18.S096 Problem Set 3 Spring 2018
Due Date: 3/9/2018
Where: On Stellar, prior to 11:59pm

Collaboration on homework is encouraged, but you will benefit from independent effort to solve the problems before discussing them with other people. **You must write your solution in your own words. List all your collaborators.**

1. **MLE for Truncated $Poisson(\lambda)$ Distribution**

   Suppose sample observations $X_1, \ldots, X_n$ from a $Poisson(\lambda)$ distribution are truncated so that the observations are $Y_1, \ldots, Y_n$ where

   $$Y_i = \begin{cases} 0, & if \quad X_i = 0 \\ 1, & if \quad X_i > 0 \end{cases}$$

   1(a) Derive the MLE for λ given the truncated sample $Y_1, \ldots, Y_n$.

   1(b) Using the R function $rpois()$ to generate random Poisson variates, conduct a Monte Carlo simulation comparing the sampling distribution of the Truncated MLE to the sampling distribution of the regular MLE.

   - Simulate sampling distributions for two cases of $\lambda : \lambda = 2$, and $\lambda = 1/5$.
   - Consider two cases of the sample size: $n = 50$ and $n = 200$.

   For each case of the simulation (choice of lambda and choice of sample size) compare these distributions by constructing a parallel boxplot and computing sample means/standard deviations of the distributions.

   1(c) For the simulation in part (b), answer the following questions:

   - How does the relative efficiency (variance ratio) of the two estimates depend on λ?
   - How does the absolute efficiency (variances) of the two estimates depend on λ and the sample size?
   - Is there an issue with the truncated MLE ever being infinite. If so, how should this be taken in account with the comparisons?

2. The R function $fitdistr()$ in the R package $MASS$ fits univariate distributions by maximum likelihood. As detailed in $help(fitdistr)$, the syntax is:

```
fitdistr(x, densfun, start, ...)
```

with Arguments

- *x*: A numeric vector of length at least one containing only finite values.
- *densfun*: Either a character string or a function returning a density evaluated at its first argument. Distributions "beta", "cauchy", "chi-squared", "exponential", "f", "gamma", "geometric", "log-normal", "lognormal", "logistic", "negative binomial", "normal", "Poisson", "t" and "weibull" are recognised, case being ignored.
- *start*: A named list giving the parameters to be optimized with initial values. This can be omitted for some of the named distributions and must be for others (see Details).

The output *Value* of the function is an object of class "fitdistr", a list with four components:

- *estimate* : the parameter estimates,
- *sd*: the estimated standard errors,
- *vcov*: the estimated variance-covariance matrix, and
- *loglik*: the log-likelihood.

The following R code simulates a sample from the Gamma distribution and applies *fitdistr*() to estimate the distribution parameters:

```
> #
> library(MASS)
> set.seed(1)
> samplesize=100
> x=rgamma(samplesize,shape=3, rate=1)
> fit_gamma_mle<-fitdistr(x,densfun="gamma")
> print(fit_gamma_mle)

     shape       rate
  3.7673500   1.2919621
 (0.5109475) (0.1874392)

> names(fit_gamma_mle)

[1] "estimate" "sd"       "vcov"     "loglik"   "n"

> fit_gamma_mle$estimate

   shape     rate
3.767350 1.291962

> fit_gamma_mle$sd
```

```
    shape      rate
0.5109475 0.1874392

> fit_gamma_mle$vcov

           shape       rate
shape 0.26106737 0.08952941
rate  0.08952941 0.03513346

> fit_gamma_mle$loglik

[1] -173.1452

> fit_gamma_mle$n

[1] 100
```

2(a) Generate 4 samples from a $Gamma(shape = 3, rate = 2)$ distribution with sample sizes: 100, 400, 800, 1600. Apply $fitdistr()$ to compute the mles and standard errors for each sample

```
> # Generate random samples from a
> #     Gamma(shape=3,rate=2) distribution
> set.seed(1)
> list.samplesize<-c(100,400,800,1600)
> for (samplesize in list.samplesize){
+   assign(paste("sample.gamma",samplesize,sep="."),
+          rgamma(samplesize, shape=3,rate=2))}
> mlefit.sample.gamma.100<-fitdistr(sample.gamma.100,
+                                   densfun="gamma")
> mlefit.sample.gamma.400<-fitdistr(sample.gamma.400,
+                                   densfun="gamma")
> mlefit.sample.gamma.800<-fitdistr(sample.gamma.800,
+                                   densfun="gamma")
> mlefit.sample.gamma.1600<-fitdistr(sample.gamma.1600,
+                                   densfun="gamma")
```

2(b) How does the *sd* of the mle's depend on the sample size? Is this consistent with maximum likelihood theory/asymptotics?

2(c) Define the function $mydgamma()$ to compute the density of a gamma distribution applying its formula:

$$f(x \mid \alpha, \beta) = \frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{\alpha-1}e^{-\beta x},\ x > 0.$$

Apply $fitdistr()$ with this user-defined density to compute the mle's for the four samples. Confirm that you obtain results consistent with part (a).

2(d) Use the R function $microbenchmark()$ in $library(microbenchmark)$ to compare the computation times of the two options

$$densfun = "gamma" \text{ and } densfun = mydgamma$$

Is one option generally faster on average than the other? If so, what would explain the difference?

3. **Laplace Distribution**

Let $X_1, X_2, \ldots X_n$ be a random sample from the $Laplace(\mu, b)$ distribution with density:

$$f(x \mid \mu, b) = \frac{1}{2b} e^{-\dfrac{|x-\mu|}{b}}, \ -\infty < x < \infty.$$

with two parameters:

μ (location)

b (scale)

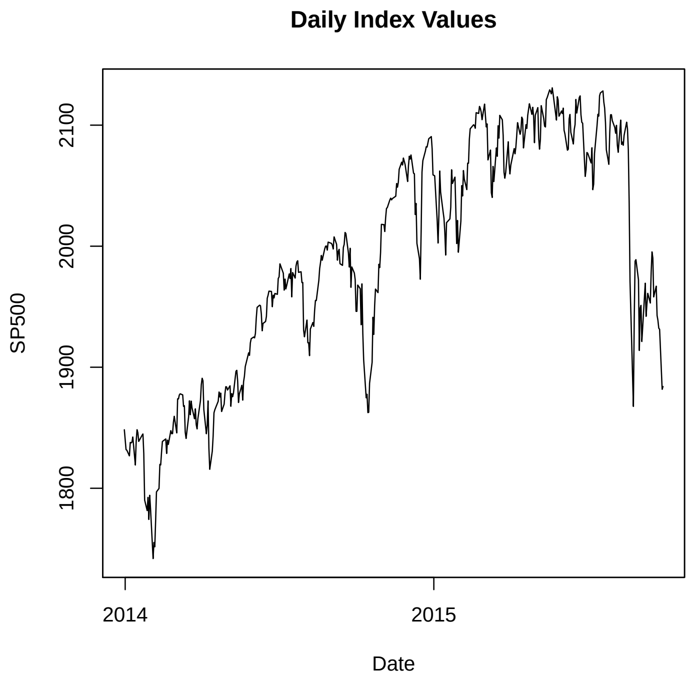
3(a) Derive formulas for the method-of-moments estimates of μ and b.

3(b) Derive formulas for the maximum-likelihood estimates of μ and b.

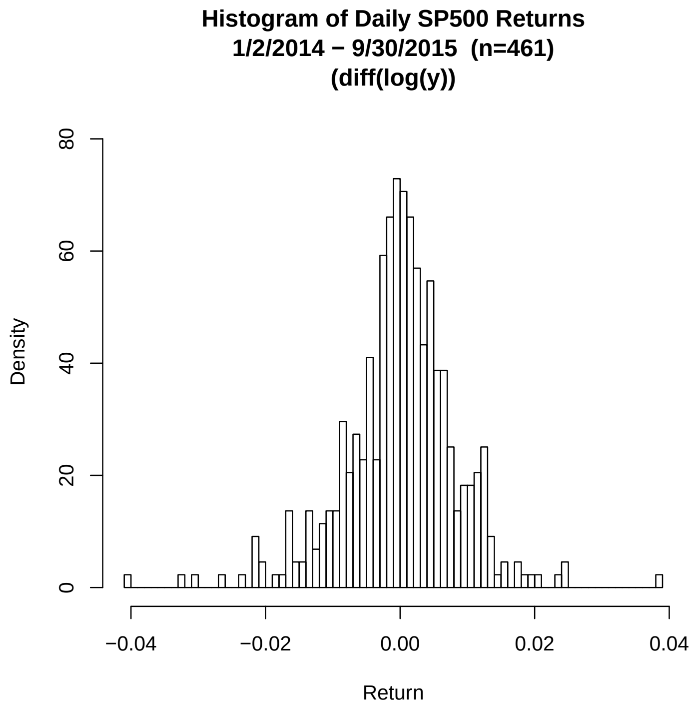
3(c) The file $SP500.csv$ has daily values of the S&P 500 stock index from 1/2/2014 to 9/30/2015.

- In R, install the package "zoo" (for time series) and use $read.zoo()$ to read the data into R.
- Plot the time series
- Compute the logarithmic daily returns and plot the histogram.

```
> # install.packages("zoo")
> library(zoo)
> SP500<-read.zoo(file="SP500.csv")
> par(mfcol=c(1,1))
> plot(SP500, main="Daily Index Values", xlab="Date")
```




```
> y<-diff(log(SP500))
> hist( y, breaks=100,
+       ylab="Density",xlab="Return",
+       freq=FALSE,ylim=c(0,84.),
+       main=paste(c("Histogram of Daily SP500 Returns",
+                    "1/2/2014 - 9/30/2015  (n=461)",
+                    "(diff(log(y))"),collapse="\n"))
>
```



3(d) Fit the parameters of the Laplace distribution by method-of-moments and by maximum likelihood; report the estimates and draw the fitted density from each method on the histogram.

3(e) Define the R function *dlaplace*() to compute the density function for a Laplace distribution

```
> dlaplace<-function(x, location=0,scale=1){
+   dx=(0.5/scale)* exp(-abs(x-location)/scale)
+ }
```

Use the *R* function $fitdistr()$ to fit the Laplace distribution by maximum likelihood. Comment on the consistency of the mle's from $fitdistr()$ and those computed using the exact formulas in (d).

3(f) Compare the mle and method-of-moments estimates (how big are the differences in terms of the sd's of the mle's output by $fitdistr()$).

4. The workings of the R function $fitdistr()$ can be understood by studying the function definition. Print the function definition by typing just the function name

   $>$ fitdistr

   at the R console without any parentheses. To study/edit the function you can use the built-in object/function editor called $fix()$, i.e.,

> fix(myfitdistr)

However, many find $fix()$ cumbersome to use when editing/revising functions. Instead, make a copy of the function $fitdistr()$ by copying the function definition into a separate R script file. In this new file, edit the first line to give the function assignment a new name, i.e.,

```
myfitdistr<-function (x, densfun, start, ...)
{
  myfn <- function(parm, ...) -sum(log(dens(parm, ...)))
  ...
  (rest of function)
}
```

For convenience, such a file is included in the problem set materials.

Edit the function script file to include comments explaining the code at the following if-blocks:

4(a) if-block

if (distname == "poisson")

4(b) if-block

if (distname == "normal")

4(c) if-block

if (distname == "gamma")

5. **ML estimation of Normal Linear Regression model using Newton's Method**

Consider

$$\vec{y} = X\vec{\beta} + \vec{\epsilon}, \quad \vec{\epsilon} \sim N_n(\vec{0}, \sigma^2 \mathrm{I}_m)$$

where X is $n \times p$ design matrix and $\vec{\beta} \in R^p$ is regression parameter; $\sigma^2 > 0$ is error variance.

- Assume σ^2 known.
- Set $\vec{\beta}_0 \in R^p$ arbitrarily.

5(a) Give explicit solution for $\vec{\beta}_1$ using Newton algorithm.

5(b) Verify $\vec{\beta}_1$ equals $\hat{\beta}$, the MLE of β.

5(c) Verify that solution independent of σ^2, so it applies for unknown σ^2.