SS2864B, 2020

Assignment #5 due to April 3, 2020

Instructions Submit an electronic version (pdf, words) of your solutions (appropriately annotated with comments, plots, and explanations; notice that neatness counts) to owl. Save all your R codes in one script (or markdown) file with proper comments and submit it as well to owl.

1. Let $f(x) = (\cos x)^2$ for $0 < x < 2\pi$.

   (a) Graph the function $f(x)$.

   (b) Use Monte Carlo integration with uniform$[0, 2\pi]$ sample size 1,000,000 to find the area under $f(x)$ on the range $0 < x < 2\pi$, and to find a 95% confidence interval for the area.

   (c) Use trigonometry or calculus to find the same area exactly. Did the confidence interval cover the true value?

2. In this question, you will do some resampling and show results in graphics. This is related to bootstrap technique. The population distribution is a normal with $\mu = 10$ and $\sigma^2 = 4$. The statistic is the sample mean. Hence in theory we know exactly what the density function of the sample mean is.

   (a) Simulate a sample, say x, with sample size n=100. Report its mean, sd, min, and max.

   (b) Use R functions **sample** and **replicate** to resample x 50000 times with replacement. The statistic is the sample mean and the output is booted.data. Find the mean, sd, min, and max of booted.data.

   (c) Plot the histogram of booted.data. Please double the cells of histogram since the default one is too small. Please plot as a density plot since the theoretical density will be added in the next step. Comment the shape and center of this distribution.

   (d) Plot the histogram of booted.data-mean(x) with twice number of default cells. Please plot as a density plot. Add the theoretical density function of the $\bar{X} - \mu$ to the histogram with different line type and color. Comment out your findings.

   (e) Repeat the procedures from (a) to (d) two additional times to check consistency.

3. For an odd number of data $x = (x_1, \ldots, x_n)$, the minimizer of the object function

$$f(\theta|x) = \sum_{i=1}^{n} |x_i - \theta|$$

   is the sample median of x.

   (a) Build an object function **my.obj** with arguments **theta** and **x**. The return value is $\sum_{i=1}^{n} |x_i - theta|$. Try to vectorize your codes without any looping.

   (b) Use the R function **optimize** to find the min value of the object function $f(\theta|x)$ for a given data x. Please implement it as an R function with arguments **x** and **interval**=(min(x), max(x)). The return value is the theta that minimizes the f. Test your function with the dataset 3, 7, 9, 12, 15, 18, 21.

(c) Use the R function **nlminb** to find the min value of the object function $f(\theta|x)$ for a given data x. Please implement it as an R function with arguments **x** and **start**=mean(x). The return value is the theta that minimizes the f. Test your function with the dataset 3, 7, 9, 12, 15, 18, 21.

(d) Test your functions from (b) and (c) with the dataset 1, 3, 7, 9, 12, 15, 18, 21. With the function from (b), use three different **wider** intervals. With the function from (c), use three different **start** values. What are your findings? For such a dataset, plot f with theta from min(x) and max(x) and conclude your findings.

4. In this question you will use the R function **nlminb** to fit the Huron water levels with AR(1) time series model. The time series **huron** records the mean water levels from 1860 to 1986. Use the following steps to carry out your study.

   (a) Open the dataset **huron.R** in owl and copy/paste into R to create an R object **huron**. Compute its mean, sd, min, max, and median. Do boxplot and time series plot and comment your findings.

   (b) Let huron2=huron-mean(huron). Use **nlminb** to find the MLE of par[1] with the following negative log likelihood function

   ```
   log.likelihood=function(par, x){
       n=length(x)
       v=x[1]^2
       for (i in 2:n)
           v=v+(x[i]-par[1]*x[i-1])^2
       return(v/par[2]+n*log(par[2]))
   }
   ```

   Make sure to choose a proper start value and the ranges that par[1] and par[2] within.

   (c) Based on the estimator par[1] obtained in (b), compute the predicted values of **huron2** as

   ```
   pred.huron2=huron2
   for (i in 2:length(huron2))
       pred.huron2[i] = par[1]* huron2[i-1]
   ```

   Then do a time series plot on **huron2** and add **pred.huron2** into it with different color. Comment your findings.