Math 475 Final Project

Ruidong Chen, Student ID: 488314

Xinyang Feng, Student ID: 490138

Ziwei Liu, Student ID: 489909

Zhuoyun Chen, Student ID: 499214

1. Introduction and Motivation

1.1. Sampling

Sampling is a statistical method that involves selecting a subset of individuals from a larger population and using that subset to represent the entire population. This is typically done to reduce the amount of data that needs to be collected and analyzed, as collecting and analyzing data from the entire population can be time-consuming, expensive, and impractical. Sampling is a crucial component of statistical analysis, as it allows researchers to make inferences about the population based on the characteristics of the sample.

1.2. Sampling from Sphere

Sphere sampling is a method of randomly sampling points from the surface of a sphere. It is often used in computer graphics and numerical simulations to generate random points on the surface of a sphere for various purposes, such as lighting, shading, and particle effects.

Sphere sampling method, which involves selecting observations from the population at random, but with a probability that is proportional to their distance from a specified point, known as the center of the sphere. This method is used to ensure that the observations in the sample are representative of the population, rather than being biased towards observations that are close to the center of the sphere.

There are several different techniques for generating random points on a sphere, including:

- **Monte Carlo method**: This method involves randomly selecting points within a bounding box that encloses the sphere, and then rejecting any points that fall outside the sphere. This method is relatively simple, but it can be inefficient if the bounding box is much larger than the sphere.
- **Stratified sampling**: This method involves dividing the surface of the sphere into a grid of equally sized cells, and then randomly selecting one point from each cell. This method can be more efficient than the Monte Carlo method, but it may not produce as uniformly distributed a set of points.
- **Importance sampling**: This method involves generating points on the sphere according to a probability density function (PDF) that reflects the relative importance of different areas of the sphere. This method can be more efficient than other methods if the PDF is

carefully chosen to match the distribution of points needed for a particular application.

1.3. Advantages and Practical Applications

Sampling points from the surface of a sphere is useful in many applications because it allows you to generate random points that are evenly distributed over the surface of the sphere. This can be useful in a variety of contexts, including:

- **Computer graphics**: In computer graphics, random points on the surface of a sphere can be used to simulate various effects, such as lighting, shading, and particle systems. Sampling points from the surface of a sphere allows you to generate these effects in a way that is evenly distributed over the surface of the sphere, which can produce more realistic and aesthetically pleasing results.
- **Numerical simulations**: In numerical simulations, random points on the surface of a sphere can be used to represent various quantities, such as positions, velocities, or forces. Sampling points from the surface of a sphere allows you to represent these quantities in a way that is evenly distributed over the surface of the sphere, which can improve the accuracy and reliability of the simulation.
- **Statistics and machine learning**: In statistics and machine learning, random points on the surface of a sphere can be used to generate samples for various purposes, such as testing hypotheses or training models. Sampling points from the surface of a sphere allows you to generate these samples in a way that is evenly distributed over the surface of the sphere, which can improve the reliability and generalizability of the results.

1.4. Make inference about the humidity or temperature of the earth

When considering the earth, we can regard this as a sphere. Hence, we can sample humidity or temperature from the history records. We can divide this sphere into plenty of parts, and into different areas. For example, we can divide the earth into ocean part and land part. Then we can apply some sampling methods to these points based on the condition. To be more detailed, we can consider the ocean currents and altitude.

2. Box Muller's Approach and the Problem

2.1.Definition

The Box-Muller approach is a method for generating random points in two dimensions that are uniformly distributed over the unit circle. It is often used to generate random points on the surface of a sphere in three dimensions.

The Box-Muller approach works by generating two independents, uniformly distributed random variables x and y, and then using these values to define a point (x, y) on the unit circle. The resulting point will be uniformly distributed over the surface of the unit circle.

$$x = \sqrt{(-2) * \log(x)} * \cos(2\pi y)$$
$$y = \sqrt{(-2) * \log(x)} * \sin(2\pi y)$$

To generate random points on the surface of a sphere in three dimensions, the Box-Muller approach can be extended to generate three independents, uniformly distributed random variables x, y, and z, and then use these values to define a point (x, y, z) on the unit sphere. The resulting point will be uniformly distributed over the surface of the unit sphere.

$$x = \sqrt{(-2) * \log(x)} * \cos(2\pi y)$$
$$y = \sqrt{(-2) * \log(x)} * \sin(2\pi y)$$
$$z = \sqrt{(-2) * \log(z)}$$

2.2.Advantages

One of the main advantages of the Box-Muller approach is that it is relatively simple to implement and can be easily extended to higher dimensions. The formulas for generating points on the unit circle and the unit sphere are straightforward and easy to remember, which makes the Box-Muller approach a convenient choice for many applications.

Moreover, the Box-Muller approach generates points that are uniformly distributed over the surface of the unit circle or unit sphere. This can be useful in many applications where you need to generate random points that are evenly distributed over the surface of a sphere, such as in computer graphics or numerical simulations.

In addition, the Box-Muller approach is based on well-known and widely used mathematical functions, such as the logarithm, square root, cosine, and sine functions, which makes it a widely accepted and understood method.

2.3.Disadvantages

For disadvantages of the Box-Muller approach, it can be less efficient than other methods for generating random points on the surface of a sphere. In particular, the Box-Muller approach can be slower than other methods because it requires the computation of the logarithm and square root functions, which can be computationally expensive.

Also, Box-Muller approach requires the generation of two or three independent, uniformly distributed random variables, which can also be computationally expensive. This can be particularly problematic if the number of random points that you need to generate is very large.

In addition, the Box-Muller approach may not always produce points that are as uniformly

distributed over the surface of the sphere as other methods, such as stratified sampling or importance sampling. This can be a problem if you need to generate many points that are evenly distributed over the surface of the sphere.

Overall, the Box-Muller approach is a simple and effective method for generating random points on the surface of a sphere, but it may not always be the most efficient or accurate method depending on the specific needs of the application. Therefore, we need to consider more methods to deal with sphere sampling.

3. The Method of Sampling from the Surface of a Sphere

3.1. Definition

3.1.1. D-spere and D-ball

A unit d-dimensional sphere is defined as follows:

$S^d = \{x \in R^{d+1} : |x| = 1\}$

And a unit d-dimensional ball is defined as follows:

$B^d = \{x \in R^d : |x| \leq 1\}$

Based on this definition, the perimeter of a circle is 1-sphere, and the interior of the circle is 2-ball. The surface of a ball is 2-sphere, and the interior of the ball is 3-ball. Also, the d signifies the degrees of freedom.

3.1.2. Uniform Random Sampling

For uniform random sampling, this means that all possible elements of S have an equal probability to be selected based on discreate probability. Besides, it also means the likelihood of an element falling in any subinterval is discrete proportional to the length of the subinterval based on continuous probabilities. Also, a simple and common method to draw from a normal distribution is the Box-Muller algorithm.

To briefly understand why this works, consider the function:

$$f(u,v) = f(z) = (\frac{1}{\sqrt{2\pi}} e^{-0.5u^2})(\frac{1}{\sqrt{2\pi}} e^{-0.5v^2})$$

After algebra simplifications:

$$f(u,v) = f(z) = \frac{1}{2\pi} e^{-0.5(u^2+v^2)} = \frac{1}{2\pi} e^{-0.5|z|^2}$$

This shows that the probability distribution of z only depends on its magnitude and not any other direction.

Thus, it must be symmetric and must correspond to a uniform distribution on the circle.
This argument can be generalized to any d.

3.2. Sphere Sampling Methods

3.2.1. Rejection Method

The rejection method is a technique for generating random points on the surface of a sphere that involves randomly sampling points within a bounding box that encloses the sphere, and then rejecting any points that fall outside the sphere. This method is often referred to as the "Monte Carlo method," as it is based on the principles of Monte Carlo simulation.

To implement the rejection method for sampling points from the surface of a sphere, here is the process:

a) Define a bounding box that encloses the sphere. This can be any rectangular region that surrounds the sphere, but it should be as small as possible to minimize the number of rejected points.
b) Generate a random point (x, y, z) within the bounding box using a method for generating uniformly distributed random numbers.
c) Check whether the point (x, y, z) falls within the sphere. You can do this by computing the distance from the point to the center of the sphere and comparing it to the radius of the sphere. If the distance is less than or equal to the radius, the point falls within the sphere; otherwise, it falls outside the sphere.
d) If the point falls within the sphere, keep it and add it to your sample. If the point falls outside the sphere, reject it and generate a new point. Repeat this process until you have generated the desired number of points.

3.2.2. Polar + Radial CDF

The Polar + Radial CDF method is a technique for generating random points on the surface of a sphere that involves sampling points using a probability density function (PDF) that reflects the relative importance of different areas of the sphere. This method is known as an "importance sampling" method, as it involves generating points according to their importance or relevance to the application.

To implement the Polar + Radial CDF method for sampling points from the surface of a sphere, here is the process:

a) Define a PDF that reflects the relative importance of different areas of the sphere. This can be any function that assigns a higher probability to areas of the sphere that are more important or relevant to the application.
b) Generate two independents, uniformly distributed random variables u and v.
c) Use the PDF and the random variables u and v to compute the polar angle θ and the radial distance r, where $\theta = 2 * \pi * u;\ r = \sqrt{v}$

d) Finally, use the polar angle θ and the radial distance r to compute the Cartesian coordinates (x, y, z) of the point on the surface of the sphere.

$x = r * \cos(\theta)$

$y = r * \sin(\theta)$

$z = \sqrt{1 - r^2}$

3.2.3. Concentric Map

Concentric map sampling is a technique for generating random points on the surface of a sphere that involves dividing the surface of the sphere into concentric rings or shells, and then sampling points uniformly within each ring or shell. This method is often referred to as "stratified sampling," as it involves dividing the surface of the sphere into strata or layers.

To implement the concentric map method for sampling points from the surface of a sphere, here is the process:

a) Divide the surface of the sphere into concentric rings or shells using a predetermined number of rings or shells and a spacing between them.
b) Generate two independents, uniformly distributed random variables u and v.
c) Use the random variables u and v to determine which ring or shell the point should be sampled from.
d) Within the chosen ring or shell, generate a random point (x, y, z) uniformly over the surface of the ring or shell using a method such as the Polar + Radial CDF method we mentioned previously.

3.2.4. Exponential Distribution

Exponential distribution is another method we can use to generate random points within a sphere. This method involves generating random points in a cube, and then rejecting any points that fall outside of the sphere.

Exponential distribution method is valid for balls of any dimension is to draw d random variates from the normal distribution and a single random variable from exponential distribution usually with $\lambda = 0.5$. Also, a simple and common method to draw from an exponential distribution with parameter lambda is via the negative logarithm of uniform random variate divided by λ.

3.2.5. Dropped Coordinates

Dropped Coordinates is a new method first introduced by Harman and Lacko in 2010 and then proven by Voelker in 2017. It involves generating three random numbers between -1 and 1, and using them as the coordinates (x, y, z) of a point within the unit sphere (i.e., a sphere with radius 1). The point is then scaled by the desired radius of the sphere to obtain a point within the target

sphere.

4. Simulation and data analysis and data visualization.

We will put all data analysis and data visualization results along with some problems extension discuss and implement in the next page.

1:Rejection Method

This is probably the most intuitive method and is fast for 2-ball.

However, it will get a bad rap because when we apply this method to high dimensions, it can become very inefficient.

pseudo code:

1: Select x,y ~ U(-1,1)

2: If ($x^{2+y}2>1$) then reject and go to step 1.

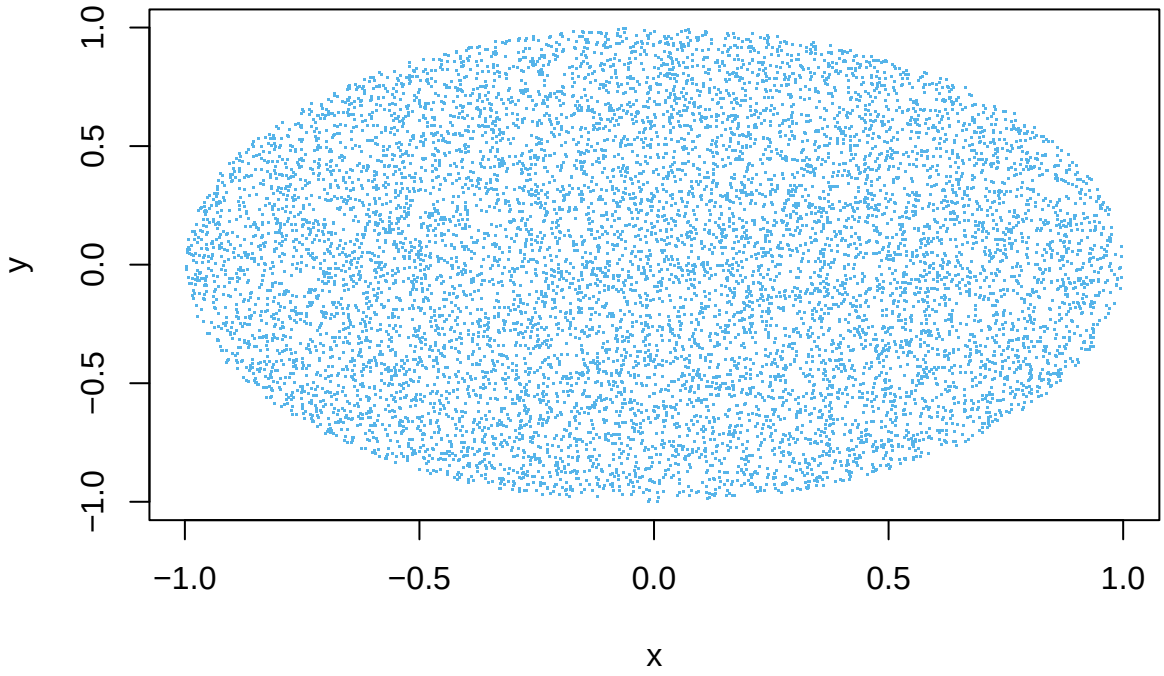
3: Return (x,y)

2:Polar + Radial CDF

For balls of any dimension d, a very common method of uniformly picking form a d-ball is to first select a random directional unit-vector from (d-1)-sphere and then multiply this vector by a scaler radial multiplicative factor.

The crucial aspect of this algorithm is the multiplicative factor square root function. This is because area which corresponds to the CDF grows according to r square. Therefore, we need to apply the inverse of this.

```
set.seed(1)
u=runif(10000)
v=runif(10000)
r=u^0.5
theta=2*pi*v
x=r*cos(theta)
y=r*sin(theta)
colors <- c('#56B4E9')
plot(x,y,col=colors,pch = '.')
```



3:Concentric Map

This variance reduction method is motivated by the idea that ideal mappings from square-based random variate uv-space to the coordinate xy-space should be area-preserving, two-way continuous and with low distortion.

```
set.seed(1)
u=runif(10000)
v=runif(10000)
if (u==0 && v==0) return(0,0)
```

```
## Warning in u == 0 && v == 0: 'length(x) = 10000 > 1' in coercion to 'logical(1)'
```

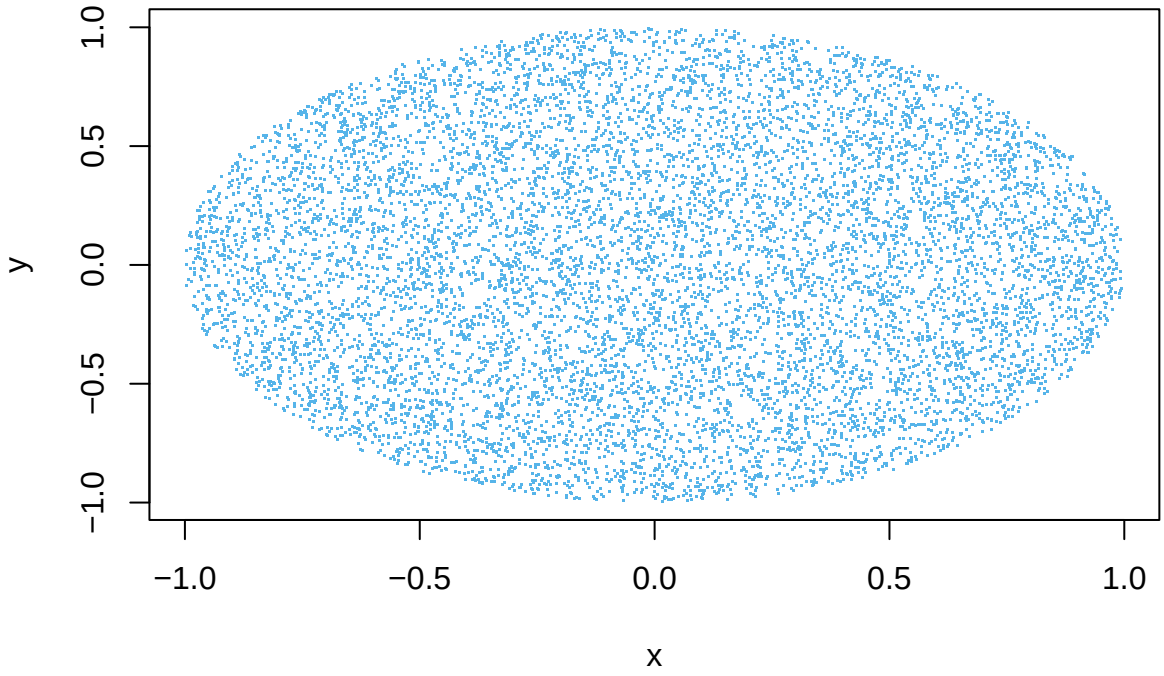
```
theta=0
r=1
a=2*u-1
b=2*v-1
r=numeric(10000)
theta=numeric(10000)

for (i in 1:10000){
  if (a[i]*a[i]>b[i]*b[i]){
    r[i]=a[i]
    theta[i]=pi/4*b[i]/a[i]
  }else{
    r[i]=b[i]
```

```
    theta[i]=pi/2-pi/4*a[i]/b[i]
  }
}

x=r*cos(theta)
y=r*sin(theta)

colors <- c('#56B4E9')
plot(x,y,col=colors,pch = '.')
```

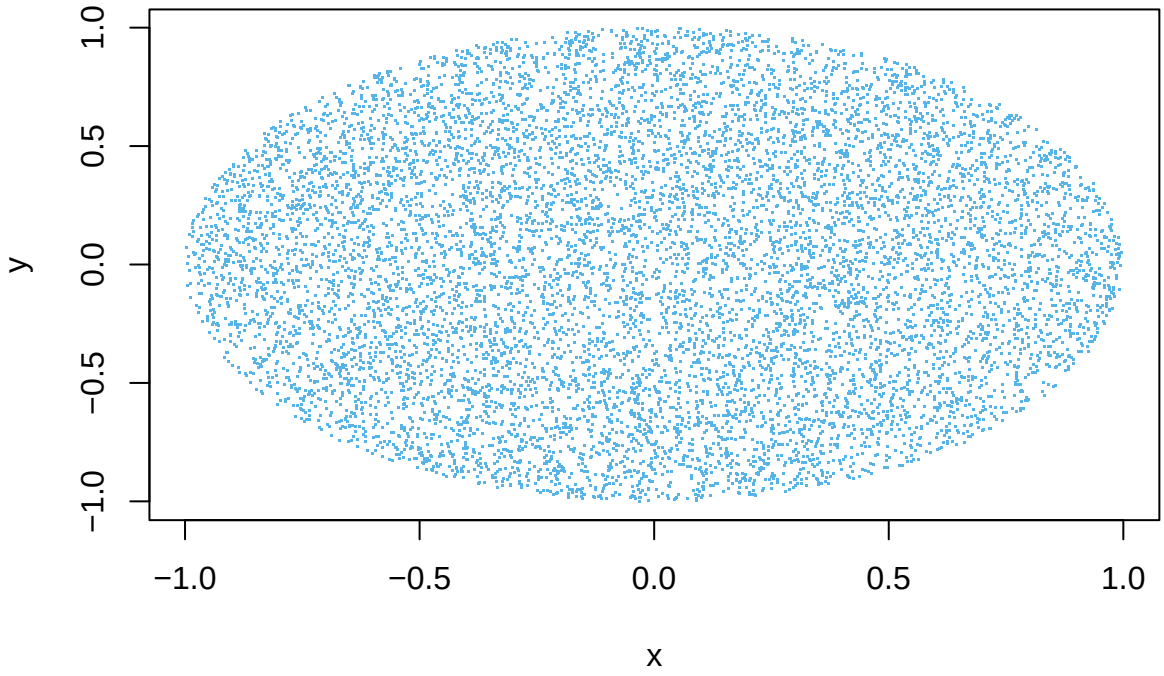


4:Exponential Distribution

Another method that is valid for balls of any dimension is to draw d random variates from the normal distribution and a single random variable from exponential distribution usually with lambda=0.5.

Also, a simple and common method to draw from an exponential distribution with parameter lambda is via the the negative logarithm of uniform random variate divided by lambda.

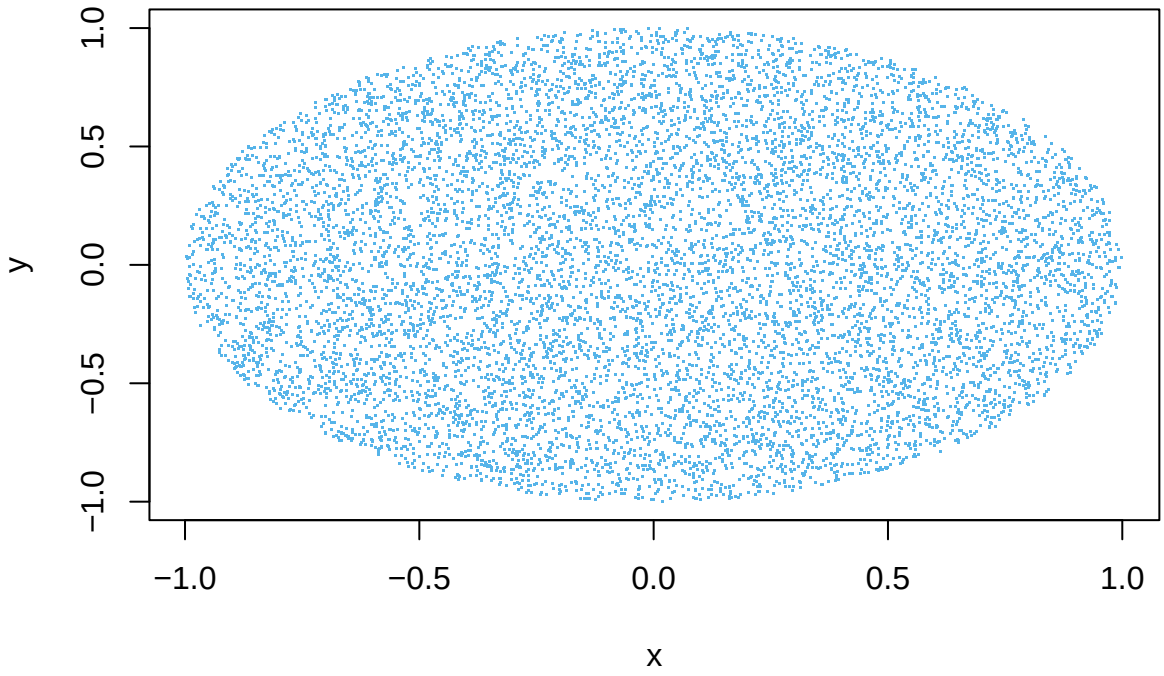
```
u=rnorm(10000)
v=rnorm(10000)
e=rexp(10000,0.5)
denom=(u*u+v*v+e)^0.5
x=u/denom
y=v/denom
colors <- c('#56B4E9')
plot(x,y,col=colors,pch = '.')
```



5:Dropped Coordinates

A new method first introduced by Harman and Lacko in 2010 and then proven by Voelker in 2017 that is not famous is that if (x1,x2,x3,x4) is a random vector uniformly distributed on the 3-sphere, the random vector (x1,x2) is uniformly distributed in the 2-ball.

```
a=rnorm(10000)
b=rnorm(10000)
c=rnorm(10000)
d=rnorm(10000)
norm=(a*a+b*b+c*c+d*d)^0.5
x=c/norm
y=d/norm
colors <- c('#56B4E9')
plot(x,y,col=colors,pch = '.')
```



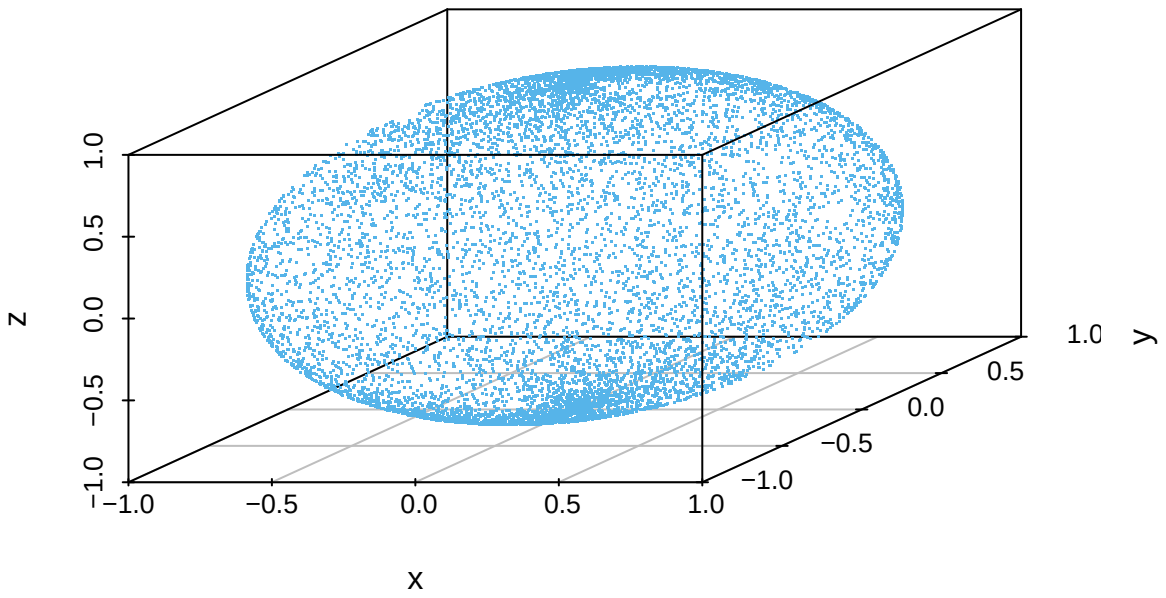
Uniformly sampling a 2-sphere which is the surface of a sphere

We can uniformly sample points on the surface of a sphere.

polar

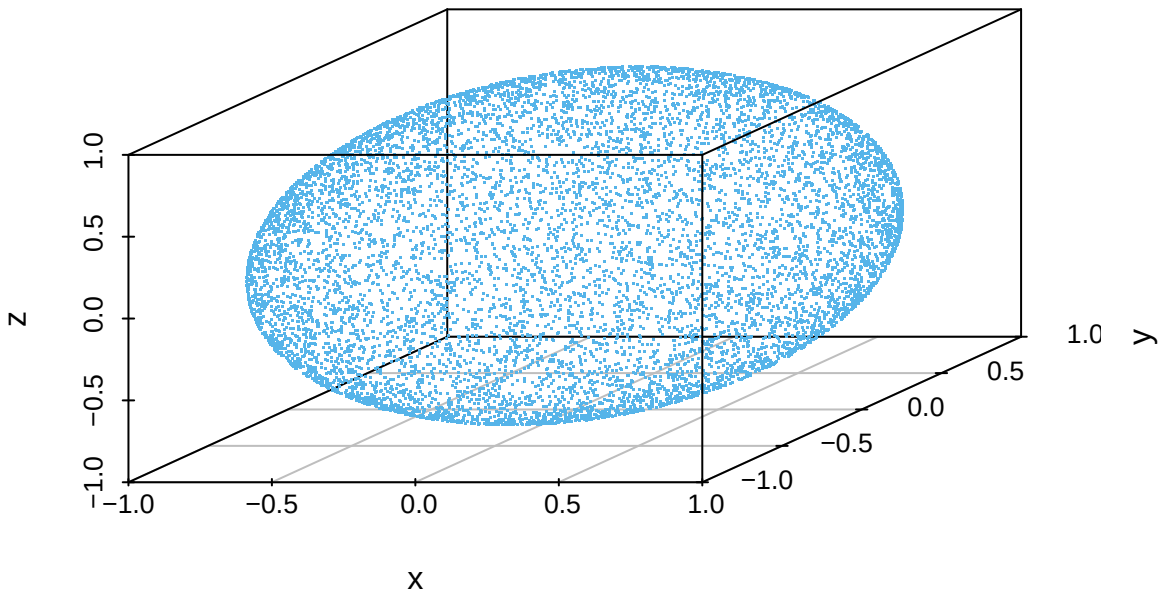
This is the natural generalization.

```
library(scatterplot3d)
u=runif(10000)
v=runif(10000)
theta=2*pi*u
phi=acos(2*v-1)
x=sin(theta)*cos(phi)
y=sin(theta)*sin(phi)
z=cos(theta)
colors <- c('#56B4E9')
scatterplot3d(x,y,z,color=colors,pch = '.')
```



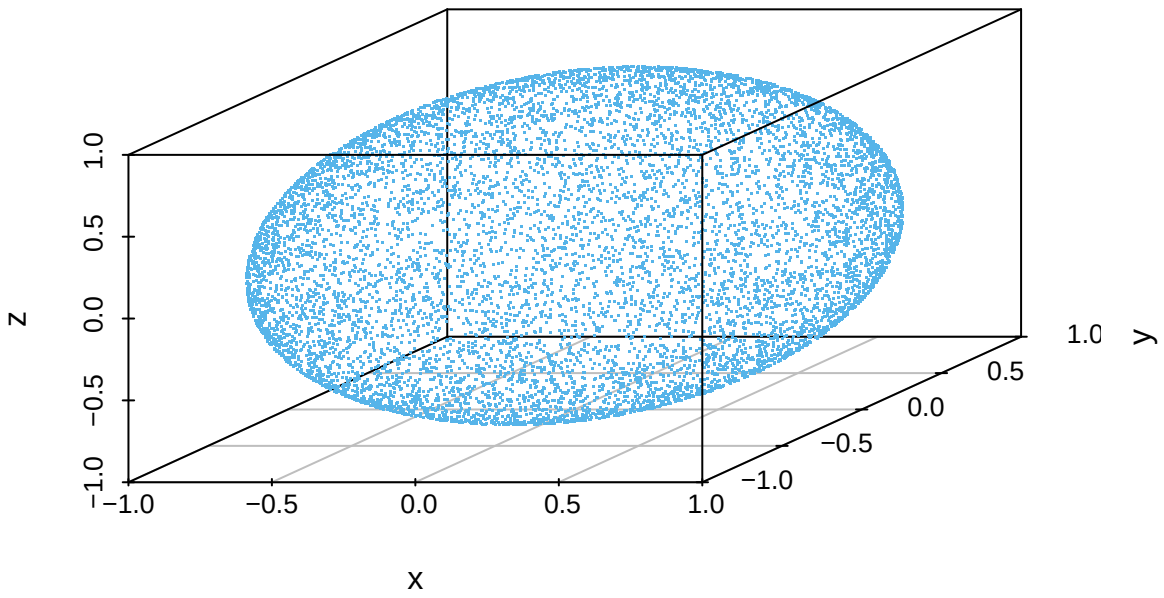
We can also use u=cosine theta which does not use inverse trig functions as follows:

```
library(scatterplot3d)
u=2*runif(10000)-1
phi=2*pi*runif(10000)
z=u
x=cos(phi)*(1-z^2)^0.5
y=sin(phi)*(1-z^2)^0.5
colors <- c('#56B4E9')
scatterplot3d(x,y,z,color=colors,pch = '.')
```



Muller method

```
library(scatterplot3d)
a=rnorm(10000)
b=rnorm(10000)
c=rnorm(10000)
norm=(a*a+b*b+c*c)^0.5
x=a/norm
y=b/norm
z=c/norm
colors <- c('#56B4E9')
scatterplot3d(x,y,z,color=colors,pch = '.')
```

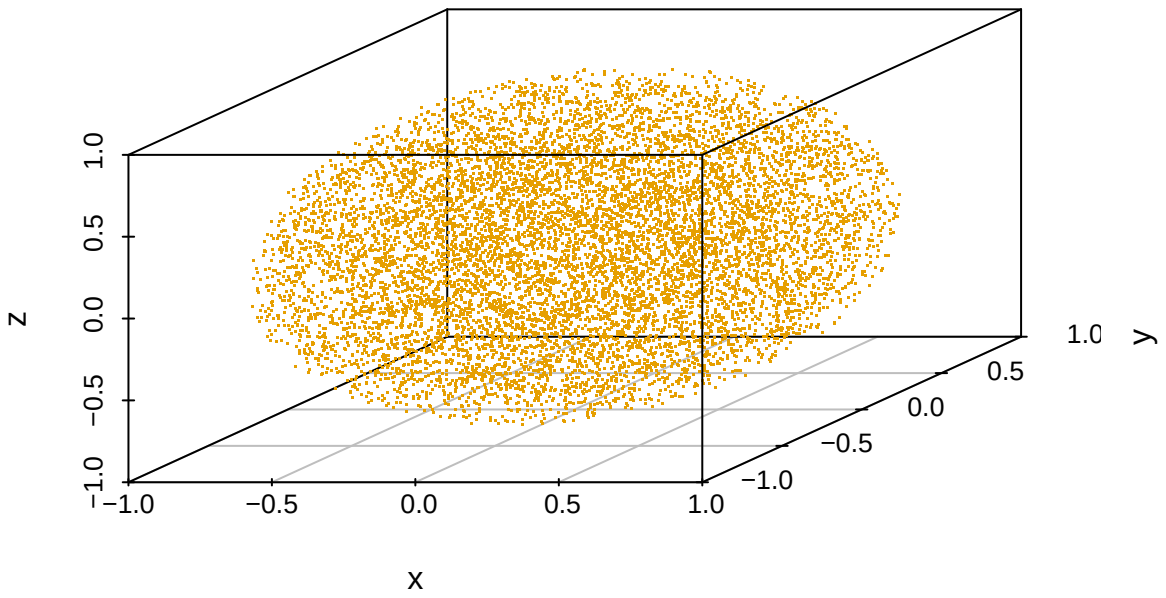


Uniformly sampling a 3-ball which is interior of a sphere

We will now uniformly sample points inside a sphere.

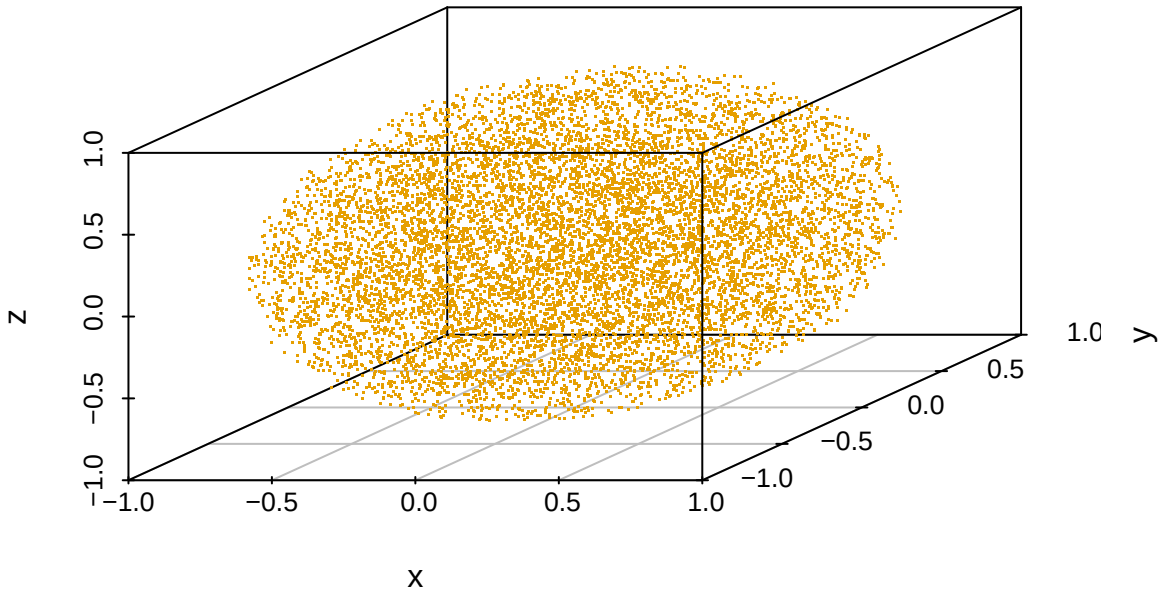
Muller method for 3-ball

```
library(scatterplot3d)
a=rnorm(10000)
b=rnorm(10000)
c=rnorm(10000)
r=runif(10000)^(1/3)
norm=(a*a+b*b+c*c)^0.5
x=r*a/norm
y=r*b/norm
z=r*c/norm
colors <- c("#E69F00")
scatterplot3d(x,y,z,color=colors,pch = '.')
```



polar method for 3-ball

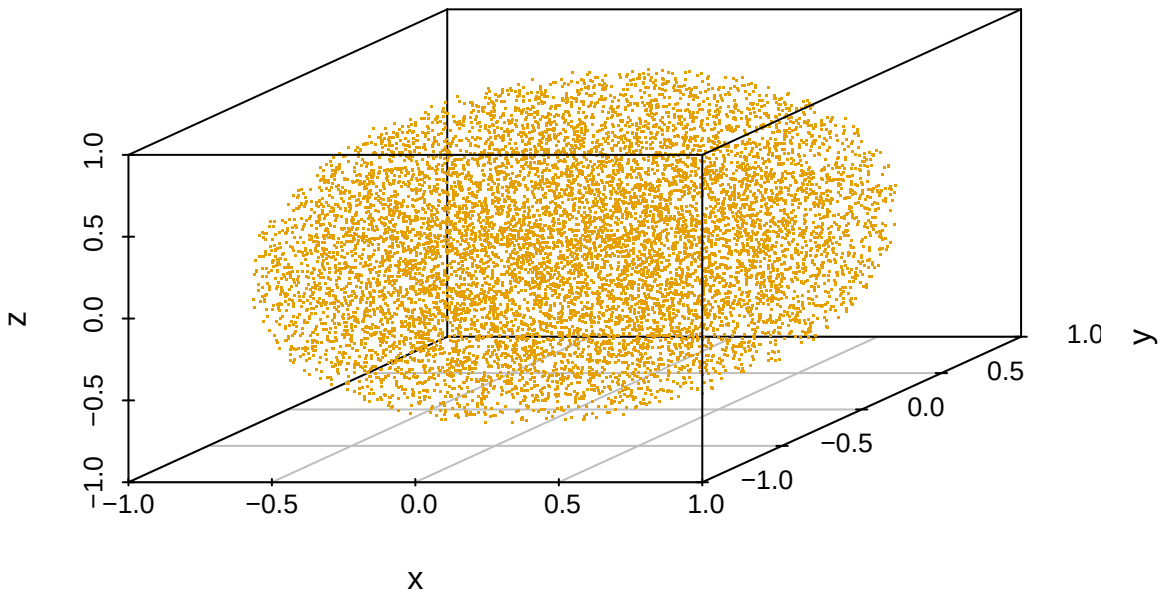
```
library(scatterplot3d)
u=2*runif(10000)-1
phi=2*pi*runif(10000)
r=runif(10000)^(1/3)
x=r*cos(phi)*(1-u^2)^0.5
y=r*sin(phi)*(1-u^2)^0.5
z=r*u
colors <- c("#E69F00")
scatterplot3d(x,y,z,color=colors,pch = '.')
```



Exponential Distribution for 3-ball

This is a similar method as 2-sphere.

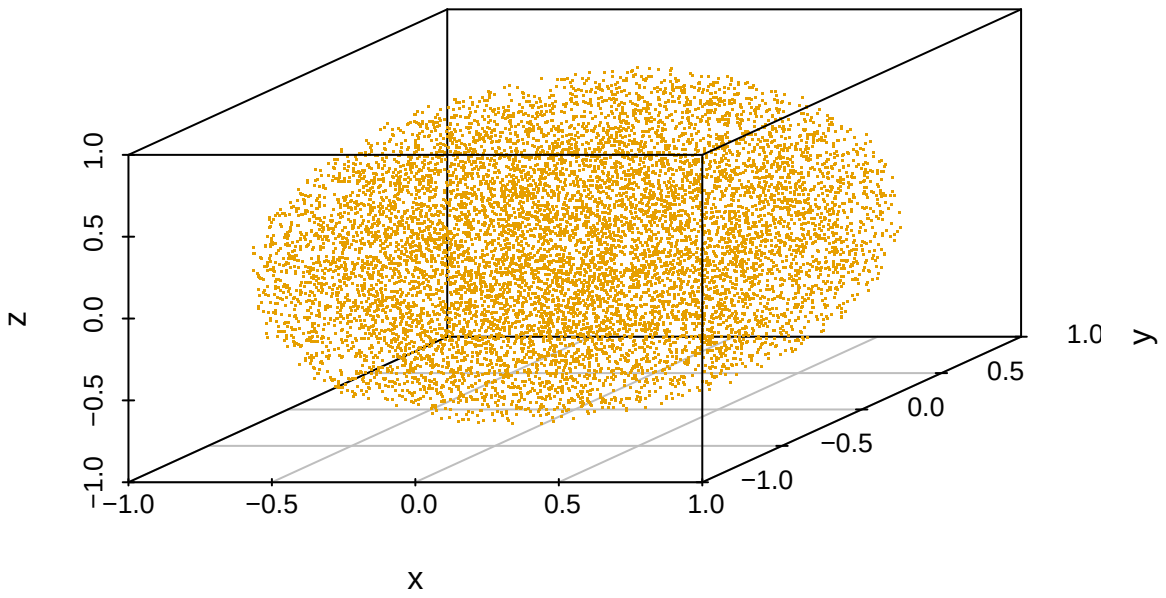
```
library(scatterplot3d)
a=rnorm(10000)
b=rnorm(10000)
c=rnorm(10000)
e=rexp(10000,rate=0.5)
denom=(e+a*a+b*b+c*c)^0.5
x=a/denom
y=b/denom
z=c/denom
colors <- c("#E69F00")
scatterplot3d(x,y,z,color=colors,pch = '.')
```



Dropped coordinates for 3-ball

Also, we have random vectors (a,b,c,d,e) uniformly distributed on the 4-sphere, the random vector (c,d,e) is uniformly distributed in the 3-ball and we dropped the first two coordinates.

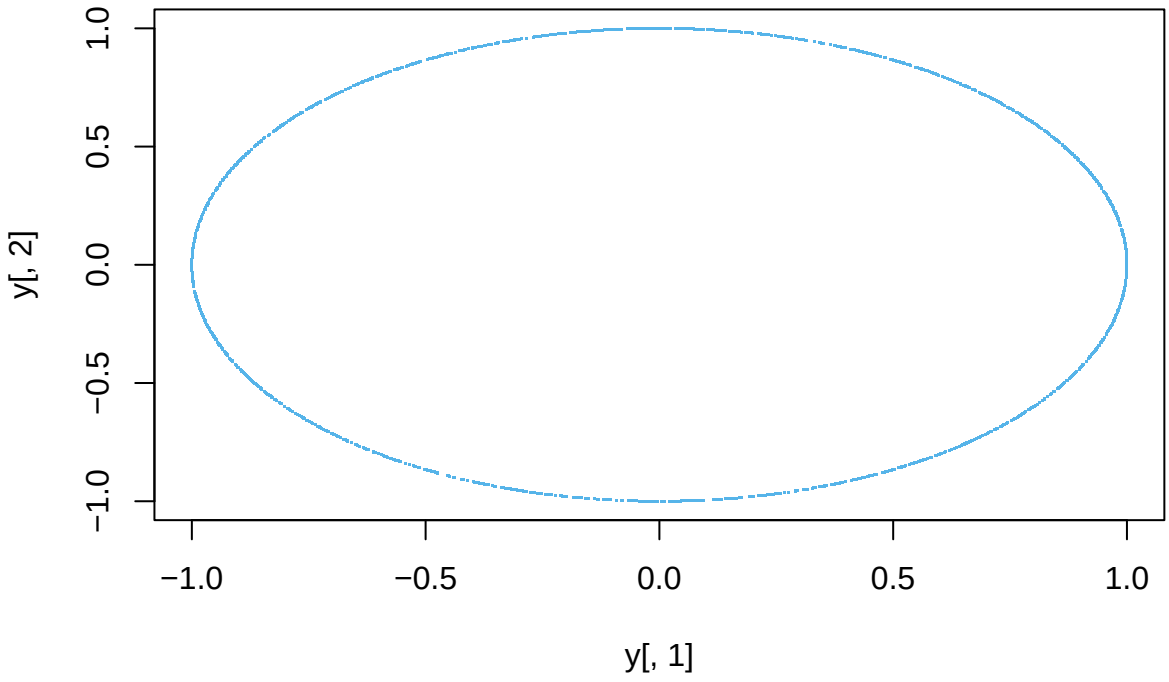
```
library(scatterplot3d)
a=rnorm(10000)
b=rnorm(10000)
c=rnorm(10000)
d=rnorm(10000)
e=rnorm(10000)
norm=(a*a+b*b+c*c+d*d+e*e)^0.5
x=c/norm
y=d/norm
z=e/norm
colors <- c("#E69F00")
scatterplot3d(x,y,z,color=colors,pch = '.')
```



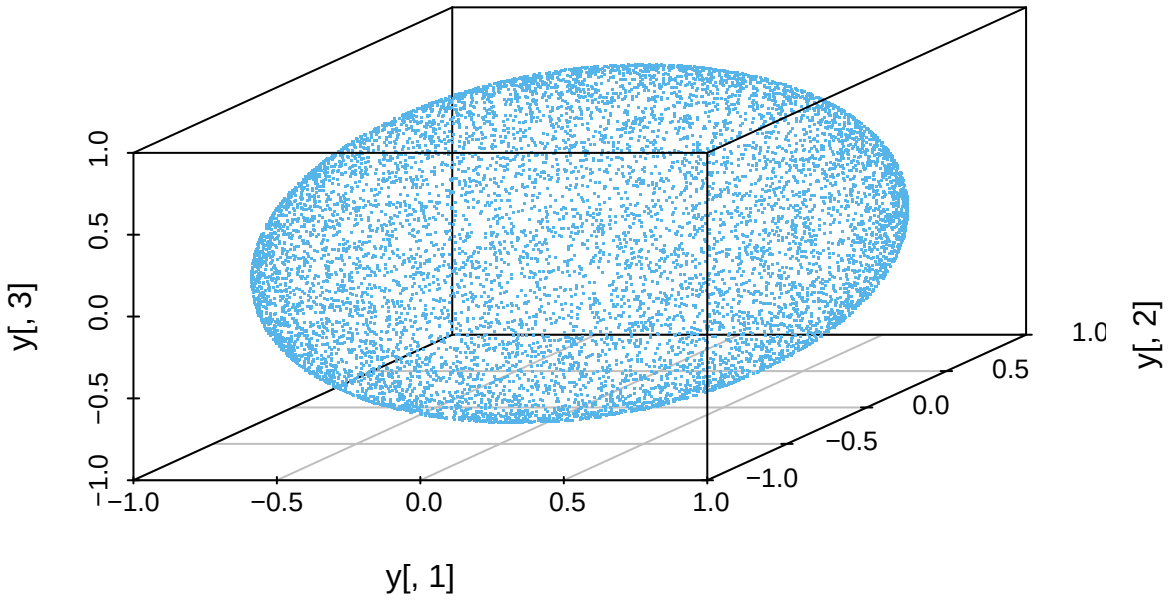
Other method

```
#install.packages("scatterplot3d")
```

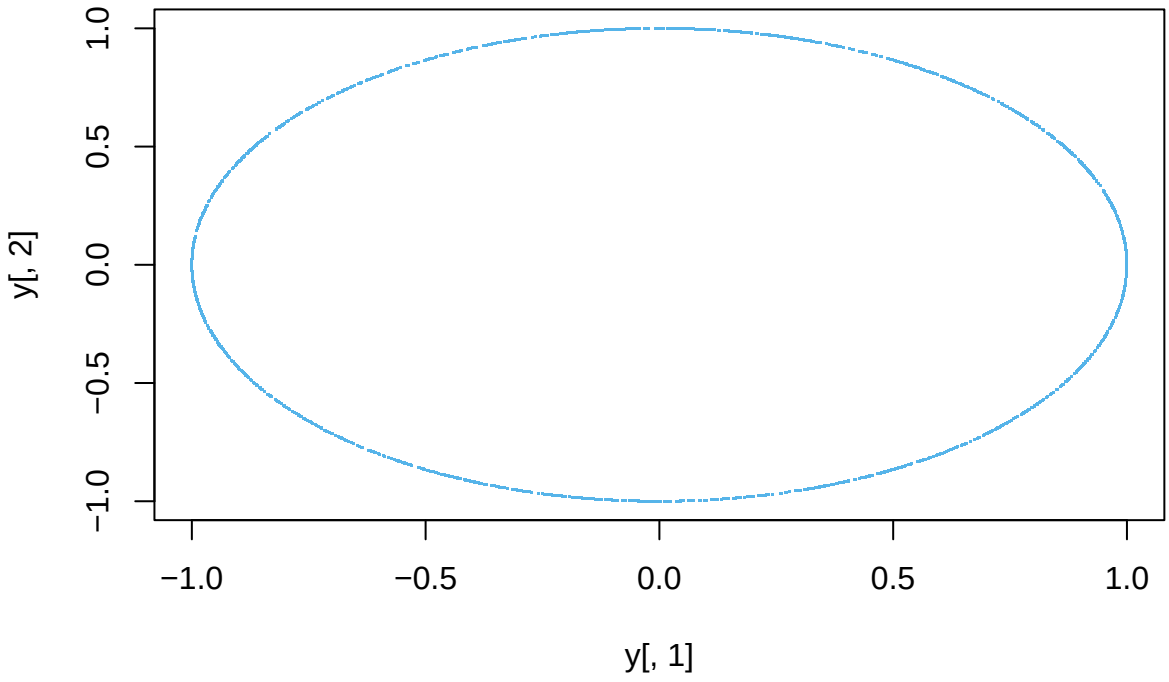
```
#r2 space
#multivariate approach
library(scatterplot3d)
n=2000
k=2
x=matrix(rnorm(n*k),ncol=k)
y=x/sqrt(rowSums(x*x))
colors <- c('#56B4E9')
plot(y[,1],y[,2],col=colors,pch = '.')
```




```
# r3 space
#multivariate approach
n=10000
k=3
x=matrix(rnorm(n*k),ncol = k)
y=x/sqrt(rowSums(x*x))
colors <- c('#56B4E9')
scatterplot3d(y[,1],y[,2],y[,3],color=colors,pch = '.')
```



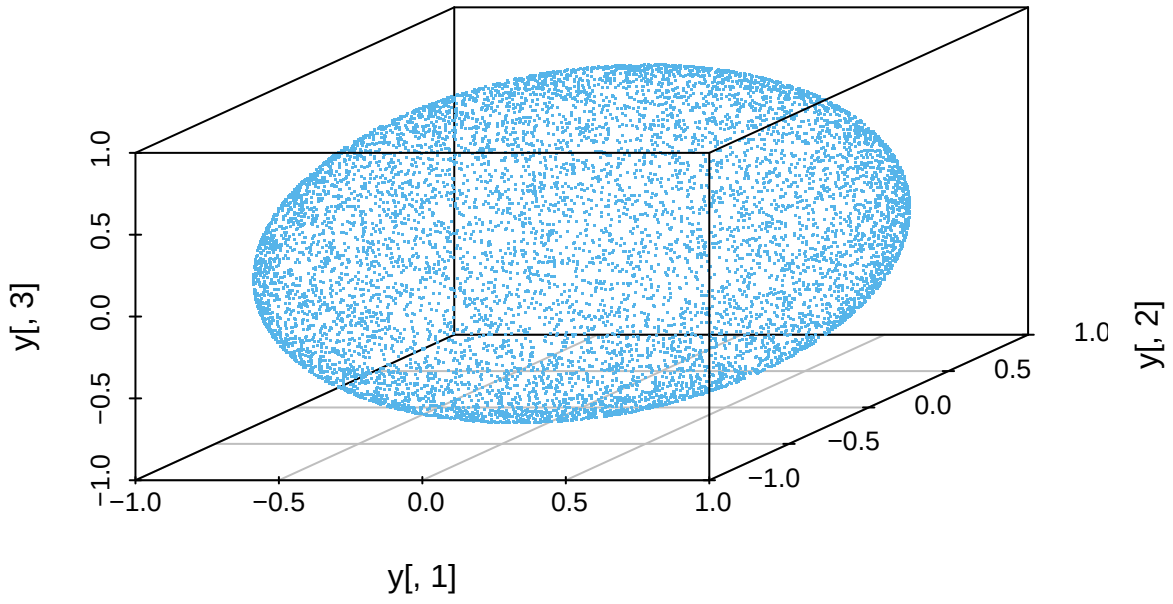

```r
#r2 space
#random numbers in a square
n=2000
k=2
x=matrix(runif(k*ceiling(n*4/pi),-1,1),ncol = k)
x=x[x[,1]^2+x[,2]^2<=1,]
y=x/sqrt(rowSums(x*x))
colors <- c('#56B4E9')
plot(y[,1],y[,2],col=colors,pch = '.')
```




```
dim(y)
```

```
## [1] 1987    2
```

```
#r3 space
#random numbers in a cube
n=10000
k=3
x=matrix(runif(k*ceiling(n*6/pi),-1,1),ncol = k)
x=x[x[,1]^2+x[,2]^2+x[,3]^2<=1,]
y=x/sqrt(rowSums(x*x))
colors <- c('#56B4E9')
scatterplot3d(y[,1],y[,2],y[,3],color=colors,pch = '.')
```




```
dim(y)
```

```
## [1] 9880    3
```

Generalization: How to generalize to n-dimensional sphere or ball sampling? Using Muller Method.We can do it by random numbers in a cube method, but its efficiency drops quickly after d=3.

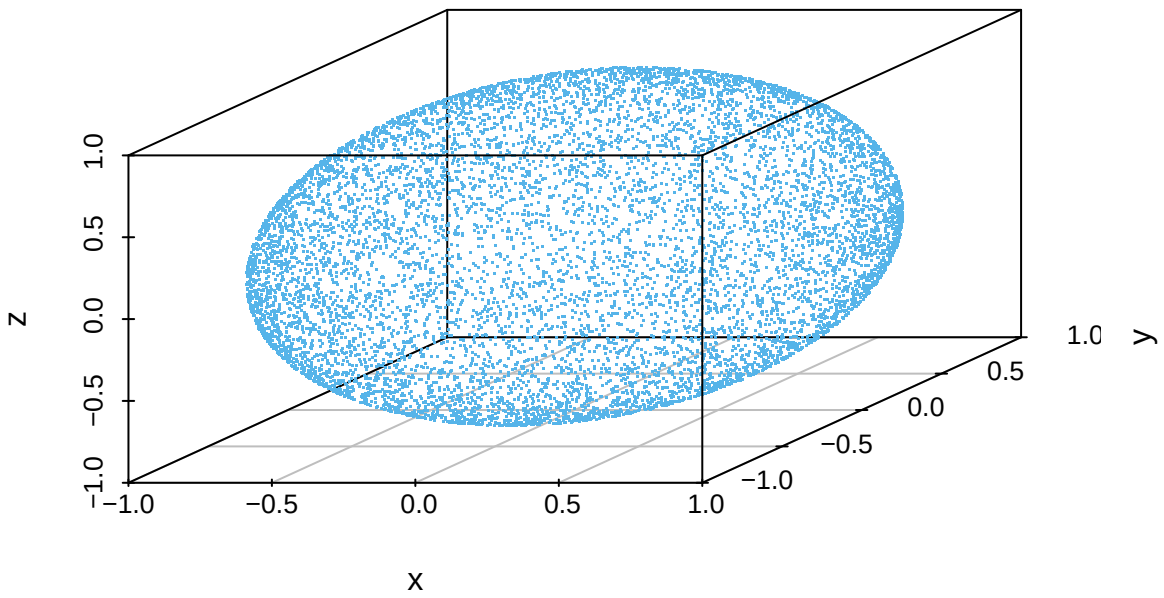
Start from sphere sampling.

Muller Method

```
d_dimension_sphere = function(D) {
    samples = sapply(1:D, function(d) {
        rnorm(n=10000)
    })
    radii = apply(samples, 1, function(s) {
        sqrt(sum(s ^ 2))
    })
    samples = samples/radii
samples
}
```

```
points.sphere = d_dimension_sphere(D=3)
x = points.sphere[, 1]
y = points.sphere[, 2]
```

```
z = points.sphere[, 3]
colors <- c('#56B4E9')
scatterplot3d(x,y,z,color=colors,pch = '.')
```

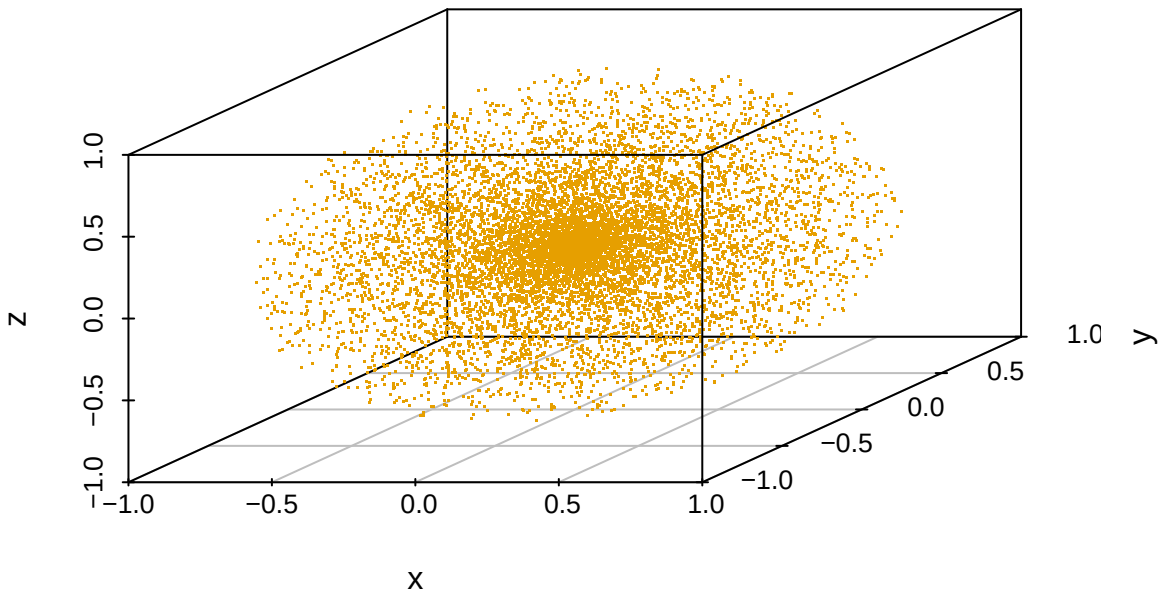


Ball sampling:

Muller Method Similar to the 3-ball Muller, it just extend to higher dimension.

```
d_dimension_ball = function(D) {
    samples = sapply(1:D, function(d) {
        rnorm(n=10000)
    })
    radii = apply(samples, 1, function(s) {
        sqrt(sum(s ^ 2))
    })
    new.radii = runif(10000) / radii
    samples = samples * new.radii
samples
}
```

```
points.sphere = d_dimension_ball(D=3)
x = points.sphere[, 1]
y = points.sphere[, 2]
z = points.sphere[, 3]
colors <- c('#E69F00')
scatterplot3d(x,y,z,color=colors,pch = '.')
```

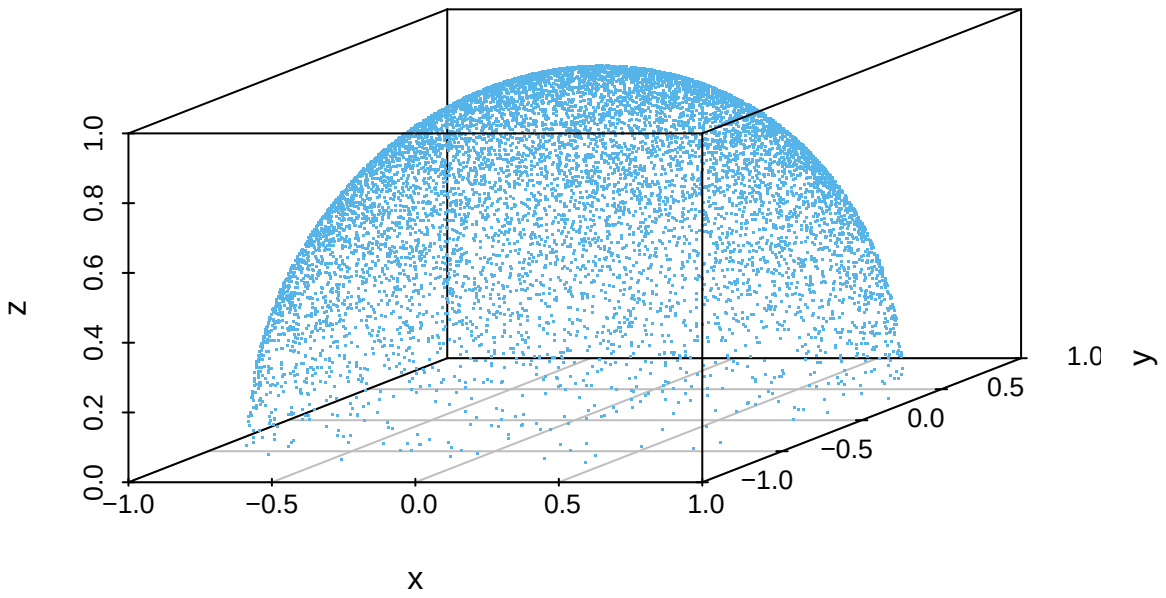


Extension: How to sample from unit hemisphere surface? (Given one of the coordinates is positive)

Cosine-weighted hemisphere This is the natural generalization. Main difference from the sphere is the interval of theta to make sure theta range from ($-\pi/2$,$\pi/2$). In such a case, $\cos(\theta)>0$.

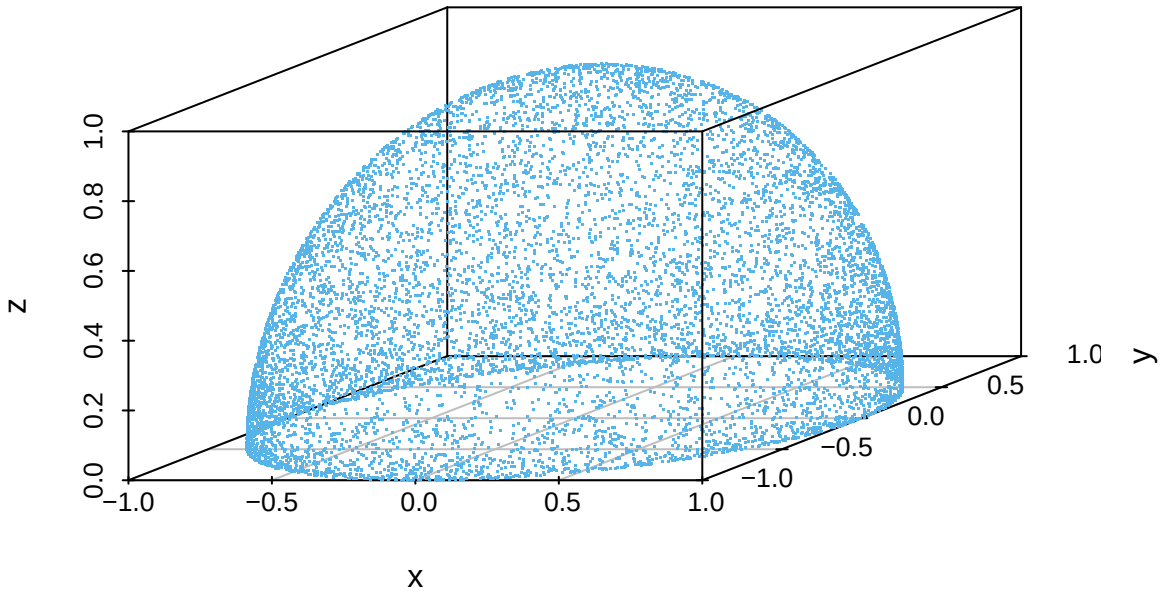
```
#r-3 space
u=runif(10000)
v=runif(10000)
theta=acos(sqrt(u))
phi=2*pi*v

x=sin(theta)*cos(phi)
y=sin(theta)*sin(phi)
z=cos(theta)
colors <- c('#56B4E9')
scatterplot3d(x,y,z,color=colors,pch = '.')
```



We can also use u=cosine theta which does not use inverse trig functions as follows:

```
u=runif(10000)
v=runif(10000)
a = (1 - 2*u)^2
b = sqrt(1 - a*a)
phi = 2*pi*v
x = b*cos(phi)
y = b*sin(phi)
z = a
colors <- c('#56B4E9')
scatterplot3d(x,y,z,color=colors,pch = '.')
```


z
0.0
0.2
0.4
0.6
0.8
1.0
−1.0
−0.5
0.0
0.5
1.0
x
−1.0
−0.5
0.0
0.5
1.0
y