

# PHYS 210, Assignment 8

Create a new directory somewhere in your home directory with the name `yourusername_assignment_8` to store the files you will create for this assignment. To hand the assignment in, copy the directory with your results to `/home2/phys210/yourusername/`. Make sure it's there and has the right permissions (read and execute for everyone, write for you).

## 1 Data fitting and plotting

1. Read our trusty sample data. Specifically, read the `x`, `y`, and `yerror` columns:

```
# N x y xerror yerror
1 1.24 3.14 0.012 0.52
2 1.26 3.89 0.012 0.52
3 1.28 4.04 0.012 0.53
4 1.30 4.33 0.012 0.53
5 1.33 4.67 0.012 0.54
6 1.35 5.08 0.012 0.54
```

```
# Data from 10:15:39 on September 12 2016
```

2. Write a function that takes variables  $b$ ,  $c$ , and  $x$  as arguments and calculates  $y = bx + c$ , i.e., a linear function.
3. Use `scipy.optimize.curve_fit` to fit the data you loaded in step 1 with the function you defined in step 2. Read the lecture notes (week 4.1) and the `scipy` reference ([http://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve\\_fit.html](http://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html)) to learn what arguments `scipy.optimize.curve_fit` takes. Make sure to include the `y` errors on the data points. Assume that the errors are absolute errors, i.e., you need to pass `absolute_sigma=True`.
4. Plot the data (including the `y` error bars) and the linear fit. Save the plot as `linear_fit.pdf`.

Put your commented code in a file called `linear_fit.py`.

## 2 Scatter plots and histograms

1. Create random points drawn from a bivariate normal distribution using, e.g., `numpy.random.multivariate_normal`. Read up on how to use this function ([http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.multivariate\\_normal.html](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.multivariate_normal.html)).
2. Make a scatter plot of your random points. Save the plot as `bivariate_scatter_plot.pdf`.

3. Make a histogram plot for the x and y coordinates. Save it as `bivariate_hist_plot.pdf`.
4. Make a single figure, showing the scatter plot in the top left corner, the x histogram in the bottom left corner, and the y histogram in the top right corner. This should look something like figure 1.

Put your commented code in a file called `scatter_hist_plot.py`.

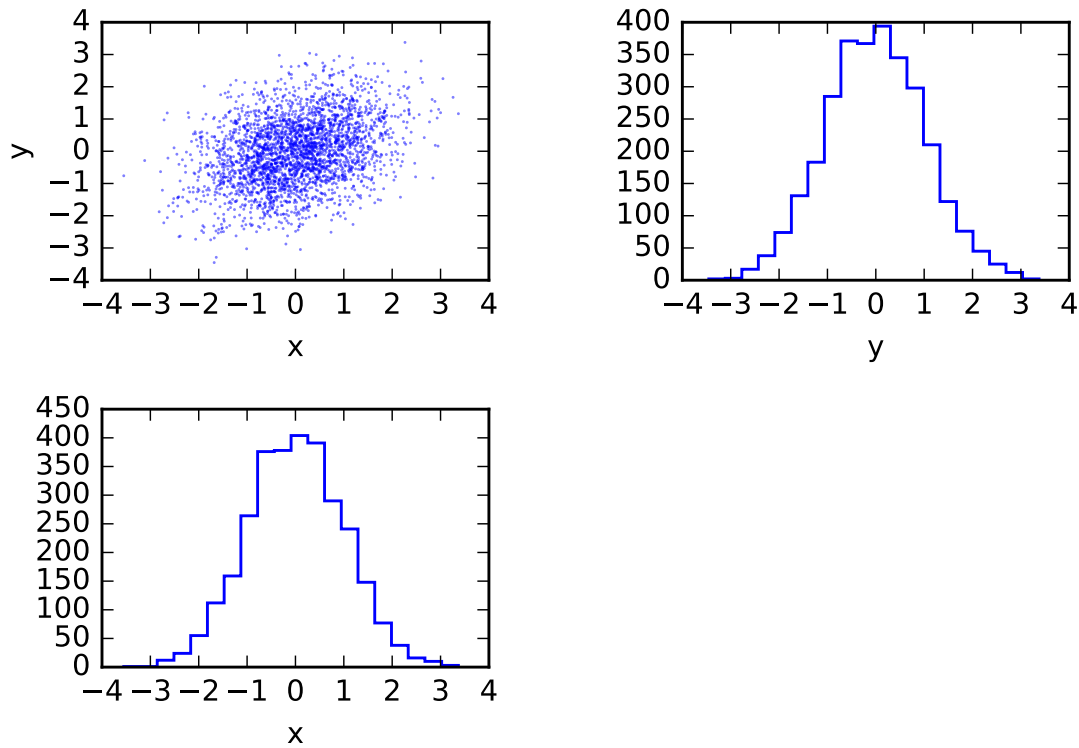


Figure 1: Random points drawn from a bivariate normal distribution.