ECS 132 Homework 3

Problem A:

Download the DNC e-mail co-recipient dataset from canvas. Here we will explore the degree distribution of the resulting graph.

- The format seems to be

recip1ID recip2ID nummsgs

- One might treat this as a weighted graph, but we will not do so. Ignore the third column.
- The data seems to be duplicated, with (a,b) and (b,a) both appearing in the data. This is odd, as the third column differs, lets take only the rows in which recip1ID < recip2ID. This indeed cuts the data in half.
- Explore how well a power law fits the data, as follows. Let m_i denote the count of recipients having degree i in the data, i = 1,2,3,... The form of the pmf (check this!) implies that a plot of $\log(m_i)$ against log(i) should look like a straight line. There will be points above and below the line, due to sampling variation, but the trend should look linear. Make this plot, and comment.
- **NOTE:** In this and all future work for the course, you must use R to generate your graphs. This can be either base R, **ggplot2** (appendix in our book) or **lattice**.
- Assuming a power law, estimate γ. (We use the term *estimate* here because the data is only a sample from a population (real or conceptual). Use R's **lm()** function for this. We will study this function in detail later, but for instance the following would find the intercept and slope of a line fit through the points (1,1), (2,2), (3,4):

```
> x <- 1:3
> y <- c(1,2,4)
> lm(y ~ x)
```

- Re-do your earlier plot with the fitted line superimposed. Do not hard-code the intercept and slope from above; instead, call **coef()** on the object returned by **lm()**.

Problem B:

Here you will develop "d,p,q,r" functions for a certain distribution family, in the sense of e.g. Sec. 4.4.1.

- We'll call the family "accum" for "accumulate." The setting is that of repeatedly rolling a pair of dice. The random variable X is the number of rolls needed to achieve an accumulated total of at least k dots. So for instance the support of X ranges from ceiling(k/12) to ceiling(k/2). This is a one-parameter family.
- The call forms will be

```
daccum(i,k)
paccum(i,k)
qaccum(m,k)
raccum(nreps,k)
```

-
- The 'd', 'p' and 'q' functions must be exact, i.e. not computed via simulation, but a recursive call is fine. Finding 'p' and 'q' from 'd' is fine.
- For 'q', note the comment following (4.31).

Problem C: Written HW no simulation.

Part 1

Suppose the number of bugs per 1,000 lines of code has a Poisson distribution with mean 4.5. Let's find the probability of having more than 90 bugs in 20 sections of code, each 1,000 lines long. We'll assume the different sections act independently in terms of bugs.

Please show the z-table lookup required to solve this problem if you didn't have R, now show the R function call.

Part 2:

Part A.
Generate 8 random numbers from a geometric distribution with p=.65.

Part B.
Find the 94% confidence interval for E[x]

Part 3.

Part A.
Generate 8 random numbers from an exponential distribution with mean =32.

Part B.
You are looking at the population of salmon in the Sacramento River and are interested in the weight of the salmon.

You catch and release 8 salmon and weigh them. Pretend that the weight data you collect for the 8 samples is the data from Part A where each number is the weight in lb.

Find the 94% confidence interval for the average weight salmon population.

Part 4.

Given two populations of salmon and tuna you have been told the length of the two fish *has no difference* however you sample the Sacramento river and see that the average length of tuna is 15in and salmon is 14in with st.dev of 1 and 2 respectively (samples size of 15 and 20 respectively). Test the alternative hypothesis with $\alpha = .04$.