MOONRISE OVER NORTHERN LIGHTS Northern Alaska, March 2011 (Ben Hattenbach/Los Angeles, California)

Statistics and Plotting

COMP 364 - Lecture 19
March 7th, 2012
Mathieu Perreault

A faster way to read in CSV files

```
import csv
r = csv.reader(fh,delimiter=’,’)
```

```
r = csv.reader(open(sys.argv[1]), delimiter='\t')
for row in r:
  # Items are accessed at row[0], row[1], etc...
```

A reader is an iterable object that produces lists containing the data in a given row.

Correlation

- Indicates a relation of dependence between two sets of data (or random variables)

$$\rho_{X,Y} = \operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y},$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{(n-1)s_x s_y} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2 \sum_{i=1}^{n}(y_i-\bar{y})^2}},$$

- Sometimes an interesting measure to include on a plot

- See http://docs.scipy.org/doc/numpy/reference/generated/numpy.corrcoef.html

Line of best fit

- Polynomial curve that best fits the dataset. Depending on the experiment being performed, different degree polynomials apply.

- Be careful of overfitting!

```
import numpy as np
import matplotlib.pyplot as plt

# Linear regression
params = np.polyfit(x, y, 1) # Pick your degree
yp = np.polyval(params, x)

plt.plot(x,y,’g.’) # Plot your real data
plt.plot(x,yp)     # Plot the best fit
```

Standard Deviation and others

- Measure the variability of your dataset with respect to the mean.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \quad \text{where} \quad \mu = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

- numpy can help you calculate this:

```
import numpy as np

stddev = np.std(y)
```

- Also available in numpy: `var, average, median, mean`
  - http://docs.scipy.org/doc/numpy/reference/routines.statistics.html

Changing the limits of your axes

- matplotlib allows you to change the limits of your plot
- http://matplotlib.sourceforge.net/api/pyplot_api.html#matplotlib.pyplot.xlim

```
import matplotlib.pyplot as plt

plt.xlim(-1.0, 1.0)
plt.ylim(0, 50.0)

plt.xlim(xmax=1.0) # Also possible
```

Legend vs. Text

- matplotlib allows you to put up a legend associated with a line

```
import matplotlib.pyplot as plt

plt.plot(x,y)
plt.legend([‘sin(x)’])
```

- But you can also put up a text label, if appropriate. Choose location based on the scale of your plot, and available whitespace (don’t put over data!)

```
plt.plot(x,y)
plt.text(1.0, 5.0, ‘Correlation: 0.3’)
# Example at x=1.0, y=5.0
```

Horizontal and vertical lines

- matplotlib allows you to add vertical and horizontal lines, pretty easily.

```
matplotlib.pyplot.axhline(y=0, xmin=0, xmax=1, hold=None, **kwargs)
matplotlib.pyplot.axvline(x=0, ymin=0, ymax=1, hold=None, **kwargs)
```

- The min and max bounds are fractions of the axis. For example, setting ymax=0.5 on a vertical line will make it go up the the middle of the screen.