Numerical optimization using R

- To find the minimum of a real-valued function numerically, the R commands are `optimize` (for a univariate function) and `optim` (for a multivariate function).
  - The usage of `optimize`/`optim` can be found by running `help(optimize)`/`help(optim)` in R.

- Example 1. Find the maximum of $f(x) = -(x-1)^2$ on $[-1, 2]$.

  Sol. Running the R commands

```
mf <- function(x){ (x-1)^2 }
optimize(mf, lower = -1, upper =2)
```

  gives

```
$minimum
[1] 1

$objective
[1] 0
```

  The above result means that $-f(x)$ is minimized at $x = 1$ on $[-1, 2]$ and the minimum $-f(1)$ is 0. Thus the maximum of f is 0 on $[-1, 2]$.

- Example 2. Find the maximum of $f(x, y) = 1 - (x-2)^2 - (y-3)^2$.

  Sol. Running the R commands

```
mf <- function(z){
  x <- z[1]
  y <- z[2]
  ans <- -1+(x-1)^2+(y-3)^2
  return(ans)
}
optim(c(0,0), mf)
```

  gives

```
$par
[1] 0.999943 3.000062

$value
[1] -1

$counts
function gradient
```

```
       69          NA

$convergence
[1] 0

$message
NULL
```

The above result means that $-f(x,y)$ is minimized at $(x,y) = (0.999943, 3.000062)$ (near $(0,0)$) and the minimum is -1. Thus the maximum of f is 1 (near $(0,0)$).

- Example 3. Suppose that $(X_1,\ldots,X_n)$ is a random sample from the gamma distribution with PDF $f_{\alpha,\beta}$, where $\alpha > 0$, $\beta > 0$

$$f_{\alpha,\beta}(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta} I_{(0,\infty)}(x)$$

for $x \in (-\infty,\infty)$, where $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}dx$ for $\alpha > 0$. Let $(\hat{\alpha},\hat{\beta})$ denote the MLE of (α,β).

(a) Write an R function with input vector `x` and output vector `y`, where `x` is the data vector $(X_1,\ldots,X_n)$ and `y` is the the MLE $(\hat{\alpha},\hat{\beta})$.

(b) Find the approximate MSE of the MLE of α when $n = 1000$, $\alpha = 2$ and $\beta = 1$ based on 10000 errors of $\hat{\alpha}$.

Sol. Let L be the likelihood function, then

$$L(a,b) = \frac{1}{(\Gamma(a)b^a)^n}\left(\prod_{i=1}^n X_i\right)^{a-1} e^{-\sum_{i=1}^n X_i/b}$$

for $a > 0$, $b > 0$. It can be shown that $L(a,b) \le L(a,\bar{X}/a)$ for $b > 0$, where $\bar{X} = \sum_{i=1}^n X_i/n$. Thus the MLE $(\hat{\alpha},\hat{\beta})$ can be found by

$$\hat{\alpha} = \arg\max_a \log(L(a,\bar{X}/a))$$

and

$$\hat{\beta} = \bar{X}/\hat{a}.$$

- For Part (a), we will define a function $g(a) = \log(L(a,\bar{X}/a))/n$ and find the minimizer of -$g(a)$ with $a = \lambda/(1-\lambda)$ for $\lambda \in (0,1)$ using `optimize`. The R function is given below.

```
get_mle <- function(x){
 bar.x <- mean(x)
 bar.lnx <- mean(log(x))
 g <- function(a){ -log(gamma(a)) - a*log(bar.x/a) + (a-1)*bar.lnx - a }
 g1 <- function(lambda){ -g(lambda/(1-lambda)) }
 lambda.hat <- optimize(g1, lower=0, upper=1)$minimum
 alpha.hat <- lambda.hat/(1-lambda.hat)
 beta.hat <- bar.x/alpha.hat
 return(c(alpha.hat, beta.hat))
}
```

The R function `gamma` is the Γ function.

- For Part (b), running the R commands

```
res <- matrix(0, 10000, 2)
for (i in 1:10000){
 set.seed(i)
 x <- rgamma(1000, shape=2, scale=1)
 res[i,] <- get_mle(x)              #res[i,]: the i-th row of res
}
mean((res[,1] - 2)^2)               #approximate MSE for MLE of alpha
                                    #res[,1]: first column of res
```

gives 0.007136904, which is an approximate MSE for $\hat{\alpha}$ when $n = 1000$, $\alpha = 2$, $\beta = 1$.