University of Cape Town
Department of Statistical Sciences
Statistical and High Performance Computing (2019)

Time allowed: 3 hours Total Marks: 50

Instructions:

- This is an open-book examination. You are allowed to consult hard-copy material you have brought into the examination venue, however no soft-copy material will be allowed.
- You may not use the internet during the course of the examination.
- Vula may only be used to download and upload the files required for the examination. You may not consult any other materials that is stored on Vula.
- Download the R Notebook template required for the examination from the 'Exam 2019' folder on Vula. Use this notebook to answer the questions (in any order) by writing R code chunks as well as explanatory comments where needed.
- Save both the R Notebook and the html/pdf version of your output as 'StudentNumber.Rmd' and 'StudentNumber.nb.html'/'StudentNumber.pdf'. Create an R object in your workfile named 'stdnumber' to store your student number. Ensure that you include comments in your code as well as the question number that the code relates to.
- Below you are explicitly told to create certain objects with specific names. Do so! The names of some of the required objects are shown using single quotes in brackets after the question.
- Ensure that you add comments to your code to explain the logic of your code where necessary!
- **Do not include any `rm` commands in your Rmd file!**
- **At the end of the examination ensure that your R code compiles!!!**

Question 1 [15 Marks]

Install the `gapminder` package to answer the following questions.

```
require(gapminder)
```

(a) Extract the `year` and `gdpPercap` values for `Angola`. Name the two variable names `Year` and `Gdp` respectively and store them both in a **dataframe** named 'Ang_gpd'. **[2]**

(b) How many continents have a median GDP per capita (gdpPercap) larger than that of Angola? ('gdp_g_Ang') **[2]**

(c) Which continents have a median gdpPercap larger than that of Angola? ('gdp_great') **[2]**

(d) Obtain the five number summary of gdpPercap for each of the countries in the `gapminder` object. Store the result as a **list** named 'gdp_fivenum'. **[1]**

(e) Use 'gdp_fivenum' to extract the median gdpPercap for each of the countries. ('med_gdp_count') **[4]**

Now identify which country has the 70^{th} largest median gdpPercap value. Create a character object with the specific country name. ('names_med_70')

(f) How many countries in the `gapminder` object have the small letter `e` in their name more than twice? (e.g. `Greece`) ('n_letter')

Hint:

```
?strsplit
```

[4]

Question 2 [12 Marks]

Suppose that the probability density function of a random variable X is

$$f(x, a) = ax^3 e^{-20x} \quad \text{for } x > 0.$$

(a) Use the 'integrate' function to show that $a = 26\ 666\frac{2}{3}$. ('a') **[3]**

```
f <- function(x,a){
 #add lines of code here
 ...
}

?integrate
```

(b) Create two functions named 'ex' and 'ex2' to calculate $x \times f(x, a)$ and $x^2 \times f(x, a)$ respectively. **[2]**

```
ex <- function(x,a){
  #add lines of code here
  ...
}

#do the same for ex2
```

(c) Now use numerical integration to show that $\text{var}(X) = 0.01$. ('varX') **[3]**

(d) Note that X is a Gamma random variable with shape parameter 4 and rate parameter 20. Write a function (named 'max_samp') to obtain the empirical distribution of the **maximum** value of a random sample of size 500 that is drawn from this particular Gamma distribution.

```
max_samp<- function(nsims){
 #nsims = the number of simulations undertaken
 #only add one argument named 'nsims'
 #add in more lines of code


}
```

Now use the 'max_samp' function to obtain 10 000 samples. Store the samples in an object named 'max_dist'. **[3]**

(e) Calculate the median of 'max_dist' and store the results in a object named 'med_max'. **[1]**

Question 3 [10 Marks]

A fair coin is tossed until two consecutive 'heads' are observed. Estimate the mean number of coin tosses.

Example sequences of the above experiment are shown below.

```
#H = Head is observed, T = Tail is observed

H T H H #number of tosses = 4
H T T T H H #number of tosses = 6
T H T T T T H T T H T H H #number of tosses = 13
```

Steps:

(a) Write a function named 'flip' that undertakes the appropriate sampling to output a numeric 1 if a 'head' is observed after the toss of a coin and a numeric 0 otherwise. **[2]**

```
flip<-function(){
 #don't add any arguments!
 #add lines of code here
}
```

(b) Write a function named 'rows_to_HH' that returns the number of tosses required for one experiment (i.e. until you observe two consecutive heads). **[6]**

```
rows_to_HH <- function(){
   #don't add any arguments!
   #add in lines of code

   #Note that you only have to return the
   #length of the sequence of 0's and 1's
   #name this object flip_count
   return(flip_count)
}
```

(c) Generate 10 000 random sequences using 'rows_to_HH' and store the results in an object named 'rand_HH'. Now estimate the mean number of tosses required until you observe two consecutive heads. ('mean_flips') **[2]**

Question 4 [13 Marks]

Single-season occupancy models are used to estimate the occupancy probability and conditional detection probability of a species in a particular region during a short period of time. The likelihood function for the model is

$$L(\psi, d|\boldsymbol{y}) = \prod_{i=1}^{n} \left[\psi \binom{K}{y_i} d^{y_i}(1-d)^{1-y_i} + (1-\psi) I(y_i = 0) \right]$$

where

- $\boldsymbol{y}$ is a vector that contains the number of species observed at each of the surveyed sites;
- n denotes the number of sites surveyed while K represents the number of times a site is surveyed;
- ψ $(0 < \psi < 1)$ denotes the probability that a species occurs at a site; while
- d $(0 < d < 1)$ denotes the conditional probability of detecting the species at a site if the site is occupied.

Use the following functions when attempting to answer the questions below.

```
occ_data <- function(psi, d, nsites, K){
 #simulate the number of species observed at each site
 #psi = occupancy prob (0<psi<1)
 #d = detection prob (0<d<1)
 #n = number of sites (integer)
 #K = number of surveys per site (integer)
 set.seed(1)

 #true occupancy
 z <- rbinom(nsites, size=1, prob=psi)

 #presence absence data
 y <- matrix(0, nrow=nsites, ncol=K)

 pres_index <- which(z==1)

 #only update y at locations where z=1
 for (i in pres_index){#the site visits
  for (j in 1:K){ y[i, j] <- rbinom(1, size=1, prob=d) }#end j
 }#end i

 y <- matrix(apply(y, 1, sum), ncol=1)
 return(y)
}
```

```
nlogl <- function(par, K, Y){
 #The negative loglikelihood function
 #par = [psi, d] (double)
 #K = the number of surveys per site (integer)
 #Y = the number of detections at each site (vector/matrix)

 pres_abs <- as.numeric(Y==0)
 psi <- par[1];   d <- par[2];  nsites <- length(Y)

 logl  <-0
 for (i in 1:nsites){
  t1_i <- psi*choose(K, Y[i])*( d^Y[i] )*( (1-d)^(K-Y[i]) )
  t2_i <- (1-psi)*pres_abs[i]
  likelihood_i <-  t1_i + t2_i
  logl <- logl + log(likelihood_i)
 }#end i
 return(-logl)
}
```

Below we consider the following strategy to estimate ψ.

(a) Assume that $\psi = 0.6$, $d = 0.8$, $n = 200$ and $K = 5$. Now use 'occ_data' to generate an example data set and name the object 'data_eg'. **[1]**

(b) Randomly split 'data_eg' into two data sets of equal size. Ensure that none of the elements in 'data_eg' are repeated and store the result in a matrix of dimension 100×2 named 'split_data'. **[2]**

(c) Calculate the maximum likelihood estimators $(\hat{\psi}_j,\ j = 1, 2)$ and the associated variances $(\hat{\sigma}^2_j,\ j = 1, 2)$ for both data sets using `optim`.

- Set the starting values of your `optim` call to `c(0.5, 0.5)`.
- Store the two `optim` calls as 'fit1' and 'fit2' respectively.
- If both `optim` calls converge, store the $\hat{\psi}_j$ values in an object named 'mu_psi' and the associated variances in an object named 'var_psi'.
- If either of your `optim` calls does not converge, set 'mu_psi' and 'var_psi' equal to `NA`.

[8]

(d) Use an appropriate `if` statement to estimate ψ as follows:

$$\hat{\psi} = \begin{cases} \dfrac{\hat{\sigma}^2_1\hat{\psi}_1 + \hat{\sigma}^2_2\hat{\psi}_2}{\hat{\sigma}^2_1 + \hat{\sigma}^2_2} & \text{if both } \texttt{optim} \text{ calls converge} \\ \texttt{NA} & \text{otherwise.} \end{cases}$$

Store your solution in an object named 'psi_hat'. **[2]**