Phys 238: Uncertainties resulting from Random, Independent errors

In[86]:=
```
Clear["Global`*"]
```

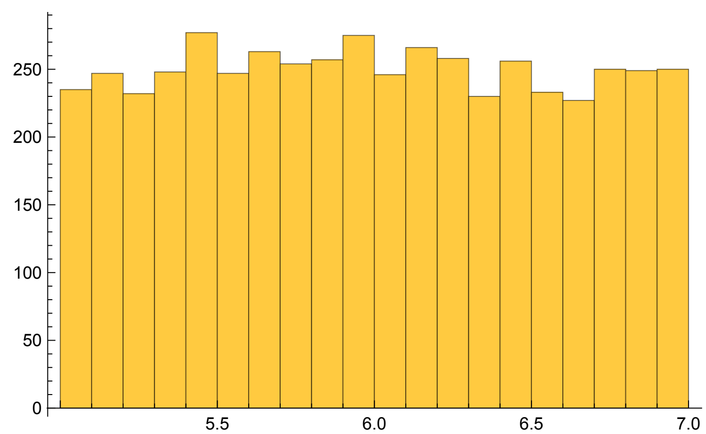
Individual Random Numbers

In[87]:=
```
RandomReal[{5 , 7}] (* Generates a random number between 5 and 7. *)
```
Out[87]=
```
5.52889
```

In[88]:=
```
RandomReal[{5, 7}, 10] (* Generate a bunch (10) of them *)
```
Out[88]=
```
{6.54766, 6.25998, 6.01958, 6.7685,
 6.60214, 6.18854, 6.71273, 5.40059, 6.63115, 6.17335}
```

In[89]:=
```
vals = RandomReal[{5, 7}, 5000]; (* A big bunch *)
```

In[90]:=
```
Histogram[vals]
```
Out[90]=



Repeating a single measurement that results from the sum of many individual random numbers should give a normal distribution.

Define a function that represents one data point that is a result of many independent random numbers.

In[91]:=
```
onepoint := Mean[RandomReal[{5, 7}, 200]]
```

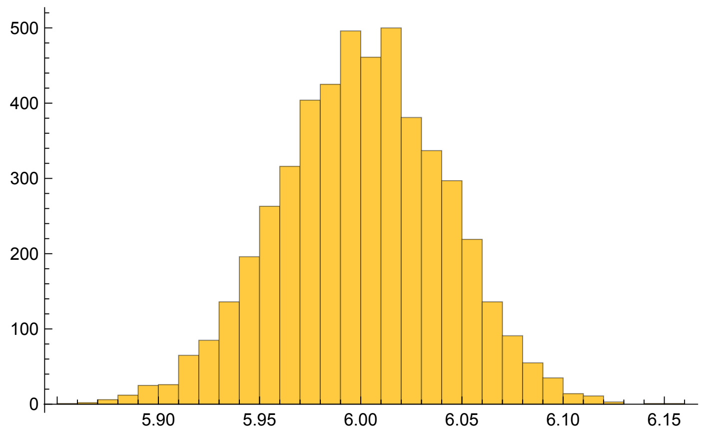
This returns a different measurement each time you try it.

In[92]:=
```
onepoint
```
Out[92]=
```
6.04731
```

In[93]:=
```
onepoint
```
Out[93]=
```
5.95473
```

Now imaging doing 5000 such measurements.

In[94]:=
```
data = Table[onepoint, {i, 1, 5000}];
```
In[95]:=
```
Histogram[data]
```
Out[95]=



In[96]:=
```
μ = Mean[data]
σ = StandardDeviation[data]
```
Out[96]=
```
6.00058
```
Out[97]=
```
0.0410498
```

How many are within ± 1 σ of the mean? Use the ‘Select’ function to select data points within σ of μ.

In[98]:=
```
Length[Select[data, μ - σ ≤ # ≤ μ + σ &]] / Length[data];
StringForm["``, or ``%", %, 100 * N[%]]
```
Out[99]=

$\frac{3371}{5000}$, or 67.42`%

Now try to model this distribution by a Gaussian. Getting a manual list of the histogram numbers is a bit tedious. The HistogramList[ ] function actually gives two separate lists, first is a list of bin boundaries, and second is the list of counts in each bin.

In[100]:=
```
{bins, counts} = N[HistogramList[data]]
```
Out[100]=
```
{{5.85, 5.86, 5.87, 5.88, 5.89, 5.9, 5.91, 5.92, 5.93,
  5.94, 5.95, 5.96, 5.97, 5.98, 5.99, 6., 6.01, 6.02, 6.03, 6.04, 6.05,
  6.06, 6.07, 6.08, 6.09, 6.1, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16},
 {1., 2., 6., 12., 25., 26., 65., 85., 136., 196., 263., 316., 404., 425., 496.,
  461., 500., 381., 337., 297., 219., 136., 91., 55., 35., 14., 11., 3., 0., 1., 1.}}
```

In[101]:=

```
bins
```

Out[101]=

```
{5.85, 5.86, 5.87, 5.88, 5.89, 5.9, 5.91, 5.92, 5.93, 5.94,
 5.95, 5.96, 5.97, 5.98, 5.99, 6., 6.01, 6.02, 6.03, 6.04, 6.05,
 6.06, 6.07, 6.08, 6.09, 6.1, 6.11, 6.12, 6.13, 6.14, 6.15, 6.16}
```

In[102]:=

```
counts
```

Out[102]=

```
{1., 2., 6., 12., 25., 26., 65., 85., 136., 196., 263., 316., 404., 425., 496., 461.,
 500., 381., 337., 297., 219., 136., 91., 55., 35., 14., 11., 3., 0., 1., 1.}
```

Interleaving them is straightforward, if verbose. We will assign each height to the center point of each bin by constructing a new table.

In[103]:=

```
hist = Table[{(bins[[i]] + bins[[i + 1]]) / 2., counts[[i]]},
   {i, 1, Length[counts]}];
```

Enter an unnormalized gaussian (Eq. 5.20 in Taylor's text):

In[104]:=

$$\text{gauss[A\_, }\mu\text{\_, }\sigma\text{\_, x\_] := }\frac{A}{\sigma\,\text{Sqrt}[2\,\pi]}\,\text{Exp}\left[-\frac{(x-\mu)^2}{2\,\sigma^2}\right]$$

In[105]:=

```
Clear[A, μ, σ]
(* The initial fit attempt does always work well.  It will be useful to
 seed the process with guesses at the mean and standard deviation. *)
result = NonlinearModelFit[hist, gauss[A, μ, σ, x],
  {A,
   {μ, Mean[data]},
   {σ, StandardDeviation[data]}
  }, x]
```

Out[106]=

FittedModel[$484.\, e^{-293.\ \ll 1 \gg}$]

In[107]:=

```
result["BestFitParameters"]
```

Out[107]=

{A → 50.0606, μ → 6.00128, σ → 0.04129}

In[108]:=

```
result["ParameterConfidenceIntervalTable"]
```

Out[108]=

	Estimate	Standard Error	Confidence Interval
A	50.0606	0.647771	{48.7337, 51.3875 }
μ	6.00128	0.00061693	{6.00002, 6.00255 }
σ	0.04129	0.000616944	{0.0400262 , 0.0425537 }

These are very close to the calculated mean and standard deviation:

In[109]:=

```
{Mean[data], StandardDeviation[data]}
```

Out[109]=

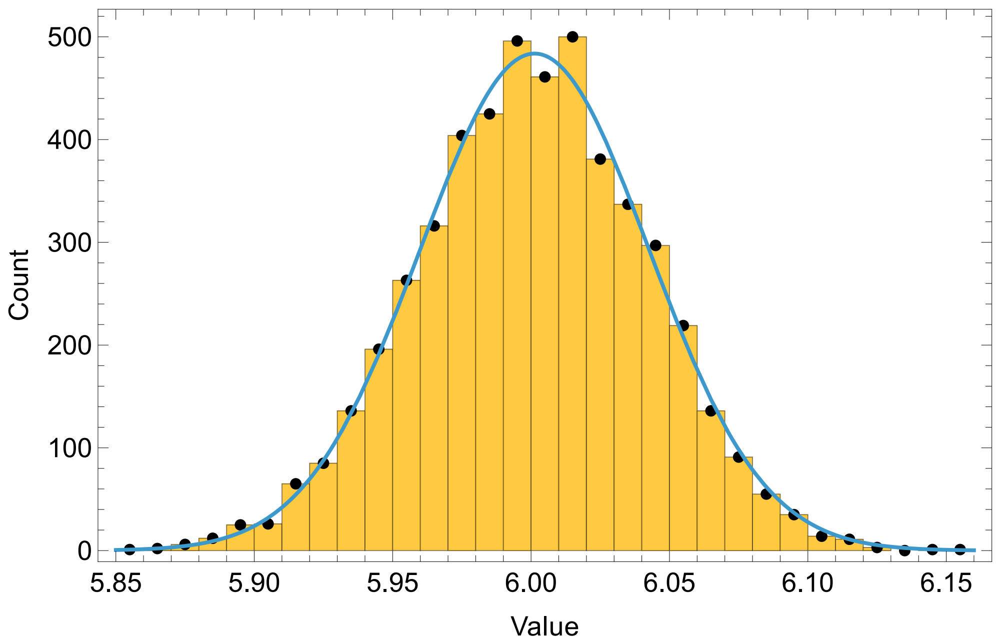
```
{6.00058, 0.0410498}
```

Show the histogram, the data used in the fit, and the best-fit gaussian all on the same graph. Add useful titles.

In[110]:=

```
randerrplot = Show[{
   Histogram[data],
   ListPlot[hist, PlotStyle → Black],
   Plot[result[x], {x, Min[bins], Max[bins]}]},
  Frame → True, FrameLabel → {"Value", "Count"},
  LabelStyle → Larger, ImageSize → Scaled[0.8]]
```

Out[110]=



In[111]:=

```
SetDirectory[NotebookDirectory[]] (* Save it in this directory *)
```

Out[111]=

```
/Users/doughera/238/2025/Mathematica-dev
```

In[113]:=

```
Export["randerrplot.pdf", randerrplot]
```

Out[113]=

```
randerrplot.pdf
```