MathsNET
A joined up approach to
teaching and learning
mathematics

Understanding random variables

0.1 Level 1

Use the blocks to generate a uniform random variable, X. Plot a point on the graph at $(1, X)$.

0.2 Level 2

Use the blocks to generate 50 uniform random variables. You should plot each of these random variables on the graph. The ith random variable you generate, X_i, should be at the coordinate (i, X_i).

0.3 Level 3

A Bernoulli random variable (or Bernoulli trial) can take one of two values 0 or 1. For this type of random variable the probability that the random variable takes a value of 1 is equal to p and the probability that the random variable takes a value of 0 is equal to $(1-p)$. Use the blocks to generate 50 random variables from a bernoulli distribution with $p = 0.75$. You should plot each of these random variables on the graph. The ith random variable you generate, Y_i, should be at the coordinates (i, Y_i).

0.4 Level 4

We use a binomial random variables to describe the result from an experiment in which N bernoulli trials are performed. The binomial random variable tells you how many of these trials were sucessful. We can generate a binomial random variable by generating N bernoulli random variables and adding them all together. Use the blocks to generate 50 random variables from a binomial distribution with $p = 0.75$ and $N = 5$. You should plot each of these random variables on the graph. The ith random variable you generate, Y_i, should be at the coordinates (i, Y_i).

0.5 Level 5

We use geometric random variables to describe the result from experiments in which bernoulli trials are performed until sucess is acchieved. A geometric random variable measures how many trials were required to achieve a sucess. We can generate a geometric random variable by generating bernoulli random variables until sucess is achieved. The geometric random variable is then equal to the number of trials we performed. Use the blocks to generate 50 random variables from a geometric distribution with $p = 0.25$ You should plot each of these random variables on the graph. The ith random variable you generate, Y_i, should be at the coordinates (i, Y_i).

0.6 Level 6

Consider a random variable that can take one of three possible values 0, 1 or 2. Suppose that the probabilities for each of these outcomes are $P(X = 0) = 0.2$, $P(X = 1) = 0.3$ and $P(X = 2) = 0.5$. Use the block that generates a uniform random variable and an if-elseif block as well as the other blocks to generate 50 random variables from this distribution. You should plot each of these random variables on the graph. The ith random variable you generate, Y_i, should be at the coordinates (i, Y_i).

0.7 Level 7

The probability mass function for the poisson random variable is $f_X(x) = \frac{\lambda^x}{x!}e^{-\lambda}$ and x can take any integer value that is greater than or equal to zero. Use the blocks to generate 50 random variables from a poisson distribution with $\lambda = 2$. You should plot each of these random variables on the graph. The ith random variable you generate, Y_i, should be at the coordinates (i, Y_i).

0.8 Level 8

If we generate a set of random variables $\{X_i\}$ it is often useful to calculate the sample mean using $\mu = \frac{1}{N}\sum_{i=1}^{N} X_i$, where N is the number of random variables in our set. This quantity is still random as it is the sum of a set of random variables but as we will see in subsequent exercises it is easier to make predictions about the value of this sum. Use the blocks to calculate 10 μ values. Each of these μ values should be calculated by adding together 5 uniform random variables. You should plot each of these values for μ on the graph. The ith μ value you generate, Y_i, should be at the coordinates (i, μ_i).

0.9 Level 9

Drawing graphs is computationally expensive and this business of adding a new point every time we generate a new random variable is slowing our codes down enormously. To avoid this expense use the blocks to generate 10 normal random variables from a distribution with $\mu = 0$ and $\sigma = 1$. This time, however, instead of plotting them instantly store them in a list and only plot the graph once all the data is generated.