1

Introduction

During the years that I've been preparing Volume 4B, I've often run across basic techniques of probability theory that I would have put into Section 1.2 of Volume 1 had I been clairvoyant enough to anticipate them in the 1960s ...

— Donald Knuth preface to Volume 4B of The Art of Computer Progamming.

By the end of this chapter you should:

- Know the aims of this course.

Please consult the Electronic Course Profile for details concerning assessment.

A **random experiment** is a process whose outcome cannot be determined in advance.

Examples of random experiments:

- The number of collisions in a hash table.
- Time for a bug in a program to be found/reported.
- The number of comparisons required by the quicksort algorithm to sort a list of items.

To handle *randomness*, we need **models** for random experiments.

Dogfight

Suppose that Alice, Bob, and Carol are fighting in an air battle.

- In each round, each survivor fires one shot. Alice fires first, then Carol fires, and then Bob fires.
- Anyone hit drops out of the battle immediately.

- On any shot aimed at an opponent:
  - Alice hits with probability $\frac{2}{5}$. (Meaning on average 2 out of every 5 shots Alice fires will hit their target.)
  - Bob hits with probability $\frac{1}{2}$.
  - Carol never misses.

Question: Where should Alice fire in the first round?

Answer:

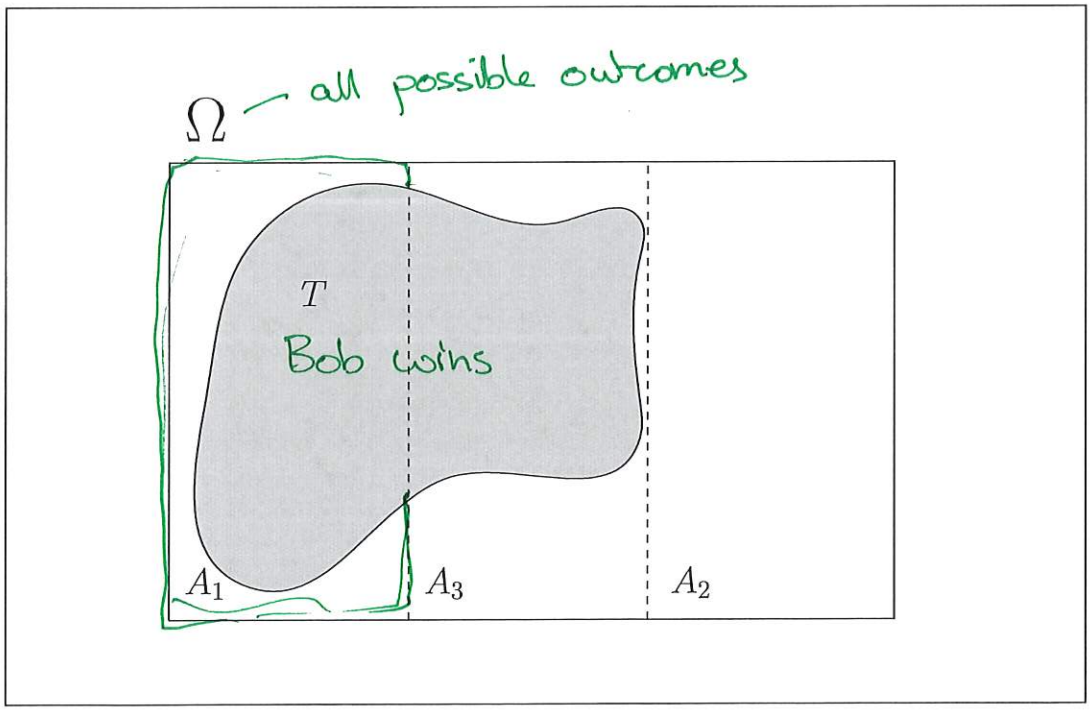
Note that if Alice fires at Bob and hits him, then Alice will immediately be shot by Carol and lose. So in the first round Carol is the only target to consider.

First, suppose that Alice was to fire at Carol and hits her (which occurs with probability $\frac{2}{5}$). Alice would then be Bob's target, and the battle would continue with shots alternating between Alice and Bob until a hit is scored. Define the **event**

- $T \equiv$ Bob wins the one-on-one battle with Alice.

We also need to consider the **events**:

- $A_1 \equiv$ Bob hits with his first shot
- $A_2 \equiv$ Bob misses with 1st shot AND Alice hits with (second) return shot.
- $A_3 \equiv$ Both Bob and Alice miss (Bob's 1st and Alice's return)



If A_3 occurs, then the next round begins under the same conditions; hence,

$$\mathbb{P}(T \mid A_3) = \mathbb{P}(T)$$

here the function $\mathbb{P}$ takes an event (e.g. $T \mid A_3$) and gives its probability. When we write $T \mid A_3$ we are supposing that the event A_3 has already happened, and we are now interested in the event T *conditional* on the fact that A_3 has occurred. Also,

$$\mathbb{P}(T \mid A_1) = 1 \quad \text{and} \quad \mathbb{P}(T \mid A_2) = 0$$

because if Bob hits with his first shot then Alice is out of the game and Bob wins, but if Bob misses with his first shot and then Alice hits with her first shot then Bob is out of the game.

So

$$\begin{aligned}\mathbb{P}(T) &= \mathbb{P}(T \mid A_1)\mathbb{P}(A_1) + \mathbb{P}(T \mid A_2)\mathbb{P}(A_2) + \mathbb{P}(T \mid A_3)\mathbb{P}(A_3) \\ &= 1 \times \tfrac{1}{2} + 0 \times \mathbb{P}(A_2) + \mathbb{P}(T) \times \tfrac{1}{2} \times \tfrac{3}{5}\end{aligned}$$

$$\mathbb{P}(T) = \tfrac{1}{2} + \tfrac{3}{10}\,\mathbb{P}(T)$$

which implies

$$\tfrac{7}{10}\,\mathbb{P}(T) = \tfrac{1}{2}$$

$$\mathbb{P}(T) = 5/7 .$$

Second, suppose that Alice was to fire at Carol and misses her. Then Carol will certainly fire at Bob because $\frac{2}{5} < \frac{1}{2}$ and Carol is a rational person. Since Carol is such a good shot, she will 'almost surely' hit Bob. So in this case Alice only wins if Alice's second shot hits Carol in the second round, which happens with probability $\frac{2}{5}$.

Hence, missing Carol (on purpose) gives Alice a better chance since

$$\underbrace{1 - \mathbb{P}(T)}_{\text{Alice wins against Bob}} = 1 - \frac{5}{7} = \frac{2}{7} < \frac{2}{5}.$$

In summary, since Bob is a sufficiently better shot than Alice, Alice should let Carol take out Bob and hope for a good outcome on her single shot against Carol.

This example is adapted from [5, p. 68].

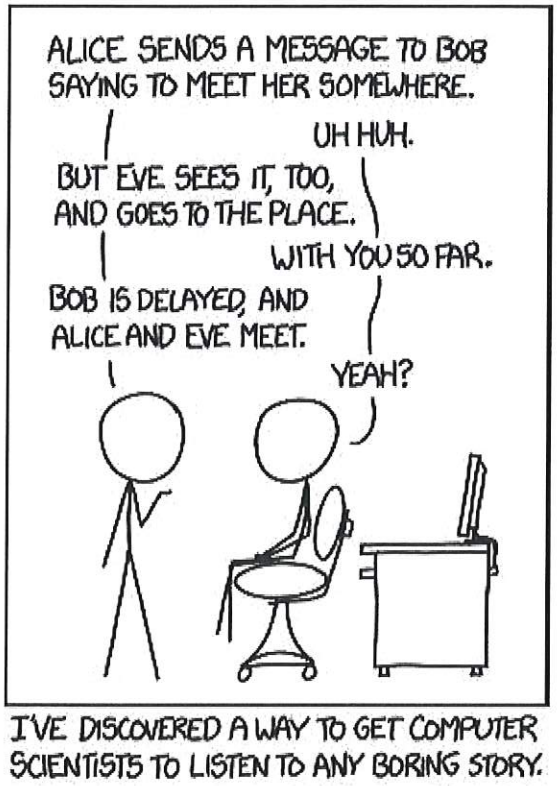


Figure 1.1: From `www.xkcd.com/1323/`

$\log(x^y) = y\log(x)$

Software effort estimation

$\log(xy) = \log(x) + \log(y)$

Cost-estimation is a difficult problem in software engineering. A basic model that relates the person-hours a project requires and a measure of the complexity of the project is

$$\text{Effort} = a \times (\text{Project complexity})^b,$$

$\log(\text{Effort}) = \log(a) + b \log(\text{Project complexity})$

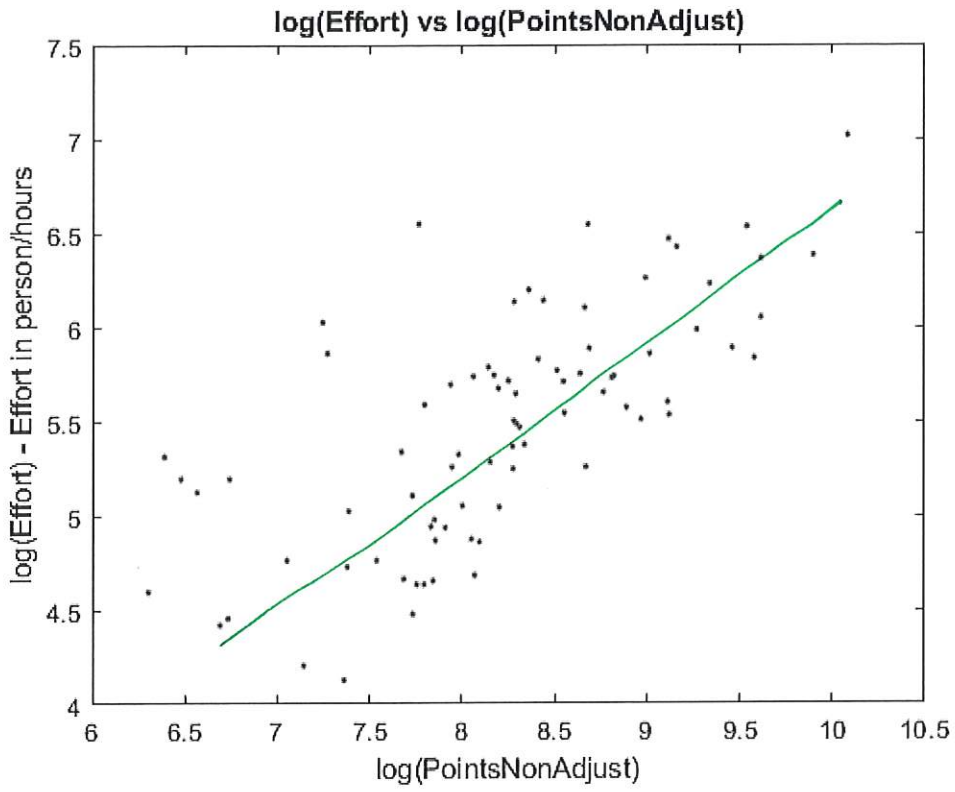
where a and b are parameters that can be estimated from historical data. For the purpose of this demonstration we take the non-adjusted function point technique as the measure of complexity. Jean-Marc Desharnais[1] surveyed 10 organisations on 81 management information systems development projects completed between 1983 and 1988. The data is presented in the scatterplot below on the log-log scale.

Question: How can we estimate a and b from this data?

MATLAB

```
desh = readtable('desharnais.xlsx');
desh.logEffort = log(desh.Effort);
desh.logPNA = log(desh.PointsNonAdjust);
plot(desh.logEffort,desh.logPNA,'.')
title('log(Effort) vs log(PointsNonAdjust)')
xlabel('log(PointsNonAdjust)')
ylabel('log(Effort) - Effort in person/hours')
```

[1]The file `deshardnais.xlsx` is on Blackboard. The original data from Desharnais's Masters thesis is available from `http://promise.site.uottawa.ca/SERepository/datasets-page.html`.



scatter plot

Notice that log(*Effort*) is roughtly a linear function of the log(*Project Complexity*).

Question: Suppose that a we have a project whose log(*Project Complexity*) is 6. How many person hours do we expect the project to take? How certain would we be of this estimate.

The following code can be used to perform an *ordinary least squares* analysis of this dataset in MATLAB. First place the file 'desharnais.xlsx' in your current working directory, and then execute the following code.

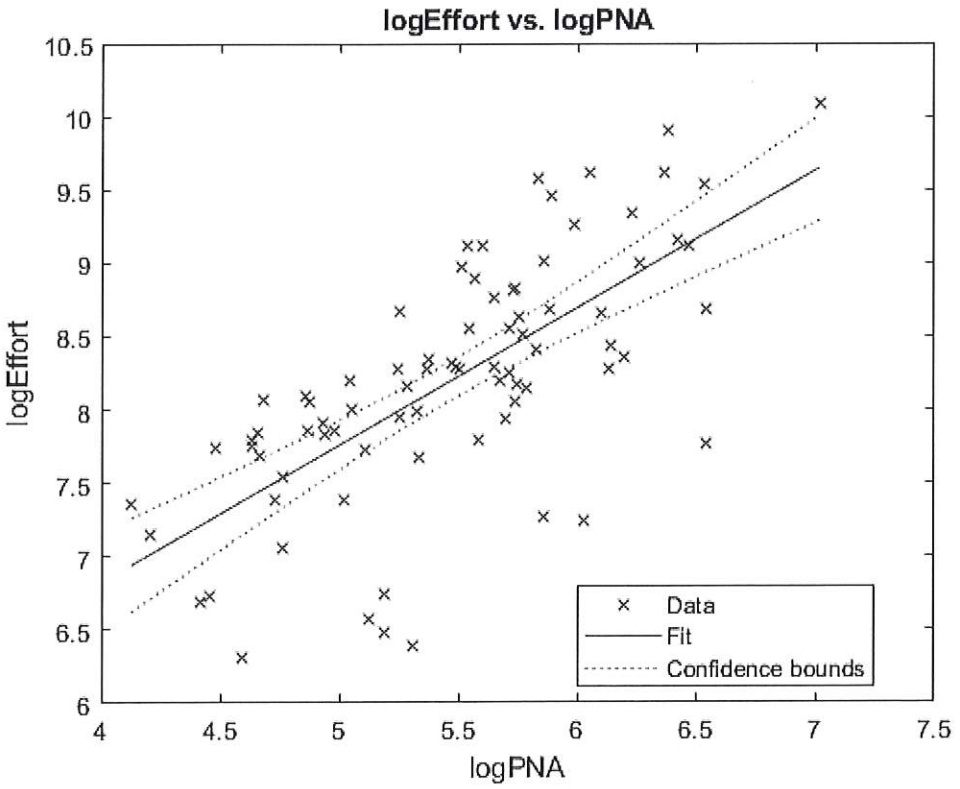
```
1 lm = fitlm(desh, 'logEffort~logPNA')
2 plot(lm)
```

You'll see the following output and figure.

```
Linear regression model:
    logEffort ~ 1 + logPNA

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                                                    log(a) =
    (Intercept)      3.0775     0.59842    5.1427    1.9218e-06
    logPNA           0.93612    0.10863    8.6178     5.428e-13
                     ≈ b

Number of observations: 81, Error degrees of freedom: 79
Root Mean Squared Error: 0.599
R-squared: 0.485,  Adjusted R-Squared 0.478
F-statistic vs. constant model: 74.3, p-value = 5.43e-13
```



Having obtained estimates for a and b, our task is not finished. We need to ask ourselves:

Question: Is the data consistent with the original model?

If the data is not consistent with the model, then our estimates and inferences will be worthless.

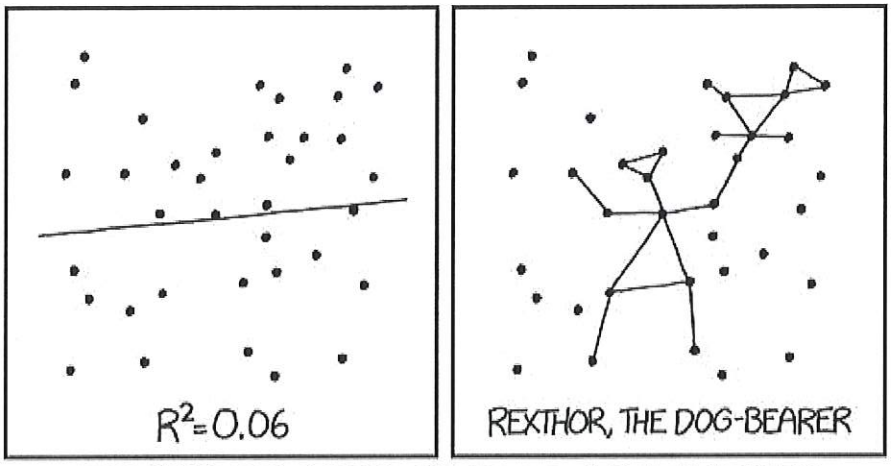


Figure 1.2: From `www.xkcd.com/1725/`