HW week 7

W203 - Section 4

Brad Andersen

The Meat

Suppose that Americans consume an average of 2 pounds of ground beef per month.

(a) Do you expect the distribution of this measure (ground beef consumption per capita per month) to be approximately normal? Why or why not?

Assuming that the "Americans" referenced in the question represents a population and not a sample, *no*, I do not expect the distribution of this measure to be normal. Because of the Central Limit Theorem, we can expect *sample* distributions to be normal, provided the sample size is sufficiently large (i.e. $n > 30$). However, the Central Limit Theorem does not hold for *population* distributions.

(b) Suppose you want to take a sample of 100 people. Do you expect the distribution of the sample mean to be approximately normal? Why or why not?

Yes, I expect the distribution of the sample mean to be approximately normal because of the Central Limit Theorem. Two criteria associated with anticipating a normal distribution are satisfied: the 100 people are a selected *sample* and are not the population, and the sample size is greater than 30.

(c) You take a random sample of 100 Berkeley students to find out if their monthly ground beef consumption is any different than the nation at large. The mean among your sample is 2.45 pounds and the sample standard deviation is 2 pounds. What is the 95% confidence interval for Berkeley students?

The 95% confidence interval for the sample can be calculated using the following equation:

$$\big(\bar{X} - 1.96 \cdot \frac{s}{\sqrt{n}}, \bar{X} + 1.96 \cdot \frac{s}{\sqrt{n}}\big)$$

Substituting values for the sample mean, standard deviation and size:

$$\big(2.45 - 1.96 \cdot \frac{2}{\sqrt{100}}, 2.45 + 1.96 \cdot \frac{2}{\sqrt{100}}\big)$$

The confidence interval is as follows:

$$\big(2.058, 2.842\big)$$

GRE Scores

Assume we are analyzing MIDS students' GRE quantitative scores. We want to construct a 95% confidence interval, but we *naively* uses the famous 1.96 threshold as follows:

$$\big(\bar{X} - 1.96 \cdot \frac{s}{\sqrt{n}}, \big(\bar{X} + 1.96 \cdot \frac{s}{\sqrt{n}}\big)$$

What is the real confidence level for the interval we have made, if the sample size is 10? What if the sample size is 200?

The code snippet in this document's appendix will calculate the confidence level for samples of various sizes assuming a z-critical value of 1.96.

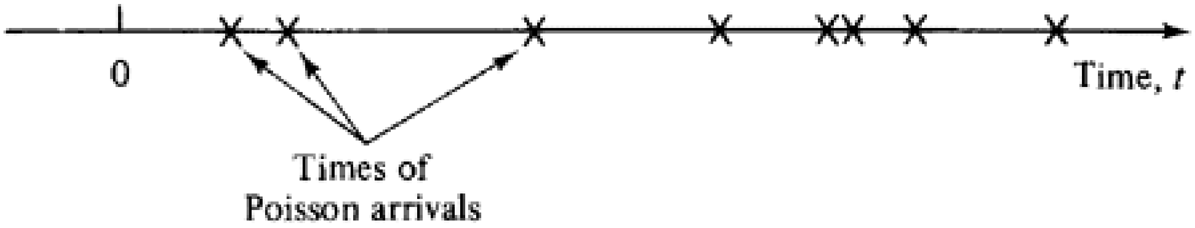

Figure 1: Events over time

```
> derive_ppoint(z_critical = 1.96, n = 10)
[1] 0.959
> derive_ppoint(z_critical = 1.96, n = 200)
[1] 0.974
```

Percentile points for the interval detailed above using sample sizes of 10 and 200 are 95.9 and 97.4, respectively. Because the sample size of the first calculation is less than 30, we can anticipate that the sample does not follow a normal curve, and its t distribution will be more spread out on the x axis. Because the sample size of the second calculation is greater than 30, the Central Limit Theorem states that the distribution should be close to approaching a normal distribution. The can be seen with the percentile point of 97.4, which is very close to a percentile point of 97.5 associated with 95% of the area under a normal distribution lying within 1.96 standard deviations of the mean.

Maximum Likelihood Estimation for an Exponential Distribution

A Poisson process is a simple model that statisticians use to describe how events occur over time. Imagine that time stretches out on the x-axis, and each event is a single point on this axis(see Figure 1).

The key feature of a Poisson process is that it is *memoryless*. Loosely speaking, the probability that an event occurs in any (differentially small) instant of time is a constant. It doesn't depend on how long ago the previous event was, nor does it depend on when future events occur. Statisticians might use a Poisson process (or more complex variations) to represent:

- The scoring of goals in a world cup match
- The arrival of packets to an internet router
- The arrival of customers to a website
- The failure of servers in a cluster
- The time between large meteors hitting the Earth

In live session, we described a Poisson random variable, a discrete random variable that represents the number of events of a Poisson process that occur in a fixed length of time. However, a Poisson process can be used to generate other random variables.

Another famous random variable is the exponential random variable, which represents the time between events in a Poisson process. For example, if we set up a camera at a particular intersection and record the times between car arrivals, we might model our data using an exponential random variable.

The exponential random variable has a well-known probability density function,

$$f(x|\lambda) = \lambda e^{-\lambda x}$$

Here, λ is a parameter that represents the rate of events.

Suppose we record a set of times between arrivals at our intersection, $x_1, x_2, ...x_n$. We assume that these are independent draws from an exponential distribution and we wish to estimate the rate parameter λ using maximum likelihood.

Do this using the following steps:

a. Write down the likelihood function, $L(\lambda)$. Hint: We want the probility (density) that the data is exactly $x_1, x_2, ..., x_n$. Since the times are independent, this is the probability (density) that $X_1 = x_1$, times the probability (density) that $X_2 = x_2$, and so on.

$$\begin{aligned} L(\lambda) &= f(x_1, x_2, ..., x_n|\lambda) \\ &= (\lambda e^{-\lambda x_1}) \cdot (\lambda e^{-\lambda x_2}) \cdot, ..., (\lambda e^{-\lambda x_n}) \\ &= \lambda^n e^{-\lambda \sum_{i=1}^{n} x_i} \end{aligned}$$

b. To make your calculations easier, write down the log of the likelihood, and simplify it.

$$\begin{aligned} \log_e(L(\lambda)) &= \log_e(\lambda^n e^{-\lambda \sum_{i=1}^{n} x_i}) \\ &= \log_e(\lambda^n) + \log_e(e^{-\lambda \sum_{i=1}^{n} x_i}) \\ &= n \cdot \log_e(\lambda) - \lambda \sum_{i=1}^{n} x_i \end{aligned}$$

c. Take the derivative of the log of likelihood, set it equal to zero, and solve for λ. How is it related to the mean time between arrivals?

Taking the partial derivative of the log of likelihood function:

$$\frac{\partial \log_e(L(\lambda))}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i$$

Solving for λ:

$$\begin{aligned} \frac{n}{\lambda} - \sum_{i=1}^{n} x_i &= 0 \\ \frac{n}{\lambda} &= \sum_{i=1}^{n} x_i \\ \frac{1}{\lambda} &= \frac{\sum_{i=1}^{n} x_i}{n} \\ \hat{\lambda}_{MLE} &= \frac{n}{\sum_{i=1}^{n} x_i} \end{aligned}$$

This is the reciprocal of the mean time between arrivals.

d. Suppose you get the following vector of times between cars:

```r
times = c(2.65871285, 8.34273228, 5.09845548, 7.15064545,
          0.39974647, 0.77206050, 5.43415199, 0.36422211,
          3.30789126, 0.07621921, 2.13375997, 0.06577856,
          1.73557740, 0.16524304, 0.27652044)
```

Use R to plot the likelihood function. Then use optimize to approximate the maximum likelihood estimate for λ. How does your answer compare to your solution from part c?

I've attempted this quite a number of times, and am yet to arrive at a solution that I feel is correct. When using the maximum likelihood estimation equation, above, and calculating the mean of the *times* vector, I arrive at nothing similar to what I have calculated on various attempts from part c.

My plots tend to resemble cumulative distribution functions, where the function begins very close to 0 on the y axis, then travels upwards before peaking and plateauing at $y = 1$.

Appendix

```
# Default percentile point associated with 95% confidence assuming a normal
# distribtion
DEFAULT_PPOINT <- 0.975

# Calculates the percentile point value associated with a t distribution
# given the desired z critical value and the sample size
derive_ppoint <- function(z_critical, n) {

  degrees_freedom <- n - 1

  # Iterate through percentile point values, beginning with what we know
  # is the value for 95% confidence and decrementing by thousandths
  for (p in seq(DEFAULT_PPOINT, 0, by=-0.001)) {

    current_z_critical <- qt(p, degrees_freedom)

    # If the calculated z critcal value is less than that desired, return
    # the associated percentile point
    if (!is.finite(current_z_critical)) {
      return(NaN)
    }
    if (current_z_critical <= z_critical) {
      return(p)
    }
  }
}
```