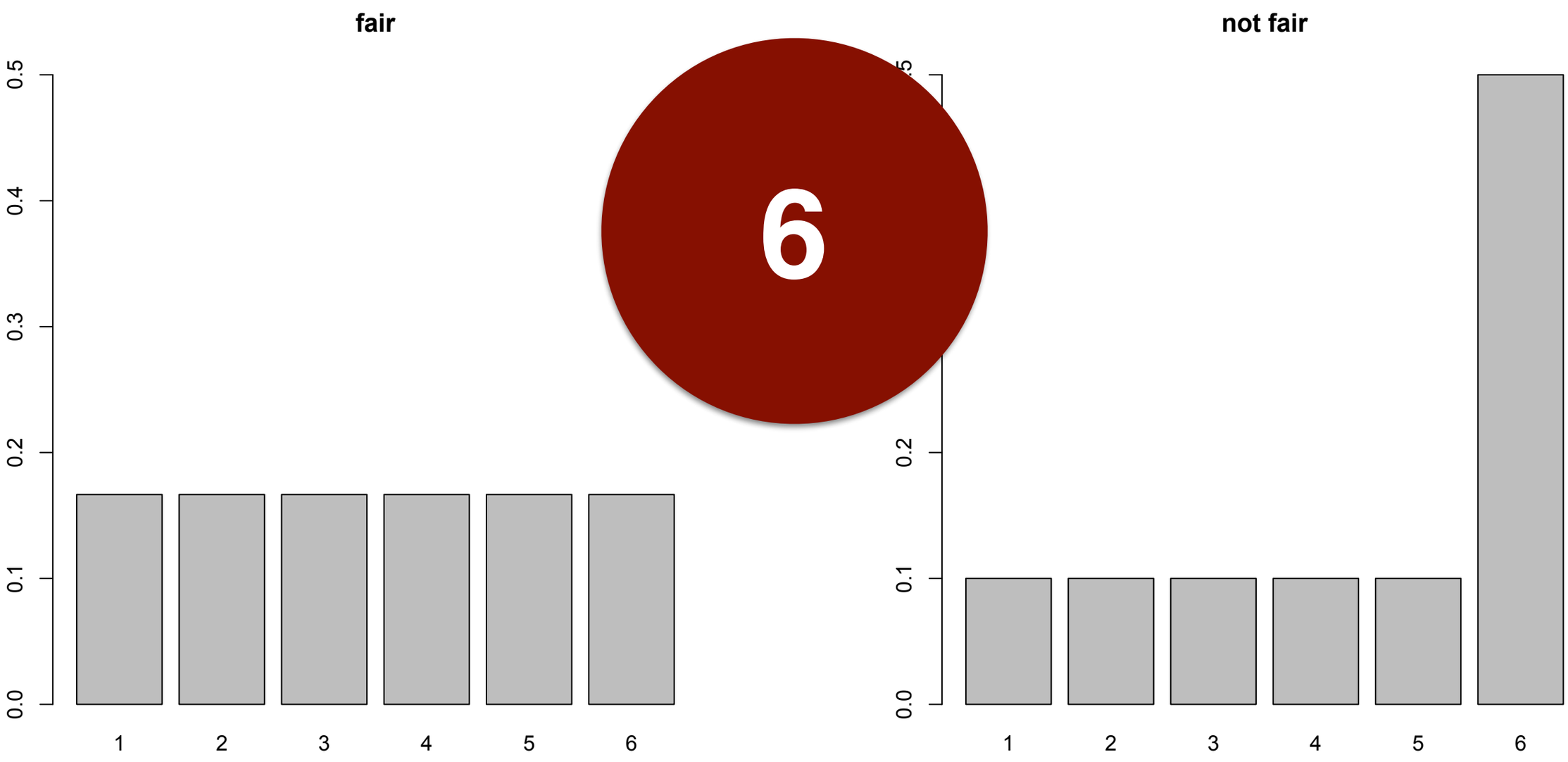


Probability



Probability

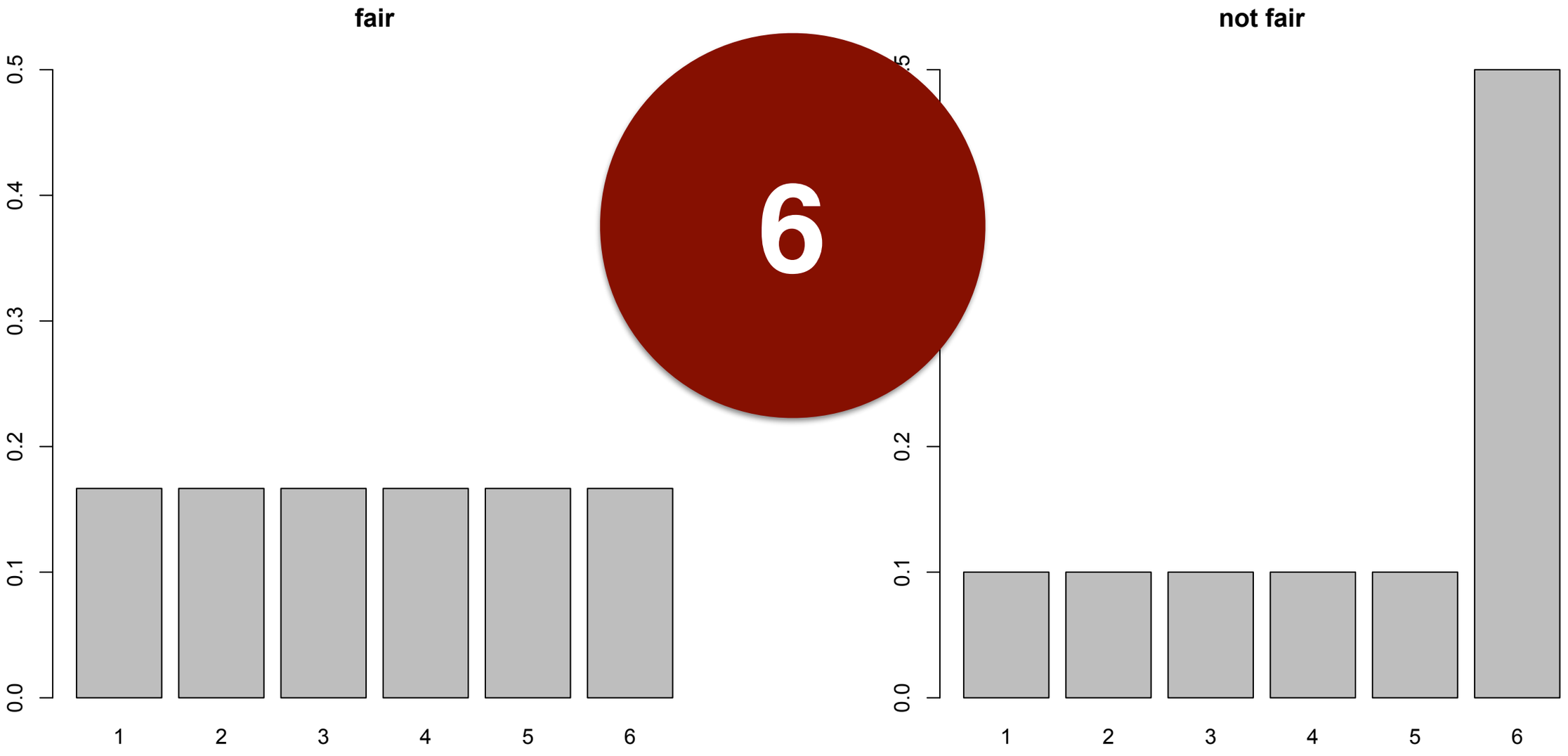
2



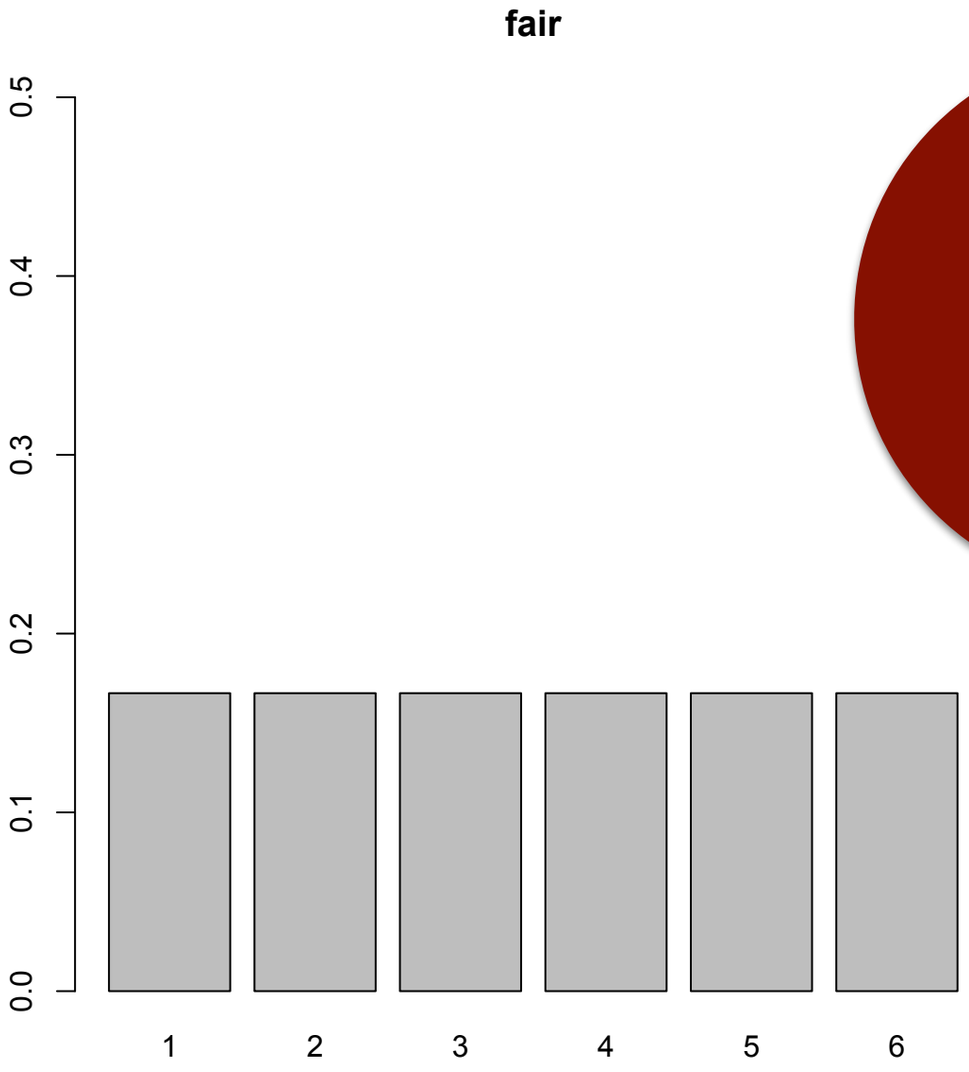
6

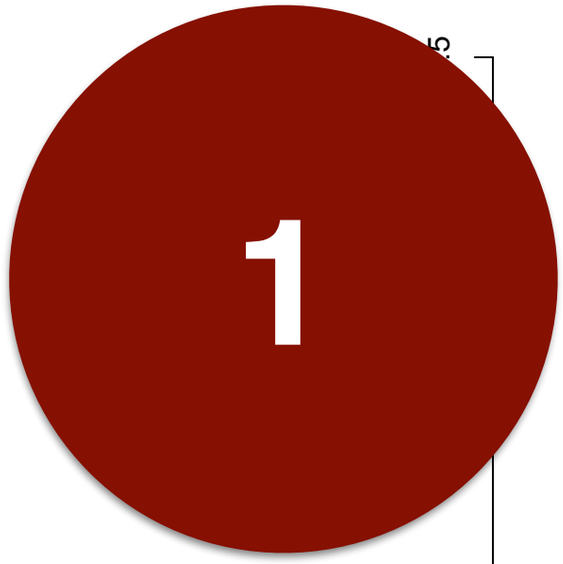
Probability

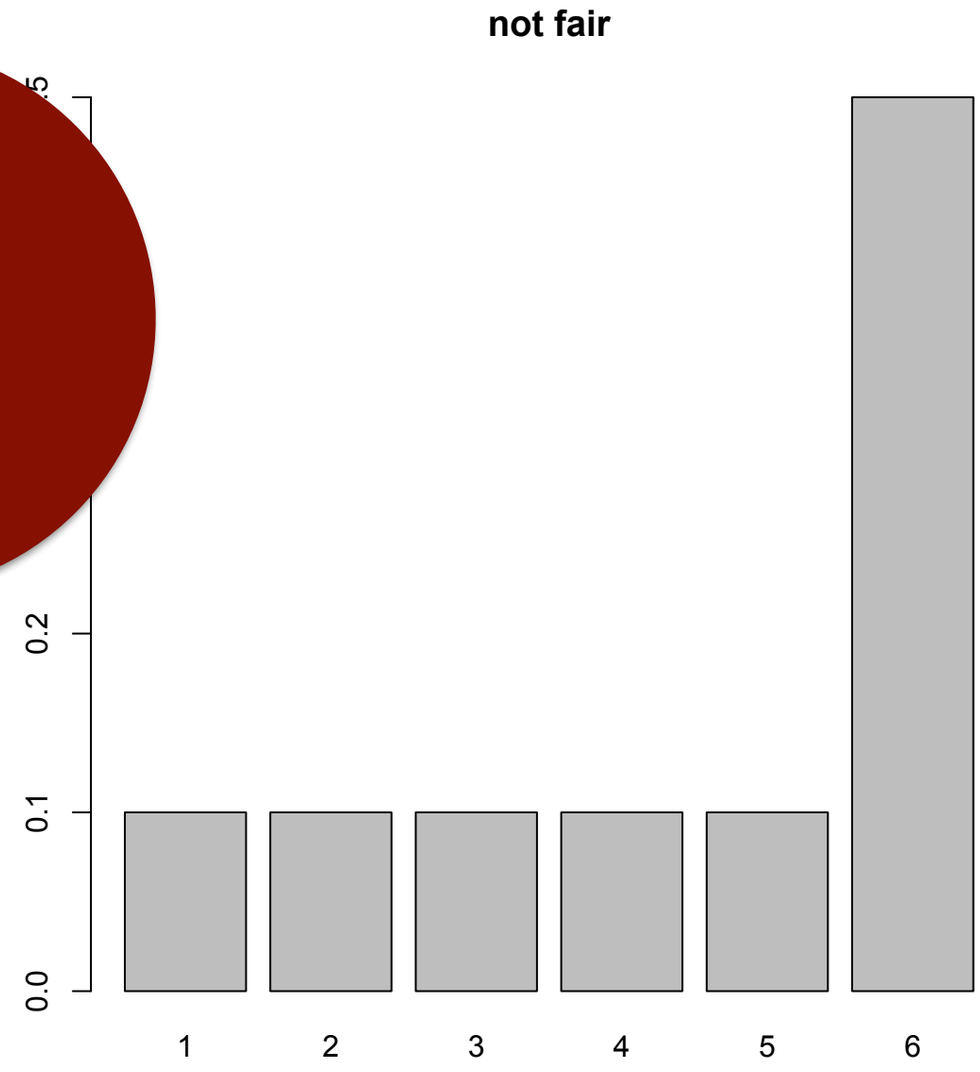
2 6



Probability

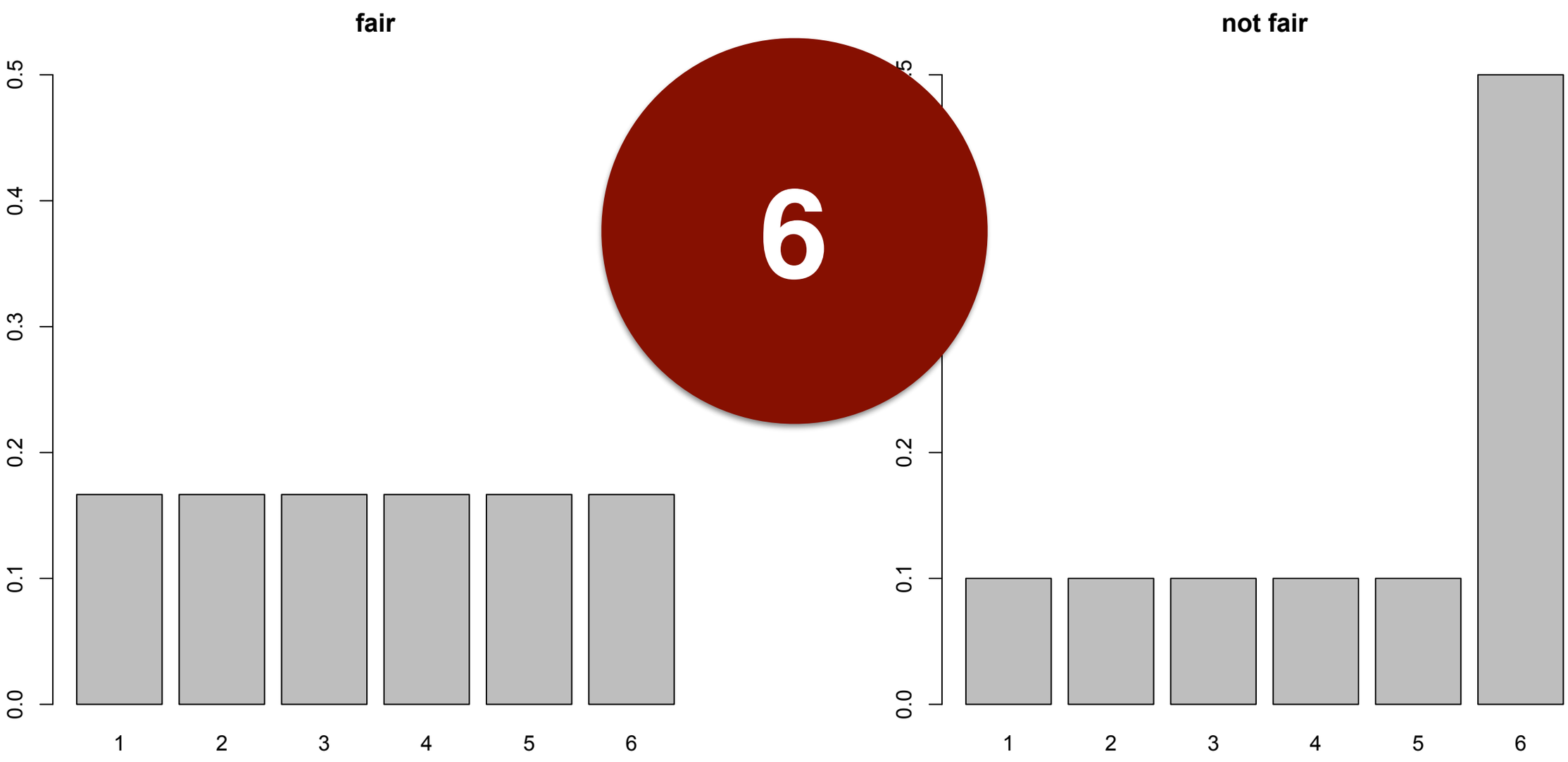






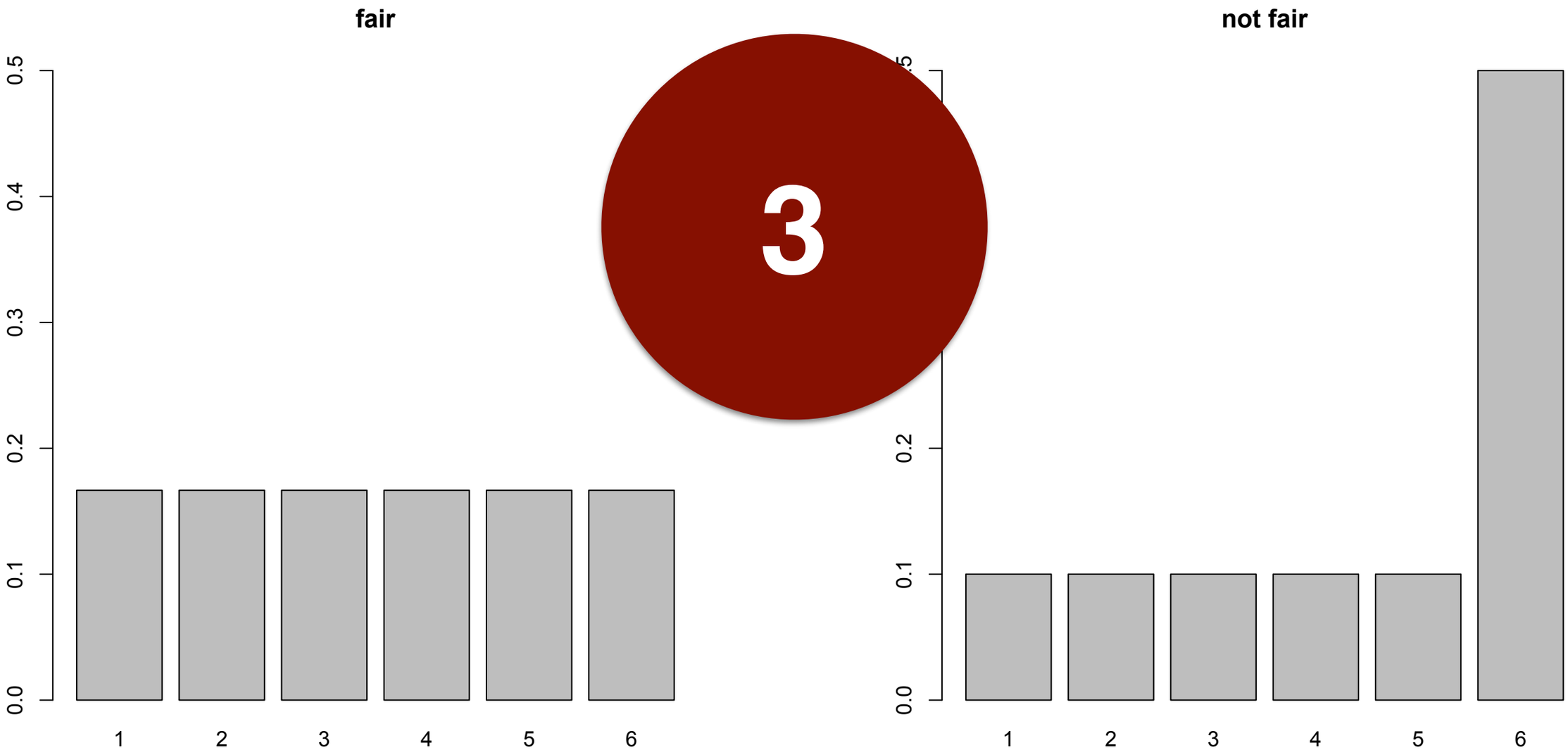
Probability

2 6 6 1



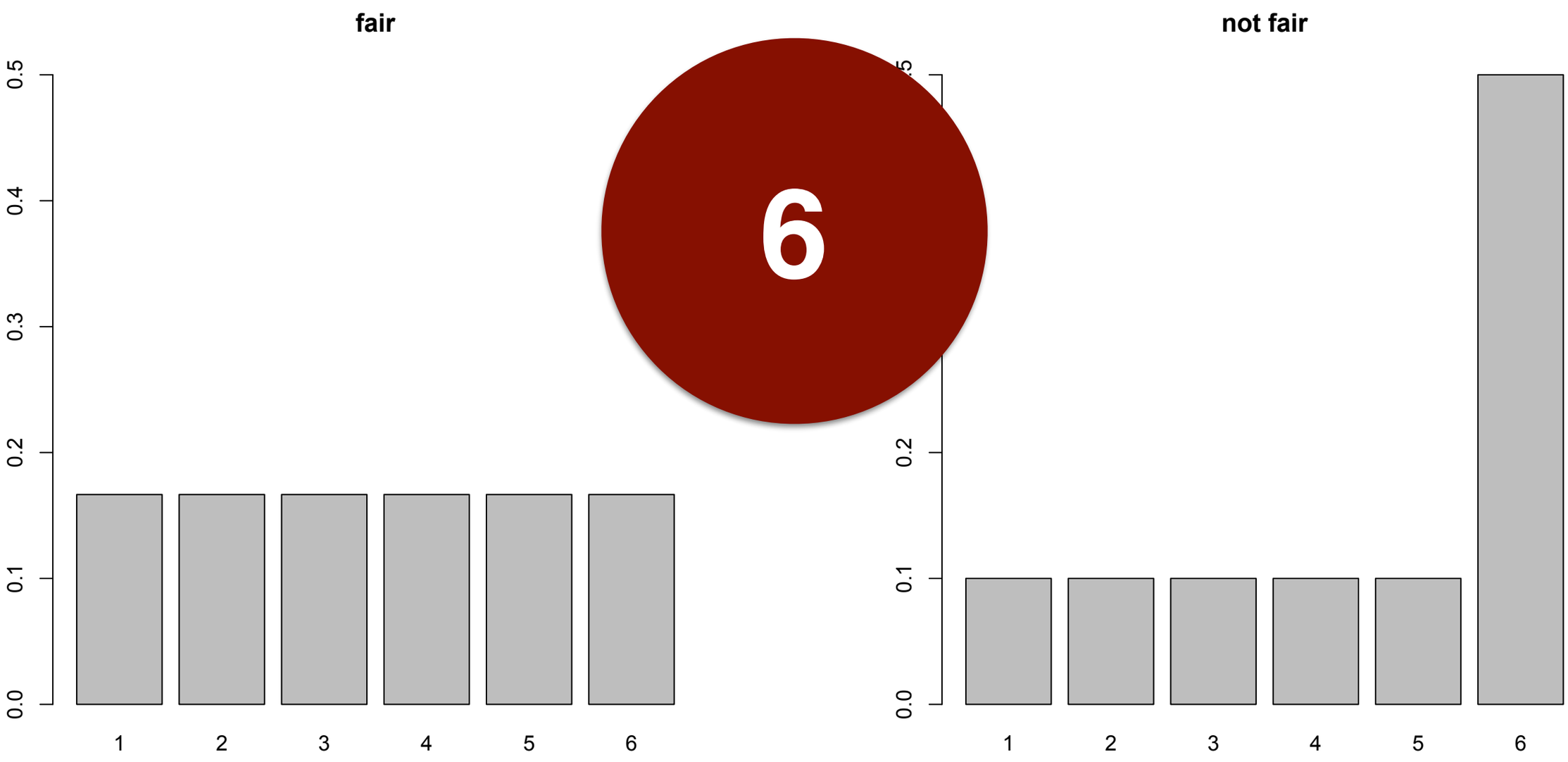
Probability

2 6 6 1 6



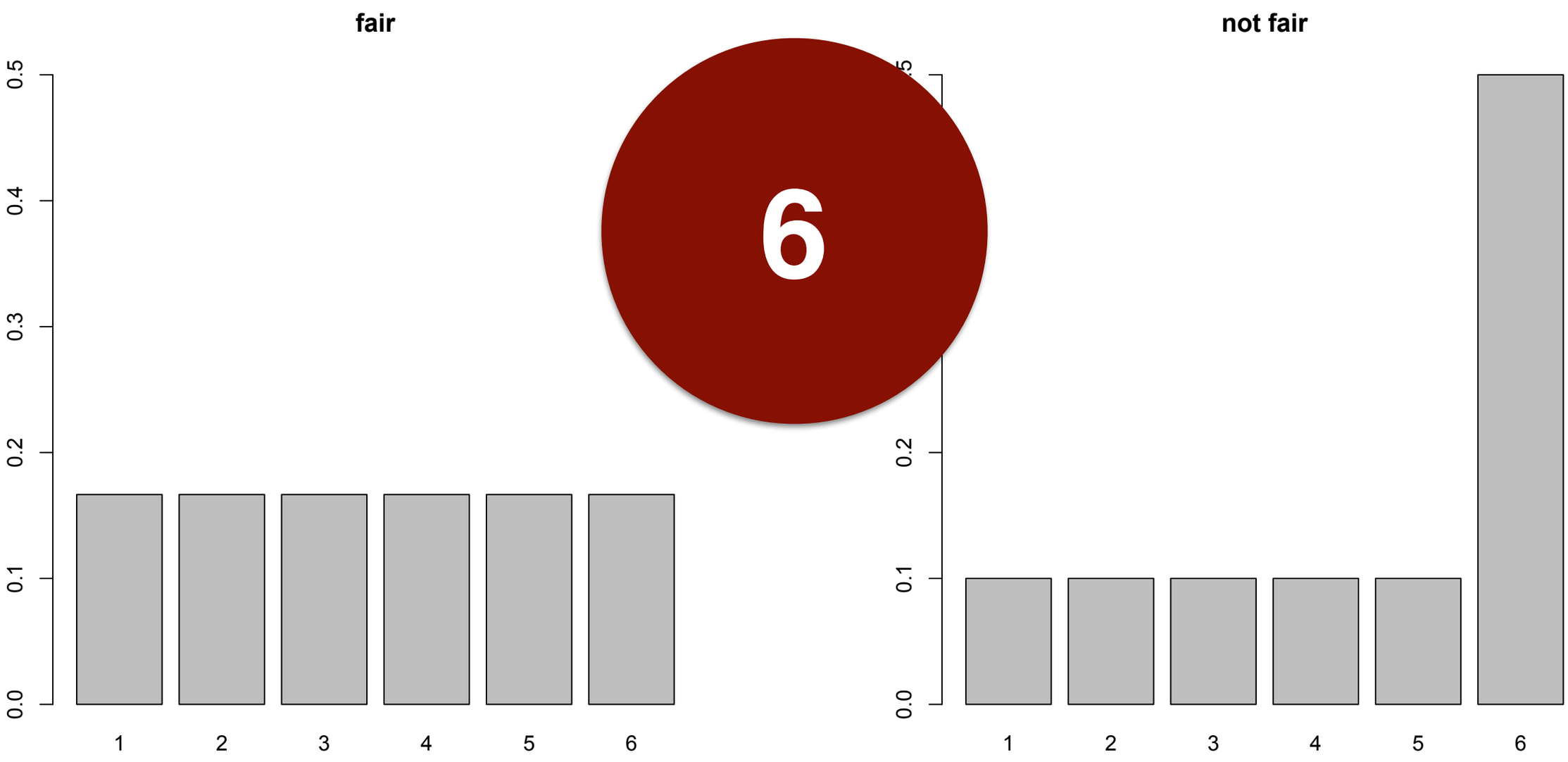
Probability

2 6 6 1 6 3



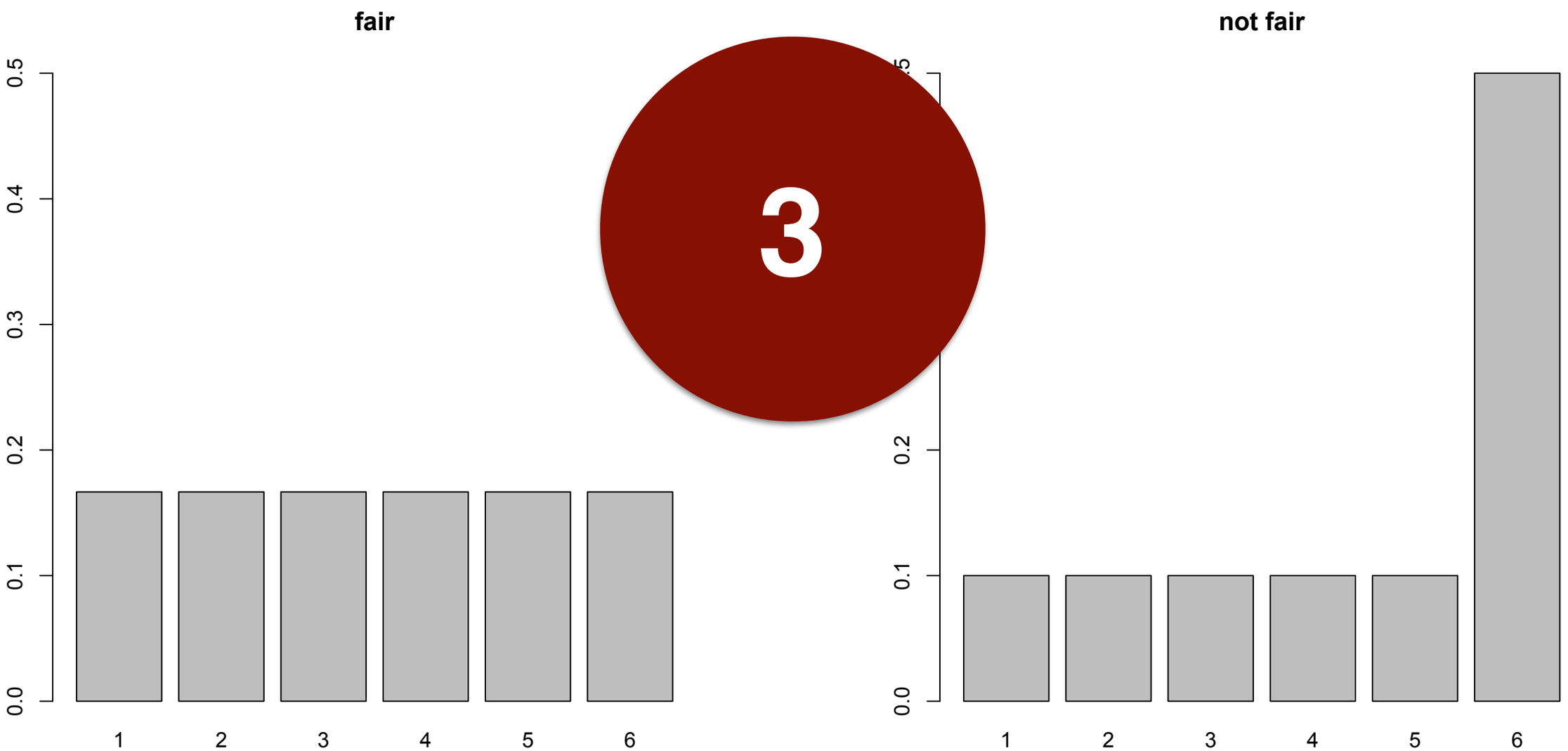
Probability

2 6 6 1 6 3 6



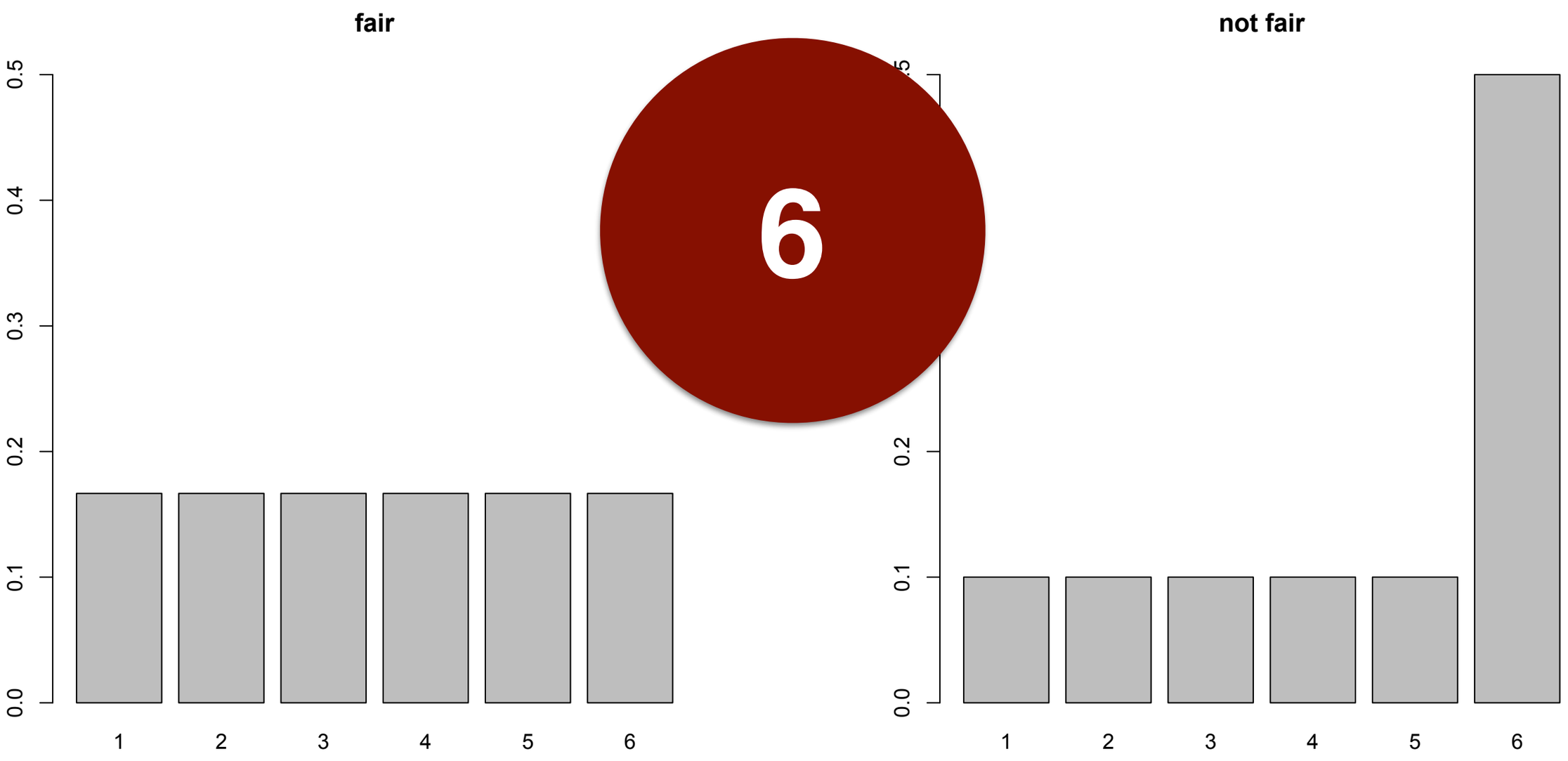
Probability

2 6 6 1 6 3 6 6



Probability

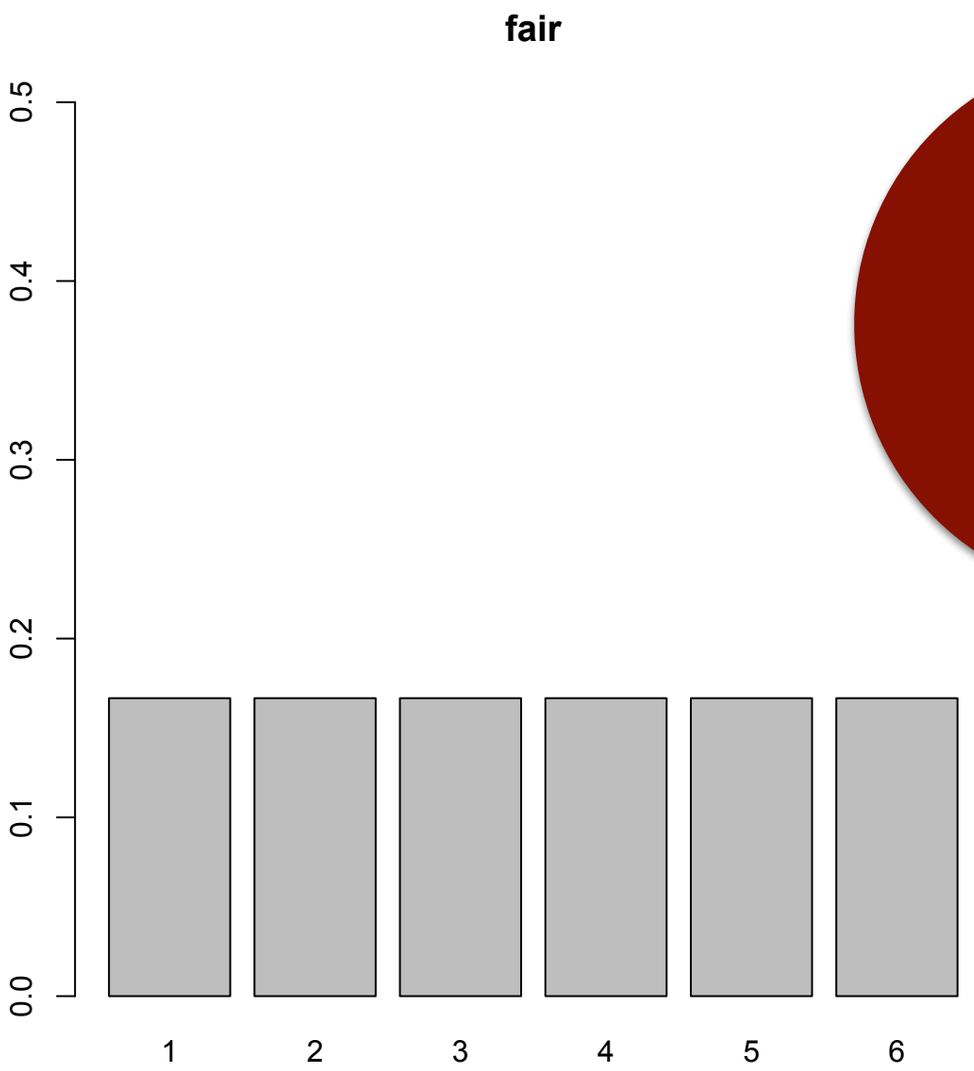
2 6 6 1 6 3 6 6 3

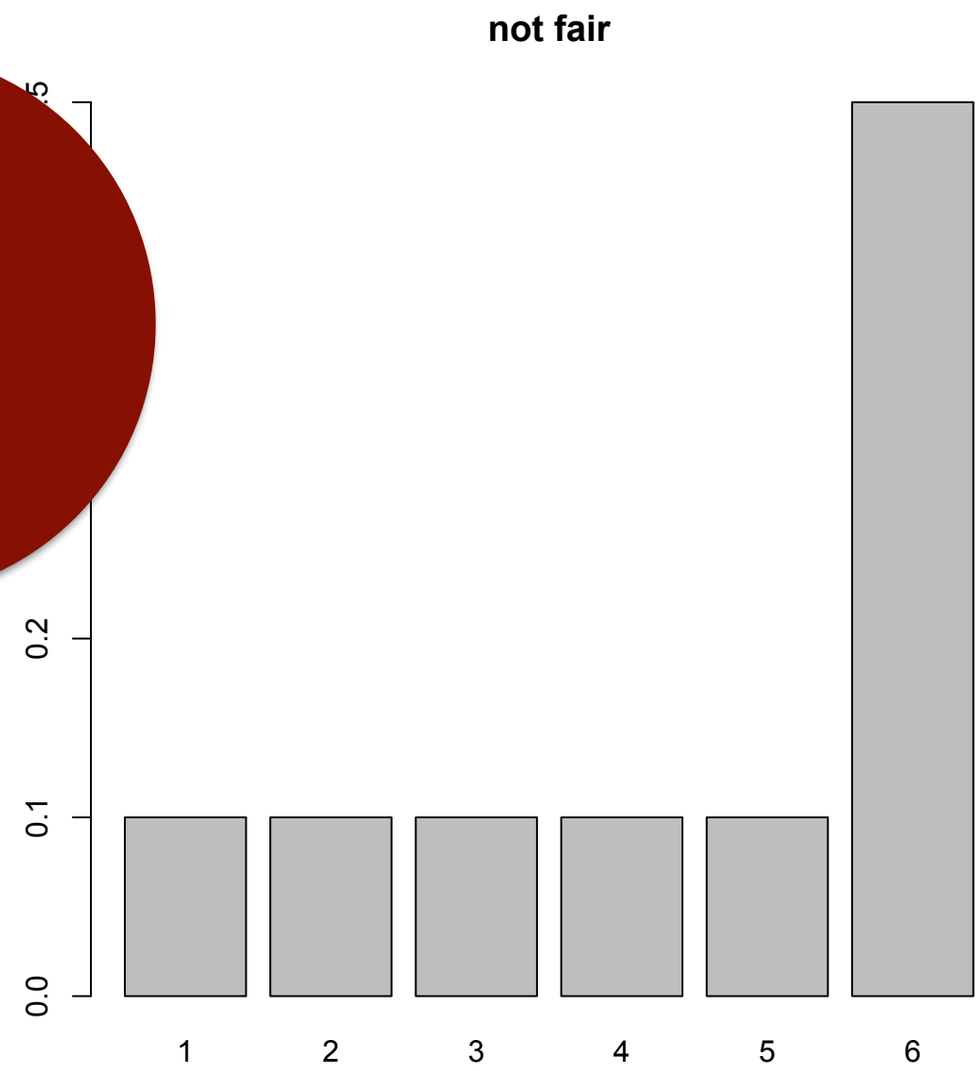


Probability

2 6 6 1 6 3 6 6 3 6

?





Independence

- Two random variables are independent if:

$$P(A, B) = P(A) \times P(B)$$

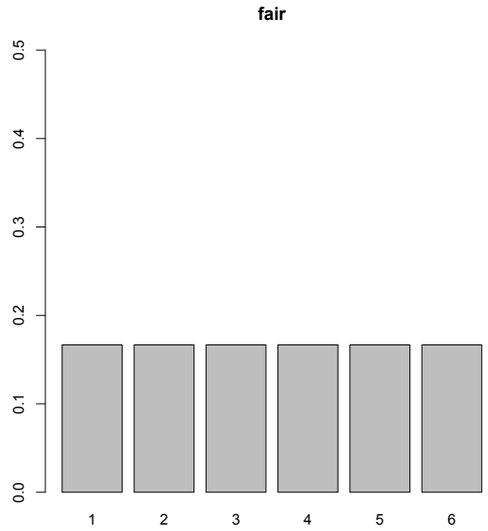
- In general:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{N} P(x_i)$$

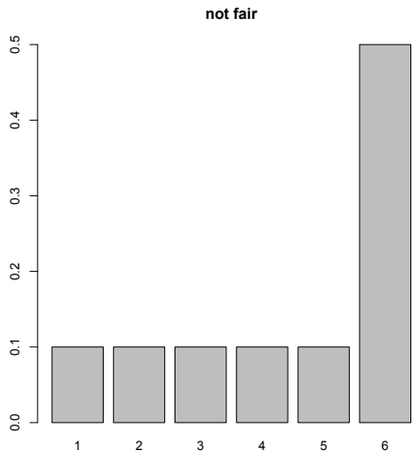
- Information about one random variable (B) gives no information about the value of another (A)

$$P(A) = P(A \mid B) \qquad P(B) = P(B \mid A)$$

Data Likelihood

P(2 6 6 | 

) =.17 x .17 x .17
= 0.004913

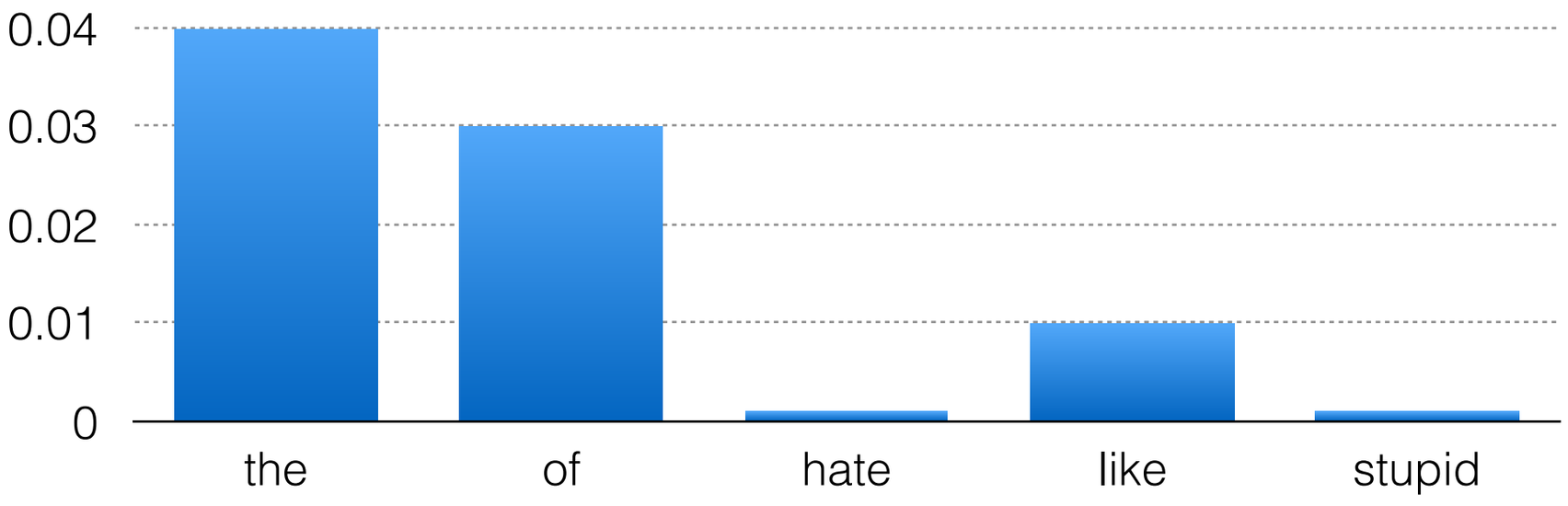
P(2 6 6 | 

) = .1 x .5 x .5
= 0.025

Data Likelihood

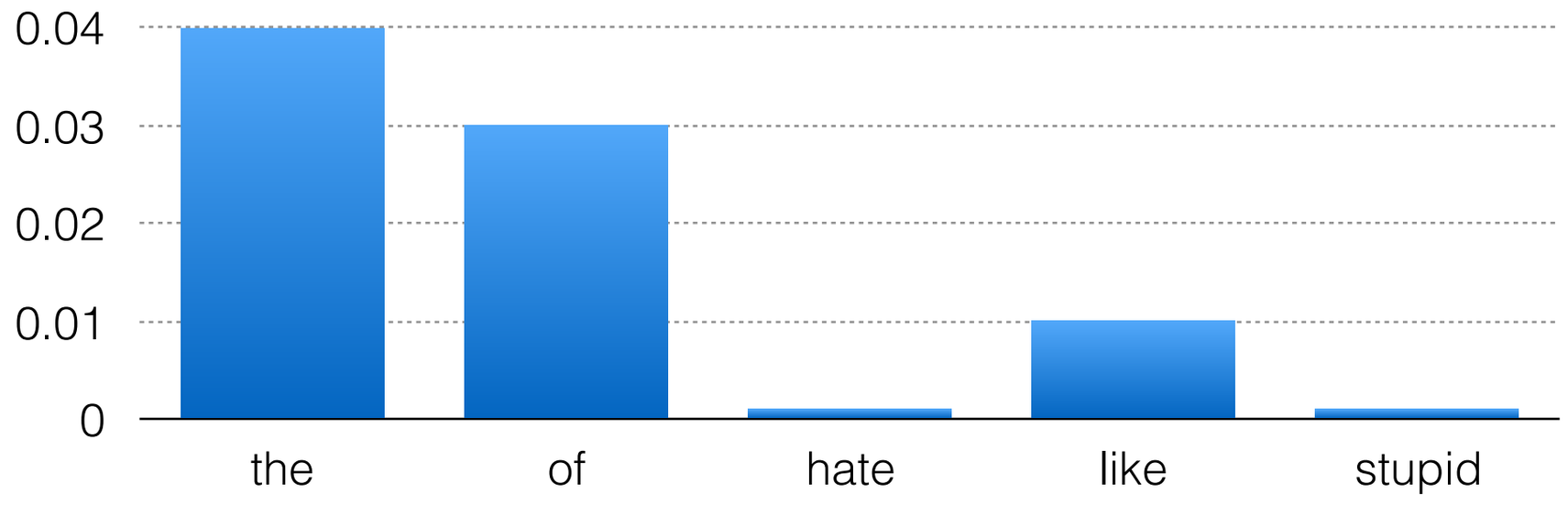
- The likelihood gives us a way of discriminating between possible alternative parameters, but also a strategy for picking a single best* parameter among all possibilities

Word choice as weighted dice

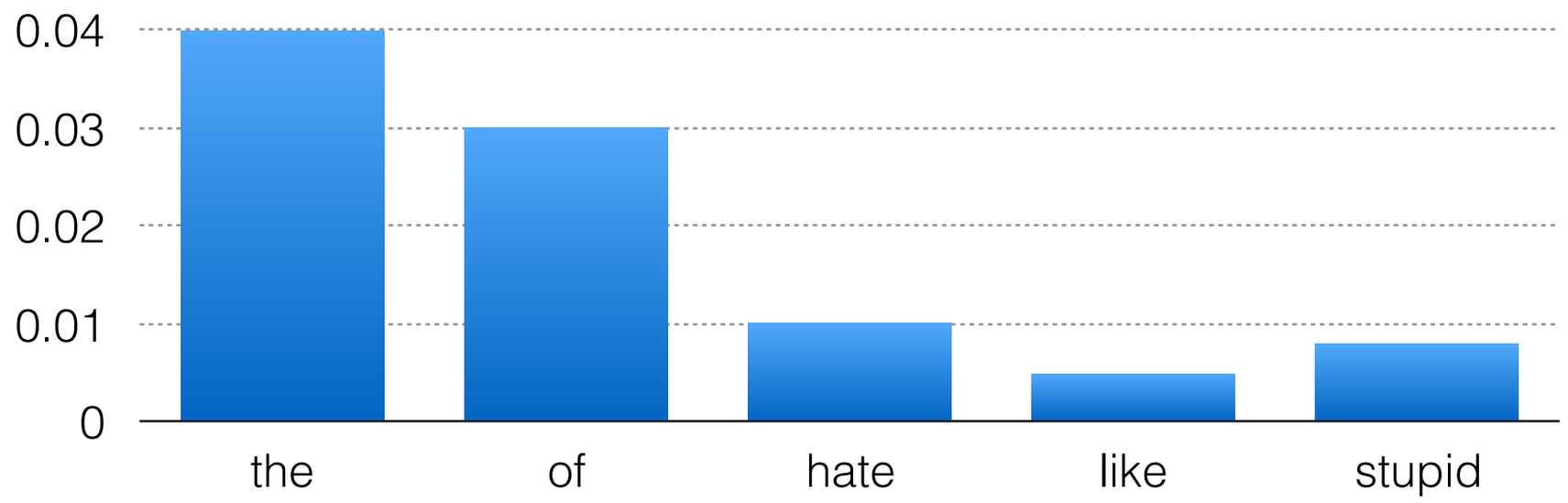


Unigram probability

positive reviews



negative reviews



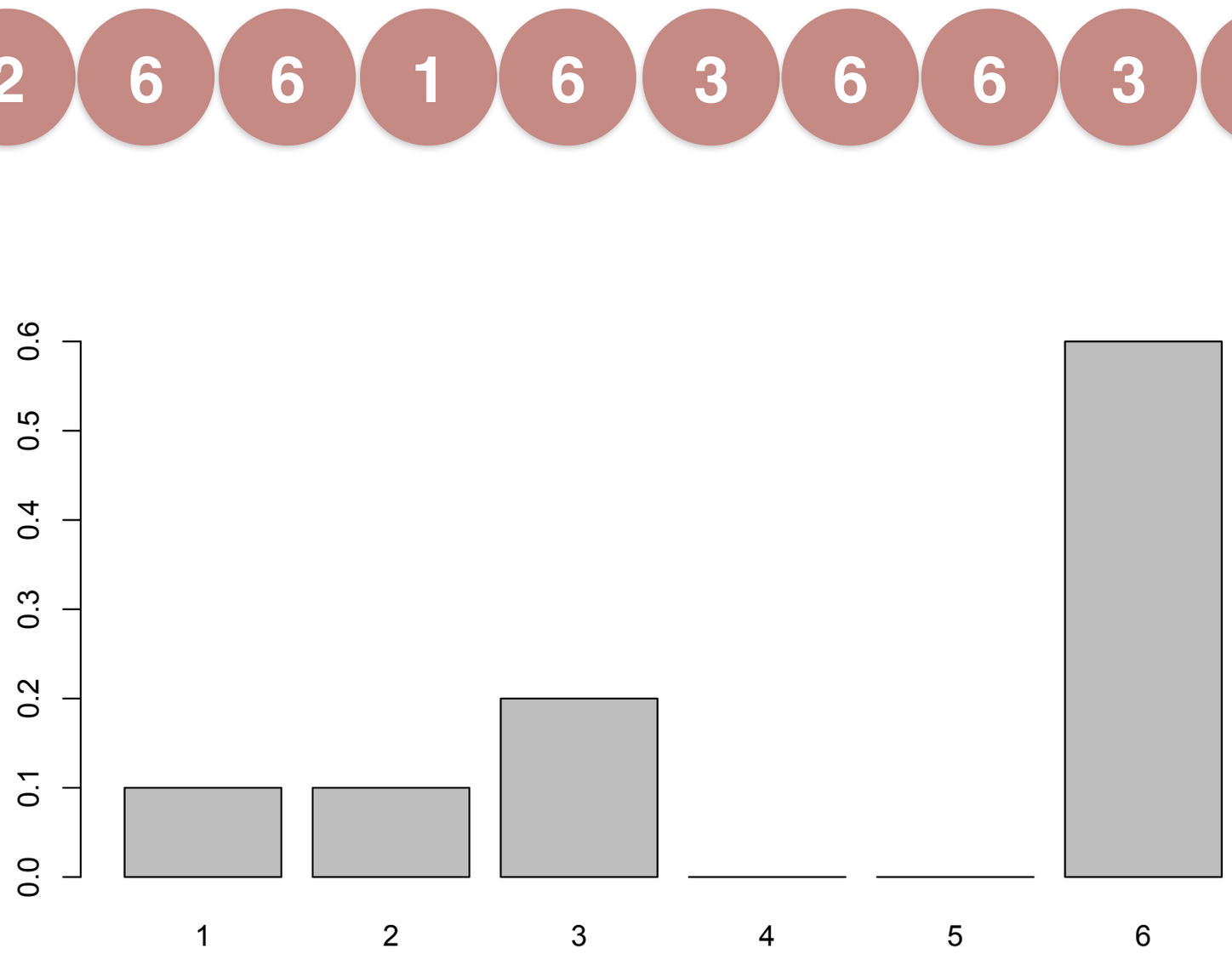
$$P(X = the) = \frac{\#the}{\#\text{total words}}$$

Maximum Likelihood Estimate

- This is a maximum likelihood estimate for P(X); the parameter values for which the data we observe (X) is most likely.

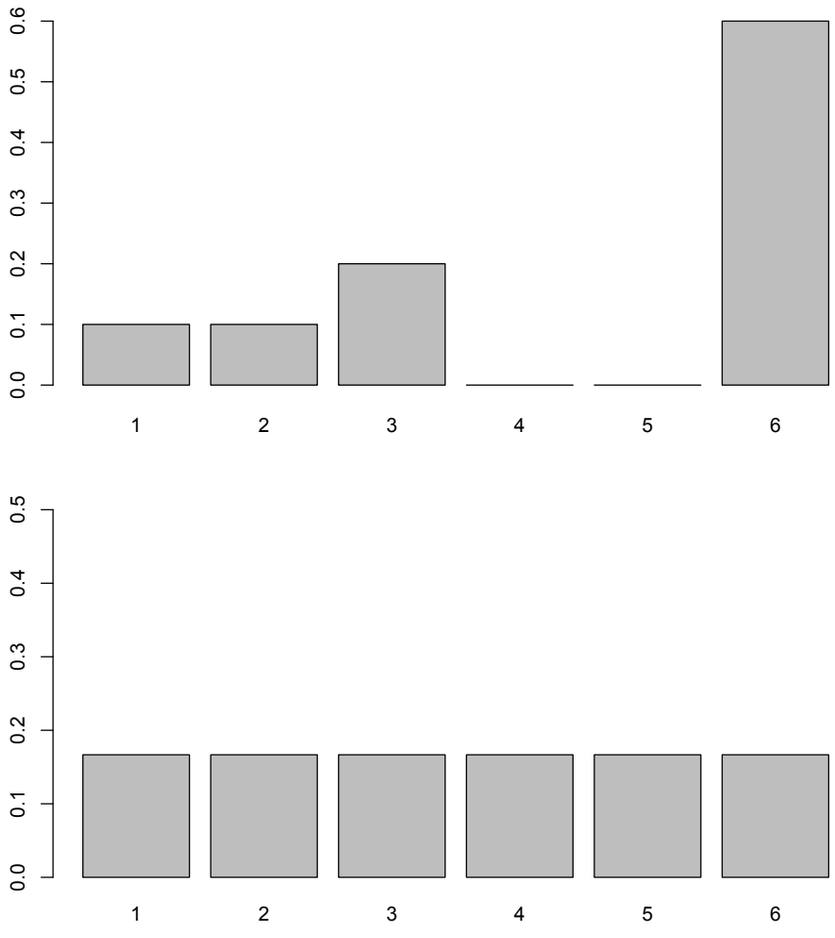
Maximum Likelihood Estimate

2 6 6 1 6 3 6 6 3 6



$X =$ 2 6 6 1 6 3 6 6 3 6

θ_1



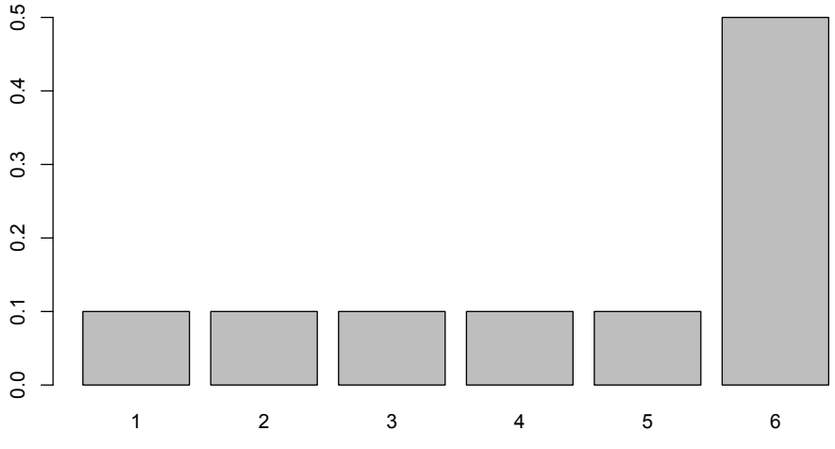
$P(X \mid \theta_1) = 0.0000311040$

θ_2

$P(X \mid \theta_2) = 0.0000000992$

(313x less likely)

θ_3



$P(X \mid \theta_3) = 0.0000031250$

(10x less likely)

Conditional Probability

$$P(X = x|Y = y)$$

- Probability that one random variable takes a particular value *given* the fact that a different variable takes another

$$P(X_i = hate \mid Y = \oplus)$$

Sentiment analysis

“really really the worst movie ever”

Independence Assumption

really really the worst movie ever

P(really, really, the, worst, movie, ever) =
P(really) x P(really) x P(the) … P(ever)

Independence Assumption

really really the worst movie ever

We will assume the features are independent:

$$P(x_1, x_2, x_3, x_4, x_6, x_7 \mid c) = P(x_1 \mid c)P(x_2 \mid c) \ldots P(x_7 \mid c)$$

$$P(x_i \ldots x_n \mid c) = \prod_{i=1}^{N} P(x_i \mid c)$$

A simple classifier

really really the worst movie ever

Y=Positive		Y=Negative	
P(X=really | Y=⊕)	0.0010	P(X=really | Y=⊖)	0.0012
P(X=the | Y=⊕)	0.0551	P(X=the | Y=⊖)	0.0518
P(X=worst | Y=⊕)	0.0001	P(X=worst | Y=⊖)	0.0004
P(X=movie | Y=⊕)	0.0032	P(X=movie | Y=⊖)	0.0045
P(X=ever | Y=⊕)	0.0005	P(X=ever | Y=⊖)	0.0005

A simple classifier

really really the worst movie ever

P(X = “really really the worst movie ever” | Y = ⊕)

P(X=really | Y=⊕) x P(X=really | Y=⊕) x P(X=the | Y=⊕) x P(X=worst | Y=⊕) x P(X=movie | Y=⊕) x P(X=ever | Y=⊕)

= 6.00e-18

P(X = “really really the worst movie ever” | Y = ⊖)

P(X=really | Y=⊖) x P(X=really | Y=⊖) x P(X=the | Y=⊖) x P(X=worst | Y=⊖) x P(X=movie | Y=⊖) x P(X=ever | Y=⊖)

= 6.20e-17 ✅

Aside: use logs

- Multiplying lots of small probabilities (all are under 1) can lead to numerical underflow (converging to 0)

$$\log \prod_i x_i = \sum_i \log x_i$$

A simple classifier

- The classifier we just specified is a maximum likelihood classifier, where we compare the likelihood of the data under each class and choose the class with the highest likelihood

Likelihood: probability of data (here, under class y)

$$P(X = x_i \ldots x_n \mid Y = y)$$

Prior probability of class y

$$P(Y = y)$$

Image credit: https://thebossmagazine.com/will-it-blend/

Naive Bayes Classifier

Bayes' Rule

Prior belief that Y = y
(before you see any data)

Likelihood of the data
given that Y=y

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{\sum_y P(Y = y)P(X = x|Y = y)}$$

Posterior belief that Y=y given that X=x

Bayes' Rule directly applied to our example

Prior belief that Y = positive (before you see any data)

Likelihood of "really really the worst movie ever" given that Y= positive

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{\sum_y P(Y = y)P(X = x|Y = y)}$$

Posterior belief that Y=positive given that X="really really the worst movie ever"

This sum ranges over y=positive + y=negative (so that it sums to 1)

Likelihood: probability of data (here, under class y)

$$P(X = x_i \ldots x_n \mid Y = y)$$

Prior probability of class y

$$P(Y = y)$$

Posterior belief in the probability of class y after seeing data

$$P(Y = y \mid X = x_i \ldots x_n)$$

Naive Bayes Classifier

$$\frac{P(Y = \oplus)P(X = \text{“really } \ldots\text{”} \mid Y = \oplus)}{P(Y = \oplus)P(X = \text{“really } \ldots\text{”} \mid Y = \oplus) + P(Y = \ominus)P(X = \text{“really } \ldots\text{”} \mid Y = \ominus)}$$

Let's say P(Y=⊕) = P(Y=⊖) = 0.5

(i.e., both are equally likely a priori)

$$\frac{0.5 \times (6.00 \times 10^{-18})}{0.5 \times (6.00 \times 10^{-18}) + 0.5 \times (6.2 \times 10^{-17})}$$

$$P(Y = \oplus \mid X = \text{“really } \ldots\text{”}) = 0.088$$

$$P(Y = \ominus \mid X = \text{“really } \ldots\text{”}) = 0.912$$

Naive Bayes Classifier

- To turn probabilities into a classification decisions, we just select the label with the highest posterior probability

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P(Y \mid X)$$

$$P(Y = \oplus \mid X = \text{“really } \ldots\text{”}) = 0.088$$

$$P(Y = \ominus \mid X = \text{“really } \ldots\text{”}) = 0.912$$

Taxicab Problem

“A cab was involved in a hit and run accident at night. Two cab companies, the Green and the Blue, operate in the city. You are given the following data:

- 85% of the cabs in the city are Green and 15% are Blue.
- A witness identified the cab as Blue. The court tested the reliability of the witness under the same circumstances that existed on the night of the accident and concluded that the witness correctly identified each one of the two colors 80% of the time and failed 20% of the time.

What is the probability that the cab involved in the accident was Blue rather than Green knowing that this witness identified it as Blue?”

(Tversky & Kahneman 1981)

Prior Belief

- Now let's assume that there are 1000 times more positive reviews than negative reviews.
  - P(Y= negative) = 0.000999
  - P(Y = positive) = 0.999001

$$\frac{0.999001 \times (6.00 \times 10^{-18})}{0.999001 \times (6.00 \times 10^{-18}) + 0.000999 \times (6.2 \times 10^{P}(-17)}$$

$$P(Y = \oplus \mid X = \text{“really } \ldots\text{”}) = 0.990$$
$$P(Y = \ominus \mid X = \text{“really } \ldots\text{”}) = 0.010$$

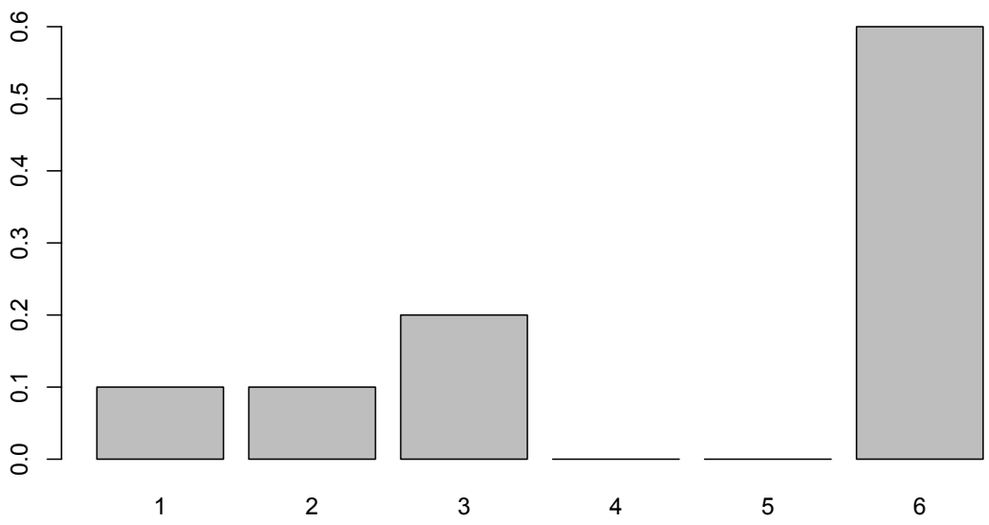
Priors

- Priors can be informed (reflecting expert knowledge) but in practice, but priors in Naive Bayes are often simply estimated from training data

$$P(Y = \oplus) = \frac{\#\oplus}{\#\text{total texts}}$$

Smoothing

- Maximum likelihood estimates can fail miserably when features are never observed with a particular class.



What's the probability of:

2 4 6

Smoothing

- One solution: add a little probability mass to every element.

maximum likelihood estimate

$$P(x_i \mid y) = \frac{n_{i,y}}{n_y}$$

$n_{i,y}$ = count of word i in class y
n_y = number of words in y
V = size of vocabulary

smoothed estimates

$$P(x_i \mid y) = \frac{n_{i,y} + \alpha}{n_y + V\alpha}$$

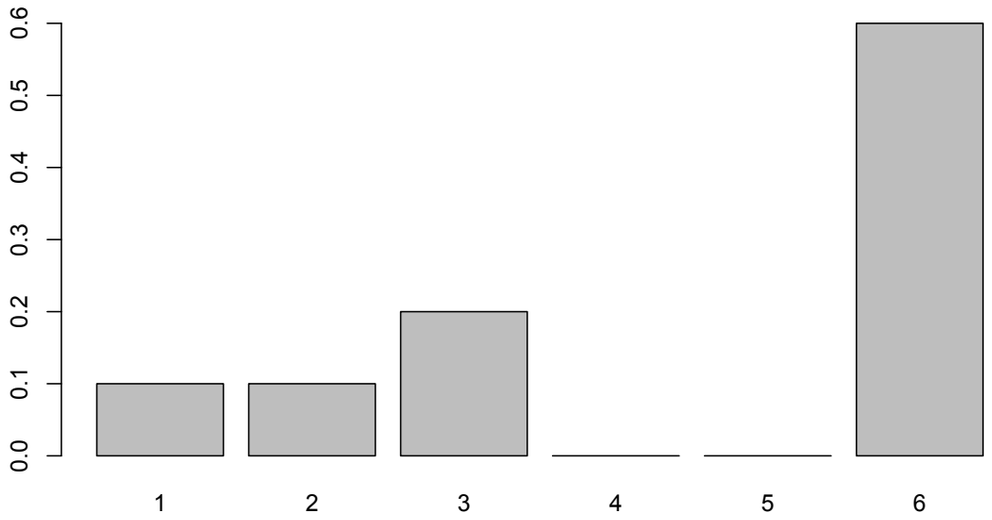
same α for all x_i

$$P(x_i \mid y) = \frac{n_{i,y} + \alpha_i}{n_y + \sum_{j=1}^{V} \alpha_j}$$

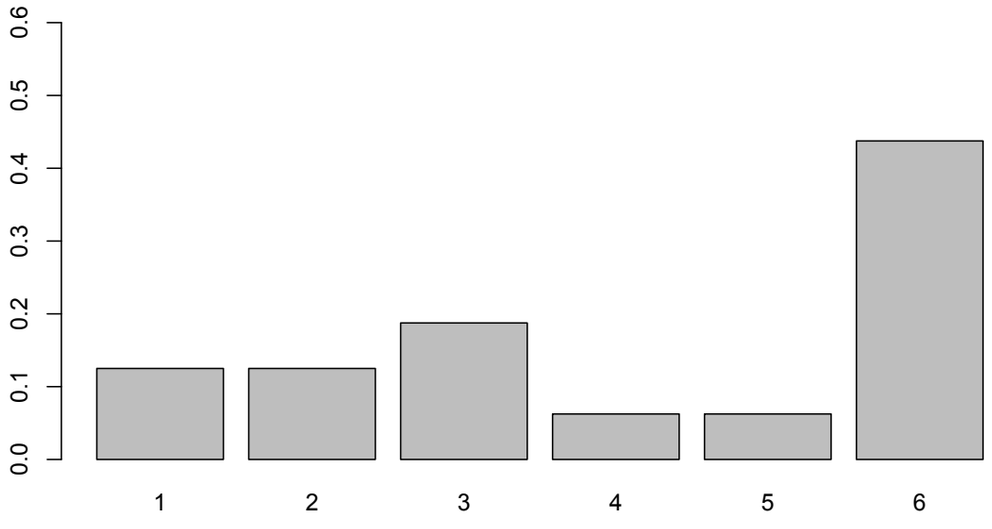
possibly different α for each x_i

Smoothing

MLE



smoothing with α =1



Naive Bayes training

Training a Naive Bayes classifier consists of estimating these two quantities from training data for all classes y

$$P(Y = y|X = x) = \frac{P(Y = y)P(X = x|Y = y)}{\sum_y P(Y = y)P(X = x|Y = y)}$$

At test time, use those estimated probabilities to calculate the posterior probability of each class y and select the class with the highest probability

- Naive Bayes' independence assumption can be killer
- One instance of *hate* makes seeing others much more likely (each mention contributes the same amount of information)
- We can mitigate this problem by not reasoning over counts of tokens and instead using binary values to indicate a word's presence or absence

	Apocalypse now	North
the	1	1
of	0	0
hate	0	9
genius	1	0
bravest	1	0
stupid	0	1
like	0	1
…		

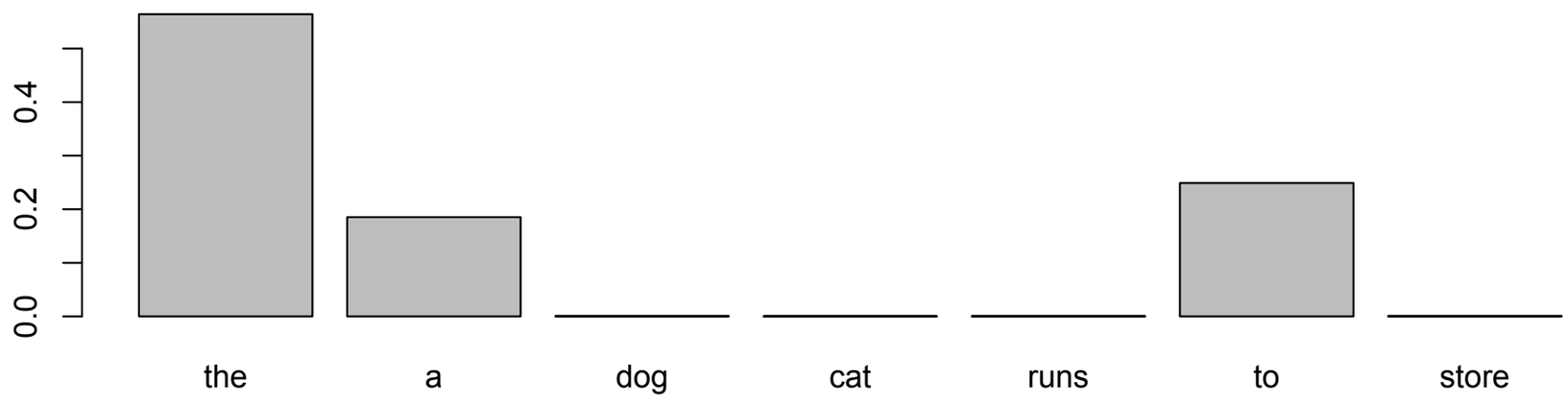
- Naive Bayes' independence assumption can be killer
- One instance of *hate* makes seeing others much more likely (each mention contributes the same amount of information)
- We can mitigate this problem by not reasoning over counts of tokens and instead using binary values to indicate a word's presence or absence

	Apocalypse now	North
the	1	1
of	0	0
hate	0	~~9~~ 1
genius	1	0
bravest	1	0
stupid	0	1
like	0	1
…		

Multinomial Naive Bayes

Discrete distribution for modeling count data (e.g., word counts; single parameter θ

$\theta =$



the	a	dog	cat	runs	to	store
3	1	0	1	0	2	0
531	209	13	8	2	331	1

Multinomial Naive Bayes

Maximum likelihood parameter estimate

$$\hat{\theta}_i = \frac{n_i}{N}$$

	the	a	dog	cat	runs	to	store
count n	531	209	13	8	2	331	1
θ	0.48	0.19	0.01	0.01	0.00	0.30	0.00

Bernoulli Naive Bayes

- Binary event (true or false; {0, 1})
- One parameter: p (probability of an event occurring)

$$P(x = 1 \mid p) = p$$

$$P(x = 0 \mid p) = 1 - p$$

Examples:

- Probability of a particular feature being true (e.g., review contains “hate”)

$$\hat{p}_{mle} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

Bernoulli Naive Bayes

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	p_{MLE}
f_1	1	0	0	0	1	1	0	0	0.375
f_2	0	0	0	0	0	0	1	0	0.125
f_3	1	1	1	1	1	0	0	1	0.750
f_4	1	0	0	1	1	0	0	1	0.500
f_5	0	0	0	0	0	0	0	0	0.000

Bernoulli Naive Bayes

	Positive				Negative					
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$p_{MLE,P}$	$p_{MLE,N}$
f_1	1	0	0	0	1	1	0	0	0.25	0.50
f_2	0	0	0	0	0	0	1	0	0.00	0.25
f_3	1	1	1	1	1	0	0	1	1.00	0.50
f_4	1	0	0	1	1	0	0	1	0.50	0.50
f_5	0	0	0	0	0	0	0	0	0.00	0.00

Tricks for SA

- Negation in bag of words: add negation marker to all words between negation and end of clause (e.g., comma, period) to create new vocab term [Das and Chen 2001]

- I do not [like this movie]

- I do not like_NEG this_NEG movie_NEG

Sentiment Dictionaries

- MPQA subjectivity lexicon (Wilson et al. 2005) http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

- LIWC (Linguistic Inquiry and Word Count, Pennebaker 2015)

pos	neg
unlimited	lag
prudent	contortions
supurb	fright
closeness	lonely
impeccably	tenuously
fast-paced	plebeian
treat	mortification
destined	outrage
blessing	allegations
steadfastly	disoriented

Homework 1 Preview

- You'll actually *implement* Naive Bayes from scratch
- And implement another classifier
- Due January 30th, so it's useful to finish Homework 0 early


SHURE

Let's set up your KaggleInClass for HW0

https://www.kaggle.com/c/si630w20hw0

Your task: pretend none of the webpages have emails and upload a solution that has `None` for every page.

Looking forward to a great semester with you all!

jurgens@umich.edu

Office: NQ 3385