In[ ]:=
```
Remove["Global`*"]
```

Generate a set of values that follow a Gaussian probability distribution then fit a binning of the data.

Choose values for mean u and standard deviation std. Generate n random numbers between -1 and +1.

In[ ]:=
```
u = 10
std = 1
```

Out[ ]=
```
10
```

Out[ ]=
```
1
```

In[ ]:=
```
SeedRandom[10]
```

Out[ ]=
```
RandomGeneratorState[ Method: ExtendedCA
                      State hash: 8 826 066 408 037 144 672 ]
```

In[ ]:=
```
n = RandomReal[{-1, 1}, 10 000]
```

Out[ ]=
```
{0.335833, 0.667747, -0.0773686, -0.033475, 0.904067,
 ··· 9990 ···, -0.766481, -0.979913, 0.0712523, -0.527152, -0.00248799}
```

large output | show less | show more | show all | set size limit...

Turn the list into a list that is distributed according to the gaussian e^-t^2

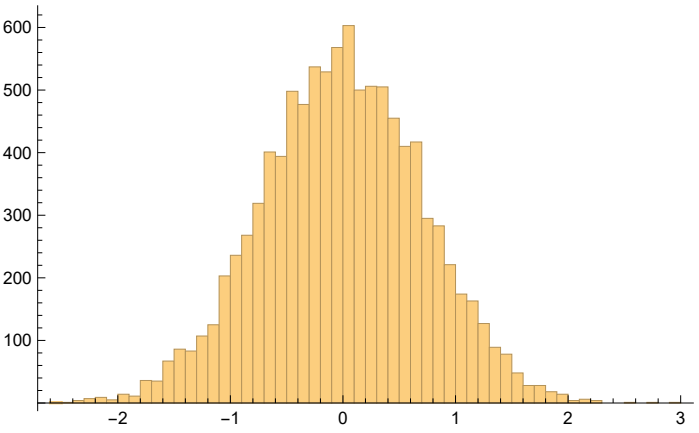
In[ ]:=
```
ni = InverseErf[n]
```

Out[ ]=
```
{0.307003, 0.685601, -0.0686739, -0.0296751, 1.17726,
 ··· 9990 ···, -0.842407, -1.64383, 0.0632299, -0.507599, -0.00220493}
```

large output | show less | show more | show all | set size limit...

In[ ]:=
```
Histogram[ni, 50]
```

Out[ ]=

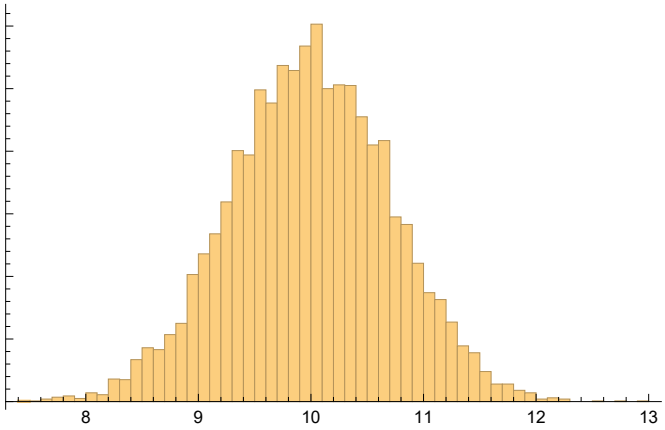


Turn n list into a list containing select u and std values

In[ ]:=
```
l = ni * std + m;
```

In[ ]:=
```
Histogram[l, 50]
```

Out[ ]=



Extract histogram frequencies and centers of the bins

In[ ]:=
```
xlo = Min[l]
xhi = Max[l]
dx = (xhi - xlo) / 50
```

Out[ ]=
```
7.48287
```

Out[ ]=
```
12.9811
```

Out[ ]=
```
0.109966
```

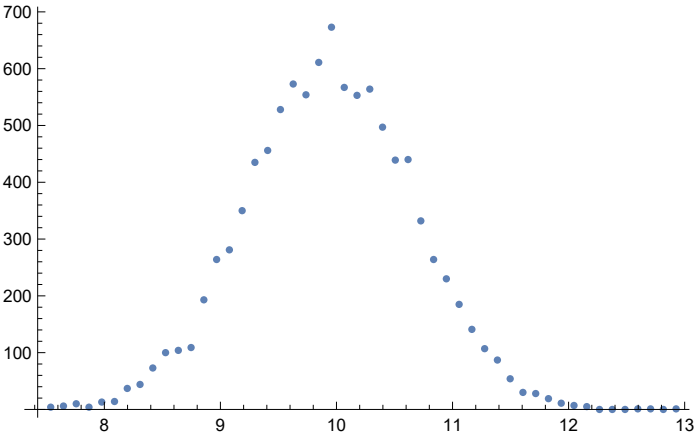
In[ ]:=
```
r = Range[xlo, xhi, dx];
rr = r + .5 * dx;
```

In[ ]:=
```
bc = BinCounts[l, {rr}];
cn = Delete[rr, 51];
```

Plot of frequencies vs bin center

In[•]:=
```
d = Transpose@{cn, bc};
ListPlot[d]
```

Out[•]=



Integrating the Gaussian probability distribution P

In[•]:=
```
P = 1 / (std * Sqrt[2 Pi]) * E^(-((x - m)^2) / (2 (std^2)))
```

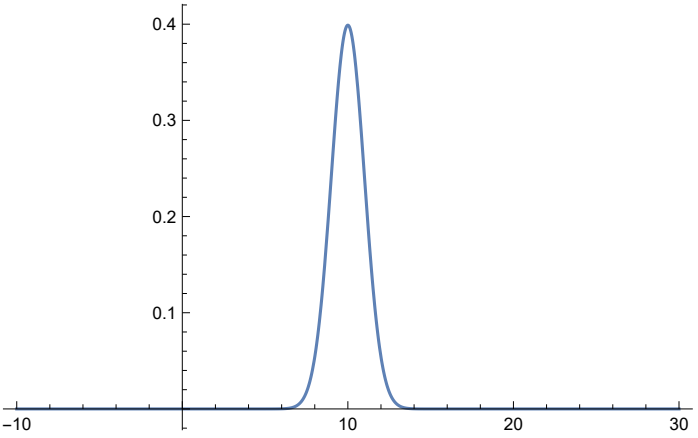
Out[•]=
$$\frac{e^{-\frac{1}{2}(-10+x)^2}}{\sqrt{2\pi}}$$

In[•]:=
```
NIntegrate[P, {x, -100, 100}]
```

Out[•]=
```
1.
```

In[•]:=
```
Plot[P, {x, -10, 30}, PlotRange → Full]
```

Out[•]=



In[•]:=
```
px = FindFit[d, p * P, p, x]
```

Out[•]=
```
{p → 1266.63}
```

In[•]:=
```
pfit = p /. px[[1]]
```

Out[•]=
```
1266.63
```

In[ ]:=
```
Show[Plot[pfit * P, {x, -10, 30}, PlotRange → Full, PlotStyle → Gray],
 ListPlot[d, PlotRange → Full, PlotStyle → Dashed]]
```

Out[ ]=