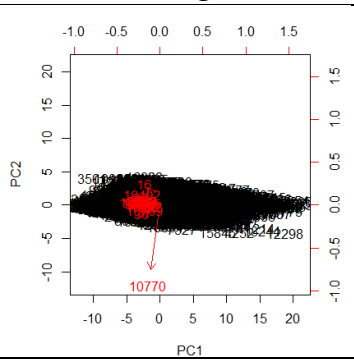
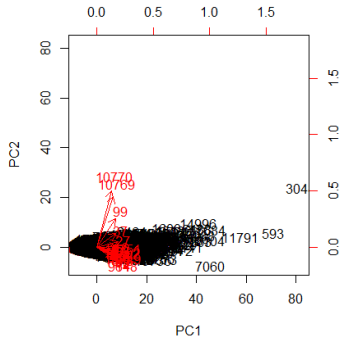
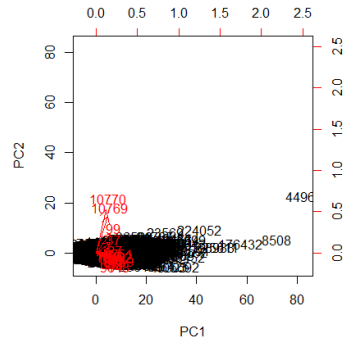
Movie Data Analysis

The purpose of this analysis was to elucidate patterns in genres of movies watched by users, which was specifically accomplished here by unsupervised cluster analysis.

Movie rating data frame and movie-genre mapping data frame was loaded for 265847 service users and 20 different genres. The two data frames were combined to produce a LONG data structure with a column of users, ratings, and genres. The first step was to construct a measure of a user's appeal for each genre. Pipeline functions in R on the combined data frame yielded two measures: the mean rating for each genre and the number of views of each genre.

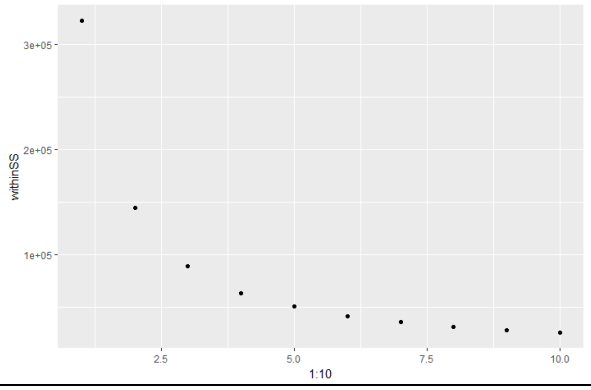
Figure 1: Principal Component Biplots for Different Genre Scoring Metrics

Mean Rating Metric	Count Metric	Combined Metric
		

Principal component analysis on both of these measures was used to identify salient principal component directions (Figure 1) and reduce problem dimensionality. Based on mean rating alone, the data varied mainly in a single principal component direction. To use count alone would ignore important information that could be contained in the rating values. Thus, a combined metric was constructed as the product of count and mean rating for each user for each genre. For the combined metric, the explained variation is captured well by the first principal component (86.2%) and the second principal component (5.04%).

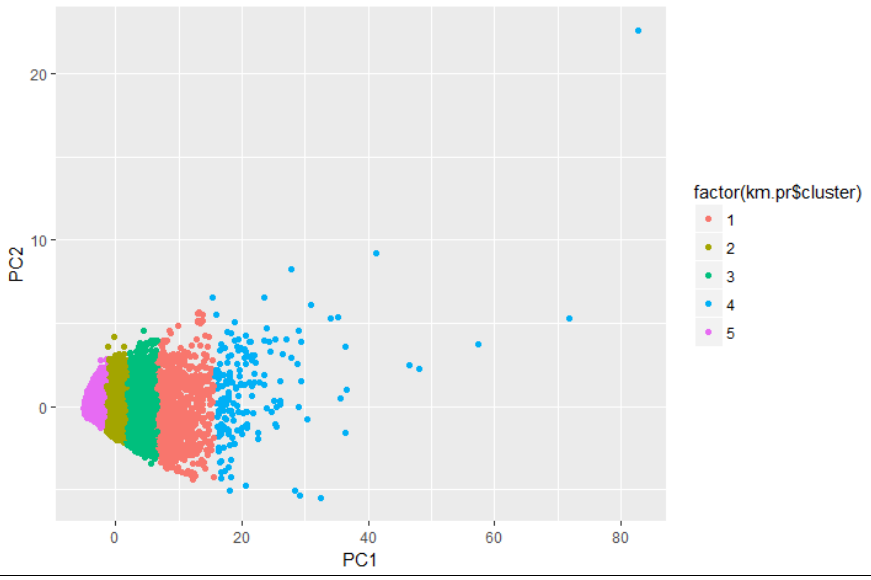
The second principal component direction loads more heavily on Documentary, Foreign, and TV Movie, whereas the first principal component loads more on a broad array of genres.

Figure 2: Cluster Size vs. Total Sum of Squares Variation Within



To understand patterns in individual users based on their principal component scores, a cluster analysis was performed using kmeans. To determine the number of groups to select, kmeans was run for cluster size one through ten, and the total sum of squares within groups (quantity to minimize) was plotted to determine a cutoff (Figure 2).

Figure 3: Kmeans Cluster Overlaid on PCA Biplot for Combined Metric



Cluster size of 5 was chosen as an appropriate cutoff, and the clusters were overlaid on the principal component biplot (Figure 3). To interpret, users with low score in the first principal direction component and high in the second, are more likely to watch TV documentaries on foreign affairs such as what might be found on the History Channel. Additionally, there are five groups of users lying along the first principal component. Users with larger principal component scores in the first direction tend to watch a broader set of movies compared to users with low principal component scores. For example, user 157155 has a first principal component score of

22.17 which means they enjoy a larger set of genres compared to user 53471 with a first principal component score of 2.84.

The clusters need to be explored further to determine which categories are enjoyed exactly by more selected users.

The analysis has two major flaws which influence the results. First, it is assumed that combinations of genres do not influence the rating given by a user to a movie. For each user, movie rating was copied uniformly to each genre associated with that movie. However, in reality the rating should be weighted to each genre associated with that movie. Second, the analysis omits users who have not seen every genre (NaN values in matrix). The fact that users have not seen every genre in itself contains pattern information which was not analyzed here.

Appendix:

```
library(tidyverse)
library(jsonlite)
library(dplyr)
library(ggplot2)

#Parse movies_metadata.csv into movie_genre.Rdata using abstract_movie_genre.R code.

#Load data
load('movie_genreID_genre.Rdata')
load('ratings.Rdata')
#ratings<-read_csv('movie_rating_data.csv')

#Combine ratings and movie_genre by movieId vector
combined<-inner_join(ratings,movie_genre)
#yields data.frame of rating of each user of each genre of each movie
attach(combined)

#Create vector of genre_id's k (covariates)
genres<-as.integer(levels(factor(combined$genre_id)))
k<-length(genres)

#Find total num of users n (observation)
n<-length(unique(combined$userId))

ratingsMeans<-combined %>% group_by(userId,genre_id)%>%
summarise(mean(rating))%>%spread(genre_id,'mean(rating)')
user.genre.rating1<-ratingsMeans[,-1]

viewsCount<-combined %>% group_by(userId,genre_id)%>%
summarise(n())%>%spread(genre_id,'n()')
user.genre.rating2<-viewsCount[,-1]

user.genre.rating<-ratingsMeans[,-1]*viewsCount[,-1]

pr<-prcomp(na.omit(user.genre.rating),scale=TRUE)
biplot(pr,scale=0)
pr.varExplained<-pr$sdev^2
pve<-pr.varExplained/sum(pr.varExplained)
data<-pr$x[,1:2]
```

```
#choosing number of clusters
withinSS<-NULL
for (i in 1:10){
  withinSS[i]<-kmeans(data,i,nstart=20)$tot.withinss
}
SSplot<-ggplot(as.data.frame(withinSS),aes(x=1:10,y=withinSS))+geom_point()
SSplot
km.pr<-kmeans(data,5,nstart=20)


#km<-kmeans(na.omit(user.genre.rating),10,nstart=20,algorithm='MacQueen')

clusterPlot<-
ggplot(as.data.frame(data),aes(x=PC1,y=PC2))+geom_point(aes(color=factor(km.pr$cluster)))
clusterPlot
```