Stat 624: Final

Complete all problems in a folder in the git repository labeled "Final". Have one R file with all the code for each problem clearly labeled and in order. Name this file MyFinal.R. You will also need an additional file for C or C++ code. Name that file ForFinal.c or ForFinal.cpp. No formal write-up is required.

Problem 1

For an exponential distribution with density function $f(x_i|\theta) = \frac{1}{\theta}e^{-x_i/\theta}$, the maximum likelihood estimate of θ is $\bar{x}$. The variance of data that follow the exponential distribution is θ^2, so another possible estimate for θ is $\sigma = \sqrt{\sum(x_i - \mu)^2/n}$. If an exponential likelihood was to be paired with an inverse gamma prior with parameters α and β, then the posterior distribution of $\theta|x$ has a distribution of an inverse gamma with parameters $\alpha + n$ and $\beta + \sum x_i$. Then the Bayesian estimate of θ under squared error loss is $(\beta + \sum x_i)/(\alpha + n - 1)$.

(a) Using C++, create an Rcpp function that takes an argument for the data and for the prior values of α and β and returns values for all three estimators. Source the code in R to be used for other parts of this project.

(b) Perform a simulation study to explore the bias and mean squared error of the three estimators. Use a true value for θ of 10 and simulated data sets of size 30. Use a prior of $\alpha = 3$ and $\beta = 10$ for the Bayesian estimate.

(c) Another way to establish estimator accuracy is using coverage of the estimator's confidence interval. Write a function that takes an argument for the data, the prior values of α and β, and the true value of θ. This function should return values in (0,1) indicating if the bootstrap confidence intervals contained the truth for all three estimators.

(d) Determine coverage probability using a simulation study. This may be more computationally time consuming than the previous simulation study, so permission officially granted to limit the number of any Monte Carlo iterations to 1000 for this part. Otherwise, use all the same fixed values as part (b) in the simulation. Asses Monte Carlo error of coverage probability.

(e) Make a conclusion about your findings in how it relates to how good the estimators are.

Problem 2

One model for population growth is

$$y_i = \frac{\beta_1}{1 + \exp(\beta_2 + \beta_3 x_i)} + \epsilon_i$$

where y_i is the population at time x_i and β_1, β_2, and β_3 are unknown constants. ϵ_i is normally distributed with mean 0.

(a) The data file pop.csv contains U.S. population in millions by year for consecutive years between 1961 and 2018. Using this data, fit the population growth model above to estimate values for β_1, β_2, and β_3. To do this, find the estimates that minimize the sum of squared residuals of the regression function. A few things will be very helpful to keep in mind. You don't need to code your own Newton Raphson, but if you use a built in R function such as optim, using the "BFGS" method and inputing your own gradient function is pretty much necessary. By using the gradient, starting values aren't as fickle, but β_1 should be greater than the maximum population or it fails. Try several different starting values until you feel that you've found good estimates.

(b) Project the population in 2020.

(c) We want to use bootstrapping to create a confidence interval for the projection in part (b), but bootstrapping assumes independence, which we do not have due to the natural time dependence of this the data. To fix this, a variant of bootstrapping is often used called block bootstrapping. Instead of resampling each individual data point, resample predefined blocks of consecutive data points. Follow this algorithm:

(1) Split the data into K blocks that are as close in size as possible

(2) Sample K of the blocks with replacement to make a bootstrapped data set.

(3) Fit the model with the bootstrapped data set

(4) Repeat steps 2 and 3 a large number of times.

In our context, for each bootstrapping iteration, estimate the parameters and compute the projection so you can get uncertainty about the projection. Use 10 blocks. This is an approximation method, so don't get too caught up in the details (i.e. the last two blocks aren't as large as the rest; the resampled data set is larger or smaller than my original data set).

(d) Write code that creates a figure that clearly illustrates the data, the fitted line, the projection, and the uncertainty of the projection.