Year 9 Mathematics Investigation 2 – Sampling and Distributions

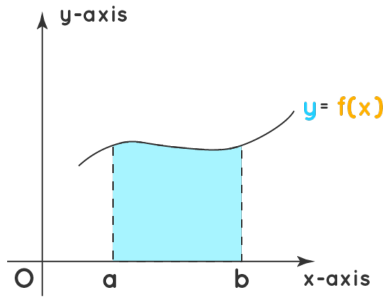
Introduction

Purpose
In this assignment you will use random sampling to explore the technique of function integration. It will also include knowledge of geometry, image processing and probability.

Learning Goals
- Understand the difference between an exact and approximate solution.
- Understand how random sampling can be used to efficiently approximate a problem solution.
- Learn about different probability distributions.

1 Approximating Function Integration Using Random Selection

The integral of a function refers to the area under the graph of a function, such as the area in blue shown below.



Within calculus, determining the integral of a function is referred to as integrating a function. In this assignment you will not learn the details of how to integrate functions, instead you will learn the concept and then you will approximate it by using random sampling.

Problem 1: A person is travelling at a constant velocity (v) of 10 m/s. How far do they travel in 20 seconds?

Approach 1: Use substitution to solve for displacement (s) (this is similar to distance).

$$v = \frac{s}{t} (Equation\ 1)$$

$$10 = \frac{s}{20}$$

$$10 \times 20 = s = 200\,metres$$

Approach 2: Taking the integral of the function (you will not be required to do this). Let y represent velocity and x represent time.

$$y = 10\ (Equation2)$$

$$\int_0^{20} 10dx = 10(20) - 10(0) = 200\,metres$$

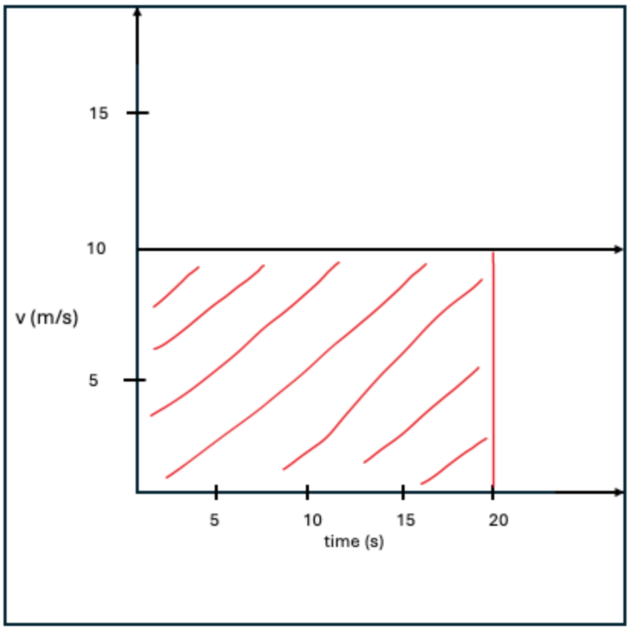


Figure 1

Above is a visual representation of what the integral calculation did, it calculated the area under the curve which was the answer to the problem between the x values of 0 and 20.

This was an alternate approach to just solving for s in equation 1. It is also easy to verify that our integral is correct as the area under the curve is just a rectangle, and length times width gives us 200.

You might be wondering why integration is such a big deal if we could solve this problem otherwise?

The answer is we can find areas under graphed functions, even when the graphed line produces a form more complicated than a straight line (like in the very first diagram).

Remember, you will not be required to integrate equations in this investigation (we will save that for year 12). The above overview is intended to show you what is possible.

1.1 Approximating Integration

1.1.1 Approach 3: The final approach we will look at for solving problem 1 will not give us an exact solution but instead will give us an approximate one.

We can approximate the solution to the integral of equation 2 by using repeated random sampling. Using this technique of repeated random sampling is your main task for part 1.

Below is a visual representation of using this approach to solve problem 1. Pairs of random values are generated repeatedly from 0 to 20 inclusive. One of each pair corresponds to the x dimension and the other to the y dimension, together each of these randomly generated values produce a coordinate on the Cartesian plane.

A ratio is produced by taking the total count of coordinates that are generated which fall between our function and the x-axis (corresponds to the blue dots) and then dividing this by the total amount of coordinates we generated (the coordinates of red and blue dots). This ratio is then multiplied to the total area in order to estimate the area of the region under the curve.

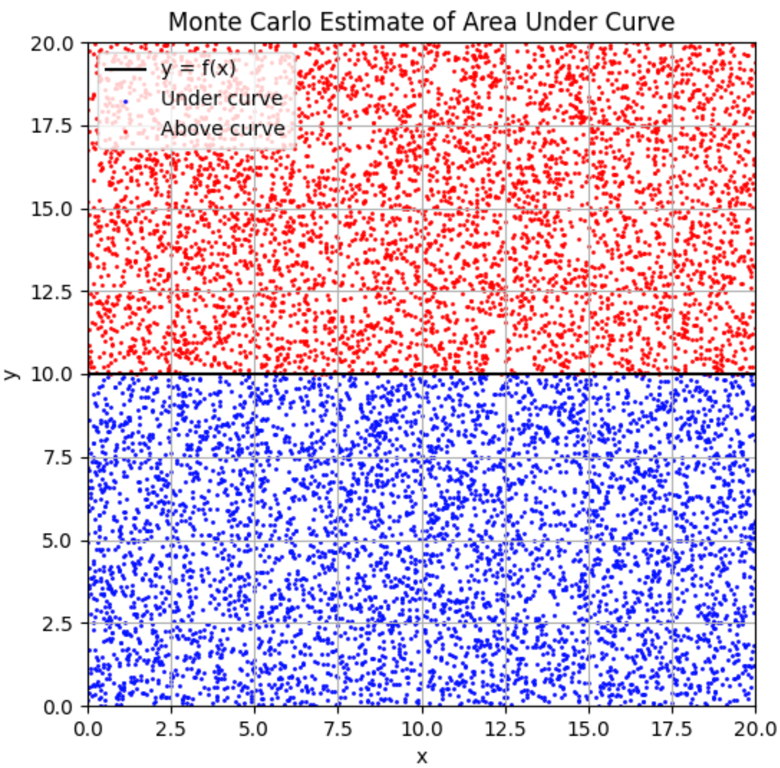


Figure 2

```
Monte Carlo Estimate: 198.64000
Exact Value: 200.00000
Error: 1.36000
```

The “Monte Carlo Estimate” is the name given to the process above which gives an approximation of integration. As you can see, the approximation in this run was close, and only off by 1.36. What is the most obvious way to improve this?

View example code file “SimpleUniform.py” to see how this image and approximation was generated. Use this file to help provide an approximate solution to problems 2 and 3.

1.1.2 Problem 2: The rate of change of temperature is modelled by equation 3. Approximate the integral of this function for $0 \leq t \leq 3$.

Rate of change T:
$$T = t^2 (Equation 3)$$

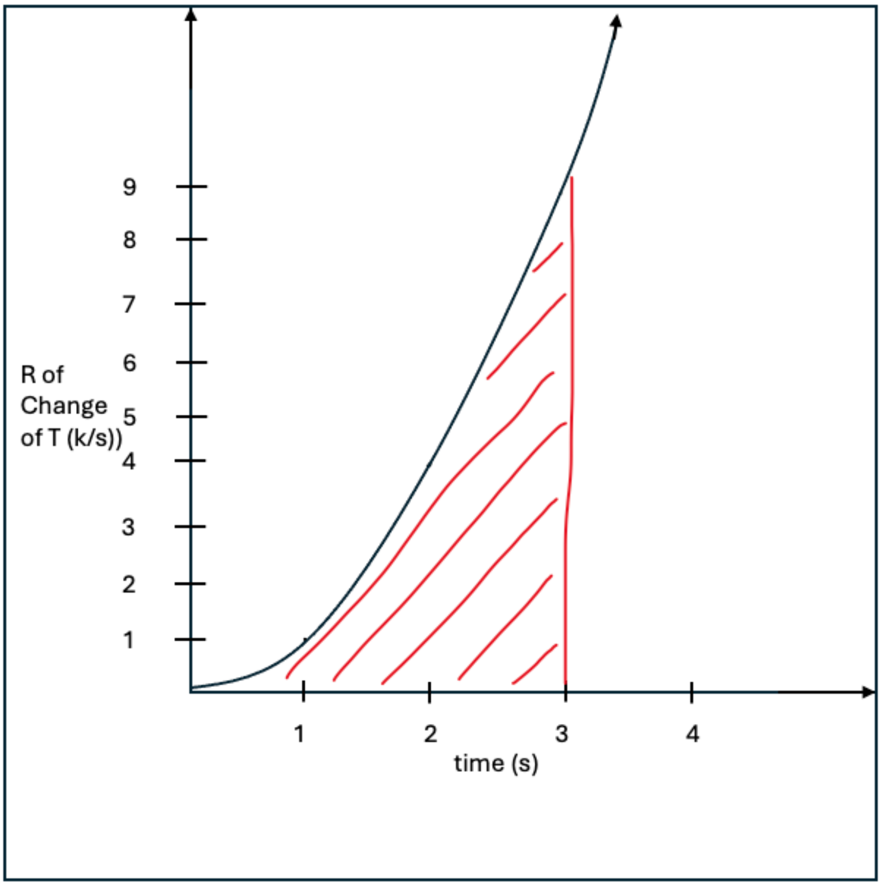
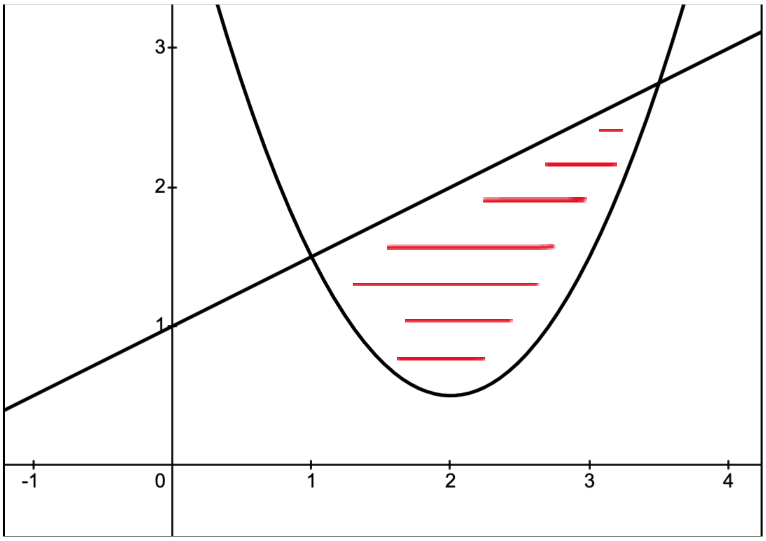


Figure 3

- Your code should generate an image of the graphed function which visually represents the sampling you have done just as the sampling down in “SimpleUniform.py” shows a visual representation. Your slides should include this image along with a snip of your code and the approximation your code generated. **(5 marks)**

1.1.3 Problem 3: Integration allows us to also determine the space between graphed functions by taking the difference of the areas produced between the line and the curve. Approximate the highlighted area between the following functions: $y = \frac{1}{2}x + 1$ and $y = (x-2)^2 + 0.5$ for $0 \leq x \leq 4$.



- Your code should generate an image of the graphed functions which visually represents the sampling you have done just as the sampling down in "SimpleUniform.py" shows a visual representation. Your slides should include this image along with a snip of your code and the approximation your code generated. **(5 marks)**

1.2.Sampling Pixel Values from an Image.

1.2.0: Pixel Tutorial (sample.png)

In this short tutorial you will:

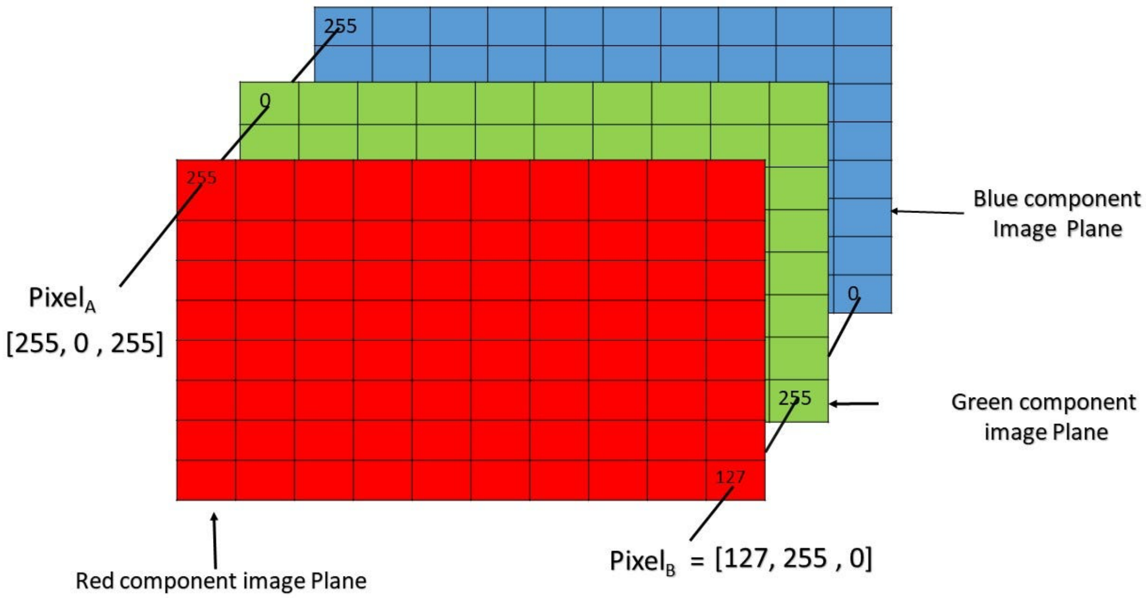
- Learn how colours are represented in digital images.
- Learn to edit pixel values to change image colour.

You will combine this knowledge with approximation techniques in later sections.

Download "Pixel_Tutorial.py" and "sample.png". Ensure they are both in the same folder on your computer, and then run the Python file. It will display "sample.png" twice, but one time will be after it has been edited. Some things to consider:

- How is colour represented in the image?
    - What number ranges are used?
- How was colour changed in the image?
- How can you check the colour of a pixel in the image?
- How could you identify if a region of a particular image is a certain colour?

To assist with your understanding, below is a diagram of an 8x10 RGB image. Computer images in computers are generally represented by three two-dimensional arrays with integer values in each array cell being between 0 and 255. For RGB images, the first value is red and the last is Blue. For BGR images the order of reversed. You can think of an array as a table.



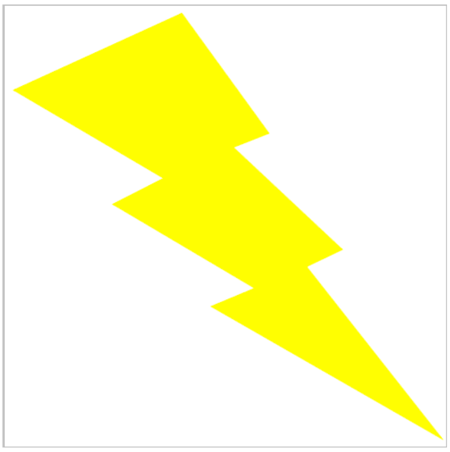
Pixel of an RGB image are formed from the corresponding pixel of the three component images

- Your presentation should include a slide with your answers to the above questions. **(5 marks)**

2.1 Estimating the Area of an Irregular Polygon

2.1.1 Determining area (bolt.png)

Using the techniques and knowledge already explored. Write an algorithm that includes random sampling which estimates the percentage of the image below which is covered by the lightning bolt. Sample no more than 5000 data points.



- Your code should generate an image which shows how "bolt.png" was sampled. For example, "SimpleUniform.py" generates an image which shows the sampling by representing each point that was sampled as a coloured dot. Include a snapshot of this image in your slides along with your approximate percentage and a snip of your code. Also include either a brief text or flowchart description of how your algorithm works. **(5 marks)**

3 Image Based Sampling

3.1.1 Ratio of Board to Scoring Area - Exact (3.1.png)

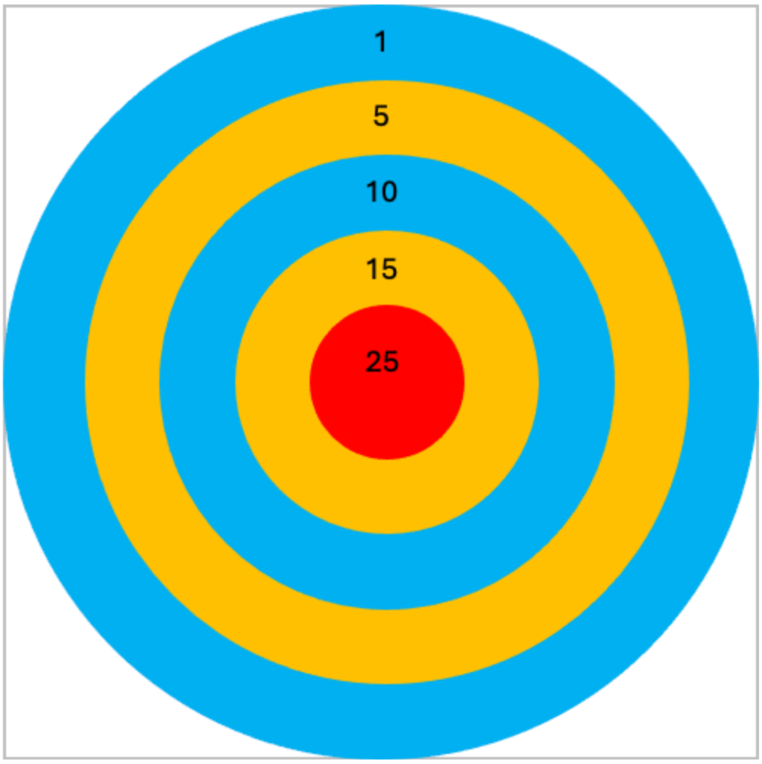
Work out an exact value of the ratio of the scoring area to the total area of the board (Coloured area is the scoring area). Hint: Assume board has dimensions 1 unit by 1 unit.

- Include your working and solution in your presentation. **(5 marks)**

3.1.2 Ratio of Board to Scoring Area - Approximate (3.1.png)

Now produce an algorithm that will use random sampling to approximate the ratio of the scoring area to the total area of the board.

- Your code should generate an image which shows how "3.1.png" was sampled. In your slides include a screenshot of your code and your approximated ratio in your slides. Also include either a brief text or flowchart description of how your algorithm works. **(5 marks)**



3.1.3 Probability of Scoring 10 or more - Exact

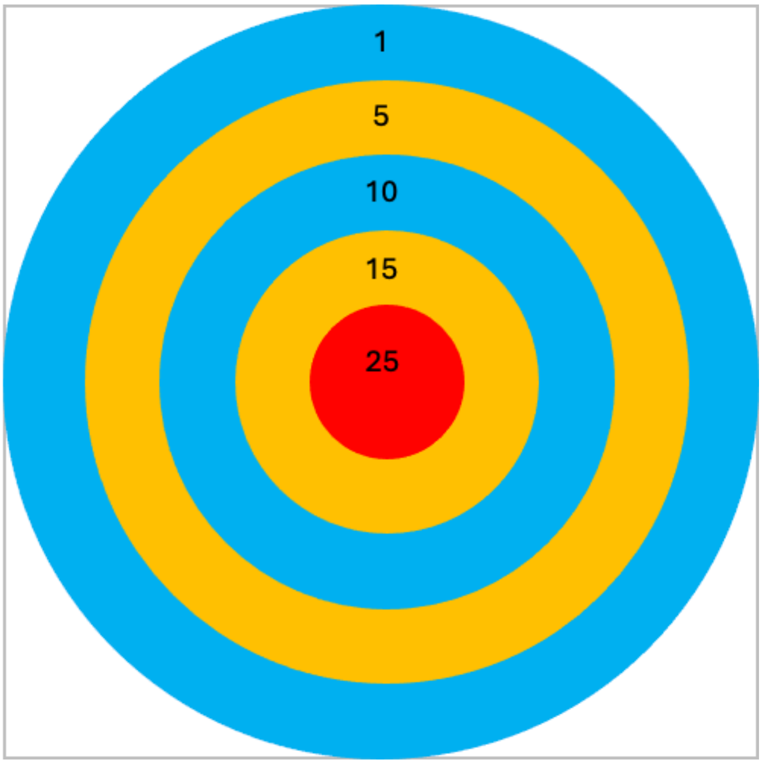
If the inner red circle has a radius of r_1, and each circle on the board is larger than the immediately smaller circle by 2 r_1. Work out the exact likelihood that a person will score 10 or more.

- Include your working and solution in your presentation. **(5 marks)**

3.1.4 Probability of Scoring 10 or more - Approximate (3.1.png)

Write an algorithm that provides an estimate that a person will score 10 or more. Just like in section 3.1.3, you should assume that each circle in the board is larger than the immediately smaller circle by 2 r_1, with r_1 being the radius of the small red circle.

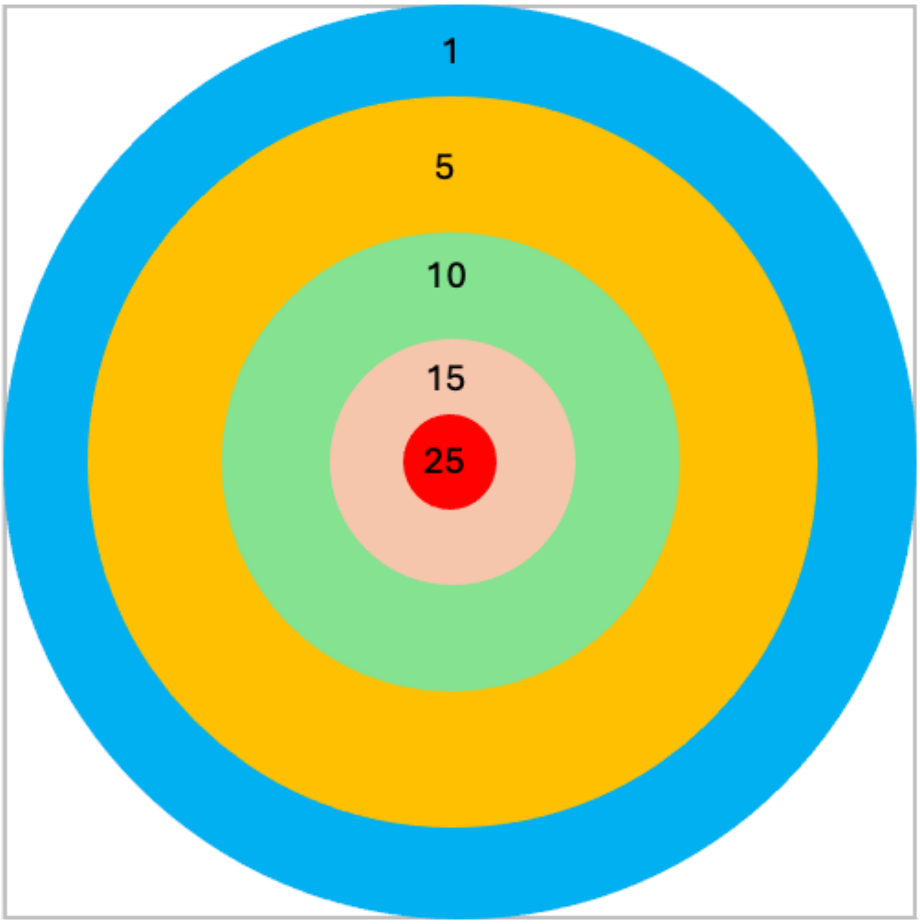
- Your code should generate an image which shows how "3.1.png" was sampled. In your slides include a screenshot of your code and your approximated ratio in your slides. Also include either a brief text or flowchart description of how your algorithm works. **(5 marks)**



3.1.5 Determining Coloured Regions (3.3.png)

In the following image, the size of the radius of each circle is ambiguous, yet each ring is of a distinct colour. Use random sampling to approximate the number of pixels of each colour in the image (blue, gold, green, peach, red).

- Your code should generate an image which shows how "3.3.png" was sampled. In your slides include a screenshot of your code and your approximated number of pixels in your slides. Also include either a brief text or flowchart description of how your algorithm works. **(5 marks)**



Note: The RGB colours for the different regions are:

- Blue: (59,163,234)
- Gold: (247,188,43)
- Green: (138,219,138)
- Peach (238,193,165):
- Red: (237,49,25)

4 Distribution Exploration

In this section you will continue random sampling from images, but you will graph the results as histograms and consider the shape of the distributions formed.

4.1.1 Time until outcome

Write a program that will sample pixel values from "0.5_2.png". This program should run an experiment where the number of trials required until a red pixel is selected is saved. You should run this experiment 20000 times and save the results in a text file. Graph the results (referred to as Graph A later) into a probability distribution using matplotlib or excel.

- In your slides include a copy of the graph you generated from your results and a snip of the code that you used to generate the data. Also include either a brief text or flowchart description of how your algorithm works. **(5 marks)**

4.1.2 Average of Times

In 4.1 you should have generated 20000 results which you then represented in a distribution. Write a program that will load in these results and group them into collections of 20 and then take the average of each 20. For example, the results 1 to 20 would be summed and divided by 20 and this average would serve as the first data point of a new set of results, results 21 to 40 would be summed and divided by 20 and this average would serve as the second data point of a new set of results. Your code should end up producing a second set of results that has 1000 data values. Graph the results (referred to as Graph B later) into a probability distribution using matplotlib or excel.

- In your slides include a copy of the graph you generated from your results and a snip of the code that you used to generate the data. Also include either a brief text or flowchart description of how your algorithm works. **(5 marks)**

4.1.3 Distribution Conversion

Your Graph A should show what is known as a geometric distribution, your Graph B should show what is known as a normal distribution – feel free to google these distributions to compare the shape of your graphs. Investigate why taking the average of your results from 4.1.1 in batches caused a normal distribution to be generated for Graph B.

- Include a slide which explains why taking the average of your results generated a normal distribution. **(5 marks)**

5 Live Testing

The final part of the assignment will involve live testing in test conditions. To prepare for the live section, please ensure you are familiar with the concepts this investigation explores and that you have understood how your code works. You will not have to write code, but you will need to understand the general algorithms which you have built and the related concepts and be able to write out relevant explanations.

(15 marks)

6 Git

As with past assignments you are expected to use Git to track your progress. At the outset of your project, you should initialise a git repository in a folder named in the following form:
FirstName_Surname_9_Cat_2

So John Smith's folder will be named as follows: John_Smith_9_Cat_2. You will initialised a git repository in this folder and it will be where you store all your assignment files. When working through this assignment, you may decide whether you produce one big program that has different modules that complete separate sections, or whether you make a smaller python program for each section.

You should regularly make commits as you progress through the assignment. Each commit should have a description of what progress you have made. Frequent and detailed commits are required to earn marks for this section. Consideration will also be given to use of comments and code readability.

(25 marks)

Presentation

You will present your assignment to a small group as you have for past coding assignments. This presentation should go for roughly 12 minutes per person. Please have your code ready to be run for the presentation so you may demonstrate its effectiveness to your group.

Submission

Your assignment is due on the 24th of October, a submission link will be emailed out nearer to the date. You are not permitted to edit your presentation slides after you have submitted your project.