Exercise 9 - Statistical modeling with maximum likelihood

Work with a partner to complete the tasks below and submit your results via a pull request on GitHub by the beginning of tutorial next Friday.

To begin this week, one of the partners should fork the TA's Exercise 9 Github repo and provide collaborative access to the other partner. Clone the forked repo so that you have the required files. Be sure to commit regularly to show how you arrived at your solutions and demonstrate coding effort by both partners.

1. In contrast to the study we read this week (Kelly *et al.* 2014), many laboratory experiments are designed to test the specific effect of a treatment with all other variables controlled. These experiments will have replicated treatment and control sets of experimental units. Often to test a hypothesis in this context, a **t-test** is used to evaluate whether there is a statistically significant difference in the mean of measurements taken from the treatment and control experimental units. One way to conduct a significance test within the maximum likelihood framework is with something called a **likelihood ratio test**. In this approach, the likelihood of a null model with a single parameter describing the mean behavior of all experimental units (`y=B0+error`) is compared to the likelihood of a model with an additional parameter that describes the difference between treatment and control groups (`y=B0+B1*treat+error`). If the difference in likelihood between the two models is large enough, we can say that our treatment had a statistically significant effect. We test for statistical significance by asking whether two times the difference in likelihoods (D) is large relative to a *chi-squared distribution* with one degree of freedom. This test can be accomplished with `pchisq(q=D,df=1,lower.tail=FALSE)` in `R` or `1-scipy.stats.chi2.cdf(x=D,df=1)` in `Python`.

Use a likelihood ratio test to determine which of three mutations significantly reduced the expression of *ponzr1*, a gene involved in the formation of the glomerulus in a developing zebrafish kidney. Messenger RNA copy counts for the experiment are available in "ponzr1.csv".

2. The maximum likelihood approach is very flexible when choosing both the deterministic component and the probability distribution. For this problem we'll stick with a normal distribution, but let's try a non-linear model. Growth rates of bacteria in culture can be modeled as a function of the concentration of a limiting resource provided. Generally, these relationships have a saturating relationship between growth rate and resource supply known as the *Monod Equation* – $\mu = \mu_{max}\frac{S}{(S+K_s)}$. Where μ is growth at a resource concentration S, μ_{max} is the absolute maximum growth rate and K_s is the resource concentration at which growth is half of maximum.

Given the data provided in "MmarinumGrowth.csv" estimate the maximum growth rate (μ_{max}) and the half-saturation constant (K_s).

3. As you'll see next week, when recreating biological processes in simulation models we often make strong simplifying assumptions. The degree to which we simplify the representation of a biological process in a simulation model is often guided by observational or experimental data. One way to quantify the simplicity, or conversely complexity, of a simulation model is the number of parameters used. Imagine we want to model decomposition of leaves (d) in the soils of a forest as a function of water availability (measured as soil moisture, M_s). For this aspect of our simulation model we want to keep the model as simple (i.e. fewest parameters) as possible, but also want to capture the "shape" of the relationship between decomposition and soil moisture.

Using the observations in "leafDecomp.csv" determine whether we should use a constant rate for all soil moistures ($d = a$), a linear response of decomposition to soil moisture ($d = a + bM_s$), or a hump-shaped response of decomposition to soil moisture ($d = a + bM_s + cM_s^2$).

Hint: You can use the likelihood ratio test to compare these models too. This is because the likelihood ratio test can be used to compare any set of models that are subsets of each other. You know one model is a subset of another if you can set a parameter equal to zero or some non-zero constant and have the same equation for both models. Also note that the degrees of freedom used in the *chi-squared distribution* is determined by the difference in the number of parameters between the two models. In the t-test example above, that will always be 1, but in this case it will be 1 or 2 depending on which models you are comparing.

Devise a plan for splitting up the work and generating the required code. Do this in parrallel, not sequentially. Don't forget to check and edit each other's code. Remember to frequently `add-commit` locally and `push-pull` to GitHub to avoid conflicts. Also, remember you don't have to be in the same place at the same time to work on this collaboratively thanks to GitHub!!!

Turning in your assignment via GitHub

Once you have committed all changes to your local Git repos and pushed all of those commits to the forked repo on GitHub, you can "turn in" your assignment using a `pull request`. This can be done from the GitHub repo website. When viewing the forked repo, select "Pull requests" in the upper middle of the screen, then click the green "New pull request" button in the upper right. You'll then see a screen with a history of commits for you and your collaborator, select the green "Create pull request button". In the text box next to your user icon near the top of the page, remove whatever text is there and add "owner's last name - collaborator's last name submission", but obviously substitute your last names. If I and Ann Raiho worked on the project together the text would read "jones-raiho submission". Then click the green "Create pull request" button. **Only one of you will need to create a pull request.**