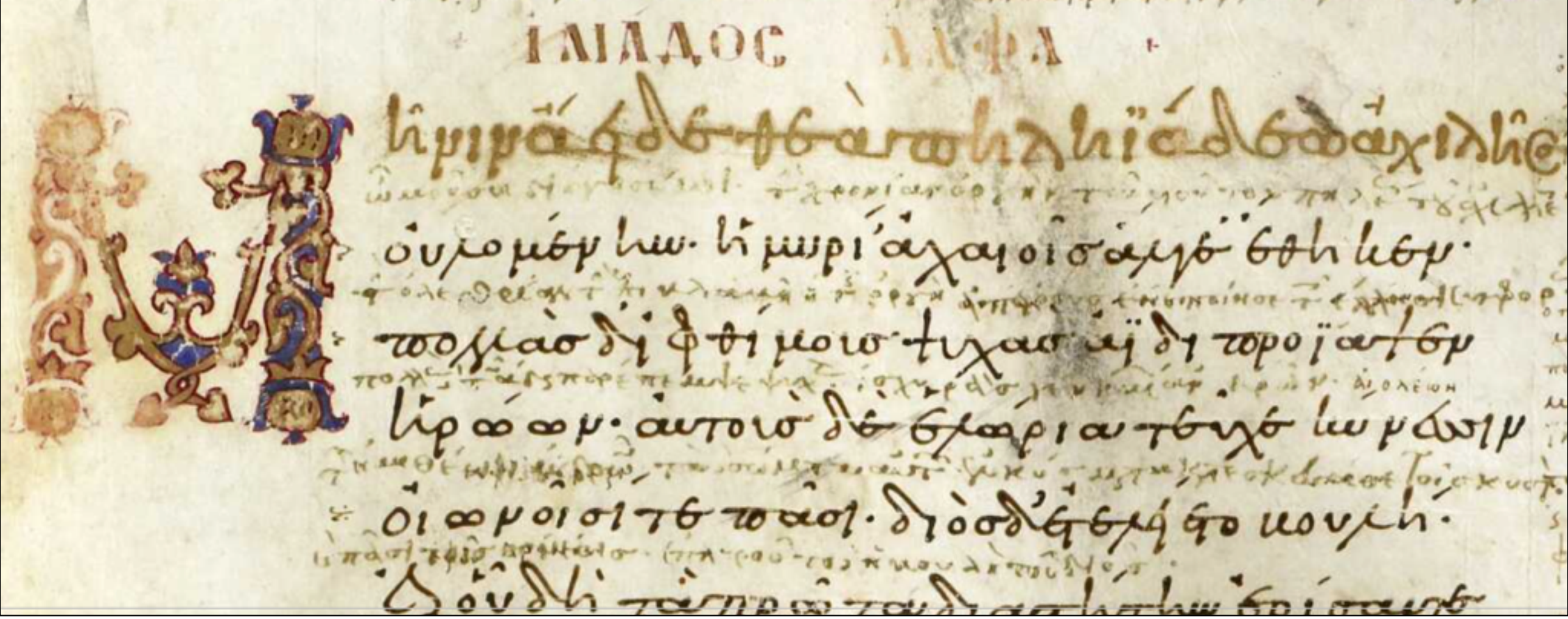


Natural Language Processing

Info 159/259

Lecture 2: Text classification 1 (Aug 29, 2017)

David Bamman, UC Berkeley

Quizzes

- Take place in the first 10 minutes of class:
    - start at 3:40, end at 3:50
- We drop 3 lowest quizzes and homeworks total. For Q quizzes and H homeworks, we keep (H+Q)-3 highest scores.

Classification

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 documents
$\mathcal{Y}$ = {english, mandarin, greek, …}

x = a single document
y = ancient greek

Classification

$$h(x) = y$$

$$h(\mu\eta\nu\iota\nu\ \acute{\alpha}\varepsilon\iota\delta\varepsilon\ \theta\varepsilon\grave{\alpha}) = \textit{ancient grc}$$

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 characters in unicode point range 0370-03FF:
$\hat{h}(x)$ = greek

Classification

Supervised learning

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

Text categorization problems

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.

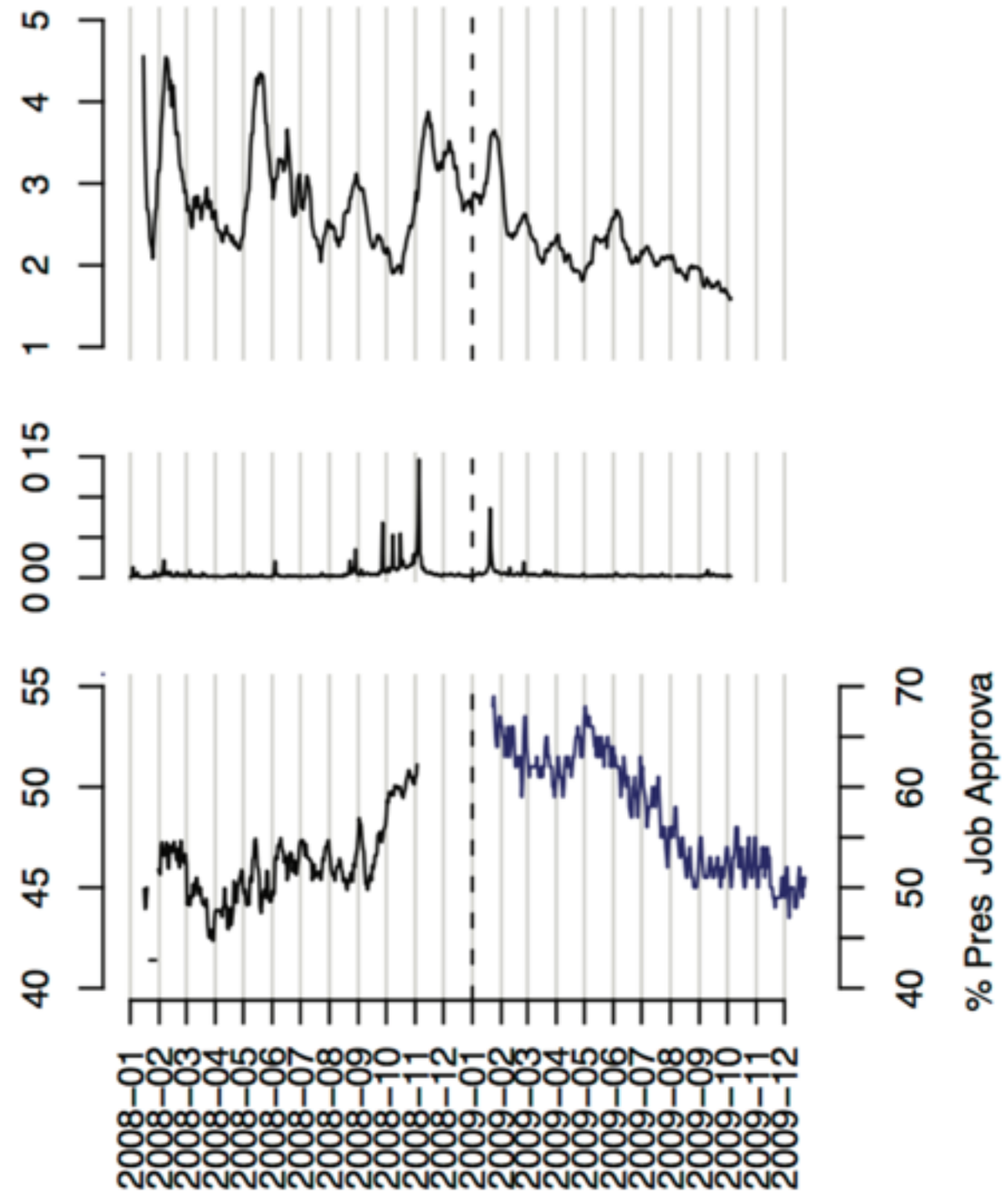
Hu and Liu (2004), “Mining and Summarizing Customer Reviews”

Sentiment analysis

- Political/product opinion mining

Twitter sentiment →

Job approval polls →



O'Connor et al (2010), "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series"

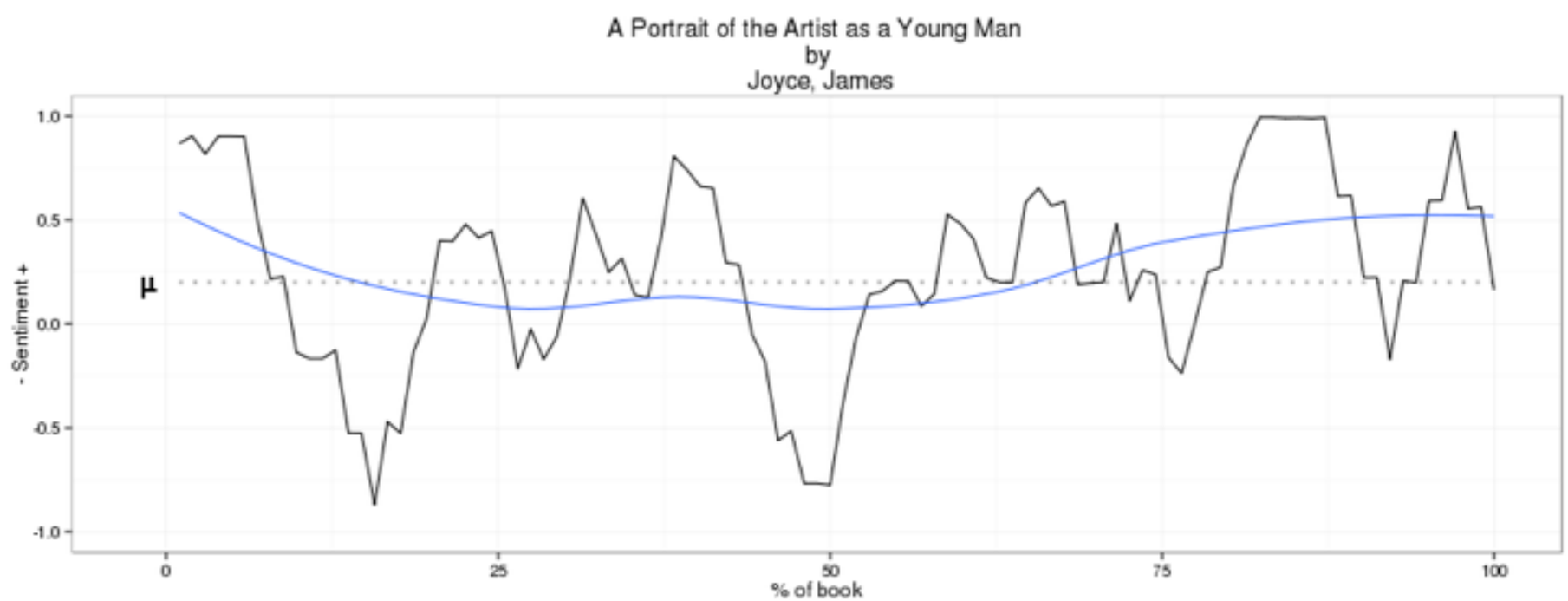
Figure 9: The sentiment ratio for *obama* (15-day window), and fraction of all Twitter messages containing *obama* (day-by-day, no smoothing), compared to election polls (2008) and job approval polls (2009).

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

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

$\hat{h}(x)$

- 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 SA

- 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 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)

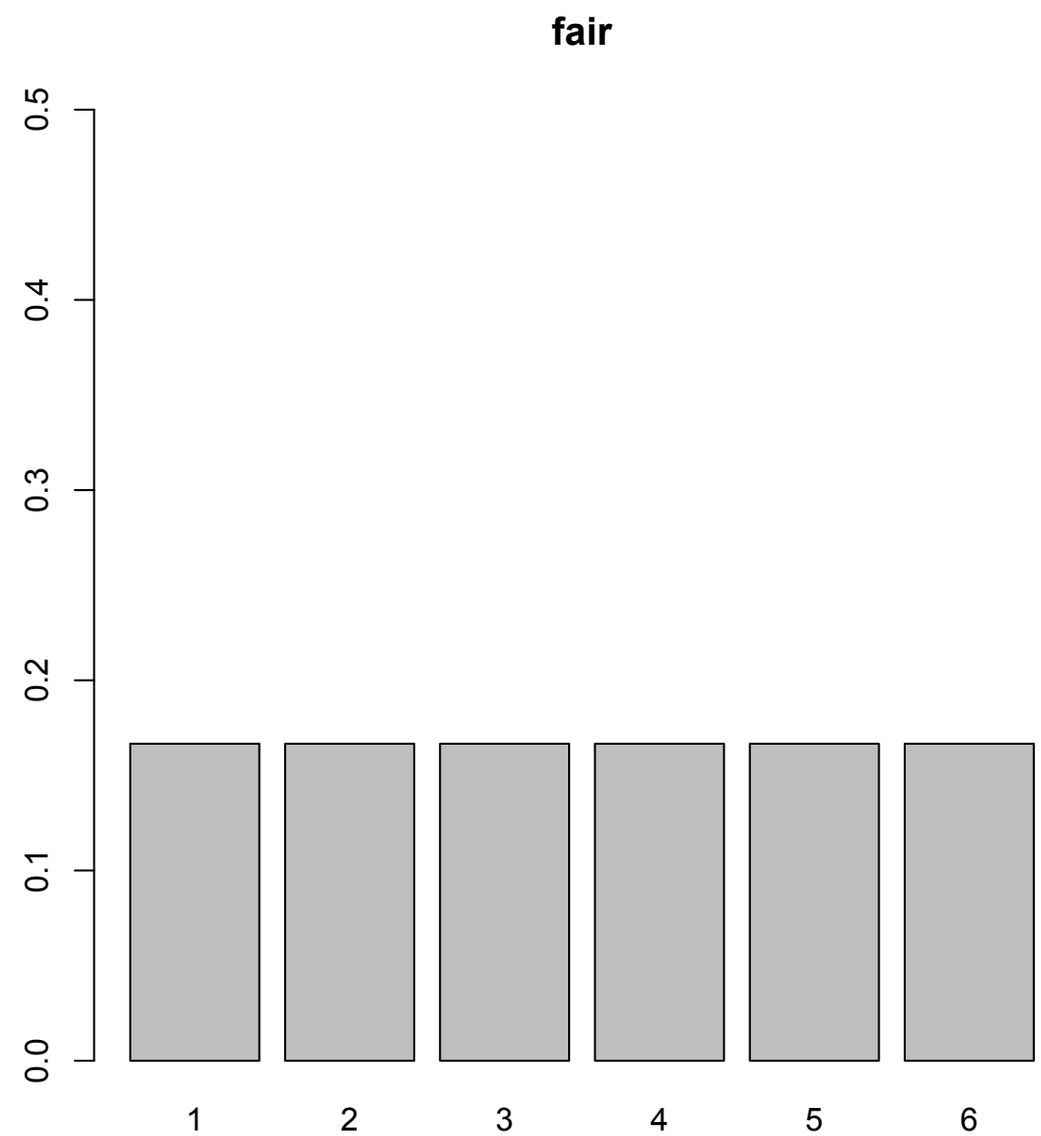
$$X \in \{1, 2, 3, 4, 5, 6\}$$

Two conditions:

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

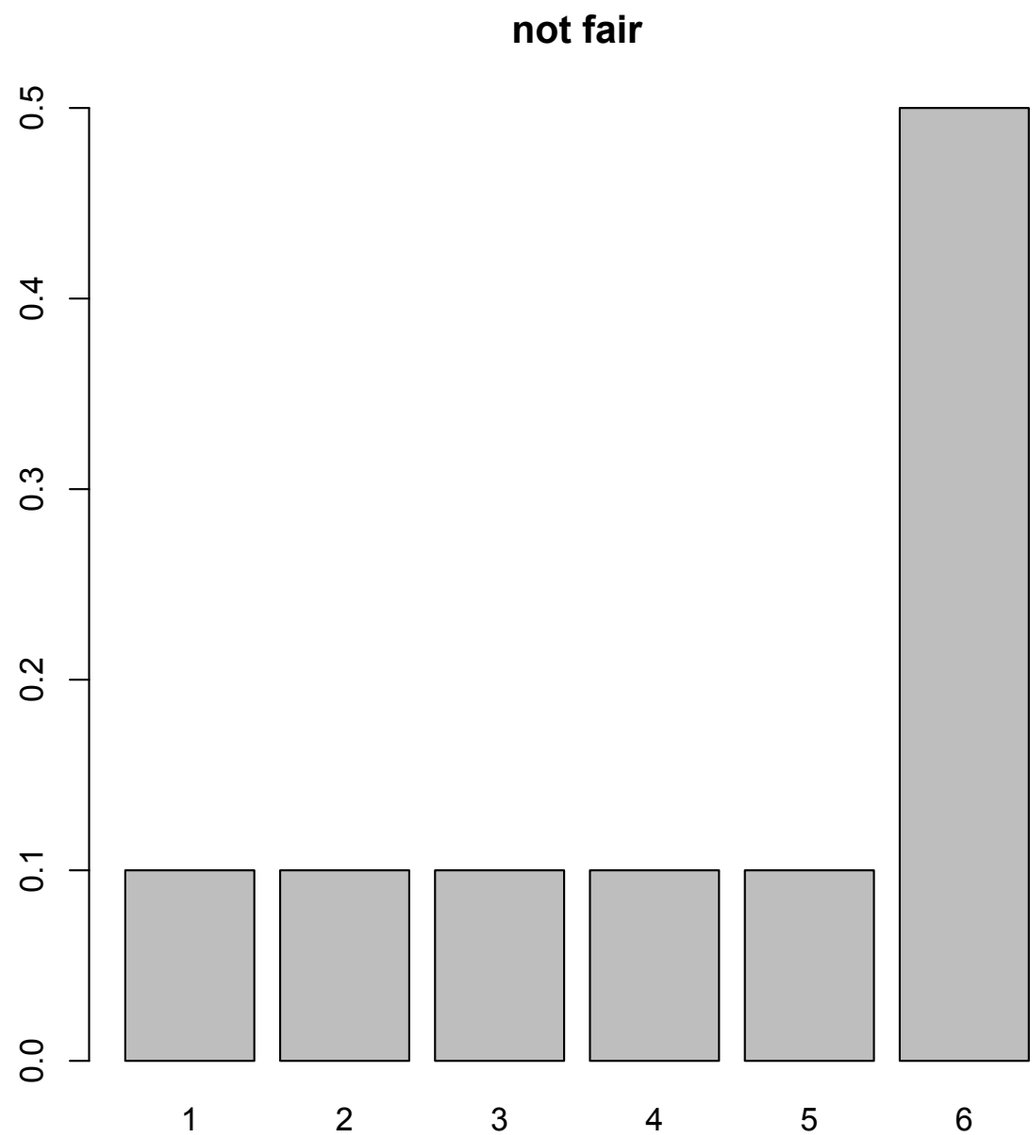
Fair dice

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



Weighted dice

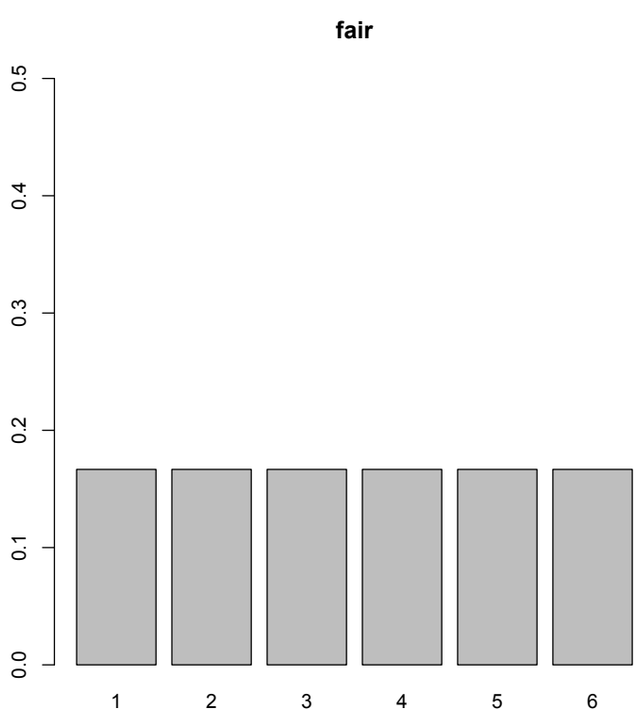
$$X \in \{1, 2, 3, 4, 5, 6\}$$



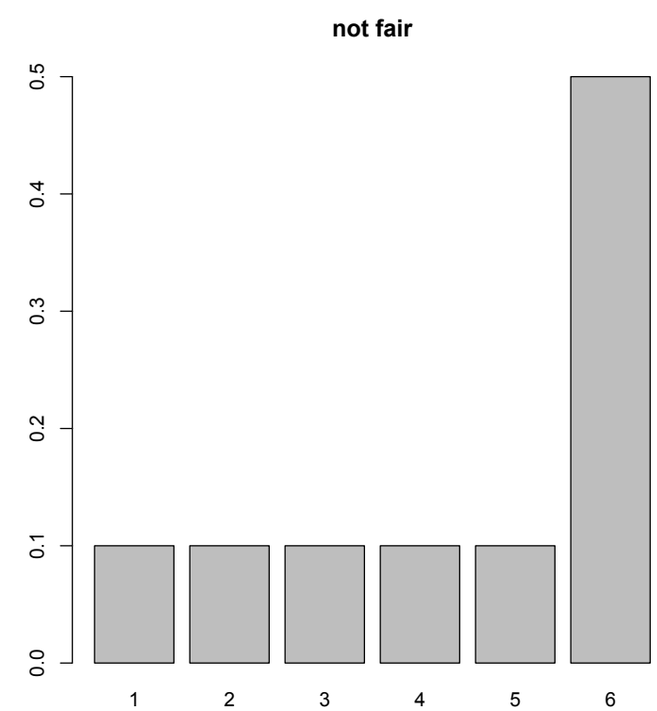
Inference

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

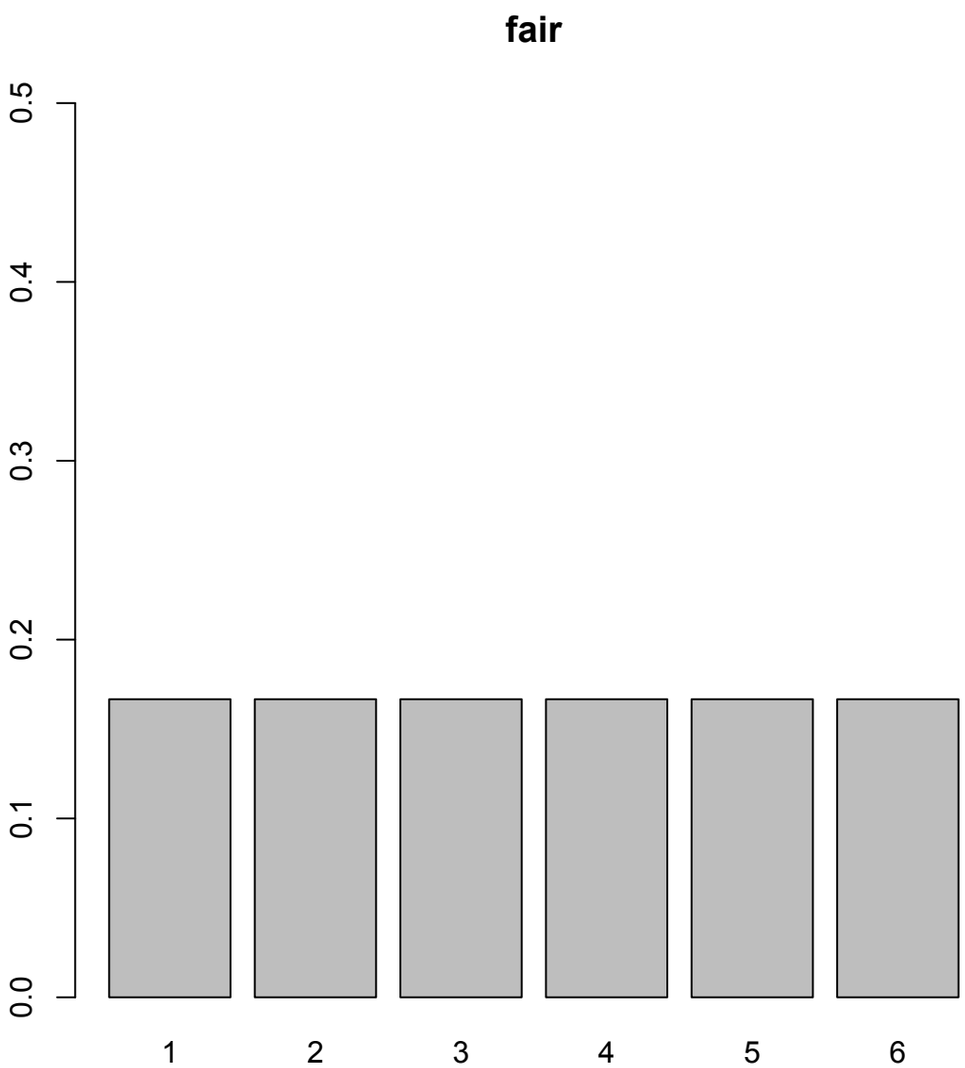
We want to *infer* the probability distribution that generated the data we see.

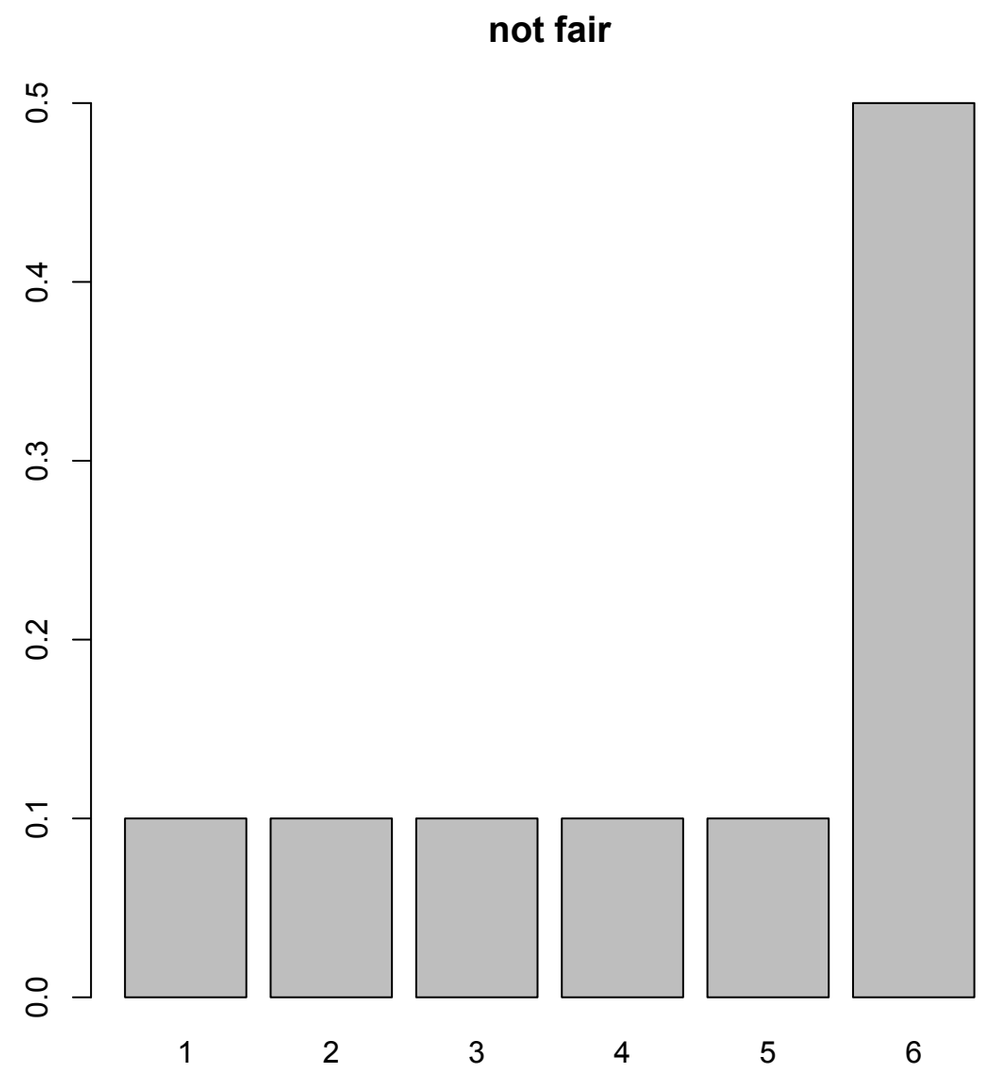


?

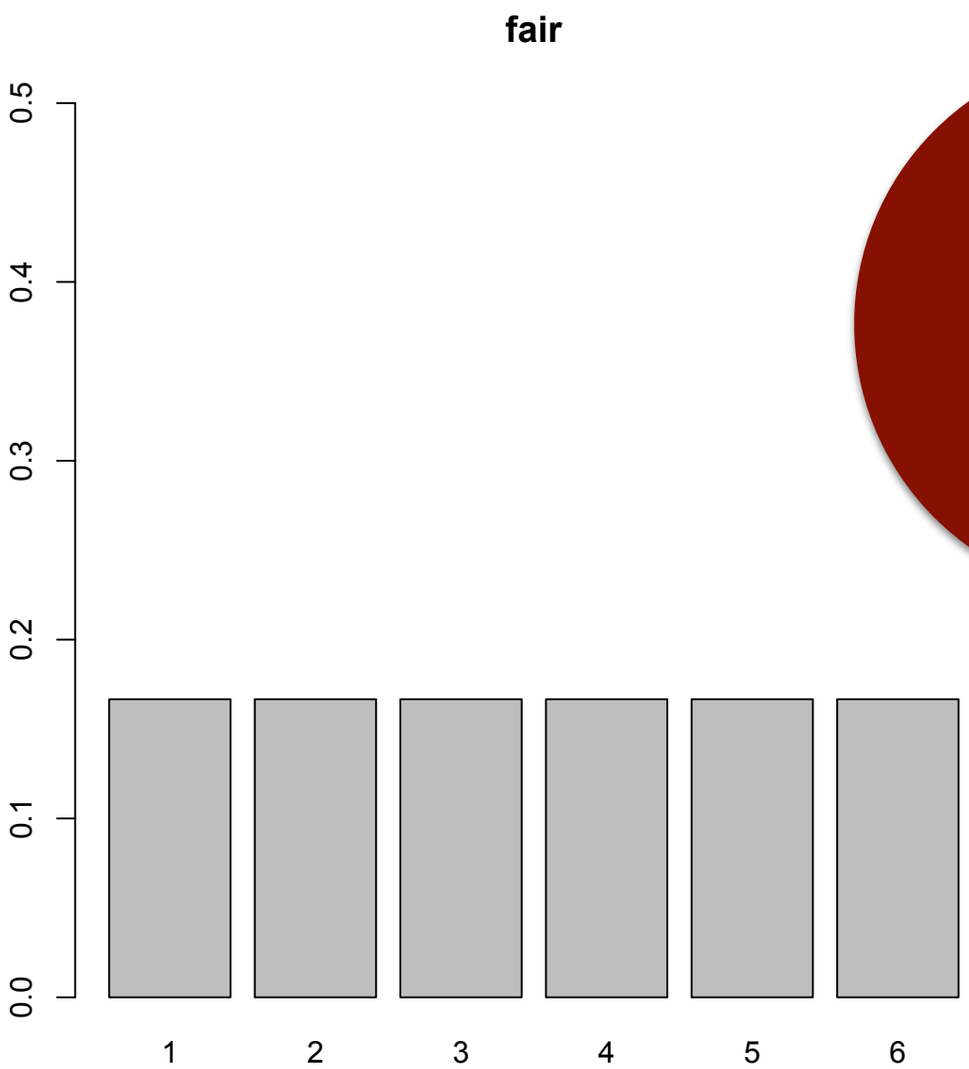


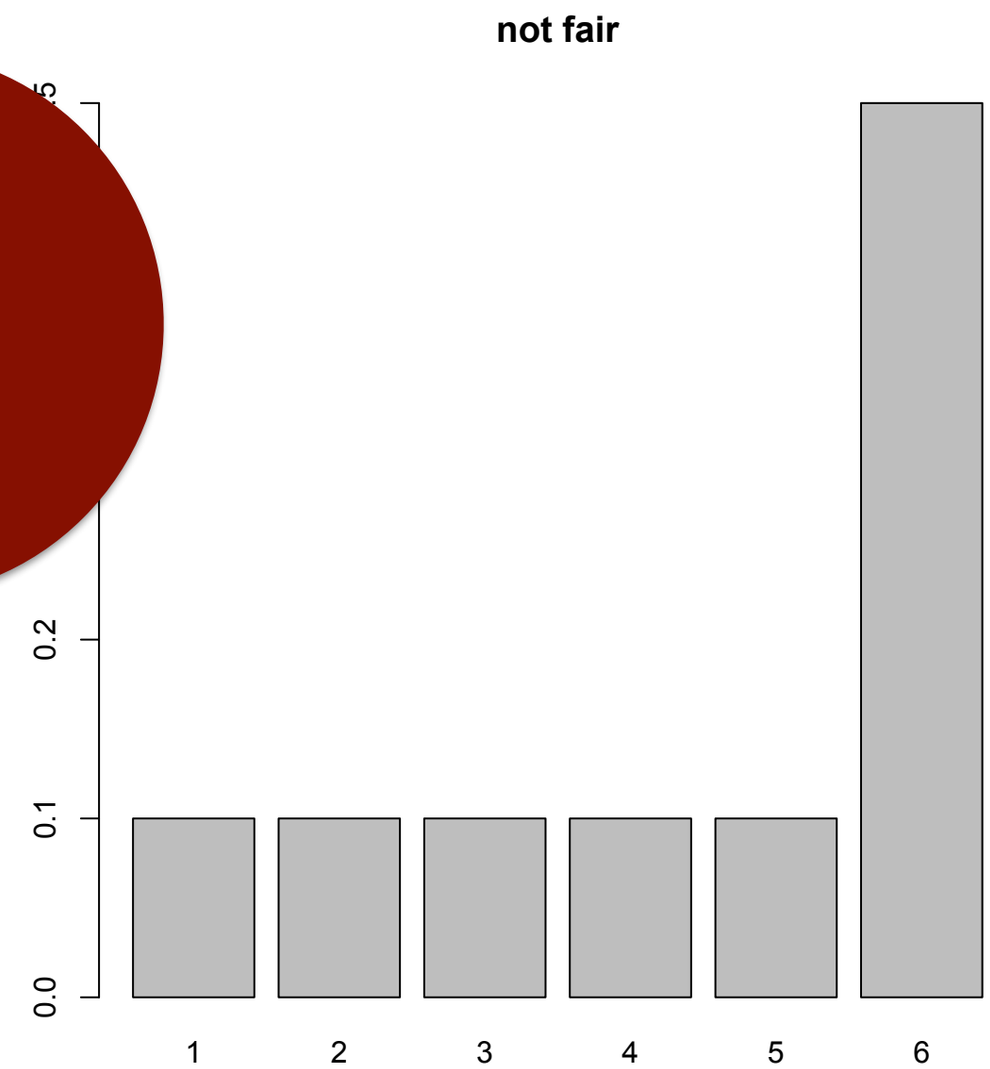
Probability





Probability

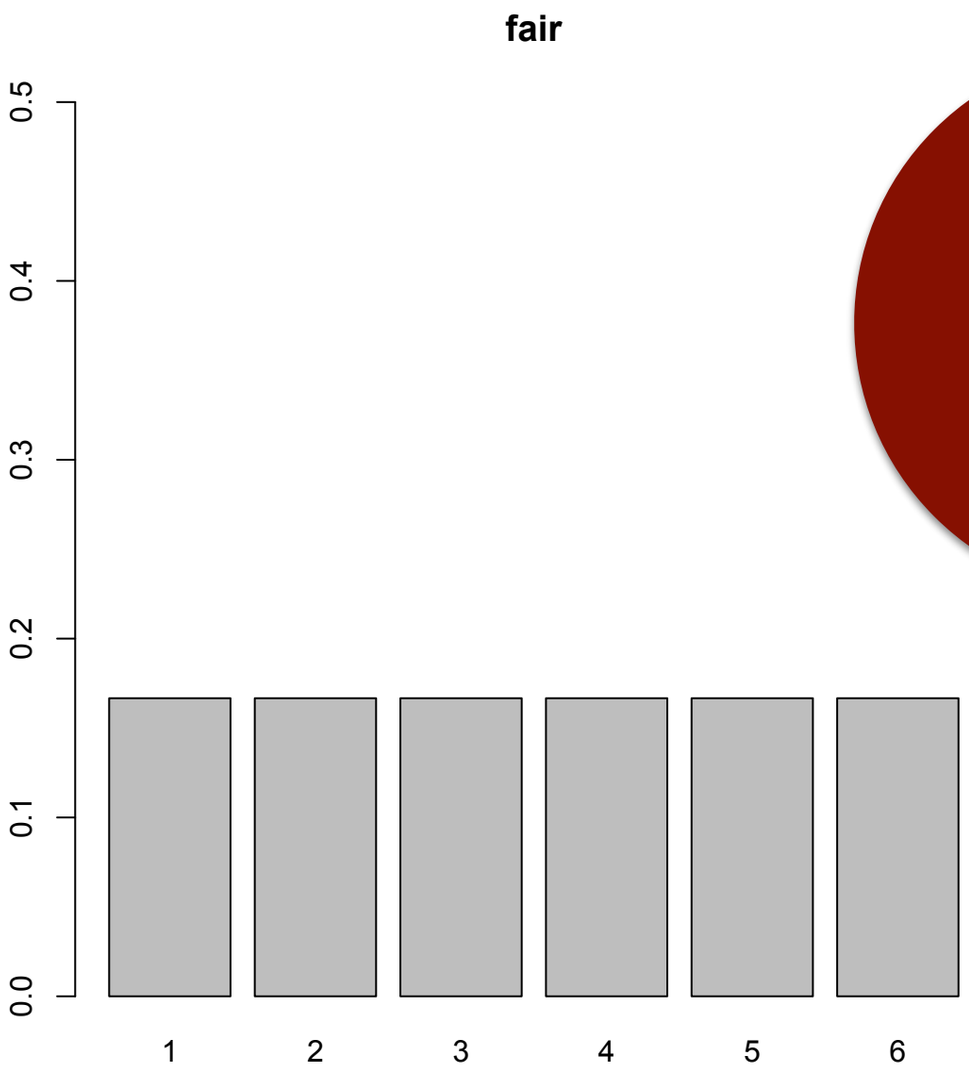




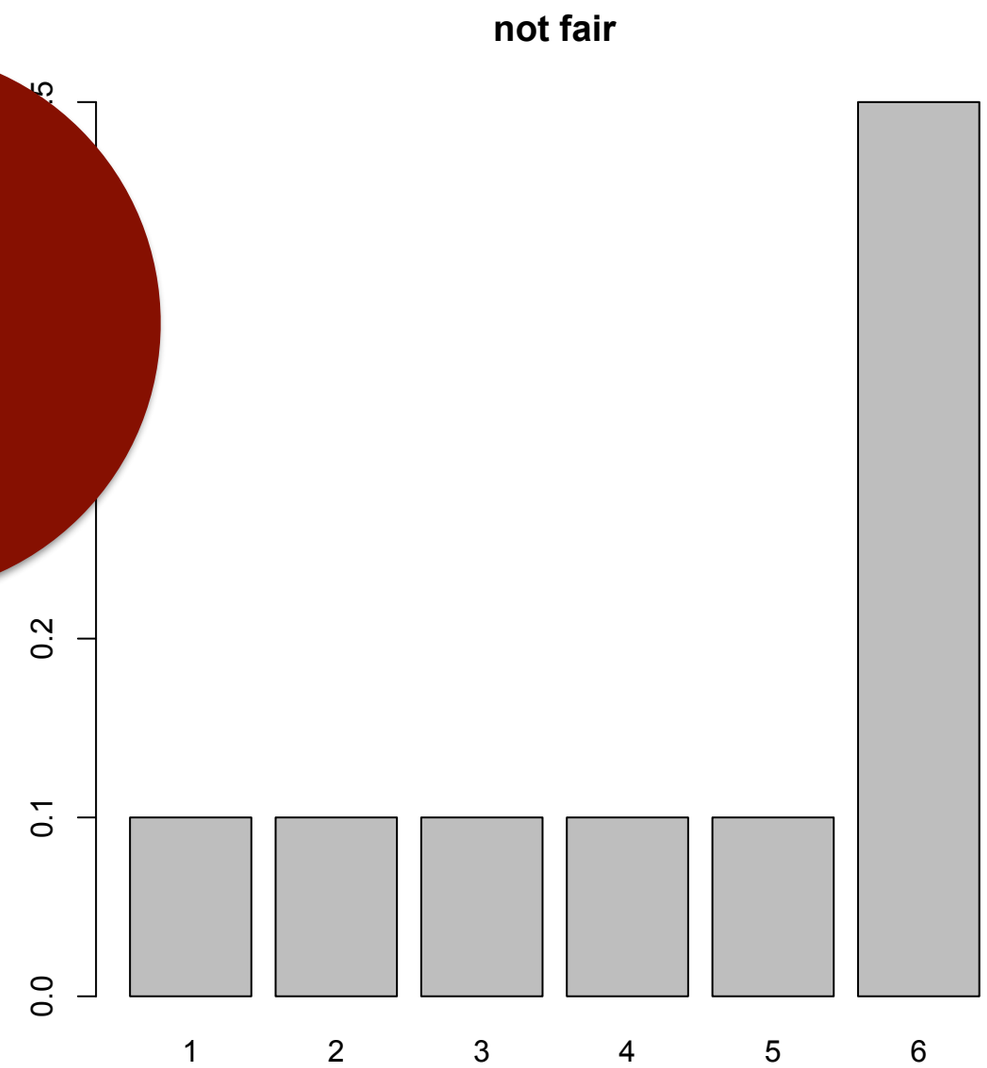
2

Probability

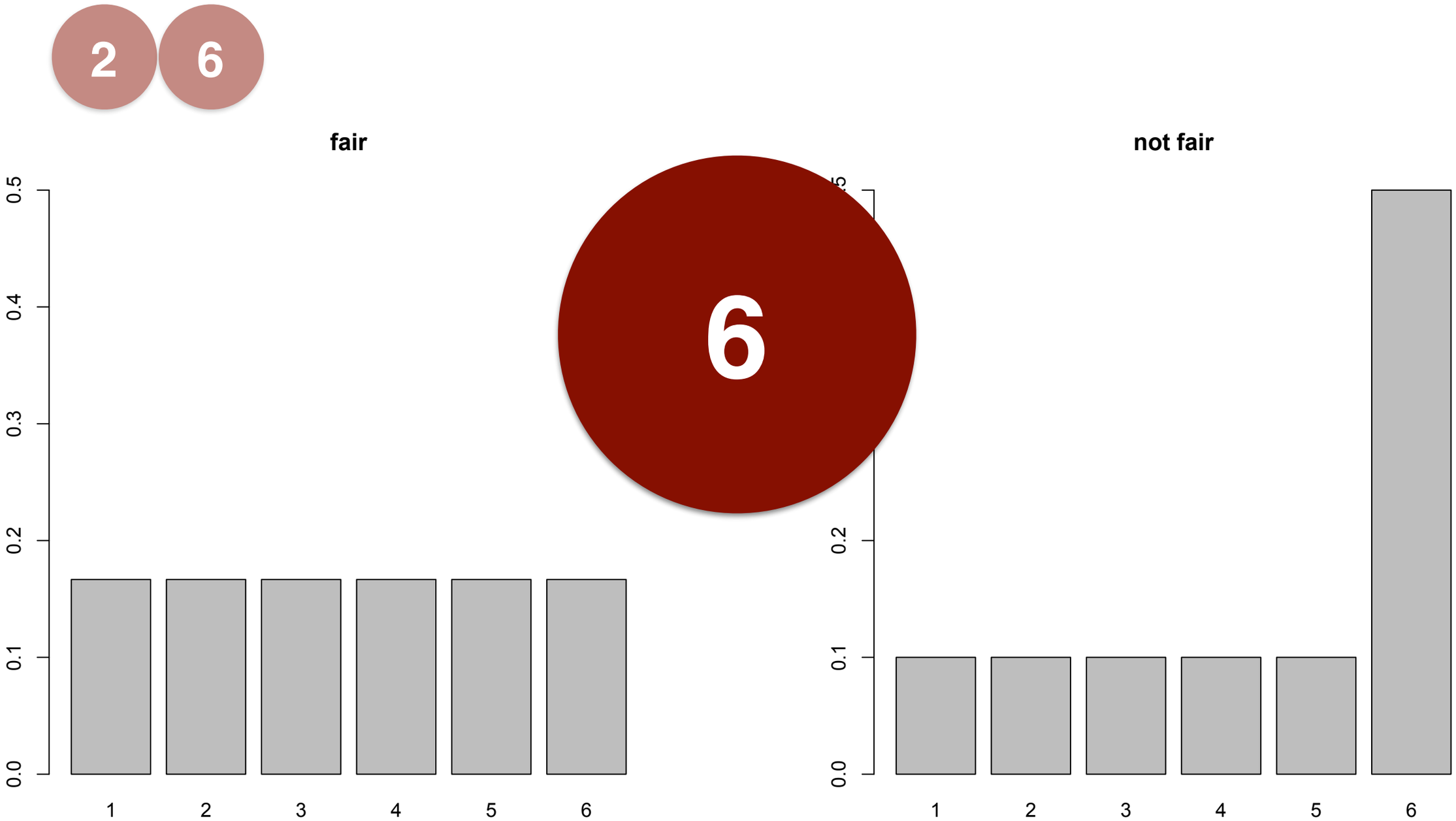
2



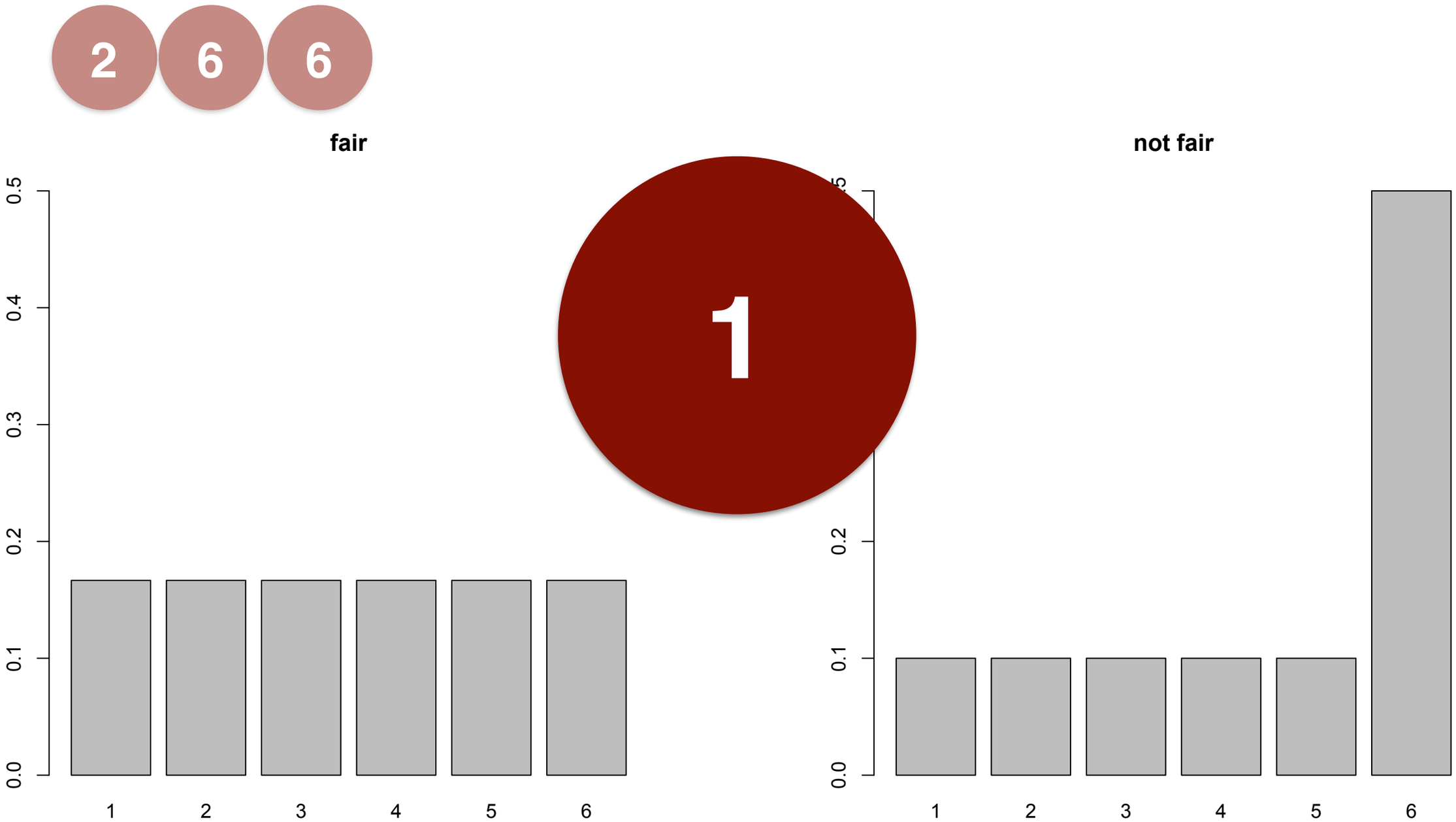
6



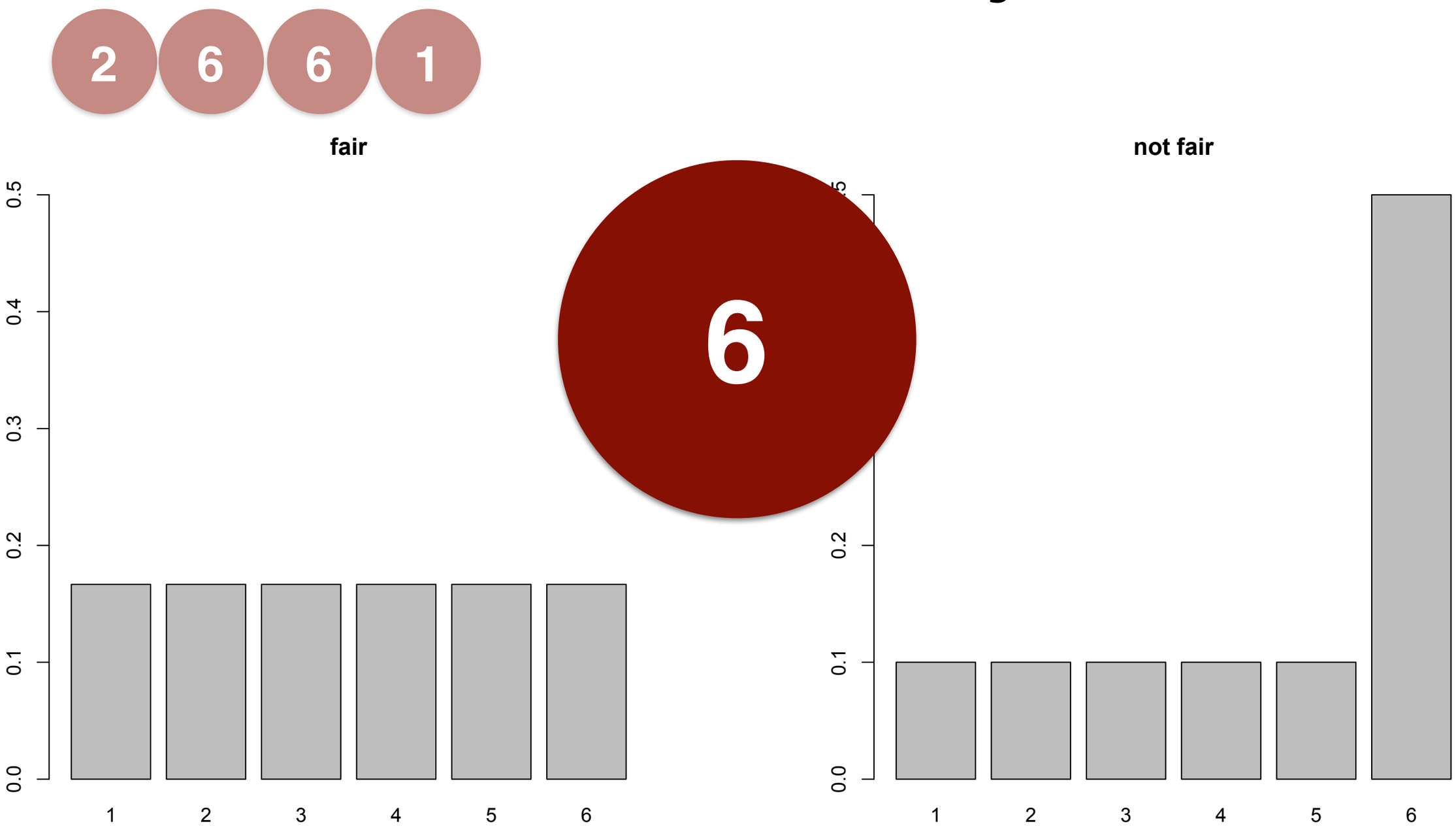
Probability



Probability

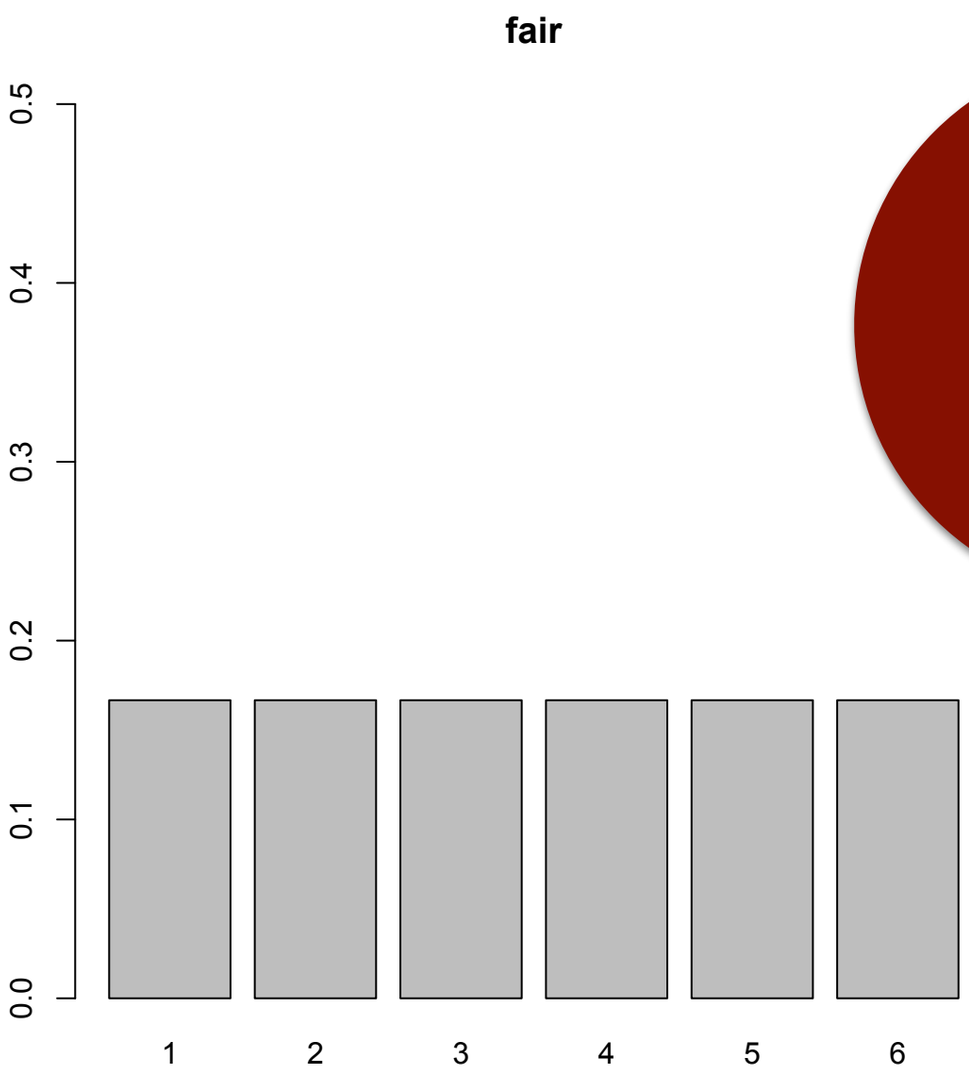


Probability

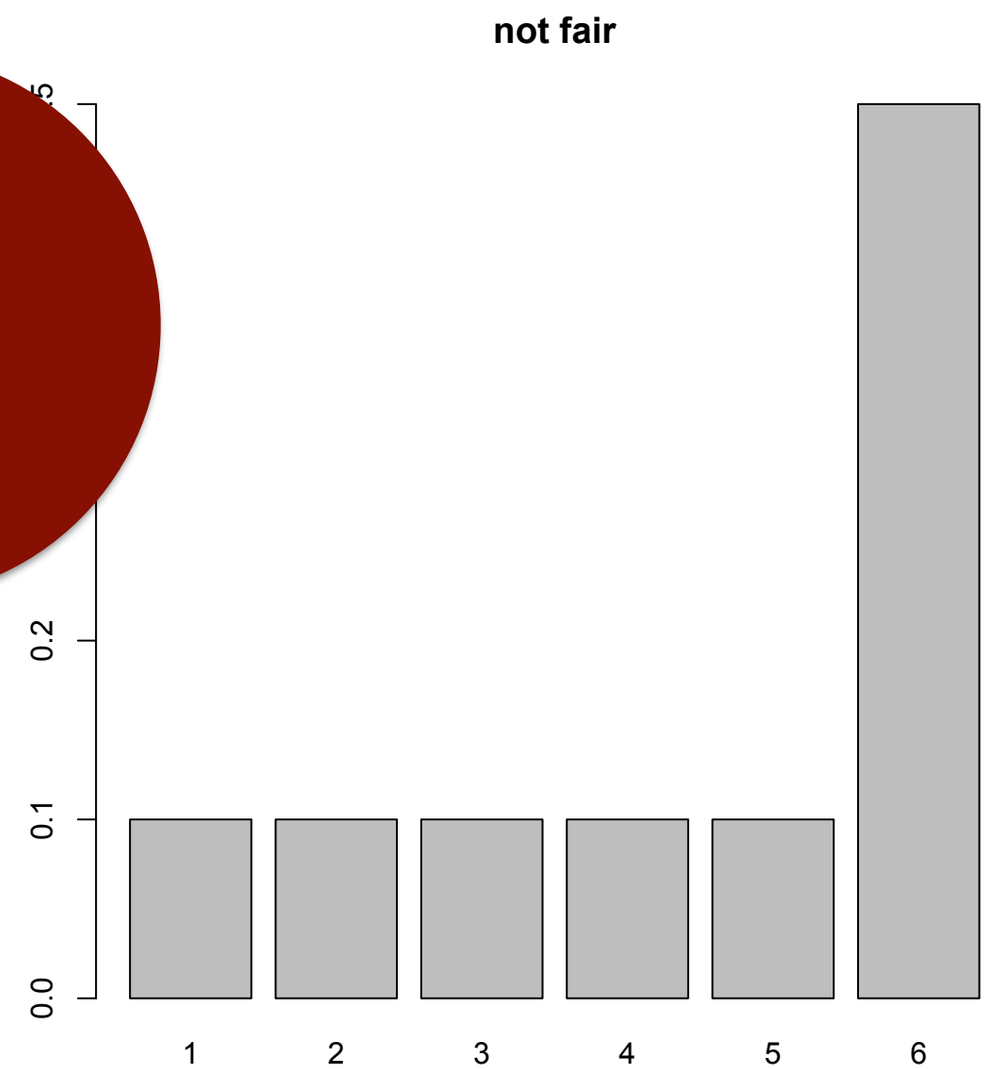


Probability

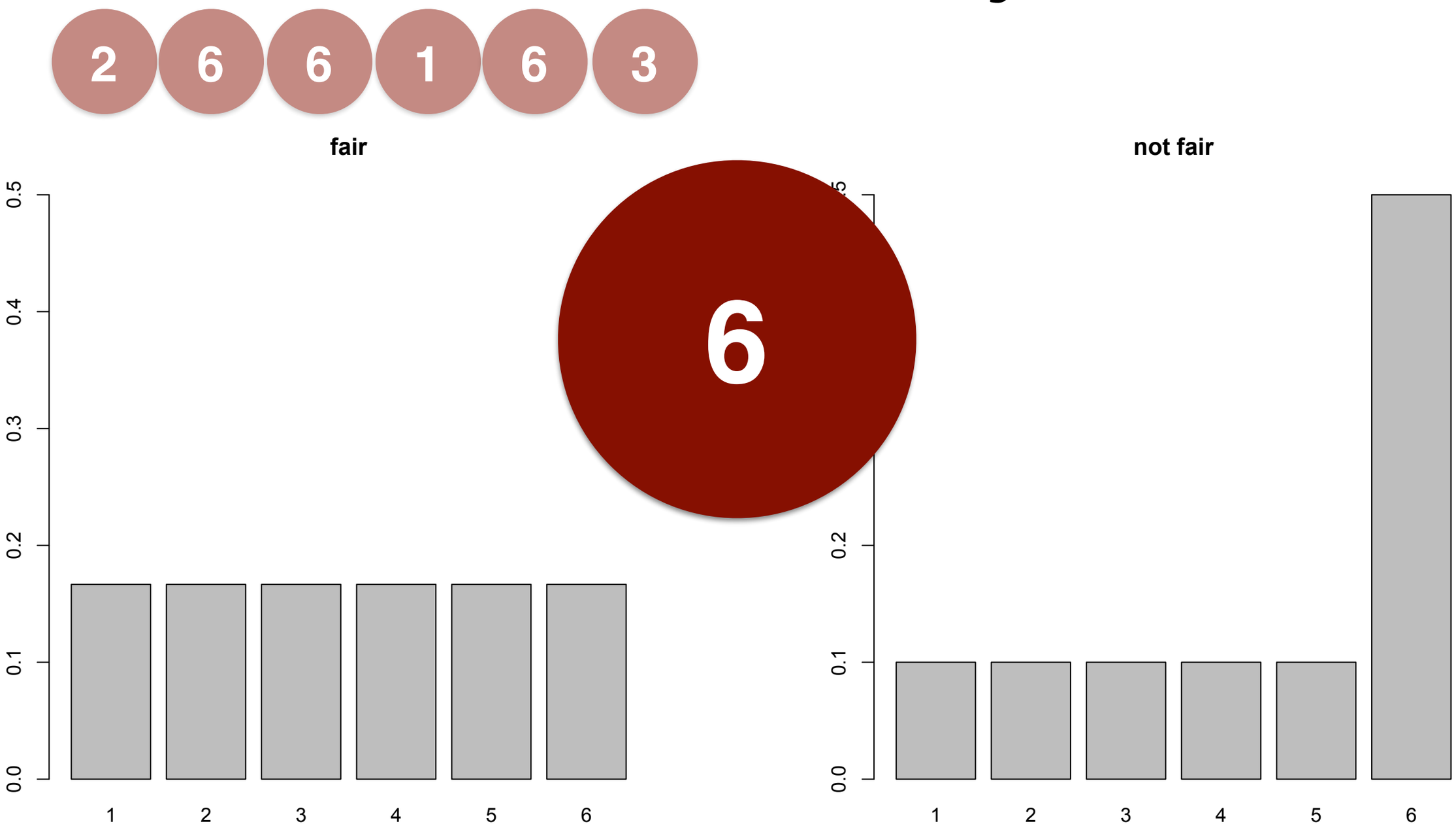
2 6 6 1 6



3

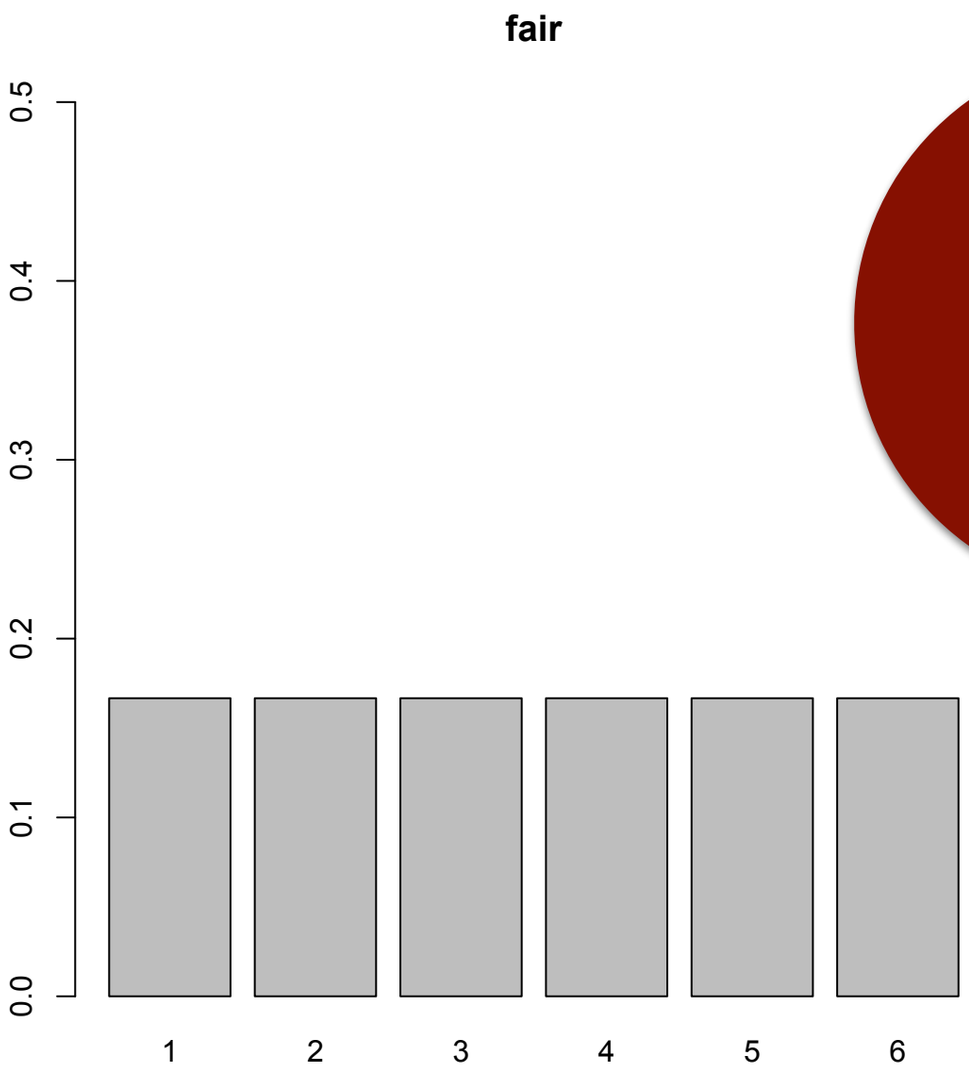


Probability

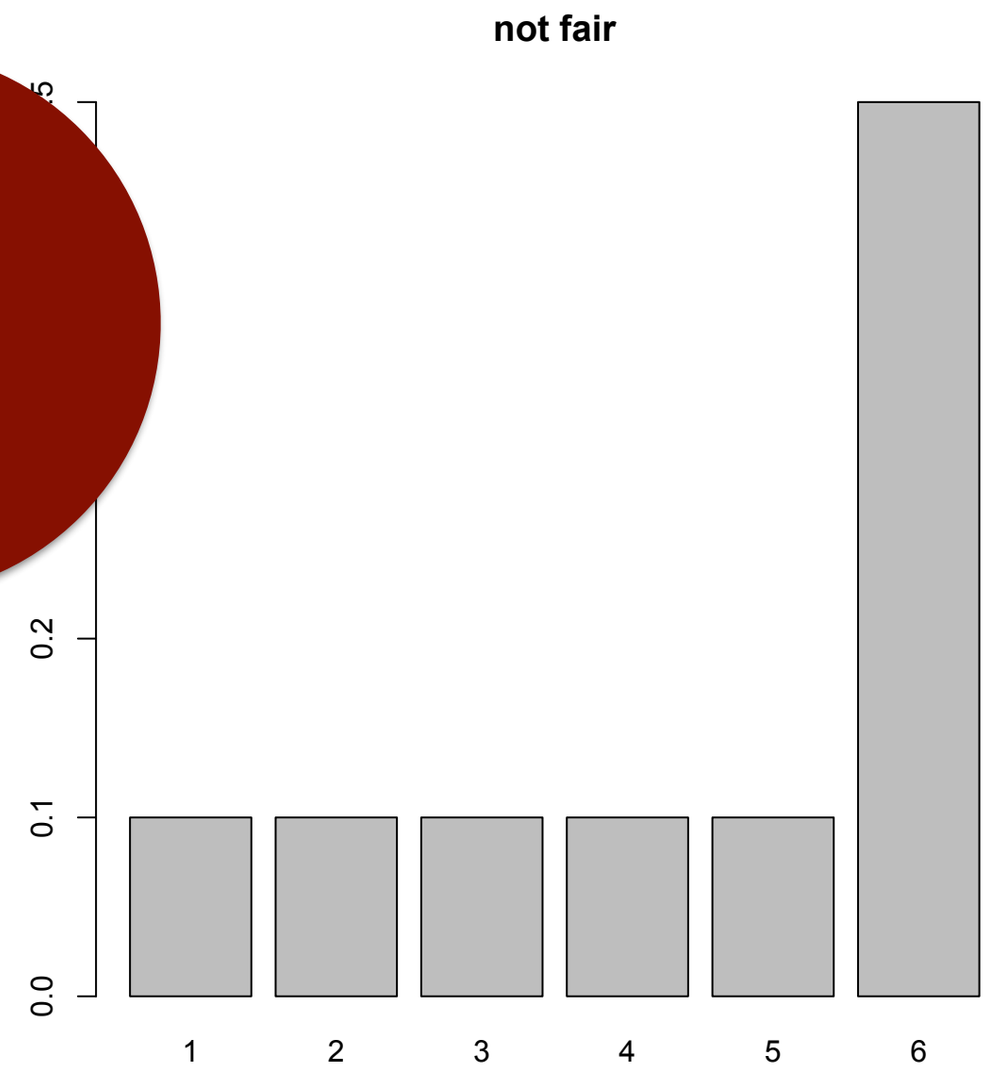


Probability

2 6 6 1 6 3 6

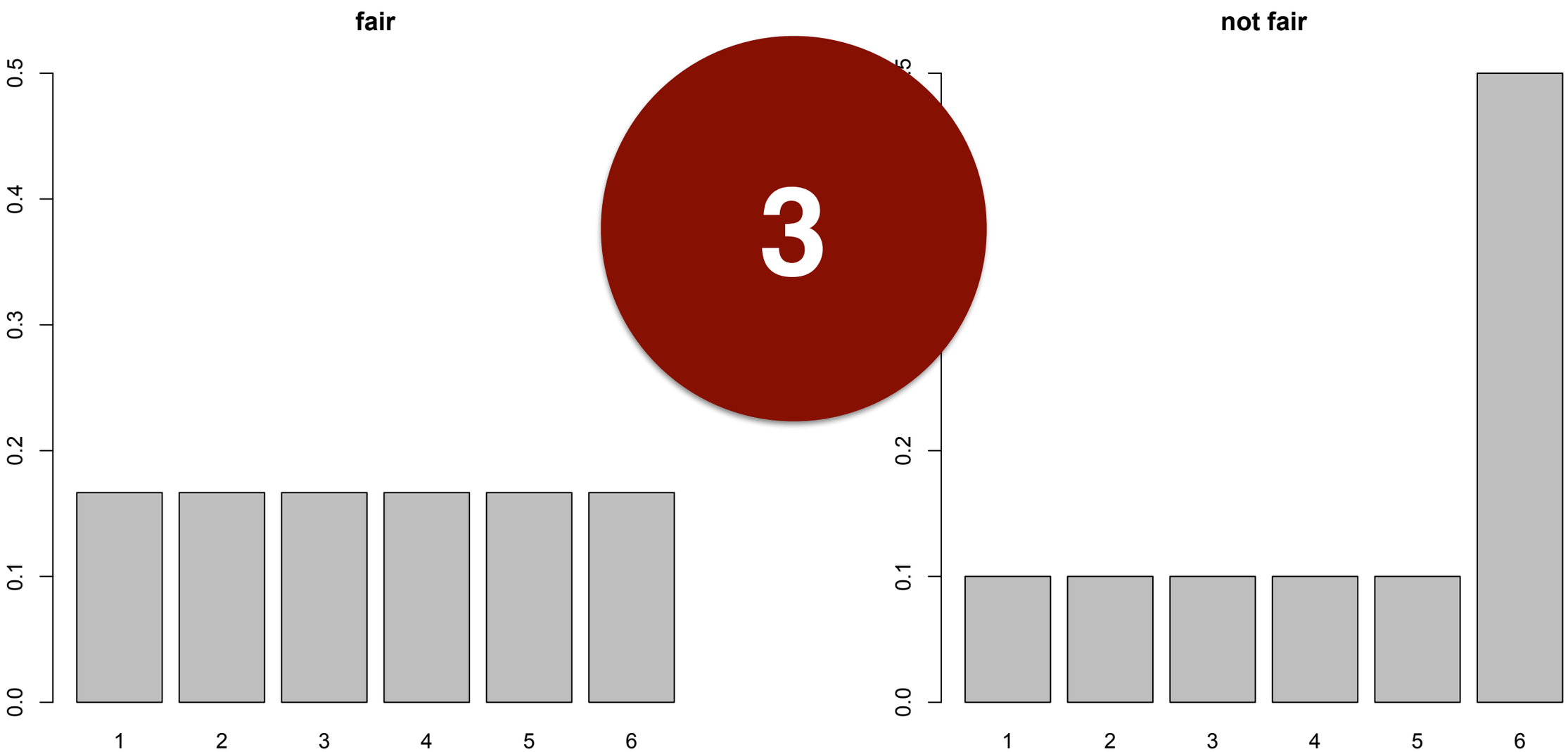


6



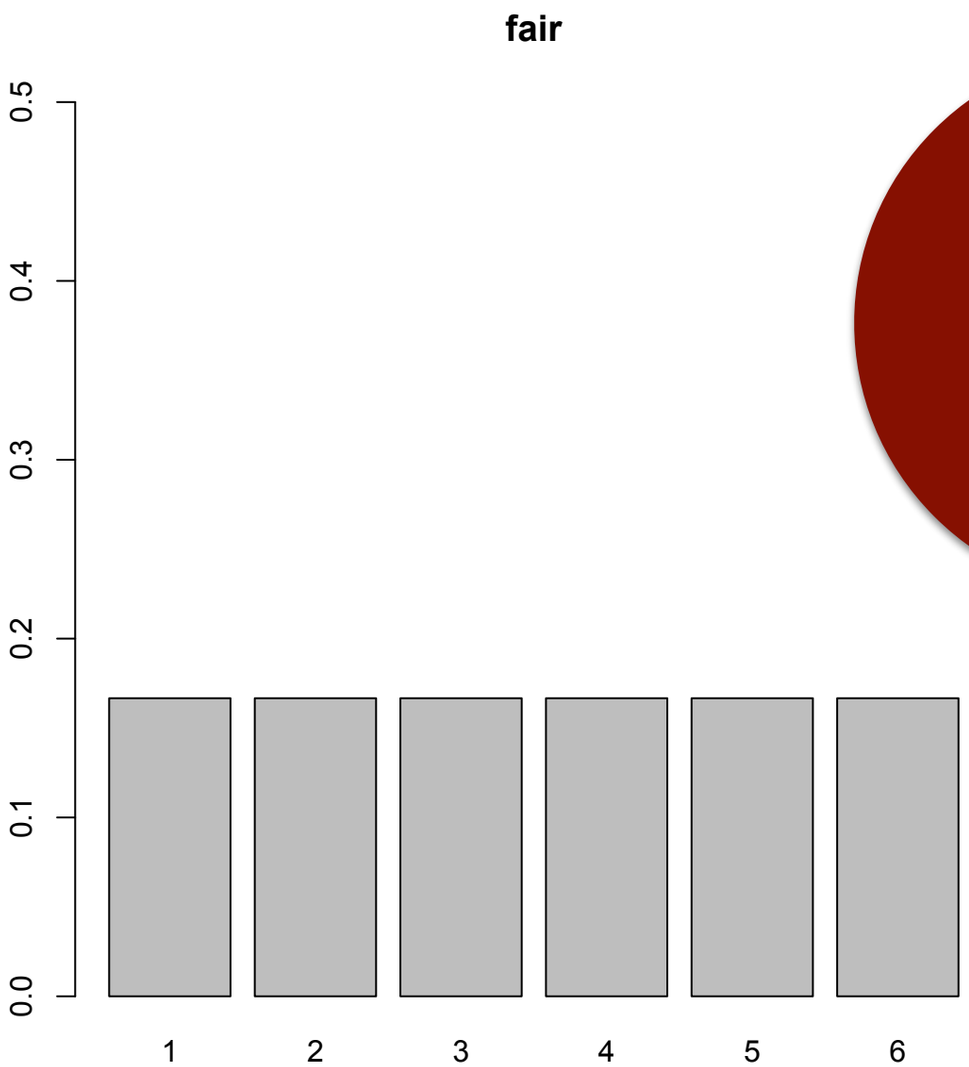
Probability

2 6 6 1 6 3 6 6

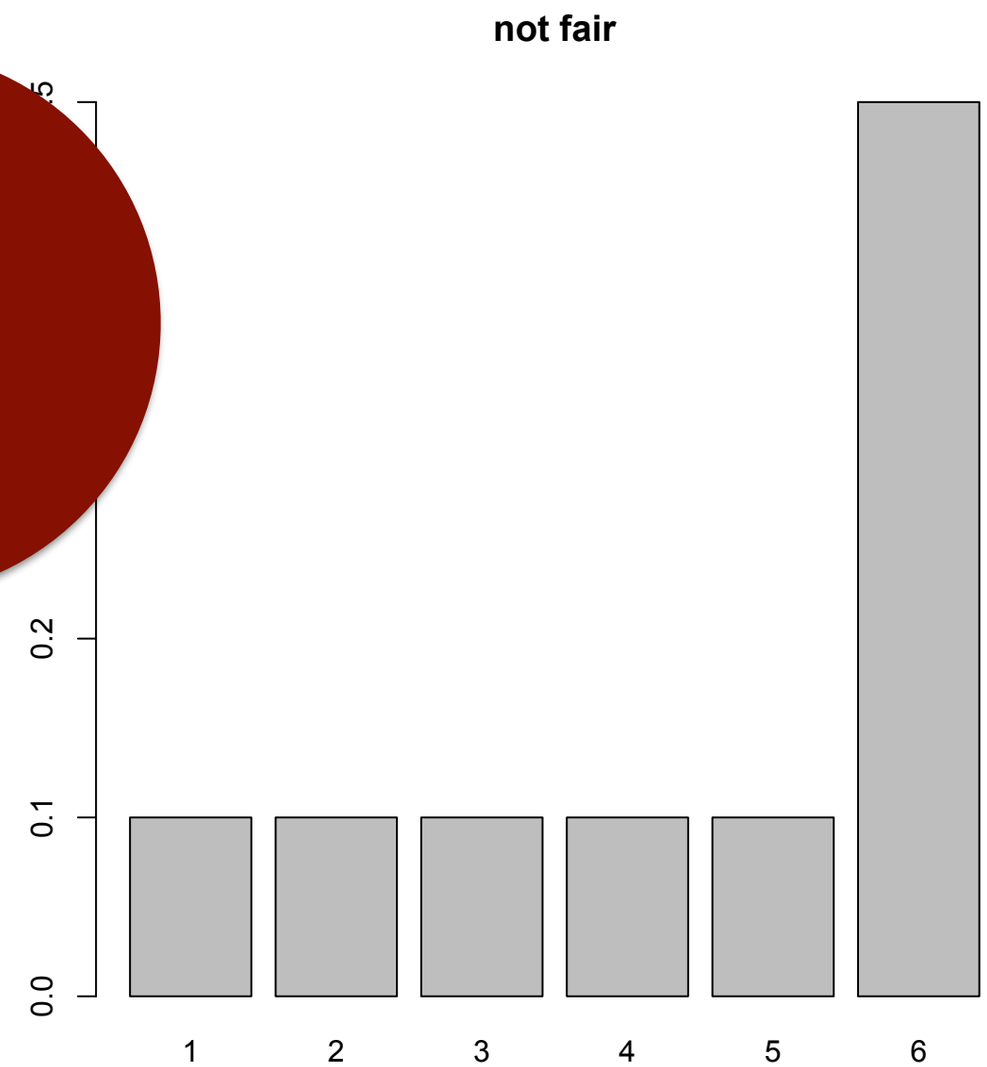


Probability

2 6 6 1 6 3 6 6 3



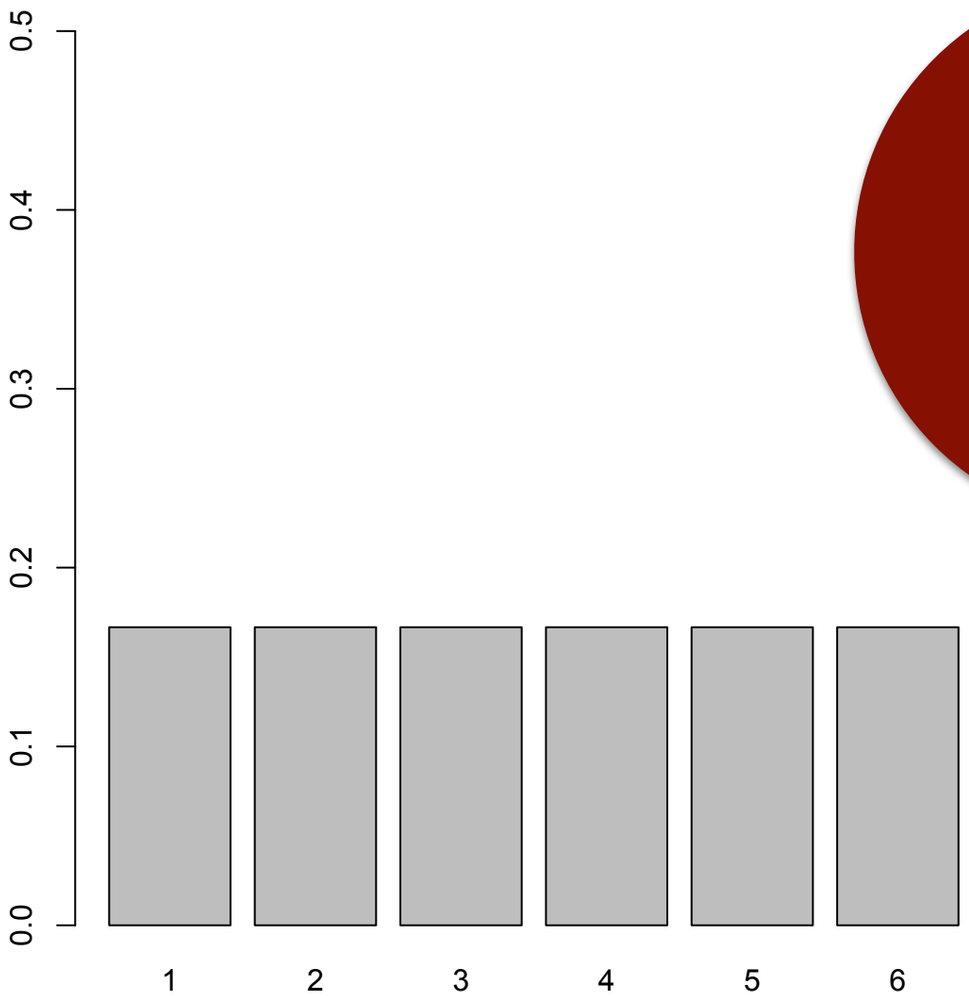
6



Probability

2 6 6 1 6 3 6 6 3 6

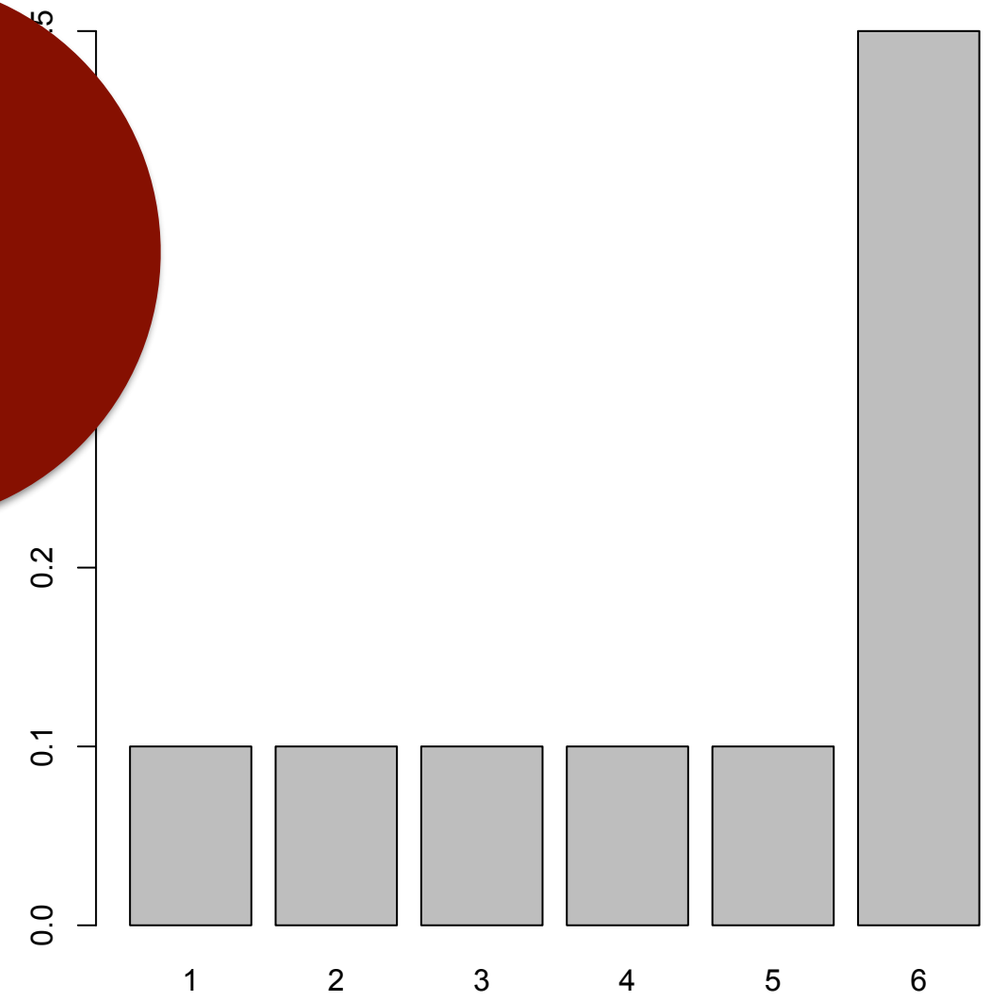
fair



1

?

not fair



15,625

Independence

- Two random variables are independent if:

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

- In general:

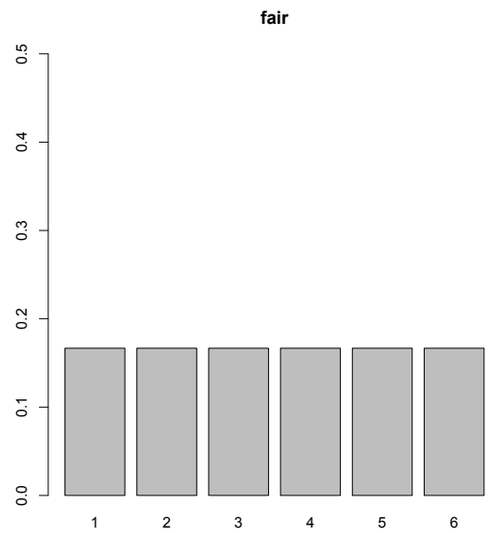
$$P(x_1, \ldots, x_n) = \prod_{i=1}^{N} P(x_i)$$

- 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

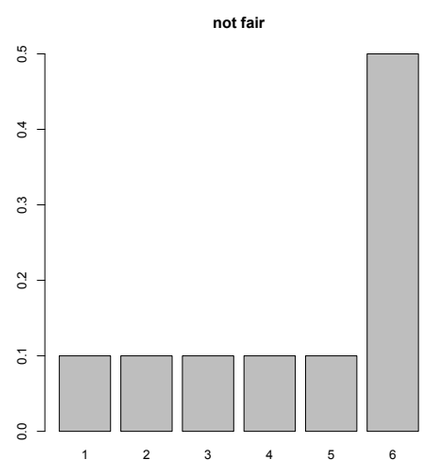
P(2 6 6 |



)

=.17 x .17 x .17
= 0.004913

P(2 6 6 |



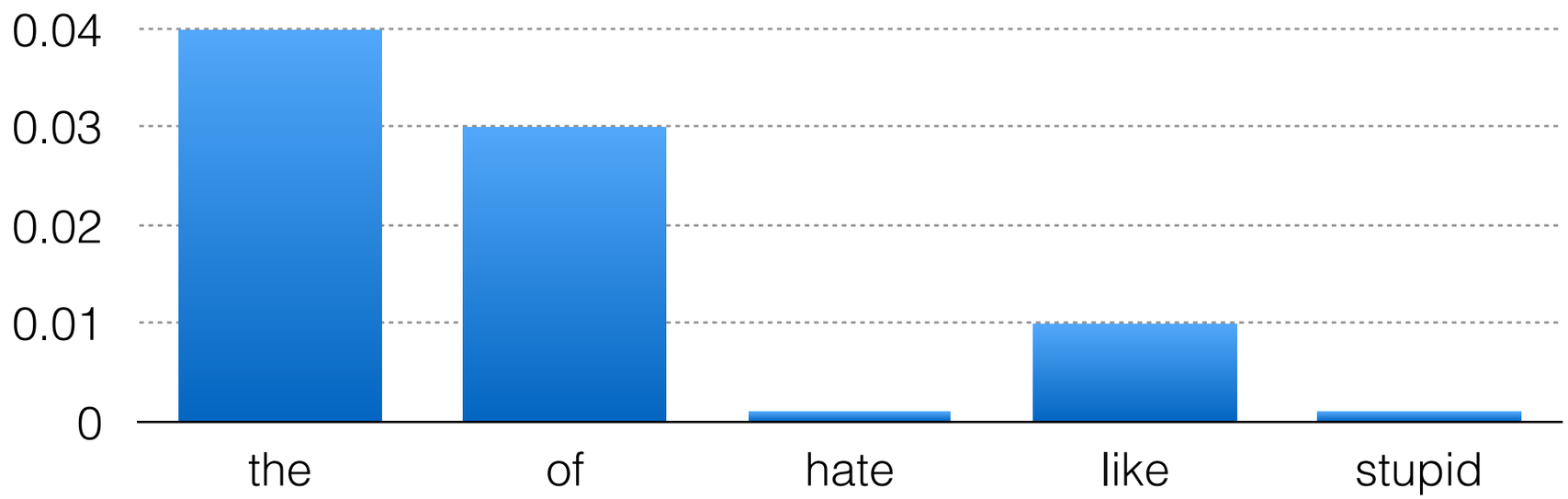
)

= .1 x .5 x .5
= 0.025

Data Likelihood

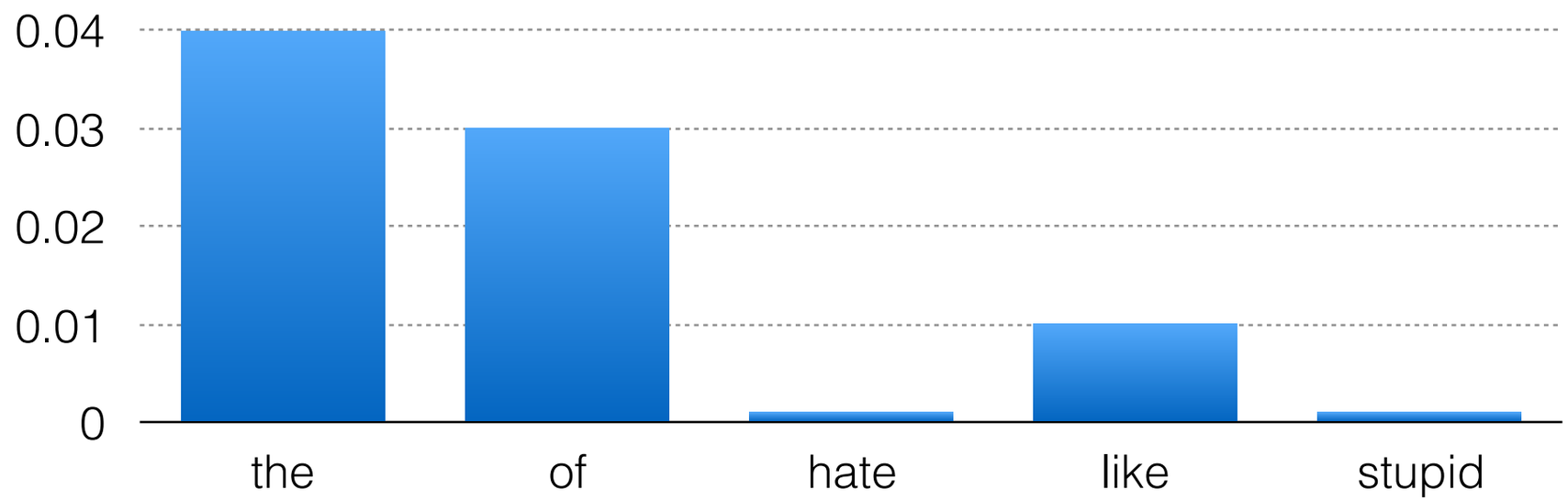
- The likelihood gives us a way of discriminating between possible alternative parameters, but also a strategy for picking a single best* parameter among all possibilities

Word choice as weighted dice

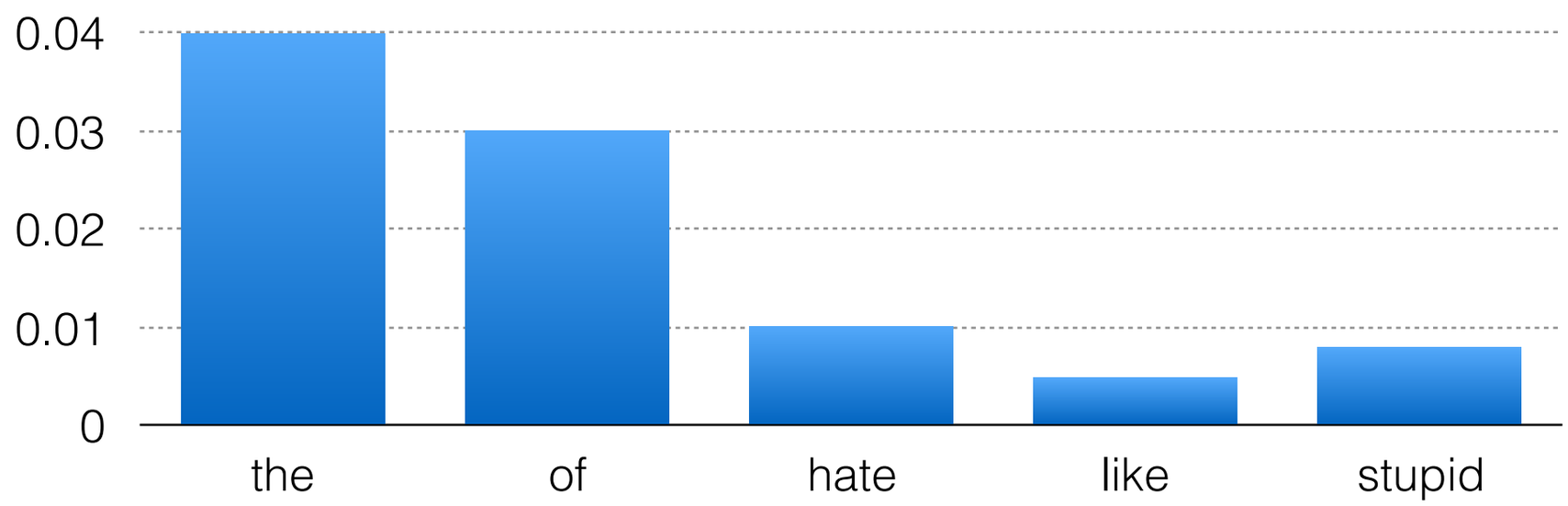


Unigram probability

positive reviews



negative reviews



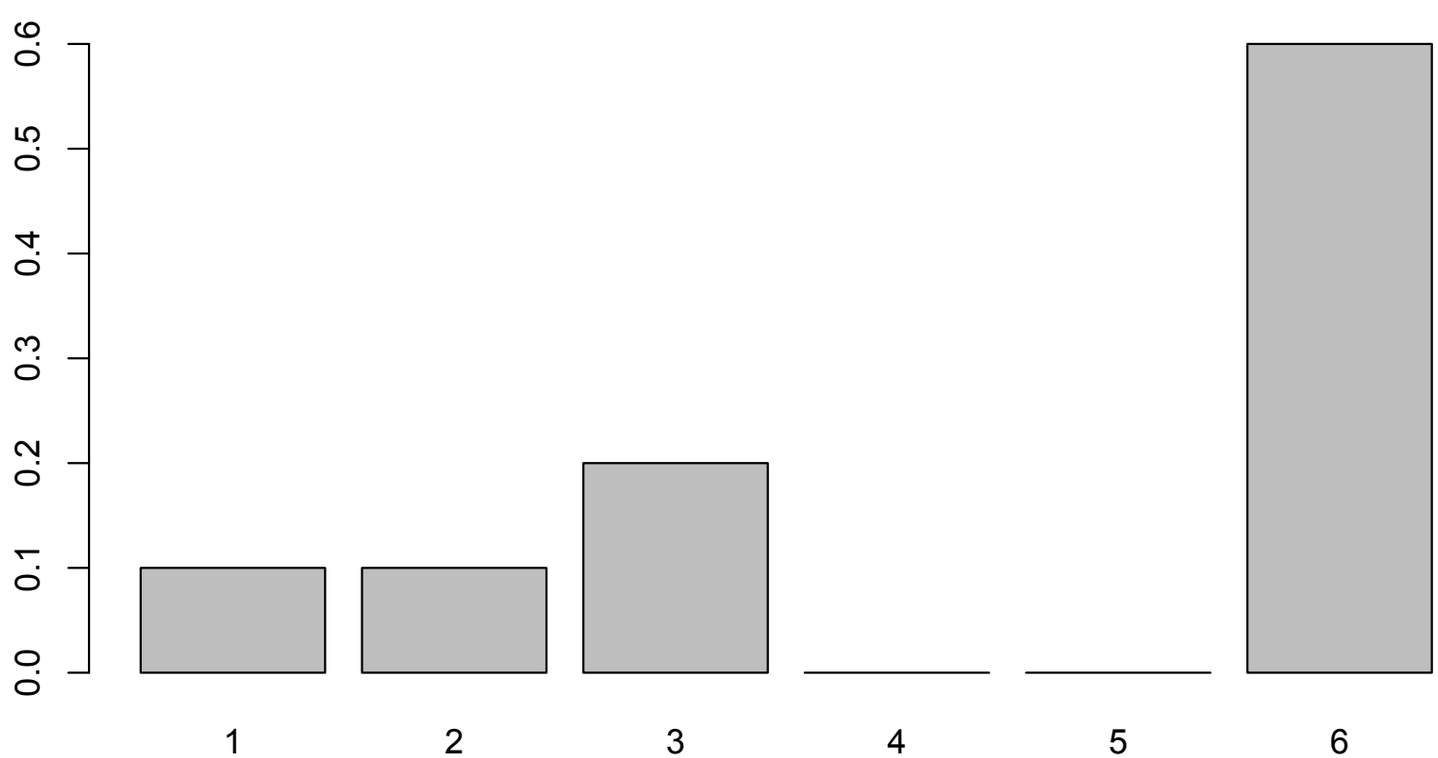
$$P(X = the) = \frac{\#the}{\#\text{total words}}$$

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.

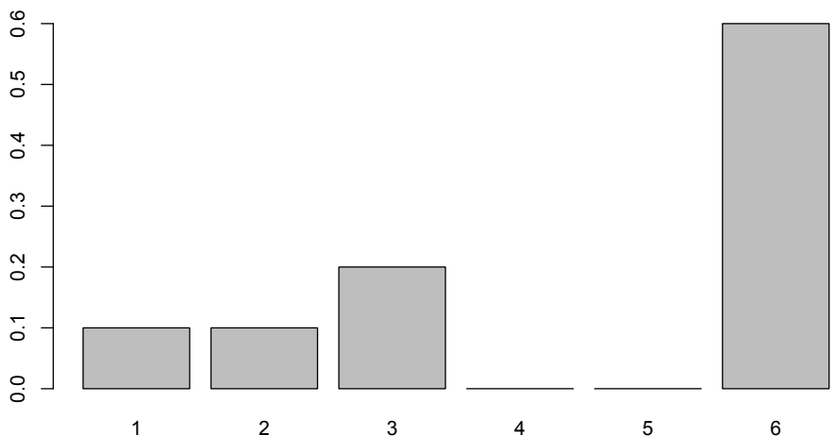
Maximum Likelihood Estimate

2 6 6 1 6 3 6 6 3 6



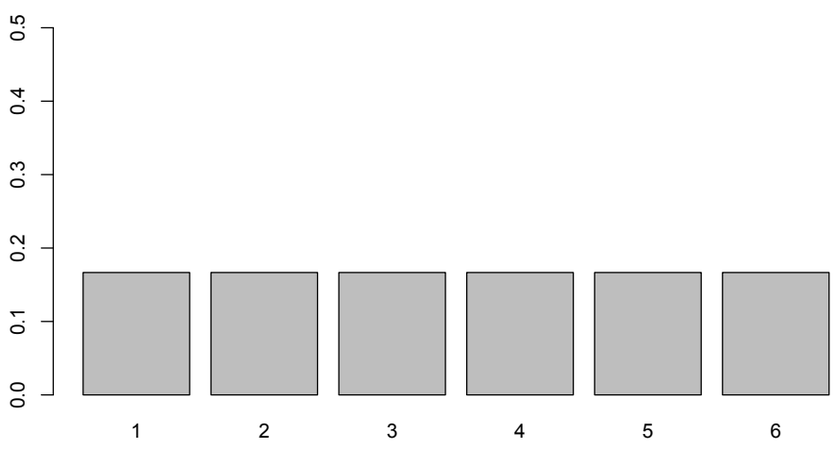
2 6 6 1 6 3 6 6 3 6

θ_1



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

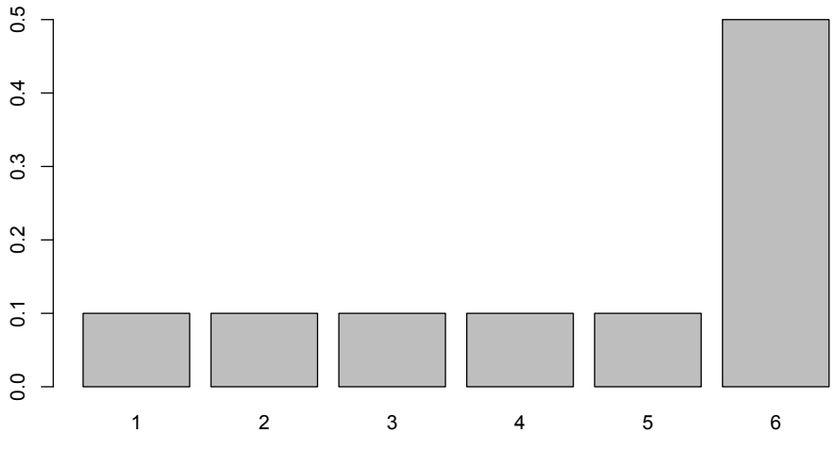
θ_2



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

(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

x_1 x_2 x_3 x_4 x_5 x_6

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=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

$$P(Y = y)$$

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

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

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

Let's say P(Y=⊕) = P(Y=⊖) = 0.5

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

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

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

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

Naive Bayes Classifier

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

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

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

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

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

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

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

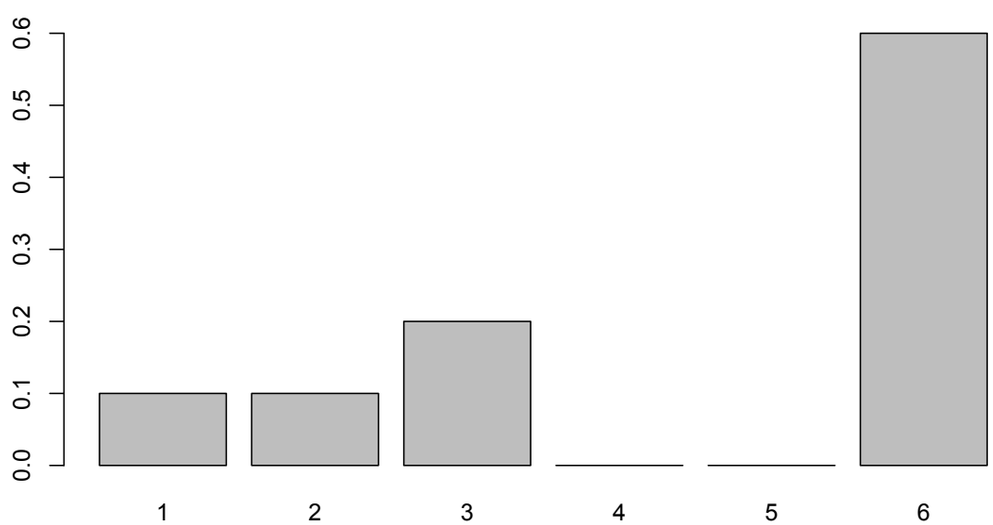
Priors

- 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

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

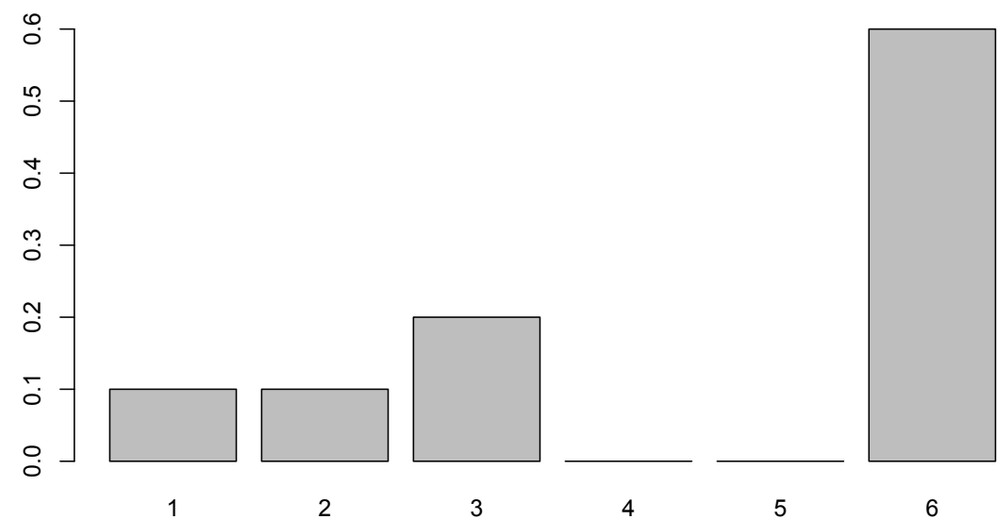
same α for all x_i

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

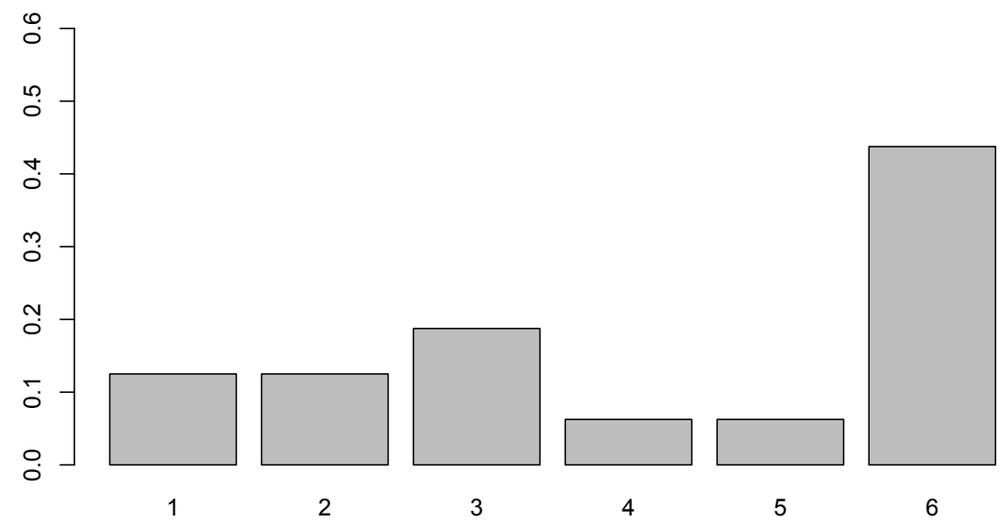
possibly different α for each x_i

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

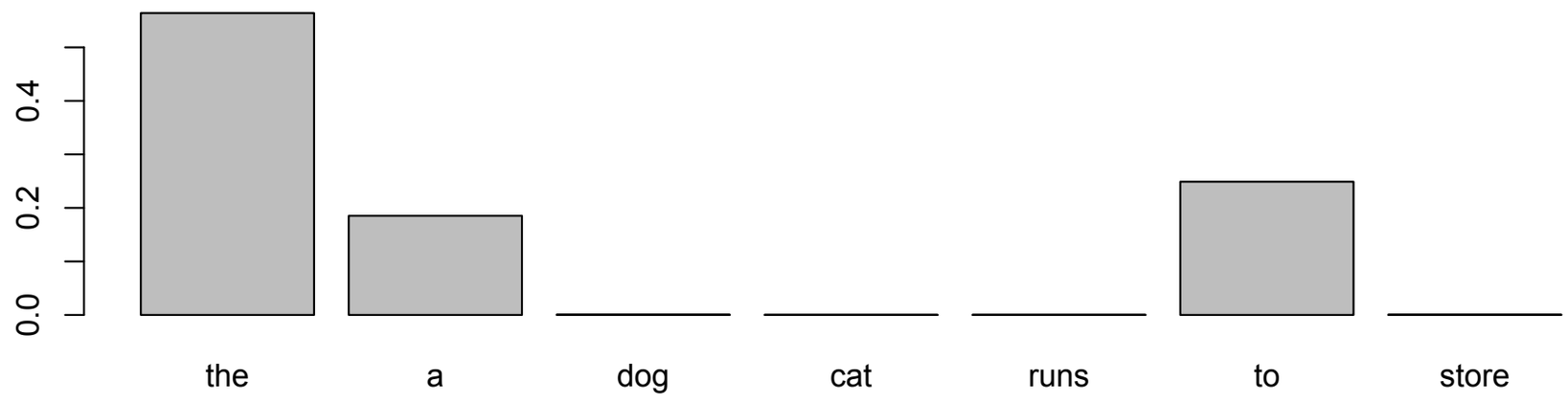
- Naive Bayes' independence assumption can be killer
- One instance of *hate* makes seeing others much more likely (each mention does contribute the same amount of information)
- We can mitigate this by not reasoning over counts of tokens but by their presence absence

	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: due 9/4

Annotate the sentiment by the writer toward the people and organizations mentioned