Probability and Random Processes

ECS 315

Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

1 Probability and You

Office Hours:

BKD, 6th floor of Sirindhralai building

Wednesday	14:30-15:30
Friday	14:30-15:30

Everything we do, everything that happens around us, obeys the laws of **probability**.

We can no more escape them than we can escape gravity… "Probability," a philosopher (Bishop Butler) once said, "is the **very guide of life**."

We are all gamblers who go through life making countless bets on the outcome of countless actions.

[Gardner, 1986]

Life is random

In 2005, this statement (which is true)

Life is random

was on display all over the world…

Life is random

Life is random.

[https://www.youtube.com/watch?v=0d8aVzQY4Zo]

Enjoy uncertainty.
Life is random.
Enjoy uncertainty.
Life is random.
Enjoy uncertainty.
240 songs. A million different ways. iPod shuffle
Give chance a chance.
Life is random.
Give chance a chance.
Life is random.
Life is random.
Meet iPod shuffle.
The unpredictable new member of the iPod family.
240 songs. A million different ways.
From $99. Mac + PC.
iPod shuffle
Enjoy uncertainty.

iPod shuffle

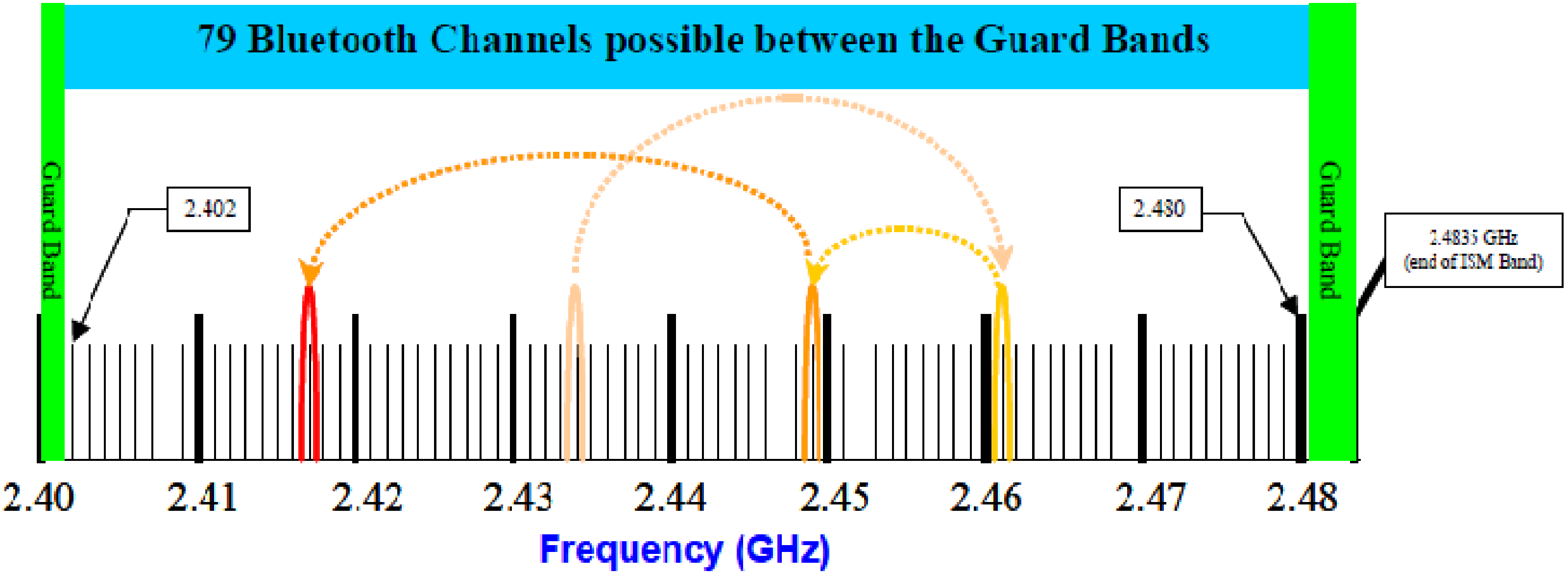
Life is random.

Applications of Probability Theory

- The subject of probability can be traced back to the 17th century when it arose out of the study of **gambling games**.
- The range of applications extends beyond games into business decisions, insurance, law, **medical tests**, and the social sciences.
- The **stock market**, "the largest casino in the world," cannot do without it.
- The **telephone network**, call centers, and airline companies with their randomly fluctuating loads could not have been economically designed without probability theory.

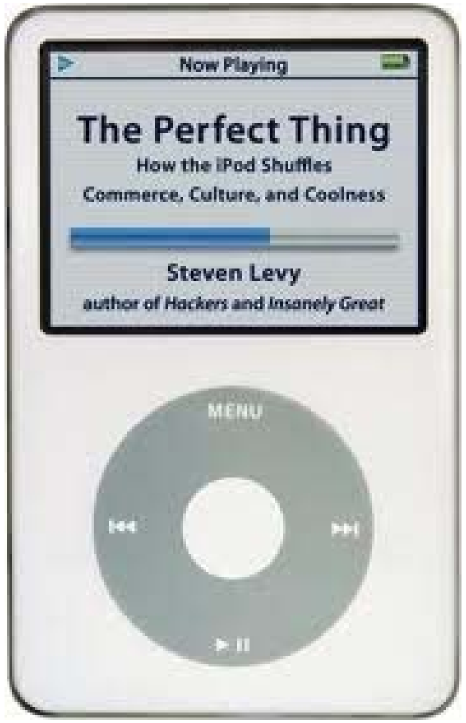
FHSS Example: Bluetooth

- The band at 2.4 GHz is divided into 79 channels.
- A Bluetooth device, hops frequency at a rate of 1600 hops per second, randomly selecting a channel of 1 MHz to operate.



[https://sites.google.com/site/nearcommunications/adaptative-frequency-hopping]

“The Perfect Thing”



Now Playing

The Perfect Thing

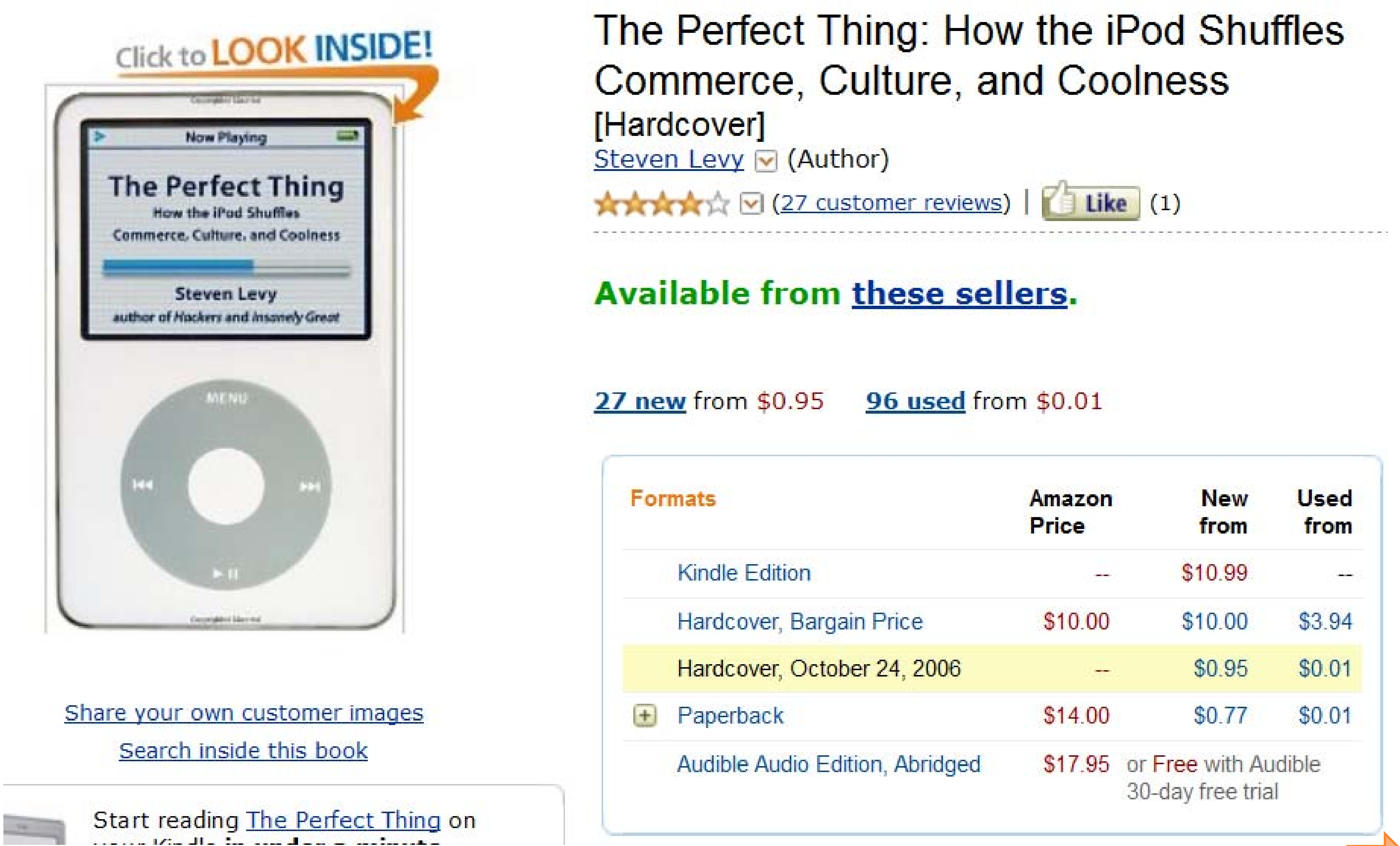
How the iPod Shuffles Commerce, Culture, and Coolness

Steven Levy

author of *Hackers* and *Insanely Great*

What is this?

“The Perfect Thing”



Share your own customer images

Search inside this book

Start reading The Perfect Thing on

The Perfect Thing: How the iPod Shuffles Commerce, Culture, and Coolness [Hardcover]

Steven Levy (Author)

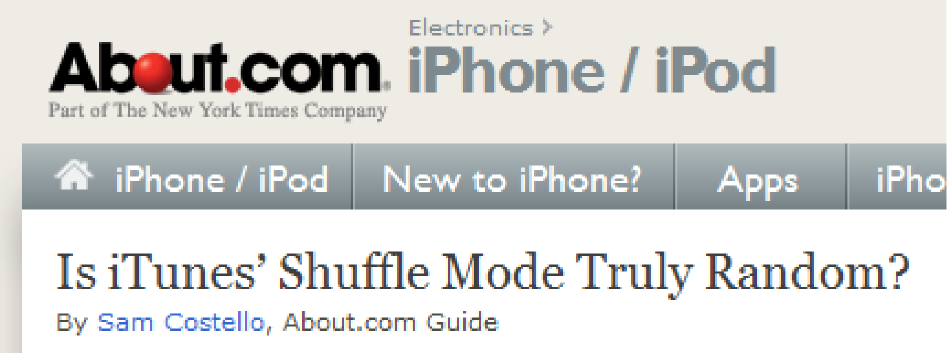
★★★★☆ (27 customer reviews) | Like (1)

Available from these sellers.

27 new from $0.95 **96 used** from $0.01

Formats	Amazon Price	New from	Used from
Kindle Edition	--	$10.99	--
Hardcover, Bargain Price	$10.00	$10.00	$3.94
Hardcover, October 24, 2006	--	$0.95	$0.01
⊞ Paperback	$14.00	$0.77	$0.01
Audible Audio Edition, Abridged	$17.95	or Free with Audible 30-day free trial	

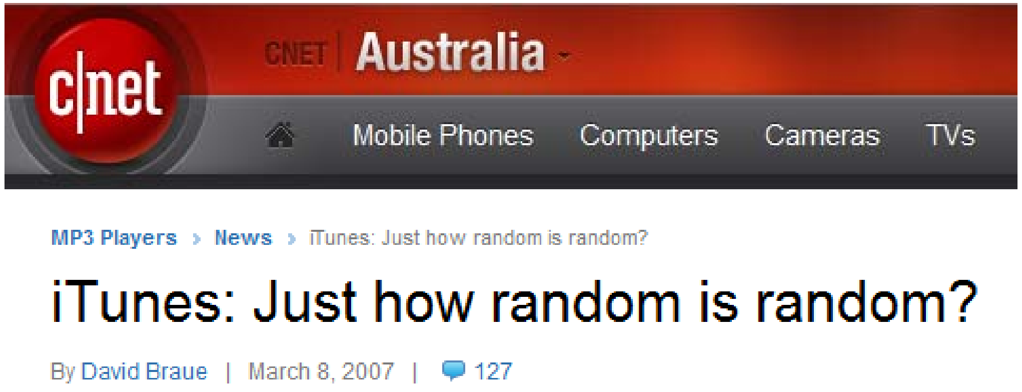
What about the shuffle function?



Is iTunes' Shuffle Mode Truly Random?

By Sam Costello, About.com Guide

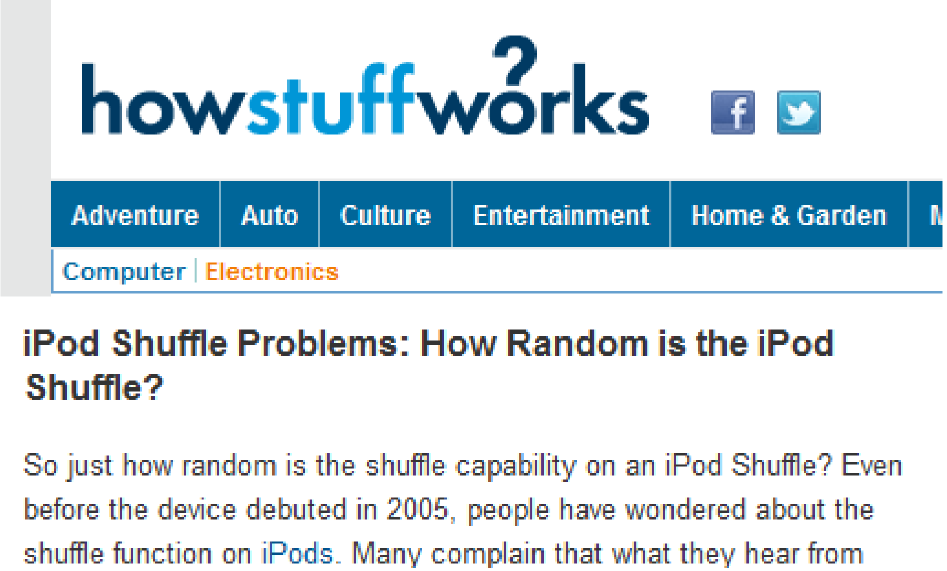
http://ipod.about.com/od/advanceditunesuse/a/itunes-random.htm



iTunes: Just how random is random?

By David Braue | March 8, 2007 | 127

http://www.cnet.com.au/itunes-just-how-random-is-random-339274094.htm



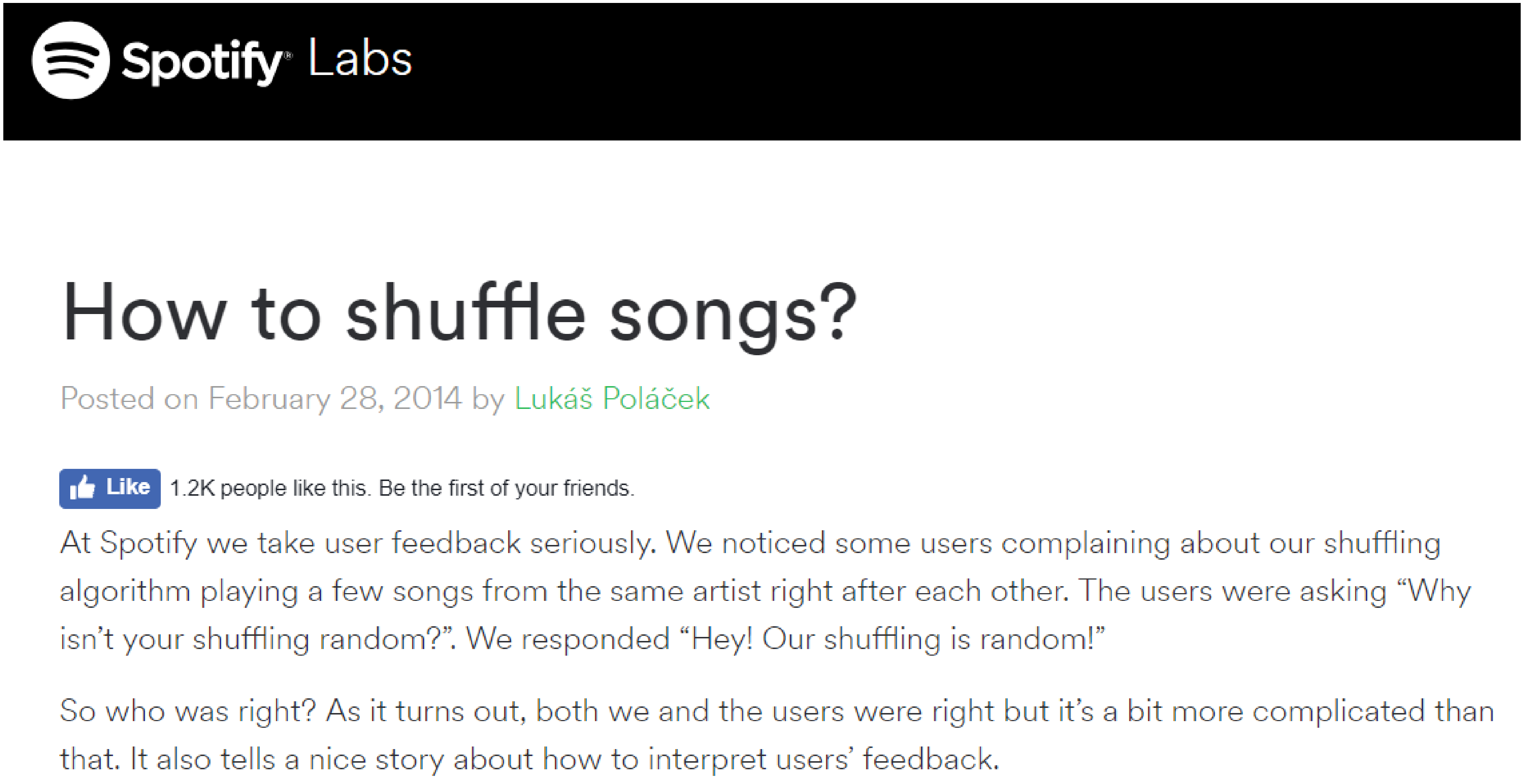
iPod Shuffle Problems: How Random is the iPod Shuffle?

So just how random is the shuffle capability on an iPod Shuffle? Even before the device debuted in 2005, people have wondered about the shuffle function on iPods. Many complain that what they hear from

http://electronics.howstuffworks.com/ipod-shuffle2.htm

What about the shuffle function?

[https://www.youtube.com/watch?v=W-wHXfHxRUA]



Spotify Labs

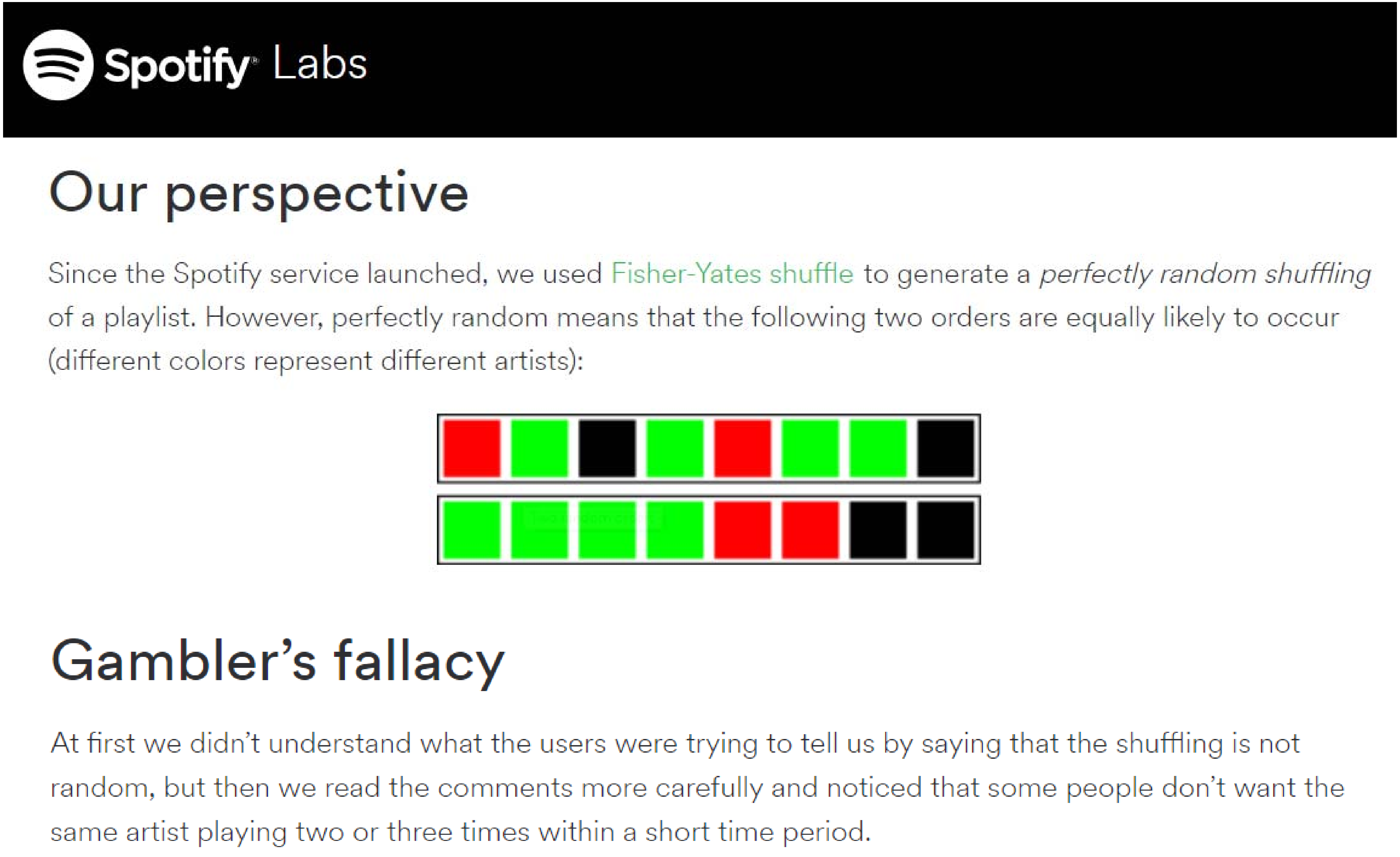
How to shuffle songs?

Posted on February 28, 2014 by Lukáš Poláček

At Spotify we take user feedback seriously. We noticed some users complaining about our shuffling algorithm playing a few songs from the same artist right after each other. The users were asking “Why isn’t your shuffling random?”. We responded “Hey! Our shuffling is random!”

So who was right? As it turns out, both we and the users were right but it’s a bit more complicated than that. It also tells a nice story about how to interpret users’ feedback.

[https://labs.spotify.com/2014/02/28/how-to-shuffle-songs/]



Spotify Labs

Our perspective

Since the Spotify service launched, we used Fisher-Yates shuffle to generate a *perfectly random shuffling* of a playlist. However, perfectly random means that the following two orders are equally likely to occur (different colors represent different artists):

🟥🟩⬛🟩🟥🟩🟩⬛

🟩🟩🟩🟩🟥🟥⬛⬛

Gambler's fallacy

At first we didn't understand what the users were trying to tell us by saying that the shuffling is not random, but then we read the comments more carefully and noticed that some people don't want the same artist playing two or three times within a short time period.

[https://labs.spotify.com/2014/02/28/how-to-shuffle-songs/]

USA Currency Coins

- Penny = 1 cent (Abraham Lincoln)

- Nickel = 5 cents (Thomas Jefferson)

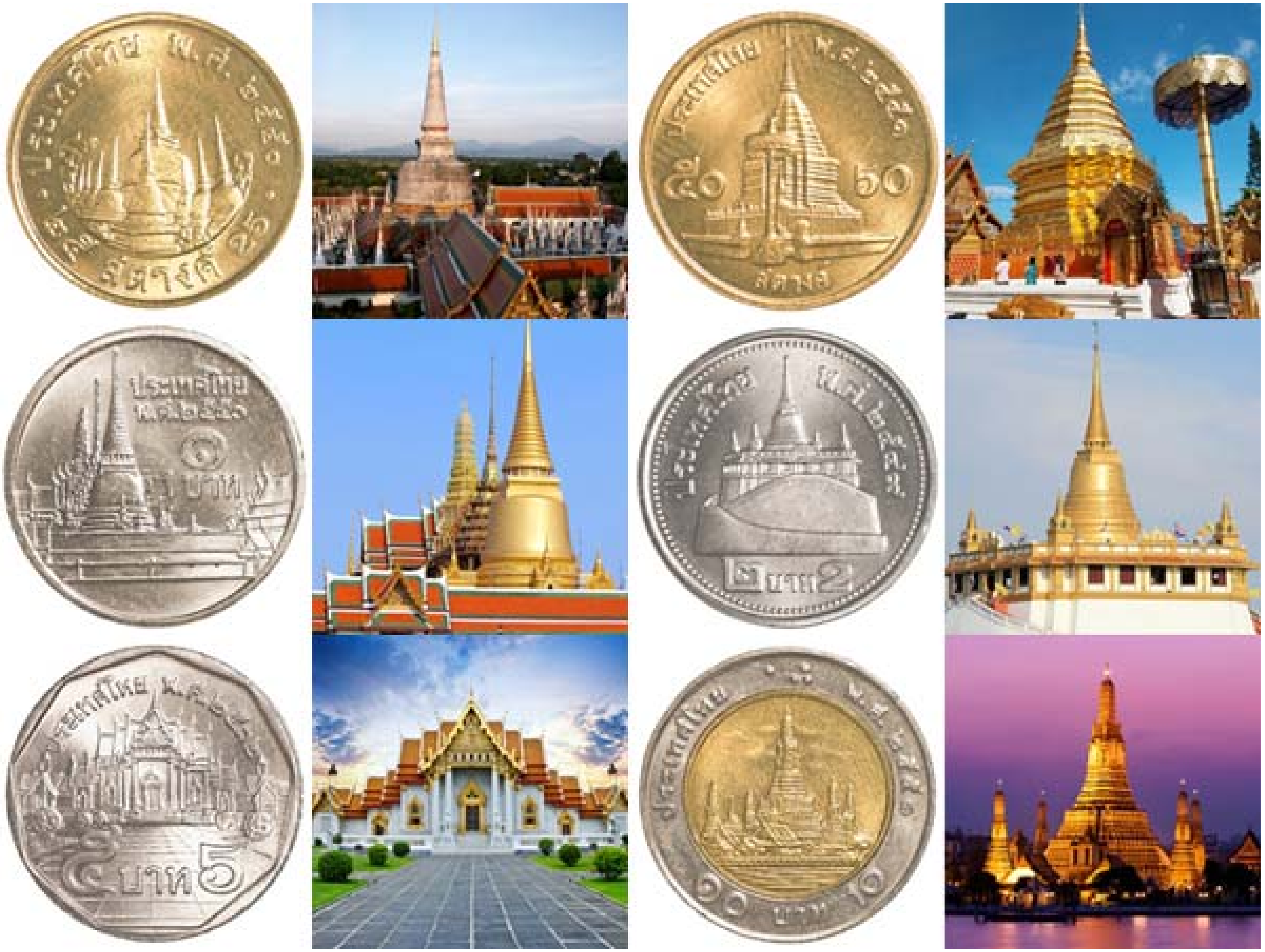
- Dime = 10 cents (Franklin D. Roosevelt)

- Quarter = 25 cents (George Washington)

Thai Coins

Front

Back



[http://travel.kapook.com/view75126.html]

`randi` function

- Generate uniformly distributed **pseudorandom integers**
- `randi(imax)` returns a scalar value between 1 and `imax`.
- `randi(imax,m,n)` and `randi(imax,[m,n])` return an *m*-by-*n* matrix containing pseudorandom integer values drawn from the discrete uniform distribution on the interval [1,`imax`].
  - `randi(imax)` is the same as `randi(imax,1)`.
- `randi([imin,imax],...)` returns an array containing integer values drawn from the discrete uniform distribution on the interval [`imin`,`imax`].

randi function: examples

Coin Tosses:

```
>> randi([0,1])
ans =        T, H
     0
>> randi([0,1],10,2)
ans =
     1     0
     1     0
     1     0
     1     1
     1     1
     0     0
     1     1
     0     0
     1     0
     0     0
```

Dice Rolls

```
>> randi([1,6])
ans =
     5
>> randi([1,6],10,2)
ans =
     5     1
     2     1
     3     3
     3     6
     4     3
     5     4
     5     2
     2     5
     5     2
     4     4
```

`randi` function: examples

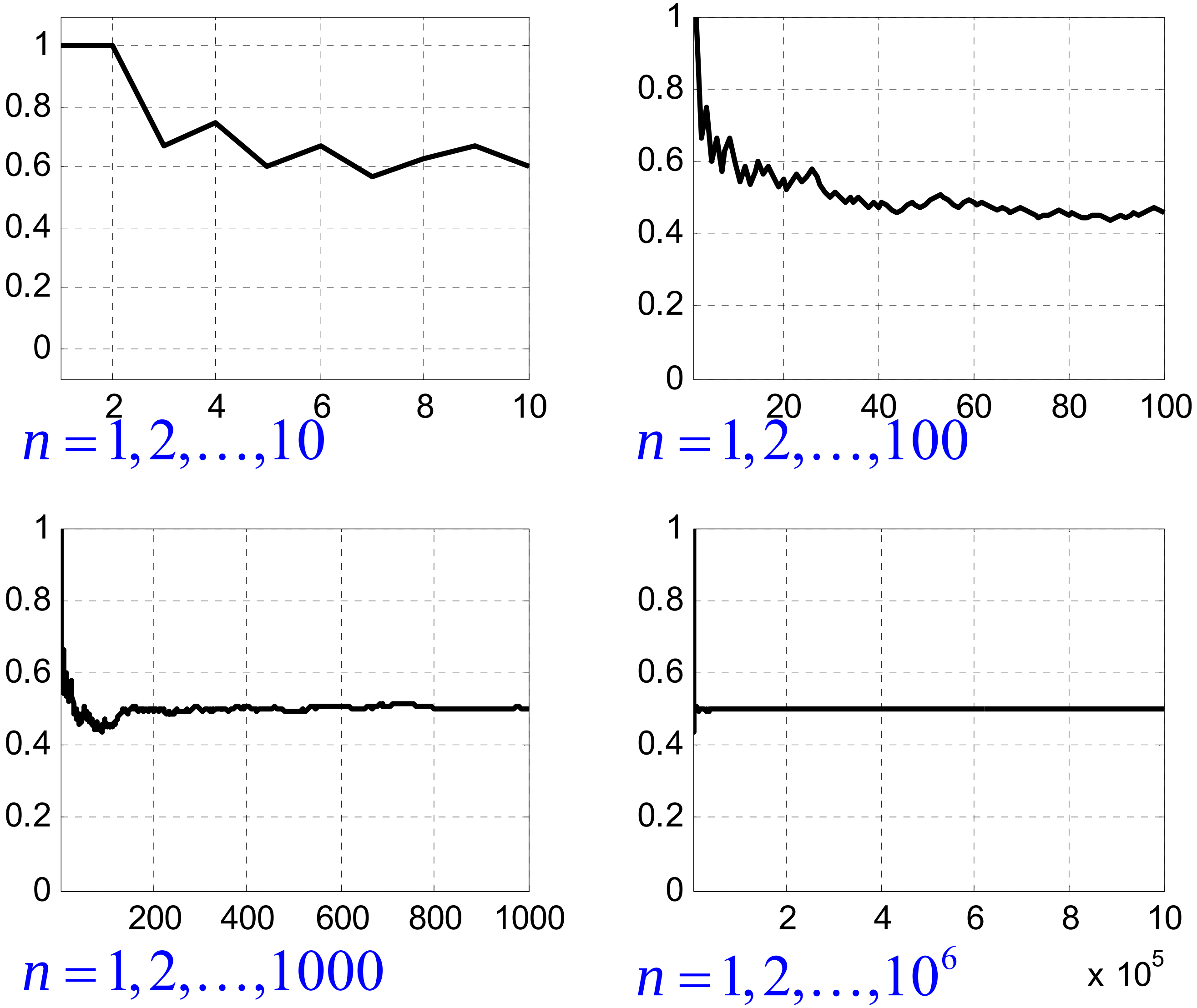
Coin Tosses:

```
>> S = ['T','H']
S =
TH
>> S(randi([1,2]))
ans =
H
>> S(randi([1,2],10,2))
ans =
TT
HH
HT
TT
HT
TT
TH
HT
HH
HT
```

Coin Tossing: Relative Frequency

$$\frac{N(A,n)}{n}$$

If a fair coin is flipped a large number of times, the **proportion** of heads will tend to get closer to 1/2 as the number of tosses increases.



$n = 1, 2, \ldots, 10$

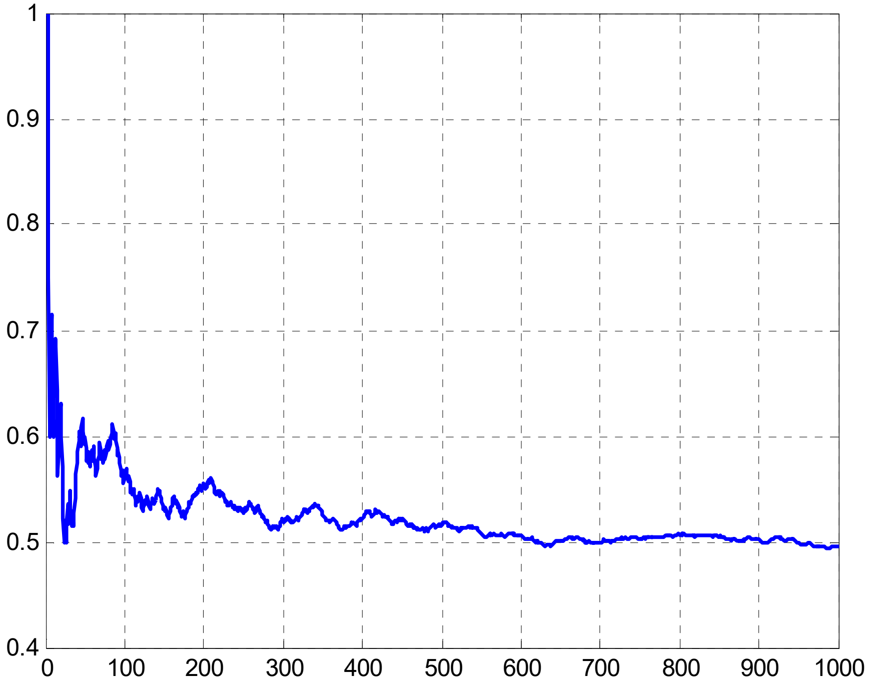
$n = 1, 2, \ldots, 100$

$n = 1, 2, \ldots, 1000$

$n = 1, 2, \ldots, 10^6$

Coin Tossing: Relative Frequency

```
close all; clear all;
N = 1e3; % Number of trials (number of times that the coin is tossed)
s = randi([0,1],1,N);   % Generate a sequence of N Coin Tosses.
                        % The results are saved in a row vector s.
NH = cumsum(s); % Count the number of heads
plot(NH./(1:N),'LineWidth',1.5); grid on % Plot the relative frequencies
```

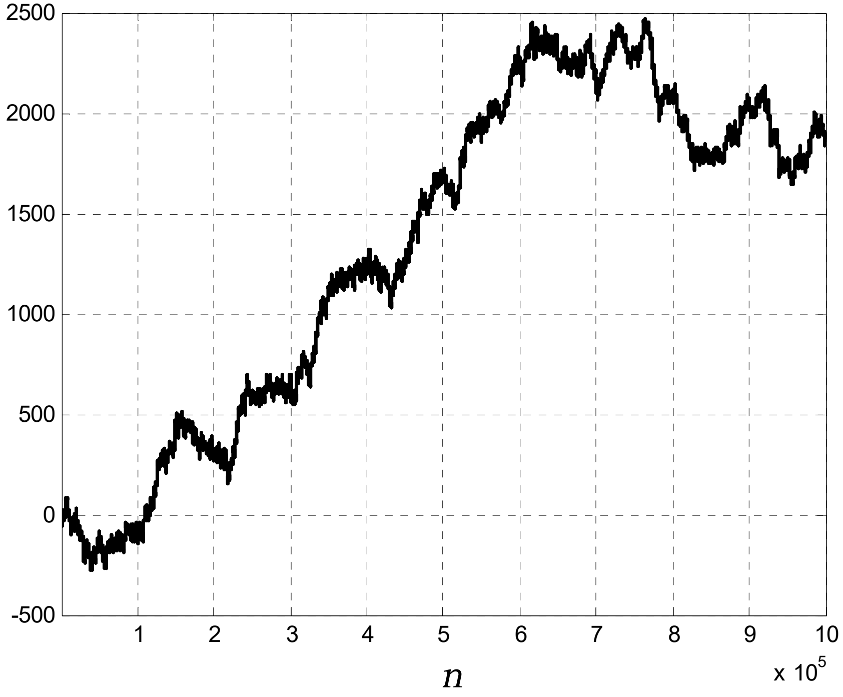


Coin Tossing: Relative Freq. vs. #H-#T

This statement does not say that the difference between #H and #T will be close to 0.

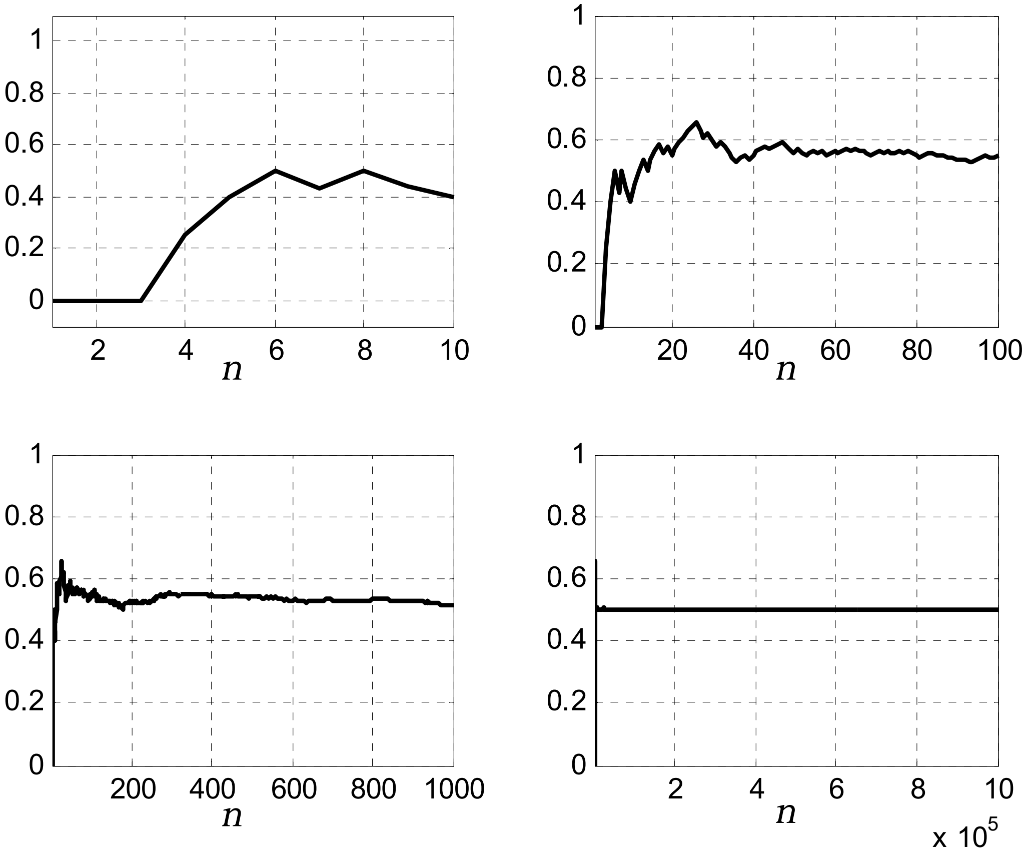
If a fair coin is flipped a large number of times, the **proportion** of heads will tend to get closer to 1/2 as the number of tosses increases.

The **difference** between #H and #T will **not** converge to 0.

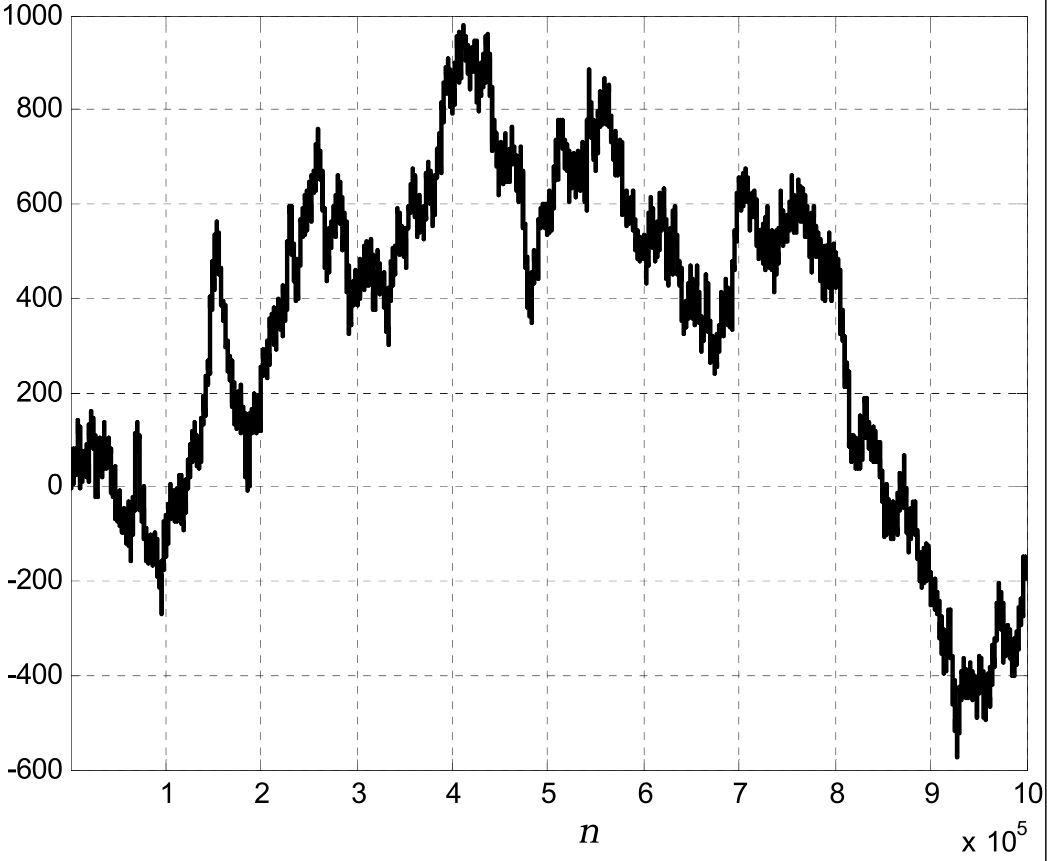


Another Experiment

Relative Freq.

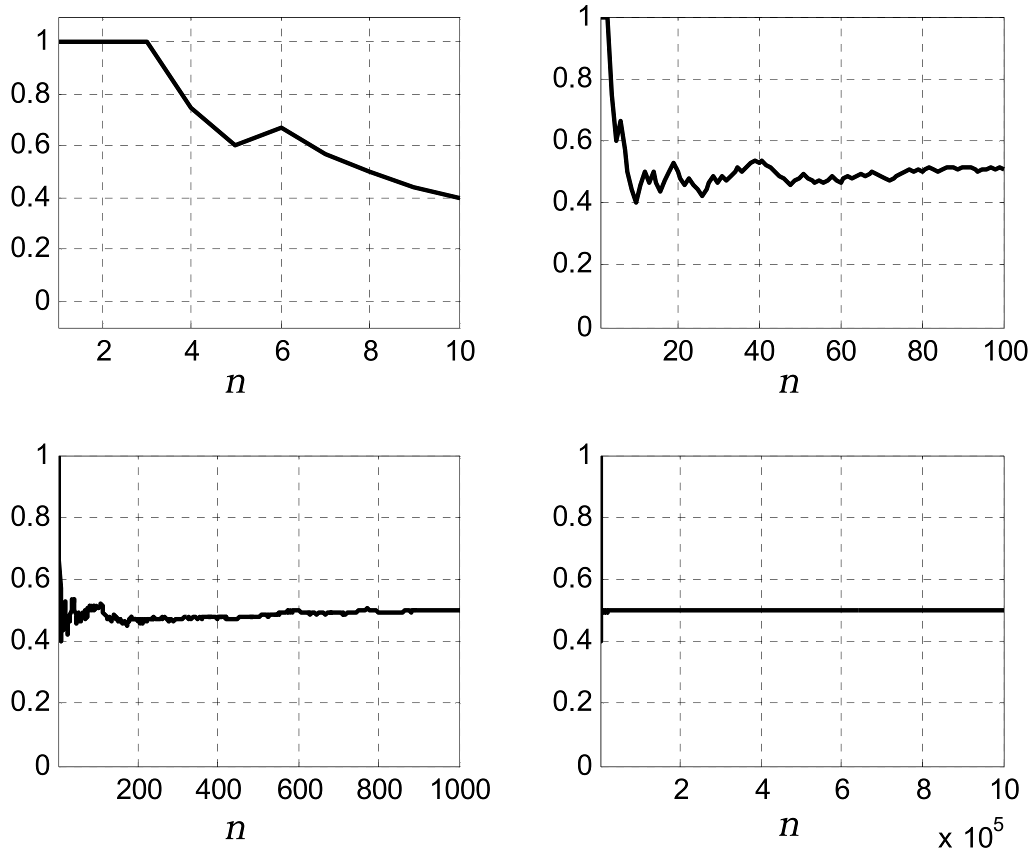


#H-#T

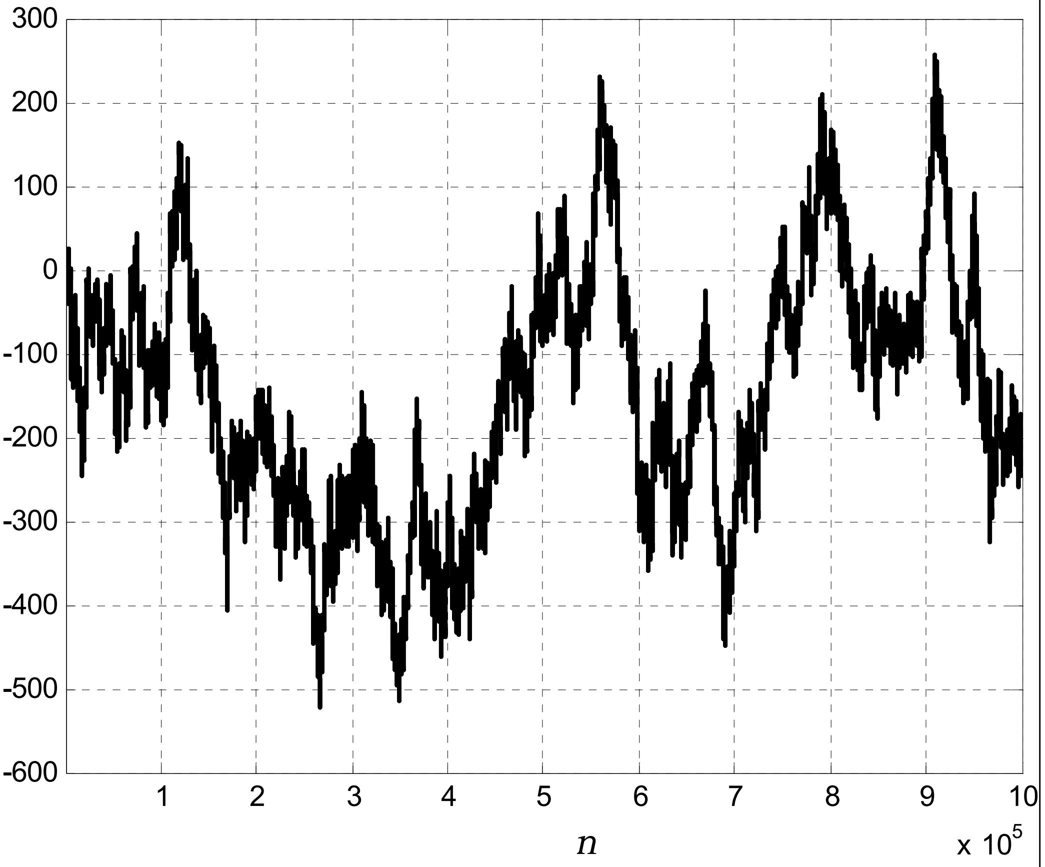


Another Experiment

Relative Freq.



#H-#T


300
200
100
0
-100
-200
-300
-400
-500
-600
1
2
3
4
5
6
7
8
9
10
n
x 10^5