

Homework_1_soln

January 29, 2025

1 PHYS 310 HW 1 Solutions

```
[1]: import numpy as np
import matplotlib.pyplot as plt
```

1.1 Hughes & Hayes: 2.2

Mean:

$$\mu = \frac{1}{N} \sum_i x_i$$

Standard Deviation:

$$\sigma_{N-1} = \sqrt{\frac{1}{N-1} \sum_i (x_i - \mu)^2}$$

Standard Error (Standard deviation of the mean):

$$\alpha = \frac{\sigma_{N-1}}{\sqrt{N}}$$

```
[2]: sensitive = np.asarray([5.33,4.95,4.93,5.08,4.95,4.96,5.02,4.99,5.24,5.25,5.
    ↪23,5.01]) #A/W

#(i)
mu_manual = sum(sensitive)/len(sensitive)
mu = np.mean(sensitive) #Two ways of doing the same thing

print('(i) The mean sensitivity of the photodiode is ',mu_manual, '= %.
    ↪5f'%(mu), 'A/W')
# The %.5f in a string followed by %(number) magic keeps 5 decimal places to
    ↪read more easily

#(ii)
sig_manual = np.sqrt(sum((sensitive - mu)**2)/(len(sensitive)-1))
sig = np.std(sensitive, ddof=1)
#REMEMBER TO USE ddof = 1, because we will only ever be able to take a sampling
    ↪of all the data there ever could be
```

```

print('(ii) The standard deviation of the sensitivity measurements is_
↳',sig_manual, '= %.5f'%(sig), 'A/W')

#(iii)
alpha = sig/np.sqrt(len(sensitive))
print('(iii) The standard error of the sensitivity measurements is ',alpha, '=_
↳%.5f'%(alpha), 'A/W')

#(iv) Our addition
#Now we know what precision to actually print .2f
print('(iv) The Sensitivity of the Photodiode is %.2f +/- %.2f A/W'%(mu,alpha))

```

(i) The mean sensitivity of the photodiode is 5.078333333333334 = 5.07833 A/W
(ii) The standard deviation of the sensitivity measurements is 0.14357977404617803 = 0.14358 A/W
(iii) The standard error of the sensitivity measurements is 0.04144791059787326 = 0.04145 A/W
(iv) The Sensitivity of the Photodiode is 5.08 +/- 0.04 A/W

1.2 H&H: 2.3

The standard error, or standard deviation of the mean, is given by Eq.(2.7):

$$\alpha = \frac{\sigma_{N-1}}{\sqrt{N}}$$

To decrease α by a factor of 10, the denominator must be increased by the same factor, which means that N must increase by a factor of 100. Translating to the described experiment, this means that data should be collected for 100 minutes (assuming that everything in the experiment is stable for that length of time).

1.3 Pendulum Problem

The standard deviation is an estimate of timing accuracy.

Experiment A: 12 sets of 10 swings gives a result of $T_{10} = 28.39 \pm \frac{0.04}{\sqrt{12}}$ seconds for timing for 10 swings. That is distributed to the 10 swings to give:

$$T_1 = \frac{28.39 \pm \frac{0.04}{\sqrt{12}}}{10} \text{ s}$$

$$T_1 = 2.839 \pm 0.001 \text{ s}$$

Experiment B: The single measurement of 120 swings is assumed to have the same timing accuracy, giving a result of $T_{120} = 340.61 \pm \frac{0.04}{\sqrt{1}}$ seconds for timing for 120 swings. That is distributed to the 120 swings to give:

$$T_1 = \frac{340.61 \pm \frac{0.04}{\sqrt{1}}}{120} \text{ s}$$

$$T_1 = 2.8384 \pm 0.0003 \text{ s}$$

Distributing a fixed uncertainty over many cycles in one measurement (Experiment B) is more precise than taking more measurements that compound effects from the uncertainty (Experiment A). It is often hard to take one measurement that captures many instances of the phenomenon we want.

```
[3]: print('Experiment A: %.3f +/- %.3f s'%(28.39/10,0.04/np.sqrt(12)/10))
      print('Experiment B: %.4f +/- %.4f s'%(340.61/120,0.04/120))
```

```
Experiment A: 2.839 +/- 0.001 s
Experiment B: 2.8384 +/- 0.0003 s
```

1.4 Estimates

Getting an intuitive idea of what the data should give you before you run it through statistical formulae is a good skill.

We can sort the data from smallest to largest, that gives us an idea of the range where the center value is probably close to the average. It goes from about 3.9s to 4.5s, spanning 0.6s so the middle should be around $3.9+0.3=4.2$ s. The median of the 15 measurements (7 values lower and 7 values higher) is 4.19365 s so that double-check worked well.

For the “rough and ready” estimate of standard deviation, the estimate of the mean is 0.3 s away from the ends and we’ll use 2/3 of that : 0.2s.

```
[4]: period = np.asarray([4.1075, 4.39831, 4.19365, 4.20259, 4.26921, 4.13037, 3.
    ↪97548, 4.51314, 4.01286,
    4.0101, 4.15578, 4.35153, 4.30801, 4.21082, 3.94315])

print('The sorted data from smallest to largest:',np.sort(period))
print('The median value is:',np.median(period))
print('The Estimate might be a mean of: %f and standard deviation of 0.2'%np.
    ↪median(period))
print('The actual mean: %f and standard deviation: %f'%(np.mean(period),np.
    ↪std(period, ddof =1)))
```

```
The sorted data from smallest to largest: [3.94315 3.97548 4.0101 4.01286
4.1075 4.13037 4.15578 4.19365 4.20259
4.21082 4.26921 4.30801 4.35153 4.39831 4.51314]
The median value is: 4.19365
The Estimate might be a mean of: 4.193650 and standard deviation of 0.2
The actual mean: 4.185500 and standard deviation: 0.164020
```

The Mean and Standard Deviation of this data is well estimated by the median and 2/3 of half of the range.

```
[5]: %load_ext version_information
      %version_information
```

```
[5]:
```

Software	Version
Python	3.12.8 64bit [Clang 14.0.6]
IPython	8.30.0
OS	macOS 15.2 arm64 arm 64bit
Wed Jan 29 16:33:02 2025 EST	