

Searching and choosing in the *open* Web

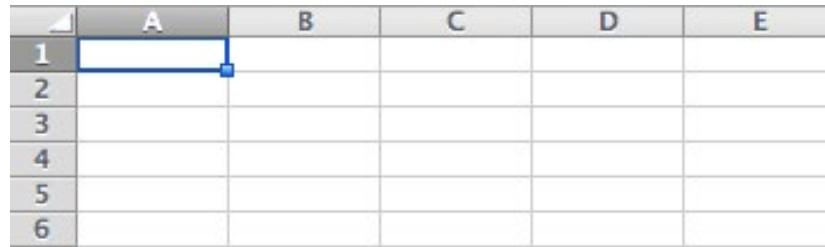
it::unimi::sps::webcomm



Data presentation: Spreadsheet

- ◆ A spreadsheet is a collection of data organized as row of cells:

The cell "A1"



	A	B	C	D	E
1					
2					
3					
4					
5					
6					

- ◆ Each cell can contains a value or a "way" to determines its value, a *function*.
- ◆ Functions create *relations* between cells.
- ◆ Collecting data create *questions* and the problem to find *answers*

Functions with complete knowledge

- ◆ The function `Max()` returns the max value in a set of given values.
- ◆ The input set on a spreadsheet is well defined and clear; we can provide the exact (optimal) solution for the *problem* *Max*

Functions with incomplete Knowledge

- ◆ Sometime on the real world it is not possible to collect the whole data set:
 - ◆ Data set too big, ex: *the average age of the world population.*
 - ◆ Data set extension unknown because hidden into a too big population: *The number of games owned by Italian owners of a Commodore 64 console.*
 - ◆ Lack of time for task execution: *Find the best candidate by deep interview for a job*
- ◆ These are problems with *incomplete Knowledge*

Data analytics is about taking decision.

- Financial market:
 - When to buy a stock?
 - When to sell a stock?
- Real life:
 - When to buy a house?
 - When is the best time for BREXIT?
- Our life:
 - Which data scientist to hire?
 - When to return to work while in vacations?
 - When to get married?

Data analytics

- Analyse data to take decisions
 1. We have data, we analyse the data, we need take decisions on the analysed data
 2. We have data, but... more data is generated every second, minutem hour etc.
 3. We need to take decision on existing data, and data that are about to come...
- When to stop and analyse to take decisions?
- Optimal stopping strategy ([link](#))

The *secretary* problem

- ◆ An administrator wants to hire the best secretary out of n rankable applicants for a position.
- ◆ The applicants are interviewed one by one in random order.
- ◆ During the interview, the administrator can rank the applicant among all applicants interviewed so far, but is unaware of the quality of yet unseen applicants.
- ◆ **A decision about each particular applicant is to be made immediately after the interview. Once rejected, an applicant cannot be recalled.**

What is the best stopping strategy?

The *secretary* problem (contd)

- ◆ Why the secretary problem is meaningful abstraction for web communications:
- ◆ data is flowing, cannot be easily saved, there's non finite domain to refer to.

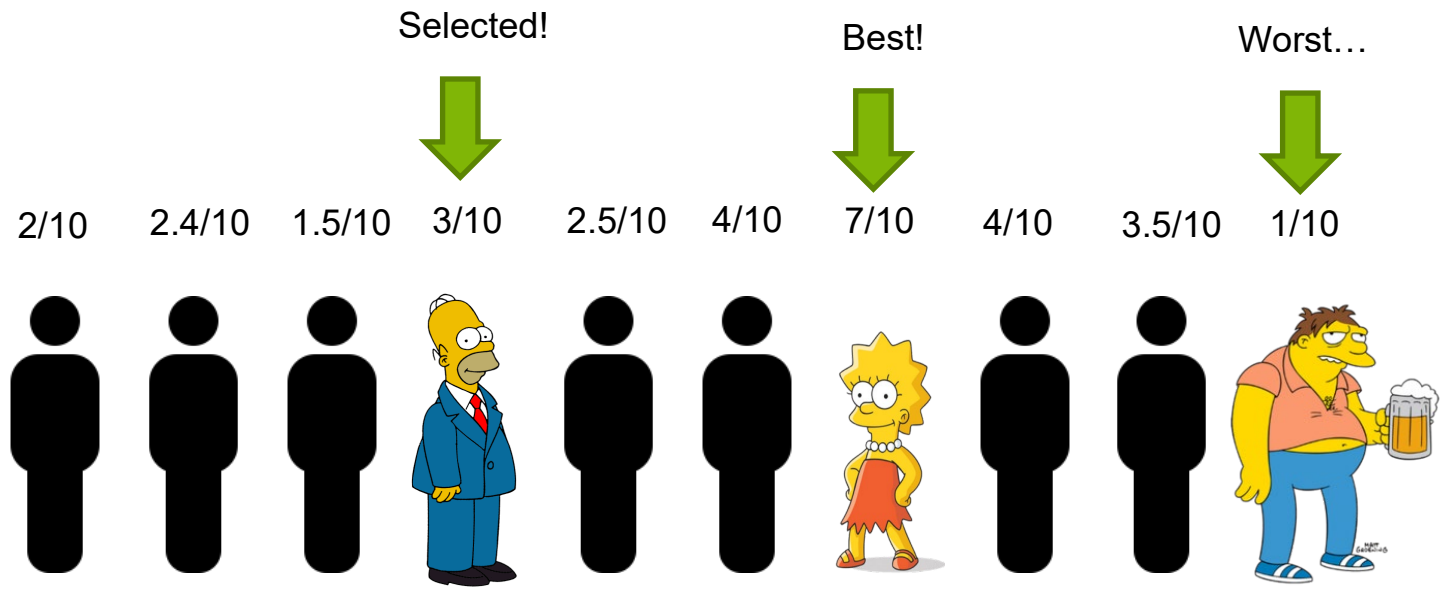
SP and Psychology

- ◆ [...] people tend to stop searching too soon.
- ◆ This may be explained, at least in part, by the cost of evaluating candidates.
- ◆ In real world settings, this might suggest that **people do not search enough** whenever they are faced with problems where the decision alternatives are encountered sequentially

Cfr. https://en.wikipedia.org/wiki/Secretary_problem#Experimental_studies

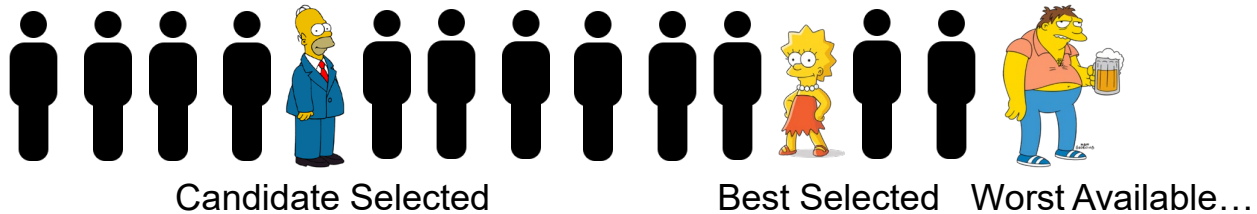
The *secretary* problem (contd)

- ◆ Why the secretary problem is meaningful abstraction for web communications:
- ◆ data is flowing, cannot be easily saved, there's non finite domain to refer to.



The secretary problem

- n applications for a position.
- A function that ranks the applicant's quality (if seen altogether)
- Solution:
 - A choice of applicant that maximizes rank.
- Constraints:
 - Applicants are interviewed one by one, in random order.
 - A decision about each particular applicant is to be made immediately after the interview.



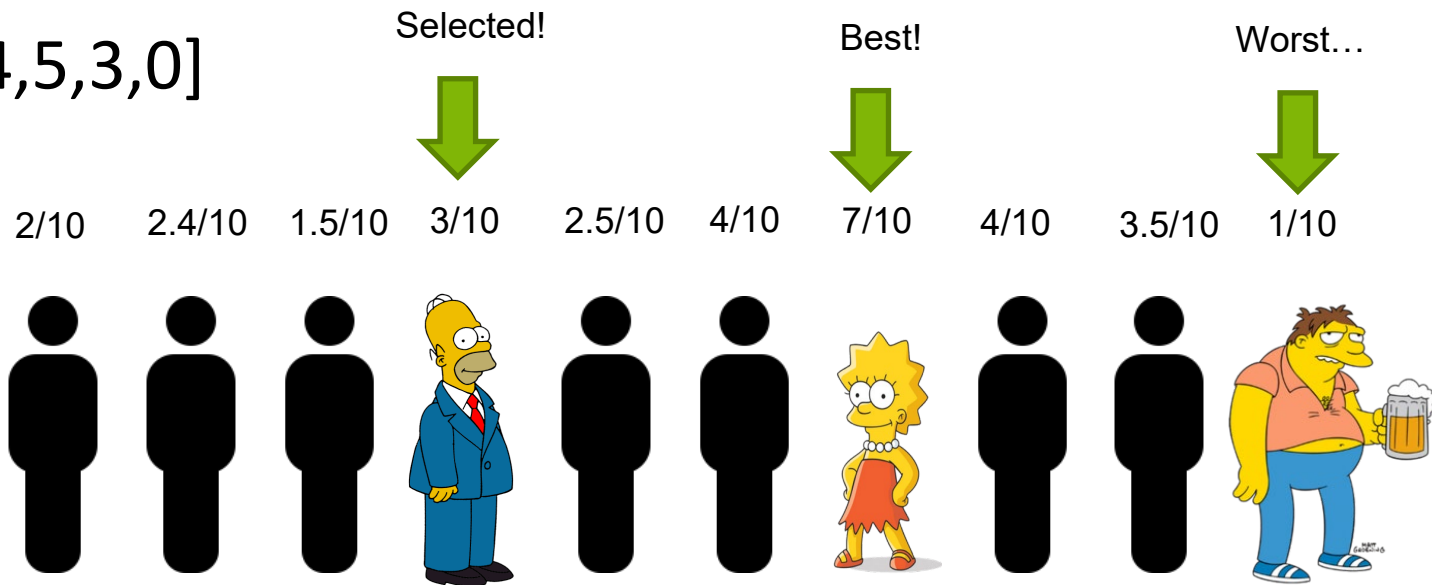
Definition

- Although we can rank the applicant among those interviewed so far, we are unaware of the quality of yet-unseen applicants.
- No strategy, not even waiting until the last applicant, seems to work in all cases.
- We need a method to select a good applicant, most of the times.

$rank(el) =$
of elements in the collection that are $\leq el$

Example

- Cumulative grade point average is an average over [0..4.0]
- Applicants = [2.6, 3.0, 3.4, 3.8, 3.0, 2.5]
- For hiring, only the ranking counts!
- Apps = [1,2,4,5,3,0]



Discussion

- Apps = [1,2,4,5,3,0]
- Hire the first applicant who's:
 - Better than the previous one?
 - Better than the previous two?
 - Above a previously defined threshold? 3.9?

Solution I

- Decide on the basis of what we see in the interviews.
- Use the first r interviews as calibration of an experimental threshold.
- Carry on and hire the first who's above the threshold.
 - $r = 3$; $A = [2.6, 3.0, 3.4]$; $[3.8, 3.0, 2.5]$
- This could go wrong in a number of ways...
 - $A' = [2.6, 2.5, 3.0]$; $[3.4, 3.8, 3.0]$
 - $A'' = [3.4, 3.8, 3.0]$; $[2.6, 2.5, 3.0]$
- We need probabilities...

Probabilities I

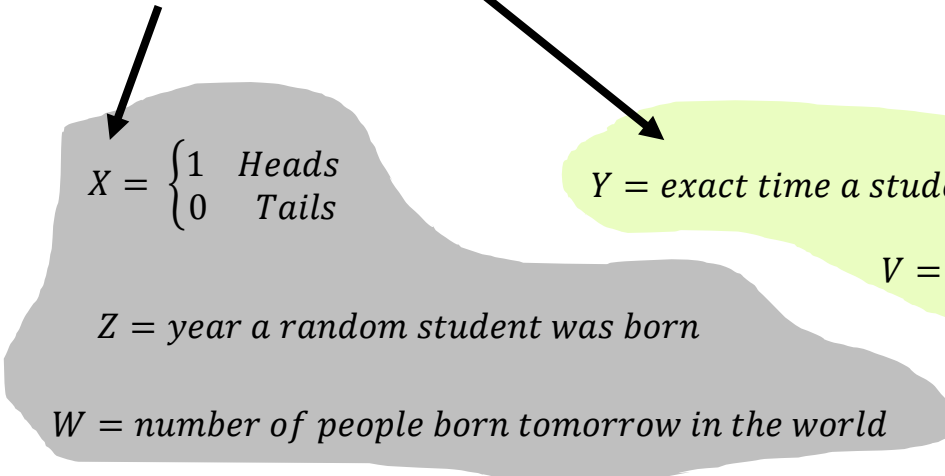
- A probability space consists of:
 - A sample space Ω of all possible outcomes.
 - A set of events F , where each event is a set of containing zero or more outcomes.
 - The assignment of probabilities to the events:
 - Function $\text{Pr}[\cdot]$ from events to probabilities.

Probabilities II

- A probability function:
 - The probability of an event is a non-negative real number:
 - $\Pr[e] \geq 0$
 - The probability that at least one of the events in the sample space will occur is 1.
 - The probability of a set of mutually-exclusive events is the sum of the single probability space.
 - A random variable X is a measurable function $X : \Omega \rightarrow S$ from the sample space Ω to another measurable space S called the [state space](#).

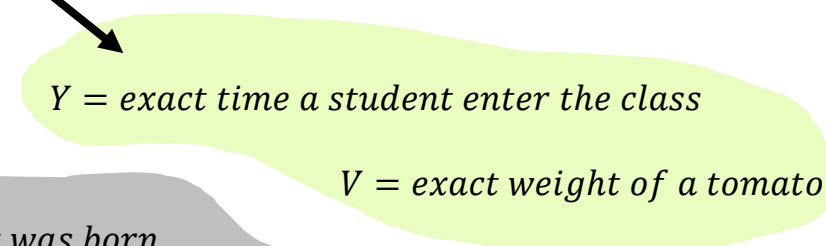
Example!

- Random variables:
 - Discrete: Distinct/ separate values
 - Continuous: Any value that belongs to an interval


$$X = \begin{cases} 1 & \text{Heads} \\ 0 & \text{Tails} \end{cases}$$

$Z = \text{year a random student was born}$

$W = \text{number of people born tomorrow in the world}$



$Y = \text{exact time a student enter the class}$

$V = \text{exact weight of a tomato}$

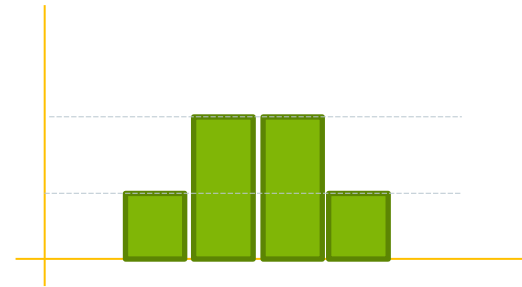
A pragmatic view

- When the range of X is finite [...], the random variable is called a discrete random variable, if X is infinite then it is called continuous random variable.
- Its distribution can be described by a probability mass function, which assigns a probability to each value in the range of X .
- * Probabilities are outside computers, seen as deterministic machines.
- For us probabilities will be a compact way to count:

$$\Pr[X = out.] = \frac{\textit{pos. cases}}{\textit{all cases}}$$

Flip a coin!

- I flip 3 times a coin (fair)!
- $X = \#$ of heads after the 3 flips!



The secretary problem II

- When the instance is presented as a permutation of ranks, we can compute the probability that our threshold strategy will be successful:

$$\begin{aligned} P(r) &= \sum_{i=1}^n P(\text{applicant } i \text{ is selected} \cap \text{applicant } i \text{ is the best}) \\ &= \sum_{i=1}^n P(\text{applicant } i \text{ is selected} | \text{applicant } i \text{ is the best}) \times P(\text{applicant } i \text{ is the best}) \\ &= \left[\sum_{i=1}^{r-1} 0 + \sum_{i=r}^n P\left(\begin{array}{l} \text{the best of the first } i-1 \text{ applicants} \\ \text{is in the first } r-1 \text{ applicants} \end{array} \middle| \text{applicant } i \text{ is the best} \right) \right] \times \frac{1}{n} \\ &= \sum_{i=r}^n \frac{r-1}{i-1} \times \frac{1}{n} = \frac{r-1}{n} \sum_{i=r}^n \frac{1}{i-1}. \end{aligned}$$

The secretary problem III

Candidates n	Optimal Sample size k	Ratio k/n	Probability of Success
3	1	0.333333333	0.366
4	1	0.25	0.347
5	2	0.4	0.367
6	2	0.333333333	0.366
7	3	0.428571429	0.363
8	3	0.375	0.368
9	3	0.333333333	0.366
10	4	0.4	0.367
11	4	0.363636364	0.368
12	4	0.333333333	0.366
13	5	0.384615385	0.368
14	5	0.357142857	0.368
15	6	0.4	0.367
16	6	0.375	0.368
17	6	0.352941176	0.368

Pr[X=best] converges toward $1/e \approx 0.368$

Solving the secretary problem!

```
#Solving the probabilities for the Secretary problem
```

```
import math
```

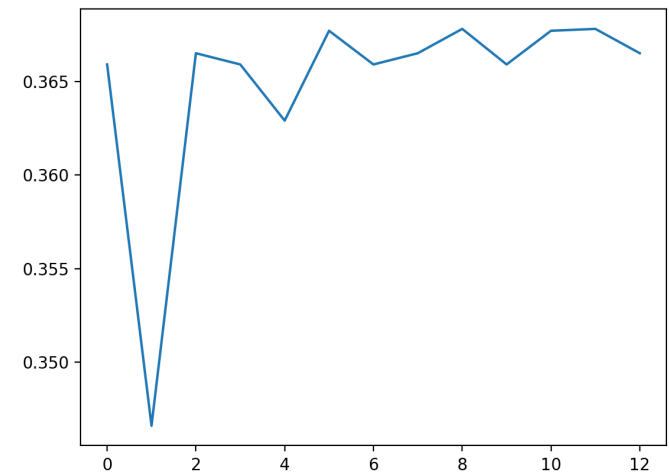
```
cand_list = range(3,16)
```

```
optimal_sample_size_k = [int(round(i/2.71,0)) for i in cand_list]
```

```
ratio_k_n = [round(optimal_sample_size_k[i]/cand_list[i],4) for i in  
range(len(optimal_sample_size_k))]
```

```
approx_prob_success_p = [round((i*math.log(i)*-1),4) for i in ratio_k_n]
```

```
print(approx_prob_success_p)
```



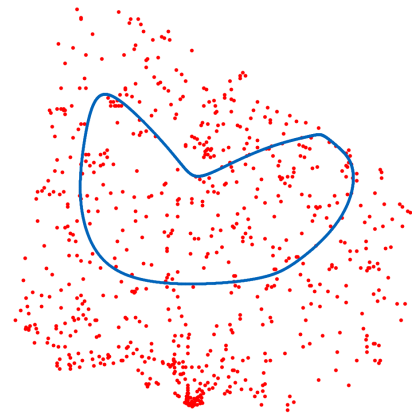
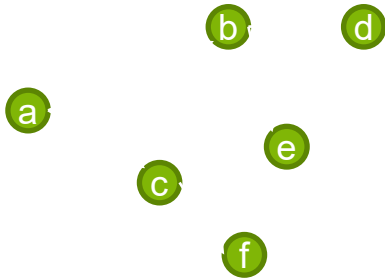
On-line Algorithm

- The golden rule algorithm is correct for secretary, with a certain probability.
- The sequence of operations is determined by the instance, however cost is always $\leq O(n)$.
- We don't know the instance beforehand hence we can't look ahead.
- However, the size n of the instance is always available.
- What if a flow of external data is available?
- This is the task of Online algorithms and dataflow/datastream programming.

Online algorithms Examples

"Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?"

- Travelling salesman problem (Offline...)



The garden of Probability and Stats

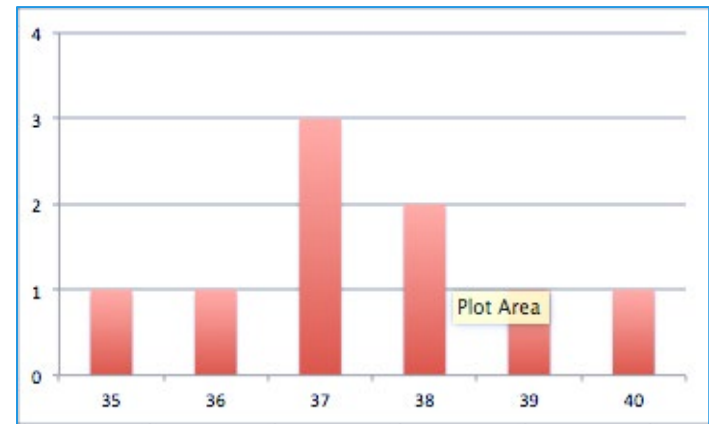


The source of Knowledge

- ◆ A sensor/probe returns one of a finite set of possible values
 - ◆ Thermometer: A number into $34.5 \div 43.5$ with step of 0.1.
 - ◆ Dice: 1,2,3,4,5,6
 - ◆ Political ballot: one of two candidates
- ◆ We can repeat measurement various times, collecting a set of *observations*, a **dataset**.
- ◆ Analyzing observations, we can try to infer some knowledge of the world the data came from.

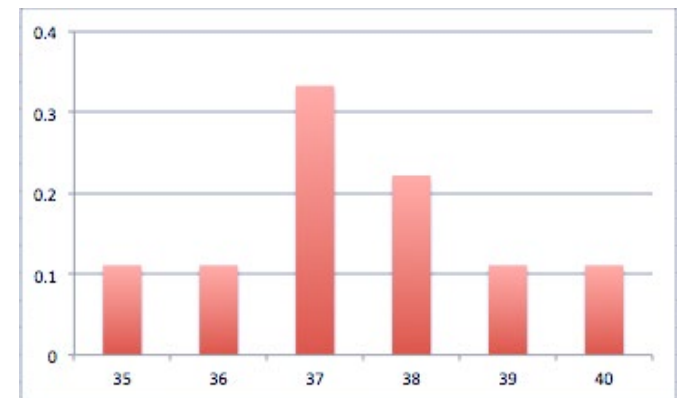
Frequency and frequency histogram

- Frequency: How many times a particular value happened in my observations?
- Frequency histogram: How my frequency are spread among my observations?
 - given this observations: {37,35,36,37,37,38,40,38,39}
 - $Fr(35)=1$, $Fr(36)=1$, $Fr(37)=3$
 $Fr(38)=2$, $Fr(39)=1$, $Fr(40)=1$
 - $FrHist(35 \div 40) = \{1, 1, 3, 2, 1, 1\}$



... toward Knowledge

- Frequency normalization: reformat histogram in order to *hide* the dataset size, and *try to generalize*:
 - given this observations: {37,35,36,37,37,38,40,38,39}
 - #observation = 9
 - NormlizedFr(35)=1/9, NormFr(36)=1/9,
Norm Fr(37)=3/9, NormFr(38)=2/9,
NormFr(39)=1/9, NormFr(40)=1/9
 - NormalizedFrHist(35 ÷ 40)
= {1/9, 1/9, 3/9, 2/9, 1/9, 1/9}



The important of having multiple observations

- ◆ Many observations you made, more your observations are near to the reality (*the Law of large numbers*)
- ◆ How many observations are needed? The importance of selecting a good population in which make observations.
- ◆ **Bias** can deviate data:
 - ◆ I tend to use thermometer when I'm sick so my average temperature from that observations dont represent my *real* avarege temperature.
 - ◆ Usually young people dont reply to the home phone; interviews with this chanel tend to reach more adults.
 - ◆ What about “**algorithmic bias?**”

Mean vs. Median

- ◆ Mean: the simplest average: sum of all values divided by number of observations
 - + easy to calculate
 - + can be adapted, with math transformations
 - for low # of observations, it tends to be *biased by outliers*
- ◆ Median: the observation in the middle, i.e. ordering observation by value, it is the observation value who have the same number of observation before and after itself
 - + less sensible to outliers respect Average
 - requires an ordering step (expensive to compute)

From small to large: probability

- ◆ Informally: ratio of the # of *good* observable values over # of *possible* observable values (*sample space*).
 - ◆ Dice:
 - ◆ possible observable values: {1,2,3,4,5,6}
 - ◆ Probability of "5": 1/6
 - ◆ Coin:
 - ◆ possible observable values: {"head","tail"}
 - ◆ Probability of "head": $\frac{1}{2}$
- ◆ formally, $\text{Pr}: S \rightarrow [0..1]$ (0=impossible, 1=certain) s.t. its integral (sum over S) is 1.

Exercise

The Probability of seeing a 'six' when throwing two dice:

♦ possible observable values:

<1,1>, <1,2>, <1,3>, <1,4>, <1,5>, <1,6>

<2,1>, <2,2>, <2,3>, <2,4>, <2,5>, <2,6>

<3,1>, <3,2>, <3,3>, <3,4>, <3,5>, <3,6>

<4,1>, <4,2>, <4,3>, <4,4>, <4,5>, <4,6>

<5,1>, <5,2>, <5,3>, <5,4>, <5,5>, <5,6>

<6,1>, <6,2>, <6,3>, <6,4>, <6,5>, <6,6>

♦ good observable values:

<6,1>, <6,2>, <6,3>, <6,4>, <6,5>, <6,6>, <1,6>, <2,6>, <3,6>, <4,6>, <5,6>

♦ $\Pr(\text{"seeing a 6"}) = 11/36 \cong 0.3$

Epilogue: the $1/e$ -strategy

- ◆ The best known strategy for the secretary problem is “37% rule:”
- ◆ Let N be the number of applicants
- ◆ Interview the first N/e applicants and fix the threshold score t ($e=2.718\dots$)
- ◆ Interview the remaining candidates; hire the first whose score $> t$.
- ◆ $\Pr[X=\max] = 1/e = 0.3678\dots$

- ◆ What could possibly go wrong???

Final considerations

- 💧 The Web is open-domain: hard to fix the sample space (denominator)
- 💧 A phenomenon ('seeing a 6') might have more than one explanation: hard to 'go back' to the original happening
- 💧 We try to *maximise the impact of communication* by either
 - Increasing frequencies (numerator)
 - Re-shaping the user base (denominator)
- 💧 Better interfaces
- 💧 Statistical tests that allow to estimate impact: [A/B testing](#)