Homework 3. Stat 202a. Due Mon, Nov 11, 11:59pm.

You must work on the homework INDEPENDENTLY! Collaborating on this homework will be considered cheating. **Late homeworks will not be accepted!** Your homework solution should be a single PDF document. The first pages should be your *output* from the problems. After that, on subsequent pages, include all your *code* for these problems. Email your homework to statgrader@stat.ucla.edu .

1. Approximation of an infinite series in C.

It is well known that 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 +/- ... = ln(2).
Write a C function called *alt2(n)* that computes the first *n* terms in this series, as a function of *n*. Call your C function from *R* to evaluate *alt2(n)* for various *n*. Using *R*, plot *alt2(n)* vs. *n*, for *n* ranging from some small number up to 1 million. You may set up your range of the y-axis in a way that you feel is appropriate. You do not need to show alt2(n) for all values of n and should not plot alt2 for very small values of n if they are off the plot.

2. Kernel density estimation in C and plotted in R.

Write a C function to compute a Gaussian kernel density estimate for univariate data. The inputs to the function should be two integers, m and n, a vector g of m gridpoints at which to calculate the estimates, a vector x consisting of the n observed data points, and a vector y of length m which will contain the resulting density estimates.

Gather data on all earthquakes of magnitude at least 3.0 in the longitude range -122.0 to -118.0 and latitude range 34.0 to 38.0, from Jan 1, 1960 to Oct 1 2024, from http://service.scedc.caltech.edu/eq-catalogs/date_mag_loc.php . Input the data into R. Use minimum magnitude = 3.0, maximum magnitude = 9.0, min depth = -5km, max depth = 100km, event type = earthquake, geographic type = local. Take this vector of earthquake magnitudes, and use your C function to make a kernel density estimate of the earthquake magnitudes, using a Gaussian kernel with bandwidth selected using the rule of thumb suggested by Scott (1992). You may calculate this bandwidth in *R*. Let $\{m_1, m_2, ..., m_{100}\}$ = a vector of 100 equally spaced magnitudes spanning the observed range of magnitudes in your dataset, compute your kernel estimates on this grid using the C function, and plot your kernel density estimates $\hat{f}(m_1)$, $\hat{f}(m_2)$, ..., $\hat{f}(m_{100})$.

3. Approximation of an integral in C.

Consider the integral from 0 to xmax of the shifted Pareto density,
$f(x) = (p-1)\, c^{p-1}\, (x+c)^{-p}$, for $x \ge 0$, and f(x) = 0 otherwise,
where c>0 and p>1 are parameters.

Let *c = 4* and *p = 3*. Write a C function called *paretoint(xmax,c,p)* that approximates this integral over a grid of 1 million values ranging from *x = 0* to *xmax*. Note that technically *paretoint()* is not only going to be a function of *xmax, c*, and *p*, but will also have another input variable which will store the result. Call your C function from *R* to evaluate *paretoint(xmax,c,p)* for various choices of *xmax* between 10 and 1000 (you do not need to calculate paretoint for every integer between 10 and 1000, but choose around 10-15 numbers between 10 and 1000), and for *c =4* and *p = 3* each time. Using *R*, plot *paretoint(xmax,4,3)* vs. *xmax*, for *xmax* ranging from 10 up to 1000. You may set up your range of the y-axis in a way that you feel is appropriate.

Output: Your output for this assignment should be a pdf document containing the following, in this order.

Figure 1. A plot of alt2(n) versus n, for several values of n ranging up to 1 million.

Figure 2. A plot of your kernel density estimates $\hat{f}(m_1)$, $\hat{f}(m_2)$, ..., $\hat{f}(m_{100})$ versus m.

Figure 3. A plot of *paretoint(xmax,4,3)* vs. *xmax*, for *xmax* ranging from 10 to 1000.

All of your code, at the end.