Stats 598z: Midterm exam 1

Important:
Write you name and PUID on all sheets, and include the number of sheets
There are 7 questions, question 1 has 8 points, all the rest have 7.
Attempt all questions, and when appropriate include a brief justification of your answer
Don't spend time polishing your answers as the main idea is more important.
Also, not all questions are equally easy, even if they are for the same points.
For maximum points use vectorization, but you won't lose too many points by looping

1. `movies` is a dataframe, each row giving information about a different movie. Its columns are `title, genre, director, year_of_release, imdb_rating, budget` and `box_office_earning`. Write `R` code to give:
   (a) the number of movies in this dataset (call this `len`), and the average IMDB rating of all movies.
   (b) the average profit (defined as `box_office_earning` minus `budget`) of all movies.
   (c) how many movies have `genre` equal to "`thriller`", and their titles.
   (d) how many movies were released between 1990 and 2000 (inclusive).
   (e) What does the command `sort(movies$budget)[-(6:(len-5))]` do?

2. (a) Briefly explain coercion and recycling. Give the output of `c(4,TRUE) + 5`, specifying their roles.
   (b) Explain the difference between `|` and `||`. What is the output of `c(0,1,5,0,1) | c(0,0,0,NA,NA)`?
   (c) For a list `my_list`, what is the difference between `my_list[1]` and `my_list[[1]]`?
   (d) What is the result of `my_vec <- 1:8; my_vec[c(TRUE,FALSE,TRUE)]`, including error/warning messages?

3. (a) `my_mat` is a matrix of both positive and negative numbers. Write a few lines of `R` to print a sub-matrix consisting only of the rows of `my_mat` whose elements sum to a positive value.
   (b) Explain briefly the relation between a dataframe and a list.
   (c) What is the output of `matrix(c(1,2,3) > c(2,3), nrow=2,ncol=3)`?
   (d) `locn` is a vector of doubles of length `n`. Write `R` code to create a matrix `dist_mtrx`, whose `(i,j)`th element equals $(\texttt{locn[i]-locn[j]})^2$.

4. (a) Briefly explain why vectorization is faster than looping.
   (b) Briefly explain the difference between `*` and %*%.
   (c) `my_obs` is a vector of numbers interpreted as $(\texttt{len1}, \texttt{obs}_1, \texttt{obs}_2, \ldots, \texttt{obs}_{\texttt{len1}}, \texttt{len2}, \texttt{obs}_1, \texttt{obs}_2, \ldots, \texttt{obs}_{\texttt{len2}}, \ldots)$. Thus, the first number gives the length of a set of observations that follows it, this set is followed by the length of the next set, and so on. Write a few lines of `R` code to return the mean of each set, so that $(3, 1, 2, 3, 2, 5, 3)$ returns $(2, 4)$. You can use loops.

5. (a) For the movies dataset of question 1, give ggplot commands to plot budget versus year_of_release for different genres, each genre having a different color.

   (b) In the dataset movies, each director can have multiple movies. Write R code to provide the 10 directors with the highest average IMDB scores across their movies. Here, a useful command might be unique, which accepts a vector as input, and returns the unique elements of the vector. You can use loops.

   (c) Briefly explain what ridge regression is and why it is useful.

6. (a) Give R code to fit a linear model giving box_office_earning as a function of budget, genre, and year_of_release. Use any geometry of your choice.

   (b) Suppose you decided to also include the additional variable imdb_rating to predict box_office_earning. Comment on the following two lines:

      i. The resulting model will have a smaller error on the training data: True/False/Cannot tell.

      ii. The resulting model will have a smaller error on the test data: True/False/Cannot tell.

   (c) You want to use one of the previous two models to make predictions for a new movie that's coming out this year. Briefly explain how would you pick the best model using the data in movies.

7. (a) Recall that rnorm(n) returns n Gaussian variables with mean 0. Write an R function that accepts two inputs thresh and trunc. The function keeps sampling Gaussians until their sum crosses thresh, or the number of Gaussians equals trunc, whichever occurs first. At this point it returns all the Gaussians generated so far. You can use loops.

   (b) Suppose you run the lines:

```
my_func <- function(a=1,b,c,d=3) {
   b/a + d/c
 }
 a<-2; b<-3; c<-1;d<-4
 my_func(d,c,b,a)
```

   What is the output? Explain why.

   (c) Suppose you further run the line:

```
my_func(,c=2,3)
```

   What is the output? Explain why.

   (d) Suppose you further run the line:

```
my_func(c=2,3)
```

   What is the output? Explain why.