Introduction to Maximum Likelihood Estimation in R

Neba Nfonsang
University of Denver

In [1]:

```
require(tidyverse)
```

```
Loading required package: tidyverse
Loading tidyverse: ggplot2
Loading tidyverse: tibble
Loading tidyverse: tidyr
Loading tidyverse: readr
Loading tidyverse: purrr
Loading tidyverse: dplyr
Conflicts with tidy packages ---------------------------------
------------------
filter(): dplyr, stats
lag():    dplyr, stats
```

Maximum Likelihood

- Maximum likelihood estimation tries to estimate the paramter of the distribution that generated the sample data by finding which parameter value is most likely to generate the data among possible parameter values.
- You need to have some sample data that will be used to estimate the population parameter.
- You need to know the distribution that generated the sample data. That is, we need to assume that the data was generated by a specific distribution with certain parameters. The parameters of the distribution should be the parameters of interest that we are trying to estimate.
- The type of random variable associate with the data can give us a hint about what distribution generated the data.
- When the distribution is known or assumed, we can then specify the density function of the distribution.
- The density function is then used to compute the joined probability for the data for each parameter value. The joint probability density of the data for each parameter value is called the likelihood of the parameter.
- The joint density of the data for each parameter value is the product of the densities of the data points for each parameter value.
- Basically, we are using the likelihood function to compute how likely each parameter value generated the data.

- We then maximize the likelihood function by finding the parameter value with the highest likelihood. This parameter value is the maximum likelihood estimate for the parameter we are seeking to estimate.
- The assumption of the maximum likelihood estimation is that, the observations in the sample are independent and are identically distributed. That is, the sample is assumed to be an iid sample.

Normally Distributed Data

- Suppose the sample, x = c(5, 6, 8, 10) follows a normal distribution.
- We can find the mean estimate of the parameter that generated the sample.
- Since the normal distribution has more than one parameter, each time we are estimating one of the parameters, we can treat the other parameter as a constant.

In [2]:

```r
x = c(5, 6, 8, 10)

parameters = seq(from = 5, to=10, 0.1)
parameters
```

```
5   5.1   5.2   5.3   5.4   5.5   5.6   5.7   5.8   5.9   6   6.1   6.2   6.3   6.4   6.5   6.6
6.7   6.8   6.9   7   7.1   7.2   7.3   7.4   7.5   7.6   7.7   7.8   7.9   8   8.1   8.2   8.3
8.4   8.5   8.6   8.7   8.8   8.9   9   9.1   9.2   9.3   9.4   9.5   9.6   9.7   9.8   9.9   10
```

In [3]:

```r
num.param = length(parameters)
likelihoods = rep(NA, num.param)
for (i in 1:num.param){
    likelihoods[i] = prod(dnorm(x = x, mean=parameters[i], sd=1))
    # we can use any sd value since we are keep sd constant,
    # we should still get the same mean estimate for any sd value used
}


ind=which.max(likelihoods)
parameters[ind]
```

```
7.2
```

In [4]:

```
# estimate the population mean directly
mean(x)
```

7.25

Binomial Data

- Imagine the sample, x = c(0, 0, 1, 0, 1) contains data about individuals and this data represents whether an individual is female (x=1) or not female (x=0).
- We can find the proportion of females in the population from which this sample was drawn using maximum likelihood.
- The question we are trying to answer is, among some reasonable parameter values (proportion), which of the parameter value is most likely to have generated the data?
- Since the Binomial random variable is number of sucess, there is only one value of x so we will not be finding joint probability, we will just use the density function directly.

In [5]:

```
x = c(0, 0, 1, 0, 1)

parameters = seq(from = 0, to=1, 0.1)


num.param = length(parameters)
likelihoods = rep(NA, num.param)
for (i in 1:num.param){
    likelihoods[i] = dbinom(x = sum(x), size = length(x), prob = parameter
}

ind=which.max(likelihoods)
parameters[ind]
```

0.4

In [6]:

```
# estimate the propability of a single trial directly
sum(x)/length(x)
```

```
0.4
```

Conclusion

- This lesson was just to give us an intuition of how maximum likelihood works. In these examples examined, we could estimate the parameter values directly without using maximum likelihood estimation. However, there are certain models that are more complicated, where the parameter values cannot be estimated directly, so maximum likelihood estimation has an advantage in such situations.
- Note that there are some likelihood functions that do not converge or do not have a closed form. That means, the function cannot be maximized analytically or using calculus.
- Also, the likelihood function is more computationally difficult to maximimize especially when we are using calculus, so most of the times, the log likelihood function is used instead, for easy computations.
- Maximizing the likelihood function gives the same results as maximizing the log of the likelihood function.
- If we are using some optimization algorithm, specifically a minimization algorithm to maximize the likelihood function, we should rather minimize the negative of the likelihood function. Minimizing the negative of a function is the same as maximizing the function.
- In R, we can use the optimize() or nlm() functions to maximize the likelihood function or log likelihood function.