

# Task3

October 31, 2020

## 1 Task 3: Maximum Likelihood Estimator

### 1.0.1 Daniel Alonso & Ander Iturburu

#### 1.1 Computation of MLE using user-written function and *optimize()*

First of all, we are going to calculate the MLE of gamma distribution by an user-written function and then, finding the maximum of the vector with *optimize()*

```
[ ]: s = 3
r = 1
x = rgamma(5000,shape=s,scale=r)
L = function(v) {
  shape <- v[1]
  scale <- v[2]
  -sum(log(dgamma(x, shape, scale)))
}
optim(c(.5,.5),L)
```

\$par 1. 3.01992716429069 2. 1.01686193133783

\$value 9176.21258836363

\$counts function 67 gradient <NA>

\$convergence 0

\$message NULL

So, the estimators predicted by Maximum Likelihood are  $k = 3.01992716429069$  and  $\theta = 1.01686193133783$ , which are near to our actual values.

#### 1.2 Computation of MLE by *maxLik()*

The main advantage of using *maxLik()* function over the common used *mle()* is that we can implement the **Newton-Raphson** algorithm in order to find our estimator. This means that we can also implement the **Berndt-Hall-Hausman** algorithm, a widely used algorithm for Maximum Likelihood problems.

The *maxLik()* function has the following structure:

```
[ ]: maxLik ( logLik , grad = NULL , hess = NULL , start , method ,
3 constraints =NULL , ...)
```

These arguments are the following:

- **logLik**: The log-likelihood function to optimize.
- **grad**: The gradient of log-likelihood. If this element is *NULL*, we will use numeric gradient.
- **hess**: The hessian of log-likelihood. If this element is *NULL*, we will use numeric hessian.
- **start**: The start point in order to find our estimators. It is a numeric vector.
- **method**: The maximisation method. If it is missing, automatically the function will select a method that fits.
- **constraints**: The default argument is *NULL* for unconstrained maximization.
- ...: further arguments, such as *control* are passed to the selected maximisation routine.

Then, we are going to apply MLE using *maxLik()* function.

```
[ ]: install.packages("maxLik", repos="http://R-Forge.R-project.org")
```

Installing package into ‘/usr/local/lib/R/site-library’  
(as ‘lib’ is unspecified)

also installing the dependencies ‘zoo’, ‘miscTools’, ‘sandwich’

```
[ ]: library("maxLik")
```

Loading required package: miscTools

Please cite the 'maxLik' package as:

Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood estimation in R. Computational Statistics 26(3), 443-458. DOI 10.1007/s00180-010-0217-1.

If you have questions, suggestions, or comments regarding the 'maxLik' package, please use a forum or 'tracker' at maxLik's R-Forge site:  
<https://r-forge.r-project.org/projects/maxlik/>

```
[ ]: s = 3
r = 1
x = rgamma(5000, shape=s, scale=r)
```

```
[ ]: LL = function(v) {
  shape <- v[1]
  scale <- v[2]
  sum(log(dgamma(x, shape, scale)))
}
```

```
[ ]: r=maxLik(LL, start=c(shape=0.5,scale=0.5),method="NR")
```

```
Warning message in dgamma(x, shape, scale):  
"NaNs produced"
```

```
[ ]: summary(r)
```

```
-----  
Maximum Likelihood estimation  
Newton-Raphson maximisation, 10 iterations  
Return code 2: successive function values within tolerance limit  
Log-Likelihood: -9176.212  
2 free parameters  
Estimates:  
      Estimate Std. error t value Pr(> t)  
shape  3.01922    0.05743   52.57 <2e-16 ***  
scale  1.01649    0.02103   48.34 <2e-16 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
-----
```

```
[ ]: r=maxLik(LL, start=c(shape=0.5,scale=0.5))
```

```
Warning message in dgamma(x, shape, scale):  
"NaNs produced"
```