# #homework (4 questions, A-D. Make sure to expand all bullets!)

- A: non-parametric smoothing (1)
  - Start by downloading and running the code for the cell cycle example in https://github.com/stephens999/stat34800/blob/master/analysis/cell_cycle.Rmd You may need to install some packages etc...
  - From the examples in the code (and, if desired, using further investigation of your own) settle on one version of the wavelet smoothing, and one version of the trend filtering. Try to select what you think will be the best -- or at least a satisfactory -- version of each.
    - Then, using some kind of cross-validation based approach, compare the accuracy of these two methods on the data from 10 genes that are provided in the repository. You are free to define your own measure of accuracy, but you should provide a report of what you did, and explain which method did best. Include plots that compare the results for both methods for all 10 genes.
  - Return now to our original goal. The question we want to answer is this: which genes show greatest evidence for varying in their expression through the cell cycle? For now we are not worried about "statistical significance" - it is enough to rank the genes. Using the results from the previous part, and some reasonable numeric criteria which you should explain, provide a ranking of the genes from the strongest evidence to weakest evidence. Check that your ranking by a numeric criteria seems reasonable (or, at least, not unreasonable) based on a visual assessment of the plots you made in the previous part.

choose one more accurate method

- B: non-parametric smoothing (2)
  - This question aims to get some intuition into trend filtering through simple simulations. It arises from me thinking "what circumstances might reduce the performance of trend filtering?". This is usually a good question to ask before applying a method!
  - 1. Constant mean case
    - Simulate some data with $y = u + e$ where:
    - `mu = rep(0,1000)` `
    - $e_j \sim N(0,1)$ independently $j = 1, \ldots, 1000$
    - Apply trend filtering to estimate the mean vector $\mu$. Plot the data with the estimate of $\mu$ overlaid.
  - 2. Step function
    - repeat 1, but with `mu = c(rep(0,500), rep(5,500))` . That is, the mean vector is a step function.

- You should find that in the second case (2) trend filtering finds the big change in mean accurately, but the estimate of $\mu$ in the regions where it is not changing becomes less smooth. (If this does not happen, try another random number seed.) Can you explain why?
  - Hint: you might get more insight by examining the results for multiple values of \(\lambda\) in each case, not only the "optimal" \(\lambda\) chosen by CV.

- C: Bayesian Calculations: Dirichlet multinomial
  - This goal here is to introduce you to the conjugate analysis for multinomial data. It might help to begin by refreshing your memory about the beta-binomial result, and then read an introduction to the Dirichlet distribution, which is a multivariate generalization of the beta distribution. I've also included a link to the multinomial distribution if you need a refresher on that.
    - https://stephens999.github.io/fiveMinuteStats/bayes_beta_binomial.html
    - https://stephens999.github.io/fiveMinuteStats/dirichlet.html
    - https://en.wikipedia.org/wiki/Multinomial_distribution
  - Let $X = (X_1, \ldots, X_k) \sim Mult(n, p)$ be a vector of multinomially-distributed count data on $k$ classes, where $n$ is a fixed and known sample size, and $p = (p_1, \ldots, p_k)$ is to be estimated. (So $X_1 + \cdots + X_k = n$ and $p_1 + \cdots + p_k = 1$).
  - Assume that the prior distribution for $p$ is Dirichlet$(\alpha_1, \ldots, \alpha_k)$. Show that the posterior distribution for $p$ is also Dirichlet, and find its posterior mean.
  - Now assume the special case $\alpha_1 = \alpha_2 = \cdots = \alpha_k$, and all are equal to $\alpha$ say. Write a function that, given data $X_1, \ldots, X_k$, first estimates $\alpha$ by maximum likelihood (you will need to use a numerical optimizer for this), and then returns the posterior mean of $p$ for that value of $\alpha$. [Essentially this is an Empirical Bayes approach to estimating $p$.]
    - Hint: the marginal distribution of \(X\) given \(n,\alpha\) (integrating out the vector \(\p\)) is known as the "Dirichlet-multinomial" distribution. You can read about it here: https://en.wikipedia.org/wiki/Dirichlet-multinomial_distribution, where a formula for the probability mass function \(p(X | \alpha)\) is given.

- D: density estimation
  - Following the ideas on cross validation marked #cv in the lecture notes above, write a function to automatically select an optimal bin-width for a histogram (where here optimal is measured in terms of its accuracy as a density estimator). You can assume the data are on $[0, 1]$ and that the bins are to be of equal width, spanning 0 to 1.

- Illustrate your function by plotting the resulting histogram for several simulated data sets. Use at least one dataset where the true density varies dramatically and one where the data have uniform density. Compare your results with the default of the `hist()` function in R. Comment on any good or bad features of your histograms compared with the default.
- One problem with a histogram as a density estimate is that if a bin $j$ contains no data then the density estimate is 0. (Note that this is not necessarily a problem for visualizing the data: if a bin contains no data, then it may be fine to show that! But as a density estimate it can be unsatisfactory to estimate the density as 0.) One possible solution is to replace the maximum likelihood estimate of $p_j$ used in constructing a histogram with the Empirical Bayes estimate (posterior mean) implemented in part C. Write a function to implement this modified density estimate, and see whether it improves the performance of the density estimate in your cross-validation experiment.