

Rethinking Interaction with Literate Computing

Clemens Nylandsted Klokmose & Roman Rädle
Aarhus University



A *probability distribution* defines how likely different states of a random variable are. The probability distribution of a random variable X is written:

$$P(X)$$

and is shorthand for the expression

$$\Pr(X = x),$$

i.e. that the variable X takes on the specific value x .

Random variables can be continuous (e.g. the height of a person) or discrete (the value showing on the face of a dice). The distribution of a discrete variable is described with a *probability mass function* (PMF) which gives each outcome a specific value. A continuous variable has a *probability density function* (PDF) which specifies the spread of the probability as a continuous function.

A probability distribution must assign probabilities in the range 0 (impossible) to 1 (definite) and the PMF or PDF **must** sum/integrate to exactly 1 as the random variable under consideration must take on *some* value.

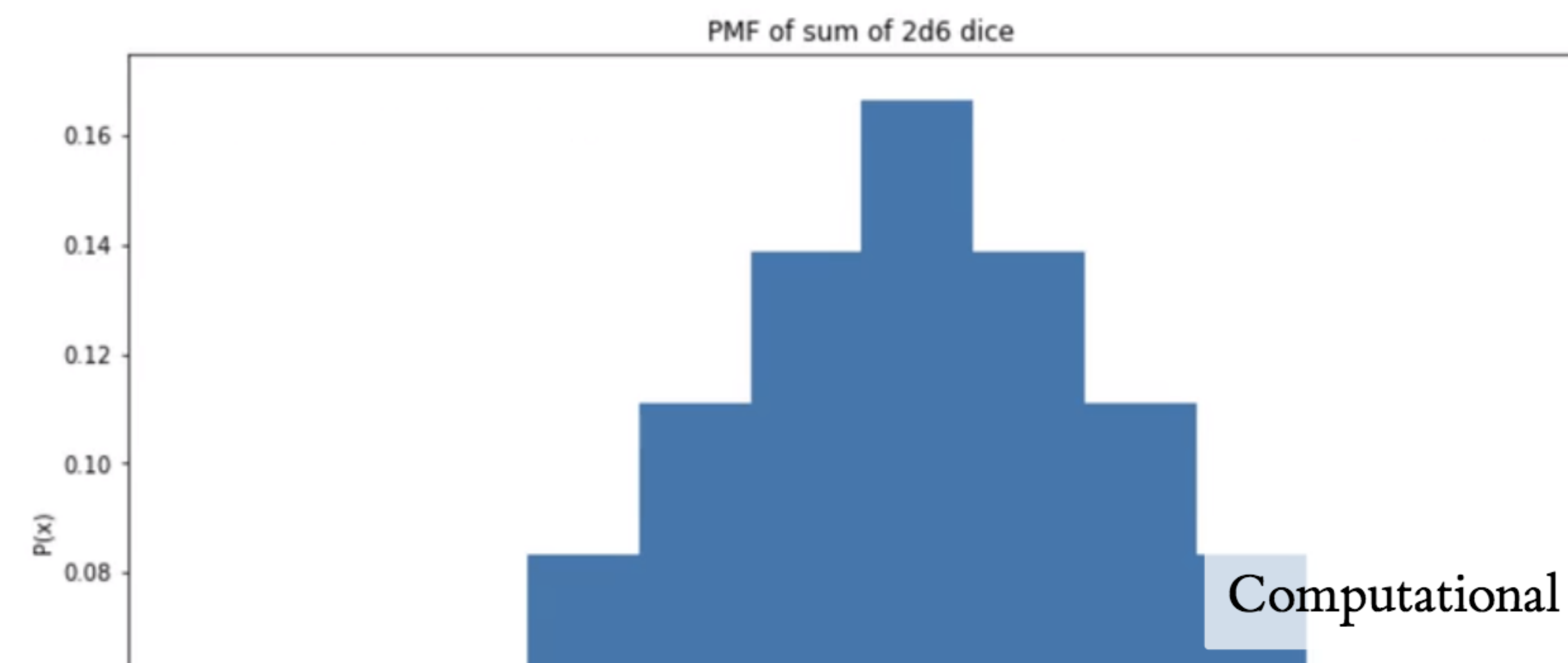
PMF example: sum of dice rolls

A very simple discrete PMF is the expected value of the sum of two six-faced die. In this case, $P(X) = P(D_1 + D_2)$ for two uniform discrete variables $D_1, D_2 \in \{1, 2, 3, 4, 5, 6\}$

```
In [8]: 1 # the PMF of the sum of two dice rolls
2 def two_dice():
3     # form the sum of the cross product of these possibilities
4     roll_two = [i+j for i in range(1,7) for j in range(1,7)]
5     # now plot the histogram
6     pmf, edges, patches = plt.hist(roll_two, normed=True, bins=range(1,14))
7     print("Sum of PMF %.2f" % np.sum(pmf)) # sum of probability should be *exactly* 1.0
8     plt.title("PMF of sum of 2d6 dice")
9     plt.xlabel("Sum of rolls x")
10    plt.ylabel("P(x)")
```

```
In [9]: 1 two_dice()
```

Sum of PMF 1.00



literate computing

is the act of weaving “*a narrative directly into a live computation, interleaving text with code and results to construct a complete piece that relies equally on the textual explanations and the computational components.*”

Codestrates combines Literate Computing with Webstrates

Collaborative computation, extension, and development of applications

Prototyping similar to code playgrounds, but with potential to become usable applications with persistent states

Development environment for Webstrates that goes beyond paradigmatic application-document model

Interactive notebooks

```
10 if (unit=="K") { dist = dist * 1.09344 }
11 if (unit=="M") { dist = dist * 0.8684 }
12 return dist
13 }
```

Analysis of Movement Speeds

The average movement speeds are `mean-walking (unset)` m/s for walking, `mean-running (unset)` m/s for running, and `mean-cycling (unset)` m/s for cycling.

`result-plot (unset)`

Statistics

```
1 const calculateMean = (data) => {
2   let sum = 0;
3   data.forEach(v => {
4     sum += v.distance;
5   });
6   return (sum / data.length).toFixed(2);
7 }
```

Plot Code

```
1 Walking Speed Data #geo-data-walking class:
2 {
3   "measure": 0,
4   "timestamp": 1491298018342,
5   "distance": 3.33
6 },
7 {
8   "measure": 1,
9   "timestamp": 1491298018342,
10  "distance": 3.39
11 },
12 {
13  "measure": 2,
14  "timestamp": 1491298020142,
15  "distance": 3.42
16 },
17 {
18  "measure": 3,
19  "timestamp": 1491298021942,
20  "distance": 3.42
21 },
22 {
23  "measure": 4,
```

Teaching programming

Opgave 1

Nedenfor er der en knap, og i kassen ved siden af vil vi gerne have vist hvor mange gange der er trykket på knappen. Hvis der er trykket under tre gange, skal teksten være sort, over tre gange `green`, over seks gange `orange` og over ni gange `red`.

Vi har hjulpet jer lidt. Funktionen `click` bliver kaldt hver gang der klikkes på knappen med teksten 'Klik på mig!'. Funktionen `output` på `helper` objektet skriver output ud til højre for knappen. Første argument er farven og andet argument er et antal der skal skrives ud.

Når du har skrevet noget kode skal du klikke på i menuen over koden for at få koden kørt. Hvis du trykker på i samme menu kan du få konsollen med evt. fejlmeddelelser frem. Du kan også bruge Chromes indbyggede JavaScript konsol.

Eksempleresultat (hold musen over)

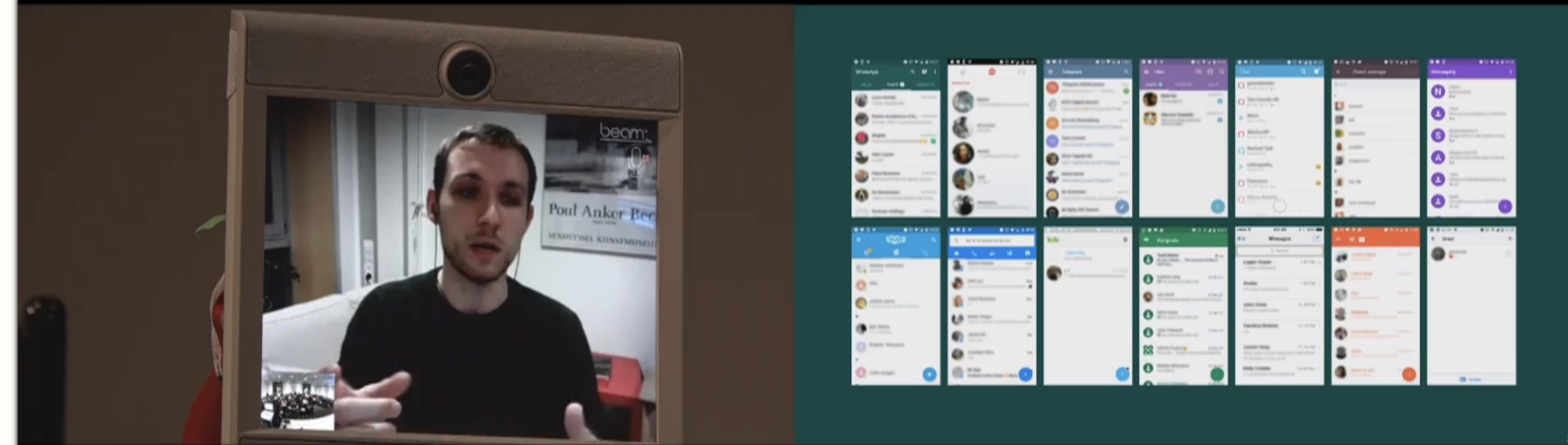
Hint: Overvej if-sætninger.

Klik på mig! Her kommer der (måske?) output.

```
1 var helper = require("#opgave1-helper");
2
3 var counter = 0; // Tæl denne værdi op hver gang der klikkes
4 var color = "black"; // Brug denne til at gemme farvewærdien
5 var click = function() {
6
7   /* ===== SKRIV DIN KODE HER ===== */
8
9   helper.output(color, counter);
10 };
11
12 helper.whenButtonClickedCallThisFunctionPlease(click);
```

Opgave 2

Application development



Physical computing

