Hypothesis testing

Statistics and Data Science

Spring 2025

Goals for today: you should be able to...

- Outline the process of hypothesis testing
- **lecture 18/19 notebook**:
  - Perform Bayesian hypothesis testing
  - **IMPORTANT**: install bokeh if you haven't already, by doing

```
mamba install bokeh
```

at a command prompt

Project meetings remaining

- Five people hadn't as of Sunday night
- I need to know your schedules to set up meetings with you...

Review: Classical (frequentist) hypothesis testing, so far

1) Define two mutually exclusive hypotheses: a *null hypothesis*, H_0, and an *alternative* or *research* hypothesis, H_1.

- Our goal is to test whether the data is consistent with the null hypothesis - we need to be able to calculate probabilities for it

2) Define the ***significance level*** $\boldsymbol{\alpha}$ we will use to decide if H_0 will be excluded, **and** the statistic we will use

- e.g., if we consider a hypothesis to be excluded if our results would happen less than 1% of the time if H_0 is true, then α=0.01
- **Important to do this before running statistics on the data!!!**

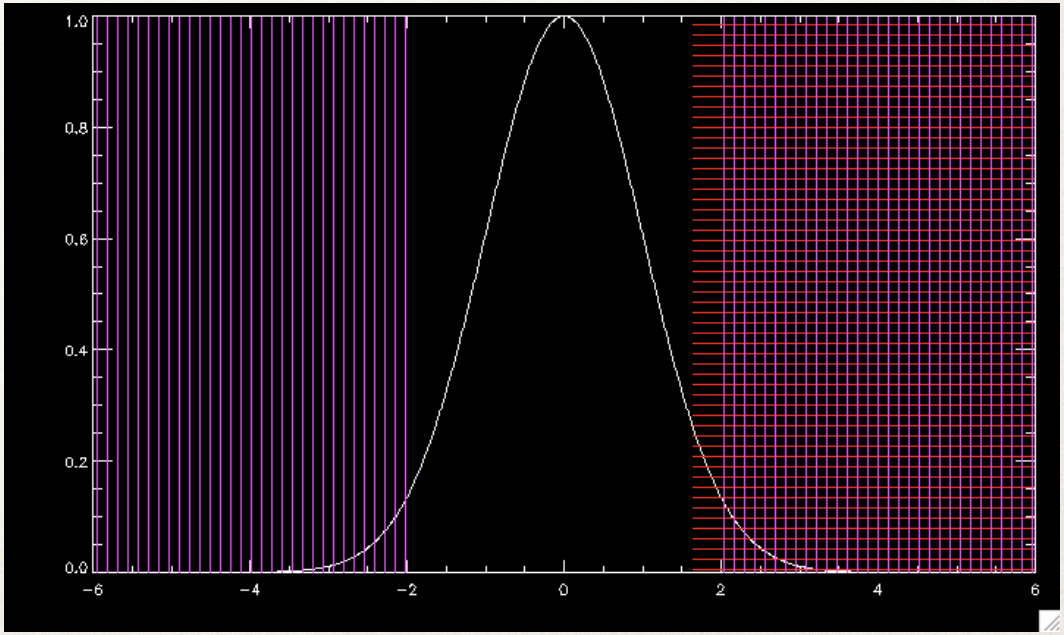
Classical (frequentist) hypothesis testing

- The data may end up favoring the alternative hypothesis H_1 over H_0, but generally we are **not** testing H_1 (unless it's simply the opposite of H_0).
- Our choice of H_1 **will** determine if we should apply a ***one-sided test*** (e.g., checking if the mean of one sample is larger than the mean of the second) or ***two-sided*** (checking if they differ, but not favoring one direction over the other).
  - This, plus our choice of value for α, defines the ***region of rejection*** within the full range of possible values (the "sampling distribution")
    - The region of rejection should comprise a fraction α of the probability distribution under H_0.

One-sided vs. two-sided tests

- In a two-sided test, the rejection region is split into two parts, each containing $\alpha/2$ of the probability - so further into the tails of the distribution.

- Plotted are one-sided (**red**) and two-sided (**purple**) 5% significance regions for a Gaussian.



Classical (frequentist) hypothesis testing

3) Measure the value of the test statistic, and calculate the probability, under H_0, that a value as extreme as the value observed **or more extreme** occurred: this is typically called the ***p-value***. H_0 is rejected if the p-value is lower than the chosen significance level α.

- We can express significance levels in terms of numbers of sigma deviation, $n\sigma$; then $\alpha = 1 - \mathrm{erf}(n/\sqrt{2})$, where *erf(x)* is the error function evaluated at x (`scipy.special.erf` in python).

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

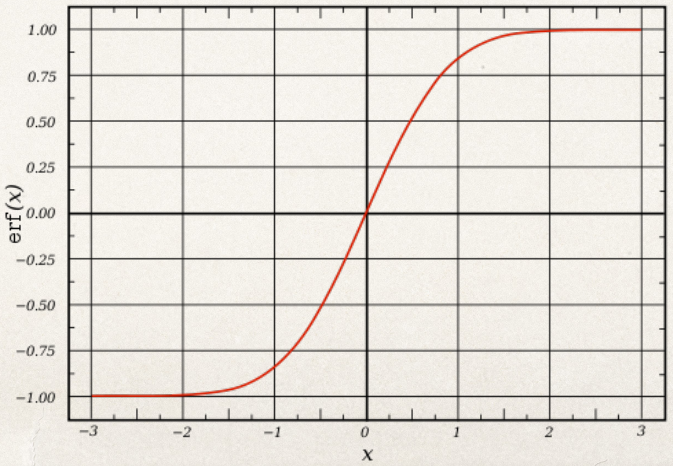
- erf(x) is closely related to the integral of a Gaussian; the integral from -∞ to x is

$$\Phi(x) = \frac{1}{2} + \frac{1}{2} \mathrm{erf}\left(x/\sqrt{2}\right)$$

Calculating equivalent sigmas

- Conversely, `stats.norm.ppf(p)` gives the value of x such that $N(0,1)$ will be less than x with probability p.
- Hence, `stats.norm.ppf(1-α/2)` is the distance from zero (in both directions) you have to go to have total probability 1-α: so e.g. you could say that getting a p value of 0.05 or less has equivalent rarity to a 1.96+ sigma event.

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,dt.$$



How do we interpret the p-value?

- If the p-value is less than α, we could say:
  - The null hypothesis is rejected at significance α
  - We have a statistically significant result
  - The probability that the difference/relationship we see occurred by chance if H_0 is true is less than α
- It would **NOT** be appropriate to say:
  - The probability that H_0 is true is p
  - The probability that we got the data we got by chance is p
  - The probability that we are rejecting H_0 incorrectly is p
  - The probability that H_1 is true is 1-p

How do we interpret the p-value?

- If the p-value is greater than α, we could say:
  - We do not observe a statistically significant difference
  - More data may be required to reject the null hypothesis

How do we choose α?

- As usual, there are tradeoffs in choosing α. If α is small, it is difficult to reject H_0, even when it is false. If α is large, we may reject H_0 even when it is true.
  - If we reject H_0 when it is true, we have made a ***Type I error*** (or "false positive"). This should happen a fraction α of the time.
  - If we accept H_0 when it is false, we have made a ***Type II error.*** If this happens a fraction of the time β, a test has *power* equal to $1-\beta$. β is sometimes called the 'false negative rate'; it depends on our choice of H_1.
- A common goal is to have power > 0.8 . In choosing α, though, we should consider all costs/losses.
- Common standards: 'significant' = $p < 0.05$, 'highly significant' = $p<0.01$ or 0.003 (equivalent to probability of a $>3\sigma$ event for a Gaussian random variable); in particle physics generally 5σ evidence (corresponding to $\alpha=6.0\text{x}10^{-7}$) is required.

Application: confidence intervals

- We previously found that a 95% confidence interval for the Hubble Constant, treating all recent measurements as equal, was [68.9,71.0] km/sec/Mpc.
- Recall the true value should lie within a measured confidence interval 95% of the time (presuming that the measurements are all really measuring the same thing with no overall systematic error).
- So if our null hypothesis were that the Hubble Constant is 50, we could reject it with significance α=0.05; as if the null were true, there would be 95% probability that the value 50 should fall within any measured 95% confidence interval.
- A 99% confidence interval turns out to be [68.5,71.0]; 99.9% is [68.2, 71.5]. So the null hypothesis is rejected at a very high level of significance.

Summary: Frequentist hypothesis testing

1) Define a *null hypothesis*, H_0, we are trying to reject (and can calculate probabilities for) and an *alternative* or *research* hypothesis, H_1, which determines if we are checking for positive/negative differences (*one-sided test*), or any sort (*two-sided*).

2) Define the *significance level* α we will use to decide if H_0 us excluded (e.g. if we only reject H_0 if our results would happen less than 1% of the time if H_0 is true, then α=0.01), and the statistic to be used. This, plus the nature of the alternative hypothesis H_1 defines the *region of rejection* for the statistic: i.e., the range of values that would reject the null hypothesis

3) Measure the value of the test statistic, and calculate the probability, under H_0, that a value as extreme as the value observed from the data *or more extreme* would occur: the *p-value*. H_0 is rejected if the *p*-value is lower than the chosen significance level α.

The replication crisis

- It has recently become apparent that many 'statistically significant' results in psychology, social sciences, medicine, etc. do not show up as statistically significant when experiments are repeated
  - E.g.: the Open Science Collaboration tried to replicate 100 psychology experiments; only 36% yielded p-value < 0.05 , vs. 97% of the original papers (https://ppw.kuleuven.be/okp/_pdf/Nosek2015ETROP.pdf)
- Common problems:
  - p-hacking: Performing many different tests and reporting only the significant outcomes
  - Publication bias: Null results don't get published
  - Selection bias: e.g., stopping a study when the desired results are obtained
  - Low-power studies: may give a low p-value by chance
- p-values can still be useful, but it is good to treat them with caution.
  - Best practice is always to provide a confidence interval for effect size as well

Review before we go on

- Suppose you have a set of 100 measurements that you know are distributed as $N(\mu,2^2)$, but μ is not known. In your groups, discuss:
  - What might we call μ?
  - What is the true standard deviation (i.e., the standard deviation of the distribution used to generate the data)?
  - How does the standard deviation of the mean relate to this quantity? How do we interpret that value?
  - How would we determine the sample standard deviation? What does it tell us about?
  - How would we estimate the standard error (sample standard deviation of the mean)? What does it tell us about?

Comparing means of two samples

- Let's start with a simple case: two Normally-distributed samples with the same standard deviation. Do they have the same mean?
  - E.g., we could compare the mean stellar mass for two different sets of galaxies with similar measurement errors (e.g., those that are redder in color vs. those that are bluer); or the distributions of energies for two different set of events from a particle experiment
  - A simple case: let's compare 18 data drawn from each of N(0,1) vs. N(1,1) - so the means are $\mu_1 = 0$ and $\mu_2 = 1$ - and see if the difference in means is statistically significant

```
ndata=18
data1=random.randn(ndata)
data2=random.randn(ndata)+1
```

Now plot histograms of data1 and data2 in the same plot, using the same binning for each.

Confidence intervals

- We can use the observed mean and standard deviation for each sample to get a confidence interval for the mean of each.
- Modify the below code to calculate values for sigma1 and sigma2 (reminder: sample std. deviation is **`np.std`** with **`ddof=1`**)

```
mean1=np.mean(data1)
mean2=np.mean(data2)
sigma1=??? # want the standard deviation of the mean of data1
sigma2=??? # want the standard deviation of the mean of data2
print('means',mean1,mean2)
print('sigmas',sigma1,sigma2)
tfactor=stats.t.ppf(1-0.025,ndata-1)
print('CI 1',mean1-tfactor*sigma1,mean1+tfactor*sigma1)
print('CI 2',mean2-tfactor*sigma2,mean2+tfactor*sigma2)
```

Confidence interval for the difference of means

- Apply propagation of errors based on `sigma1` and `sigma2` to figure out the uncertainty in, and a confidence interval for, (`mean2-mean1`)…
  - (**Note**: the *t* factor needs to be recalculated; in comparing two samples of size n_1, n_2, the *t* statistic for the difference will be distributed as the *t* distribution with (n_1+n_2-2) degrees of freedom)

```
tfactor=stats.t.ppf(1-0.025,2*ndata-2)
mean_diff=???
sigma_diff=???
print('CI diff',???)
```

- **Evaluate: are the data consistent with a hypothesis of no difference?**

Setting a lower limit

- If our alternative hypothesis is that $\mu_2 > \mu_1$, we should perform a one-sided test
- Equivalently, we can calculate a **lower limit** on the difference and see if it contains zero
- In this case, the t value we care about is the value at the 5th/95th percentile:

```
tfactor=stats.t.ppf(1-0.05,2*ndata-2)
```

- The lower limit on the difference will be the mean value minus tfactor * the uncertainty in the difference in means:

```
print('1-sided Confidence Interval / Lower Limit: ',???)
```

- Are the data consistent with a hypothesis of no difference?

Bayesian analysis

- Alternatively, we could assess whether there is a difference by applying Bayes' theorem:

  $p(\text{model} \mid \text{data}) = p(\text{data} \mid \text{model})\, p(\text{model}) / p(\text{data})$

- Our model will be that we have two datasets, x_i and y_j, described by Gaussians with the same σ and means μ_1, μ_2. We then want to test whether $\mu_2 > \mu_1$.

- The likelihood will be:

$$\prod_i (1/(2\pi\sigma^2)^{1/2}) \exp(-(x_i-\mu_1)^2/2\sigma^2) \prod_j (1/(2\pi\sigma^2)^{1/2}) \exp(-(y_j-\mu_2)^2/2\sigma^2)$$

- We'll use Jeffreys priors for μ_1, μ_2 and σ.

Bayesian analysis

- Then one can show that:

$$p(\mu_1, \mu_2, \sigma \mid \text{data}) \propto \sigma^{-(2n+1)} \exp(-A/\sigma^2)\text{, where}$$

$$A = \sum_{i=1}^{n} (x_i-\mu_1)^2/2 + \sum_{j=1}^{n} (y_i-\mu_2)^2/2$$

- To test if $\mu_2 > \mu_1$, we can change variables to $u = \mu_1$ and $v = \mu_2-\mu_1$, and test whether $v>0$.

- To do this, we need to **marginalize** over σ and u to get the probability distribution for v alone.

$$p(v \mid \text{data}) = \int\int p(u, v, \sigma \mid \text{data})\, d\sigma\, du$$

$$= 2 \int \Gamma(n)\, A^{-n}\, du$$

Bayesian analysis

$$p(v \mid \text{data}) = \int 2\,\Gamma(n)\,A^{-n}\,du \qquad A = \sum_{i=1}^{n} (x_i-\mu_1)^2/2 + \sum_{j=1}^{n} (y_i-\mu_2)^2/2$$

$$= \sum_{i=1}^{n} (x_i-u)^2/2 + \sum_{j=1}^{n} (y_i-(u+v))^2/2$$

```
# Only set up the A array if we haven't calculated it before -- this can be slow.
try:
    A
except NameError:
```

Set up a grid in u and v:

```
nbin=501
u = np.linspace(-5.,5.,nbin)
v = np.copy(u)
```

Bayesian analysis

$$p(v \mid \text{data}) = \int 2\,\Gamma(n)\,A^{-n}\,du \qquad A = \sum_{i=1}^{n} (x_i-\mu_1)^2/2 + \sum_{j=1}^{n} (y_i-\mu_2)^2/2$$

$$= \sum_{i=1}^{n} (x_i-u)^2/2 + \sum_{j=1}^{n} (y_i-(u+v))^2/2$$

Calculate *A* on that grid.

```
A=np.zeros(nbin,nbin)
for i in arange(nbin):
    for j in arange(nbin):
        A[j,i]=(np.sum((data1-u[i])**2)+np.sum((data2-u[i]-v[j])**2))/2.
```

Note: we could make the code faster by making **nbin** x **nbin** x **ndata** arrays containing the value of **data1**, **data2**, **u** or **v** at each pixel using **np.meshgrid** and **np.tile**, and summing over the ndata direction).

Bayesian analysis

$$p(v \mid \text{data}) = \int 2\,\Gamma(n)\,A^{-n}\,du$$

- We can now calculate the probability of any values of u and v, $p(u,v)$, from A :

```
from scipy.special import gamma
prob_uv=2*gamma(ndata)*A**(-ndata)
```

- We would like to now inspect the value of $p(u,v)$ as a function of the two variables. For that we need to be able to display images.
- We will do this using the Bokeh package. Bokeh allows you to make interactive displays of plots (e.g., with the ability to zoom in or out) in 2 or 3D, images, etc. This can be very useful for exploring your data!
- For the Bokeh website, see https://bokeh.org/ . Tutorials are available from the tutorial link at the top of that page.

Viewing an image in Bokeh

```
# Do imports for bokeh
from bokeh.plotting import figure, output_file, show, output_notebook
from bokeh.models.mappers import
         LinearColorMapper,LogColorMapper,EqHistColorMapper
 if 0:
# Bokeh command telling it to put the plot in this notebook
     output_notebook()

# Set up tooltips from Bokeh so we can read off values
#       wherever we point the cursor
     p = figure(tooltips=[("u", "$x"), ("v", "$y"), ("value", "@image")])
     p.x_range.range_padding = p.y_range.range_padding = 0
```

Viewing an image in Bokeh

```
# Choose logarithmic color scaling; you could instead try the linear or EqHist color
mappers
    color_mapper = LogColorMapper(palette="Turbo256", low=prob_uv.min(),
high=prob_uv.max())

# Set up the image display, with axis ranges from -5 to +5
# dw and dh set the plot size within the notebook
    p.image(image=[prob_uv], x=-5, y=-5, dw=10, dh=10, level="image",
            color_mapper=color_mapper)

# show the image, interactively
    show(p)
```

Viewing an image without Bokeh

```
# list color maps available in matplotlib
from matplotlib import colormaps
list(colormaps)

# display an image of prob_uv, with the plasma color map

if 0:
    plt.imshow(np.log(prob_uv),cmap='plasma',extent=(-5,5,-5,5))
    plt.xlabel('u')
    plt.ylabel('v')
```

Now we want to plot $p(u)$ and $p(v)$

```
if 0:
# Set up tooltips from Bokeh so we can read off values wherever we point the
cursor,
#   if we point close enough to the curve.  Set plot width/height.
    p = figure(width=400,height=200,tooltips=[("v", "$x"), ("p(v)", "$y")])
    p.x_range.range_padding = 0
    p.y_range.range_padding = 0.1

# Plot the figure.   p.line(v,prob_v,line_width=2,legend_label='p(v)',color='blue',alpha=0.5)
p.line(u,prob_u,line_width=1.5,legend_label='p(u)',color='red',
    line_dash='dashed')
# Show the plot, interactively
    show(p)
```

- Or see the notebook for matplotlib equivalent...

Things to look for

- **Evaluate by eye and discuss with your group:**
- **Are the Bayesian results peaked where you expect**? Note that $\mu_1=0$ corresponds to $u=0$, and $\mu_2-\mu_1=1$ corresponds to $v=1$.
- **Are the Bayesian results consistent with the true values** (i.e., is the probability nonnegligible at those values of u and v?)?

Bayesian analysis

$$p(v \mid \text{data}) = \int 2\,\Gamma(n)\,A^{-n}\,du \qquad A = \sum_{i=1}^{n} (x_i-\mu_1)^2/2 + \sum_{j=1}^{n} (y_i-\mu_2)^2/2$$

$$= \sum_{i=1}^{n} (x_i-u)^2/2 + \sum_{j=1}^{n} (y_i-(u+v))^2/2$$

- Now, it's time to evaluate our results as Bayesians.
- To what degree do we believe that v>0?

```
wh=np.where(v > 0)
p_gt_0=np.sum(prob_v[wh])/np.sum(prob_v)
```

- Calculate the posterior probability that v>0 and the odds of that proposition:

```
print(p_gt_0,p_gt_0/(1-p_gt_0))
```

How does this compare to the confidence intervals?

- We earlier got confidence intervals for μ_2-μ_1 from the observed means and standard errors: e.g.,

```
signif=0.05
tfactor=stats.t.ppf(1-signif/2, ndata-1)
sigma_diff=sqrt(sigma1**2+sigma2**2)
print('CI diff',(mean2-mean1)-tfactor*sigma_diff,(mean2-mean1)+tfactor*sigma_diff)
```

- What significance level do we need to go to for the confidence interval for μ_2-μ_1 to barely include 0? Adjust `signif` and see.

Is there any correspondingly sensitive frequentist method?

- We've seen that when we incorporate everything we know about the two probability distributions in a Bayesian formalism, we got stronger constraints on whether the two samples differ than we got from the confidence intervals.
- Is there a frequentist method that has similar power (= probability of ruling out the null hypothesis when it is false) as the Bayesian analysis?

The Likelihood Ratio Test

- If we ignore our prior on σ (which should have only a tiny effect here), $p(u, v, \sigma \mid \text{data})$ would be identical to the likelihood, $p(\text{data} \mid u, v, \sigma)$.
- We can then calculate the maximum likelihood value across all the cases where H_0 holds (i.e., when $v=0$), and the maximum where H_1 holds (when $v > 0$); we can then define the likelihood ratio Λ as:
  $\Lambda = \max (L (\text{data} \mid H_0) \; / \; \max (L (\text{data} \mid H_1))$

  where L denotes the likelihood. It is calculated from the data, and hence is a statistic.
- If H_0 has k fewer free parameters than H_1 (here, $k=1$), it turns out that $-2 \ln \Lambda$ should be approximately distributed as a chi-squared distribution with k degrees of freedom.