Xiaobo Zhang
CS 57300
2/18/2015
Code

1. PCA

```
data<- read.table("yelp.dat",header=TRUE,sep=";",comment.char="",quote="")

x<-data[c(3:4,8:9,14:44)]
logx<-x;
logx['review_count'] <- log(logx['review_count'])

pca_old<-princomp(x)
screeplot(pca_old,type='line')
pca_old$loadings[,1]
summary(pca_old)



pca_new<-princomp(logx)
screeplot(pca_new,type='line')
pca_new$loadings[,1]
summary(pca_new)

sample_data<- x[sample(1:nrow(x),100,replace=FALSE),]
pca_sample_old<-princomp(sample_data)
screeplot(pca_sample_old,type='line')
pca_sample_old$loadings[,1]
summary(pca_sample_old)

sample_data_log<-logx[sample(1:nrow(logx),100,replace=FALSE),]
pca_sample_new<-princomp(sample_data_log)
screeplot(pca_sample_new,type='line')
pca_sample_new$loadings[,1]
summary(pca_sample_new)
```

2. Score and search

```
data<- read.table("yelp.dat",header=TRUE,sep=";",comment.char="",quote="")
newdata = data[c(4,42)]
pca = princomp(newdata)
mean_data = scale(newdata,center=T,scale=F)

seq1 = seq(-0.95,0.95,by = 0.05)
variance = c()

for (v1 in seq1){
   v2 = sqrt(1-v1^2)
   b1 = c(v1,v2)
   transformdata = as.matrix(mean_data) %*% as.matrix(b1)
   diff = max(transformdata)- min(transformdata)
   variance = c(variance,diff)
}

plot(seq1, variance)
```

3. Transformation and association

Part1 and 2

```
data<- read.table("yelp.dat",header=TRUE,sep=";",comment.char="",quote="'")
trim <- function (x) gsub("^\\s+|\\s+$", "", x)

#find first 30 category
category <-c()
for(item in unlist(data['categories'])){
   temp <- unlist(strsplit(item,",",fixed=T))
   for(i in temp){
      if(!is.element(i,category)){
         category <- c(category,i)
      }
   }
}
category <- category[1:30]
category <- trim(category)

#find top 30 city
city<-data['city']
city<-table(city)
city<-sort(city,decreasing = TRUE)
city<-city[1:30]
city<-names(city)
city<-trim(city)
#create binary city matrix
city_matrix <- matrix(nrow = length(data[,1]),ncol = length(city),dimnames = list(c(),city))
for (i in 1: length(city)){
   for (j in 1:length(data[,1])){
      if (city[i] == trim(as.character(unlist(data[j,]['city'])))){
         city_matrix[j,i]=1
      }else{
         city_matrix[j,i]=0
      }
   }
}
#create binary category matrix
category_matrix<- matrix(nrow = length(data[,1]),ncol = length(category),dimnames =
list(c(),category))
for (i in 1: length(category)){
   for (j in 1:length(data[,1])){
      if (regexpr(category[i],trim(as.character(unlist(data[j,]['categories'])))) > 0){
         category_matrix[j,i]=1
      }else{
```

```
            category_matrix[j,i]=0
        }
    }
}
Part3
chi_matrix = matrix(nrow =30,ncol = 30,dimnames    = list(city,category))
p_matrix = matrix(nrow =30,ncol = 30,dimnames    = list(city,category))
for (i in 1 :length(city)){
    for(j in 1 : length(category)){

        b_matrix = matrix(c(0,0,0,0),nrow =2)
        for (k in 1:length(data[,1])){
            if (city_matrix[k,i] == 0 & category_matrix[k,j] == 0){
                b_matrix[1] = b_matrix[1] + 1
            }
            if(city_matrix[k,i] == 0 & category_matrix[k,j] == 1){
                b_matrix[2] = b_matrix[2] + 1
            }

            if(city_matrix[k,i] == 1 & category_matrix[k,j] == 0){
                b_matrix[3] = b_matrix[3] + 1
            }
            if (city_matrix[k,i] == 1 & category_matrix[k,j] == 1){
                b_matrix[4] = b_matrix[4] + 1
            }
        }

        result = chisq.test(b_matrix)$statistic
        p = chisq.test(b_matrix)$p.value
        if(is.nan(result)){
            chi_matrix[i,j] =0
            p_matrix[i,j] = 0
        }else{
            chi_matrix[i,j] = result
            p_matrix[i,j] = p
        }

    }
}
top5 <- order(chi_matrix,decreasing=T)[1:5]

for (i in top5){
    t = which(chi_matrix==chi_matrix[i],arr.ind=T)
    x = t[1]
```

```
    y = t[2]

    cat("city: ",city[x],"\n")
    cat("category: ",category[y],"\n")
    cat("chi_square: ",chi_matrix[i],'\n')
    cat("p: ",p_matrix[i],'\n')
}

Part4
agood <- cbind(city_matrix[,3],category_matrix[,7])
amax <- cbind(city_matrix[,3],category_matrix[,1])

amax_score <- c()
agood_score <- c()

sequence <- c(16,32,64,128,256,1024,2048,4096,8192)

for (n in sequence){
    testsample <- sample(1:length(data[,1]),n)

    b_matrix = matrix(c(0,0,0,0),nrow =2)
    for (k in testsample){
        if (amax[k,1] == 0 & amax[k,2] == 0){
            b_matrix[1] = b_matrix[1] + 1
        }
        if(amax[k,1] == 0 & amax[k,2] == 1){
            b_matrix[2] = b_matrix[2] + 1
        }

        if(amax[k,1] == 1 & amax[k,2] == 0){
            b_matrix[3] = b_matrix[3] + 1
        }
        if (amax[k,1] == 1 & amax[k,2] == 1){
            b_matrix[4] = b_matrix[4] + 1
        }
    }
    result = chisq.test(b_matrix)$statistic
    if(is.nan(result)){
        amax_score <- c(amax_score,0)
    }else{
        amax_score <- c(amax_score,result)
    }

}
```

```
for (n in sequence){
   testsample <- sample(1:length(data[,1]),n)

   b_matrix = matrix(c(0,0,0,0),nrow =2)
   for (k in testsample){
      if (agood[k,1] == 0 & agood[k,2] == 0){
         b_matrix[1] = b_matrix[1] + 1
      }
      if(agood[k,1] == 0 & agood[k,2] == 1){
         b_matrix[2] = b_matrix[2] + 1
      }

      if(agood[k,1] == 1 & agood[k,2] == 0){
         b_matrix[3] = b_matrix[3] + 1
      }
      if (agood[k,1] == 1 & agood[k,2] == 1){
         b_matrix[4] = b_matrix[4] + 1
      }
   }
   result = chisq.test(b_matrix)$statistic
   if(is.nan(result)){
      agood_score <- c(agood_score,0)
   }else{
      agood_score <- c(agood_score,result)
   }

}

plot(sequence,amax_score,col='red',type='l',ylab = "chi square score",main='size vs chi_square score')
lines(sequence,agood_score,col='green',type='l')
legend("topleft", legend = c("good attribute",'best attribute'), col=c('red','green'), lty=c(1,1))
```

4. Hypothesis testing

```
newdata <- cbind(data['review_count'],data['stars'],city_matrix[,1],city_matrix[,2],category_matrix[,1],category_matrix[,4])
colnames(newdata)[3]<-'las vegas'
colnames(newdata)[4]<-'phoenix'
colnames(newdata)[5]<-'Indian'
colnames(newdata)[6]<-'Chinese'

las_score <- c()
phoenix_score <- c()

indian_score <-c()
chinese_score <- c()
for(i in 1:length(newdata[,1])){
   row=newdata[i,]
   if (row[3] == 1){

      las_score <- c(las_score,row[2])
   }
   if(row[4] == 1){
      phoenix_score <- c(phoenix_score,row[2])
   }
}

mean_las = mean(unlist(las_score))
mean_phoenix = mean(unlist(phoenix_score))

barplot(c(mean_las,mean_phoenix),names.arg=c("Las Vegas","Phoenix"),ylim=c(0,5),ylab = "score",main="Average restaurant score in Las Vegas and Phoenix")

indian_score <-c()
chinese_score <- c()
for(i in 1:length(newdata[,1])){
   row=newdata[i,]
   if (row[5] == 1){

      indian_score <- c(indian_score,row[1])
   }
   if(row[6] == 1){
      chinese_score <- c(chinese_score,row[1])
   }
}
```

```
mean_indian = mean(unlist(indian_score))
mean_chinese = mean(unlist(chinese_score))

barplot(c(mean_indian,mean_chinese),names.arg=c("Indian","chinese"),ylim = c(0,50),ylab = "review count",main="Average review count between indain and chinese food")
```