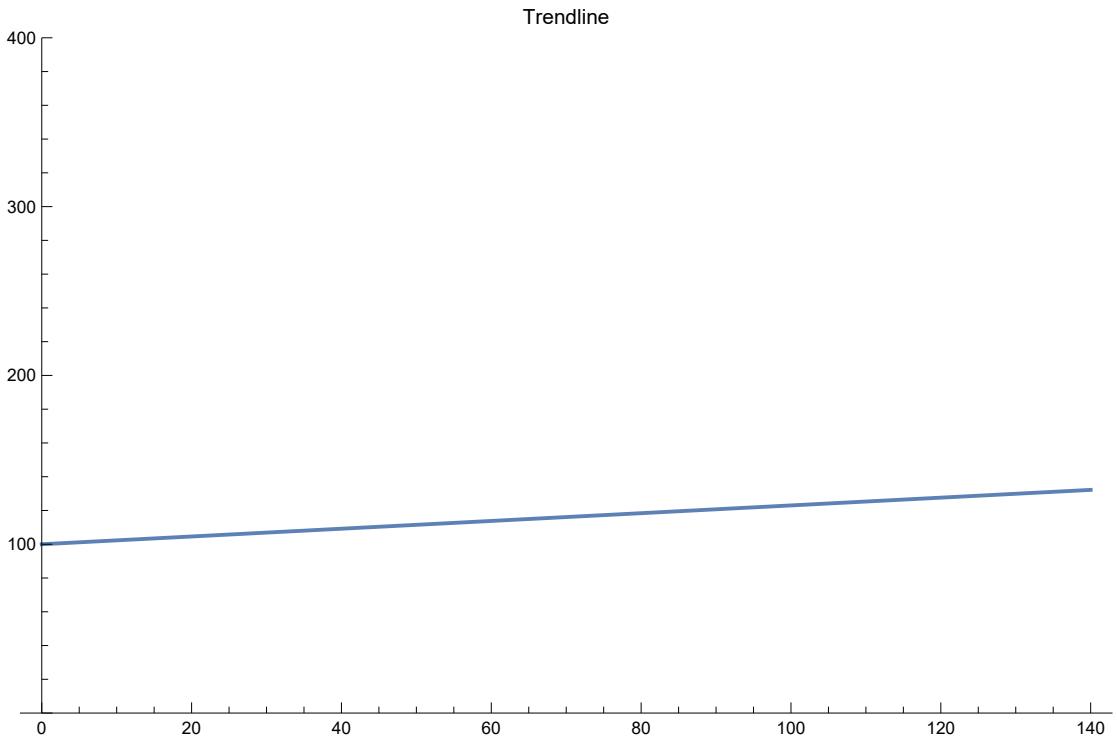
Quackery 2, Correlation Plot

Running $N = 10^4$ Monte Carlo simulations to see the actual linear coefficient, assuming that the scatter of IQ is normally distributed with mean $IQ = 100$ and $\sigma_{IQ} = 15$ as is typical for the literature.

[Image]

```
In[•]:= f = a * x + b /. {a → .23, b → 100}
Out[•]=
100 + 0.23 x

In[•]:= Plot[f, {x, 0, 140}, PlotRange → {0, 400}, PlotLabel → "Trendline", ImageSize → Large]
Out[•]=
```



Assume that number of participants (IQ) is normally distributed. In reality, need to trace number of dots and manually count how many participants.

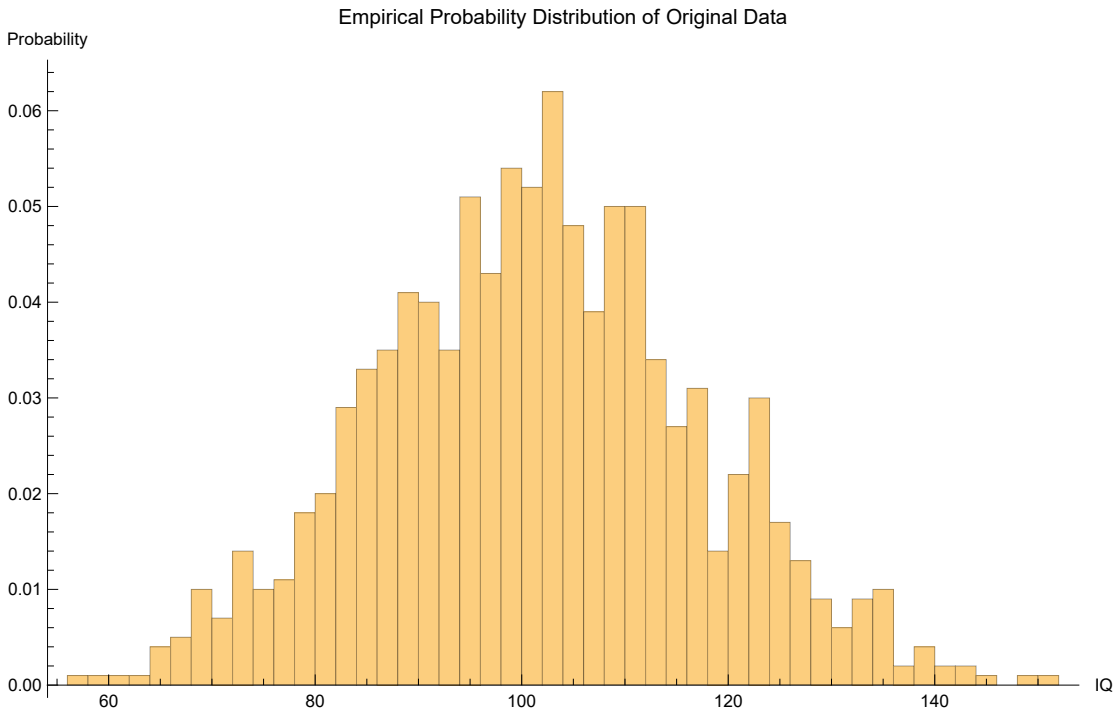
IQ being normally distributed is quite controversial, however, I am only using this based on how the scatterplot looked and try to replicate their scatterplot.

In[•]:=
```
ta = Round[RandomVariate[NormalDistribution[100, 15], 1000]];

dist = EmpiricalDistribution[ta];

Histogram[ta, 30, Probability,
 PlotLabel → "Empirical Probability Distribution of Original Data",
 AxesLabel → {"IQ", "Probability"}, ImageSize → Large]
```

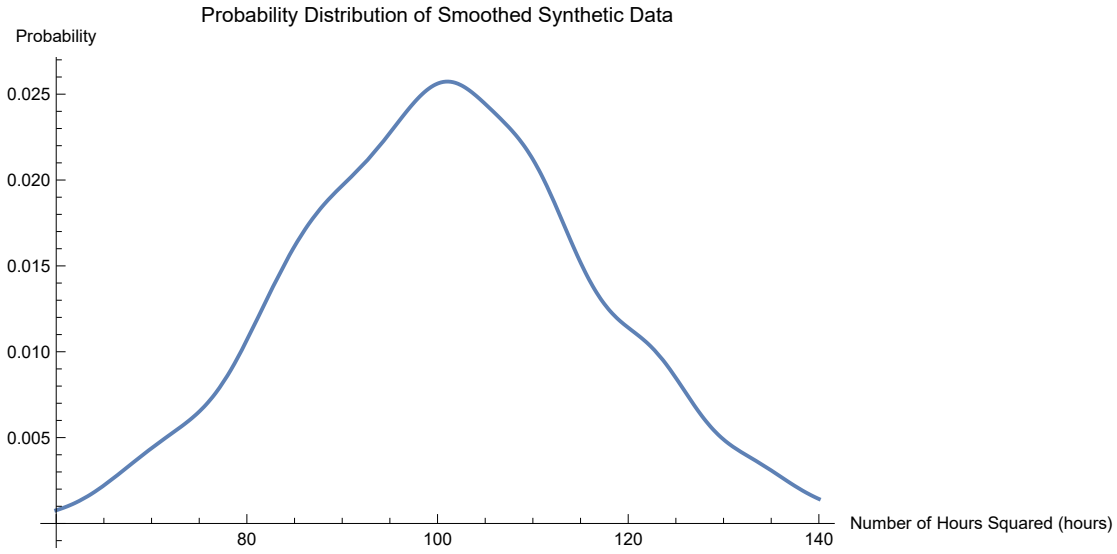
Out[•]=



In[•]:=
```
dist2 = SmoothKernelDistribution[ta];
```

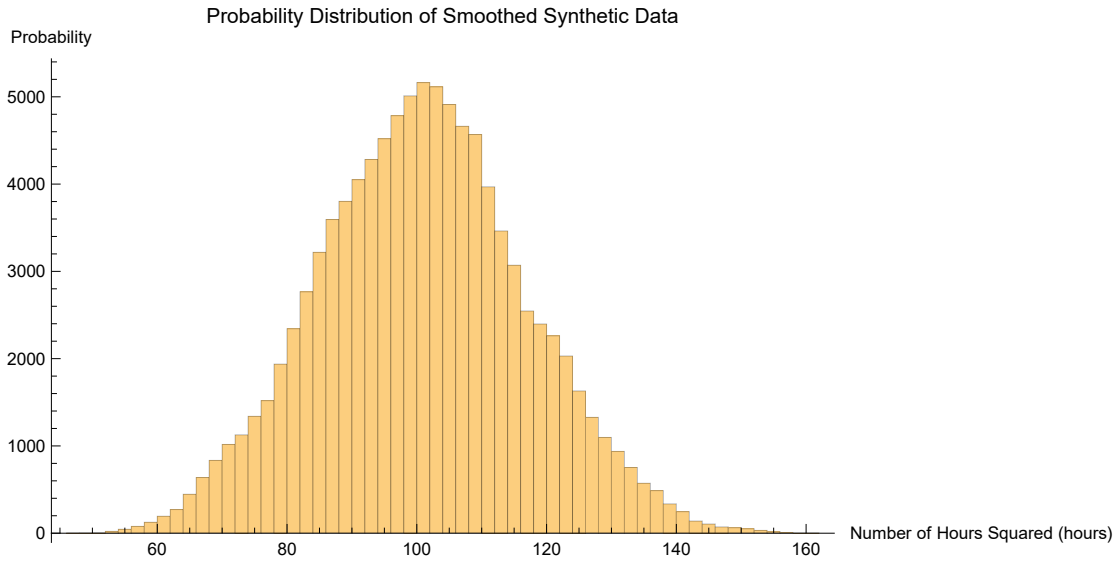
In[•]:=
```
Plot[PDF[dist2, x], {x, 60, 140},
 PlotLabel → "Probability Distribution of Smoothed Synthetic Data",
 AxesLabel → {"Number of Hours Squared (hours)", "Probability"}, ImageSize → Large]
```

Out[•]=



In[•]:=
```
(*Take50bins,showthediscretenumberofsamplestomakesureitisabout20000*)
Histogram[RandomVariate[dist2, 10^5], 50,
 PlotLabel → "Probability Distribution of Smoothed Synthetic Data",
 AxesLabel → {"Number of Hours Squared (hours)", "Probability"}, ImageSize → Large]
```

Out[•]=



In[•]:=
```
f = a * x + b /. {a → .23, b → 100}
```

Out[•]=

$$100 + 0.23\,x$$

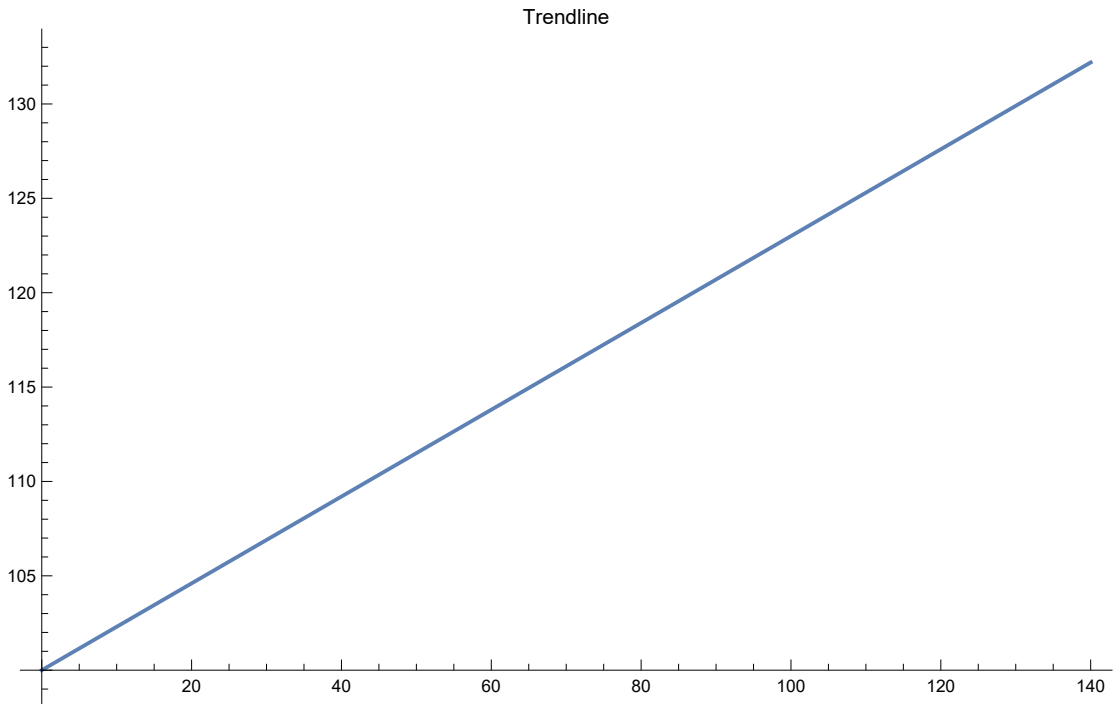
In[•]:=
```
f1 = f /. x → ta;
```

In[•]:=
```
reg = Transpose[{ta, f1}];
```

In[•]:=
```
lm = LinearModelFit[reg, {1, x}, x];
```

In[•]:=
```
Plot[lm[x], {x, 0, 140}, PlotLabel → "Trendline", ImageSize → Large]
```

Out[•]=



In[•]:=
```
f1 // Length
```

Out[•]=
```
1000
```

In[•]:=
```
reg1[sigma_] :=
 Transpose[{ta, f1 + RandomVariate[NormalDistribution[0, sigma], Length[f1]]}]
```

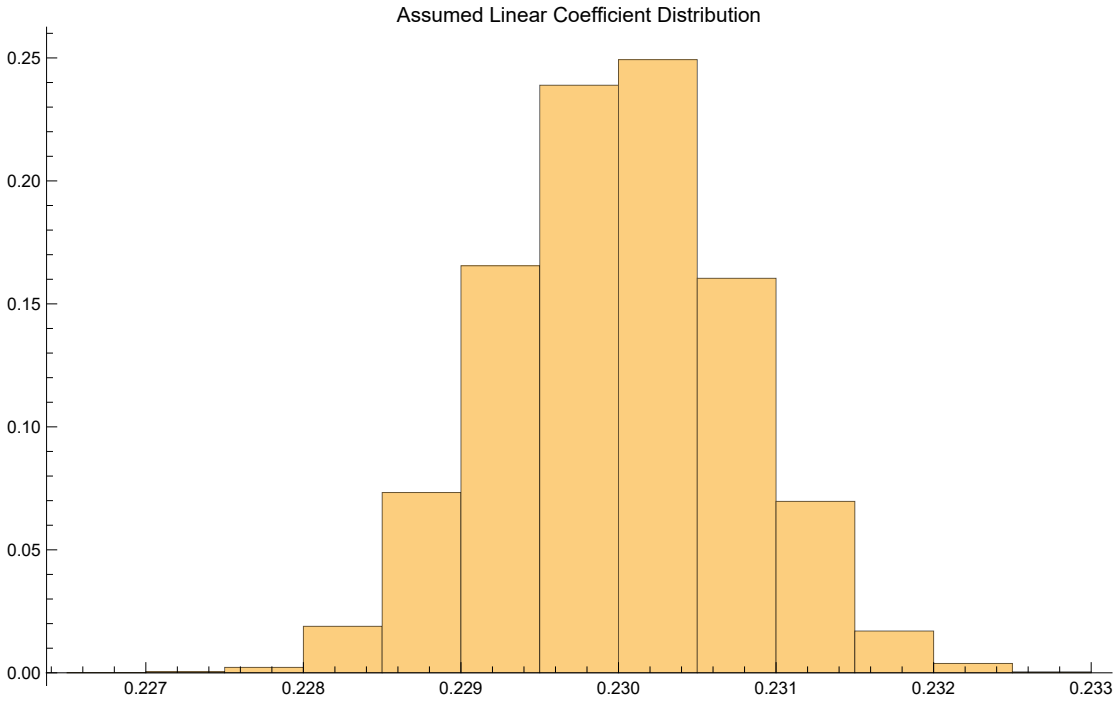
This is the key step, you add Gaussian noise to the function, and then tune the standard deviation σ_{noise} to get the *p*-value they want.

In[•]:=
```
sig = 0.375;(*Tuningthenoisetogetp<0.001*)
res = Table[lm2 = LinearModelFit[reg1[sig], {1, x}, x];
   lm2[x][[2]][[1]], {z, 1, 10^4}];
{Mean[res], StandardDeviation[res], Skewness[res], Kurtosis[res]}
```

Out[•]=
```
{0.229996, 0.000750371, 0.0127748, 2.91856}
```

In[•]:=
```
pl0 = Histogram[res, Automatic, "Probability",
  PlotLabel → "Assumed Linear Coefficient Distribution", ImageSize → Large]
```

Out[•]=

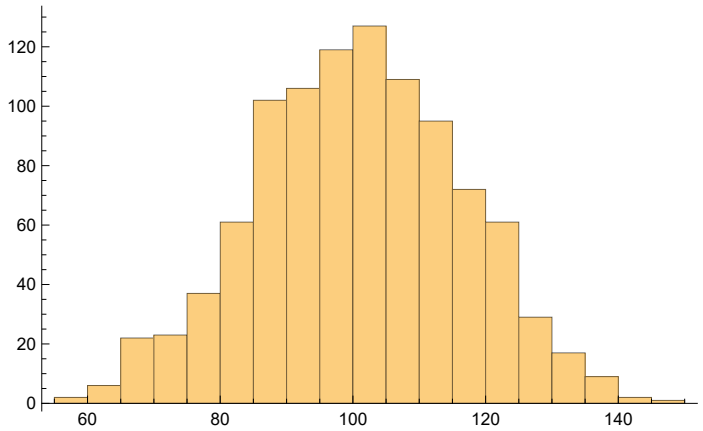


Now, use the smooth kernel normal distribution using the empirical length of about 1000 samples, and then generate a new table.

In[•]:=
```
ta := RandomVariate[dist2, Length[f1]]
 (*Takethesmoothkernel,taketheempiricallength,generatetable*)
```

In[•]:=
```
Histogram[ta]
```

Out[•]=



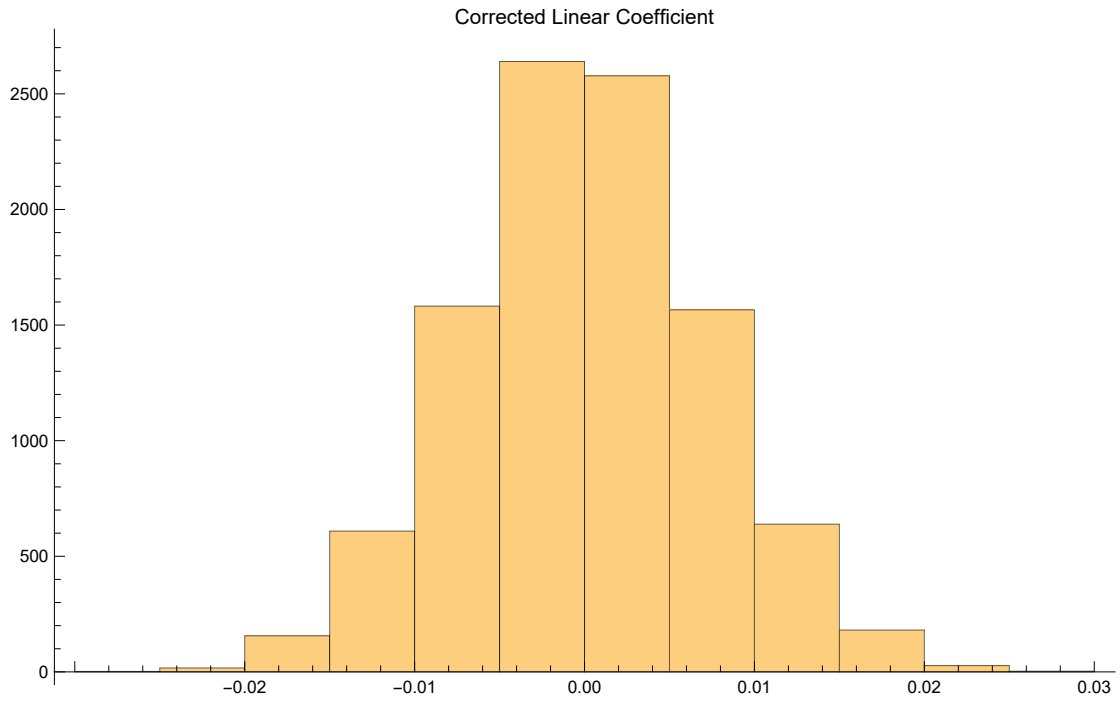
Watch what happens to the linear coefficient.

lm2[x][[2]][[1]]: The first element of the coefficients is extracted. This would correspond to the coefficient of the linear term in the fitted model.

In[•]:=
```
ta2 = Table[lm2 = LinearModelFit[reg1[0.375], {1, x}, x]; lm2[x][[2]][[1]], {z, 1, 10^4}];
```

In[•]:=
```
pl2 = Histogram[ta2, PlotLabel → "Corrected Linear Coefficient", ImageSize → Large]
```

Out[•]=


Corrected Linear Coefficient
2500
2000
1500
1000
500
0
−0.02
−0.01
0.00
0.01
0.02
0.03