1. Introduction
2. Context/Background information
   1. overview of goodness of fit and different testing crietrions (one dimensional)
      1. analysis of how well each can differentiate between two samples from same distribution with small differences
      2. comparison of % of failed p values in exact sample
3. Requirements and analysis
   1. problem statement
   2. list of inference problems
4. design and implementation
   1. overview of the website
   2. overall structure of the code base
   3. weights
      1. how different one dimensional tests were adjusted for weighted case
   4. p value
      1. how p values are evaluated in one/two sample ks test
      2. how a single p value is extended to samples of p value (and percentage of which that has passed the test)
         1. how the initial sample is divided into either overlapping and non overlapping samples and its correlation
         2. (lack of) correlation between sample size and % passed
   5. basic sampling algorithms
      1. how importance sampling, mcmc, rejection algorithms were designed and how they work
      2. their relative performance with respect to distribution of p values (and % of p values above critical value)
      3. their relative performance in estimating the mean (time and first sample size to get the difference between true mean and estimation below some small number) of known distribution
         1. how basing the algorithm on different ‘easy’ distribution effects its performance
      4. strength and limitations of each algorithms
   6. overview of inference problems
   7. benchmark inference algorithm by pymc
      1. speculation/analysis of how pymc estimates posterior distribution
   8. how basic sampling algorithms could be expanded to inference algorithms
      1. their performance with respect to the benchmark with different problems
   9. multivariate statistical test
      1. how empircal cdf is calculated for n dimensional distribution
      2. two ways in which ks test was extended to n dimensional case and their limitations (need to ask Fred)
      3. how n dimensional distributions are tested in the code base (with respect to exact cdf/ exact sample)
   10. overview of the second problem
       1. benchmark problems to test the unknown inference algorithm
          1. why these problems were chosen (overlap with 4.8.1?)
       2. implementation of the second problem
          1. how the benchmark problems were chosen and how these can be used to determine how good user’s inference algorithm is in general (planned but not implemented yet)
5. testing
   1. testing the goodness of fit, i.e. does the return of S&P come from normal distribution?
   2. examples of tests with samples from pymc3 (or from anglican if time is availabe)
6. evaluations
   1. limitations
      1. lack of benchmark for certain n dimensional problems, problems limited to those with known exact solutions, etc
   2. future work
      1. sketch of database to store user’s results