Physics 238: Fitting a Normal Distribution to the Pendulum Data

Introduction

In this notebook, you will see how to import your data, create a histogram, and fit the normal distribution to your data. I have used this notebook to introduce a number of powerful *Mathematica* functions that I hope will be useful beyond this experiment. Please make extensive use of *Mathematica*'s built-in Help function to learn more about any of the functions used here. Also feel free to add in extra steps where you print out various arrays to see what is happening.

For your final submitted work, however, please trim out all unnecessary or duplicate printouts. Make it concise, but accurate and readable.

Getting Started

Clear all data and functions.

```
In[10]:= Clear["Global`*"]
```

Importing Data

You need to tell *Mathematica* where to find your data file. If you store your notebook in the same directory as your data, you can use the following trick to set *Mathematica*'s default directory. Otherwise, you have to either give the full pathname or give the pathname relative to *Mathematica*'s default.

This command assumes you have previously extracted the period values (with the pendulum-period.nb notebook) and saved them in a local file.

Note the trailing ';' on each line. If you omit them, *Mathematica* prints out the result of the command. Try it and see.

```
In[11]:= SetDirectory[NotebookDirectory[]];
         data = Import["pendulum-20250129-period.csv", "CSV"];
```

Look at the first few elements of the array:

```
In[13]:= data⟦Range[5]⟧
Out[13]=
         {{1.93468}, {1.93477}, {1.93487}, {1.93471}, {1.9345}}
```

Mathematica has decided to treat each row of the data file as a list (in curly braces) of length 1. Look at the dimensions of the array. Mathematica thinks the data is 516 lists, each of length one.

In[14]:=
```
Dimensions[data]
```
Out[14]=
```
{516, 1}
```

We want just a single array of 516 elements. The way to get rid of the inner lists is to use the Flatten[] command:

In[15]:=
```
data = Flatten[data];
data⟦Range[5]⟧
```
Out[16]=
```
{1.93468, 1.93477, 1.93487, 1.93471, 1.9345}
```

Now the data is a simple list. We can find out how many data points we have by finding the length of the array.

In[17]:=
```
Length[data]
```
Out[17]=
```
516
```

Alternatively, if you know the data is just a single number per line (as it is here) you can tell Mathematica that the data is a simple list by telling the Import[ ] function to treat it as a list:

In[18]:=
```
data = Import["pendulum-20250129-period.csv", "List"];
```

Now the data is imported directly as a simple list. No Flatten[ ] command is needed.

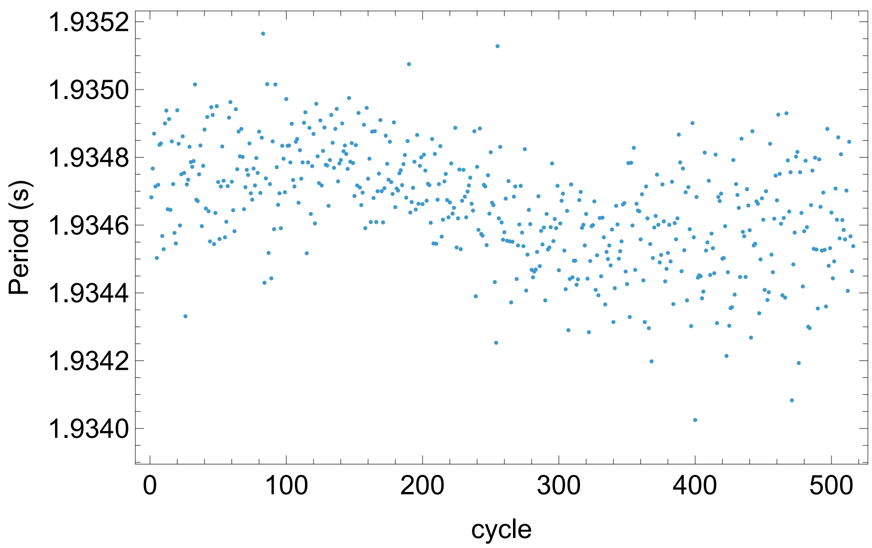
In[19]:=
```
data⟦Range[5]⟧
```
Out[19]=
```
{1.93468, 1.93477, 1.93487, 1.93471, 1.9345}
```

Eliminating Bad Points

If there are any outlier points, they should be easy to spot in a graph. Most likely, someone walked past the apparatus. More generally, it is often reasonable to compute the spread of the data and throw out spurious points that are many standard deviations away from the mean. Don't do this unless you have good reason to suspect those points are bad, however!

In[20]:=
```
ListPlot[data, PlotRange → All, Frame → True, LabelStyle → Larger,
  FrameLabel → {"cycle", "Period (s)"}, ImageSize → Scaled[0.7]]
```

Out[20]=



Make a list of the minimum, maximum, average, and standard deviation for our data.

In[21]:=
```
{min = Min[data], max = Max[data], avg = Mean[data], σ = StandardDeviation[data]}
```

Out[21]=
```
{1.93403, 1.93517, 1.93466, 0.000168487}
```

How far away from the mean are the min and max points? Compare them to the standard deviation:

In[22]:=
```
(max - avg) / σ
```

Out[22]=
```
3.02333
```

In[23]:=
```
(min - avg) / σ
```

Out[23]=
```
-3.74276
```

If any points are more than 5 σ away from of the mean, we could make a case for deleting them. One or two points likely won't make much statistical difference either way. Also, in this experiment, the data sometimes get erratic after a long time. That didn't happen here, but sometimes it does. The commands below show how to consider only at most the first 600 data points and then select only those points within 5 σ of the mean. You can adjust the '600' and '5' number as appropriate for a particular data set based on your graph above.

In[24]:=
```
data = If[Length[data] > 600, data[[Range[600]]], data];
(* Keep at most 600 points *)
data = Select[data, Abs[# - avg] < 5 σ &];
Length[data]
```

Out[26]=
```
516
```

Now compute our statistics and display them as a simple list.

In[27]:=
```
min = Min[data]; max = Max[data]; avg = Mean[data];
σ = StandardDeviation[data]; δT = σ / Sqrt[Length[data]];
{min, max, avg, σ, δT}
```

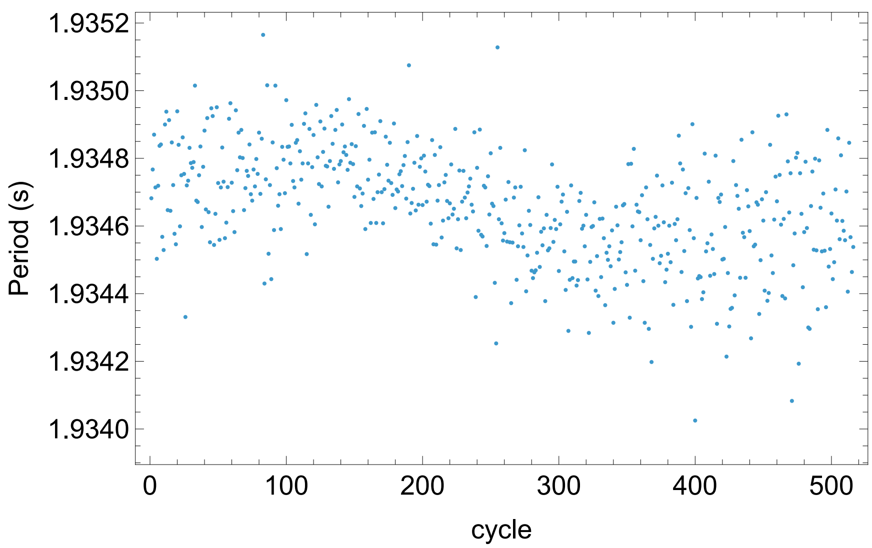
Out[29]=

$\{1.93403, 1.93517, 1.93466, 0.000168487, 7.41724 \times 10^{-6}\}$

This is our best representation of the good data.

In[30]:=
```
dataplot = ListPlot[data, PlotRange → All, Frame → True, LabelStyle → Larger,
  FrameLabel → {"cycle", "Period (s)"}, ImageSize → Scaled[0.7]]
```

Out[30]=



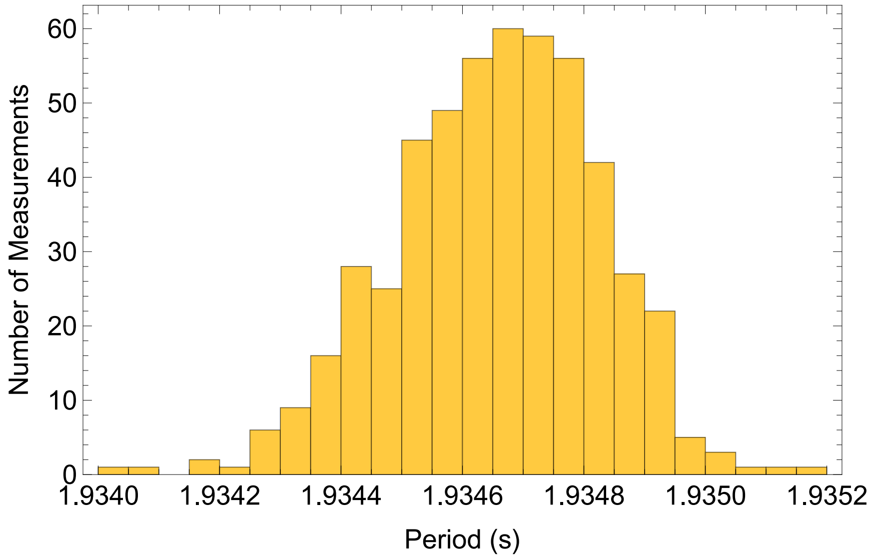
Making a Histogram

Mathematica can make a nice histogram. It often (but not always) chooses reasonable bins. For this data, I have overridden the default and picked a bin size of 0.0005s.

In[31]:=
```
binsize = 0.00005;
```

In[32]:=
```
histplot = Histogram[data, {binsize}, Frame → True, LabelStyle → Larger,
  FrameLabel → {"Period (s)", "Number of Measurements"}, ImageSize → Scaled[0.7]]
```

Out[32]=



To fit this data to a normal distribution, see the ‘randerrors.nb’ notebook.