Machine Learning

Statistics and Data Science

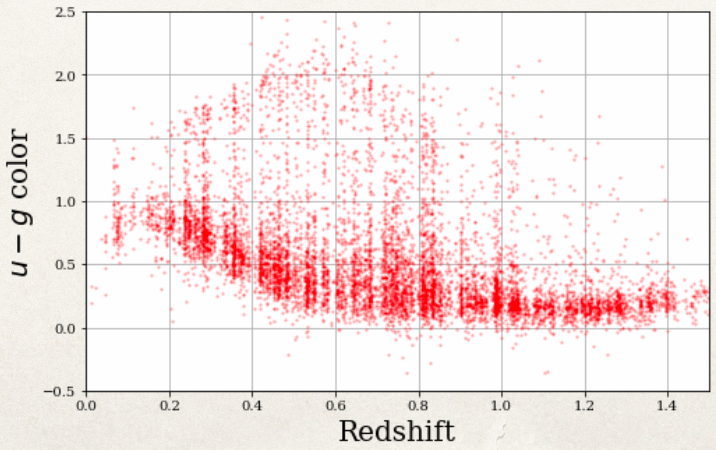
Spring 2025

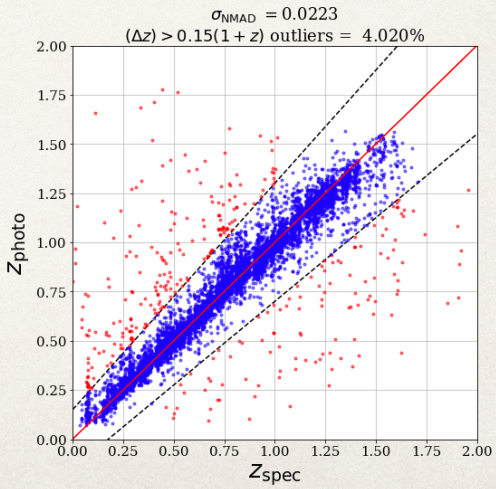
Goals for today: you should be able to...

- **lecture 15/16 notebook**:
  - Apply machine learning methods for classification and regression
- Propagate errors in one quantity to errors in another quantity
- Outline the process of hypothesis testing

Review: the problem we are working on: Photometric Redshifts

- Redshift (z) measurements allow us to determine how far back in Universe's history we are looking
- ***Photometry*** = measuring how bright something is
- ***magnitude*** is a measure of flux: e.g.: r = r-band magnitude = -2.5 log10 (r flux / r_0)
- In astronomy, ***color*** is the difference between magnitudes in 2 filters (with bluest specified first)
  - e.g.: g - r = g-band magnitude - r-band magnitude = - 2.5 log10 (g flux / r flux)





Where we left off: do not train and test with the same data!

```
if 0:

    # use the RF regressor trained on the training set, but
    #   apply it to the training set instead of the test set
    z_phot = regrf.predict(data_train)
    z_spec = z_train

    plot_and_stats(z_spec,z_phot)
```

- **How do the statistics (NMAD + outlier rate) differ when evaluated on the training set, as compared to when using an independent test set?**

Cross-validation

- With a 50-50 training/testing split, we are always training significantly more poorly than we would with the full dataset, and can only use half the data to evaluate how well we are doing. ***K-fold cross-validation*** provides a way around this.
- In k-fold cross-validation, we split the data into k subsets. We loop over the subsets, training with all but one and testing with the other; in the end, we get the performance of training with a fraction *(k-1)/k* of the data, but are able to get test statistics based on the entire dataset.
- This is easy to do in `scikit-learn`, but does require running the training ***k*** times ...
  - Note: we can search multi-dimensional grids of ML algorithm parameters in an automated way and optimize parameters with cross- validation with `sklearn.model_selection.GridSearchCV`; see https://scikit-learn.org/stable/auto_examples/model_selection/plot_grid_search_digits.html for an example.

Walking through the code: cross-validation

```
if 0:

    # use the RF regressor trained on the training set, but
    #   apply it to the training set instead of the test set
    z_phot = regrf.predict(data_train)
    z_spec = z_train

    plot_and_stats(z_spec,z_phot)
```

- **Compare the performance (NMAD, outlier rate...) from random forest regression with 5-fold cross-validation vs. training with only 50% of the sample.**

What does the plotting code do?

```
#This is a function that makes a plot of photometric redshift
#    as a function of spectroscopic redshift
#    and calculates key statistics.  It will save us a lot of work.
def plot_and_stats(zspec,zphot):

    x = np.arange(0,5.4,0.05)

# define differences of >0.15*(1+z) as non-Gaussian 'outliers'
    outlier_upper = x + 0.15*(1+x)
    outlier_lower = x - 0.15*(1+x)

    mask = np.abs((z_phot - z_spec)/(1 + z_spec)) > 0.15
    notmask = ~mask

#Standard Deviation of the predicted redshifts compared to the data:
    std_result = np.std((z_phot - z_spec)/(1 + z_spec), ddof=1)

#Normalized MAD (Median Absolute Deviation):
    nmad = 1.48 * np.median(np.abs((z_phot - z_spec)/(1 + z_spec) -np.median((z_phot - z_spec)/(1 + z_spec))))

#Percentage of delta-z > 0.15(1+z) outliers:
    eta = np.sum(np.abs((z_phot - z_spec)/(1 + z_spec)) > 0.15)/len(z_spec)

    #Median offset (normalized by (1+z); i.e., bias:
    bias = np.median(((z_phot - z_spec)/(1 + z_spec)))
    sigbias=std_result/np.sqrt(0.64*len(z_phot))
```

What does the plotting code do?

```
# make photo-z/spec-z plot
plt.figure(figsize=(8, 8))

#add lines to indicate outliers
plt.plot(x, outlier_upper, 'k--')
plt.plot(x, outlier_lower, 'k--')
plt.plot(z_spec[mask], z_phot[mask], 'r.', markersize=6,  alpha=0.5)
plt.plot(z_spec[notmask], z_phot[notmask], 'b.',  markersize=6, alpha=0.5)
plt.plot(x, x, linewidth=1.5, color = 'red')
plt.title(f'NMAD:  {nmad:6.4f} Delta z >0.15(1+z) outlier rate:{eta*100:6.3f} %', fontsize=18)
plt.xlim([0.0, 2])
plt.ylim([0.0, 2])
plt.xlabel(r'$z_{\mathrm{spec}}$', fontsize = 27)
plt.ylabel(r'$z_{\mathrm{photo}}$', fontsize = 27)
plt.grid(alpha = 0.8)
plt.tick_params(labelsize=15)
plt.show()
```

Warning: ML methods extrapolate poorly!

```
if 0:
    # split the sample at the median magnitude; note that
    #      smaller magnitude means brighter!
    is_bright = r_mag < 23.15

    # we can use a logical statement as a mask to select only
    #       those array elements where it is true
    data_bright = data_colmag[is_bright]
    z_bright = data_z[is_bright]

    # or negate it to select where it is false
    data_faint=data_colmag[~is_bright]
    z_faint = data_z[~is_bright]

    #train the regressor with the bright data
    regrf.fit(data_bright,z_bright)

    # run on the faint test set
    z_phot = regrf.predict(data_faint)
    z_spec = z_faint

    plot_and_stats(z_spec,z_phot)
```

- In the notebook we present a variety of scenarios in which the training & test data differ in brightness, color, and redshift.
- **Assess: which of these causes the worst problems / worst predictions?**

Random Forest for classification

```
if 0:
    from sklearn.ensemble import RandomForestClassifier

    # set up the classifier object
    classrf = RandomForestClassifier(n_estimators=50)

    # fit a classifier intended to separate objects at
    #       redshift > 0.75 from those at < 0.75
    classrf.fit(data_train,z_train > 0.75)

    # predict the classifications for the test set
    class_predict = classrf.predict(data_test)

    # test which objects are selected as being at high redshift
    ishiz = class_predict == True

    #plot redshift histograms for each sample
    bins = np.linspace(0,1.5,150)
    a,b,c=plt.hist(z_test[ishiz],bins=bins,histtype='step',label = 'High z')
    a,b,c=plt.hist(z_test[~ishiz],bins=bins,histtype='step',label = 'Low z')
    plt.xlabel('Redshift')
    plt.legend()
```

- Classification with machine learning techniques in **`scikit-learn`** works much like regression; but the target array will consist of True and False, instead of a continuous variable.

Optimizing parameters of a scikit-learn algorithm

```
from sklearn.metrics import mean_squared_error,median_absolute_error
if 1:
    ntree_test = 10,25, 50,100
    mse=np.copy(ntree_test)*0.
    mad=np.copy(ntree_test)*0.

    for i,n in enumerate(ntree_test):
        # define the RF object with our choice of number of trees
        regrf = RandomForestRegressor(n_estimators = n, \
                                      max_depth = 30, max_features = 'auto')
        # do training AND prediction on the whole sample
        #     using cross-validation
        predicted = cross_val_predict(regrf,data_colmag,data_z, cv=5)
        #calculate mean squared error
        mse[i]=mean_squared_error(data_z,predicted)
        #calculate MAD
        mad[i]=median_absolute_error(data_z,predicted)

    plt.plot(ntree_test, mse,label='MSE')
    plt.plot(ntree_test, mad,label='NMAD')
    plt.legend()
    plt.xlabel('Number of trees')
```

- In general, it is best to test whether your results could be improved by changing the basic free parameters of the algorithm. You do need to decide what kind of loss you want to optimize though...

Things to keep in mind when applying machine learning methods

- Machine Learning algorithms can be very good at handling data like that which they were trained on, but can extrapolate very poorly
  - In the notebook, you can see what happens when you test with data that is identical to the training set, and also for data that is systematically different in at least some way
  - Always be sure to think about whether the data you will be applying the algorithm to will really match the training data!
- Imbalanced training sets can also be an issue: if 99.9% of the training data is in one class, always outputting that class as the solution could yield a small loss and get favored...

Propagation of errors

- Suppose we know the uncertainty in one quantity. Can we predict the uncertainty in related quantities? This is a crucial question for experiment design!
- As an example, suppose we measure quantity x, but really want to know quantity y. How small do the errors in x need to be to make the errors in y small enough?

Propagation of errors

- Suppose we make measurements of variable x, giving us estimates of the mean & standard deviation, μ & σ, of the Normal distribution it is drawn from.

- What will be the probability distribution for $y = x + b$, where b is a constant? It must be the case that:

  $\text{mean}(y) = \boldsymbol{E}(y) = \boldsymbol{E}(x+b) = \int f(x)(x+b)dx = \boldsymbol{E}(x) + b = \mu + b$

  $\text{variance}(y) = \boldsymbol{E}((y-(\mu+b))^2 = \boldsymbol{E}((x+b-(\mu+b))^2 = \boldsymbol{E}(x-\mu)^2 = \sigma^2$

- so if x is described by $N(\mu,\sigma^2)$, then y is described by $N(\mu+b,\sigma^2)$.

Propagation of errors

- Similarly, what will be the probability distribution for y = ax, where a is a constant? It must be the case that:

  $\text{mean}(y) = \boldsymbol{E}(y) = \boldsymbol{E}(ax) = \int f(x)(ax)dx = a\boldsymbol{E}(x) = a\mu$

  $\text{variance}(y) = \boldsymbol{E}((y-(a\mu))^2 = \boldsymbol{E}(ax-a\mu)^2 = \boldsymbol{E}a^2(x-\mu)^2 = a^2\,\boldsymbol{E}(x-\mu)^2 = a^2\sigma^2$

- so: if x is described by $N(\mu,\sigma^2)$, then y is described by $N(a\mu,a^2\sigma^2)$.

- In fact, if $y=ax+b$, then if x is described by $N(\mu,\sigma)$, y will be described by $N(a\mu+b,a^2\sigma^2)$; if x is Normally-distributed, so will y be.

The more general case

- Suppose y is some function of x: $y=g(x)$.
- Then (if conditions like differentiability hold) Taylor's theorem tells us that y can be accurately described over some small interval about some value of x, x_0, as just a linear function of x :

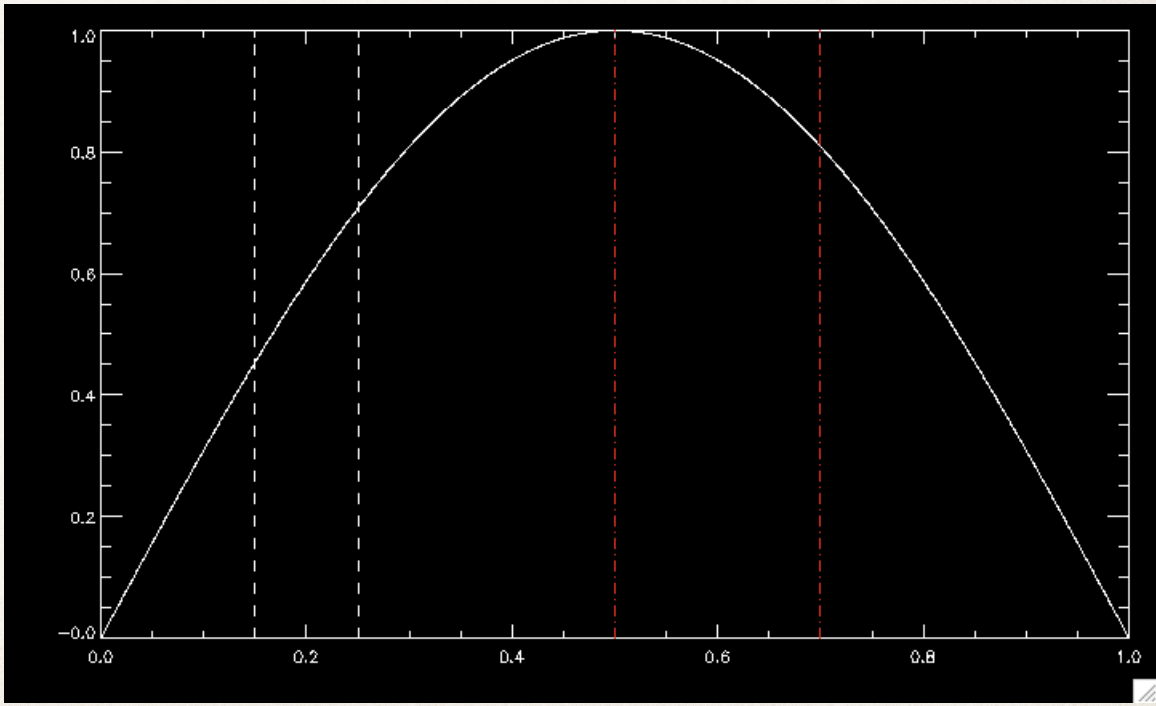
$$y \approx g(x_0) + (dg/dx)(x-x_0)$$

Then $(y-g(x_0))=y-y_0 \approx a\,(x-x_0)$; so if $x_0=\mu$, $(x-x_0)$ is distributed as $N(0, \sigma^2)$, and $y-y_0$ will be approximately described by:

$$N(0, a^2\sigma^2)=N(0, (dg/dx)^2\,\sigma^2)$$

The more general case

- Via Taylor's theorem, this will be accurate so long as $(d^2g/dx^2)\ \sigma^2$ is small - so if the 2-sigma range is the one shown by the white dashed lines, this as an excellent approximation, while if it's the red dashed lines, the accuracy will be less.



Example:

- Suppose we have estimated the brightness of a "standard candle" - an object whose luminosity we know - with 10% error and want to infer its distance. What will the fractional uncertainty in the distance be?
- brightness ∝ Luminosity / distance^2, so $d \propto b^{-1/2}$
- Let $d = a\, b^{-1/2}$. Then $\mathrm{d}d/\mathrm{d}b = -a/2\ b^{-3/2}$, so $\sigma(d) = a/2\ b^{-3/2}\,\sigma(b)$; and so

$$\sigma(d)/d = \frac{\frac{a}{2} b^{-\frac{3}{2}} \sigma(b)}{a b^{-\frac{1}{2}}} = 1/2\ \sigma(b)/b$$

- Hence, if we know b to 10%, we have determined the distance to 5%, roughly.

More generally:

- Let f be some function of a set of variables x_j.
- If all of the variables except for x_i are constant, the previous expression gives $\sigma^2(f)=(df/dx_i)^2\,\sigma^2(x_i)$
- Recall that, if x and y are independent, Normally distributed variables, $x+y$ is Normally distributed with $\sigma^2(x+y)=\sigma^2(x)+\sigma^2(y)$; so, holding one variable at a time constant, we find:

$$\sigma^2(f) = \sum_i \left(\frac{\partial f}{\partial x_i}\right)^2 \sigma^2(x_i)$$

- e.g., if $f=x_1x_2$, then $\sigma^2(f) = x_2^2\,\sigma^2(x_1) + x_1^2\,\sigma^2(x_2)$

Let's work out a more complicated case...

- The rate at which cosmic density fluctuations grow is proportional to $f = \Omega_m^{\gamma}$, where Ω_m is the density of matter in the Universe relative to the critical density at which the geometry of the Universe is flat, and $\gamma = 0.56$ (regardless of other cosmological parameters) if General Relativity is correct.
- If we make a measurement of f accurate to 10% (so $\sigma_f / f = 0.1$), what will be the uncertainty in γ (i.e., σ_γ)?
- **Solve this symbolically, then assume $\Omega_m = 0.3$ and evaluate.**

Some standard cases

- The most useful case is probably $f=x^A y^B \ldots$
- Then you can show that $(\sigma_f/f)^2=A^2(\sigma_x/x)^2+B^2(\sigma_y/y)^2+\ldots$
- Hence, in the single-variable case, the fractional error in f will be A times worse than the fractional error in x, etc. E.g. for $d \propto b^{1/2}$, we found $\sigma(d)/d = 1/2\ \sigma(b)/b$

$f = aA^{\pm b}$	$\frac{\sigma_f}{f} = b\frac{\sigma_A}{A}$
$f = a\ln(\pm bA)$	$\sigma_f = a\frac{\sigma_A}{A}$
$f = ae^{\pm bA}$	$\frac{\sigma_f}{f} = b\sigma_A$
$f = a^{\pm bA}$	$\frac{\sigma_f}{f} = b\ln a\sigma_A$

Source: wikipedia.org

Application: Error in the weighted mean

- Suppose we have a weighted mean, so $f=\Sigma\, w_i\, x_i$, where w_i is the weight applied to the ith value, x_i.
- Then by propagation of errors $(\sigma_f)^2=\Sigma w_i^2(\sigma_i)^2$
- Typically, we will use weights proportional to $\frac{1}{\sigma_i^2}$, as that is the optimal weighting for the mean of measurements with different errors (which we found before).
- For a mean, we want $\Sigma\, w_i=1$, so $w_i = \frac{\frac{1}{\sigma_i^2}}{\Sigma \frac{1}{\sigma_i^2}}$
- Then $(\sigma_f)^2=\Sigma w_i^2(\sigma_i)^2 \quad = \quad (\frac{1}{\Sigma \frac{1}{\sigma_i^2}})^2 \sum \frac{1}{\sigma_i^4}\sigma_i^2 \quad = \quad \frac{1}{\Sigma \frac{1}{\sigma_i^2}}$

Note that, if all the $\sigma_i=\sigma$, this gives $(\sigma_f)^2=\sigma^2/n$

Hypothesis Testing

Applying statistics to answer questions

- We've talked about how to measure and interpret a variety of statistics. How do we apply them to answer scientific questions?
  - Does my measurement agree or disagree with the results of X from the literature?
  - Are two sets of measurements (e.g. the color or mass distributions of populations of different types of galaxies) detectably different, or not?
  - Is the observed data consistent with some theoretical model?
- These are all problems of hypothesis testing.

Methods of hypothesis testing

- As in previous cases, hypothesis testing works somewhat differently in Bayesian and Frequentist approaches:

- **Bayesian**: Given the observed data and what I know about the underlying distributions, what are the odds that X is true?
- **Frequentist**: If X were true, what is the probability we would measure the value of some particular statistic observed, or any value more extreme?

Parametric vs. nonparametric methods

- As when we were defining confidence intervals, where an X% credible interval is just the portion of the posterior distribution containing X% of probability, Bayesian tests are generally easier to derive/write down immediately from knowing the distributions we're dealing with.
- However, there exist tests of hypotheses based on a frequentist formalism that do not rely on knowing anything about underlying distributions, called ***nonparametric tests.***
- There are a variety of hypotheses we can test with only minimal assumptions in the Frequentist formalism that are difficult to test in a Bayesian way.

Classical (frequentist) hypothesis testing

1) Define two mutually exclusive hypotheses: a *null hypothesis*, H_0, and an *alternative* or *research* hypothesis, H_1. Our goal is generally to test whether the data is consistent with the null hypothesis - so we must be able to calculate the probability distribution of a statistic of interest, **if** H_0 is true.

- In medicine, say, it's generally simple to define these: e.g., H_0= a particular medicine causes no significant improvement, and H_1 = it does improve things.
- In physics and astronomy, H_0 might be that the predictions from a particular model are consistent with the data, or that some factor (like the environment a galaxy or bacterium is located in) makes no difference to the properties we observe.

Null hypotheses

- Many classic statistical tests use a null hypothesis that there is no difference between 2 (or N) samples / distributions.
- We can generally apply those tests even if we want to test whether the difference is consistent with some specific nonzero value (say, 3) - we'd then just subtract 3 from the values for the 2nd sample, and use a test for whether we are consistent with no difference.
- Basically, the null hypothesis is the statement that we got the results we see purely by random chance, and do not need any additional explanation.

Classical (frequentist) hypothesis testing

2a) Define the ***significance level* α** we will use to decide if H_0 will be excluded (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).

- *Then* decide on an appropriate statistic we can use to test H_0 (e.g., the difference in the mean environment [density of surrounding galaxies] for two samples, if H_0 is that environment makes no difference).
- **It is necessary to decide on α before looking at the results of any tests.** It can be very tempting to change the value of α to match the results of a test, or to keep doing tests till one gives the result we expect/are looking for.

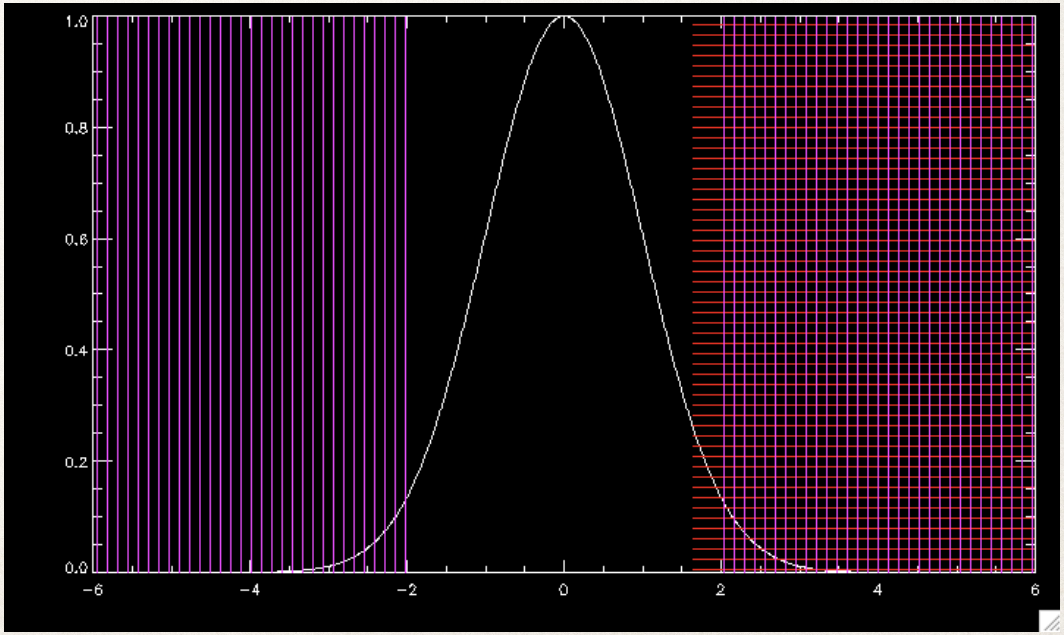
Classical (frequentist) hypothesis testing

- 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).

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

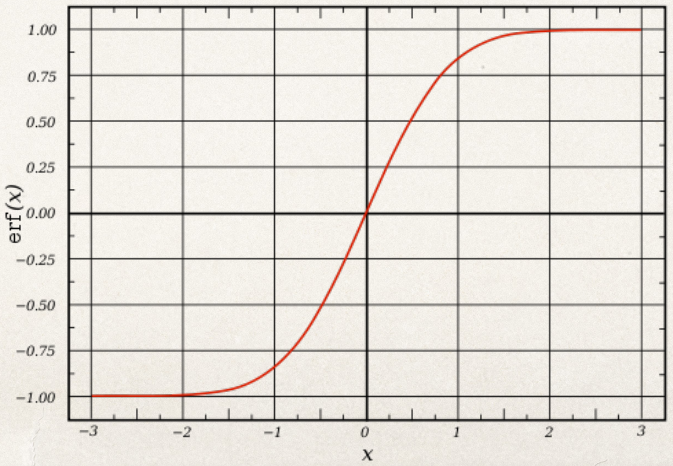
- erf(x) is closely related to the integral of a Gaussian; the integral from -∞ to x is

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

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?