Problem Set 3: Graphics

Shane Mueller

1. Create a new dotchart function based on the matplot version we did in class (do not use the built-in dotchart function). It should:

a. have the following arguments:

`mydotchart(data,labels, colors,main, xlab,ylab,xlim,ylim,lty,normalize,col,pch,cex,subsets)`

b. Take a matrix of values rather than just a single series. Each column should represent a different series, so data <- cbind(c(1,3,5),c(10,2,3)) should work.

c. Permits normalization if normalize parameter is TRUE. If so, it will rescale everything to the range 0 to 1.0, based on the smallest versus largest value in the data matrix. Assume all values will be positive, and you can do this for a data matrix as follows: `data <- (data - min(data))/(max(data)-min(data))`

d. col, pch, cex, lty, etc. should do something reasonable, like control the size of individual points, entire series, or the whole chart.

Demostrate how this works with the following data and the set of examples given, as well as others you feel relevant.

Hints:

- if you use rmarkdown, you can control figure size with the fig.width and fig.height arguments, which are specified in inches:

```
##figures here will be 3x4 inches
```

- Don't worry too much if there are conditions that won't work–just be sure it works for the example data below. For instance, if you give it a different data set, maybe the xlim or ylim won't work well by default. Ideally you can choose default values that work well, but if you can't use ones that work.

http://stackoverflow.com/questions/28370249/correct-way-to-specifiy-optional-arguments-in-r-functions

You can test your function on a data set like this:

```
set.seed(100)
data2 <- data.frame(q1=sample(letters[1:10],100,replace=T),
                    q2=sample(letters[1:10],100,replace=T),
                    q3=sample(letters[1:10],100,replace=T),
                    q4=sample(letters[1:10],100,replace=T),
                    q5=sample(letters[1:10],100,replace=T))

datatable2<-apply(data2,2,table)
```

In addition, validate other parameters work as expected.

Demonstrate it works with the following:

```
mydotchart(datatable2)
mydotchart(datatable2[,1:2])
mydotchart(datatable2[,1])  ##this might break because it is interpreted as a vector
mydotchart(as.matrix(datatable2[,1]))  #this should work
mydotchart(datatable2[,1:3])
mydotchart(datatable2,col=1:5)
mydotchart(datatable2,col=1:5,pch=16)
mydotchart(datatable2,col=1:5,pch=16,cex=2.5,main="Everything",xlab="Value", ylab="Category")
mydotchart(datatable2,col=1:5,pch=16,cex=2.5,main="Everything normalized",xlab="Value", ylab="Category"
```

2. Correlating word frequency with SCRABBLE scores

The following data frame specifies the English letter frequency of letters, the points earned in Scrabble, and the number of Scrabble tiles.

```
lf <- c(8.167,1.492,2.782,4.253,12.702,2.228,2.015,6.094,
        6.966,0.153,0.772,4.025,2.406,6.749,7.507,1.929,
        0.095,5.987,6.327,9.056,2.758,0.978,2.36,0.15,1.974,0.074)/100
pts <- c(1,3,3,2,1,4,2,4,1,8,5,1,3,1,1,3,10,1,1,1,1,4,4,8,4,10)
tiles <- c(9,2,2,4,12,2,3,2,9,1,1,4,2,6,8,2,1,6,4,6,4,2,2,1,2,1)
lf.table <- data.frame(LETTERS,
                       freq=lf,
                       points=pts,
                       ntiles=tiles)
```

For any word, you can split it into its letters, and then compute some statistics based on this scoring. The following computes the sum of the inverse letter frequency of the letters, the total scrabble points, the mean numbers of tiles of the letters in the word, and the length of the word:

```
scoreme <- function(word)
{

  lets <- strsplit(splus2R::upperCase(word),"")[[1]]
  data <- matrix(0,ncol=4,nrow=length(lets))

  for(i in 1:length(lets))
  {
    index <- which(lets[i]==LETTERS)
    data[i,1] <- lf.table$freq[index]
    data[i,2] <- lf.table$points[index]
    data[i,3] <- lf.table$ntiles[index]

  }
  list(suminvfreq= sum(1/data[,1]),
       points=sum(data[,2]),
       meantiles=mean(data[,3]),
       length=length(lets))
}
```

You can access these like this, which gives you some statistics based on the letters in a word:

```
horses <- scoreme("HORSES")
horses
```

```
## $suminvfreq
## [1] 85.91667
##
## $points
## [1] 9
##
## $meantiles
## [1] 6
##
## $length
## [1] 6
```

```
print(horses$points)
```

```
## [1] 9
```

The following lists a set of words, along with their rank frequency (lower meaning more frequent), and their total frequency (number of occurrences in a large corpus). For each word, compute the four statistics above using the scoreme function. You can add the results of the scoreme function to each row of the test data frame, starting by adding empty variables:

```
test <- read.table(text='rank word frequency
1081  CUP      1441306
2310  FOUND     573305
5285  BUTTERFLY 171410
7371    brew       94904
11821 CUMBERSOME 39698
17331 useable        17790
18526 WHITTLE    15315
25416 SPINY       7207
27381 uppercase         5959
37281     halfnaked         2459
47381     bellhop       1106
57351     tetherball        425
7309        attic       2711
17311     tearful       542
27303     tailgate  198
37310     hydraulically         78
47309     unsparing         35
57309     embryogenesis     22 ',header=T)[,c(2,1,3)] ##reorder colums

test$meantiles <- NA
test$suminvfreq <- NA
test$points <- NA
test$length <- NA
```

- Make a set of plots showing both rank frequency and total frequency by each of the four statistics based on individual letters (a total of 8 figures). Compute the correlation between each pairing using the cor() function, and add this to the plot either in the title or on the main plot area (use paste() and round() if necessary). Be sure all axis and figures are labeled and described using human language.
- Discuss whether you see any relationships between either of the two corpus statistics on word frequency and any of the three letter-based statistics
- If you see a relationship, try to suggest why you might be seeing it. Explain why it is positive or negative.
- Discuss relative advantages of looking at rank frequency versus raw frequency. Hint: higher correlations are not always more meaningful.