Phys 238 : Reading .CSV Files With *Mathematica*

Getting Started

Mathematica remembers definitions, even if you delete them from the notebook. This can sometimes be confusing, so it's often a good idea to simply clear everything and start anew. This is especially true if you use the Evaluation -> Evaluate Notebook menu item.

```
In[•]:= Clear["Global`*"] (* Clear all variable and function definitions *)
```

Reading a CSV file

It is not always obvious where Mathematica will look for data files or export graphs. It is often useful to keep your data in the same directory as the notebook. The following command sets the default directory to be the same as the directory for this notebook.

```
In[•]:= SetDirectory[NotebookDirectory[]]
Out[•]= /Users/doughera/238/2025/Mathematica-dev
```

Mathematica has many different ways to read and interpret .csv files. One useful way is to Import the data into what Mathematica calls a "Dataset", which is a structured object with it's own special query language. Here's how it works for the pendulum data.

In[ ]:=
```
fulldata = Import["pendulum-20250129.csv", "Dataset"]
```

Out[ ]=

Latest: Time (s)	Latest: GateState	Latest: Period (s)
0.139623	1	—
0.222001	0	—
1.14797	1	—
1.23138	0	—
2.0743	1	1.93468
2.15672	0	—
3.08287	1	—
3.16634	0	—
4.00907	1	1.93477
4.09148	0	—
5.01748	1	—
5.10137	0	—
5.94394	1	1.93487
6.02656	0	—
6.95224	1	—
7.03638	0	—
7.87866	1	1.93471
7.96142	0	—
8.88701	1	—
8.97117	0	—

rows 1–20 of **2068**

To extract the period data, we want to pick {All rows, 3rd column}, but we can also refer to the column by name:

```
In[ ]:= data = fulldata[All, "Latest: Period (s)"]
```

Out[ ]=

—	—	—	—
1.93468	—	—	—
1.93477	—	—	—
1.93487	—	—	—
1.93471	—	—	—
1.9345	—	—	—
1.93472	—	—	—
1.93484	—	—	—
1.93484	—	—	—
1.93457	—	—	—
1.93453	—	—	—
1.9349	—	—	—
1.93494	—	—	—
1.93465	—	—	—
1.93491	—	—	—
1.93464	—	—	—
1.93485	—	—	—
1.93472	—	—	—
1.93458	—	—	—
1.93455	—	—	—

elements 1–80 of **2068**

For this particular data set, this still has the problem that most of the rows are empty. It is possible to filter them out while importing, but a more generally useful approach is to use the Select[ ] function to select only data elements that are numbers. (Select can also be used for much more complex testing, but this is sufficient for here.)

In[ ]:=
```
data = Select[data, NumberQ]
```

Out[ ]=

1.93468	1.93477	1.93487	1.93471	1.9345	1.93472
1.93484	1.93484	1.93457	1.93453	1.9349	1.93494
1.93465	1.93491	1.93464	1.93485	1.93472	1.93458
1.93455	1.93494	1.93484	1.9346	1.93475	1.93486
1.93475	1.93433	1.93472	1.93473	1.93483	1.93479
1.93477	1.93479	1.93502	1.93468	1.93467	1.93475
1.93484	1.9346	1.93478	1.93488	1.93465	1.93492
1.93464	1.93455	1.93495	1.93492	1.93454	1.93464
1.93495	1.93473	1.93456	1.93471	1.93483	1.93473
1.93456	1.93461	1.93472	1.93492	1.93496	1.93473
1.93464	1.93458	1.93494	1.93477	1.93488	1.9348
1.93488	1.9348	1.93465	1.93471	1.93476	1.93475
1.93484	1.93469	1.93468	1.93477	1.93472	1.9348
1.93475	1.93488	1.9347	1.93486	1.93516	1.93443
1.93474	1.93502	1.93452	1.93472	1.93444	1.93485
1.93459	1.93502	1.93477	1.93466	1.93469	1.93459
1.93483	1.9348	1.9347	1.93497	1.93483	1.93484
1.93478	1.9349	1.93473	1.93471	1.93485	1.93485
1.93467	1.93482	1.93474	1.93478	1.9349	1.93493
1.93452	1.93478	1.93489	1.93463	1.93477	1.93487

elements 1–120 of **516**

Summary of Import:

For this data set, here are the relevant commands,

In[143]:=
```
SetDirectory[NotebookDirectory[]]
```

Out[143]=
```
/Users/doughera/238/2025/Mathematica-dev
```

In[144]:=
```
fulldata = Import["pendulum-20250129.csv", "Dataset"];
```

In[145]:=
```
data = Select[fulldata[All, "Latest: Period (s)"], NumberQ];
```

(Optional) Save the period data for later importing.

In[162]:=
```
Export["pendulum-20250129-period.csv", data]
```
Out[162]=
```
pendulum-20250129-period.csv
```

Statistical Analysis

In[146]:=
```
npts = Length[data]
```
Out[146]=
```
516
```

In[147]:=
```
T = Mean[data]
```
Out[147]=
```
1.93466
```

In[148]:=
```
σT = StandardDeviation[data]
```
Out[148]=
```
0.000168487
```

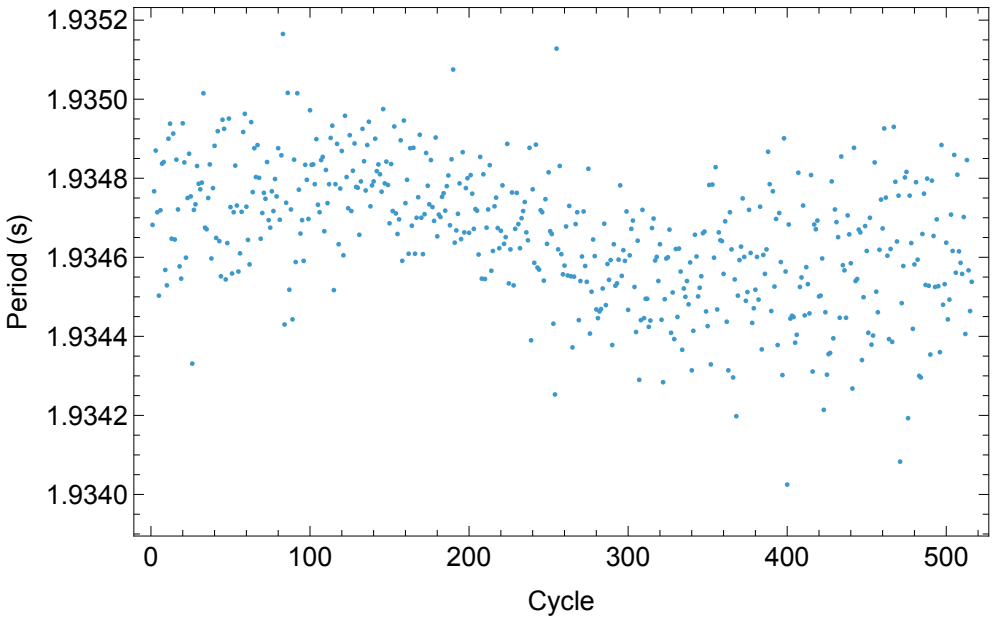
In[149]:=
```
δT = σT / Sqrt[npts]
```
Out[149]=
$$7.41724 \times 10^{-6}$$

Mathematica can express this as a value with uncertainty using the Around[ ] function:

In[150]:=
```
Around[T, δT]
```
Out[150]=
$$1.93465(6 \pm 7)$$

In[151]:=
```
Tplot = ListPlot[data, LabelStyle → Larger, Frame → True,
  FrameLabel → {"Cycle", "Period (s)"}, ImageSize → Scaled[0.8]]
```

Out[151]=

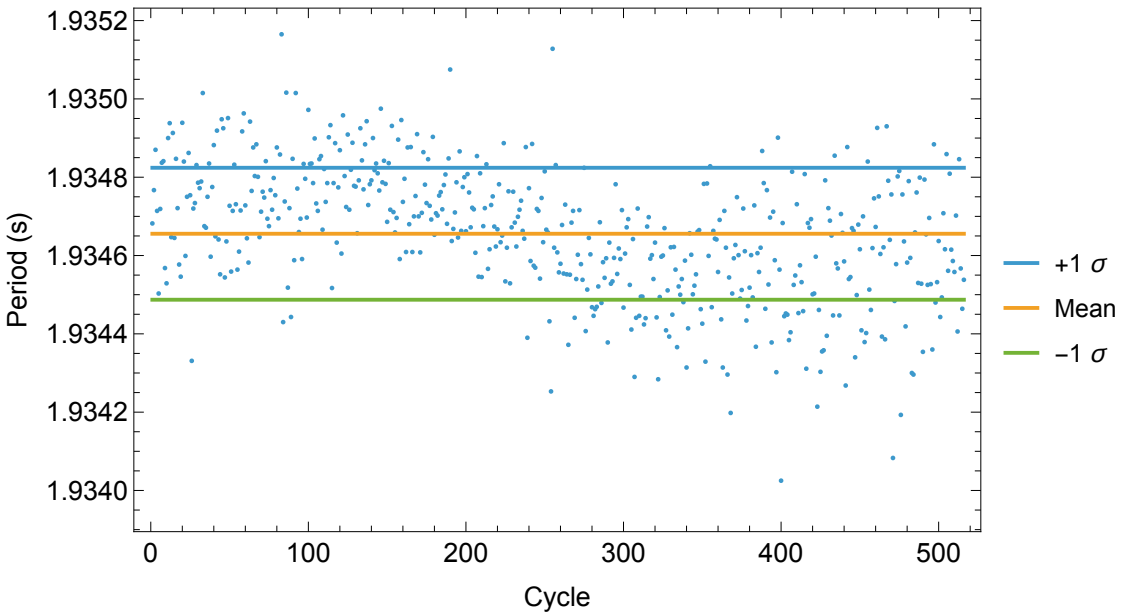


Further explorations

We can illustrate the ±1 σ range on the plot:

In[180]:=
```
TCompPlot = Show[{Tplot,
      Plot[{T + σT, T, T - σT},
    {cycle, 1, Length[data]}, PlotLegends → {"+1 σ", "Mean", "-1 σ"}]
  }]
```

Out[180]=



In[181]:=
```
Export["pendulum-period-plot.pdf", TCompPlot]
```

Out[181]=
```
pendulum-period-plot.pdf
```

We can also count the numbers with a particular range with the Select[ ] function. This function steps through the 'data' array and assigns the special symbol '#' to the current element of the array. (The '&' must also be included at the end of the comparison. See the 'Pure Functions' entry in the on-line help for more information.)

In[174]:=
```
within1 = Select[data, T - σT ≤ # ≤ T + σT &];
100. * Length[within1] / npts
```

Out[175]=
```
67.8295
```

In[176]:=
```
within2 = Select[data, T - 2 σT ≤ # ≤ T + 2 σT &];
100. * Length[within2] / npts
```

Out[177]=
```
95.5426
```

In[178]:=
```
within3 = Select[data, T - 3 σT ≤ # ≤ T + 3 σT &];
100. * Length[within3] / npts
```

Out[179]=
```
99.4186
```