Based on
Bertrand's Paradox (with 3blue1brown) - Numberphile
https://www.youtube.com/watch?v=mZBwsm6B280

VERSION 1: Pick two points (angles) on the unit circle, create the chord connecting them. Ask if it is greater than or less than sqrt(3).

In[1]:=
```
NumSamples = 1000;
GTCount = 0;
For[i = 1, i ≤ NumSamples, i++,
Angle1 = RandomVariate[UniformDistribution[{0, 2 Pi}]];
Angle2 = RandomVariate[UniformDistribution[{0, 2 Pi}]];
r = Sqrt[(Cos[Angle1] - Cos[Angle2]) ^ 2 + (Sin[Angle1] - Sin[Angle2]) ^ 2];
If[r > Sqrt[3], GTCount++,];
](* end for loop over samples *); Print[N[GTCount / NumSamples]]
```

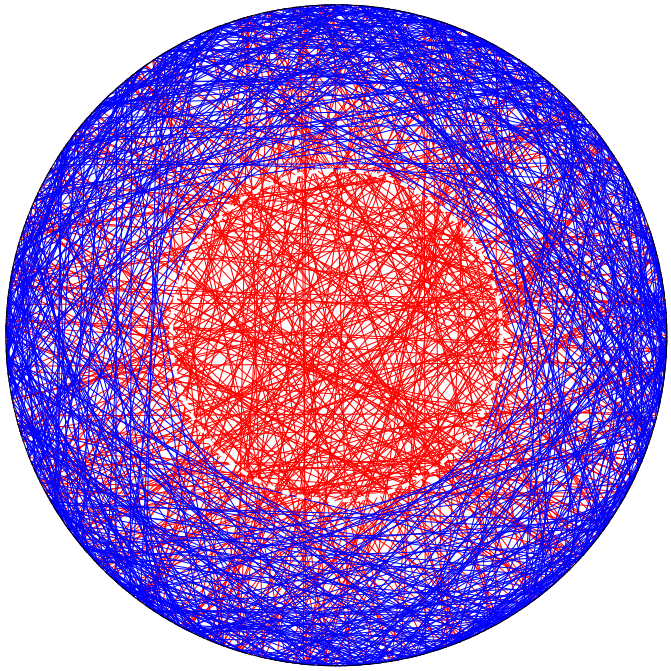
```
0.362
```

In[4]:=
```
NumSamples = 1000;
GTCount = 0;
smaller = {}; bigger = {};
For[i = 1, i ≤ NumSamples, i++,
Angle1 = RandomVariate[UniformDistribution[{0, 2 Pi}]];
Angle2 = RandomVariate[UniformDistribution[{0, 2 Pi}]];
r = Sqrt[(Cos[Angle1] - Cos[Angle2]) ^ 2 + (Sin[Angle1] - Sin[Angle2]) ^ 2];
If[r > Sqrt[3],
GTCount++;
AppendTo[bigger, Line[{{Cos[Angle1], Sin[Angle1]}, {Cos[Angle2], Sin[Angle2]}}]],
AppendTo[smaller, Line[{{Cos[Angle1], Sin[Angle1]}, {Cos[Angle2], Sin[Angle2]}}]]
];
](* end for loop over samples *); Print[N[GTCount / NumSamples]]
```

```
0.301
```

In[8]:=
```
Show[Graphics[{Red, bigger}], Graphics[{Blue, smaller}],
 Graphics[Circle[{0, 0}, 1]], Graphics[{White, Thick, Dashed, Circle[{0, 0}, 0.5]}]]
```

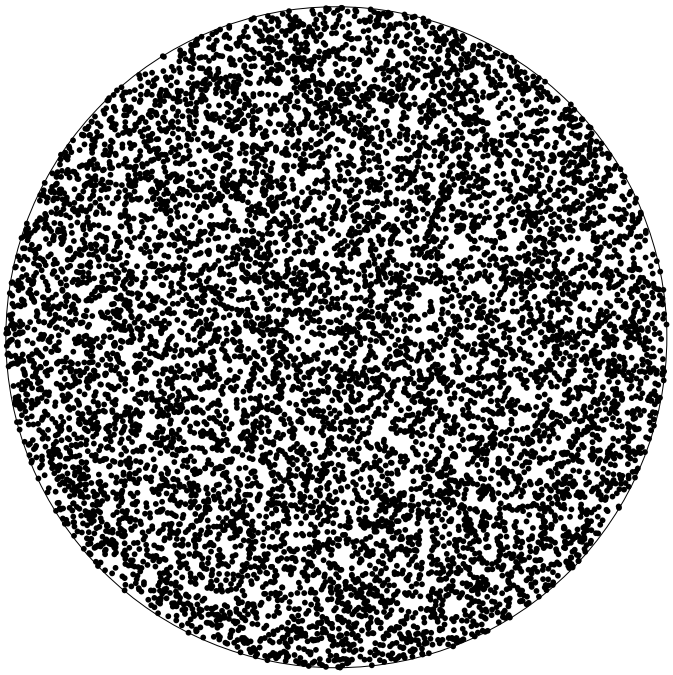
Out[8]=



(Looks like chords going with a midpoint inside a circle with radius 1/2 fit criterion)
VERSION 2: Pick a random point, make it the midpoint of a chord, etc.

In[26]:=
```
NumSamples = 10 000;
MyPoints = {};
For[i = 1, i ≤ NumSamples, i++,
MyAngle = RandomVariate[UniformDistribution[{0, 2 Pi}]];
MyRadius = Sqrt[RandomVariate[UniformDistribution[{0, 1}]]]; (* Note the sqrt*)
AppendTo[MyPoints, Point[{MyRadius * Cos[MyAngle], MyRadius * Sin[MyAngle]}]];
](* end for loop over samples *); Show[Graphics[MyPoints], Graphics[Circle[{0, 0}, 1]]]
```

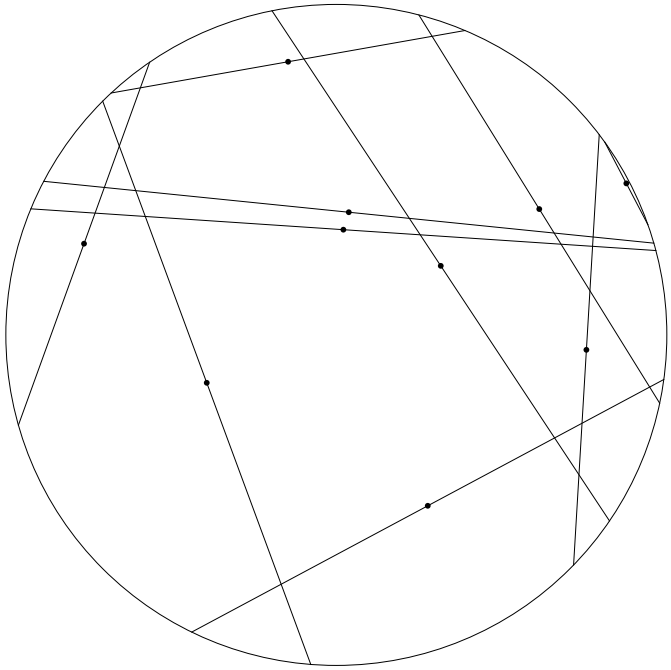
Out[28]=



First image point is a distance r from the center on the horizontal. Then a point on the unit circle at an +/- angle of ArcCos[r] would have same horizontal as the original point. These would be the endpoints of the chord. Then rotate the points by the angle.

In[9]:=
```
NumSamples = 10;
MyPoints = {};
MyChords = {};
For[i = 1, i ≤ NumSamples, i++,
MyAngle = RandomVariate[UniformDistribution[{0, 2 Pi}]];
MyRadius = Sqrt[RandomVariate[UniformDistribution[{0, 1}]]]; (* Note the sqrt*)
AppendTo[MyPoints, Point[{MyRadius * Cos[MyAngle], MyRadius * Sin[MyAngle]}]];
x1 = MyRadius * Cos[MyAngle] - Sin[ArcCos[MyRadius]] * Sin[MyAngle];
y1 = MyRadius * Sin[MyAngle] + Sin[ArcCos[MyRadius]] * Cos[MyAngle];
x2 = MyRadius * Cos[MyAngle] + Sin[ArcCos[MyRadius]] * Sin[MyAngle];
y2 = MyRadius * Sin[MyAngle] - Sin[ArcCos[MyRadius]] * Cos[MyAngle];
AppendTo[MyChords, Line[{{x1, y1}, {x2, y2}}]]
](* end for loop over samples *);
Show[Graphics[MyPoints], Graphics[MyChords], Graphics[Circle[{0, 0}, 1]]]
```

Out[12]=

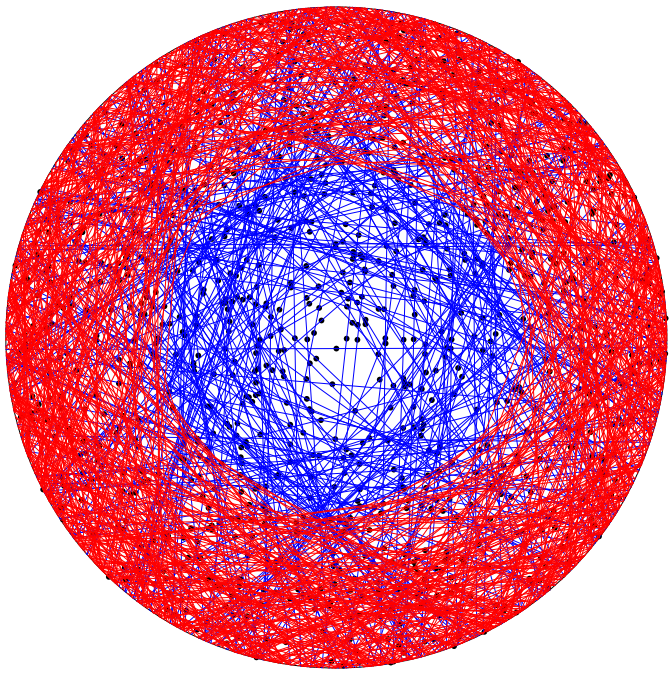


In[19]:=
```
NumSamples = 1000;
GTCount = 0;
MyPoints = {};
MyChords1 = {};
MyChords2 = {};
For[i = 1, i ≤ NumSamples, i++,
MyAngle = RandomVariate[UniformDistribution[{0, 2 Pi}]];
MyRadius = Sqrt[RandomVariate[UniformDistribution[{0, 1}]]]; (* Note the sqrt*)
AppendTo[MyPoints, Point[{MyRadius * Cos[MyAngle], MyRadius * Sin[MyAngle]}]];
x1 = MyRadius * Cos[MyAngle] - Sin[ArcCos[MyRadius]] * Sin[MyAngle];
y1 = MyRadius * Sin[MyAngle] + Sin[ArcCos[MyRadius]] * Cos[MyAngle];
x2 = MyRadius * Cos[MyAngle] + Sin[ArcCos[MyRadius]] * Sin[MyAngle];
y2 = MyRadius * Sin[MyAngle] - Sin[ArcCos[MyRadius]] * Cos[MyAngle];
r = Sqrt[(x2 - x1)^2 + (y2 - y1)^2];
If[r > Sqrt[3],
GTCount++;
AppendTo[MyChords1, Line[{{x1, y1}, {x2, y2}}]],
AppendTo[MyChords2, Line[{{x1, y1}, {x2, y2}}]]
];
](* end for loop over samples *); Print[N[GTCount / NumSamples]];
```

```
0.253
```

In[25]:=
```
Show[Graphics[MyPoints], Graphics[Circle[{0, 0}, 1]],
 Graphics[{Blue, MyChords1}], Graphics[{Red, MyChords2}]]
```

Out[25]=



In[29]:=
```
Simplify[Sin[ArcCos[x]]]
```

Out[29]=

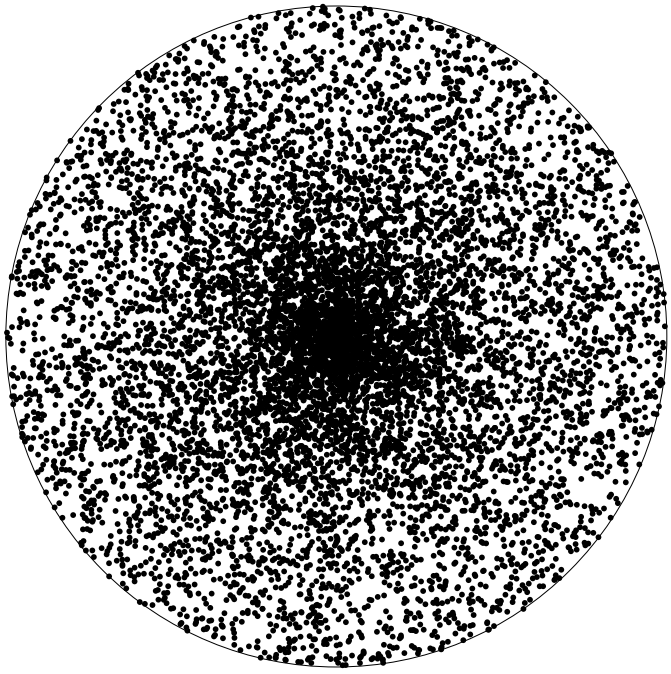
$$\sqrt{1 - x^2}$$

The image above seems have a dearth of chords through the middle. Though the points of the chord seem to be distributed evenly throughout, the chords do not seem to be.

VERSION 3: Pick an Angle (a radial direction), then pick a position along that radial direction (from a uniform distribution). This approach seems to "over-select" points from the center. But appears to yield a more evenly distributed set of chords.

In[142]:=
```
NumSamples = 10 000;
MyPoints = {};
For[i = 1, i ≤ NumSamples, i++,
MyAngle = RandomVariate[UniformDistribution[{0, 2 Pi}]];
MyRadius = RandomVariate[UniformDistribution[{0, 1}]]; (* Note took out sqrt*)
AppendTo[MyPoints, Point[{MyRadius * Cos[MyAngle], MyRadius * Sin[MyAngle]}]];
](* end for loop over samples *); Show[Graphics[MyPoints], Graphics[Circle[{0, 0}, 1]]]
```

Out[144]=

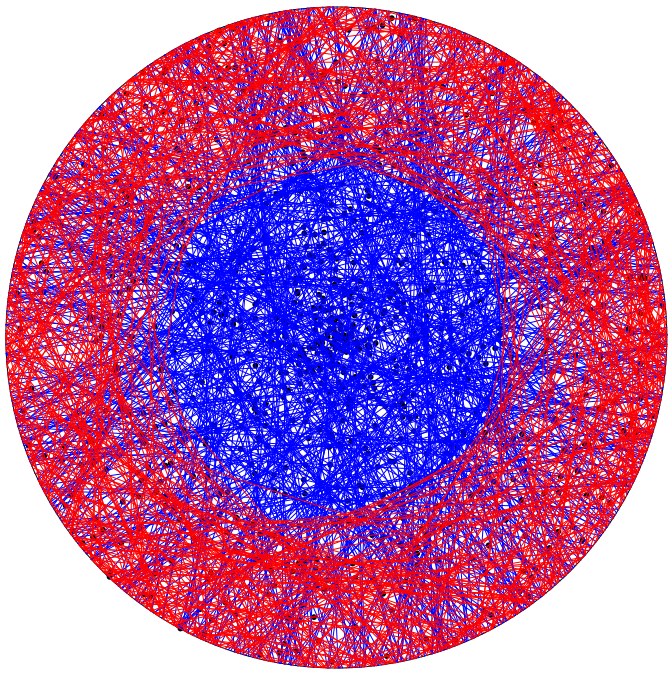


In[30]:=
```
NumSamples = 1000;
GTCount = 0;
MyPoints = {};
MyChords1 = {};
MyChords2 = {};
For[i = 1, i ≤ NumSamples, i++,
MyAngle = RandomVariate[UniformDistribution[{0, 2 Pi}]];
MyRadius = RandomVariate[UniformDistribution[{0, 1}]]; (* Note took out sqrt*)
AppendTo[MyPoints, Point[{MyRadius * Cos[MyAngle], MyRadius * Sin[MyAngle]}]];
x1 = MyRadius * Cos[MyAngle] - Sin[ArcCos[MyRadius]] * Sin[MyAngle];
y1 = MyRadius * Sin[MyAngle] + Sin[ArcCos[MyRadius]] * Cos[MyAngle];
x2 = MyRadius * Cos[MyAngle] + Sin[ArcCos[MyRadius]] * Sin[MyAngle];
y2 = MyRadius * Sin[MyAngle] - Sin[ArcCos[MyRadius]] * Cos[MyAngle];
r = Sqrt[(x2 - x1) ^ 2 + (y2 - y1) ^ 2];
If[r > Sqrt[3],
GTCount++;
AppendTo[MyChords1, Line[{{x1, y1}, {x2, y2}}]],
AppendTo[MyChords2, Line[{{x1, y1}, {x2, y2}}]]
];
](* end for loop over samples *); Print[N[GTCount / NumSamples]];
```

```
0.499
```

In[36]:=
```
Show[Graphics[MyPoints], Graphics[Circle[{0, 0}, 1]],
 Graphics[{Blue, MyChords1}], Graphics[{Red, MyChords2}]]
```

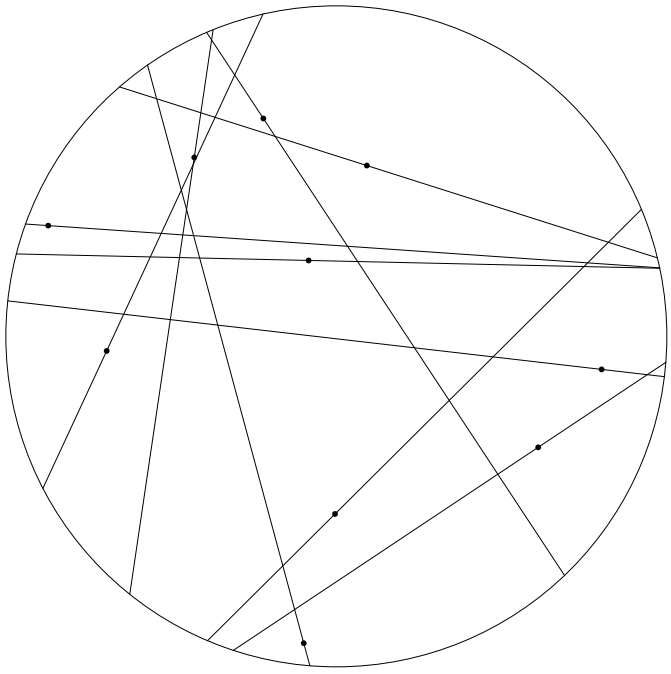
Out[36]=



VERSION 4: Pick a random point in the circle. Pick a random angle (slope). Create a chord that goes through that point having that slope. Unlike Version 2, the point is not the midpoint of the chord.

In[37]:=
```
NumSamples = 10;
MyPoints = {};
MyChords = {};
For[i = 1, i ≤ NumSamples, i++,
MyAngle = RandomVariate[UniformDistribution[{0, 2 Pi}]];
MyRadius = Sqrt[RandomVariate[UniformDistribution[{0, 1}]]]; (* Note the sqrt*)
MyAngle2 = RandomVariate[UniformDistribution[{0, 2 Pi}]];
root1 = -MyRadius * Cos[MyAngle2] + Sqrt[1 + MyRadius ^ 2 * ((Cos[MyAngle2]) ^ 2 - 1)];
root2 = -MyRadius * Cos[MyAngle2] - Sqrt[1 + MyRadius ^ 2 * ((Cos[MyAngle2]) ^ 2 - 1)];
AppendTo[MyPoints, Point[{MyRadius * Cos[MyAngle], MyRadius * Sin[MyAngle]}]];
x1 = (MyRadius + root1 * Cos[MyAngle2]) * Cos[MyAngle] - root1 * Sin[MyAngle2] * Sin[MyAngle];
y1 = (MyRadius + root1 * Cos[MyAngle2]) * Sin[MyAngle] + root1 * Sin[MyAngle2] * Cos[MyAngle];
x2 = (MyRadius + root2 * Cos[MyAngle2]) * Cos[MyAngle] - root2 * Sin[MyAngle2] * Sin[MyAngle];
y2 = (MyRadius + root2 * Cos[MyAngle2]) * Sin[MyAngle] + root2 * Sin[MyAngle2] * Cos[MyAngle];
AppendTo[MyChords, Line[{{x1, y1}, {x2, y2}}]]
](* end for loop over samples *);
Show[Graphics[MyPoints], Graphics[MyChords], Graphics[Circle[{0, 0}, 1]]]
```

Out[41]=



In[42]:=
```
NumSamples = 1000;
GTCount = 0;
MyPoints = {};
MyChords1 = {};
MyChords2 = {};
For[i = 1, i ≤ NumSamples, i++,
MyAngle = RandomVariate[UniformDistribution[{0, 2 Pi}]];
MyRadius = Sqrt[RandomVariate[UniformDistribution[{0, 1}]]]; (* Note the sqrt*)
MyAngle2 = RandomVariate[UniformDistribution[{0, 2 Pi}]];
root1 = -MyRadius * Cos[MyAngle2] + Sqrt[1 + MyRadius ^ 2 * ((Cos[MyAngle2]) ^ 2 - 1)];
root2 = -MyRadius * Cos[MyAngle2] - Sqrt[1 + MyRadius ^ 2 * ((Cos[MyAngle2]) ^ 2 - 1)];
AppendTo[MyPoints, Point[{MyRadius * Cos[MyAngle], MyRadius * Sin[MyAngle]}]];
x1 = (MyRadius + root1 * Cos[MyAngle2]) * Cos[MyAngle] - root1 * Sin[MyAngle2] * Sin[MyAngle];
y1 = (MyRadius + root1 * Cos[MyAngle2]) * Sin[MyAngle] + root1 * Sin[MyAngle2] * Cos[MyAngle];
x2 = (MyRadius + root2 * Cos[MyAngle2]) * Cos[MyAngle] - root2 * Sin[MyAngle2] * Sin[MyAngle];
y2 = (MyRadius + root2 * Cos[MyAngle2]) * Sin[MyAngle] + root2 * Sin[MyAngle2] * Cos[MyAngle];
r = Sqrt[(x2 - x1) ^ 2 + (y2 - y1) ^ 2];
If[r > Sqrt[3],
GTCount++;
AppendTo[MyChords1, Line[{{x1, y1}, {x2, y2}}]],
AppendTo[MyChords2, Line[{{x1, y1}, {x2, y2}}]]
];
]; Print[N[GTCount / NumSamples]];
```

```
0.59
```

In[48]:=
```
Show[Graphics[MyPoints], Graphics[Circle[{0, 0}, 1]],
 Graphics[{Blue, MyChords1}], Graphics[{Red, MyChords2}]]
```

Out[48]=