```
Name: ___________________________________
```

Computing in Statistics, STAT:5400
Midterm 2, Fall 2015

Submit your answers in the ICON drop box as an .Rnw file and .pdf file produced using Sweave, with your name as author. If you can't get your .Rnw file to compile, submit it anyway and include your R output in a separate text file.

In your document, have a named section for each problem, and, where needed, a numbered list of answers to multipart questions. You don't have to type any other text except where needed to answer a question.

1 Simulation studies

Given a random sample of n observations $y_1,\ y_2,\ \ldots, y_n$, the maximum likelihood estimator of the variance in the population from which the sample was drawn is

$$m \ = \ \frac{\sum(y_i - \bar{y})^2}{n}$$

This estimator is known to be biased. The unbiased estimator is

$$s^2 \ = \ \frac{\sum(y_i - \bar{y})^2}{n-1}$$

Carry out a simulation study to compare these two estimators with respect to *mean squared error* for the case when

> the true population distribution is Gamma with shape parameter 8 and rate parameter 2
>
> the sample size is $n = 10$

Set your number of replicate datasets large enough that the standard error of your mean-squared-errors estimates is no larger than 0.04.

```
> set.seed(9)
> n <- 10
> S <- 1000
> mygammas <- matrix( rgamma( n * S, 8,2), nrow=S)
> ssq <- apply(mygammas,1,var)
> mle <- ((n-1)/n) * ssq
> truevar <- 8 / (2^2)
> ssqse <-  (ssq - truevar )^2
> stderrssq <- sqrt( var(ssqse) / S )
> mlese <- (mle - truevar)^2
> stderrmle <- sqrt( var(mlese) / S )
> results <- rbind( c( mean( ssqse ), mean( mlese )),
+             c(stderrssq, stderrmle) )
> rownames(results) <- c( "Mean squared Errors","Standard Errors of MSE")
> colnames(results) <- c( "s^2", "mle" )
> print(results)
```

```
                              s^2        mle
Mean squared Errors    1.15565503 0.98000189
Standard Errors of MSE 0.09549936 0.07047994
```

Standard errors are too big by a factor of about 2.5, so we need to rerun with at least $2.5^2 = 6.25$ times as many replicate datasets.

```
> set.seed(9)
> n <- 10
> S <- 6250
> mygammas <- matrix( rgamma( n * S, 8,2), nrow=S)
> ssq <- apply(mygammas,1,var)
> mle <- ((n-1)/n) * ssq
> truevar <- 8 / (2^2)
> ssqse <-  (ssq - truevar )^2
> stderrssq <- sqrt( var(ssqse) / S )
> mlese <- (mle - truevar)^2
> stderrmle <- sqrt( var(mlese) / S )
> results <- rbind( c( mean( ssqse ), mean( mlese )),
+             c(stderrssq, stderrmle) )
> rownames(results) <- c( "Mean squared Errors","Standard Errors of MSE")
> colnames(results) <- c( "s^2", "mle" )
> print(results)
```

```
                              s^2        mle
Mean squared Errors    1.20024452 1.01097732
Standard Errors of MSE 0.03635198 0.02671013
```

2 Jackknife and Bootstrap

Use your first simulated dataset from the previous problem as your data for this problem. The statistic of interest is the unbiased estimator of variance (s^2 as defined above).

1. Use the jackknife to estimate the bias in s^2 in your gamma dataset of size 10. You may either use the `jackknife` function in the `bootstrap` package or use other code to implement the jackknife. Show your R code and its output.

2. Next, use the nonparametric bootstrap for the same purpose. You may either use the `boot` function in the `boot` package or other code to implement the nonparametric bootstrap. Show your R code and its output.

3. It is often said that the jackknife is an approximation to the bootstrap. Comment briefly on which procedure performs better in this problem.

```
> library(boot)
> library(bootstrap)
> data <- mygammas[1,]
> jackout <-  jackknife(data, var)
> print(jackout)

$jack.se
[1] 1.044081

$jack.bias
[1] 0

$jack.values
 [1] 2.305184 2.536698 2.538855 2.504174 2.166944 2.220637 2.515750 2.097822
 [9] 2.468824 1.337734

$call
jackknife(x = data, theta = var)

> myvar <- function(x,ind) {var(x[ind])}
> bootout <- boot(data, myvar, R=1999)
> print(bootout)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = data, statistic = myvar, R = 1999)


Bootstrap Statistics :
    original      bias    std. error
t1* 2.269262 -0.2415807   0.8491776
```

```
>
```

Since s^2 actually is an unbiased estimator, the jackknife is correct to estimate the bias at 0. In this case, jackknife performs better than bootstrap.

3 Root-finding

Consider the function

$$f(x) \ = \ x^4 - 6x^3 + 5x^2 - x - 1$$

1. Plot this function on the interval (1,7). Show your R code and the plot.
2. Explain briefly why the bisection algorith could be used to find the root of this function on the interval (1,7).

   f is a continuous function. f(1) is negative and f(7) is positive.
3. Find the root of this function on the interval (1,7) in a two-step process:
   (a) First, use the bisection algorithm to find an interval of width not greater than 0.75 that contains the root.
   (b) Then use `uniroot` or any other optimization function of your choice to find the root to within ±0.00001.

```
> f <- function(x)  x^4 - 6 * x^3 + 5 * x^2 -x - 1
> print(f(1))

[1] -2

> print(f(7))

[1] 580

> x <- seq(1,7,by=0.1)
> plot(x, f(x), type="l")
> source("bisection.R")
> bisection.out <- bisection( f, 1,7, 0.75, 15)
> newinterval <- c( bisection.out$a, bisection.out$b)
> print(newinterval)

[1] 4.75 5.50

> print(uniroot(f, newinterval, tol=0.00001))
```

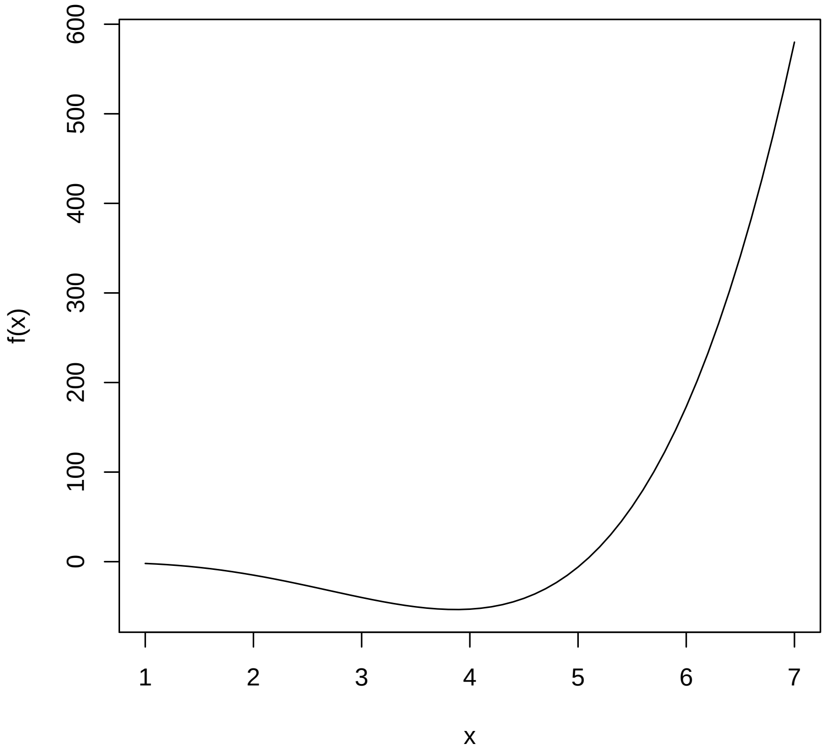
```
$root
[1] 5.058341

$f.root
[1] -0.0001784718

$iter
[1] 5

$init.it
[1] NA

$estim.prec
[1] 5e-06
```


f(x)
600
500
400
300
200
100
0
1
2
3
4
5
6
7
x