```
setwd("/Users/sarpotd/Desktop/Coursera/Recommender Systems/week2/Assignment2/")
data1 <- read.csv("recsys-data-ratings.csv", header=FALSE)
colnames(data1) <- c("User", "Movie", "Rating")


data2 <- data1[sort(data1$Movie,index.return=TRUE)$ix,]

data3 <- table(data2$Movie,data2$User)
write.table(data3,file="a",sep =",",row.names=TRUE,col.names=TRUE)
```

Notes:

Simple Analysis

```
> table(data3["121", ] == 1 & data3["122", ] == 1)["TRUE"]/table(data3["121", ] == 1)
["TRUE"]
     TRUE
0.9480046
```

```
movie <- "1572"
movie_corr <- function(c) (table(data3[movie, ] == 1 & data3[c, ] == 1)["TRUE"]/
table(data3[movie, ] == 1)["TRUE"])
temp <- as.matrix(mapply(movie_corr,1:nrow(data3)))
colnames(temp) <- "Prob"
data4 <- cbind(data3,temp)
data5 <- round(data4[sort(data4[,"Prob"],index.return=TRUE, decreasing=TRUE)$ix,]
[1:6,"Prob"],2)
```

Complex Analysis

```
(table(data3["121", ] == 1 & data3["122", ] == 1)["TRUE"]/table(data3["121", ] == 1)
["TRUE"])/(table(data3["121", ] == 0 & data3["122", ] == 1)["TRUE"]/table(data3["121", ]
== 0)["TRUE"])
    TRUE
4.736006
```

```
movie <- "1572"
movie_corr <- function(c) ((table(data3[movie, ] == 1 & data3[c, ] == 1)["TRUE"]/
table(data3[movie, ] == 1)["TRUE"])/(table(data3[movie, ] == 0 & data3[c, ] == 1)["TRUE"]/
table(data3[movie, ] == 0)["TRUE"]))
temp <- as.matrix(mapply(movie_corr,1:nrow(data3)))
colnames(temp) <- "Prob"
data4 <- cbind(data3,temp)
data4 <- na.omit(data4)
data5 <- round(data4[sort(data4[,"Prob"],index.return=TRUE, decreasing=TRUE)$ix,]
```

[1:5,"Prob"],2)

Examples

Suppose that you were assigned movie IDs 11, 121, and 8587. Your submission for the first part (simple formula) would be:

11,603,0.96,1892,0.94,1891,0.94,120,0.93,1894,0.93
121,120,0.95,122,0.95,603,0.94,597,0.89,604,0.88
8587,603,0.92,597,0.90,607,0.87,120,0.86,13,0.86
...and your submission for the second part (advanced formula) would be:

11,1891,5.69,1892,5.65,243,5.00,1894,4.72,2164,4.11
121,122,4.74,120,3.82,2164,3.40,243,3.26,1894,3.22
8587,10020,4.18,812,4.03,7443,2.63,9331,2.46,786,2.39

Note that with rounding, some entries will appear to tie. Be sure to preserve the order of the output from the original algorithm.