

bayesian-analysis (/github/christianhbye/bayesian-analysis/tree/main)  
/ homeworks (/github/christianhbye/bayesian-analysis/tree/main/homeworks)



([https://colab.research.google.com/github/christianhbye/bayesian-analysis/blob/main/homeworks/HW7\\_288.ipynb](https://colab.research.google.com/github/christianhbye/bayesian-analysis/blob/main/homeworks/HW7_288.ipynb))

## Homework 7

### ***Bootstrapping, Bayesian Evidence***

This notebook is arranged in cells. Texts are usually written in the markdown cells, and here you can use html tags (make it bold, italic, colored, etc). You can double click on this cell to see the formatting.

The ellipsis (...) are provided where you are expected to write your solution but feel free to change the template (not over much) in case this style is not to your taste.

*Hit "Shift-Enter" on a code cell to evaluate it. Double click a Markdown cell to edit.*

---

### Imports

```
In [1]: import numpy as np
        from scipy.integrate import quad
        #For plotting
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
In [2]: %pylab inline
        %config InlineBackend.figure_format = 'retina'
        # general plot params
        params = {
            'axes.labelsize' : 24,
            'xtick.labelsize' : 16,
            'ytick.labelsize' : 16,
            'figure.figsize'  : (14,7),
        }
        plt.rcParams.update(params)
```

Populating the interactive namespace from numpy and matplotlib

### Mounting Google Drive locally

Mount your Google Drive on your runtime using an authorization code.

Note: When using the 'Mount Drive' button in the file browser, no authentication codes are necessary for notebooks that have only been edited by the current user.

```
In [3]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

## Problem 1 - Bootstrapping Supernova Data

In this homework, we use a compilation of supernovae data to show that the expansion of the universe is accelerating, and hence it contains dark energy. This is the Nobel prize winning research in 2011 ([https://www.nobelprize.org/nobel\\_prizes/physics/laureates/2011/](https://www.nobelprize.org/nobel_prizes/physics/laureates/2011/)) ([https://www.nobelprize.org/nobel\\_prizes/physics/laureates/2011/](https://www.nobelprize.org/nobel_prizes/physics/laureates/2011/)), and Saul Perlmutter, a professor of physics at Berkeley, shared a prize in 2011 for this discovery.

"The expansion history of the universe can be determined quite easily, using as a "standard candle" any distinguishable class of astronomical objects of known intrinsic brightness that can be identified over a wide distance range. As the light from such beacons travels to Earth through an expanding universe, the cosmic expansion stretches not only the distances between galaxy clusters, but also the very wavelengths of the photons en route. By the time the light reaches us, the spectral wavelength  $\lambda$  has thus been redshifted by precisely the same incremental factor  $z = \Delta\lambda/\lambda$  by which the cosmos has been stretched in the time interval since the light left its source. The recorded redshift and brightness of each such object thus provide a measurement of the total integrated expansion of the universe since the time the light was emitted. A collection of such measurements, over a sufficient range of distances, would yield an entire historical record of the universe's expansion." (Saul Perlmutter, <http://supernova.lbl.gov/PhysicsTodayArticle.pdf> (<http://supernova.lbl.gov/PhysicsTodayArticle.pdf>)).

Supernovae emerge as extremely promising candidates for measuring the cosmic expansion. Type I Supernovae arises from the collapse of white dwarf stars when the Chandrasekhar limit is reached. Such nuclear chain reaction occurs in the same way and at the same mass, the brightness of these supernovae are always the same. The relationship between the apparent brightness and distance of supernovae depend on the contents and curvature of the universe.

We can infer the "luminosity distance"  $D_L$  from measuring the inferred brightness of a supernova of luminosity  $L$ . Assuming a naive Euclidean approach, if the supernova is observed to have flux  $F$ , then the area over which the flux is distributed is a sphere radius  $D_L$ , and hence

$$F = \frac{L}{4\pi D_L^2}.$$

In Big Bang cosmology,  $D_L$  is given by:

$$D_L = \frac{\chi(a)}{a}$$

where  $a$  is the scale factor ( $\frac{\lambda_0}{\lambda} = 1 + z = \frac{a_0}{a}$ , and the quantity with the subscript 0 means the value at present. Note that  $a_0 = 1, z_0 = 0$ ), and  $\chi$  is the comoving distance, the distance between two objects as would be measured instantaneously today. For a photon,

$cdt = a(t)d\chi$ , so  $\chi(t) = c \int_t^{t_0} \frac{dt'}{a(t')}$ . We can write this in terms of a Hubble factor

$(H(t) = \frac{1}{a} \frac{da}{dt})$ , which tells you the expansion rate:  $\chi(a) = c \int_a^1 \frac{da'}{a'^2 H(a')} = c \int_0^z \frac{dz'}{H(z')}$ .

(change of variable using  $a = \frac{1}{1+z}$ .)

Using the Friedmann equation (which basically solves Einstein's equations for a homogenous and isotropic universe), we can write  $H^2$  in terms of the mass density  $\rho$  of the components in the universe:  $H^2(z) = H_0^2 [\Omega_m(1+z)^3 + (1 - \Omega_m)(1+z)^2]$ .

$\Omega$  is the density parameter; it is the ratio of the observed density of matter and energy in the universe ( $\rho$ ) to the critical density  $\rho_c$  at which the universe would halt its expansion. So  $\Omega_0$  (again, the subscript 0 means the value at the present) is the total mass and energy density of the universe today, and consequently  $\Omega_0 = \Omega_m$  (matter density parameter today; remember we obtained the best-fit value of this parameter in Project 1?) =  $\Omega_{\text{baryonic matter}} + \Omega_{\text{dark matter}}$ . If  $\Omega_0 < 1$ , the universe will continue to expand forever. If  $\Omega_0 > 1$ , the expansion will stop eventually and the universe will start to recollapse. If  $\Omega_0 = 1$ , then the universe is flat and contains enough matter to halt the expansion but not enough to recollapse it. So it will continue expanding, but gradually slowing down all the time, finally running out of steam only in the infinite future. Even including dark matter in this calculation, cosmologists found that all the matters in the universe only amounts to about a quarter of the required critical mass, suggesting a continuously expanding universe with deceleration. Then, using all this, we can write the luminosity distance in terms of the density parameters:

$$D_L = \frac{\chi(a)}{a} = c(1+z) \int_0^z \frac{dz'}{H(z')} = c(1+z) \int_0^z \frac{dz'}{H_0 [\Omega_m(1+z')^3 + (1 - \Omega_m)(1+z')^2]^{1/2}}$$

$$= \frac{2997.92458}{h} (1+z) \int_0^z \frac{dz'}{[\Omega_m(1+z')^3 + (1 - \Omega_m)(1+z')^2]^{1/2}} \text{ [unit of Mpc]}$$

where  $H_0 = 100 \cdot h \text{ [km} \cdot \text{s}^{-1} \text{ Mpc}^{-1}]$ .

Fluxes can be expressed in magnitudes  $m$ , where  $m = -2.5 \cdot \log_{10} F + \text{const}$ . The distance modulus is  $\mu = m - M$  ( $M$  is the absolute magnitude, the value of  $m$  if the supernova is at a distance 10pc. Then, we have:

$$\mu = 25 + 5 \cdot \log_{10} (D_L \text{ [in the unit of Mpc]})$$

In this assignment, we use the SCP Union2.1 Supernova (SN) Ia compilation.

(<http://supernova.lbl.gov/union/>) (<http://supernova.lbl.gov/union/>)

First, load the measured data:  $z$  (redshift),  $\mu$  (distance modulus),  $\sigma(\mu)$  (error on distance modulus)

```
In [ ]: # data = np.loadtxt("/content/drive/My Drive/P188_288/P188_288_HW8/sn_
data = np.loadtxt("/content/drive/My Drive/P188_288/P188_288_HW7/sn_z_1
# z
z_data = data[:,0]
# mu
mu_data = data[:,1]
# error on mu (sigma(mu))
mu_err_data = data[:,2]
```

With measurements of the distance modulus  $\mu$ , we used Bayesian analysis to estimate the cosmological parameters  $w$  and  $\Omega_m$ .

Let us assume that the universe is flat (which is a fair assumption since the CMB measurements indicate that the universe has no large-scale curvature).

$\Omega_0 = \Omega_m + \Omega_{DE} = 1$ . Then, we do not need to worry about the curvature term:

$$D_L = \frac{\chi(a)}{a} = c(1+z) \int_0^z \frac{dz'}{H(z')} = c(1+z) \int_0^z \frac{dz'}{H_0[\Omega_m(1+z')^3 + (1-\Omega_m)(1+z')^w]}$$

$$= \frac{2997.92458}{h} (1+z) \int_0^z \frac{dz'}{[\Omega_m(1+z')^3 + (1-\Omega_m)(1+z')^{3(1+w)}]^{1/2}} \text{ [unit of Mpc]}$$

where  $H_0 = 100 \cdot h \text{ [km} \cdot \text{s}^{-1} \text{Mpc}^{-1}\text{]}$ .

Assuming that errors are Gaussian (can be justified by averaging over large numbers of SN; central limit theorem), we calculate the likelihood  $L$  as:

$$L \propto \exp\left(-\frac{1}{2} \sum_{i=1}^{N_{\text{SN}}} \frac{[\mu_{i, \text{data}}(z_i) - \mu_{i, \text{model}}(z_i, \Omega_m, w)]^2}{\sigma(\mu_i)^2}\right)$$

where  $z_i, \mu_i, \sigma(\mu_i)$  are from the measurements, and we compute  $\mu_{\text{model}}$  as a function of  $z, \Omega_m, w$ .

New Material:

In this problem, assume that the universe is flat, and our two parameters are  $w$  and  $\Omega_m$ . We are going to estimate 1-D posteriors for each of these using the bootstrapping technique.

As a simple first example of bootstrapping, let's apply bootstrapping to the process of obtaining an average of some data. Suppose that we have 10 measurements of  $x$ : [3.7, 3.2, 3.3, 3.1, 3.2, 3.5, 2.9, 3.4, 3.0, 3.1]. We can randomly generate 5 samples of 10 data measurements using bootstrapping by randomly sampling with replacement:

```
In [ ]: x = np.array([3.7, 3.2, 3.3, 3.1, 3.2, 3.5, 2.9, 3.4, 3.0, 3.1])

num_samples = 5
len_x = len(x)
idx = np.random.randint(0, len_x, (num_samples, len_x))
print("After bootstrap re-sampling, here are 5 sets of 10 measurements")
print(x[idx])
```

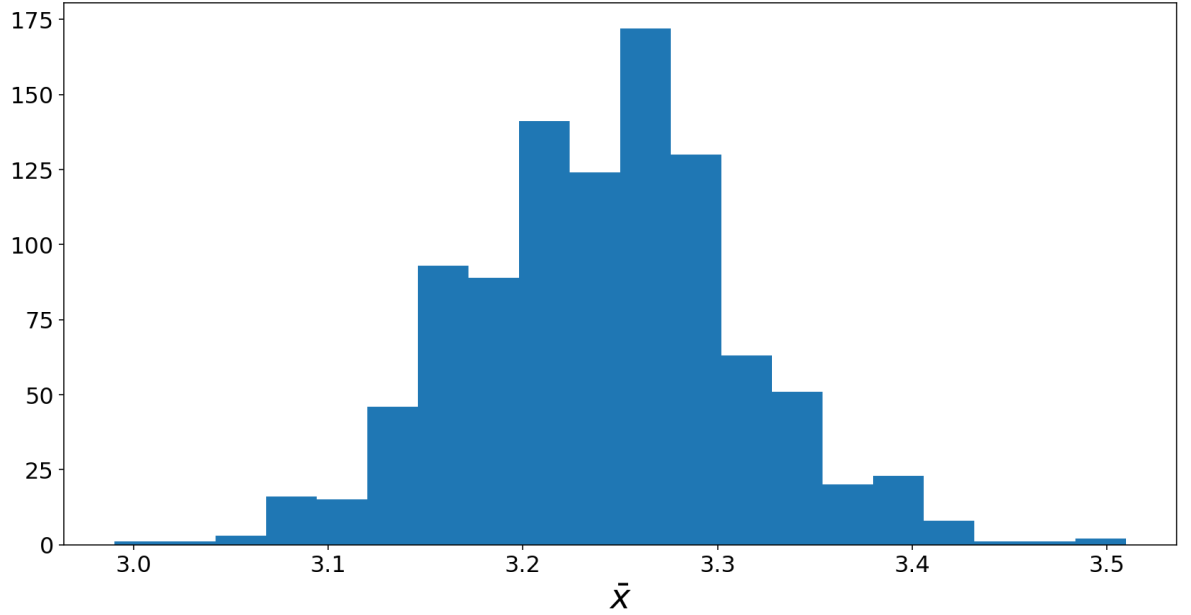
After bootstrap re-sampling, here are 5 sets of 10 measurements:

```
[[2.9 3.  3.3 3.  3.2 3.2 3.2 3.  3.2 3.3]
 [3.1 3.  3.1 3.4 3.1 3.2 3.7 3.2 3.4 3.5]
 [3.2 3.1 3.7 3.2 3.4 3.1 3.2 3.5 3.2 3.1]
 [3.1 3.1 3.7 3.2 3.1 3.  3.1 3.5 3.4 3.1]
 [3.  3.2 3.5 3.2 3.2 3.3 3.4 3.2 3.2 3.5]]
```

Now if we take the average of each one of these sets of 10 data points, we get 5 draws from the posterior of  $\bar{x}$ , as the average would be the MAP value of the average value given that data set.

1. Generate 1000 samples using bootstrapping and compute the MAP values of  $\bar{x}$  for each of them. Plot the histogram of these values to obtain a 1-D posterior of  $\bar{x}$ .

```
In [ ]: num_samples = 1000
idx = np.random.randint(0, len_x, (num_samples, len_x))
x_maps = x[idx].mean(axis=1)
plt.hist(x_maps, bins=20)
plt.xlabel(r'\bar{x}')
plt.show()
```



2. Now, use the bootstrap resampling technique to estimate the posteriors of  $\Omega_m$  and  $w$ . Take 200 bootstrap samples of supernova distance modulus measurements and their corresponding redshifts by randomly sampling from the given data. Now, think about how you can obtain the 1-d posteriors of  $\Omega_m$  and  $w$ , using maximum likelihood estimation. (Hint: you might want to try using `scipy.optimize.fmin` <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fmin.html>, this may take ~10 mins or more, you might want to report progress while looping through these 200 samples) Plot the 1-d posteriors (as histograms).

We use bootstrapping to generate 200 realizations of observations of  $\mu$  with associated errorbars. We then compute the MLE for each of the 200 experiments and then get a distribution of MLEs. With flat priors on the parameters  $\Omega_m$  and  $w$ , these histograms will be 1d-posterior estimates.

The likelihood is given as before (assuming Gaussian errors):

$$L \propto \exp\left(-\frac{1}{2} \sum_{i=1}^{N_{\text{SN}}} \frac{[\mu_{i, \text{data}}(z_i) - \mu_{i, \text{model}}(z_i, \Omega_m, w)]^2}{\sigma(\mu_i)^2}\right),$$

where

$$\mu_{\text{model}}(z, \Omega_m, w) = 25 + 5 \log_{10} \frac{D_L(z, \Omega_m, w)}{1 \text{Mpc}}$$

$$D_L = \frac{2997.92458}{h} (1+z) \int_0^z \frac{dz'}{[\Omega_m(1+z')^3 + (1-\Omega_m)(1+z')^2]^{1/2}} \quad [\text{unit of Mpc}]$$

In [ ]: *# copied from HW6*

```
c = 2997.92458

def integrand(z, omegam, w):
    """
    Function to integrate to compute D_L
    """
    x = 1 + z
    d = omegam * x**3 + (1-omegam) * x**(3 + 3 * w)
    return 1 / np.sqrt(d)

def distL(z, omegam, w, h=0.7):
    """
    Compute Distance Luminosity
    """
    z = np.array(z).ravel()
    integral = np.empty(z.size)
    for i in range(z.size):
        integral[i] = quad(integrand, 0, z[i], args=(omegam, w))[0]
    return c/h * (1+z) * integral

def dist_mod(dist_lum):
    return 25 + 5 * np.log10(dist_lum)

def minus_log_likelihood(param, z, mu, mu_sigma):
    Omegam, w = param
    if(Omegam<=0 or w>=0):
        lnL = -1.e100
    else:
        dL = distL(z, Omegam, w)
        mu_model = dist_mod(dL)
        lnL = -1/2 * np.sum((mu - mu_model)**2 / mu_sigma**2)
    return -lnL
```

In [6]: **from scipy import optimize**

```
def sn_bootstrap(z, mu, sigma_mu, nsamples=200):
    # bootstrap mu
    idx = np.random.randint(0, len(mu), (nsamples, len(mu)))
    mu_boot = mu[idx] # each row is a different realization of mu
    sigma_boot = sigma_mu[idx] # these are the associated errors
    z_boot = z[idx] # and associated redshifts
    # for each realization, we compute the MLE of omega m and w (same as
    mle = np.empty((nsamples, 2)) # empty array for MLEs
    x0 = np.zeros(2) # initial parameters
    for i in range(nsamples):
        if i % 10 == 0:
            print(f"Iteration {i}/{nsamples}")
        args = (z_boot[i], mu_boot[i], sigma_boot[i]) # function argument
        mle[i] = optimize.fmin(minus_log_likelihood, x0, args=args, disp=0)
    return mle
```

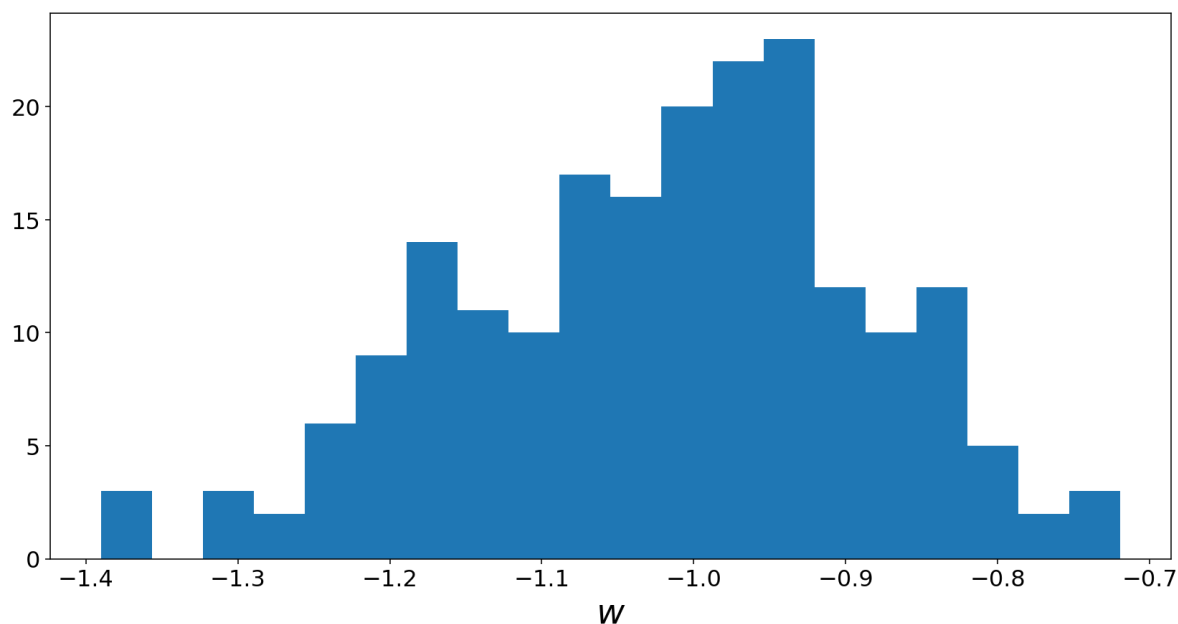
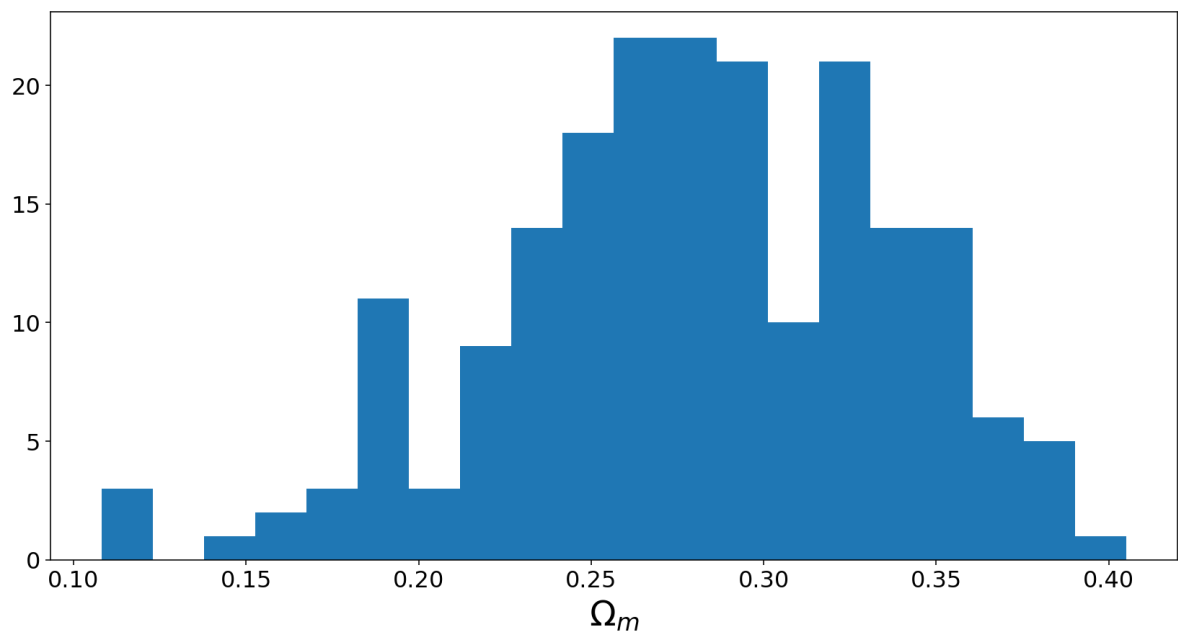
In [ ]: mle = sn\_bootstrap(z\_data, mu\_data, mu\_err\_data)

```
In [ ]: plt.hist(mle[:, 0], bins=20)
plt.xlabel(r'$\Omega_m$')
plt.show()

plt.hist(mle[:, 1], bins=20)
plt.xlabel(r'$w$')
plt.show()

# estimate parameters and 1 sigma constraints
omega, w = mle.mean(axis=0)
om_err, w_err = np.std(mle, axis=0)

# Print best-fit values and constraints
print ('$\Omega_m$ = ', omega, '+/-' , om_err)
print ('$w$ = ', w , '+/-' , w_err)
```



```
 $\Omega_m$  = 0.2797418066885927 +/- 0.05676123321300136
w = -1.0218182130058802 +/- 0.13308374414093557
```

The Bayesian evidence for hypothesis  $H$ , given data  $x$  and a model with parameters  $z$ , is given by

$$p(x|H) = \int dz p(x|z, H)p(z|H),$$

where  $p(x|z, H)$  is the likelihood and  $p(z|H)$  the prior. For a likelihood with a single mode located at  $z = \mu$ , one can approximate the evidence using the Laplace approximation to give

$$p(x|H) \approx (2\pi)^{M/2} \sqrt{\det \Sigma} p(x|\mu, H)p(\mu|H),$$

where  $M$  is the dimensionality of the parameters, and  $\Sigma$  is the covariance matrix. If there is no correlation between parameters then the determinant of the covariance matrix is given by the product of the square of the errors of each parameter

$$\det \Sigma = \prod_{i=1}^M \sigma_i^2.$$

For simplicity, we assume no correlation in this assignment.

To gain intuition for this equation one can define the prior volume as  $V_{\text{prior}} = 1/p(\mu|H)$  and the posterior volume as  $V_{\text{posterior}} = (2\pi)^{M/2} \sqrt{\det \Sigma}$ . Thus the evidence can be written as

$$p(x|H) \approx p(x|\mu, H) \frac{V_{\text{posterior}}}{V_{\text{prior}}}.$$

Hence the Bayesian evidence is approximately equal to the maximum likelihood multiplied by the posterior-to-prior volume ratio. Because the posterior volume is typically narrower than the prior, the evidence will be less than the likelihood. The posterior-to-prior volume ratio acts as a penalty on the likelihood to account for model complexity. In cases where the prior volume is much larger than the posterior volume, a large penalty will be induced.

For a hypothesis test, Bayes factor is used to compare the Bayesian evidence ratio of two hypotheses. Often one is interested in comparing some hypothesis  $H$  to a null hypothesis  $H_0$ . In such a case Bayes factor is defined as

$$B \equiv \frac{p(x|H)}{p(x|H_0)} \approx \frac{p(x|\mu, H)}{p(x|\mu_0, H_0)} \frac{V_{\text{posterior}}}{V_{\text{prior}}},$$

where in the final step we have applied the Laplace approximation. It can be seen that the Bayes factor is equal to the likelihood ratio multiplied by the posterior-to-prior volume ratio.

In this exercise we will apply Bayesian evidence in the presence of the look-elsewhere effect, which occurs when the prior volume is much larger than the posterior volume. One of the great powers of the Bayesian approach is that it has a built in correction for the look-elsewhere effect and other forms of model complexity thanks to the posterior-to-prior volume ratio.

### The look-elsewhere effect

A common problem in statistical analysis is to find evidence for a physical signal in a large, continuous parameter space, where the true position of the signal is not known a priori. By searching over a wide parameter space one increases the probability of finding large signals caused by random statistical fluctuations, as opposed to a physical source. This is known as the look-elsewhere effect -- or sometimes the "problem of multiple comparisons" in discrete cases -- and must be accounted for when performing a hypothesis test. Ignoring this effect would lead to an overestimation of the statistical significance, sometimes by a considerable amount, and thus incorrectly concluding the detection of a physical signal.

The look-elsewhere effect is prominent throughout particle physics and cosmology. One of the



most commonly known occurrences is in collider searches for new particles, for example it was a key consideration in the Higgs boson discovery. In this example, one searches a large range of masses for a resonance, without a priori knowledge of the true mass of the particle. Similarly, in astrophysical searches for particles one seeks resonances in the energy flux of various astrophysical spectra, where the true energy signature of the particle is unknown. Examples include: constraining the dark matter self-annihilation cross-section via gamma ray emission from galaxy clusters, searching for WIMPs via charged cosmic rays, searching for non-baryonic dark matter via X-ray emission from the Milky Way, and explaining the source of high energy astrophysical neutrinos. In terms of cosmology, the look-elsewhere effect occurs in searches for gravitational wave signals from black hole or neutron star mergers. Here one searches large time series for a signal, where the time and shape of the event are unknown. A further cosmological example is searching for signatures of inflation in the primordial power spectrum. The look-elsewhere effect is also prevalent in other areas of physics and beyond, for example: in astronomy it occurs when detecting exoplanets via stellar photometry, where the period and phase of the planets' transits are unknown; in biology it occurs when considering large DNA sequences to study genetic association; and in medicine it occurs when testing the effectiveness of drugs in clinical trials.

### Cosmological Inflation Example

We consider a search to find a signal for non-gaussianities in the primordial power spectrum using Planck data. A specific type of such a model adds the following oscillatory perturbation to the  $\Lambda$ CDM power spectrum,

$$P(k) = P_0(k)[1 + A \sin(2\omega k + \phi)],$$

where  $P_0(k)$  is the featureless ( $\Lambda$ CDM) power spectrum and  $A$ ,  $\omega$ , and  $\phi$  are the amplitude, frequency, and phase of the oscillatory perturbation. Specifically we search for a signal with frequency between 0 and 4,000, and phase between 0 and  $2\pi$ . We assume the signal is equally likely to appear anywhere in parameter space a priori. In this context the null hypothesis  $H_0$  corresponds to  $A = 0$ , i.e. no signal, while the alternative hypothesis  $H$  corresponds to  $A > 0$ .

In a full analysis one would fit for all 3 parameters, but, for the sake of simplicity, we will assume that the maximization over amplitude and phase has already been performed. We thus focus on

First, load the data and plot the likelihood ratio  $L/L_0 \equiv p(x|z, H)/p(x|H_0)$  as a function of frequency  $\omega$ . Note the likelihood is already evaluated at the best fit value of  $A$  and  $\phi$ .

```

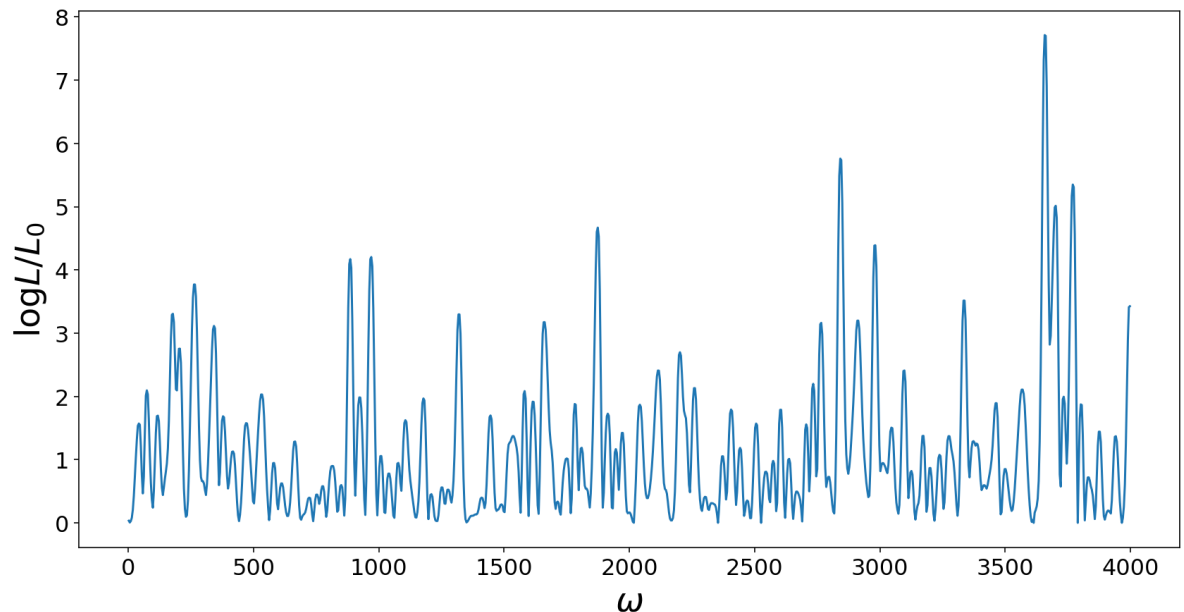
In [4]: # load the data
data = dict(np.load('/content/drive/My Drive/P188_288/P188_288_HW7/pla
omega = data['omega']
logL = data['logL']          # logL evaluated at omega
logL0 = data['logL0']

# compute log L/L0
LL = logL - logL0

# plot the data
plt.figure()
plt.plot(omega, LL)
plt.xlabel(r'$\omega$')
plt.ylabel(r'$\log L/L_0$')

```

Out[4]: Text(0, 0.5, '\$\log L/L\_0\$')



Because the search for a signal considers any frequency between 0 and 4,000, the prior volume for the frequency is very large. This leads to the likelihood being multimodal, with many (potentially all) of the peaks corresponding to noise fluctuations rather than a physical source. This is the look-elsewhere effect: by considering a large prior volume one invites the possibility of finding many large peaks which are just statistical fluctuations.

We see the maximum likelihood occurs at around  $\omega = 3660$ . Naively, one might compute the root chi-squared at this peak to determine the significance. We will start by doing this.

Approximating peaks as gaussians, the chi-squared is equal to  $2 \log L/L_0$ . The provided likelihood is discretely sampled over  $\omega$ , so to estimate the height, width, and location of the peak in frequency space a simple approach is to find the datapoint with maximum likelihood and fit a gaussian through it and adjacent data points. (As mentioned earlier, a more careful analysis would directly optimize and compute the covariance matrix of the likelihood in the space of all 3 parameters  $A$ ,  $\omega$ , and  $\phi$ .)

1. By applying a gaussian fit, or otherwise, find the location of the maximum, i.e. the best fit  $\omega$ , the value of  $\log L/L_0$  at the maximum, and the error  $\sigma_\omega$ . Print their values.

Next, make a plot of the Gaussian fit and the data around the peak.

Hint: Fit the gaussian to the likelihood ratio, not the log likelihood ratio. We have done a similar problem in HW1.5

```
In [23]: def gaussian(omega, A, omega_c, sigma_om):
    return A * np.exp(-1/2 * (omega-omega_c)**2 / sigma_om**2)

    # select a range around the max peak
    omega_max = np.argmax(LL) # index of max omega
    domega = 8 # range of omegas
    indx = np.arange(2 * domega) + (omega_max - domega) # indices

    p0 = (np.max(LL), omega[omega_max], omega[omega_max + domega] - omega[
    par = optimize.curve_fit(gaussian, omega[indx], np.exp(LL[indx])), p0=p
    sigma_omega = np.abs(par[2])

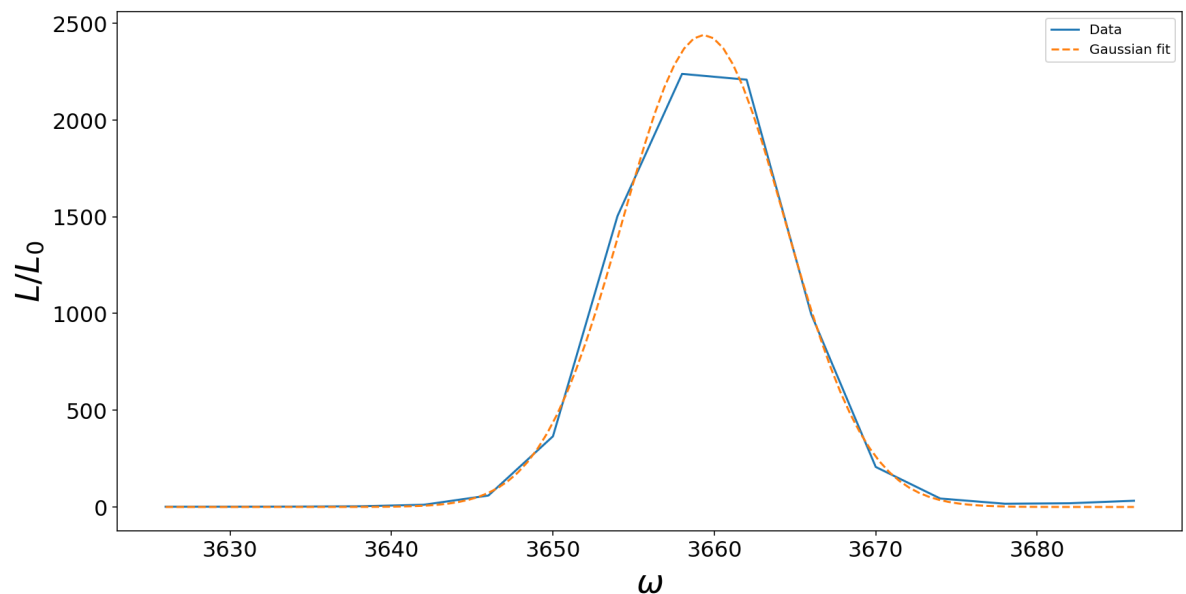
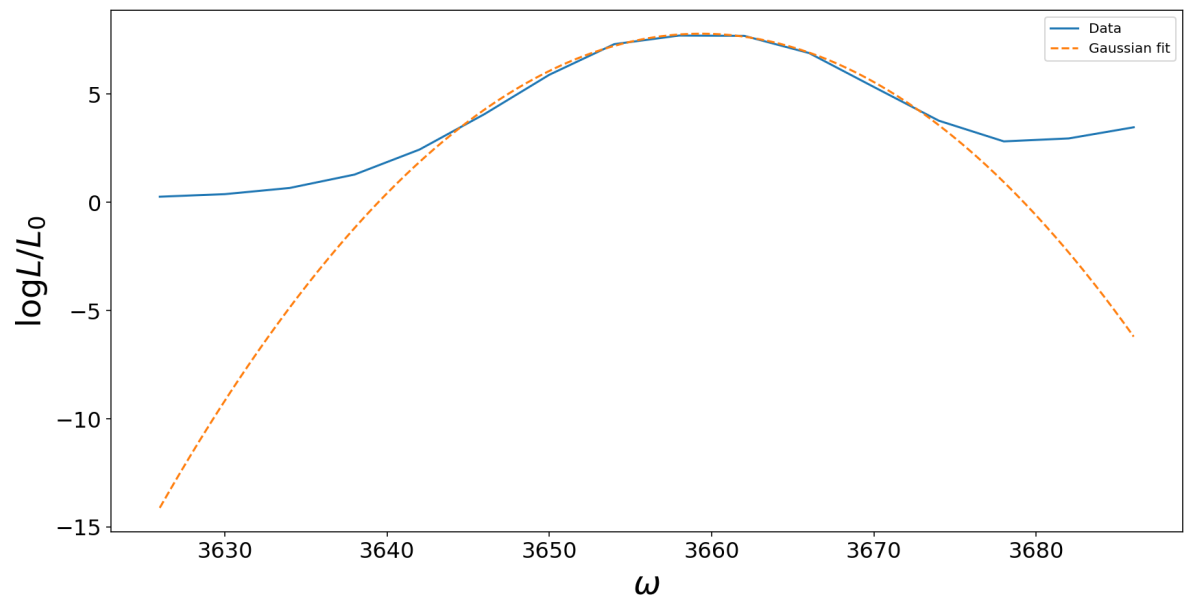
    print("Best fit:")
    print(f"omega: {par[1]:.5g}")
    print(f"log L/L0 at max: {np.log(par[0]):.5g}")
    print(f"Error on omega: {sigma_omega:.5g}")
```

```
Best fit:
omega: 3659.3
log L/L0 at max: 7.7997
Error on omega: 5.0353
```

```
In [24]: omega_grid = np.linspace(omega[indx].min(), omega[indx].max(), num=100)
fit = np.log(gaussian(omega_grid, *par))
```

```
plt.figure()
plt.plot(omega[indx], LL[indx], label="Data")
plt.plot(omega_grid, fit, ls="--", label="Gaussian fit")
plt.xlabel(r'$\omega$')
plt.ylabel(r'$\log L/L_0$')
plt.legend()
plt.show()
```

```
plt.figure()
plt.plot(omega[indx], np.exp(LL[indx]), label="Data")
plt.plot(omega_grid, np.exp(fit), ls="--", label="Gaussian fit")
plt.xlabel(r'$\omega$')
plt.ylabel(r'$L/L_0$')
plt.legend()
plt.show()
```



2. Compute the root chi-squared of the maximum peak, assuming it is a Gaussian. This gives the naive significance of the signal without considering the look-elsewhere effect. Print its value.

```
In [25]: residual = np.log(gaussian(omega[indx], *par)) - LL[indx]
chi_sq = np.mean(residual ** 2)
print(np.sqrt(chi_sq))
```

5.366694976319237

You should find the root chi-squared, to be around 4. This, naively, means that a detection at the level of 4 sigma has been made. While physicists typically set the level to announce a detection at 5 sigma, 4 sigma is still quite a significant result and would likely attract some media attention. However, the root chi-squared, and more fundamentally the likelihood ratio, does not take into account the look-elsewhere effect. Because a large prior volume introduces many spurious peaks due to statistical fluctuations, the significance must be penalized. As discussed above, Bayes factor does this for free.

3. Compute Bayes factor and print its value:

$$B \equiv \frac{p(x|H)}{p(x|H_0)} \approx \frac{p(x|\mu, H)}{p(x|H_0)} \frac{V_{\text{posterior}}}{V_{\text{prior}}},$$

where the prior volume and the posterior volume are defined as  $V_{\text{prior}} = 1/p(\mu|H)$  and  $V_{\text{posterior}} = (2\pi)^{M/2} \sqrt{\det \Sigma}$ .

Note: The Bayes factor should technically be computed by adding the contributions of all peaks, as the evidence is an integral over parameter space. However, because the maximum likelihood peak here is much higher than the others, it suffices to only consider the contribution to the Bayes factor from the maximum mode.

Hint: Use all 3 likelihood parameters,  $A, \omega, \phi$ . We provide the errors on  $A$  and  $\phi$  for the maximum peak, as well as the prior volume on the amplitude in the cell below.

```
In [26]: # given amplitude and phase information
sigma_A = 0.03901713
Vprior_A = 0.153240657

sigma_phi = 0.46561369
```

The factor  $p(x|\mu, H)/p(x|H_0)$  is simply the ratio of the likelihoods at the peak, i.e.  $L/L_0$  evaluated at the best fit  $\omega$  from above. Since we assume no correlations, the covariance matrix  $\Sigma$  is diagonal and the determinant is the product of the variances.

The prior volume of  $\omega$  is 4000 since we allow  $\omega$  to take any value in the range  $0 \leq \omega \leq 4000$  with uniform probability. Similarly, the prior volume of  $\phi$  is  $2\pi$  since  $0 \leq \phi \leq 2\pi$ . The total prior volume is therefore  $V_{\text{prior}} = V_{\text{prior},A} \cdot 4000 \cdot 2\pi$ .

```
In [27]: lh_ratio = par[0] # best fit amplitude
det_sigma_sqrt = sigma_A * sigma_phi * sigma_omega # square root of d
M = 3 # dimensionality

Vprior = Vprior_A * 4000 * 2 * np.pi
Vposterior = (2 * np.pi) ** (M/2) * det_sigma_sqrt

B = Vposterior / Vprior * lh_ratio

print(f"The Bayes factor is {B:.3f}.")
```

The Bayes factor is 0.913.

You should find the Bayes factor to be a little less than 1. Since the Bayes factor gives the ratio

of the evidence for the alternative hypothesis to the null hypothesis, a ratio less than 1 implies that there is more evidence for the null hypothesis compared to the alternative hypothesis. Hence, after accounting for the look-elsewhere effect we find that the signal is actually not worth getting excited about and is very likely a statistical fluctuation.

### **References / Extra Reading**

There are also frequentist approaches to deal with the look-elsewhere effect -- see Fergusson et al. (2015) (<https://journals.aps.org/prd/abstract/10.1103/PhysRevD.91.123506>) for such an analysis for the problem considered above. However, the Bayesian approach provides a more general framework as discussed in Bayer & Seljak (2020) (<https://iopscience.iop.org/article/10.1088/1475-7516/2020/10/009>).

---