ASTRONOMY 630

Numerical and statistical methods in astrophysics

CLASS NOTES

Spring 2024

Instructor: Jon Holtzman

1 Introduction

What is this class about?

What is statistics? Read and discuss Feigelson & Babu 1.1.2 and 1.1.3.

The use of statistics in astronomy may not be as controversial as it is in other fields, probably because astronomers feel that there is basic physics that underlies the phenomena we observe.

What is data mining and machine learning? Read and discuss Ivesic 1.1

Astroinformatics

What are some astronomy applications of statistics and data mining? Classification, parameter estimation, comparison of hypotheses, absolute evaluation of hypothesis, forecasting, finding substructure in data, finding corrrelations in data

Some examples:

- Classification: what is relative number of different types of planets, stars, galaxies? Can a subset of observed properties of an object be used to classify the object, and how accurately? e.g., emission line ratios

- Parameter estimation: Given a set of data points with uncertainties, what are slope and amplitude of a power-law fit? What are the uncertainties in the parameters? Note that this assumes that power-law description is valid.

- Hypothesis comparison: Is a double power-law better than a single power-law? Note that hypothesis comparisons are trickier when the number of parameters is different, since one must decide whether the fit to the data is sufficiently better given the extra freedom in the more complex model. A simpler comparison would be single power-law vs. two constant plateaus with a break at a specified location, both with two parameters. But a better fit does not necessarily guarantee a better model!

- Absolute evaluation: Are the data consistent with a power-law? Absolute assessments of this sort can be more problematic than hypothesis comparisons.

- Forecasting of errors: How many more objects, or what reduction of uncertainties, would allow single and double power-law models to be clearly distinguished? Need to specify goals, and assumptions about data. Common need for observing proposals, grant proposals, satellite proposals ...

- Finding substructure: 1) spatial structure, 2) structure in distributions, e.g., separating peaks in RVs or MDFS, 3) given multi-dimensional data set, e.g. abundance patterns of stars, can you identify homogeneous groups, e.g. chemical tagging?

- Finding correlations among many variables: given multi-dimensional data, are there correlations between different variables, e.g., HR diagram, fundamental plane, abundance patterns

More examples in Ivesic et al 1.1.

Note general explosion of the field in the last decade (e.g., CCA and other institutes). Familiarity with these topics is critical, and mastery even better.

Clearly, numerical/computational techniques are an important component of astrostatistics. Previous course, ASTR 575, attempted (with poor success?) to combine these....

Initial anticipation (!) of ASTR630 class topics:

- Intro: usage of data/statistical analysis in astronomy
- Basic probability and statistics. Conditional probabilities. Bayes theorem
- Statistical distributions in astronomy. Point estimators: bias, consistency, efficiency, robustness
- Random number generators. Simulating data. Handling observational data: Correlated and uncorrelated errors. Forward modeling.
- Fitting models to data. Parametric vs non-parametric models. Frequentist vs Bayesian. Least squares: linear and non-linear. Regularization.
- Bayesian model analysis. Marginalization. MCMC.
- Determining uncertainties in estimators/parameters: bootstrap and jackknife.
- Hypothesis testing and comparison
- non-parametric modeling
- Clustering analysis. Gaussian mixture models. Extreme deconvolution. K means.
- Dimensionality reduction: principal components, tSNE, etc.
- Deep learning and neural networks
- ?Fourier analysis (probably not)

Logistics: Canvas, class notes, grading TBD (homework, no exams?)
Resources:

- Ivesic et al : Statistics, Data Mining, and Machine Learning in Astronomy (2014), with Python
- Bailer-Jones: Practical Bayesian Inference (2017), with R
- Feigelson & Babu: Modern Statistical Methods for Astronomy: With R Applications (2012)
- Robinson : Data Analysis for Scientists and Engineers (2016)
- Press et al., Numerical Recipes (3rd edition, 2007)

Web sites:

- California-Harvard Astrostatistics Collaboration
- Center for Astrostatistics at Penn State: note annual summer school in Astrostatistics, also Astronomy and Astroinformatics portal
- International Computing Astronomy Group

2 Probability, basic statistics, and distributions

Why study probability?

- Probability and statistics: in many statistical analyses, we are talking about the probability of different hypotheses being consistent with data, so it is important to understand the basics of probability.
- In many circumstances, we describe variation in data (either intrinsic or experimental) with the probability that we will observe different outcomes

Different approaches to probability: classical, frequentist, subjective, axiomatic (see here):

- Classic Theory - Calculation of the ratio of favourable to total outcomes. Does not involve experiment. Computed by counting N_E events of N total possible events, where all outcomes are equally likely. All outcomes must be equally likely. Cannot handle an infinite number of possible outcomes. Example: coins, dice, cards.
- frequentist: Perform an experiment n times, testing for the occurence of event E, and define:
$$P(E) = \lim_{n\to\infty} n(E)/N$$
We must estimate from a finite number of trials. We postulate that $n(E)/n$ approaches a limit as n goes to infinity. Not possible for single events! Example: a loaded die.
- subjective: judgement based on intuition; a combination of rational statistics, personal observations, and, potentially, superstition. Easily biased by non-rational use of data. In real-world situations, often all we have! Example: weather.

Axiomatic approach: provide a logically consistent set of axioms to work with probabilities, however they are determined. Define a random experiment H (with nondeterministic outcomes), a sample description space Ω (the set of all outcomes), and an event E (a subset of the sample description space which satisfies certain constraints). Consider the following definitions of axiomatic theory:

For any outcome, E,

$$0 < P(E) < 1$$

$$P(E') = 1 - P(E)$$

where E' is the complement of E (i.e., *not* E). For mutually exclusive (no overlap) outcomes **only**:

$$P(A \text{ or } B) \equiv P(A \cup B) = P(A) + P(B)$$

If E_i are the set of mutally exclusive and exhaustive outcomes (covers all possibilities), then:

$$\Sigma P(E_i) = 1$$

These imply, for non mutally exclusive outcomes:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

where $P(A \cap B) \equiv P(A \text{ and } B) \equiv P(A, B)$ in different notations, used in different books. Note graphical representation.

If events A and B are *independent*, i.e., one has no effect on the probability of the other, then

$$P(A \cap B) = P(A)P(B)$$

This serves as a definition of independence. Note that if more than two conditions are considered (e.g., A,B,C), then independence requires that all pairs of conditions must be independent.

Caution : addition of probabilities only for mutally exclusive events! Often, it is convenient to think about the complement of events when approaching problems, and couch the problem in terms of "and" rather than "or".

Example: If there is a 20% chance of rain today and 20% chance of rain tomorrow, what is the probability that it will rain in the next two days?

Example: we draw 2 cards from a shuffled deck. What is the probability that they are both aces given a) we replace the first card, and b) we don't?

Example: consider a set of 9 objects of three different shapes (square, circle, triangle) and three different colors (red, green, blue). What is probability we will get a blue object? A triangle? What is probability that we will get a blue triangle? Are these independent?

Understand and be able to use basic axioms of probability.

2.1 Conditional probabilities

Sometimes, however, we have the situation where the probability of an event A is related to the probability of another event B, so if we know something about the other event, it changes the probability of A. This is called a conditional probability, $P(A|B)$, the probability of A given B. If $P(A|B) = P(A)$, then the results are independent.

What are some astronomy examples?

Example: Imagine we have squares, circles and triangles again, but only blue triangles. What is the probability we will have a triangle if we only look at blue objects? Thinking about this in the classical approach:

$$P(\text{blue}) = \frac{n(\text{blue})}{n}$$

$$P(\text{blue} \cap \text{triangle}) = \frac{n(\text{blue triangles})}{n}$$

$$P(\text{triangle} — \text{blue}) = \frac{n(\text{blue triangles})}{n(\text{blue})}$$

Generalizing to the axiomatic representation:

$$P(\text{triangle} — \text{blue}) = \frac{P(\text{blue} \cap \text{triangle})}{P(\text{blue})}$$

which serves as a definition of conditional probability, $P(A|B)$, the probability of A given B:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Note that, by the same reasoning:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

But, certainly, $P(A|B) \neq P(B|A)$! Given the definitions,

$$P(B|A)P(A) = P(A|B)P(B)$$

which is known as Bayes theorem. Note that this is a basic result that is not controversial, but some applications of Bayes theorem can be: the Bayesian paradigm extends this to the idea that the quantity A or B can also represent a hypothesis, or model, i.e. one talks about the relative probability of different models.

If we consider a set of k mutually exclusive events B_i that cover all possibilities (e.g., red, green and blue), then the probability of some other event A is:

$$P(A) = P(A \cap B_1) + P(A \cap B_2) + ...P(A \cap B_k)$$

$$P(A) = P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + ...P(A|B_k)P(B_k)$$

Bayes theorem then gives:

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{P(A)} = \frac{P(A|B_i)P(B_i)}{P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + ...P(A|B_k)P(B_k)}$$

This is useful for addressing many probability problems.

Example (fake!): say we can classify spiral galaxies into either star forming galxies or AGN based on line ratios, since almost all AGN have $R > R_{crit}$. Let A represent when $R > R_{crit}$. However, the test is not perfect: If you observe an AGN, $P(A|AGN) = 0.9$, but 10% of AGN fail the test (false negative). At the same time, 1% of SF galaxies also satisfy A (false positive): $P(A|SF) = 0.01$ (false positive). Also, there are many more star forming galaxies than AGN: $P(AGN) = 0.01$. If you observe a galaxy with $R_1 > R_{crit}$, what is the probability that it is an AGN?

Note the significant problem posed by false positives for a test for a rare condition.

Review:

What is $P(A \cup B)$? How is related to $P(A)$ and $P(B)$?

What is $P(A \cap B)$? How is related to $P(A)$ and $P(B)$?

What is $P(A|B)$?

What is Bayes theorem?

Example: Monty Hall, or three doors, problem. You're on a game show where you are asked to choose a door from 3 doors, labelled A, B, and C, with a car behind one of them. After you make your choice, the host, who knows where the car is, opens one of the remaining doors to show nothing, and asks whether you want to change your guess. Statistically, does it increase your chances to change or not?

Know what conditional probabilities are. Know and understand Bayes theorem, and be able to apply to probability problems.

2.2 Random variables and probability distributions

When we talk about uncertainty in science, we are often talking about quantities whose values result from a measurement of a quantity that has some random variations. These variations may results from variations of the quantity across a population (e.g., masses of stars), or they may result from uncertainties in the measurement of the quantity, or both. These quantities are called *random variables*.

Random variables can be discrete or continuous: discrete random variables take on specific distinct values, whereas a continuous random variable can take on any value. In either case, we describe the probability of getting a value with a *probability distribution function (PDF)*. A normalized probability distribution (or density)

function has the property:

$$\int p(x)dx = 1$$

for a continuous function or

$$\Sigma p(x_i) = 1$$

for a discrete function. A distribution function that cannot be normalized is called *improper*.

Distribution are often described (incompletely) by a few characteristic numbers, e.g., identifying the location and width of the distribution. Often, moments of the PDF are used:

- Mean, or expectation value: $\mu = \int xP(x)dx =< x >$
- Variance: $\int (x-\mu)^2 P(x)dx = \int x^2 P(x) - \mu^2$ (why?)
- Standard deviation: $\sigma = \sqrt{variance}$
- Skewness: $= \frac{1}{\sigma^3}\int (x-\mu)^3 P(x)dx$ (beware multiple definitions) : measures asymmetry
- Kurtosis: $= \frac{1}{\sigma^4}\int (x-\mu)^4 P(x)dx$: measures strength of wings relative to peak
- See here for some graphical illustrations of moments

Other descriptors can also be used

- Median: central value
- Mode: most common value (for discrete quantities)
- Mean absolute deviation
- percentiles: if one integrates the PDF, this gives the cumulative distribution function (CDF), i.e. the percentage of time a variable falls below a given value. This function ranges from 0 to 1. If one inverts this function, you get the quartile function, which gives the value below which some percentage of the values lie. The median is the quartile function evaluated at 0.5. A desriptor of the width of the PDF is given by the interquartile range, the difference between the quartile function evaluated at 0.75 and at 0.25: $q_{75} - q_{25}$

Note that if you have some function of the random variable, then the expectation value of that function is:

$$< f(x) >= \int f(x)P(x)dx$$

example: stellar masses and luminosities.

Some useful relations for expectation and variance operators:

$$E(x) \equiv \frac{\Sigma x_i}{N}$$

$$var(x) \equiv \frac{\Sigma(x_i - E(x))^2}{N}$$

$$E(x + k) = E(x) + k$$

$$E(x + y) = E(x) + E(y)$$

$$E(kx) = kE(x)$$

$$E(E(x)) = E(x)$$

$$var(k + x) = var(x)$$

$$var(kx) = k^2 var(x)$$

For independent variables:

$$E(xy) = E(x)E(y)$$

$$var(X + Y) = var(X) + var(Y)$$

Understand the concept of a probability distribution function (PDF), and its first several moments: mean, standard deviation, skewness, kurtosis. Know what is meant by an expectation value.

2.3 Estimators

In most case, we need to estimate parent quantities from a finite sample. Various estimators can be used for different quantites. Ideally, we want estimators that are unbiased, consistent, efficient, and robust.

- unbiased: the expectation value of estimator is equal to the quantity being estimated
- consistent : with more data, it converges to the true value

- efficient: makes good use of the data, giving a low variance about the true value of the quantity
- robust: isnt easily thrown off by data that violate your assumptions about the pdf, e.g., by non-Gaussian tails of the error distribution

It is not always possible for any single estimator to be the most optimal in all of these characteristics!

Given the definition of the mean and standard deviation as moments of the PDF, one might consider using sample moments to estimate these from a finite sample. In this case, the probability of each point is equal to 1/N: the PDF is just sampled by the relative number of points at each observed value.

So, for the mean, one would use:

$$\bar{x} = \Sigma P_i x_i = \frac{\Sigma x_i}{N}$$

(a familiar result!). To check if this is biased, we want to compute its expectation value, the average of this over multiple samples:

$$< \bar{x} >= \left\langle \frac{1}{N} \Sigma x_i \right\rangle = \frac{1}{N} \Sigma < x_i >= \mu$$

so it is unbiased.

What about the *variance* of this estimator? Using the operators above, this is found to be:

$$< (\bar{x} - \mu)^2 >= \frac{\sigma^2}{N}$$

(see, e.g., Bailer-Jones 1.3 and 2.3, also here) and gives us the uncertainty in the mean. (Note we previously got this result in ASTR 535 via propagation of uncertainties):

$$\bar{x} = \frac{\Sigma x_i}{N}$$

$$\sigma_{\bar{x}}^2 = \Sigma \frac{1}{N^2} \sigma_x^2 = \frac{\sigma^2}{N}$$

This result demonstrates that more data reduces the uncertainty! Note that this requires an estimator for the variance.

Extending to an estimator for the variance, one might expect to use:

$$S(x) = \Sigma P_i (x_i - \bar{x})^2 = \frac{1}{N} \Sigma (x_i - \bar{x})^2$$

However, let's calculate whether this is biased, by comparing it to the true variance. The expectation value of $S(x)$ is (see here):

$$< S(x) >= \sigma^2 \frac{N-1}{N}$$

This arises because we have to use the sample mean rather than the true mean, so that removes one degree of freedom; note the case of only having one point (for which we clearly can't get an estimate of the variance), or two points, for which our variance estimate is clearly too low. So if we multiply our biased estimator by $\frac{N}{N-1}$, then it would be unbiased, so an unbiased estimator of the sample variance is:

$$\hat{S}(x) = \frac{1}{N-1} \Sigma (x_i - \bar{x})^2$$

which is a common result. Note, however, that the square root of this is not an unbiased estimator of the standard deviation, because the expectation value of the square root of a distribution is not equal to the square root of the expectation value of the distribution!

This leads to the expression for the uncertainty in the mean, using our unbiased estimator for the variance:

$$SEM = \sqrt{\frac{\hat{S}(x)}{N}} = \frac{\hat{\sigma}}{\sqrt{N}} = \sqrt{\frac{1}{N(N-1)} \Sigma (x_i - \bar{x})^2}$$

One can also derive the uncertainty of the sample standard deviation: it is approximately

$$\frac{\hat{\sigma}(x)}{\sqrt{2(N-1)}}$$

(Bailer-Jones, section 2.4)

Estimator of the skew:

$$\gamma = \frac{1}{N\sigma^3} \Sigma (x_i - \mu)^3$$

and kurtosis:

$$\kappa = \frac{1}{N\sigma^4} \Sigma (x_i - \mu)^4 - 3$$

These are defined using the population mean, μ; if the sample mean is used, they would be biased and there are correction factors that can be applied.

Note that, while these estimators are unbiased and consistent, they are not necessarily the most efficient, nor are they necessarily robust. There is no single estimator that is guaranteed to be unbiased, most efficent, and most robust!

Other estimators of the mean: midrange (average of lowest and highest point). See Bailer-Jones 2.3 for a demonstration that, for a uniform distribution, midrange is more efficient than mean!

The sample mean and variance are sensitive to outliers. The median is more robust, but has larger uncertainty: $variance \sim \frac{\pi}{2}\frac{\sigma^2}{N}$, which is larger than the uncertainty in the mean: you would need more points to get the same uncertainty out of the median as from the mean.

For a more robust estimator of the variance, the mean absolute deviation (MAD, average of the absolute value of the difference between observation and mean or median) is often used, but note that it is a biased estimator; for a normal distribution with large N, $\sigma = 1.253MAD$. Another robust estimate is the interquartile range, $q_{75} - q_{25}$. Note that, for a Gaussian distribution, $\sigma = 0.7413(q_{75} - q_{25})$.

A common method to get a robust and efficient estimator of the mean is to use so-called σ-clipping: determine a robust mean and variance and use this to reject outliers, then use an unbiased and efficient estimator to get robust estimators.

Other estimators include least-squares estimators and maximum likelihood estimators. A least-squares estimator of the mean, for example, is the value, $\hat{\mu_{LS}}$ that minimizes:

$$\Sigma(x_i - \hat{\mu_{LS}})^2$$

A maximum liklihood estimator, is the value that maximizes

$$\Pi P(x_i|\hat{\mu_{MLE}})$$

where Π represents a product, in this case, of the probabilities of each individual point.

The relative quality of different estimators depends on the distribution of the quantity for which the estimator is sought. For many simple distributions, the different estimators, e.g., for the mean, yield the same quantity, but this is not always the case.

The unbiased estimator that is most efficient, i.e., gives the lowest variance, is called the minimum variance unbiased estimator (MVUE). Under some conditions, one can derive what the minimum variance of an estimator is: this is called the Cramer-Rao bound (see Feigelson & Babu 3.4.3).

Understand the concept of an estimator and the concepts of bias, consistency, efficiency, and robustness. Know the expressions for unbiased estimators of mean, standard deviation, and standard uncertainty of the mean.

2.4 Multivariate distributions

Imagine there is some joint probability distribution function of two variables, $p(x, y)$ (see ML 3.2; imagine, for example, that this could be distribution of stars as a function of effective temperature and luminosity!). Then, thinking about slices in each direction to get to the probability at a given point, we have:

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x)$$

This gives:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

which is Bayes theorem. Bayes theorem also relates the conditional probability distribution to the *marginal* probability distribution:

$$p(x) = \int p(x|y)p(y)dy$$

$$p(y) = \int p(y|x)p(x)dx$$

so

$$p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)dx}$$

The *covariance* characterizes the degree to which multiple variables are correlated. For a 2D distribution, it is defined by:

$$Cov(x, y) = \int\int P(x, y)(x - < x >)(y - < y >)dxdy$$

$$Cov(x, y) = < xy > - < x >< y >$$

For a finite sample, the sample covariance is (analagous to sample variance above):

$$\widehat{Cov}(x, y) = \frac{1}{N - 1}\Sigma(x_i - \bar{x})(y_i - \bar{y})$$

The variances and covariances can be summarized into a *covariance matrix*, in which the diagonal elements are the variances of the individual variables, and the off-diagonal elements are the covariances. By definition, this is a symmetric matrix.

If multiple variables are combined to construct some new quantity, e.g. $z = f(x, y)$, the variance in this quantity is related to the variance in the input variables by:

$$\sigma^2(z) = \left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + 2\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}Cov(x, y)$$

This is the standard formula for error propagation. Note that the covariance can be negative, leading to *reduced* scatter in the derived quantity. Also be cautious about the use of this formula: it strictly only applies in the limit of small uncertainties, and perhaps even more importantly, the shape of the distribution of uncertainties in the final variance may not be the same as the shape of the distribution in the input variables.

The error propagation formula can be generalized in matrix form for an arbitrary number of variables. Given the covariance matrix and the vector of partial derivatives, D,

$$\sigma^2(f) = \mathbf{D^T C_x D}$$

Some examples of covariance or lack of it:

- Variance from observational uncertainties
  - Consider multiple observations of an object in 2 bandpasses
  - Consider color as a function of magnitude
  - Consider the case of comparing two independent analyses of the same set of objects
- Variance from physical correlations
  - Consider measurements of cluster distances using main sequence fitting
  - Consider measurements of stellar parameters from spectra

Understand the concept of covariance and its mathematical definition. Know the expression for error propagation.

2.4.1 Correlation coefficients

Often the covariance is normalized by the standard deviation in each variable, leading to the *correlation coefficient*:

$$\rho(x,y) = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

$$\rho(x,y) = \frac{< xy > - < x >< y >}{\sigma_x \sigma_y}$$

which is 1 for perfectly correlated variables, -1 for perfectly anti-correlated variables, and 0 for uncorrelated variables.

This provides for a quantitative measurement of the amount of correlation between two quantities. This correlation coefficient is sometimes known as Pearson's correlation coefficient.

Note that a correlation between variables does not necessarily imply a causation between the two!

For a finite sample size, one can calculate the sample correlation coefficient, r. The quantity $\frac{r\sqrt{n-1}}{\sqrt{1-r^2}}$ is distributed as a t-distribution, so you can caluculate the significance of a given r. However, this does not allow for including information about uncertainty on measurements (as far as I can tell!). In that case, it is perhaps better to perform a fit to the data and assess the uncertainties on the fit coefficients.

Pearson's correlation coefficient tests for *linear* relations between quantities. A perfect nonlinear correlation will not give a value of one!

Other tests of correlation include the noparametric correlation tests: Spearman's and Kendall's. Spearman's correlation coefficient is Pearson's coefficient as applied to the *rank* of the variables, rather than the value of the variable itself. The *rank* of a variable is the sequence numbers of the sorted values.

2.5 Uniform distribution

uniform: equal probability of obtaining all numbers between some limits

2.5.1 Binomial distribution

binomial distribution: gives the probability for events that have two possible outcomes, e.g., a coin toss. The binomial distribution gives the probability that you'll get a certain number of a particular outcome, k, in n trials, given the probability of observing the desired outcome in a single event is p (note that the probability of observing the other outcome is $1 - p$.

Consider the (unlikely!) case when the first k events are all the desired outcome, followed by $n - k$ events with the other outcome. What is the probability of this?

$$p^k(1-p)^{n-k}$$

But there are many other ways that we can get k events. To get the total probability we just need to multiply this probability by the number of different *combinations* that give k of the desired outcome. How many are there?

First, consider the number of different *permutations*. For a set of n objects or trials, what is the number of different orders in which they can appear? This is given by $n!$. Now consider the number of different permutations if you draw k objects or trials. This is given by $\frac{n!}{(n-k)!}$.

Perhaps better: We want to know the number of different ways that k events can be placed into a sequence of n total events. Take the first of k events. How many positions out of n can it go into? How aboud the second one? the third? So, for k events, the number of permutations in n slots is $\frac{n!}{(n-k)!}$.

Finally consider the number of *combinations* there are if you draw k objects or trials, independent of the order. The number of permutations of the selected trials is $k!$, so the number of combinations is

$$\frac{n!}{k!(n-k)!}$$

Combining results gives us the *binomial distribution*

$$P(k|n,p) = \frac{n!}{k!(n-k)!} p^k (1-p)^{(n-k)}$$

where $P(k,n,p)$ is the probability of observing k outcomes given n trials, with p being the probability of getting the desired outcome in a single trial. Sometimes a special notation is used for the number of combinations:

$$P(k|n,p) = \binom{n}{k} p^k (1-p)^{(n-k)}$$

where $\binom{n}{k}$ is "n choose k".

Plot of binomial distribution

Examples:

- How many poker hands are there (without a draw)?
- What's the probability of getting 3 heads in 10 coin tosses?
- What's the probability of getting 3 sets of doubles in 3 tosses of a pair of dice?
- Assume that 0.1% of the objects within a subsample are quasars. You examine 100 spectra from the subsample in detail. What is the probability that one of the spectra will be that of a quasar?
- What is the probability that three will be quasars?

Mean of the binomial distribution can be derived and is:

$$\mu = np$$

Variance of the binomial distribution can also be derived:

$$\sigma^2 = np(1-p)$$

See here for the derivations.

2.5.2 Poisson distribution

consider situations in which we count objects, like detected photons. This is a *one-sided* proposition: we know the number we detect, but not the number we don't detect! However, determining the number we detect can be derived from the binomial distribution. Consider the case where we split the detection interval into very small pieces so that there are a very large number n of "trials". The probability, p, of detecting an event in a trial is very small, but from collecting data over a period, we can estimate $\lambda = np$, the mean rate of detections. The binomial distribution is then

$$P(r|\lambda/n, n) = \frac{n!}{r!(n-r)!}\left(\frac{\lambda}{n}\right)^r\left(1-\frac{\lambda}{n}\right)^{n-r}$$

In the limit of $n \to \infty$,

$$\frac{n!}{(n-r)!} \to n^r$$

and

$$\left(1-\frac{\lambda}{n}\right)^{n-r} \to \left(1-\frac{\lambda}{n}\right)^{n} = e^{-\lambda}$$

(a definition of e). In the limit $n \to \infty$, we get the Poisson distribution:

$$P(r|\lambda) = \lambda^r \frac{e^{-\lambda}}{r!}$$

Using the results for the binomial distribution in the limit $p << 1$:

$$\mu = np = \lambda$$

$$\sigma^2 = np = \lambda$$

which is the key result for a Poisson process.

Plot of Poisson distribution

Example of possible interest: derivation of the gain from a stack of images, considering what is a robust estimator of the variance (but not the standard deviation!).

2.5.3 Gaussian (normal) distribution

Many physical variables and some instrumentation uncertainties are distributed according to a Gaussian, or normal, distribution:

$$P(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

This is a symmetric function with width characterized by σ. Note that astronomers often characterize the width by the *full width at half maximum* FWHM, where $FWHM = 2\sigma\sqrt{2\ln 2} = 2.354\sigma$ The integral of a Gaussian is called the *error function* (sometimes called *erf(x)*). This yields the well-known properties that, for a Gaussian, 68% of the points should fall within plus or minus one σ from the mean, and 95.3% between plus or minus two σ from the mean.

A related distribution is the lognormal distribution, which is a Gaussian distribution in the logarithm of a variable, $x = \log y$. This would arise from the central limit theorem for variables that are combined via multiplicative operations. An astronomical example might be in a Gaussian distributions of magnitudes, e.g., in globular cluster luminosities.

2.5.4 Beta distribution

(useful distribution that gives values between 0 and 1)....

2.5.5 χ^2 distribution

2.5.6 Students t-distribution

2.5.7 Cauchy (Lorentzian) distribution

Understand what uniform, binomial, Gaussian, and Poisson distributions look like. Ideally, know the formulae; certainly, know the mean and standard deviation for Gaussian and Poisson distributions.

2.6 Central limit theorem

The Gaussian may be common because of the *central limit theorem*, which says that the sum of many *independent* variables will have a distribution that is Gaussian is form, **regardless of the distributions of each of the independent variables**.

3 Data simulation

Generating artificial data can be a powerful way to test your data analysis techniques, and is highly recommended.

Note that getting good results from simulated data is a necessary, but not sufficient, condition for testing your results: real data may contain things that you are not simulating because you might not be aware of them.

3.1 Random number generation

In many simulation cases, you may wish to simulate some population of objects, drawing your simulated sample from some underlying distribution. Perhaps you want to see whether you can recover the underlying distribution from samples of different sizes.

Be aware that making random samples is not *always* the best approach! Carefully consider whether random sampling is required. An example I have run into is comparing actual color-magnitude diagrams with model CMDs: if you make random samplings to make a model CMD, you have to use a very large number of stars to random sample low density (short lifetime) regions in the CMD. But one can do a statistical comparison with a model CMD that is constructed analytically from stellar isochrones and an IMF!

To make simulated samples requires the ability to draw a "random" sample from some distribution, so we need to understand what it means to generate a random number with a computer, and how to generate random numbers from distributions that may not be built-in to standard libraries.

Random numbers also play a significant role in numerical simulations.

There exists considerable literature on the subject, and significant improvements of the past several decades, because random number generation is a critical component of encryption algorithms. Be aware that lousy random number generators exist! So if it is important, research your random number generator. See, e.g., diehard tests.

There is a good discusion of random number algorithms in Numerical Recipes.

A simple example of a random number algorithm – but not a good one! – is the linear congruential generator, which generates a sequence of integers x_i using the recurrence relation

$$x_{i+1} = Mod(ax_i + b, m) = (ax_i + b)//m$$

where a, b, and m are large integers. Clearly, this requires a starting point, the "seed", and this is generally true of all computer random number generators. The most common modern random number generator is the Mersenne-Twister algorithm.

As is evident here, the computer random number generator is really a "pseudo-random" number generator, because computers are deterministic. If you start from the same point, you get the same sequence! Most random number generates have a method for choosing the starting point, e.g., the clock time (perhaps in milliseconds), so you get a different sequence each time you call the random number generator.

Although this sounds good, it is actually recommended that you set the seed of the random number generator yourself, as you may actually want repeatibility: as you modify your code and assess improvements, you don't want to have to worry about whether the improvements (or lack of them) comes from a different set of inputs or from the different code. Of course, you also probably want to check your code with a variety of "random" sequences.

Lowest level random number generators give *uniform deviates*, i.e., equal probability of results in some range (usually 0 to 1 for floats). E.g., python random.random, random.seed, numpy.random.random , numpy.random.seed, general random number class numpy.random.RandomState

Exercise: generate sets of increasing larger uniform deviates, plot histograms.

For many common distributions, accurate and fast ways of generating random deviates have been developed. Python implementations: numpy.random.normal, numpy.random.poisson More generally: numpy RandomState object

But what about for others? Consider the *cumulative probability distribution function*, which is the integral of the PDF. What does it look like?

Transformation method: consider cumulative distribution of desired function. This is a function that varies between 0 and 1. Choose a uniform random deviate between 0 and 1, and look up what value of the cumulative distribution this corresponds to, and you have your deviate in your function! This does require that you can integrate your function, and then invert that integral.

See NR 7.3.1

In-class exercise: generate random deviates for a "triangular" distribution: $p(x) \propto x$, for $0 <= x < 1$ and 0 otherwise. What is constant of proportionality to make this a probability distribution? Use the relation to generate random deviates, and plot them with a histogram.

What if you can't integrate and invert your function? Rejection method: choose a function that you can integrate and invert that is always higher than your desired distribution. Choose a random deviate as before and look up corresponding value in the comparison function. Calculate value of both desired function and comparison function. Choose a uniform deviate between 0 and c(x): if it is larger than f(x), reject your value and start again! This method requires two random deviates for each attempt, and the number of attempts before you get a deviate in your desired distribution depends on how close your comparison function is to your desired function. See NR 7.3.2 for a graphical representation of the rejection method.

Alternatively, if your function is hard to integrate/invert, just integrate it numerically, tabulate it, and use interpolation to do the inversion.

Understand how to use and implement the transformation method for getting deviates from any function that is integrable and invertible.

Lots of other clever ways to generate deviates in desired functions, see Numerical Recipes, chapter 7

Common distributions: Gaussian, Poisson:

$$P(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(x, \mu) = \frac{\mu^x \exp^{-\mu}}{x!}$$

For these common distributions, accurate and fast ways of generating random deviates have been developed.

Python implementations: numpy.random.normal, numpy.random.poisson

More general: numpy RandomState object

Possible in-class exercises:

- simulate some data, e.g. a linear relation (x=1,10, y=x) with Gaussian (mean=0, sigma=various) scatter. Plot it up.
- take a uniform distribution of stellar magnitudes, e.g., imagine a set of standard stars. Simulate magnitude difference between observed and input mags, for different choices of zeropoint:

$$m = -2.5 log e/s + z$$

- CMD from isochrones with Poisson scatter in colors and magnitudes.

Know how to use canned functions for generating uniform deviates, Gaussian deviates, and Poisson deviates.

3.2 Data simulation

There are three basic components in data simulation: an underlying physical model, adding the effects from observation (e.g., atmospheric and instrumental effects), and finally, adding noise.

Common simulations include imaging and spectroscopic simulations.

3.2.1 Underlying physical model

For imaging simulations, you might model an underlying distribution of objects (e.g., number density profile of stars in a star cluster, galaxies in a galaxy cluster, etc.), as well as the luminosity distribution of individual objects (point sources for stars, extended objects with radial brightness profiles for galaxies, etc.).

For spectral simulations, you might use synthetic spectra from stellar atmosphere calculations, population synthesis calculation, emission line calculations, etc.

The underlying image/spectrum should be sampled at sufficiently high resolution to accommodate modeling whatever it is you might hope to extract from the simulated data. This is probably higher resolution than you might eventually provide the simulated data at.

3.2.2 Effects of observation

For imaging observations, want to account for smearing of the spatial distribution, e.g., from seeing, diffraction, aberrations, etc., which might be a function of the location in the field of view. This is done using a characterization of the point spread function (PSF), which gives the flux distribution of the system for a point source. You might also want to consider the light loss as a function of bandpass due to absorption in the Earth's atmosphere and perhaps, the addition of sky "background".

For spectral observations, you want to account for spectral smearing due to the slit width, diffraction, seeing etc. This is done using a characterization of the line spread function (LSF), which gives the flux distribution with wavelength of a monochromatic source. Note that the LSF may or may not take a simple analytical form, although often, a simple form like a Gaussian might be used. In practice, one would derive an LSF from observation of a monochromatic source, and recognize that it might be a function of wavelength.

(Aside: note that there might be more than one thing contributing to the width of lines, e.g., the instrumental profile and also a velocity distribution of the object(s) doing the emitting.)

Sample the smoothed spectrum at the desired scale of a simulated instrument: you may need to interpolate.

3.2.3 Noise

Add background spectrum if significant. Add noise: Poisson for signal (source+background), Gaussian for readout noise.

3.3 Convolution

The process of smoothing an input by an instrumental profile (PSF or LSF) is a process of convolution.

Convolution of two functions is defined as:

$$g(x) * h(x) = \int g(x')h(x - x')dx'$$

Usually, convolution is seen in the context of *smoothing*, where $h(x)$ is a normalized function (with a sum of unity), centered on zero; convolution is the process of running this function across an input function to produce a smoothed version. Note that the process of convolution can be computationally expensive, especially in multiple dimensions; at each point in a data series, you have to loop over all of the points in the n-dimensions that contribute to the convolution integral. In some cases it can be advantageous to consider convolution in the Fourier domain, because of the *convolution theorem*, which states the convolving two functions in physical space is equivalent to multiplying the transforms of the functions in Fourier space. Multiplication of two functions is faster than convolution!

Note that convolution does not change the sampling of the underlying function, and that the smoothing function must have the same "scale" as the raw function. Of course, after smoothing, one may find that one would not want to sample the smoothed function at a fine spacing, so, e.g., if one was designing a spectrograph, you'd choose the dispersion to match the sampling of the smoothed spectrum. But to simulate spectra, you need to start with underlying spectra at sufficient sampling so that the intrinsic features are well sampled.

Numerical routines exist to do simple convolution, e.g. Python: numpy.convolve, scipy.signal.convolve, but note that these generally assume a constant smoothing kernel, which might not be the case for real astronomical instruments.

If you are coding your own convolution, you can think of the convolution in two ways:

- for each input pixel, the flux gets smeared into a range of output pixels, with different amounts according to the kernel. In this case, the edges of the input will get smeared some outside of the original range, so you have to check for that and ignore those or produce an output array that is larger (thats the full option in numpy.convolve)

- for each output pixel, the flux comes from a range of input pixels. The input pixel with the shortest wavelength that contributes adds flux according to the _rightmost_ element of the kernel: pixels of larger wavelength contribute according the elements moving leftward in the kernel. This is the reversed kernel part.

You can do this with an explicit loop (thats the slow option) or with a numpy array multiplication (the fast option) where you multiply all of the input pixels that contribute to the output pixel by the reversed kernel and sum them up.

3.4 Sampling and sampling theorem

One also wants to consider the issue of the sampling of data or simulated data. The underlying model needs to be sampled at sufficient resolution to provide an accurate simulation, but this then gets smeared by the instrumental resolution. After smearing, the data may not need to be sampled at such high resolution, and most instruments are designed such that they don't sample at a higher resolution than is needed. But what is the resolution that is needed?

For many applications, there is a physical process (e.g., seeing) which limits the amount of high spatial frequency information. When it is the case that there is an upper frequency cutoff in the power spectrum of the objects being observed (band-limited), it is possible to recover the full function from samples of it if the samples are sufficiently fine. This is known as the sampling theorem. It says that if you have a band-limited function, you can recover the function if you have samples at spacing not exceeding $0.5/\nu_c$, where ν_c is the spatial frequency at which the power spectrum goes to zero, also known as the Nyquist frequency. For example, if you say seeing wipes out scales less than 0.2 arcsec, you need to sample at 0.1 arcsec to recover the full function.

In practice, we don't measure a frequency cutoff and we don't rigorously define critical sampling, but we talk about it anyway. Generally, in astronomy, critical sampling is sampling at at least twice the full-width- half-maximum of the seeing disk (or the size of the Airy disk for diffraction limited applications). Note that this definition is pretty loose considering we have square, not circular pixels. Probably something like three samples is more reasonable.

If you can recover the underlying function, then you can sample it at different locations. You can do this using the sampling theorem by something known as sinc interpolation. This works by filtering by a box function in transform space, which is equivalent to convolution with a sinc function in real space.

However, sinc interpolation may not be accurate unless you are better than critically sampled. If you are undersampled, it fails miserably. Near critical sampling, it can lead to non-flux conserving interpolation, which you generally really want to avoid. It also may fail in the presence of significant noise.

Also, note that even for astronomical data that may be approximated as band-limited, there may be objects in the data that do not have the same spatial smearing, e.g. cosmic rays. These can turn into relatively nasty things after sinc interpolation!

3.5 Interpolation

There are certainly other interpolation techniques, such as linear interpolation, polynomial interpolation, and spline interpolation. Each makes some assumption about the underlying function to provide data at points in between the sampled data points, so you should be aware of this and make sure that the choice of interpolator does not affect your results.

You may also be interested in carrying along error information for each pixel values. This is one of the prime reasons why you might want to avoid interpolation if at all possible; interpolation generally leads to correlated errors between neighboring pixels, since the value at one location generally is derived from raw data values at several locations. You can propagate errors, but people rarely propagate covariance matrices that explicitly give the correlation.

3.6 Deconvolution

3.7 Cross correlation

3.8 Forward vs inverse modeling

Understand how you can go about simulating data. Know what convolution is and how to implement/use it. Know what the convolution theorem is.