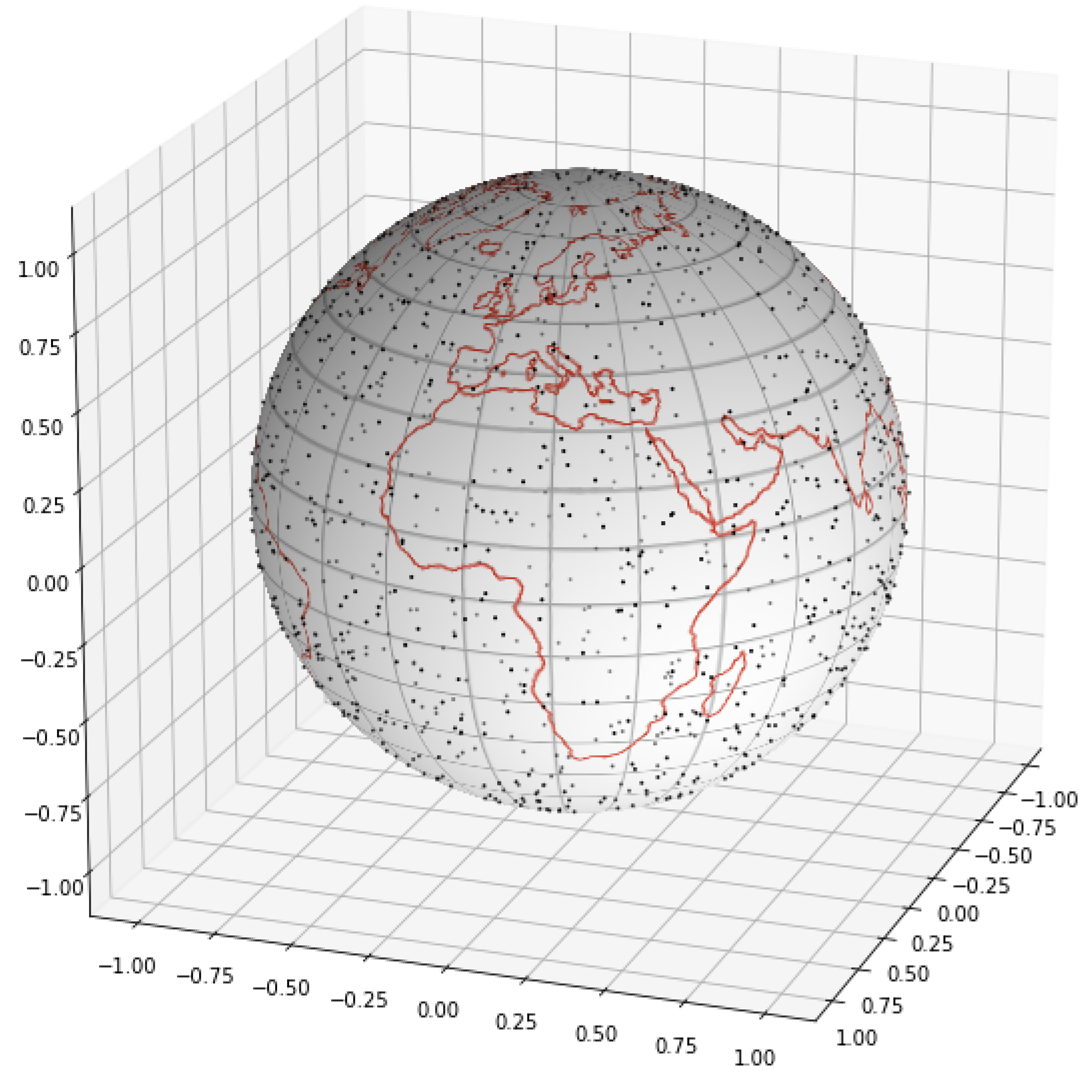
Computer Project - Seena Pourzand

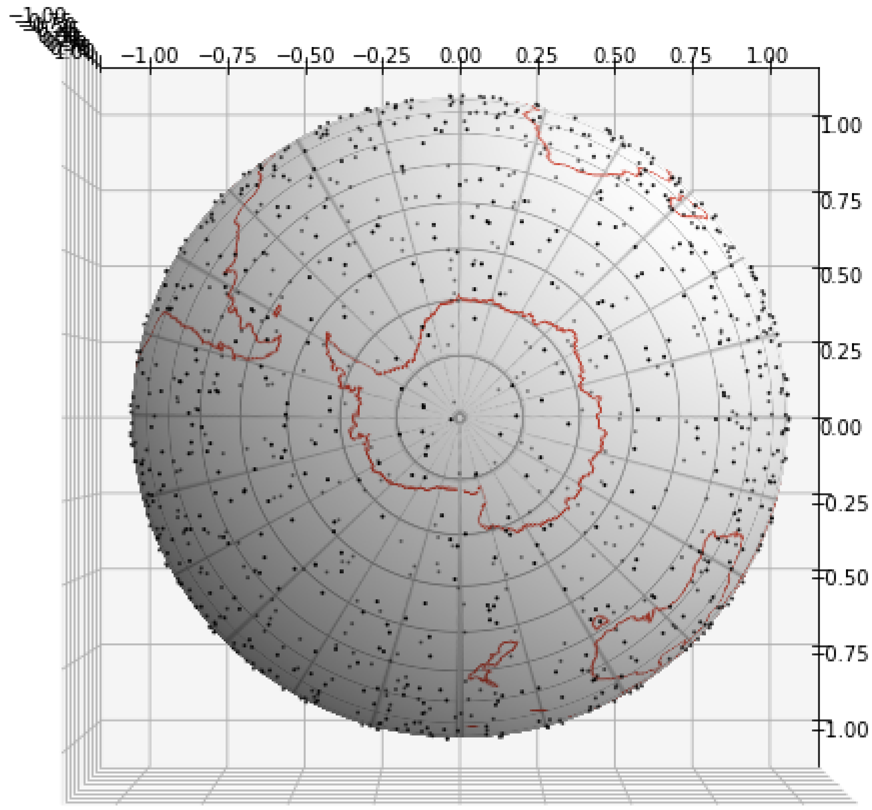
A Link to my code can be found here:
https://colab.research.google.com/drive/1epFFuHbw42wHJ0hNNrnyKo7MO1O0ob3p?usp=sharing

More indepth explanation behind the code can be found there as well as here in this write up

1. Problem 1: Find a procedure for sampling uniformly on the surface of the sphere.
    a. (a) Use computer to generate a thousand points that are random, independent, and uniform on the unit sphere, and print the resulting picture.
    b. (b) By putting sufficiently many independent uniform points on the surface of the Earth (not literally but using a computer model, of course), estimate the areas of Antarctica and Africa, compare your results with the actual values, and make a few comments (e.g. are the relative errors similar? would you expect them to be similar? if not, which one should be bigger? how does accuracy improve if you use more points? etc.)
    c. My Approach:
        i. I first generated random points for a sphere using a random gaussian distribution to ensure the points were generated on the surface of the sphere as opposed to the points being throughout the sphere, both inside and on the surface.
        ii. From there I plotted the points on a sphere with an outline of the Earth to help give context to the task at hand
        iii. Africa:



        iv. Antartica:



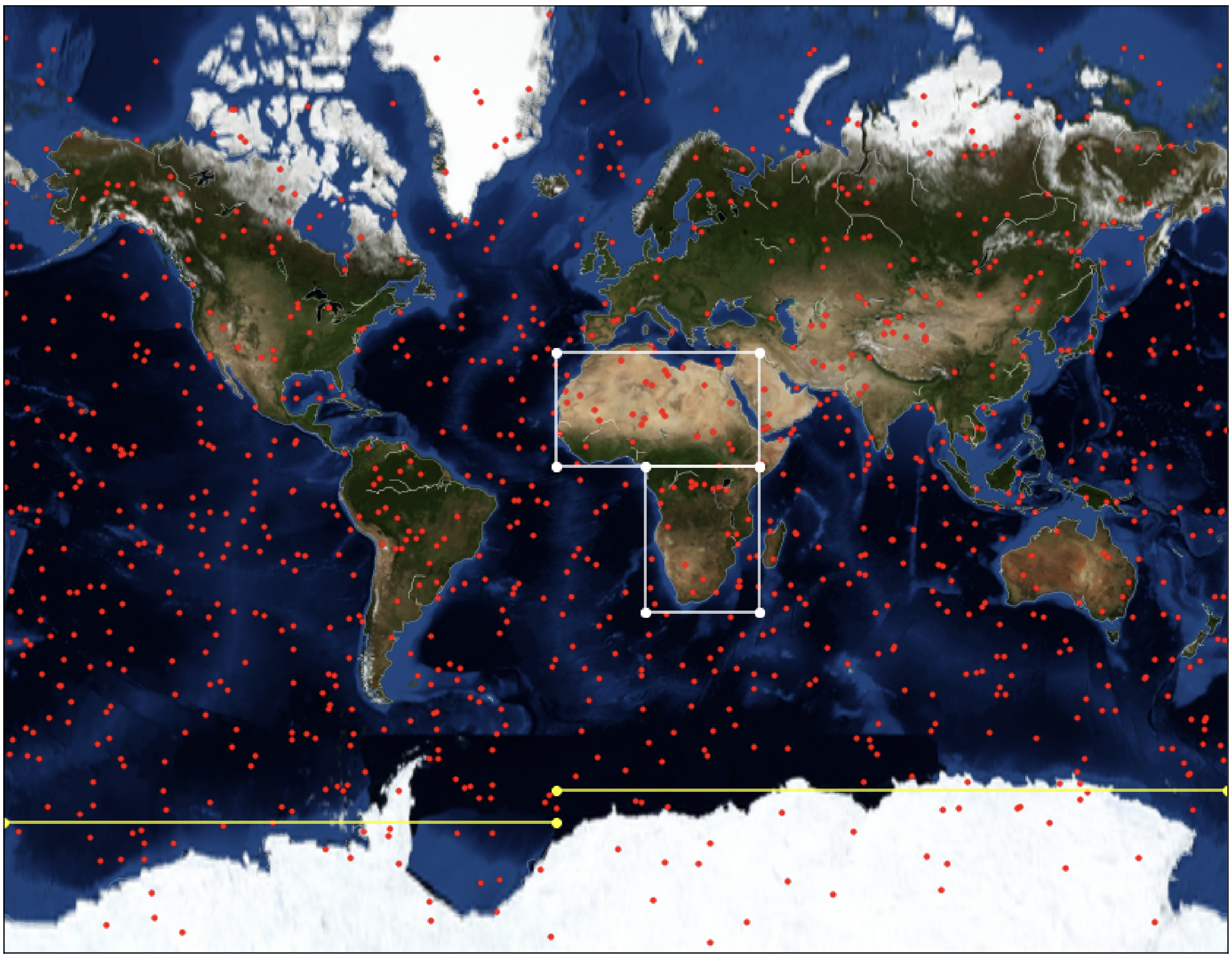
v. Next to help automate the counting process, I converted the sphere to a 2D representation using a techinque called UV mapping which uses the following formulas to convert 3D coordinates to 2D

$$u = 0.5 + \frac{\arctan 2(d_z, d_x)}{2\pi}$$

$$v = 0.5 + \frac{\arcsin(d_y)}{\pi}$$

1. where arctan2 is the element-wise arc tangent, provided with two arguments

vi. Once the coordinates where converted I then graphed them on a flattened map



vii. Additionally, I plotted bounding boxes around the continents of intersts that were as close I could get with the tools available and helped get a rough estimate of the regions. With a few if statements that check if the x and y coordinates of a point fell within these white and yellow bounding boxes, I then added them to the counter

1. Africa: My program counted 73 points of out of the 1000 on the continent, thus we can estimate 7.3% of the Earth is covered by Antartica according to my program. In actuality Antartica covers 6.0% of the Earth, so a slight overestimate
   a. Relative Error: $\frac{7.3-6}{6} * 100 = 21.167\%$
   b. Source: https://artsandculture.google.com/entity/africa/m0dg3n1?hl=en
2. Antartica: My program counted 38 points of out of the 1000 on the continent, thus we can estimate 3.8% of the Earth is covered by Antartica according to my program. In actuality Antartica covers 2.745% of the Earth, so a slight overestimate
   a. Relative Error: $\frac{3.8-2.745}{2.745} * 100 = 38.433\%$
   b. Source: https://www.weforum.org/agenda/2021/01/earth-surface-ocean-visualization-science-countries-russia-canada-china
3. Both my errors were slightly overestimates likely due to my overestimate bounding boxes that were used to count the number of points. Additonally some error was introduced from converting from 3D to 2D since there is typically warping that occurs that would have affected how some points were displayed, especially near the poles of the Earth. When using 2000s points my errors tends to stay relatively the same, dropping a bit, but indicating my counting method is likely the source of the issue

2. Problem 2: Get a computer program for distinguishing a randomly generated sequence of zeroes and ones from a cooked-up one. You are welcome to write the program yourself or use what can you find on the web or in some book. Test your program on the following two sequences: the sequence consisting of the concatenation of all numbers in binary form:
0 1 10 11 100 101 110 111 1000 . . .
and a similar sequence consisting of the concatenation of all prime numbers in binary form
0 1 10 11 101 111 1011 . . .
The first sequence (the fractional part of the Champernowne number) is known to be random when considered in base 10; the second sequence (the fractional part of the Copeland-Erd¨os constant in binary form) is known to be

random. In both cases, randomness is understood in a very specific way, and you are welcome to discuss this point too.

a. My Approach:
    i. Firstly I created methods to generate both a sequence of concatenated consecutive binary numbers from scratch. I then adapted that same method to create a sequence of concatenated prime numbers
    ii. From there I went about calculating the sum of whole sequences and dividing them by their number of digits to find their means as according to the law of large numbers, as you increase the size of the sequence it should in theory approach the mean.
    iii. When comparing the results it is clear with this test, the two given sequences are not as random as the computer generated as the computer generated was 9.3600000000027e-05 off from the mean of .5 whereas the two provided sequences had a much larger difference 0.03308847633175804(nonprime) and (0.08287478223293943) respectively

```
Mean of Concatenation of Numbers in Binary Sequence:
0.516544238165879
0.03308847633175804
Mean of Concatenation of Prime Numbers in Binary Sequence:
0.5414373911164697
0.08287478223293943
Mean of Concatenation of Computer Generated Random Sequence:
0.4999064
-9.3600000000027e-05
```

    iv. With this test, one could conclude these were non random
    v. However, using the Binary Matrix Rank Test as suggested by NIST, I get much different results as all three sequences, the computer generated random as well as the two concateanted were random as they yielded the same result of zero. The test creates a binary seqence in a 32 x 32 matrix and attempts to caluclate their rank aka seeing how many of the rows are "unique" or not made of other rows.
    vi. What is clear is that randomness in this sense is subjective to the test you run and how that defines randomness

b.