Exponential model

We consider the following univariate Exponential model, parametrized by λ, a positive real number

$$\left\{\mathcal{E}(\lambda); \lambda > 0\right\}$$

The PDF of the Exponential distribution is given by:

$$f_\lambda(x) = \lambda e^{-\lambda x} \; \mathbb{1}_{\mathbb{R}_+^*(x)}$$

To obtain a **Maximum Likelihood Estimator** (MLE), denoted $\hat{\lambda}$ for λ from a sample of n $i.i.d.$ realizations of an Exponential r.v., we first write the likelihood function (joint PDF of the sample), take the logarithm of this function, differenciate it w.r.t. the parameter and then solve for the parameter.

Maximum likelihood estimation

Likelihood and log-likelihood functions for an Exponential model

$$\mathcal{L}(\lambda \mid \mathbf{x}) = \prod_{i=1}^{n} f_\lambda(x_i) = \prod_{i=1}^{n} \lambda e^{-\lambda x_i} = \lambda^n e^{-\left(\lambda \sum_{i=1}^{n} x_i\right)}$$

$$l(\lambda \mid \mathbf{x}) = ln\big(\mathcal{L}(\lambda \mid \mathbf{x})\big) = n\ ln(\lambda) - \lambda \sum_{i=1}^{n} x_i$$

$$\frac{\partial ln \mathcal{L}(\lambda \mid \mathbf{x})}{\partial \lambda} = n \frac{1}{\lambda} - \sum_{i=1}^{n} x_i$$

Setting the derivative equal to 0 and solving for λ then yields

$$\frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0 \quad \Leftrightarrow \quad \frac{n}{\lambda} = \sum_{i=1}^{n} x_i \quad \Leftrightarrow \hat{\lambda}_{MLE} = \frac{n}{\sum_{i=1}^{n} x_i} = \frac{1}{\overline{x}}$$

Exponential MLE in R: closed-form formula

```r
# generate a random sample of n = 5000 from an
# Exponential distribution
# with parameter lambda = 1.75 (argument name = 'rate')

set.seed(1986)
n = 5000
xi <- rexp(n = n, rate = 1.75) # rate assumed unknown
head(xi)
# [1] 0.4817649 0.5847904 0.1148642 0.1820848 0.9921011 0.2747533

# Closed-form MLE
lambda_hat_formula = n / sum(xi)   # or 1 / mean(xi)
lambda_hat_formula
# [1] 1.744645

# the approximation is good with such a large sample.
```

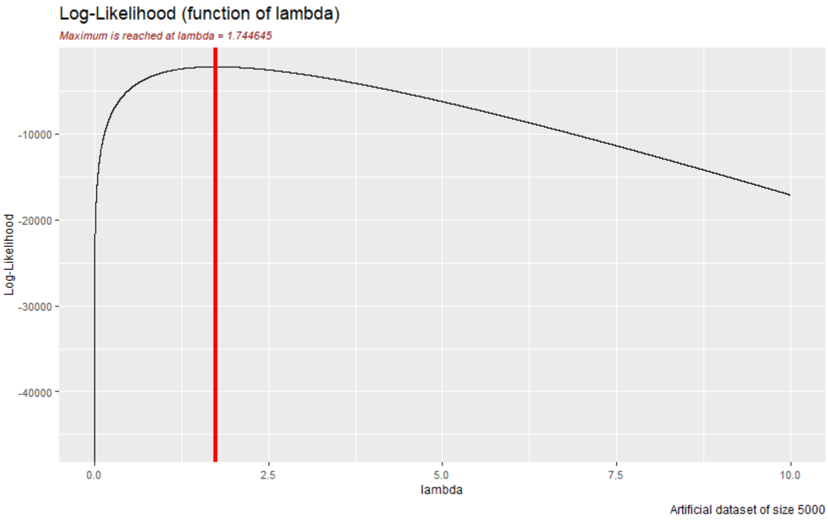
Exponential MLE in R: numerical approximation

```
# Numerical approximation of the MLE
mle = optimize(function(lambda){
                 sum(dexp(x = xi, rate = lambda, log = TRUE))
                 },
                 interval = c(0, 10),
                 maximum = TRUE,
                 tol = .Machine$double.eps^0.5)

lambda_hat = mle$maximum
lambda_hat
# [1] 1.744645

# the result is exaclty the same as when using
# the closed-form formula
```

Plot of the log-likelihood function



Exponential MLE in Python: closed-form formula

```python
from scipy import stats
import numpy as np

# generate a sample of size 5000
# in numpy, the parametrization is different from R
np.random.seed(1986)
n = 5000
Lambda = 1.75 # true parameter value, that we will estimate
xi = np.random.exponential(scale = 1/Lambda, size = n)
print(xi)
# [0.48077598 0.04599578 0.53583846 ... 0.31828503
    0.05858069 0.0721613 ]

# Closed-form MLE
lambda_hat_formula = n/sum(xi)   # or 1/statistics.mean(xi)
lambda_hat_formula
# [1] 1.71848223876711
```

Exponential MLE in Python: numerical approximation

```python
def llikelihood(Lambda):
    # log-likelihood function
    ll = -np.sum(stats.expon.logpdf(xi, scale = 1/Lambda))
    return ll

from scipy.optimize import minimize

# Numerical approximation of the MLE using minimize()
mle = minimize(llikelihood,
               x0 = 4,
               method = 'BFGS')
print(mle.x)
# [1.71848223]
```

Further reading and code

The R Project for Statistical Computing:
https://www.r-project.org/

Python:
https://www.python.org/

Accessing R and Python code:
https://github.com/JRigh/Simple-maximum-likelihood-estimation/