Download the template Jupyter notebook `HW5.ipynb` from Canvas and work from that template. You are free to use whatever packages you like.

1. **CODING:** In this problem, you will learn how to perform non-linear regression and also see how more data improves the statistical quality of the data set. There are 10 data files `run[0-9].dat` that contain hypothetical particle-physics spectral data (energy vs counts) of a search for a new particle called the *Riggs boson*. The data were taken independently in 10 distinct runs. The columns are 1) energy E (GeV) and 2) counts N (number of events in that energy bin).

```
1.0    96
2.0    99
3.0   111
4.0   124
5.0   105
6.0   102
  ...
```

   A signal of a new particle consists of a gaussian on top of a smooth background. Model the spectrum with a function given by:

$$N(E) = a + bE + cE^2 + Ae^{-(E-E_{\rm Riggs})^2/(2\sigma_E^2)} \tag{1}$$

   where $a, b, c, A, E_{\rm Riggs}, \sigma_E$ are 6 fit parameters. The first three terms make up the background, the last term represents the signal. The term σ_E represents the intrinsic energy width of the particle and is expected to be in the range 1 – 5 GeV. Theory predicts that the Riggs should exist somewhere in the 40 – 90 GeV range.

   (a) In this first part, use only the first data file `run0.dat`. Use `scipy.optimize.curve_fit` to determine the best-fit values of the parameters and the covariance matrix. Can you claim a detection of the Riggs boson? Justify your answer.

   (b) Since the 10 datasets are independent, the counts can simply be added to produce a dataset with higher statistical quality. Successively add the counts in `run1.dat`, `run2.dat`, and so on and repeat the fits. After which run can you claim a 5σ detection of the Riggs?

   (c) Using data from all 10 runs, measure the energy $E_{\rm Riggs}$ and uncertainty of the the Riggs boson. What is the final significance (in units of σ) of the detection?

2. **CODING:** Markov Chain Monte Carlo (MCMC) is a powerful tool for sampling the posterior distribution of complicated models. The best way to learn how to implement this is through an example,. In this problem, you will redo parts of HW4 Problem 2) using the Metropolis-Hastings MCMC algorithm.

   Recall that we were tasked to model the relation between the number of satellites that reenter the Earth $N_{\rm reentry}$ and the average number of sunspots $N_{\rm sunspot}$ in a given year. The data are provided in `ReentryData.dat`. We used the following model:

$$N_{\rm reentry} = a + bN_{\rm sunspot} \tag{2}$$

where a and b are the fitting parameters. Using the analytic solution to the MLE, we saw that the posterior distribution is characterized by the means and covariance matrix given by:

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 13.11 \\ 0.110 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 1.84 & -0.0141 \\ -0.0141 & 0.000169 \end{pmatrix} \tag{3}$$

(a) Construct a Markov Chain as follows:

i. Initialization: Start with the best-fit parameters $\vec{x}_{\rm old} = (13.11\ 0.110)$. Calculate the value of the likelihood $\mathscr{L}(\vec{x}_{\rm old})$ at this point. The step sizes $\delta\vec{x}$ must be chosen appropriately such that

ii. For each iteration:

- Generate a candidate random step from $\vec{x}_{\rm old}$ by drawing a pair of random numbers $\vec{r}$ from a normal distribution $\mathscr{N}(0, \delta\vec{x})$. Your candidate point is $\vec{x}_{\rm new} = \vec{x}_{\rm old} + \vec{r}$.
- Calculate the ratio $R = \mathscr{L}(\vec{x}_{\rm new})/\mathscr{L}(\vec{x}_{\rm old})$.
- Determine whether to accept or reject the candidate step:
  - If $R \geq 1$, take the proposal step (accept).
  - else if $R < 1$, then draw a random number U from a uniform distribution between 0 and 1.
    - If $U < R$, take the proposal step (accept).
    - else if $U \geq R$, do not take the step (reject).

Do 10^5 iterations and choose $\delta\vec{x}$ such that $\sim 50\%$ $(30 - 70\%)$ of the candidate steps are accepted. The output should be a Markov chain (list) of accepted steps $\{\vec{x}\}$.

(b) Using the Markov chain from above, calculate the sample means $(\bar{a}\ \bar{b})$ and covariance matrix, and check that the values are nearly identical to those from the analytic MLE.

(c) Install the plotting package `corner` in your Anaconda installation.

```
conda install corner
```

If you run into problems, see:

`http://corner.readthedocs.io/en/latest/install.html`

This package allows you to easily visualize the variance and covariance of an MCMC chain. A quick-start guide can be found here:

`http://corner.readthedocs.io/en/latest/pages/quickstart.html`

Use this package to visualize the Markov chain, which is just a sampling of the bivariate gaussian characterized by the means and covariance matrix from Equation (3).

3. **CODING, EXTRA CREDIT:** Repeat Problem 2) for the non-linear regression problem in Problem 1). Specifically, generate a 6-parameter Markov Chain, solve for the means and covariance matrix, and plot the posterior distributions with `corner`.