Sample Midterm, Statistics 133

1. What is meant by "vectorized calculations" in `R`? Provide an example.

   If we have a vector `x`, an expression such as, $x + 2$ or x^3, is vectorized in that the computation is performed on each element of the vector, i.e. 2 is added to each element of x or each element of x is cubed. There is no need to loop of each element of the vector to perform the computation.

2. Describe two important differences between a data frame and a matrix in `R`.

   A data frame is essentially a list of vectors of the same length, whereas a matrix is essentially a vector with shape information.

   Data frames can have columns/vectors that are different types, whereas all values in a matrix must be the same primitive element.

   Data frames can be indexed with $

3. Data on 37 parents of babies born at Kaiser Hospital in the 1960s is available in a data frame called parents. The variables age, ed, ht, and wt are the mother's age, education level, height and weight. The variables that start with the letter d are corresponding variables for the fathers.

```
> head(parents)
   age   ed   ht  wt  dage ded   dht  dwt marital inc
1 27 College 62 100 31 College 65 110 Married [2500, 5000)
2 33 College 64 135 38 College 70 148 Married [7000, 8000)
3 28 High School 64 115 32 Some High School NA NA Married [5000, 6000)
4 36 College 69 190 43 Some College 68 197 Married [12500, 15000)
5 23 College 67 125 24 College NA NA Married [2500, 5000)
6 25 High School 62 93 28 High School 64 130 Married [7000, 8000)
```

   Provide the return value for each of the following expressions:

```
dim(parents)
[1] 37 10
```

```
class(parents\$marital)
[1] "factor"
```

   Write an R expression to find the subset of parents where the mother is over 40.

```
parents[ parents$age > 40, ]
```

   Write an R expression using an apply function to return the class of each variable in the data frame.

```
sapply(parents, class)
```

Write one R expression using an apply function to return the number of NAs in each variable (recall that there is an is.na() function returns a logical indicating the presence of NAs)

```
sapply(parents, function(x) sum(is.na(x)))
```

4. Here is a list in R,

```
> x
$a
[1] 0.03895442 0.77658866 0.83532332

$b
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

Write one line of `R` code to extract the first row of the matrix.

```
x$b[1, ]
```

5. Suppose we have a matrix `m` in `R`, and we've just executed the following:

```
> dim(m)
[1] 5000    3
> head(m)
           [,1]       [,2]       [,3]
[1,] -2.2468718 -0.7733515 -3.4332337
[2,]  0.5771791 -0.7058552  0.8052004
[3,] -1.0125651 -0.2699696 -1.1368809
[4,] -0.2504269 -1.1205857 -0.3498572
[5,]  2.6747195  0.2550678  0.1225329
[6,]  1.0095424 -1.2900079  0.1387224
```

We need to create a vector containing the sum of the *squared* entries in each row of `m`. Write `R` code to do this in two different ways:

(a) using a `for loop`

```
sumM = rep(0, nrow(m))
for (i in 1:nrow(m) ){
  sumM[i] = sum(m[i, ]^2)
}
```

(b) using the `apply` function

```
apply(m, 1, function(x) sum(x^2))
```

6. Write down what the value of x will contain after each line of R code, if the commands are executed sequentially.

```
> x = seq(0, 8, length = 5)
0 2 4 6 8
> x[x<4] = NA
NA NA 4 6 8
> x[5] = 10
NA NA 4 6 10
> x[] = 0
0 0 0 0 0
> x = 12
12
```

7. Someone wants to study the distribution of the sum of three rolls of a die. To do this she designs a simulation study. In the first step, she writes a function to generate the sum of three random tosses of a fair die. In the second step she uses this function to generate 1,000 of these sums.

(a) Write the function for the first step.

```
sum3 = function(){
  sum(sample(1:6, 3, replace = TRUE))
}
```

(b) Write one line of code that uses the function from the first step to generate the 1,000 random sums

```
replicate(1000, sum3() )
```

8. We want to compute the sum of the absolute deviations from the median for a vector. For example for a vector `x = 1:3`, `x` has a median of 2, and the absolute deviations from the median are 1, 0, and 1 so the sum of the absolute deviations from the median is 2.

Write a function named `sadm` that computes this statistic for a vector. The function has two parameter: `x` is required and holds the numeric vector that will be operated on; and `na.rm` which determines whether NAs are to be removed from the computation. The `na.rm` parameter has a default value of `FALSE`.

```
sadm = function(x, na.rm = FALSE){
  if (na.rm) x = x[ !is.na(x) ]
  sum(abs(x - median(x)))
}
```