

Part A

Write your own functions to do the following:

```
myAvg <- function(anyVector) {  
  compute average  
  return average  
}
```

```
mySD <- function(anyVector) {  
  compute Std Dev  
  return Std Dev  
}
```

```
myRange <- function(anyVector) {  
  compute Range, calculated as Max - Min  
  return Range  
}
```

Read the CardioGoodnessFit.csv file and apply these functions to the last two columns – Income and Miles.

The output should look like the following:

```
Income (Average, SD, Range): 53719.58, 16506.68, 75019  
Miles (Average, SD, Range): 103.19, 51.86, 339
```

Part B

Consider the following code provided by Dr. Soyer. Please explain what the code in the enclosed box does in the context of the birthday problem. Please make sure that you explain why we need two loops and please explain the role of `p[i]` and `prod`.

```
## Birthday problem
```

```
K=seq(10,70,by=5)  
p=rep(0,length(K))
```

```
for (i in 1:length(K)) {  
  n = 1  
  prod=1  
  while (n<K[i]) {  
    prod=prod*(1-n/365)  
    n=n+1  
  }  
  p[i]=1-prod  
}
```

```
plot(K, p, type="l",  
      xlab="number of people",  
      ylab="Prob(same birthday)",  
      col="blue",lwd=2)
```

Please comment the portions in the enclosed boxes in the following program provided by Dr. Soyer. Please make sure that you explain the meaning and relevance of all variables in the enclosed boxes.

```
## Stock prices
# DAILY CLOSING PRICES NOV 1, 1993 TO APR 3, 2003
```

```
all=read.table("prices.txt",header=TRUE)
msft=all[,2]
ge=all[,4]
n=length(msft)
d_msft=rep(0,n)
r_msft=rep(0,n)
d_ge=rep(0,n)
r_ge=rep(0,n)
```

```
for (t in 2:n) {
  d_msft[t]=msft[t]-msft[t-1]
  r_msft[t]=d_msft[t]/msft[t-1]
  d_ge[t]=ge[t]-ge[t-1]
  r_ge[t]=d_ge[t]/ge[t-1]
}
```

```
par(mfrow=c(1,2))
plot(r_msft,
     type="l",
     main="Daily Returns 11/1/1993 to 4/3/2003",
     xlab="day",
     ylab="Returns",
     col="blue")
hist(r_msft,
     col="blue",
     xlab="Microsoft Returns",
     prob=T,
     nclass=20)
```

```
id_msft=(d_msft>0)
id_ge=(d_ge>0)

ftable(id_ge,id_msft)
```

Part C

The following code segment will be helpful in completing this portion. Run it and see what happens.

```
s = seq(from = 0.1, to = 1, length.out = 10)
mys <- ""
for (each in s) {
  a <- sprintf("%.3f", pi)
  mys <- paste(mys, a, sep=" ")
}
print(mys)
```

The following site will also prove useful:

http://www.cookbook-r.com/Strings/Creating_strings_from_variables/

Your task is to ask the user for input for n , where n is the number of trials in the binomial distribution. Based on the user's input your task is to generate a binomial probability table as shown in <http://www.statisticshowto.com/tables/binomial-distribution-table/>. For example, if the user inputs 4, your output should look like the following:

 $n=4$ [illegible]

result.