6.10 Exercises

1. Plot the function $y = 3x^2$ for $-1 \leq x \leq 3$ as a continuous line. Include enough points so that the curve you plot appears smooth. Label the axes x and y.

2. Plot the following function for $-15 \leq x \leq 15$:

   $$y = \frac{\cos x}{1 + \frac{1}{5}x^2}$$

   Include enough points so that the curve you plot appears smooth. Draw thin gray lines, one horizontal at $y = 0$ and the other vertical at $x = 0$. Both lines should appear behind the function. Label the axes x and y.

3. Plot the functions $\sin x$ and $\cos x$ *vs* x on the same plot with x going from $-\pi$ to π. Make sure the limits of the x-axis do not extend beyond the limits of the data. Plot $\sin x$ in the color orange and $\cos x$ in the color green and include a legend to label the two curves. Place the legend within the plot, but such that it does not cover either of the sine or cosine traces. Draw thin gray lines behind the curves, one horizontal at $y = 0$ and the other vertical at $x = 0$.

4. Create a data file with the data shown below.

   (a) Read the data into the Python program and plot t *vs.* y using circles for data points with error bars. Use the data in the `dy` column as the error estimates for the y data. Label the horizontal and vertical axes "time (s)" and "position (cm)." Create your plot using the `fig, ax = plt.subplots()` syntax.

   (b) On the same graph, plot the function below as a smooth line. Make the line pass *behind* the data points.

   $$y(t) = \left(3 + \frac{1}{2}\sin\frac{\pi t}{5}\right) t e^{-t/10}$$

```
Data for Exercise 4
Date: 16-Aug-2013
Data taken by Lauren and John

   t        y      dy
 1.0     2.94    0.7
 4.5     8.29    1.2
 8.0     9.36    1.2
```

```
    11.5    11.60    1.4
    15.0     9.32    1.3
    18.5     7.75    1.1
    22.0     8.06    1.2
    25.5     5.60    1.0
    29.0     4.50    0.8
    32.5     4.01    0.8
    36.0     2.62    0.7
    39.5     1.70    0.6
    43.0     2.03    0.6
```

5. Use matplotlib's function `hist` along with NumPy's functions `random.rand` and `random.randn` to create the histogram graphs shown in Fig. 9.2. See §9.2 for a description of NumPy's random number functions.

6. The data file below shows data obtained for the displacement (position) *vs.* time of a falling object, together with the estimated uncertainty in the displacement.

```
Measurements of fall velocity vs time
Taken by A.P. Crawford and S.M. Torres
19-Sep-13
time (s)     position (m)     uncertainty (m)
 0.0              0.0              0.04
 0.5              1.3              0.12
 1.0              5.1              0.2
 1.5             10.9              0.3
 2.0             18.9              0.4
 2.5             28.7              0.4
 3.0             40.3              0.5
 3.5             53.1              0.6
 4.0             67.5              0.6
 4.5             82.3              0.6
 5.0             97.6              0.7
 5.5            113.8              0.7
 6.0            131.2              0.7
 6.5            148.5              0.7
 7.0            166.2              0.7
 7.5            184.2              0.7
 8.0            201.6              0.7
 8.5            220.1              0.7
 9.0            238.3              0.7
 9.5            256.5              0.7
10.0            275.6              0.8
```

(a) Use these data to calculate the velocity and acceleration (in a Python program `.py` file), together with their uncertainties propagated from the displacement *vs.* time uncertainties. Be sure to calculate time arrays corresponding the midpoint in time between the two displacements or velocities for the velocity and acceleration arrays, respectively.

(b) In a single window frame, make three vertically stacked plots of the displacement, velocity, and acceleration *vs.* time. Show the error bars on the different plots. Make sure that the time axes of all three plots cover the same range of times (use `sharex`). Why do the relative sizes of the error bars grow progressively greater as one progresses from displacement to velocity to acceleration?

7. Starting from the code that produced Fig. 6.9, write a program using the mixed-OOP syntax introduced in §6.4.1 to produce the plot below. To create this plot, you will need to use the `sharex` feature introduced in §6.4.1, the `subplots_adjust` function to adjust the space between the two subplots, and the LaTeX syntax introduced in §6.6 to produce the math and Greek symbols. To shorten your program, try to use `for` loops where there is repetitive code.