FYS4505/9505 - Uncertainties

Autumn term 2018

Problem 1: General "quizz" on uncertainties

Which of the following numbered methods would you use to assign an uncertainty in each of the following values?

1) A theoretical (statistical) calculation
2) The standard deviation
3) The uncertainty propagation formula
4) The uncertainty value given in the publication
5) Reading accuracy of the instrument

a) The voltage of a pulse measured on your oscilloscope. (Assume you do not have a manual.)

b) The average of 7 measurements you made.

c) The half-life of 14 C, as reported in an journal article about the measurement.

d) The fraction of voters who will vote for candidate X, based on a random sample of 1000 voters.

e) The counting rate of a source, given the measured counts and the counting interval.

f) The net counting rate, given the measured counts with source in place, the measured background counts, and the counting intervals.

Problem 2: Notation

Report following values with two significant figures including the uncertainty in parenthesis notation, ie. $2.31 \pm 0.05 = 2.31(05)$.

a) 0.002743 ± 0.001077

b) $5.8762318x10^{-23} \pm 1.222x10^{-27}$

c) 35.23074 ± 1.732

d) 153079 ± 2734

Problem 3: Probability distributions

Probability distributions describe the probability of observing an event. There are several distributions that are important in physics, but in this problem we will focus on the Binomial, Normal and Poisson distributions.

a) Describe each of the probability distributions mentioned. The main goal here is to be familiar with them, not to write all there is of information about them.

b) Find data for each and fit it to the correspoding distributions. What happens if you try to fit a Gaussian dataset to a Poisson distribution, and how does the mean value change.

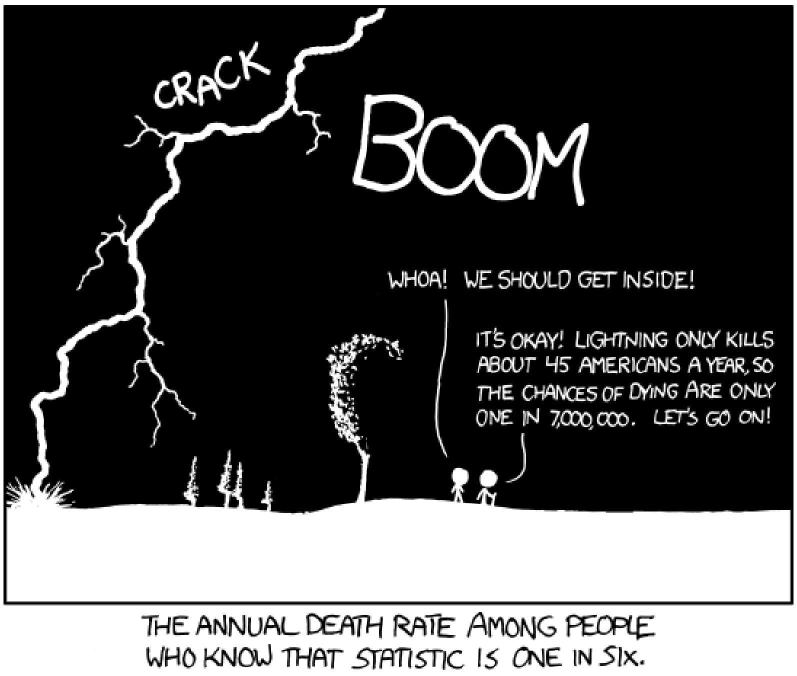
CRACK
BOOM
WHOA! WE SHOULD GET INSIDE!
IT'S OKAY! LIGHTNING ONLY KILLS ABOUT 45 AMERICANS A YEAR, SO THE CHANCES OF DYING ARE ONLY ONE IN 7,000,000. LET'S GO ON!
THE ANNUAL DEATH RATE AMONG PEOPLE WHO KNOW THAT STATISTIC IS ONE IN SIX.

Figure 1: `https://xkcd.com/795/`

Problem 4: Calculating with uncertainties

1) To start with, assume normal distributed uncertainties with no correlation. Derive the necessary relations and calculate the uncertainties on following values starting from following eq. 9.9 in the attached file on "correlation and uncertainties". [It's the "common" error propagation using the derivatives formula]

   a) $a + b$, where $a = 25(5)$ and $b = 3.11(11)$
   b) $a * b$, where $a = 25(5)$ and $b = 3.11(11)$
   c) $log(a) * b^2$, where $a = 25(5)$ and $b = 3.11(11)$

2) Five students measure two angles θ and ϕ in degrees, and obtains the result

Student	θ	ϕ
1	35	50
2	31	55
3	33	51
4	32	53
5	34	51

   1) Find the average and standard deviation for each of the two angles.
   2) Calculate the covariance $\sigma_{\theta\phi}$ and correlations.
   3) From the above results, calculate the mean of the sum of the two angles q=θ+ϕ and its standard deviation σ_q.
   4) What would happen with the standard deviation(s) if we assumed that the measurement between θ and ϕ were independent?

Problem 5: A typical application – fitting a line to data

From your measurement you derived following four independent data points
(x, y): (2,3), (7,6), (9,5), (10,11). Each of them has a relative uncertainty of 20%. You believe that these points should be fitted by a line, so $y = ax + b$.

Fit the data and create a plot where you show i) the data incl. uncertainty ii) the best fit iii) the confidence interval on the best fit. What do we mean by the latter: Well, your fit has some uncertainty, and we're interested in a graphical representation of this.

Hint: Remember to take into account correlations between your fit parameters. You either derive the uncertainties "by hand" (well, using a self-derived formula in your code), or you may use eg. *uncertainties* package to derive the values. In the latter case, quickly explain what formula it uses to derive the uncertainties. It's a quite handy tool I find. If you use it, simply define the parameters as correlated via the *correlated_values* method, and derive *nominal_values* and *std_devs* from derived quantities.

Problem 6: Realistic Data: The Giant Dipole Resonance

In both nuclear and particle physics, one is often interested in the properties of resonances. In this problem set you are provided with data from an experiment in 1976 (the data can also be accessed through the EXFOR database) and you are going to analyse the data, so that you can learn something about the resonance structure seen in the data.

a) Fit the GDR data for 239Pu with the method & program of your choice. Compare the parameters of the best fit to those of Gurevich(1976), and make sure to include the uncertainties in the plots. Plot the covariant matrix and describe what the plot tells about relation between the parameters. As can be seen in the article, a good choice for fit function f is a superposition of two *Standard Lorentzians*:

$$f_{SLO}(E) = \sigma_0 \frac{(E\Gamma)^2}{(E^2 - E_0^2)^2 + E^2\Gamma^2} \tag{1}$$

Note, that there is subtle differences between the Breit-Wigner distribution shown here, it's non-relativistic approximations and the Cauchy distribution, which is often also referred to as a Lorentz Distribution.

b) Use of Bayesian models: Eg. Monte-Carlo Markov Chains with emcee, or even pymc3 with the *No U-Turn Sampler* on the same problem. What are the correlations between the parameter? Hint: I would suggest to use emcee, as you may follow pretty much all the way along the example: `http://dfm.io/emcee/current/user/line/` - however, keep in mind that we have this easier version first. If you get hoocked, please tell me if you want to try and implement the same routines in pymc3 with a *No U-Turn Sampler*. The structure of your program could look like this (pseudo code):

```
data = import_data()

p_bounds = ... # priors for emcee;
def run_analysis(data):
    p0, pcorr = minimize(data) # some chi2/max-likelyhood
                               # to get initial parameters

    def lnlike(theta, x, y, yerr): # more flexible: lnlike(theta,*par)
        model = fSLO(...)
        # need to change this function!
        # -- see also next exercise
        lnlike = -0.5*chi2

    def lnprior(theta):
        # this means uniform priors withing [theta_min,theta_max], because
        # -> ln(const1)=const2; this is well defined for const1 > 0
        if theta is_in_bounds(p_bounds):
            return 0.
        else:
            return -np.inf

    def lnprob(theta, x, y, yerr): # more flexible: lnlike(theta,*par)
        # standard definition

    pos = p0 + 1e-4*random.randn(ndim) # initial position for emcee:
                                       # Gauss ball around p0
    sampler = run_emcee()
    # optional: plot_results(sampler)
run_analysis(data)
```

c) In the next section, we will assume that the data is actually "drawn"/discribed by the model stated above. How likely does this seems to be in the different energy regimes? What can be possible systematic discrepancies?

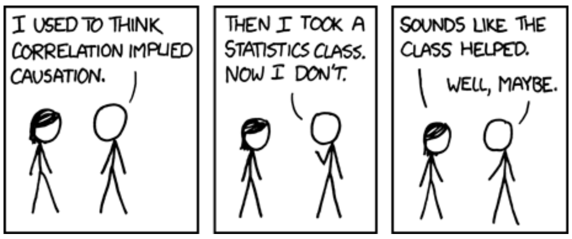
I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Figure 2: `https://xkcd.com/552/`

Problem 7 Bonus – this is where the fun starts: Generating your data & verifying an analysis method: Synthetic datasets

Introduction:
How can we be sure that the analysis method we proposed actually yields correct results? In this exercise we will generate and use synthetic datasets to verify the results form a minimization routine. For an interesting and funny discussion, I can strongly recommend the article **Data analysis recipes: Fitting a model to data** by Hogg et al.

Let's assume that we have a generative model of our data: "A generative model is a parameterized, quantitative description of a statistical procedure that could reasonably have generated the data set you have. In the case of the straight line fit in the presence of known, Gaussian uncertainties in one dimension, one can create this generative model as follows: Imagine that the data really do come from a line of the form $y = f(x) = mx + b$, and that the only reason that any data point deviates from this perfect, narrow, straight line is that to each of the true y values a small y-direction offset has been added, where that offset was drawn from a Gaussian distribution of zero mean and known variance σ_y^2."

Likewise, we assume that the data follow a superposition of two SLOs, with Gaussian uncertainties the cross-sections (i.e. y-direction). The justification of the definition of the lnlikelyhood is therefore derived between eq (9) and (11) in the work of Hogg et al. When K (see eq 11) is constant, likelyhood maximization is the same as χ^2-minimization.

Let's turn to the exercise:

Generate N artificial datasets and test the analysis method derived in the last section on it (both Frequentist and Bayesian). How can we achieve this?

- Pick a set of "true" input parameters p_{true}, to generate data following a function f. Then add the statistical noise, here: Gaussian uncertainty. Hint: The uncertainties should resemble the data, so eg. pick the same relative (x-grid and) uncertainties as given in the exp. dataset. Then add the uncertainties to the data to generate an artificial dataset.
- How could a test look like: First, make sure that the method "seem to work", eg. graphically that you seem to have enough iterations to the emcee runs. Then, we make use of the definition of the confidence (credibility) interval, which is loosely: If I repeat the measurement many times, I think that the true parameter(s) will be eg. within the 1 σ interval with a 68% probability. Check for each parameter separately, how good the estimated confidence (credibility) intervals encompass this criterion (for both methods). You may find out that you need to play a bit with the parameters of you fitting routine to get satisfactory results!

Some pseudo-code to facilitate this:

```python
def run_analysis(data):
        p0, pcorr = minimize(data) # some chi2/max-likelyhood
                                   # to get initial parameters
        ...
        sampler = run_emcee()
        p_mean, p_cred = GetCredibilityIntervals(samples)
        ...
        # arrays with entries True/False whether
        # p_true is within the quoted 1sigma limits
        h_freq = check_in_bounds(p0,pcorr,p_true)
        h_bay = check_in_bounds(p_mean,p_cred,p_true)
        return h_freq, h_bay

p0 = ...
p_bounds = ...
hist_freq = ... # to analyze whether p_true is
                # within the 1(2,3,...) sigma interval
hist_bay = ...
for i in range(N):
    data = generate_data(p_true)
    h_freq, h_bay = run_analysis(data)
    hist_freq += h_freq
    hist_bay += h_bay
hist_freq /= N
hist_bay /= N
print(hist_freq, hist_bay)
plot(hist_freq, hist_bay)
```

Problem 8 Still part of the Bonus: Under/Overestimated variance?

Analogously to the "line" example, we now want to inspect whether the reported uncertainties in the dataset are potentially under/overestimated by a constant fraction: $\sigma_{\text{quoted}} = r * \sigma_{true}$. So, say, maybe you think that the data points seem to be spread much more, than it seems reasonable when described by Gaussian uncertainties with the quoted sigma.

Test first, how well the method(s)[1] work.

- Verify, that the correct Likelihood is given by

$$\mathscr{L} = -\frac{1}{2}\sum_{i=1}^{N} \frac{(y_i - f(x_i,\theta))^2}{\sigma_{yi}^2} + \ln 2\pi\sigma_{yi}^2. \tag{2}$$

  For the prove you may follow eq. (9)–(11) in the paper of Hogg et al.. Why could we neglect the following term in the last exercise: $\ln 2\pi\sigma_{yi}^2$?

- Remember to adopt your data generation routine and the lnlike function. Remember to use the "correct(ed)" variances in the lnlike, so that if $\sigma_{\text{quoted}} = r * \sigma_{true}$, you build the χ^2 with $\sigma_{true} = \sigma_{quoted}/\Theta_r$.

How well does each of the methods work now? How meaningful are the results?

Problem 8.0.1 Full derivation: (= solution)

If the generative model is a function $f(x,\theta)$ with normal distributed uncertainties, the frequency distribution $p(y_i|x_i,\sigma_{yi},\theta)$ for a data point (x_i, y_i) is:

$$p(y_i|x_i,\sigma_{yi},\theta) = \frac{1}{\sqrt{2\pi\sigma_{yi}^2}} \exp\left(-\frac{(y_i - f(x_i,\theta))^2}{2\sigma_{yi}^2}\right), \tag{3}$$

"where this gives the expected frequency (in a hypothetical set of repeated experiments) of getting a value in the infinitesimal range $[y_i, y_i + dy]$ per unit dy" (Hogg2010). We attempt to find the function (parmeters θ), that maximize the probability of the observed data given the model – this is commonly called maximizing the likelihood of the parameters. If we assume that the data points are independently drawn, the likelihood $\mathscr{L}$ is given as the product of the conditional probabilities

$$\mathscr{L} = \prod_{i=1}^{N} p(y_i|x_i,\sigma_{yi},\theta) \tag{4}$$

Taking the logarithm,

$$\ln(\mathscr{L}) = \sum_{i=1}^{N} \ln\left(\frac{1}{\sqrt{2\pi\sigma_{yi}^2}}\right) + \ln\left[\exp\left(-\frac{(y_i - f(x_i,\theta))^2}{2\sigma_{yi}^2}\right)\right] \tag{5}$$

$$= \sum_{i=1}^{N} -\frac{1}{2}\ln \pi\sigma_{yi}^2 - \frac{1}{2}\frac{(y_i - f(x_i,\theta))^2}{\sigma_{yi}^2} \tag{6}$$

$$= -\frac{1}{2}\sum_{i=1}^{N} \frac{(y_i - f(x_i,\theta))^2}{\sigma_{yi}^2} + \ln 2\pi\sigma_{yi}^2 \tag{7}$$

Now, if the variances σ_{yi} are fixed, so they don't depend on and of the parameters in θ, the term $\ln 2\pi\sigma_{yi}^2$ will just be a constant, and we can write

$$\ln(\mathscr{L}) = K - \frac{1}{2}\chi^2. \tag{8}$$

[1]If you have used *scipy/curve_fit* up to now, you will need to change to another routine to minimize the likelyhood, eg. **minimize**, as the parameter r is not included in the function itself.

If the reported uncertainties are under(/over)estimated by a factor r, such that $\sigma_{quote} = r * \sigma_{real}$, we can try to find this factor by adding it to the parameters θ. Then we have to use (7), where we maximize the likelihood with the substitution $\sigma_{yi} \rightarrow \sigma_{quote}/r$. Note maximizing the likelyhood for

Further Reading/References

- Correlation_and_uncertainty_exampel.pdf
- A very funny and insightful article:
  David W. Hogg, Jo Bovy, and Dustin Lang. Data analysis recipes: Fitting a model to data
- More on paractical statistics, like graphical parameter estimation from χ^2 calculations:
  JV Wall. Practical statistics for astronomers-ii. correlation, data-modelling and sample comparison. *Quarterly Journal of the Royal Astronomical Society*, 37:519, 1996
  and
  J. V. Wall and C. R. Jenkins. Data modelling and parameter estimation: basics. In *Practical Statistics for Astronomers*, pages 126–150. Cambridge University Press, 2012
- On the data-set:
  G.M. Gurevich, L.E. Lazareva, V.M. Mazur, G.V. Solodukhov, and B.A. Tulupov. Giant resonance in the total photoabsorption cross section of z $\approx$ 90 nuclei. *Nuclear Physics A*, 273(2):326–340, November 1976
- emcee, the MCMC Hammer. Example ”Fitting a model to data” `http://dfm.io/emcee/current/user/line/`
- More on Bayesian statistics:
  Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL, 2014