Homework_2

Problem 1

Construct the **matrix**: code in three different ways to generate a 5 x 5 matrix that looks like this:(Hint:Try to use rbind() and cbind())

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11   16   21
## [2,]    2    7   12   17   22
## [3,]    3    8   13   18   23
## [4,]    4    9   14   19   24
## [5,]    5   10   15   20   25
```

```
# There are many ways of thing this. Think of others!
# Example 1:
my_matrix_method1 <- matrix(1:25, nrow = 5, ncol = 5, byrow = F)
```

```
# Example 2:
v=1:5
my_matrix_method2 <- cbind(v, v + 5, v + 10, v + 15, v + 20)
my_matrix_method2
```

```
##      v
## [1,] 1  6 11 16 21
## [2,] 2  7 12 17 22
## [3,] 3  8 13 18 23
## [4,] 4  9 14 19 24
## [5,] 5 10 15 20 25
```

```
# Example 3:
 v = seq(1, 21, 5)
 my_matrix_method3 <- rbind(v, v + 1, v + 2, v + 3, v + 4)
 my_matrix_method3
```

```
##   [,1] [,2] [,3] [,4] [,5]
## v    1    6   11   16   21
##      2    7   12   17   22
##      3    8   13   18   23
##      4    9   14   19   24
##      5   10   15   20   25
```

Problem 2

Find the sum of all numbers below 1000 that can be divisible by 3 or 5 (Hint: Conditionals)

```
v = 1:999 # below 1000
sum(v[v %% 3 == 0 | v %% 5 == 0])
```

```
## [1] 233168
```

Problem 3

Find the sum of the even valued terms of the Fibonacci sequence that do not exceed 4,000,000. (Hint: refer to page 110 in our week1 slides)

```
# One way of doing this
a <- 0
b <- 1
v <- c(0)
while (b <= 4000000) {
if (b %% 2 == 0) v = append(v, b)
temp <- b
b <- b + a
a <- temp
}
v
```

```
##  [1]       0       2       8      34     144     610    2584   10946
##  [9]   46368  196418  832040 3524578
```

```
sum(v)
```

```
## [1] 4613732
```

Another way

```
# Another way
i <- 2
x <- numeric(100) # guess the rough number
x[1] = 1
x[2] = 2
while (x[i] < 4e6) {
x[i+1] <- x[i-1] + x[i]
i <- i + 1
}
x <- x[-i]
sum(x[x %% 2 == 0])
```

```
## [1] 4613732
```

Problem 4

Magic squares are square matrices where the row, column, diagonal and counterdiagnoal sums all equal the same number. Use R to validate that my_mat is a magic square

```
my_mat = matrix(c(8, 3, 4, 1, 5, 9, 6, 7, 2), ncol = 3)
print(my_mat)
```

```
##      [,1] [,2] [,3]
## [1,]    8    1    6
## [2,]    3    5    7
## [3,]    4    9    2
```

```
 rowSums(my_mat)
```

```
## [1] 15 15 15
```

```
 colSums(my_mat)
```

```
## [1] 15 15 15
```

```
 sum(diag(my_mat))
```

```
## [1] 15
```

```
 sum(diag(my_mat[3:1,]))
```

```
## [1] 15
```

Problem 5

What is wrong with the following code? How would you fix it?(Hint: conditionals)

```
x <- 1
if ( x = 1 ) {cat("x is 1")
} else{
    cat("x is not 1")
  }
```

```
#The logical equality operator (==) should have been used.
```

Problem 6

Write a function that calculates the median absolute deviation (MAD) of a numeric vector. The median absolute deviation is a robust alternative to standard deviation as a measure of dispersion. It is defined for a vector X as:

```
MAD = median(|Xi - median(X)|)
```

```
MAD = function(x) {
stopifnot(is.numeric(x))
return (median(abs(x-median(x))))
}
```

Problem 7

The prime factors of 13195 are 5, 7, 13 and 29. What is the largest prime factor of the number 600851475143. (Hint: Write a function)

```
prime.factor <- function(n) {
factorization <- numeric(0)
d <- 2
while(n > 1) {
while(n %% d == 0) {
factorization <- append(factorization, d)
n <- n / d
}
d <- d + 1
}
max(factorization)
}
```

Problem 8

A palindromic number reads the same, both forward and backward. The largest palindrome made from the product of two 2-digit numbers is:

```
9009 = 91 * 99
```

Find the largest palindrome made from the product of two 3-digit numbers.

```
keep = 0
for (i in 999:100)
{
 for (j in 999:100)
 {
 palidrome = i*j
 if (as.character(i*j) == paste(rev(strsplit(as.character(i*j),"")[[1]]),
collapse="")
 & keep < palidrome ) {
 keep = palidrome
 }
 }
}
print (keep)
```

```
## [1] 906609
```

Problem 9

Explain in your own words what Vectorizing a function does and when it???s appropriate. Write a vectorized function that will examine an input array of numbers and return a logical array of whether each number is a palindrome. (Hint: Page 123 in week1 slides)

Problem 10 (In-class exercise)

1. Come up one or two questions you want to answer from Tips dataset.
2. Present your code and results.(Hint: Go over page 21-23 in week2 slides)

```
library(reshape2) # don't forget to load the package
head(tips)
```

```
##   total_bill  tip    sex smoker day   time size
## 1      16.99 1.01 Female     No Sun Dinner    2
## 2      10.34 1.66   Male     No Sun Dinner    3
## 3      21.01 3.50   Male     No Sun Dinner    3
## 4      23.68 3.31   Male     No Sun Dinner    2
## 5      24.59 3.61 Female     No Sun Dinner    4
## 6      25.29 4.71   Male     No Sun Dinner    4
```

```
summary(tips)
```

```
##    total_bill         tip             sex      smoker      day
##  Min.   : 3.07   Min.   : 1.000   Female: 87   No :151   Fri :19
##  1st Qu.:13.35   1st Qu.: 2.000   Male  :157   Yes: 93   Sat :87
##  Median :17.80   Median : 2.900                           Sun :76
##  Mean   :19.79   Mean   : 2.998                           Thur:62
##  3rd Qu.:24.13   3rd Qu.: 3.562
##  Max.   :50.81   Max.   :10.000
##      time          size
##  Dinner:176   Min.   :1.00
##  Lunch : 68   1st Qu.:2.00
##               Median :2.00
##               Mean   :2.57
##               3rd Qu.:3.00
##               Max.   :6.00
```

Example: Is there a gender difference in the tipping habits?

1. Compare the average of tips between different genders

```
dcast(tips, sex ~ ., value.var='tip', fun=mean)
```

```
##      sex        .
## 1 Female 2.833448
## 2   Male 3.089618
```

2. Put the sizes of groups into consideration

```
  dcast(tips, sex ~ size, value.var='tip', fun=mean)
```

```
##      sex        1        2        3        4    5    6
## 1 Female 1.276667 2.528448 3.250000 4.021111 5.14 4.60
## 2   Male 1.920000 2.614184 3.476667 4.172143 3.75 5.85
```

3. Compare the average of bills between different genders

```
  dcast(tips, sex ~ . , value.var='total_bill', fun=mean)
```

```
##      sex        .
## 1 Female 18.05690
## 2   Male 20.74408
```

- Open question: How can we compare between gender groups by calculating tip/total_bill?*

```
tips2<-tips

tips2$ratio<-tips2$tip/tips2$total_bill
dcast(tips2,sex~time,value.var = "ratio",fun=mean)
```

```
##      sex    Dinner     Lunch
## 1 Female 0.1693216 0.1622849
## 2   Male 0.1554065 0.1660826
```

```
dcast(tips, time ~ . , value.var='total_bill', fun=mean)
```

```
##     time        .
## 1 Dinner 20.79716
## 2  Lunch 17.16868
```
