homework1.rmd

Sam Budoff

09/18/2014

Homework 1

Cancer Data Frame

1. Load the data set into R and make it a data frame called `cancer.df`. (2 points)

```
cancer.csv<-read.csv('cancer.csv')   # load cancer.csv
cancer.df <- data.frame(cancer.csv)  # convert into dataframe
```

2. Determine the number of rows and columns in the data frame. (2)

```
nrow(cancer.df)  # count number of rows
```

```
## [1] 42120
```

```
ncol(cancer.df)  # count number of columns
```

```
## [1] 8
```

3. Extract the names of the columns in `cancer.df`. (2)

```
cnames <- colnames(cancer.df)  # create an array of character strings
cnames                         # display column names
```

```
## [1] "year"       "site"       "state"      "sex"        "race"
## [6] "mortality"  "incidence"  "population"
```

4. Report the value of the 3000th row in column 6. (2)

```
cancer.df[300,6]
```

```
## [1] 47.27
```

5. Report the contents of the 172nd row. (2)

```
cancer.df[172,]
```

```
##     year                          site  state  sex  race mortality
## 172 1999 Brain and Other Nervous System nevada Male Black         0
##     incidence population
## 172         0      73172
```

6. Create a new column that is the incidence *rate* (per 100,000) for each row.(3)

```
cancinc<-cancer.df$incidence                      # extract incidence
pop<-cancer.df$population                         # extract population
per<-(pop/10000)                                  # make population per 10000 individuals
cancrate<-cancinc/per                             #calculate rate per 10000
cancer2.df<-cbind(cancer.df, list(rate=cancrate)) # Add rate column, while renaming cancer.df
```

7. How many subgroups (rows) have a zero incidence rate? (2)

```
sum(cancer2.df$rate ==0)
```

```
## [1] 23191
```

8. Find the subgroup with the highest incidence rate.(3)

```
maxrate<- max(cancer2.df$rate)
rmaxrow <- function() {
  for (i in seq(1,42120,1)){
    if (cancer2.df[i,9] == maxrate){
      mrownum<-print(i)
      }
    }
  }                                 # this for loop allows identification of the row number of interest
rmaxrow()                           # this command displays the row number of the subgroup of interst
```

```
## [1] 5797
```

Running the function outputs the value 5797, which can be used to directly call the row of the dataframe with the highest incident rate

```
cancer2.df[5797,]
```

```
##      year     site                 state  sex  race mortality incidence
## 5797 1999 Prostate district of columbia Male Black     88.93       420
##      population  rate
## 5797     160821 26.12
```

Data Types

1. Create the following vector: `x <- c("5","12","7")`. Which of the following commands will produce an error message? For each command, Either explain why they should be errors, or explain the non-erroneous result. (4 points)

    ```
    max(x)
    ```

    This command will produce an erroneous result as it will find the maximum value string of characters, in this case "7" sort(x) *This command will sort, again, on the basis of the character string value, not the implied numerical value, in this case "12", "5", "7"* sum(x) *This command will result in an error message as characters can not be added*

2. For the next two commands, either explain their results, or why they should produce errors. (3 points)

```
y <- c("5",7,12)
```

This concatonation will create a vector of character strings becasue the "5" vector is a character type and the default of R is to make all concatonated arguments the same class type, with character having a higher priority than numberic y[2] + y[3] *This equation will fail because the numeric 7 and 12 are converted to character strings when concatonated with "5" in y. Character strings can not undergo addition*

3. For the next two commands, either explain their results, or why they should produce errors. (3 points)

```
z <- data.frame(z1="5",z2=7,z3=12)
```

This command results in teh creation of a 3 column data frame, where the first column is a list of character strings, and the other columns are lists of numerics. z[1,2] + z[1,3] *This command will result in addition of the numerics 7 and 12 to yield the numeric 19. THis occurs becasue 7 and 12 are the respective first rows of the second and third columns of the data frame z.*

Data Structures

1. $(1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1)$

```r
c(seq(1,8),seq(7,1))
```

```
##  [1] 1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
```

2. $(1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5)$

```r
x=seq(1,5)
rep(x,x)
```

```
##  [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
```

3. $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$

```r
diag(-1,3,3)+matrix(1,3,3)
```

```
##      [,1] [,2] [,3]
## [1,]    0    1    1
## [2,]    1    0    1
## [3,]    1    1    0
```

4. $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \\ 1 & 8 & 27 & 64 \\ 1 & 16 & 81 & 256 \\ 1 & 32 & 243 & 1024 \end{pmatrix}$

```
s=seq(1:4)
t(matrix(c(s,s^2,s^3,s^4,s^5),4))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    1    4    9   16
## [3,]    1    8   27   64
## [4,]    1   16   81  256
## [5,]    1   32  243 1024
```

Basic Programming

1. Let $h(x, n) = 1 + x + x^2 + \ldots + x^n = \sum_{i=0}^{n} x^i$. Write an R program to calculate $h(x, n)$ using a `for` loop. (5 points) **To use the following program type `Ztrans(x,n)` where x and n correspond, respectively, to x and n in $h(x, n) = 1 + x + x^2 + \ldots + x^n = \sum_{i=0}^{n} x^i$.**

```
w=0
z=0
Ztrans<-function(x,n){
  for (i in seq(1,n)) {
  w=x^i
  z=z+w
  }
   print(z)
  }
```

2. If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Write an R program to perform the following calculations. (5 points) **Note that the output of this function is a logical value, but answers the question.**

```
diviscounter<-function(x,y,z) {
    k=seq(1:x)
    subK<-(k%%y|z==0)
    summary(subK)[3]
  }
```

```
a. Find the sum of all the multiples of 3 or 5 below 1,000. (3, [euler1])
```

Note that the output of this function is a logical value, but answers the question.

```
k=seq(1:1000)
subK<-(k%%3|5==0)
summary(subK)[3]
```

```
##  TRUE
## "667"
```

Alternatively, using my function:

```
diviscounter(1e3,3,5)
```

```
##  TRUE
## "667"
```

```
b. Find the sum of all the multiples of 4 or 7 below 1,000,000. (2)
```

Note that the output of this function is a logical value, but answers the question.

```
m=seq(1:1e6)
subM<-(m%%4|7==0)
summary(subM)[3]
```

```
##     TRUE
## "750000"
```

Alternatively, using my function:

```
diviscounter(1e6,4,7)
```

```
##     TRUE
## "750000"
```

3. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be $(1, 2, 3, 5, 8, 13, 21, 34, 55, 89)$. Write an R program to calculate the sum of the first 15 even-valued terms. (5 bonus points, [euler2]) **To operate this function subtract 2 from the Fibronaci sequence of interest's length, becasue 1 and 2 are implied in this function. For example, if you would like the sum of the the first 15 even terms wright*** `fib(13)`.**

```
g=0
u=1
i=2
fib<- function(x) {
  while (x>0) {
    u=u+i
    x=x-1
    if (u%%2==0) {g=g+u}
    i=i+u
    x=x-1
    if (i%%2==0) {g=g+i}
  }
  print(g)
}
fib(13)                      # print sum of first 15 even values in the Fibbonacci sequence
```

```
## [1] 796
```