STAT 565, Assignment 1

assignment due 9/28

Name 1 (First, Last) ______________________________

Name 2 (First, Last) ______________________________

Name 2 (First, Last) ______________________________

Instructions:

- This is an open book, notes, computer, R, and internet assignment. This assignment is team work, teams as announced by email or in class. No communication with other persons –except me– about the assignment is allowed, including communications with real persons over internet. Submit only your team's work. If you need clarification about questions please let me know. It is okay to ask questions.
- Please see the syllabus for submission details on take-home assignments. Only one copy per team is needed. All team members should make reasonable contribution to the work. Solutions to problems in Theory section are best submitted as pdf. Solutions to problems in Practice section are to be submitted only as R programs. Please write team members names at the top of your scripts and comment your code appropriately so that it is easy to follow.
- Solutions to problems in Theory part need solid algorithmic or mathematical justifications to get full credit. No simulation solutions will be accepted unless the problem specifies otherwise. However, you can use simulations to gain insight to the problem. Use precise mathematical notation, paying particular attention to clearly denote random variables and fixed values. Define any quantity used in the solution but not given in the problem.
- Solutions to problems in Practice part need a working R program that produces a clear answer for each part of the problem. For long problems, writing

multiple R files in the form of R functions and bundling them in a main.R might be a good idea. Each problem needs at least one separate R file. When submitting by email, please send compressed folder (.zip) for programs, but do not include any output. Programs should run to produce the output.

- All questions within the Theory part have equal value. All questions within the Pratice part have equal value. Theory and Practice parts also have equal value (50% each in this assignment).

- In this assignment only, we assume that only the standard uniform is available from a computer as a random variable, unless otherwise stated in the problem. All other random variables should be obtained starting with standard uniform. For example, if a question asks a method to simulate an exponential random variable, “using R function `rexp`" is not an acceptable answer. On the other hand, feel free to use any mathematical function in R including functions for calculating combinations, permutations, Gamma function, Beta function, logarithms, and trigonometric functions.

Part I: Theory

1. Let X follow a binomial distribution with number of trials $n = 10$ and probability of success in each trial $\theta = 0.3$. Write an algorithm that requires **only one standard uniform random variable** to simulate one observation from the distribution of X so that the efficiency for use of uniform random variables in simulating X is $1 : 1$.

2. Consider the probability density function $f_X(x) = 2x$ for random variable X defined on $x \in [0, 1]$. Two candidate methods, Algorithm 1 and Algorithm 2, to simulate X are given below.

Algorithm 1.

1. Simulate $y^* \sim \mathrm{Unif}(0, 2)$
2. Simulate $u^* \sim \mathrm{Unif}(0, 1)$
3. If $y^* \leq 2u^*$, accept u^* as from $f_X(x)$, else reject and go to 1

Algorithm 2.

1. Simulate $y^* \sim \mathrm{Unif}(0, 2)$
2. Simulate $w^* \sim \mathrm{Unif}(y^*/2, 1)$
3. Accept **all** w^* as from $f_X(x)$

Assess whether Algorithm 1 and Algorithm 2 produce samples from the target distribution $f_X(x)$, supporting your answer clearly with a mathematical argument for each case.

3. In general, we want to maximize the acceptance probability when sampling a given probability distribution using the rejection method. However, as we will later see in Markov chain Monte Carlo methods, when the idea of rejection is used in conjunction with complex computational algorithms, we may want to fix the probability of acceptance to optimize other aspects of an algorithm. Assume we want to sample the target random variable $X \sim \mathrm{Unif}(-2, 2)$ using a proposal random variable $Y \sim \mathrm{Unif}(0, b)$, $b > 0$, and the symmetry of the uniform distribution. Find b so that the probability of acceptance is 0.4.

4. Assume X has probability density function $f_X(x)$, with

$$f_X(x) \propto K(e^{-x^2/2})\mathbf{I}_{\{[4,\infty)\}},$$

where K is a constant and $\mathbf{I}_{\{A\}}$ is the indicator function that takes value 1 if A and 0 otherwise. In other words the support of $f_X(x)$ is $x \in [4, \infty)$.

a. Write an algorithm to simulate from $f_X(x)$ using a **rejection method.**

b. For your choice of proposal distribution, find the optimal value of parameter (of the proposal distribution) which maximizes the probability of acceptance.

5. Show whether it is possible to use the rejection algorithm to simulate from the target distribution of a random variable whose natural logarithm is normally distributed, using a normal distribution as a proposal distribution.

Part II: Practice

1. For purposes of this problem you can use built-in random variable generating functions in R. For example, you can use `rnorm` to simulate normal random variables. In this problem we will computationally verify that in the context of linear regression we can obtain the Ordinary Least Squares estimates for the regression coefficients using a matrix decomposition method. We let

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \epsilon_i, \; i = 1, 2, \cdots, n$$

where y_i is the ith response at the ith level of predictor variables x_1 and x_2, and ϵ_i are errors satisfying Gauss-Markov conditions,

$$E(\epsilon_i) = 0, Var(\epsilon_i) = 0, \; \forall i, \; Cov(\epsilon_i, \epsilon_j) = 0, \; \forall (i, j).$$

a. Build a linear regression model and simulate a data set to work on. Let $n = 100, \beta_1 = 3, \beta_2 = 2$. In a classical linear regression model, the predictors x_{1i} and x_{2i} are fixed and for convenience we will fix these values by simulating them from $\mathrm{Unif}(0, 10)$ independently, and also simulate $\epsilon_i \sim \mathrm{Nor}(\mu = 0, \sigma = 0.5)$, for $i = 1, 2, \cdots, n$. Given values of predictors, regression coefficients, and errors, calculate the values of response variable y_i. We have simulated a data set from a linear regression model and know the true model since we set the values of parameters.

b. Run `lm` function in R (see `?lm` in R) to perform a linear regression analysis (predictors x_1, x_2 and response y) and record $\hat{\beta}$ estimates from the R output.

c. Write an R **function** that takes this data set as input and outputs $\hat{\beta}$ using `svd` function in R for Singular Value Decomposition.

d. Compare estimates $\hat{\beta}$ from part (b) and (c).

2. *Urn models* are probabilistic models that are basis for many real life discrete stochastic models. An urn model formulates a discrete data generating mechanism using various well-defined sampling schemes of balls from an urn.

Consider the following urn model: There are α_W white balls, α_R red balls, and α_B blue balls in a well-mixed urn and no other balls. Draw a ball by simple random sampling. Observe its color and put it back to the urn together with another (new) ball of the same color. For example, if a red ball is drawn, then return that ball and an additional red ball to the urn before the second draw. Continue sampling this way and stop when n draws are made. This urn scheme is known as multivariate Pólya urn (basic Pólya urn has only two colored balls).

We are interested in finding a method to simulate the random variable that captures the number of balls of each color in n draws. For example if $n = 10$, a realization of the random variable could be $(n_W = 4, n_R = 0, n_B = 6)$, $(n = n_W + n_R + n_B)$. Note that we can simulate this process directly by sampling balls from an urn as described. However, this would be computationally inefficient if n is large, we want to simulate samples for a number of $\alpha = (\alpha_W, \alpha_R, \alpha_B)$ vectors, or even generalize the model to non integer α (see part (b) below). An easier way is to find probability distributions associated with the described stochastic process and sample those distributions.

a. Write an R function that takes arguments as $(\alpha_W, \alpha_R, \alpha_B), n,$ and returns a sample from this urn model using the random variable generation methods that we learned in class and any extrensions you can devise. As an example to illustrate that your program works use $\alpha = (\alpha_W = 1, \alpha_R = 5, \alpha_B = 2)$, and $n = 100$.

b. Generalize your function to a model with k types of balls, where the input is a vector $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_k)$ (and n), where $\alpha_i \in \mathbb{R}^+$, which means that $\alpha_i > 0$ does not need to be integer.