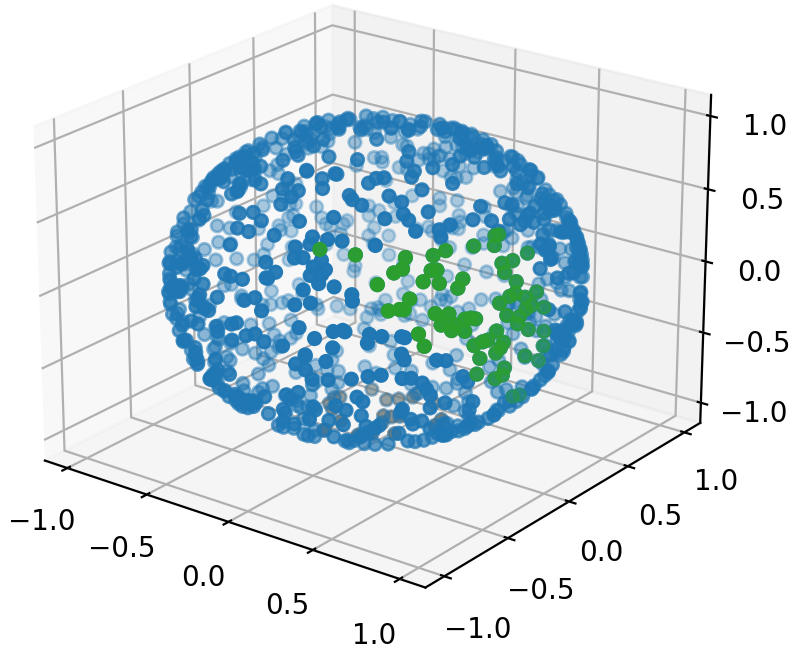
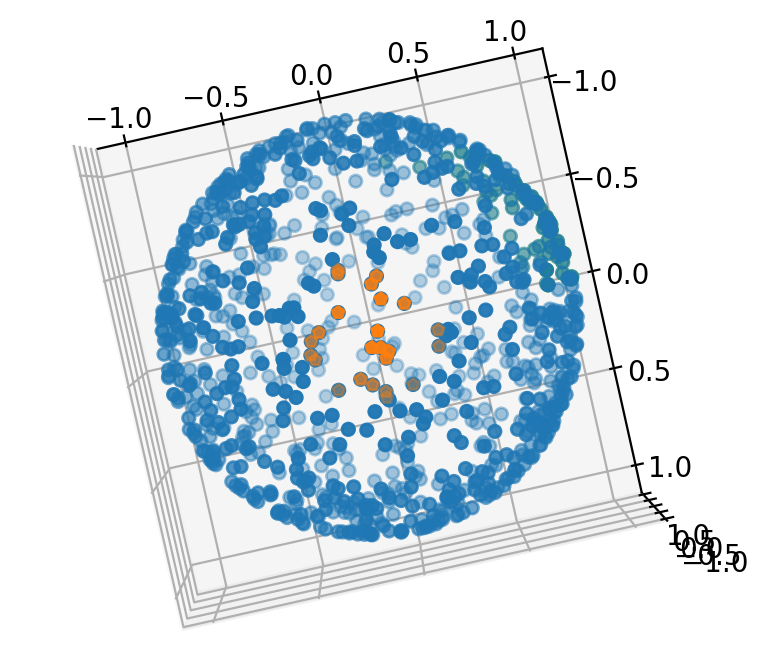
Problem 1

1. Plot points on sphere

I use three random variable x, y, z with standard normal distribution, and then I calculate lamda = sqrt(x^2 + y^2 + z^2). The point(x / lamda, y / lamda, z / lamda) must be on the surface of the sphere. Generate 1000 of those points and plot then using matplotlib.

1. Estimate area of Africa & Antarctica

I estimate the area of Africa using a triangle, whose height spans central 60% of the latitude, and width spans 1/5 of the longitude. Points within the triangle are counted as in Africa. The green points showing to the right are points in Africa. By calculating points in Africa over total points, P(Points in Africa) = 0.068. Given that the total surface area of earth is 510.1M km^2, the estimate area of Africa is 34.69M km^2.



I estimate the area of Antarctica using a round, whose height is 2.75% of the lowest latitude. Count the points in Antarctica, orange points represent points in Antarctica (shown right) By calculating points in Antarctica over total points, P(Points in Antarctica) = 0.024, so the estimate area of Antarctica is 12.24M km^2

Do the estimation for 10 times with 1000 points, 10 times with 10000 points, we get table below

1000 points:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Antarctic | 16.32 | 14.28 | 13.77 | 10.71 | 13.71 | 15.81 | 14.34 | 11.22 | 13.77 | 12.75 |
| Africa | 35.20 | 32.65 | 23.97 | 27.55 | 30.10 | 32.14 | 29.08 | 26.02 | 28.57 | 32.14 |

Antarctica: mean: 13.67 standard deviation: 1.67 actual: 13.66

Africa : mean: 29.74 standard deviation: 3.22 actual: 30.37

10000 points:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Antarctic | 12.85 | 14.28 | 13.98 | 13.77 | 14.33 | 15.00 | 13.47 | 14.84 | 13.98 | 13.79 |
| Africa | 32.19 | 30.20 | 30.04 | 31.52 | 29.79 | 31.78 | 30.55 | 30.30 | 31.47 | 30.55 |

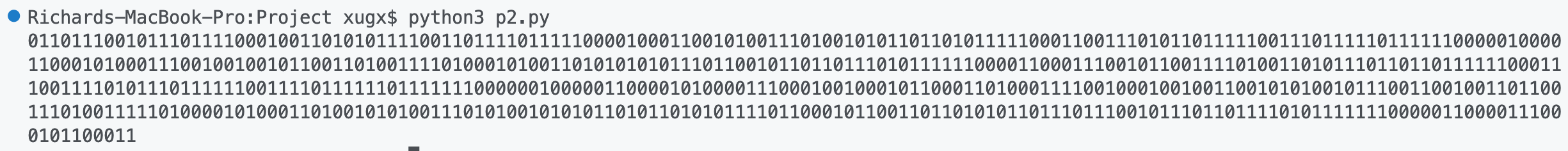
Antarctica: mean: 14.01 standard deviation: 0.60 actual: 13.66

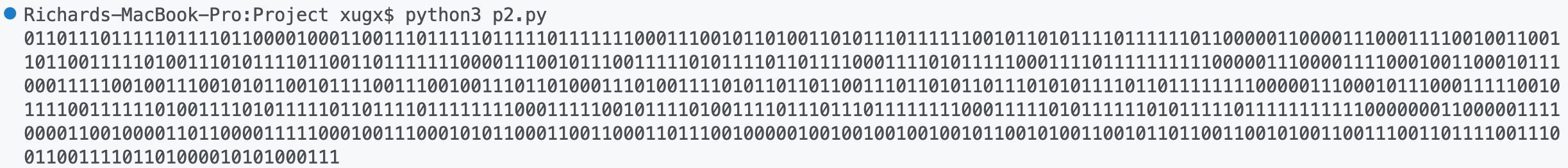
Africa : mean: 30.84 standard deviation: 0.79 actual: 30.37

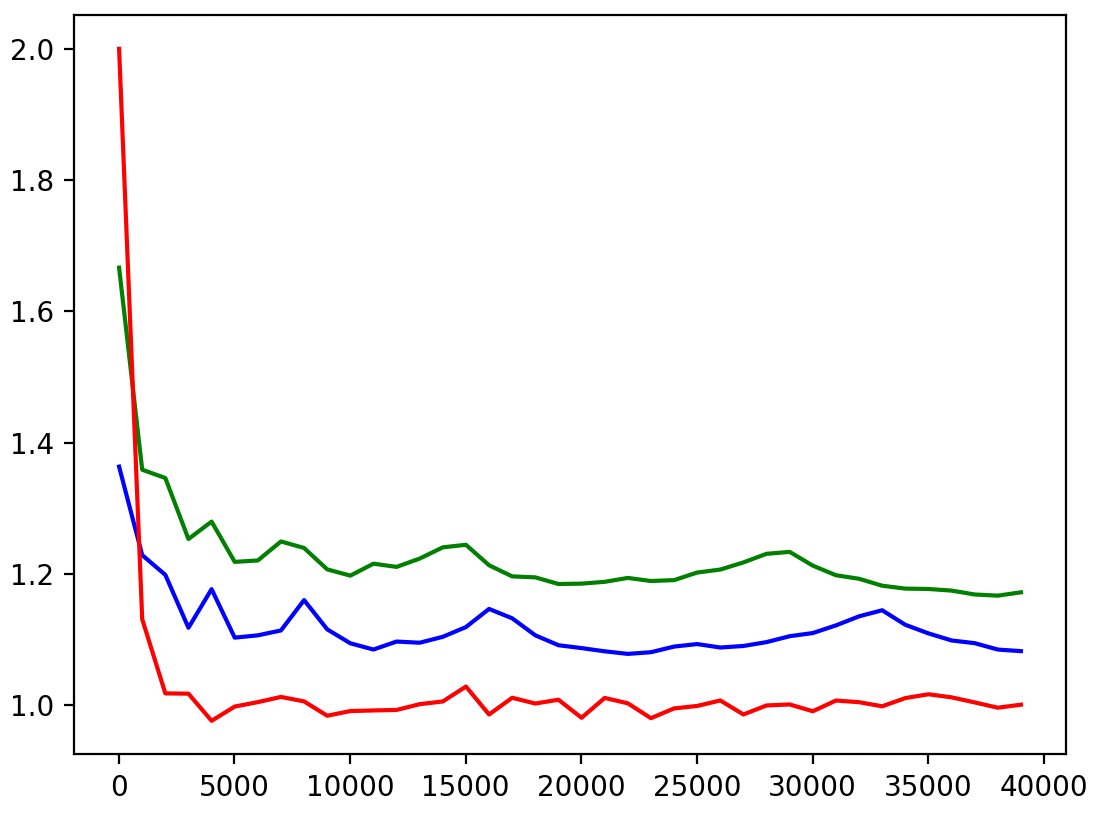
Notice that for 10000 points, the standard deviation is smaller than it with 1000 points. But still, the standard deviation of Africa is larger than Antarctica. This may because the area of Africa is larger, which cause the estimation distribute more widely. The mean is more accurate for Antarctic, this may because the way I estimate Antarctica(using circle) is more accurate and closer to the real shape than the way I estimate Africa (using triangle)

Problem 2

Let’s first explore Champernowne & Copeland-Eros constant

Champernowne constant (base 2) [concatenate first 100 number]

Copeland-Eros constant (base 2) [concatenate first 100 prime]



Test if they’re random enough. If a string of 0s and 1s are truly random, the ratio between 1 and 0 should be close to 1, and keep getting closer as input string length increase. My program plot that ratio over the number length. (shown right)

As number of concatenation increase, the ratio of 1s and 0s become closer and closer to 1, which means the data is quite random.

- The red line represent a truly randomly generated sequence, which quickly converge to 1 as number of concatenation increase.

- The blue line represent the Champernowne constant, which also converge to 1, but in a slower speed

- The green line represent Copeland-Eros constant, which converge to 1 in a lower speed than Champernowne constant

Thereby, we could conclude that Champernowne is more random than Copeland-Eros constant by comparing ratio of 1s and 0s

Also, Champernowne & Copeland-Eros converge to 1 from the top, which means the number of 1 is always larger than 0, but that difference is decreasing. This indicates that those two number may not be as random as it appear. For a truly random sequence, however, it quickly converge to 1 and fluctuate.

We can conclude that if the number length goes to infinity, Champernowne & Copeland-Eros constant are possibly random and likely to be normal number. Given input with other string, we can check if it’s random by looking at the generated graph to see if it converge or have tendency to converge to 1 as input length increase.

\* here, randomness is defined by the equality of frequency of number appeared for each digit, which correspond to the definition of normal number.