Download the template Jupyter notebook `HW6_Template.ipynb` from Canvas and work from that template. As you will see, we are introducing `pandas` in this assignment. Pay close attention to the hints in the template notebook. You can also use `numpy`, `pandas`, `scipy`, and `matplotlib`, but do not use any fitting packages such as `scipy.optimize.curve_fit` to solve for the parameters and uncertainties. As the fitting functions are linear, the solutions can be obtained entirely with `numpy`'s linear algebra operations.

1. **CODING:** (3 pts) The file `SunspotNumber.dat` contains data on the number of sunspots observed each day since January 1, 1818 through September 30, 2020. Columns are 1) year, 2) month, 3) day, and 4) number of sunspots observed (`-1` indicates that no data were taken on that day – i.e., the data is missing).

```
...
1819 03 21    0
1819 03 22   30
1819 03 23   20
1819 03 24   -1
1819 03 25   -1
1819 03 26   17
 ...
```

   (a) Calculate the monthly mean of the sunspot number. Ignore uncertainties. The output should be a `pandas` Series indexed by (year, month) with the mean number of sunspots in each month.

   (b) Repeat part a) for the yearly mean. This will be used in the next problem.

2. **CODING:** The file `SatelliteReentry.dat` contains the number of satellites in low-Earth orbits that reentered and burned in the Earth's atmosphere from 1969 to 2004.

```
1969   26
1970   25
1971   19
1972   12
1973   14
1974   21
 ...
```

   The goal of this problem is to see if there is a relation between solar activity (using sunspot number as a proxy) and the number of satellites that reenter the atmosphere. We will model the relation with a straight line given by:

$$N_{\text{reentry}} = a + bN_{\text{sunspot}} \tag{1}$$

   where a and b are the fitting parameters.

   (a) (3 pts) Taking the gaussian standard deviation of N_i to be $\sqrt{N_i}$, determine the maximum likelihood estimate of a and b.

(b) (3 pts) Determine their standard deviations σ_a and σ_b, and their covariance σ_{ab}.

(c) (1 pt) Plot the data with uncertainties and the best-fit model superimposed. Make the plot look nice - i.e., label axes, use legible markers, colors, etc.

(d) (2 pts) Calculate the χ^2 values on a grid of a and b and use `matplotlib.pyplot.contour` to plot the contours of constant χ^2 values for $\chi^2 = \chi^2_{\rm min} + 2.30, 6.17$, and 11.8. Make sure your grid and contour plot is large enough to show all three contour levels.

3. **CODING:** It is common practice in data analysis to demonstrate that your estimator produces unbiased results - i.e., the true parameters values are *on average* recovered. Do the following to show that your estimators for a and b above are unbiased.

(a) (3 pts) Write a function that draws N points $\{x_i, y_i\}$ from the model,

$$y = a + bx \tag{2}$$

where x_i is drawn from a uniform distribution in the range $[x_{\rm min}, x_{\rm max}]$ and y_i is drawn from a gaussian distribution centered on $y = a + bx_i$ with fixed standard deviation σ. Also, write a function that fits for a, b given arrays x, y.

(b) (3 pts) Using $N = 90$, $x_{\rm min} = 18, x_{\rm max} = 38, a = 200, b = 10$, and $\sigma = 10$, generate 10,000 realizations of the dataset, determine the MLE of a, b, σ_a, σ_b for each realization, and calculate the mean of a and b and their corresponding uncertainties. Show that your estimators for a and b are unbiased - i.e, the true value falls within several standard deviations from the means of a and b.

(c) (2 pts) Plot the histogram of χ^2 values of the 10,000 realization from above, and overplot an appropriately scaled $p(\chi^2; \nu)$ distribution for $\nu = 90 - 2 = 88$ degrees of freedom.