Introduction

This notebook will introduce measurements and statistics, and incorporate them into the Jupyter notebook environment. This lab will give you familiarity with some of the libraries to be used, as well as practical examples of working with data in Jupyter.

Exercise 1: Significant Figures

a) Write the following values to three significant figures:

300,000: 3.00E5

45.327: 45.3

π: 3.14

b) How many significant figures are in the following values:

1.01: 3

0.4375: 4

The counted number of students in this classroom: 2

1000.: 4

c) Perform the arithmetic with the correct significant digits in the final answer:

3.7 + 12 = 16

10 - 3.14159 = 7

100 + 1700 - 360 = 1.44E3

1.32e-7 + 4.2e-6 = 4.3E-6

15 × 6.37 = 96

1.6e-19 ÷ 9.109e-31 = 1.8e+11

$\frac{7}{16}$ = 0.4

Exercise 2: Statistics

The files All_Blue_giant_Stars.cvs and All_Blue_Stars.cvs contain the magnitudes of stars from within the Milkyway. The two data sets are selections of the same kind of stars (called Blue Horizontal Branch stars) but with different selection criteria (we call those cuts). In this exercise you will learn how to use statistics to infer inherent properties of the two data sets (this is EXACTLY what data science is).

a) Choose one file (doesn't matter which, but make sure you notate clearly in the markdown which you choose) and compute the mean $\bar{x}$ and standard deviation σ_x for different numbers of statistics. Do this for the first 10, 100, 1,000 and all lines.

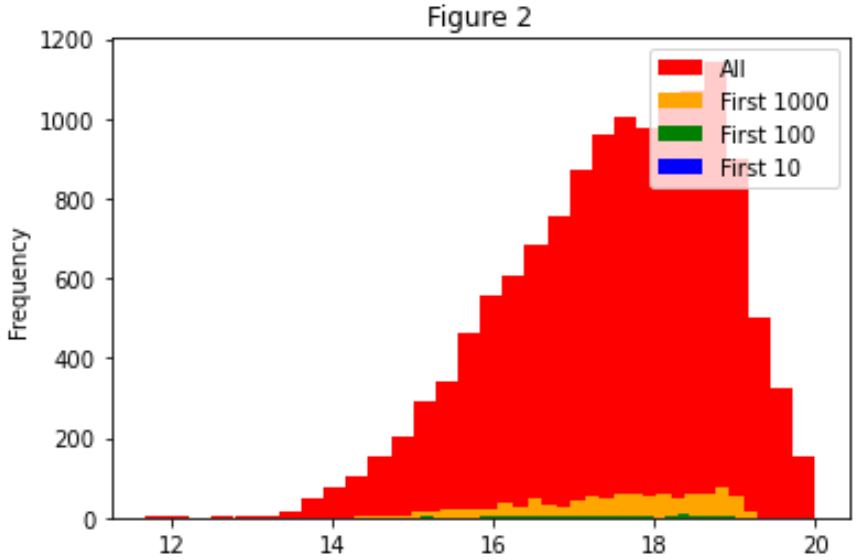
I am going to use All Blue Giant Stars data file

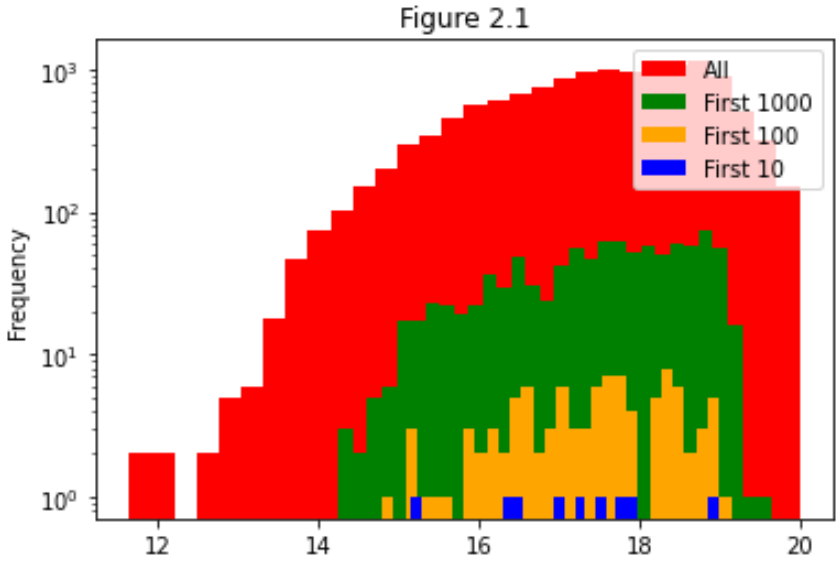
In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.odr as odr
data = pd.read_csv("/home/admin/Documents/School/ExPhys/Homewor
data = np.array(data)
print("Mean of first 10: ", np.mean(data[slice(9)]))
print("Mean of first 100: ", np.mean(data[slice(99)]))
print("Mean of first 1000: ",np.mean(data[slice(999)]))
print("Mean of all: ",np.mean(data))

print("Std. Deviation of first 10", np.std(data[slice(9)]))
print("Std. Deviation of first 100", np.std(data[slice(99)]))
print("Std. Deviation of first 1000", np.std(data[slice(999)]))
print("Std. Deviation of all", np.std(data))
```

```
Mean of first 10:  17.152835555555555
Mean of first 100:  17.379240000000003
Mean of first 1000:  17.468630040040036
Mean of all:  17.466830481001207
Std. Deviation of first 10 1.042027290814626
Std. Deviation of first 100 1.0265959358909806
Std. Deviation of first 1000 1.1448585620152465
Std. Deviation of all 1.330196324745447
```

b) Make a histogram of the data from your part a for the four different numbers of statistics, but place all of these in a single plot so they are easily comparable. Make sure to label the axes correctly, and use a different color for each histogram with a legend to denote them. If this is done right, it should resemble the lecture slides. Make two versions of this figure, one with a linear y-axis and one with a logarithmic y-axis. Place a label beneath each figure (e.g. Figure 2.1, etc), giving each a figure number so that you can refer to them by number in the discussion below.

In [3]:
```
data10 = data[slice(9)]
data100 = data[slice(99)]
data1000 = data[slice(999)]
plt.hist(data, bins=30, alpha=1, label='All', color = 'red')
plt.hist(data1000, bins=30, alpha=1, label='First 1000', color
plt.hist(data100, bins=30, alpha=1, label='First 100',color = '
plt.hist(data10, bins=30, alpha=1, label='First 10',color = 'bl
plt.ylabel('Frequency')
plt.title('Figure 2')
plt.legend(loc='upper right')

plt.show()
plt.hist(data, bins=30, alpha=1, label='All', color = 'red', lc
plt.hist(data1000, bins=30, alpha=1, label='First 1000',color =
plt.hist(data100, bins=30, alpha=1, label='First 100', color =
plt.hist(data10, bins=30, alpha=1, label='First 10',color = 'bl
plt.ylabel('Frequency')
plt.title('Figure 2.1')
plt.legend(loc='upper right')
plt.show()
```





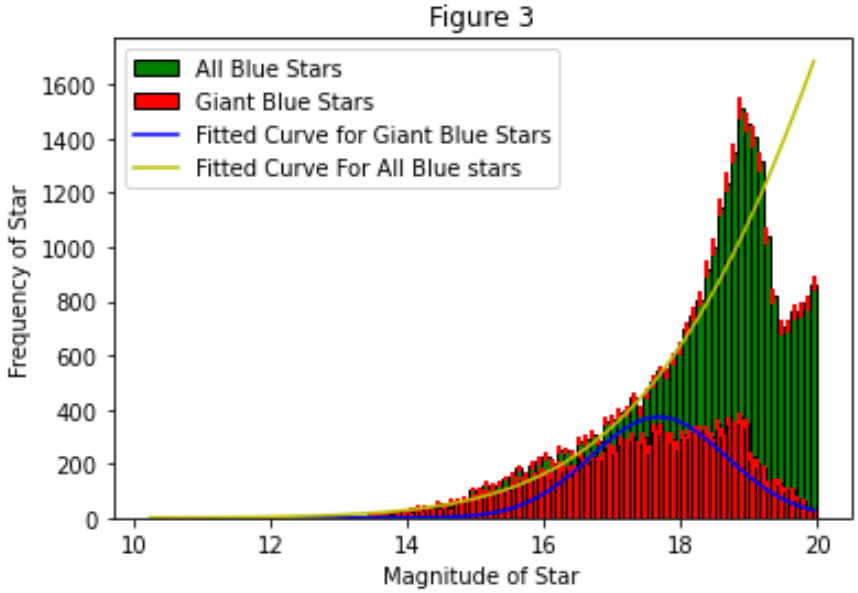
–

c) Now compute the mean x and standard deviation σ_x for all data in the second file. Make a histogram of these data showing the statistical error bar on each bin, and overlay the histogram for all data in the other file, also with statistical error bars (use colors, a legend, and a caption with figure number as in part b). Fit each of these two curves with a Guassian using the orthoganal distance regression (ODR) package. Make sure to report the best-fit values and uncertainties for each parameter as well as the χ^2/NDF.

In [5]:
```
data = pd.read_csv("/home/admin/Documents/School/ExPhys/Homewor
data = np.array(data)
data1 = pd.read_csv("/home/admin/Documents/School/ExPhys/Homewo
data1 = np.array(data1)
print("Mean of all: ",np.mean(data1))
print("Std. Deviation of all: " , np.std(data1))



counts1, bins1 = np.histogram(data1, bins=100)
bin_centers1 = (bins1[:-1] + bins1[1:]) / 2
errors1 = np.sqrt(counts1)
plt.errorbar(bin_centers1,counts1,yerr=errors1, color='red',
             elinewidth=2,linewidth=0,marker = '',markersize =
plt.hist(data1, bins=100, alpha=1, color='g', edgecolor='black'


counts, bins = np.histogram(data, bins=100)
bin_centers = (bins[:-1] + bins[1:]) / 2
errors = np.sqrt(counts)
plt.errorbar(bin_centers, counts,yerr=errors, color='red',
             elinewidth=2,linewidth=0,marker = '',markersize =
plt.hist(data, bins=100, alpha=1, color='r', edgecolor='black',

errors = np.where(errors == 0, np.nan, errors)
errors1 = np.where(errors1 == 0, np.nan, errors1)


def gaussian_func(B, x):
    """ Gaussian model: B[0] = amplitude, B[1] = mean, B[2] = s
    return B[0] * np.exp(-0.5 * ((x - B[1]) / B[2])**2)
gauss = odr.Model(gaussian_func)

noised_data = odr.RealData(bin_centers[40:95], counts[40:95], s
initial_guess = [max(counts), np.mean(bin_centers), np.std(bin_
regressed_model = odr.ODR(noised_data,gauss,beta0=initial_guess
regressed_output = regressed_model.run()
fitted_params = regressed_output.beta
param_sd = regressed_output.sd_beta
reduced_chi_squared = str(round(regressed_output.res_var,3))
fitted_curve = gaussian_func(fitted_params,bin_centers)
reduced_chi_squared = str(round(regressed_output.res_var,3))
print("Normalized Chi Squared Value for Giant Blue Stars" , red
plt.plot(bin_centers,fitted_curve,'b',label='Fitted Curve for G



noised_data1 = odr.RealData(bin_centers1[30:100], counts1[30:10
initial_guess1 = [max(counts1), np.mean(bin_centers1), np.std(b
regressed_model1 = odr.ODR(noised_data1,gauss,beta0=initial_gue
```

```
regressed_output1 = regressed_model1.run()
fitted_params1 = regressed_output1.beta
param_sd1 = regressed_output1.sd_beta

reduced_chi_squared1 = str(round(regressed_output1.res_var,3))
fitted_curve1 = gaussian_func(fitted_params1,bin_centers1)
reduced_chi_squared1 = str(round(regressed_output1.res_var,3))
print("Normalized Chi Squared Value for All Blue Stars" , reduc
print("Fitted parameters for Giant Blue Stars\n","Amplitude: ",
print("Fitted parameters for All Blue Stars\n","Amplitude: ", f
plt.plot(bin_centers1,fitted_curve1,'y',label='Fitted Curve For
plt.legend()
plt.title('Figure 3')
plt.xlabel('Magnitude of Star')
plt.ylabel('Frequency of Star')
plt.show()
```

```
Mean of all:  18.235562282547562
Std. Deviation of all:  1.294986512753307
Normalized Chi Squared Value for Giant Blue Stars 0.533
Normalized Chi Squared Value for All Blue Stars 0.351
Fitted parameters for Giant Blue Stars
 Amplitude:  372.3877049109895  Frequency:  17.67862247645448
2  Offset:  0.9930571885532568
Fitted parameters for All Blue Stars
 Amplitude:  4193.7434097396135  Frequency:  24.3949966963104
54  Offset:  3.292787395143814
```



How do the means and standard deviations compare? Why is one larger than the other. Is one data set cleaner than the other, and if so which and how can you tell? Write a short

paragraph answering all of these questions, and comment on how you use statistics to infer properties of your data, and reference your figures by the numbers you assigned.

The mean magnitude of all blue stars is higher then the mean magnitude of giant blue stars. The standard deviation of all blue stars is lower then the standard deviation of giant blue stars. For fitting to a gaussian curve, the set of all blue stars is a cleaner data set because it has a lower chi-squared with having the same degrees of freedom as the other data set. This is corroborated by viewing figure 3, the distribution of all stars in green more closely mirrors a gaussian distribution then the distribution of giant blue stars in red. Furthermore, by taking a look at the ODR-generated gaussian curve, the yellow line correlates with its data set better then the blue line does with its data.

Exercise 3: Error Analysis

The magnetic field on the axis of a coil of wire can be found using the Biot-Savart law. The coil considered here has 100 turns. You measure the diameter of the coil to be $d = 10.3 \pm 0.01$ cm. You don't have an ammeter, but you measure the resistance of the coil to be $R = 8.3 \pm 0.4\Omega$, and you measure the voltage as $V = 12.7 \pm 0.2$ V.

a) Compute the strength of the magnetic field on the coil axis 5.0 $\pm$ 0.1 cm above the coil, and give the uncertainty.

First we need to find the amperage using

$$V = IR \rightarrow \frac{V}{R} = I$$

We are given the voltage ($12.7 \pm 0.2V$) and the resistance ($8.3 \pm 0.4\Omega$), and using this we can find the current, and uncertainty of current.

$$I = \frac{12.7 \pm 0.2V}{8.3 \pm 0.4\Omega}$$

$$\Delta I = \sqrt{\left(\frac{\partial I}{\partial V}\Delta V\right)^2 + \left(\frac{\partial I}{\partial R}\Delta R\right)^2}$$

$$I = 1.53 \quad = \sqrt{((1/R)\cdot \Delta V)^2 + \left(\left(-(V/R^2)\right)\cdot \Delta R\right)^2} \quad I = (1.53 \pm 0.08)$$
$$= 0.0775779185$$
$$= 7.75779185 \times 10^{-2}$$
$$= 8 \times 10^{-2}$$

$$I = 1.53 \pm 8 \times 10^{-2}$$

With the current, we can use the Biot-Savart law for coils to derive the magnetic field

$$\Delta B = \sqrt{\left(\frac{\partial B}{\partial I}\Delta I\right)^2 + \left(\frac{\partial B}{\partial r}\Delta r\right)^2 + \left(\frac{\partial B}{\partial z}\Delta z\right)^2}$$

$$B = 0.0588831971 \quad = \sqrt{\left(\left(r^2 * (2r^2+z^2)^{(}-3/2)\right)\cdot \Delta I\right)^2 + \left(\left((2 * I * r * (2r^2+z^2)\right.\right.}$$
$$= 0.011977196$$
$$= 1.1977196 \times 10^{-2}$$
$$= 1 \times 10^{-2}$$

$$B = (6 \pm 1) \times 10^{-2}$$

Type *Markdown* and LaTeX: α^2

b) You measure the magnetic moment of a permanent magnet at this location in the B field by measuring the magnetic torque via

mechanical means, $\vec{\tau} = \vec{\mu} \times \vec{B}$, you'll make similar measurements in the magnetic moment lab. At an angle of $\theta = 45 \pm 1$ degrees, you measure the torque to be $\tau = 0.051 \pm 0.020$ Nm by averaging 10 measurements. What is the magnetic moment μ of the permanent magnet, including the

uncertainty?

Using the angle version of the cross product:

$$\vec{\tau} = \vec{\mu} \times \vec{B} \rightarrow |\vec{\mu}||\vec{B}|\sin(\theta)\vec{n} = \vec{\tau}$$

Where $\vec{n}$ is the unit vector orthogonal to $\vec{\mu}$ and $\vec{B}$. We don't care about direction, so we can rewrite this, with the values we derived above and values given, as the following

$$0.051 \pm 0.20 \text{ Nm} = \sin(45 \pm 1)(6 \pm 1)^{-2}|\mu|$$

And solving for μ, the magnetic moment, we can rearrange this:

$$\frac{0.051 \pm 0.20 \text{ Nm}}{\sin(45 \pm 1)(6 \pm 1)^{-2}} = |\mu|$$

Solving this with error propagation, we have the following:

$$\Delta m = \sqrt{\left(\frac{\partial m}{\partial T}\Delta T\right)^2 + \left(\frac{\partial m}{\partial a}\Delta a\right)^2 + \left(\frac{\partial m}{\partial B}\Delta B\right)^2}$$

$m = 1.0178844301$

$$= \sqrt{\left(\left(B * sin(a)/(sin(a) * B)^2\right) \cdot \Delta T\right)^2 + \left(\left(-(T/(sin(a) * B)^2 *\right.\right.}$$

$$= 0.7642771649$$

$$= 7.642771649 \times 10^{-1}$$

$$= 8 \times 10^{-1}$$

$$m = (1 \pm 0.8)$$

c) This experiment is a good example of a combination of statistical and systematic uncertainty. Repeated measurements of the torque will reduce the uncertainty on the torque, but not on the magnetic field or angle. Thus, the torque exhibits statistical uncertainty while the errors on the B field and angle are systematic. What will the uncertainty on the torque be by making 100 measurements? What about the uncertainty on the magnetic moment?

The uncertainty in the torque will be reduced with more measurements by a factor of $\frac{1}{\sqrt{N}}$. With N=10, it measures to be ± 0.02 Nm, and if we compare the ratio of N=10 and N=100

$$\frac{\frac{1}{\sqrt{100}}}{\frac{1}{\sqrt{10}}} = \frac{\tau_{100}}{0.02}$$

And therefore $\tau_{100} = 0.006324$

The uncertainty of the magnetic moment is systematic, and with more measurements it will remain the same.