Brock Parker
University of Arizona
Department of Astronomy & Steward Observatory
Professor Tim Eifler

Statistical and Computational Methods for Astronomy (ASTR 513)

Homework 2

September 19, 2023

---

1. **Bayes' Theorem**

   (a) **Derive Bayes' Theorem.**

The probability of the intersection of two events, $P(A \cap B)$, is commutative, as the intersection does not change based on order. Thus,

$$P(A \cap B_i) = P(B_i \cap A), \quad \text{where} \quad i = 1, \ldots, k. \tag{1}$$

Using the multiplicative rule of probability combined with the previous property, we have that

$$P(A \cap B_i) = P(A|B_i)P(B_i) = P(B_i \cap A) = P(B_i|A)P(A). \tag{2}$$

$$\implies P(B_i|A) = \frac{P(A|B_i)P(B_i)}{P(A)}. \tag{3}$$

Applying the law of total probability, we have that

$$\boxed{P(B_i|A) = \frac{P(A|B_i)P(B_i)}{P(A|B_1)P(B_1) + \ldots + P(A|B_k)P(B_k)}, \quad \text{where} \quad i = 1, \ldots, k,} \tag{4}$$

which is exactly Bayes' Theorem.
Alternatively, if B is an individual event, i.e. a data vector, the theorem can be expressed as

$$\boxed{P(B|A) = \frac{P(A|B)P(B)}{P(A)}.} \tag{5}$$

   (b) **Explain Bayes' Theorem in the context of inference, i.e., when going from data space to model (parameter) space. What are the ingredients? How are they obtained? What is difficult about priors?**

When moving from data space into model space, three different ingredients are needed. First, the probability of A, $P(A)$. This gives the probability of the data, which is typically the most computationally expensive to calculate, and often comes from integrating over the multidimensional posterior, which is not known without the posterior itself. Often, however, the absolute posterior is not needed, as we only care about the normalized probability distribution, allowing us to ignore $P(A)$, which is a constant for each experiment. However, if comparison of models is desired, this cannot be ignored, but can be subsequently calculated from the resulting normalized posterior, $P(B|A)$.
We also must know the probability of B, $P(B)$, which is the probability of the model, typically called the prior distribution. This tells us how likely our evaluated model is given our specific input parameters for each evaluation of Bayes' theorem. While the likelihood is based on the model and the data, the prior is based on previous knowledge of the problem, and constrains how well we already known each parameter. Depending on our selection of a prior, the resulting posterior can shift dramatically. As such, coming up

with properly informative priors is challenging. There is rarely a correct answer for the prior, and even attempting to make a truly uninformative prior can introduce biases for non-linear relations in the model. Prior distributions are thus often seen as subjective and biased by many, and their contribution must be carefully considered.

Finally, we need the probability of A given B, $P(A|B)$. In inference, this refers to the probability of the data given the model, i.e. how likely is it to obtain the data we did if we assume a certain model, which is known as the likelihood function. This function can take many different forms, and often is nonparametric as it depends on the data obtained for that experiment, as well as the method it was obtained in, as well as other factors. This requires knowledge about each of the parameters, and is often estimated with a generic function. In general, however, it simply gives the probability that a data point is a given distance away from the supplied model.

The result of this calculation, if all of the proper inputs are correct, is the true posterior distribution of the data, i.e. the multivariate probability density function that describes exactly how likely each parameter is given the supplied information. From this distribution, the best fit parameters can be calculated as well as the confidence intervals. Often, however, one or more of the needed ingredients, typically the likelihood function, is not known analytically, or the solution is too complicated to do analytically. In this case different numerical sampling methods can be employed to sample the posterior distribution. The most common is MCMC with the Metropolis-Hasting algorithm, although many other sampling methods exist.

(c) **For a multivariate Gaussian likelihood function, explain the ingredients. How are they obtained?**

A multivariate Gaussian likelihood function is described by the probability density function

$$f(\vec{x}) = \frac{1}{(\sqrt{2\pi})^n |\mathbf{C}|} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \mathbf{C}^{-1}(\vec{x}-\vec{\mu})}. \tag{6}$$

We define n as the number of dimensions, which is set by the number of parameters in parameter space, i.e. the length of the vectors. The vector of input parameters is $\vec{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$, which gives the input parameter value in each dimension, i.e. for each 1D marginal Gaussian density function. This is typically a subset sampled from the true sample, an observation. The vector of means is $\vec{\mu} = \begin{bmatrix} \mu_1 & \mu_2 & \dots & \mu_n \end{bmatrix}$, where each μ_i is the average of x_i over all of the rest of parameter space, i.e. the mean of the marginalized 1D Gaussian pdf. For a Gaussian, this is the most likely value that each parameter can take, the peak of the distribution. The spread of the multivariate Gaussian is dictated by the covariance matrix $\mathbf{C}$, also referred to as the generalized variance. For a typical 1D Gaussian, only one variance is needed to describe the spread. However, in higher dimensions the distribution function can vary in directions not orthogonal to the axes, requiring a basis set of vectors, n by n, to describe the spread. This is related to the intrinsic correlation between the different parameters. If two variables are truly independent, their covariance will be 0. The diagonals of $\mathbf{C}$ give the regular variance of each x_i, whereas the off diagonals give the covariances, how correlated each x_i is to each other. It can be calculated as $\mathbf{C} = C_{ij} = cov[x_i, x_j] = E[(x_i - E[x_i])(x_j - E[x_j])]$ for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, n$. This covariance matrix is the covariance of the data itself, and is ideally known exactly from the model. However, this is often not the case and $\mathbf{C}$ must be estimated from the data, introducing large errors through the inversion of the matrix.

2. **Probability Density Functions.**

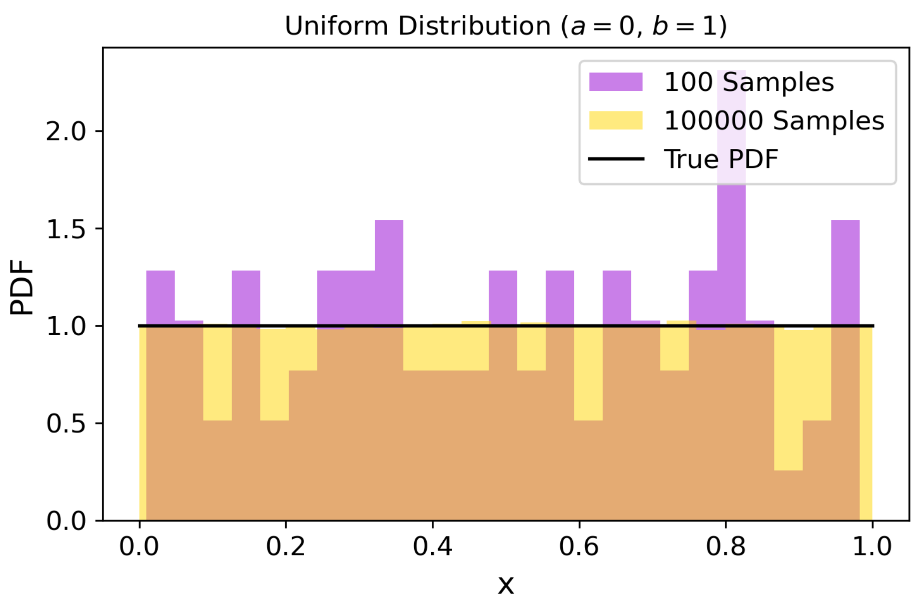
(a) **Choose 5 different probability density functions. For each of the five, draw from the distributions and create histograms with two different types of binning; plot both the historgram and the true pdf. Name the relevant parameters that determine the function and explain their role (e.g., what aspect of the distributions' shape do they control or what quantity do they represent in a use case). For 2 of the distributions, give closed form expressions (if they exist) of the moments of the pdf (only up to the 4th moment if applicable); for up to 3rd order compare to the estimated moments from the histogram data.**

---

i. The Uniform Distribution pdf is given simply by

$$f(x) = \frac{1}{b-a} \quad \text{for} \quad a < x < b \tag{7}$$

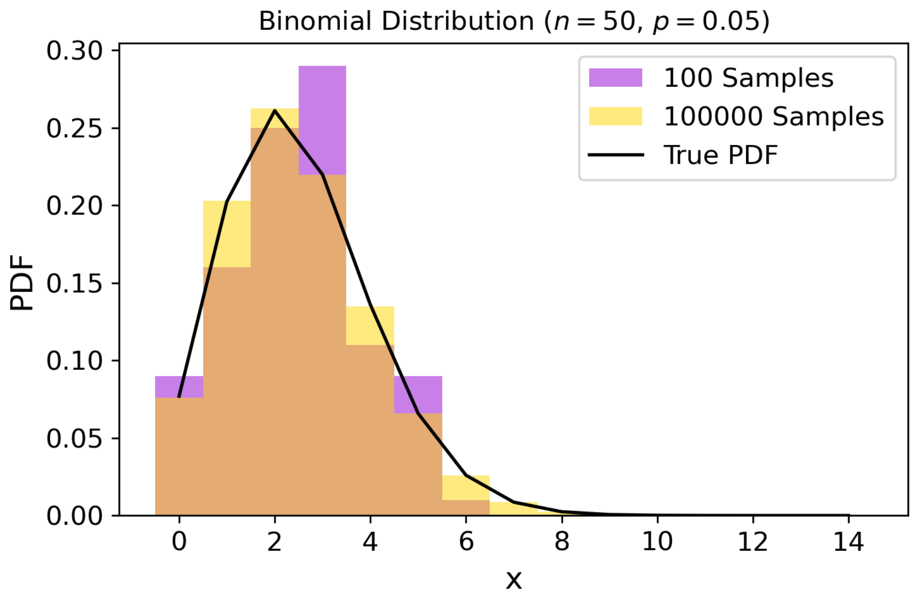
The lower domain limit is given by a, and the upper limit is given by b. The difference between these constrains the height of the distribution, where the probability of any individual point decreasing as the difference increases and vice-versa. If the domain is infinite, each point has equal probability of 0 everywhere.



ii. The Binomial Distribution pmf is given by

$$f(i) = \binom{n}{i} p^i (1-p)^{n-i} = \frac{n!}{i!(n-i)!} p^i (1-p)^{n-i}, \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{8}$$

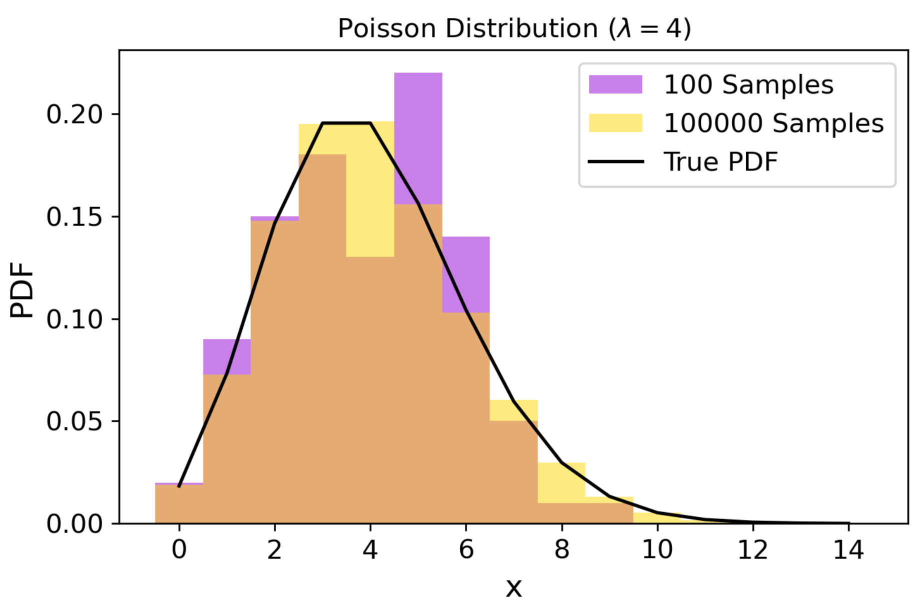
The total number of trials is given by $n \in \mathbb{N}$. As n increases, the distribution is shifted to the right and the width increases. As $n \to \infty$ while $p \sim 0.5$, the distribution approaches a normal distribution. The probability of a single success is given by $p \in (0, 1)$. For low values of p, the distribution will be skewed highly to the left; low numbers of successes are more likely. The converse is also true; the distribution will be skewed to the right for large values of p where large numbers of successes are likely. Values of $p \sim 0.5$ result in a nearly symmetric distribution.
The actual number of successes is given by i, which is a discrete number $i \in [0, n]$, and acts as the input parameter. This may represent heads in a coin flip or how many planets surround a star.



iii. The Poisson Distribution pmf is given by

$$f(i) = \frac{\lambda^i}{i!}e^{-\lambda}, \quad \text{for} \quad i = 1, 2, \dots. \tag{9}$$

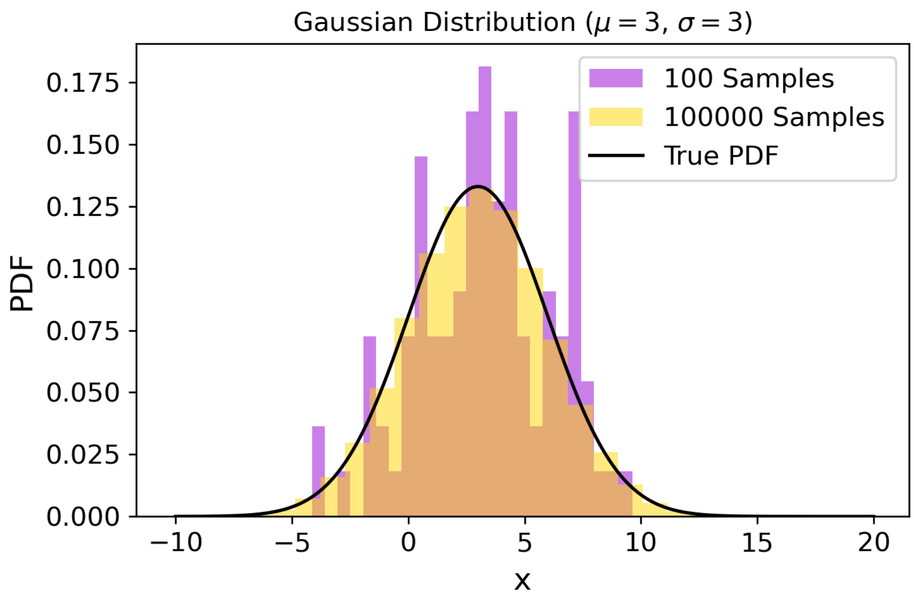
The expected rate of events is given by λ. As λ increases, the distribution is shifted towards the right, becomes more symmetric, and widens out. The expected value is also given by λ, as λ describes the expected number of events in one timescale. Additionally, as $\lambda \to \infty$, the Poission distribution approaches a Gaussian with $\sigma = \sqrt{\lambda}$.
The actual measured number of events, then, is simply given by $i \in \mathbb{N}$. This could represent photon counts or neutrino detections or anything that requires counting.



iv. The Normal, or Gaussian, Distribution is given by the pdf

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{10}$$

The mean is given by μ, which also corresponds to the location of the peak. As μ increases, the peak is shifted to the left and as μ decreases the peak is shifted to the left.
The width of the distribution is controlled by σ, which is simply the standard deviation of the data, the square root of the variance. As σ increases the width of the distribution increases, and the converse is true. No matter the value of μ or σ the distribution remains perfectly symmetrical.
The independent variable, x, can be any random variable with a given mean μ and variance σ^2. According to the central limit theorem, infinite perfect random variables will approach and result in a Gaussian distribution, and as such this applies to most random variables.



The central moments for continuous variables with continuous probability density functions $f(x)$ are calculated as

$$m_n = E[(X - E[X])^n] = \int_{-\infty}^{\infty} (x-\mu)^n f(x)\, dx \tag{11}$$

where μ is the first non-central moment, the mean, defined as $\mu = E[X] = \int_{-\infty}^{\infty} x f(x)\, dx$. For the Gaussian, this is simply $E[X] = \mu$, making the central moments

$$\mu_n = \int_{-\infty}^{\infty} (x-\mu)^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx. \tag{12}$$

Changing variable to $y = x - \mu$ results in another Gaussian with width σ and mean 0 with the same central moments, meaning that the mean has no effect on the moments about the mean. This follows from the unchanging shape of the peak as the mean changes. Thus the central moments become

$$\mu_n = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} x^n e^{-\frac{x^2}{2\sigma^2}}\, dx. \tag{13}$$

This is easily calculable from generalized Gaussian integrals, giving the generic formula

$$\mu_n = \begin{cases} 2\frac{1}{\sigma\sqrt{2\pi}} \frac{(n-1)!!}{2^{n/2+1}(2\sigma^2)^{-n/2}} \sqrt{2\sigma^2\pi} = \sigma^n (n-1)!!, & \text{for n even} \\ 0, & \text{for n odd} \end{cases} \tag{14}$$

Evaluating this for $\mu = 3$ and $\sigma = 3$, this then gives

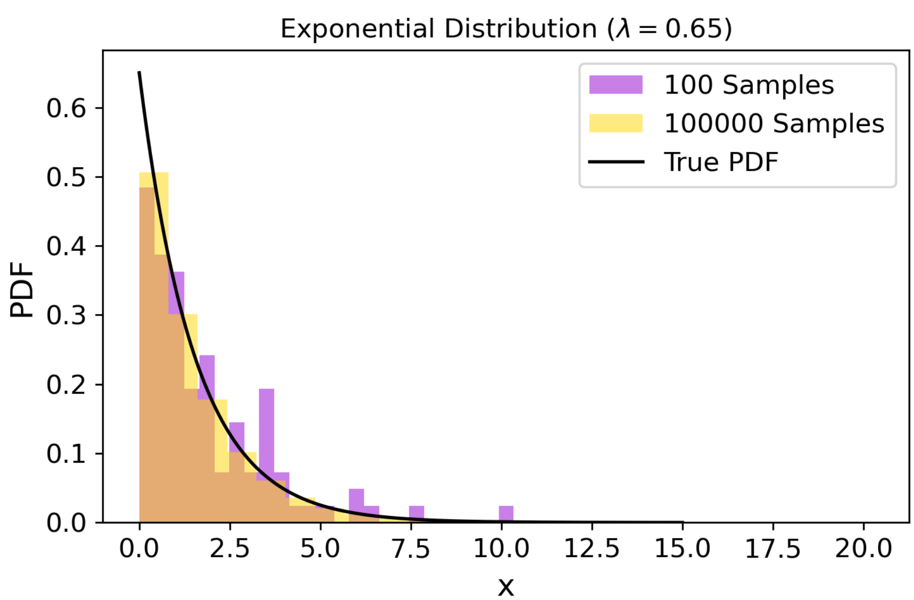
Moment (n)	Name	Analytical	Evaluated	Small Numerical	Large Numerical
0	...	1	1	1.000	1.000
1	...	0	0	0.000	0.000
2	Variance	σ^2	9	10.219	8.955
3	Skewness	0	0	16.799	-0.149
4	Kurtosis	$3\sigma^4$	243	361.152	240.689

TABLE I. Gaussian Central Moments

v. The Exponential Distribution is given by the pdf

$$f(x) = \lambda e^{-\lambda x}, \quad \text{for} \quad x >= 0 \tag{15}$$

The rate of decay is given by λ, which represents how quickly the independent variable decays for positive values. As λ increases, the distribution decays faster, and for negative values of λ, the distribution increases. As given above, λ also represents the value of the distribution at $x = 0$, and as such controls the 'height' of the distribution, where the area is simply normalized to unity. For $\lambda = 0$, the distribution is simply the uniform distribution over infinite domain.



For the exponential distribution, the mean μ is simply $E[X] = 1/\lambda$. This makes the central moments

$$\mu_n = \int_0^\infty \left(x - \frac{1}{\lambda}\right)^n \lambda e^{-\lambda x}\, dx. \tag{16}$$

Calculating this for each individual n and evaluating when $\lambda = 0.65$, this then gives

Moment (n)	Name	Analytical	Evaluated	Small Numerical	Large Numerical
0	...	1	1	1.000	1.000
1	...	0	0	0.000	0.000
2	Variance	$1/\lambda^2$	2.367	1.707	2.361
3	Skewness	$2/\lambda^3$	7.283	2.831	7.225
4	Kurtosis	$9/\lambda^4$	50.418	12.320	49.802

TABLE II. Exponential Central Moments

(b) **For 2 of the distributions, find an astronomical use case (paper) and include the link. Answer the following questions:**

- **What is the astro problem that is addressed in the paper (2 sentences)?**
- **What quantity is the random variable? What is the underlying population?**
- **What pdf is used and what moments are computed?**
- **What is the (a) statistical result of the paper?**
- **How are errors computed? What determines statistical power of the experiment? What are important systematic uncertainties?**

---

i. The first paper [2] uses simple statistical models to estimate the most likely flux per pixel for an electron multiplying charge-coupled device (EMCC). This allows individual photoelectron counting down to the shot noise limit with applications in high frame-rate astronomy and faint object spectroscopy.
The true random variable is the number of recorded photoelectrons at the end of the EMCCD multiplication process. The true underlying population is a Poissonian distribution of the incident photons originating from the observed source. This population is then convolved with the multiplication profile of the EMCCD which is exponential in nature. Combining the two allows for the statistical inference of the inherent flux rate of the object.
They compute the first non-central moment and second central moment (mean and variance) of the EMCCD multiplication distribution—the exponential—to compare the statistical power of photon counting versus assuming a fixed amplification amount. The authors also analytically compute these same moments for the final estimation of the source object flux rate from the number of frames with a threshold measured photoelectron.
The main result is that proper methods can be implemented, i.e. threshold photon counting or Bayesian inference, to allow the EMCCDs to reach the shot noise limit for very diffuse (1 photoelectron per pixel per frame) sources, as opposed to the $\sqrt{2}$ excess noise an EMCCD would produce in standard operation.
The most significant error sources arise from the underlying physics behind the distributions themselves. Both the Poissonian shot noise and the exponential amplification are inherent in the operation of the detector and cannot be mitigated or eliminated. However, they can be described precisely, and the authors do so by analytically calculating the probability of false negatives and false positives by integrating over the full description of the output of the EMCCD above and below the threshold. This thus allows precise determination of a threshold value to optimize photon counting without measuring noise.
The statistical power of the applied experiment is determined almost solely by the measured flux rate. For frames with more than one electron per pixel, photon counting is no longer at the shot noise limit, and frames with nearly zero electrons per pixel are limited by the noise of the detector (clock induced charge, read, and dark noise).
The most significant systematic uncertainty is the lack of treatment of multiple incident photoelectrons, hence the breakdown above one electron per pixel per frame. Additionally, the quantum efficiency is treated as a constant, while it truly varies across wavelengths, introducing systematic effects into the calculation of the threshold.

ii. The second paper [1] estimates the uncertainties on best fit orbital parameters for radial velocity exoplanet systems. They compare the standard resampling methods of the time and least-squares fitter to MCMC with Gibbs sampling.

The random variable(s) for this paper is the measured radial velocity of the host star. In theory this is an exact value, but it is spread out by numerous different sources (line broadening, observational limits), all of which average out to become approximately Gaussian, which is how the errors on the data points are reported.

Two different PDFs are utilized. The first is the uniform distribution, which is sampled from as the prior distributions for the parameters, some of which are in logarithmic space. The second PDF is the Gaussian pdf, which is assumed for the data point errors as well as the likelihood function in the Bayesian analysis. No moments are computed.

The main statistical result of the paper is to confirm the validity of the (at the time) new MCMC sampling method in comparison to historical resampling methods. For the majority of systems, the uninformative priors results in nearly identical posterior distributions. However, in higher dimensional systems, such as those with two or more planets, the distributions can vary wildly as other methods fail to explore the parameter space properly given computational constraints. They also show the successful retrieval of non-Gaussian and correlated posteriors from purely Gaussian input parameters. They report that many of the literature reported errors for the calculated parameters are greatly over- or under-estimated from the traditional methods.

The errors, specifically on the posterior parameters, are computed purely as confidence intervals. However, estimating these confidence intervals is not trivial in very high dimension parameter space, necessitating the need for MCMC sampling methods. They detail methods other than MCMC and resampling to calculate the posterior confidence intervals, but write all other methods off as too computationally intensive in high dimensions.

The statistical power of the experiment comes from the combination of constraining data as well as proper sampling of parameter space. While data for multi-planet systems contains the traces of long-period planets, traditional methods are unable to make use of this data efficiently. Additionally, because many of the parameters have informative priors that were not taken advantage of, the statistical power, in terms of the confidence intervals of the posterior, increases as priors incorporate known data.

The main systematic effect at play with the resampling method is the additional noise added during the resampling from added parameter correlations. This additional variation is not accounted for and can skew results. As well, the 'uninformative' priors undoubtedly influence the results in unexpected ways, particularly for those in logarithmic space. Other systematic effects may arise from uncertainties in Earth's velocity and the data collection and reduction pipeline.

---

[1] Ford, E. B. Quantifying the Uncertainty in the Orbits of Extrasolar Planets. *AJ 129*, 3 (Mar. 2005), 1706–1717.

[2] Harpsøe, K. B. W. and Andersen, M. I. and Kjægaard, P. Bayesian photon counting with electron-multiplying charge coupled devices (EMCCDs). *A&A 537* (Jan. 2012), A50.

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as st

plt.rc('axes', labelsize=14)
plt.rc('figure', titlesize=30)
plt.rc('xtick', labelsize=12)
plt.rc('ytick', labelsize=12)
plt.rc('legend', fontsize=12)
fig, ax = plt.subplots(1, 1, figsize=(6,4), layout='tight')
# BP Stylization parameters.

# BP Binomial.

n = 50
p = 0.05
n_samp = 100
n_samp_large = 100000
k = np.arange(0, 15)
# BP Necessary parameters for the binomial distribution.

ax.hist(st.binom.rvs(n, p, size=n_samp), color = 'darkviolet', alpha=0.5, histtype = 'stepfilled', density = True, bins = k.max()+1, align = 'left', range = (k.min(),k.max()+1), label = '{} Samples'.format(n_samp))
ax.hist(st.binom.rvs(n, p, size=n_samp_large), color = 'gold', alpha=0.5, histtype = 'stepfilled', density = True, bins = k.max()+1, align = 'left', range = (k.min(),k.max()+1), label = '{} Samples'.format(n_samp_large))
# BP Plotting binomial distribution at different number of samplings.

ax.plot(k, st.binom.pmf(k, n, p), color = 'k', label = 'True PDF')
# BP Plotting true binomial pdf.
ax.legend(loc = 'upper right')
ax.set_xlabel('x')
ax.set_ylabel('PDF')
ax.set_title('Binomial Distribution ($n={}$, $p={}$)'.format(n,p))
fig.tight_layout()
plt.savefig(r'/home/baparker/GitHub/Coursework/ASTR-513/HW/HW2/HW2_2_a.png',dpi = 250)
plt.show()
# BP Plotting parameters.

# BP Poisson.

fig, ax = plt.subplots(1, 1, figsize=(6,4), layout='tight')
# BP Generating plotting axis.

lam = 4
# BP Necessary parameters for the poisson distribution.

ax.hist(st.poisson.rvs(lam, size=n_samp), color = 'darkviolet', alpha=0.5, histtype = 'stepfilled', density = True, bins = k.max()+1, align = 'left', range = (k.min(),k.max()+1), label = '{} Samples'.format(n_samp))
ax.hist(st.poisson.rvs(lam, size=n_samp_large), color = 'gold', alpha=0.5, histtype = 'stepfilled', density = True, bins = k.max()+1, align = 'left', range = (k.min(),k.max()+1), label = '{} Samples'.format(n_samp_large))
# BP Plotting poisson distribution at different number of samplings.

ax.plot(k, st.poisson.pmf(k, lam), color = 'k', label = 'True PDF')
```

```python
# BP Plotting true poisson pdf.
ax.legend(loc = 'upper right')
ax.set_xlabel('x')
ax.set_ylabel('PDF')
ax.set_title('Poisson Distribution ($\lambda={}$)'.format(lam))
fig.tight_layout()
plt.savefig(r'/home/baparker/GitHub/Coursework/ASTR-513/HW/HW2/HW2_2_b.png',dpi = 250)
plt.show()
# BP Plotting parameters.

# BP Gaussian.

fig, ax = plt.subplots(1, 1, figsize=(6,4), layout='tight')
# BP Generating plotting axis.

mu = 3
sig = 3
x = np.linspace(-10, 20, 1000)
# BP Necessary parameters for the gaussain distribution.

sample = st.norm.rvs(mu, sig, size=n_samp)
sample_large = st.norm.rvs(mu, sig, size=n_samp_large)
# BP Sampling the Gaussian distribution with two different sample numbers.

ax.hist(sample, color = 'darkviolet', alpha=0.5, histtype = 'stepfilled', density = True, bins =
↪  25, label = '{} Samples'.format(n_samp))
ax.hist(sample_large, color = 'gold', alpha=0.5, histtype = 'stepfilled', density = True, bins =
↪  25, label = '{} Samples'.format(n_samp_large))
# BP Plotting gaussian distribution at different number of samplings.

ax.plot(x, st.norm.pdf(x, mu, sig), color = 'k', label = 'True PDF')
# BP Plotting true gaussian pdf.
ax.legend(loc = 'upper right')
ax.set_xlabel('x')
ax.set_ylabel('PDF')
ax.set_title('Gaussian Distribution ($\mu={}$, $\sigma={}$)'.format(mu, sig))
fig.tight_layout()
plt.savefig(r'/home/baparker/GitHub/Coursework/ASTR-513/HW/HW2/HW2_2_c.png',dpi = 250)
plt.show()
# BP Plotting parameters.

mom = st.moment(sample, moment=[0,1,2,3,4])
mom_large = st.moment(sample_large, moment = [0,1,2,3,4])

print('----Gaussian Distribution----')
print('Small Distribution: mu_0={:.3f}, mu_1={:.3f}, mu_2={:.3f}, mu_3={:.3f},
↪  mu_4={:.3f}'.format(*mom))
print('Large Distribution: mu_0={:.3f}, mu_1={:.3f}, mu_2={:.3f}, mu_3={:.3f},
↪  mu_4={:.3f}'.format(*mom_large))


# BP Exponential.

fig, ax = plt.subplots(1, 1, figsize=(6,4), layout='tight')
# BP Generating plotting axis.

lam = 0.65
```

```
x = np.linspace(0, 15, 1000)
# BP Necessary parameters for the exponential distribution.

sample = st.expon.rvs(scale = 1/lam, size=n_samp)
sample_large = st.expon.rvs(scale = 1/lam, size=n_samp_large)
# BP Sampling distribution at two different sizes.

ax.hist(sample, color = 'darkviolet', alpha=0.5, histtype = 'stepfilled', density = True, bins = 25, label = '{} Samples'.format(n_samp))
ax.hist(sample_large, color = 'gold', alpha=0.5, histtype = 'stepfilled', density = True, bins = 25, label = '{} Samples'.format(n_samp_large))
# BP Plotting exponential distribution at different number of samplings.

ax.plot(x, st.expon.pdf(x, scale = 1/lam), color = 'k', label = 'True PDF')
# BP Plotting true exponential pdf.
ax.legend(loc = 'upper right')
ax.set_xlabel('x')
ax.set_ylabel('PDF')
ax.set_title('Exponential Distribution ($\lambda={}$)'.format(lam))
fig.tight_layout()
plt.savefig(r'/home/baparker/GitHub/Coursework/ASTR-513/HW/HW2/HW2_2_d.png',dpi = 250)
plt.show()
# BP Plotting parameters.

mom = st.moment(sample, moment=[0,1,2,3,4])
mom_large = st.moment(sample_large, moment = [0,1,2,3,4])

print('----Exponential Distribution----')
print('Small Distribution: mu_0={:.3f}, mu_1={:.3f}, mu_2={:.3f}$, mu_3={:.3f}, mu_4={:.3f}'.format(*mom))
print('Large Distribution: mu_0={:.3f}, mu_1={:.3f}, mu_2={:.3f}$, mu_3={:.3f}, mu_4={:.3f}'.format(*mom_large))

# BP Uniform.

fig, ax = plt.subplots(1, 1, figsize=(6,4), layout='tight')
# BP Generating plotting axis.

a = 0
b = 1
x = np.linspace(a, b, 1000)
# BP Necessary parameters for the uniform distribution.

ax.hist(st.uniform.rvs(size=n_samp), color = 'darkviolet', alpha=0.5, histtype = 'stepfilled', density = True, bins = 25, label = '{} Samples'.format(n_samp))
ax.hist(st.uniform.rvs(size=n_samp_large), color = 'gold', alpha=0.5, histtype = 'stepfilled', density = True, bins = 25, label = '{} Samples'.format(n_samp_large))
# BP Plotting binomial uniform at different number of samplings.

ax.plot(x, st.uniform.pdf(x), color = 'k', label = 'True PDF')
# BP Plotting true uniform pdf.
ax.legend(loc = 'upper right')
ax.set_xlabel('x')
ax.set_ylabel('PDF')
ax.set_title('Uniform Distribution ($a={}$, $b={}$)'.format(a, b))
fig.tight_layout()
plt.savefig(r'/home/baparker/GitHub/Coursework/ASTR-513/HW/HW2/HW2_2_e.png',dpi = 250)
```

```
plt.show()
# BP Plotting parameters.
```

```
>>> ----Gaussian Distribution----
>>> Small Distribution: mu_0=1.000, mu_1=0.000, mu_2=10.291, mu_3=16.799, mu_4=361.152
>>> Large Distribution: mu_0=1.000, mu_1=0.000, mu_2=8.955, mu_3=-0.149, mu_4=240.689
>>>
>>> ----Exponential Distribution----
>>> Small Distribution: mu_0=1.000, mu_1=0.000, mu_2=1.707, mu_3=2.831, mu_4=12.320
>>> Large Distribution: mu_0=1.000, mu_1=0.000, mu_2=2.361, mu_3=7.225, mu_4=49.802
```