QUESTION 4: Game of Thrones Analysis

a) Using R, determine which directors directed the five most highly rated episodes.

```
got <- read.csv("GOT.csv")
gotbestrate5 <- got[order(got$IMDb_Rating, decreasing = TRUE), ]
gotbestrate5["Director"][1:5,]
```

1) David Nutter
2) Miguel Sapochnik
3) Miguel Sapochnik
4) Miguel Sapochnik
5) Matt Shakman

b) Using R, determine the names of the highest and lowest rated episodes

```
got <- read.csv("GOT.csv")
gotbestrate5 <- got[order(got$IMDb_Rating, decreasing = TRUE), ]
gotbestrate5["Episode_Name"][1:1,]
gotbestrate5["Episode_Name"][73:73,]
```

Highest: The Rains of Castamere
Lowest: The Iron Throne

c) Using R, calculate and report the average IMDb_Rating for each unique writer-director pairing. Which pairing produces the highest rated episodes, on average? Which pairing produces the lowest rated episodes, on average?

```
# sort director-writer pairs
gotpairs <- got[order(got$Director,got$Writer, decreasing = TRUE), ]
droplevels(gotpairs)

# This for loop will go through every row of gotpairs and
# calculate the highest average among the director-writer pairs
episodes <- seq(2,73)
count <- 1
curdirector <- gotpairs$Director[1]
curwriter <- gotpairs$Writer[1]
currating <- gotpairs$IMDb_Rating[1]
curavg <- currating
highavg <- currating
lowavg <- currating
indexhigh <- 0
indexlow <- 0

for (val in episodes) {
  if (droplevels(curdirector) == droplevels(gotpairs$Director[val]) &&
      droplevels(curwriter) == droplevels(gotpairs$Writer[val])) {
    count <- count + 1
    currating <- currating + gotpairs$IMDb_Rating[val]
```

```
    curavg <- currating / count
  }
  else {
    if (highavg < curavg) {
      highavg <- curavg
      indexhigh <- val - 1
    }
    if (lowavg > curavg) {
      lowavg <- curavg
      indexlow <- val - 1
    }
    count <- 1
    curdirector <- gotpairs$Director[val]
    curwriter <- gotpairs$Writer[val]
    currating <- gotpairs$IMDb_Rating[val]
    curavg <- currating
  }
}

cat("Highest rating pair with rating ", highavg)
print(gotpairs$Director[indexhigh],max.levels = 0)
print(gotpairs$Writer[indexhigh],max.levels = 0)

cat("Lowest rating pair with rating ", lowavg)
print(gotpairs$Director[indexlow],max.levels = 0)
print(gotpairs$Writer[indexlow],max.levels = 0)
```

Highest pair with average rating 9.8
Director: Matth Shakman
Writer: David Benioff & D.B Weiss

Lowest pair with average rating 4.1
Director: David Benioff & D.B Weiss
Writer: David Benioff & D.B Weiss

d) Construct an 8×10 matrix of IMDb_Rating values where rows correspond to Season and columns correspond to Number_in_Season and element (i, j) corresponds to the IMDb rating of episode j within season i. If a particular season does not have a particular episode (e.g., season 7 does not have an episode 9), the corresponding matrix element should be NA. Label the rows S1, S2, . . . , S8 and label the columns Ep1, Ep2, . . . , Ep10. Print out this matrix

```
# season 7 does not have 8,9,10
# season 8 does not have 7,8,9,10
got <- read.csv("GOT.csv")
gotrate <- got["IMDb_Rating"]
gotratevec <- matrix(as.numeric(gotrate[1:73,]))

M <- append(gotratevec, c(NA,NA,NA), after=67)
M <- append(M, c(NA,NA,NA,NA), after=76)
M <- append(gotratevec, c(NA,NA,NA), after=67)
M <- append(M, c(NA,NA,NA,NA), after=76)
M <- matrix(M, nrow=8,ncol=10,byrow=TRUE,
            dimnames=list(c("S1","S2","S3","S4","S5","S6","S7","S8"),
```

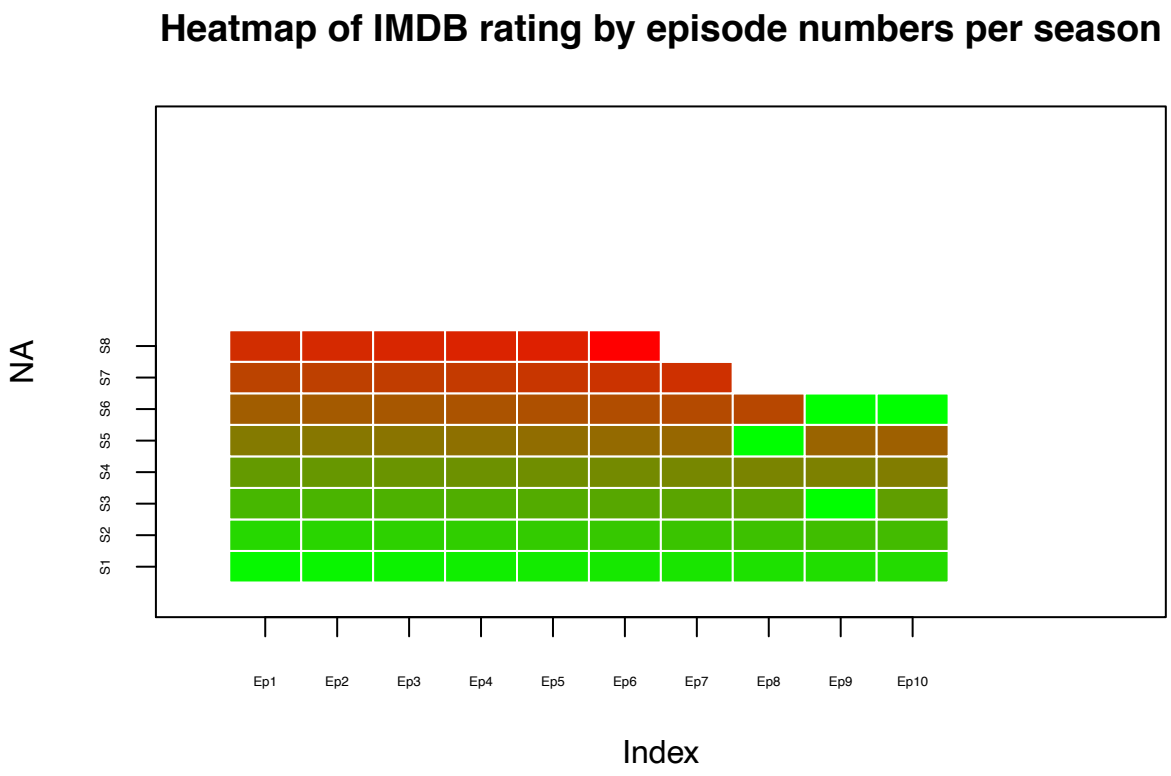
```
                          c("Ep1","Ep2","Ep3","Ep4","Ep5",
                            "Ep6","Ep7","Ep8","Ep9","Ep10")))
M
```

$$\begin{pmatrix} 9.0 & 8.8 & 8.7 & 8.8 & 9.1 & 9.2 & 9.3 & 9.1 & 9.6 & 9.5 \\ 8.9 & 8.6 & 8.9 & 8.9 & 8.9 & 9.1 & 9.0 & 8.9 & 9.7 & 9.4 \\ 8.9 & 8.7 & 8.9 & 9.6 & 9.0 & 8.9 & 8.8 & 9.1 & 9.9 & 9.2 \\ 9.1 & 9.7 & 8.9 & 8.9 & 8.8 & 9.7 & 9.2 & 9.7 & 9.6 & 9.7 \\ 8.6 & 8.6 & 8.6 & 8.8 & 8.7 & 8.1 & 9.1 & 9.9 & 9.5 & 9.1 \\ 8.6 & 9.5 & 8.8 & 9.2 & 9.7 & 8.5 & 8.7 & 8.4 & 9.9 & 9.9 \\ 8.7 & 9.0 & 9.3 & 9.8 & 9.0 & 9.2 & 9.6 & NA & NA & NA \\ 7.6 & 7.9 & 7.5 & 5.5 & 6.0 & 4.1 & NA & NA & NA & NA \end{pmatrix}$$

e) Pass the matrix you constructed in part (d) into the the make.heatmap() function you developed in Question 3, thereby producing a visualization of IMDb_Rating by Season and Number_in_Season. Add an informative title to this plot. Briefly discuss the insights drawn from this plot.

```
# season 7 does not have 8,9,10
# season 8 does not have 7,8,9,10
gotrate <- got["IMDb_Rating"]
gotratevec <- matrix(as.numeric(gotrate[1:73,]))

M <- append(gotratevec, c(NA,NA,NA), after=67)
M <- append(M, c(NA,NA,NA,NA), after=76)
M <- append(gotratevec, c(NA,NA,NA), after=67)
M <- append(M, c(NA,NA,NA,NA), after=76)
M <- matrix(M, nrow=8,ncol=10,byrow=TRUE,
            dimnames=list(c("S1","S2","S3","S4","S5","S6","S7","S8"),
                          c("Ep1","Ep2","Ep3","Ep4","Ep5",
                            "Ep6","Ep7","Ep8","Ep9","Ep10")))
make.heatmap(M)
title(main="Heatmap of IMDB rating by episode numbers per season")
```



There are only several high ratings achieved within the seasons

Season 2, 4 and 7 looks to have mediocre performance since both have not achieved a high rating.

Season 8 looked to have the worst performance since it has not achieved any high ratings and also contains an episode with the worst rating of the show

f) Using R, construct the following plot:

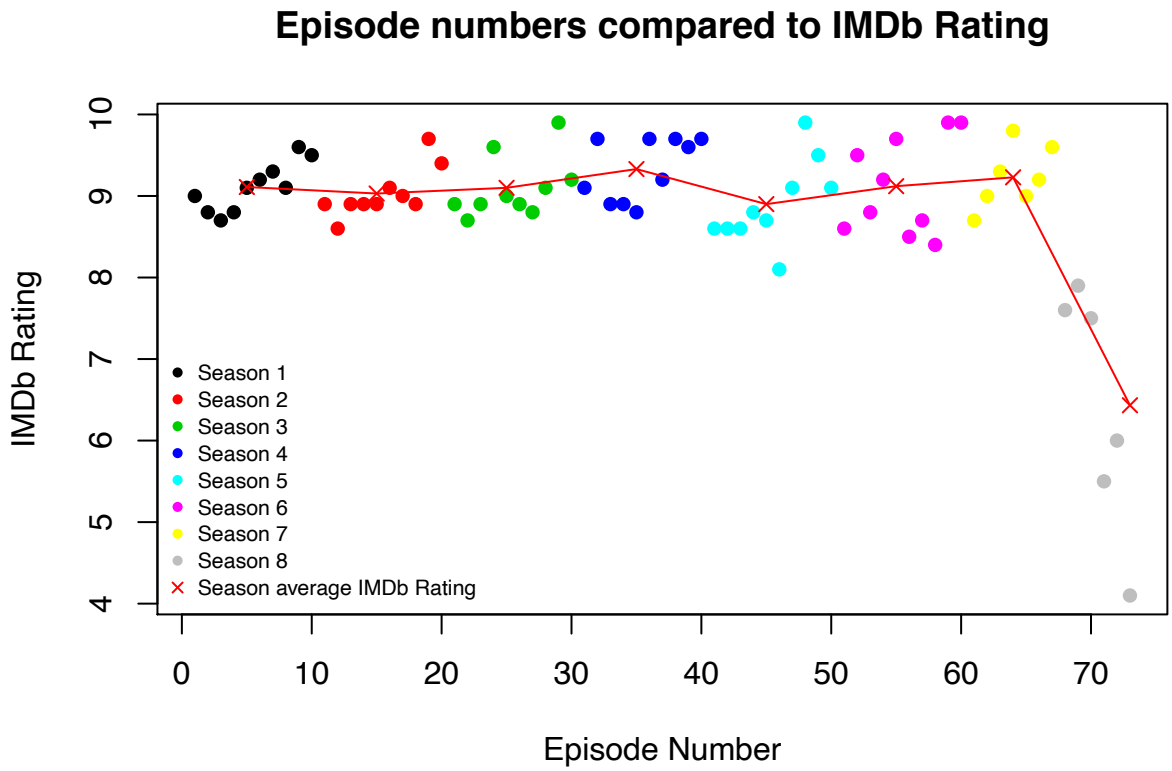
- Make a scatterplot of IMDb_Rating vs. Episode_Number with pch=16 and where the colour of the dots changes depending on the episode's season.
- Add to this plot eight red x's indicating the average episode rating in each season.
- Connect these eight red x's with red line segments.
- Add a legend, axis labels, and an informative title to the plot. Comment on any trends you observe in this plot

```
got <- read.csv("GOT.csv")
gotepisode <- got["Episode_Number"]
gotrate <- got["IMDb_Rating"]
episode <- as.numeric(gotepisode[1:73,])
imdb <- as.numeric(gotrate[1:73,])
colors = c(1,2,3,4,5,6,7,8)

# average ratings per season
avgs1 <- mean(gotrate[1:10,])
avgs2 <- mean(gotrate[11:20,])
avgs3 <- mean(gotrate[21:30,])
avgs4 <- mean(gotrate[31:40,])
```

```
avgs5 <- mean(gotrate[41:50,])
avgs6 <- mean(gotrate[51:60,])
avgs7 <- mean(gotrate[61:67,])
avgs8 <- mean(gotrate[68:73,])
avgx <- c(5,15,25,35,45,55,64,73)
avgr <- c(avgs1,avgs2,avgs3,avgs4,avgs5,avgs6,avgs7,avgs8)

# scatter plot
breaks = c(x1=10,x2=20,x3=30,x4=40,x5=50,x6=60,x7=67)
x.col = as.character(cut(episode,breaks=c(-Inf,breaks,Inf),labels=colors))
plot(episode,imdb,col=x.col, pch=16,
     xlab="Episode Number",
     ylab="IMDb Rating",
     main="Episode numbers compared to IMDb Rating")
points(avgx,avgr,pch=4,col=c(2,2,2,2,2,2,2))
lines(avgx,avgr,col=2)
legend("bottomleft", bty="n", c("Season 1", "Season 2", "Season 3",
                                "Season 4", "Season 5", "Season 6",
                                "Season 7", "Season 8", "Season average IMDb Rating"),
       col=c(1,2,3,4,5,6,7,8,2), pch=c(16,16,16,16,16,16,16,16,4),cex=0.7)
```



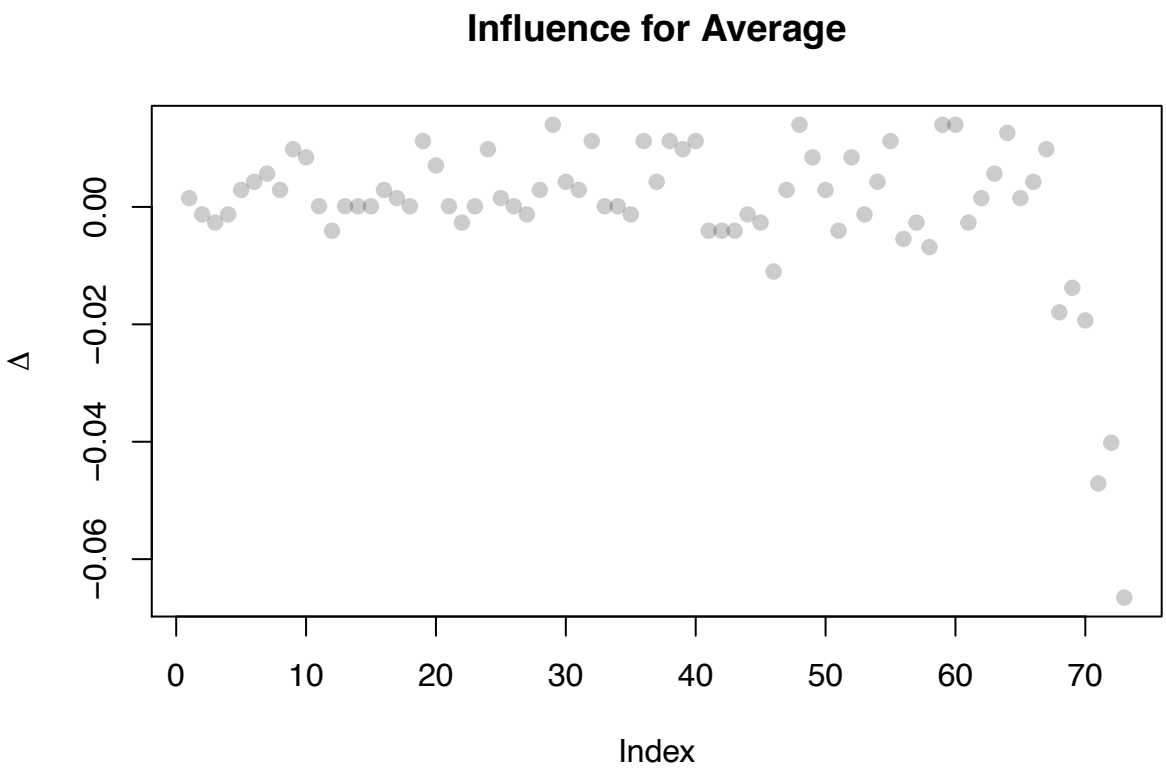
IMDB ratings were constantly high during seasons 1-7, then dropped significantly at season 8. Also it seems that ratings were lower at the beginning of each season compared to the episodes within the middle/end of the seasons on average, expect for season 8.

g) Construct an influence plot vs. observation number and identify the episode with the largest influence on the average IMDb_Rating attribute. Provide a rationale for why this particular episode is more influential than all of the others

```
got <- read.csv("GOT.csv")
gotrate <- got["IMDb_Rating"]
imdb <- as.numeric(gotrate[1:73,])
ybar <- mean(imdb)
delta = rep(0, length(imdb))
for (i in 1:length(imdb)) {
  delta[i] = ybar - mean(imdb[-i])
}

plot(delta, main="Influence for Average", pch=19,
     col=adjustcolor("black", alpha = 0.2), xlab = "Index",
     ylab = bquote(Delta))

# highest influence is episode 29, Rains of Castamere
which.max(delta)
```



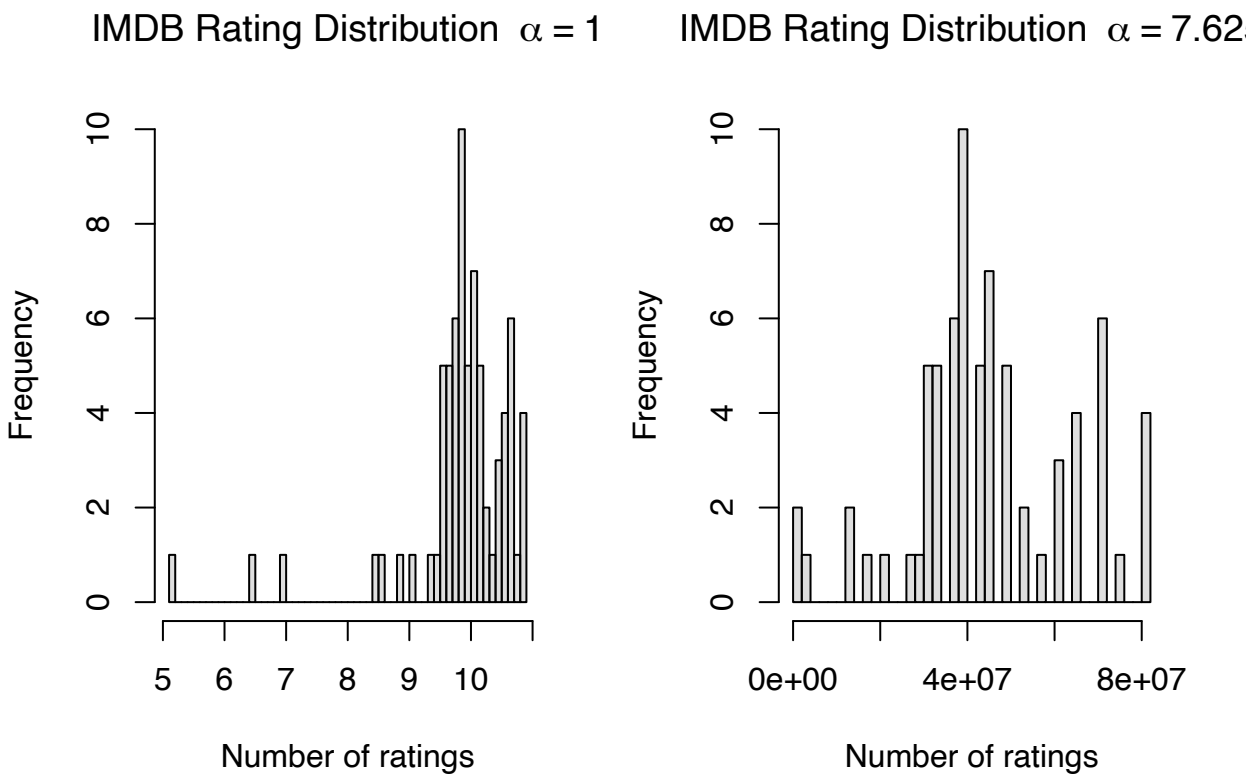
Rains of Castamere (episode 29) is the most influencial on the average IMDb rating attribute since it is the highest rated episode. Based on a media standpoint, this episode covered the Red Wedding which is the most infamous scene of the show that brought many new viewers to watch the show which does suggest episode 29 is the most influencial.

h) Using the powerfun() function from class, determine (and state) a power that makes the IMDb_Rating distribution more symmetric. Construct a 1 x 2 plot which contains histograms of the untransformed ratings and the transformed ratings using what you feel is the best value of alpha. Make sure to appropriately title and label your plots

```
got <- read.csv("GOT.csv")
gotrate <- got["IMDb_Rating"]
imdb <- as.numeric(gotrate[1:73,])

# powerfun function from lecture
powerfun <- function(x, alpha) {
  if(sum(x <= 0) > 0) stop("x must be positive")
  if (alpha == 0)
    log(x)
  else if (alpha > 0) {
    x^alpha
  }
  else -x^alpha
}

par(mfrow=c(1,2))
actual = c(1,7.625)
for (i in 1:2) {
  hist(powerfun(imdb + 1, actual[i]), col=adjustcolor("grey", alpha = 0.5),
       main=bquote(.("IMDB Rating Distribution ") ~ alpha ~ .("=") ~ .(actual[i])),
       xlab="Number of ratings", breaks=50)
}
```



i) Using the powerfun() function from class, determine (and state) powers that "straighten" the scatterplot of IMDb_Rating vs. US_Viewers. Construct a 1 x 2 plot which contains an untransformed IMDb_Rating vs. US_Viewers scatterplot and a transformed IMDb_Rating vs. US_Viewers scatterplot using what you feel are the best values of alpha_x and alpha_y. Make sure to appropriately title and label your plots

```
got <- read.csv("GOT.csv")
gotus <- got["US_Viewers"]
us <- as.numeric(gotus[1:73,])

par(mfrow=c(1,2))
a = rep(c(1,54),each=1)
b = rep(c(1,54),each=1)
for (i in 1:2) {
  plot(powerfun(imdb+1, a[i]), powerfun(us+1, b[i]), pch = 19, cex=0.5,
       col=adjustcolor("black", alpha = 0.3), xlab = "IMDB ratings",
       ylab = "US Viewers in millions",
       main=bquote(.("IMDB Rating vs US Viewers")
                   ~ alpha[x] ~ .("=") ~ .(a[i]) ~ .(",")
                   ~ alpha[y] ~ .("=") ~ .(b[i])))
}
```