hw4AnimePCA

Sai Varadharajan

5/6/2023

Files:

Box Folder: https://uwmadison.box.com/s/8oq354ju8hspa2rtql9l9ly1tzvq06r4

Dataset CSV File: https://uwmadison.box.com/shared/static/fxzjs879p27h2h3pqdmeht0f7w1foe4p.csv

Rmd File: https://uwmadison.box.com/shared/static/m538a0we3lefem96ezgf91fb2injvqu.rmd

Introduction

The dataset I will be performing PCA on to identify underlying patterns and relationships is a dataset on anime. It includes columns for names, genres, ratings, type, id, number of members in the fanbase, and number of episodes. I will be trying to identify clusters of similar anime shows based on their genre, type, and rating data, and try to reduce the dimensionality of the dataset to identify the most important features that contribute to user preferences for anime shows. Each of my two visualizations will answer an essential question (mentioned later) that contributes to this overall goal.

Preprocessing the data, Pros and Cons

In order to perform PCA on this dataset, I had to choose to make some modifications to the dataset. Firstly, I got rid of unnecessary columns like the Anime_id, and then I mutated the episodes column to make sure it was a numeric type. I also got rid of rows with missing values.

A decision I made was to mutate the Genre column to only include one genre per row. For example, an anime could have the Genre value "Mystery, Adventure,Sci-fi". I decided to edit the values so that it would only contain the first Genre, in this case "Mystery". I decided to do this to make the number of genres decrease from a few hundred to forty for simplicity's sake in my visualizations. I ran into a problem rendering the plots because there were so many genres that it was difficult to coherently visualize because the plot would try to make a specific color for each combination of genres, which ended up causing the plot not being able to be rendered by my computer at all.

A disadvantage of this decision could be that the resulting genres an anime is classified under might not be as accurate, and two different anime within the same genre could be very different, which would make it difficult to make out distinct clusters in my visualization.

```
anime_df <- read_csv("https://uwmadison.box.com/shared/static/fxzjs879p27h2h3pqdmeht0f7w1foe4p.csv")

anime_df <- anime_df %>%
  select(-anime_id,-type) %>%
  mutate(episodes = as.numeric(episodes)) %>%

  mutate(
    genre=gsub("(^\\w+).*", "\\1", genre)

  ) %>%
  na.omit()
```

Computing PCA

I made a PCA recipe, and updated the role of my Name and Genre columns to be IDs, and then created a PCA object. I then stored the components, scores, and variances separately for my visualizations. I ended up having 3 principal components.

```
pca_rec <- recipe(~., data = anime_df) %>%
  update_role(name, genre, new_role = "id") %>%
  step_normalize(all_predictors()) %>%
  step_pca(all_predictors())

pca_prep <- prep(pca_rec)
```

```
#values of components, scores, variances

components <- tidy(pca_prep,2)
scores <- juice(pca_prep)
variances <- tidy(pca_prep,2,type="variance")
```

Visualizations:

Visualization 1 - Patchwork PC

Essential Question (Answered in Key Findings Section)

What are the relationships between different anime genres based on their principal components scores?

Aspects of Design/Key Findings:

The patchwork consists of three scatterplots: PC3 vs PC2, PC3 vs PC1 and PC2 vs PC1, colored by genre. The axes are scaled based on the variance of the scores. The plot makes it possible to explore the relationships between different anime genres by visually identifying patterns and clusters of points in the plot.

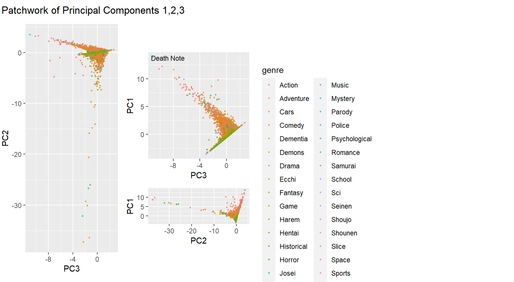
The scatterplot shows that certain genres like Drama, Romance, and Slice of Life tend to have positive PC1 scores while others like Shounen, Action, and Fantasy have negative PC1 scores, meaning that these anime have different underlying characteristics that affect their PC1 scores. Genres like Comedy, Sports, and Supernatural have positive PC3 scores, while others like Horror, Psychological, and Thriller have negative PC3 scores, which also suggest different underlying characteristics in their anime.

Some clear clusters of points are apparent in this plot, which could mean that certain genres are more closely related to each other than others. For example, there is some separation between the Shounen, Action, and Adventure genres (blue and green points) and the Romance, Drama, and Slice of Life genres (pink and purple points) along the PC1 axis. This suggests that these genres have distinct patterns of attributes that make them different from one another.

The outliers along the PC1 axis could represent anime shows that have unique combinations of attributes that do not fit neatly into any one genre. This could be a consequence of my decision to alter the Genre column

The yellow and orange points appear to be distributed more randomly across the plot and do not show clear separation from the other genres along either axis. Again, this could be a consequence of the less specific Genre column.

```
PC3vPC2 <- ggplot(scores, aes(PC3, PC2, label = name)) +
  geom_point(aes(color = genre), alpha = 0.7, size = 0.5,show.legend = FALSE) +
  geom_text_repel(check_overlap = TRUE, size = 3) +
  coord_fixed(sqrt(variances$value[2] / variances$value[1]))

PC3vPC1 <- ggplot(scores, aes(PC3, PC1, label = name)) +
  geom_point(aes(color = genre), alpha = 0.7, size = 0.5,show.legend = FALSE) +
  geom_text_repel(check_overlap = TRUE, size = 3) +
  coord_fixed(sqrt(variances$value[2] / variances$value[1]))

PC2vPC1 <- ggplot(scores, aes(PC2, PC1, label = name)) +
  geom_point(aes(color = genre), alpha = 0.7, size = 0.5) +
  geom_text_repel(check_overlap = TRUE, size = 3) +
  coord_fixed(sqrt(variances$value[2] / variances$value[1]))



PC3vPC2 + PC3vPC1 / PC2vPC1 +

  plot_annotation(title = "Patchwork of Principal Components 1,2,3")
```



Problems Encountered:

Some issues I ran into besides the Genre column was trying to facet the first visualization for different principal components. The code I tried would crash my computer, so I decided to use patchwork instead

```
# crashes computer when trying to facet

#ggplot() +
 # PC3vPC2 + facet_grid(PC1~PC2) +
  #PC3vPC1 + facet_grid(PC1~PC3) +
  #PC2vPC1 + facet_grid(PC2~PC3) +
  #ggtitle("Faceted Plot of Principal Components 1, 2, and 3") +
  #theme(axis.text = element_blank(),
   #     axis.title = element_blank())
```

Visualization 2

Essential Question:

How many principal components should be retained for further analysis and how much variance do they capture?

Answer:

As I will mention in the next section, since there is no extreme cut-off point and because there is only 30 percent captured by the components, I will retain all three components. This could be a possible consequence of the choice of narrowing down the number of different Genres.

Aspects of Design/Key Findings:

My first visualization is a patchwork of a scree plot and bar graph. The scree plot captures the positive and negative values explained by each component, while the bar graph displays the variance captured by each component on the y-axis and the component numbers on the x-axis. As we can see, the third component captures the most variance, at around 13%, while the second captures around 11% while the first captures around 10%, for a total of roughly 30 %. There doesn't seem to be an extreme cut-off point, so I will retain all three components. We can also see that the first component captures mostly positive values of the ratings and number of members, while the second captures mostly negative values for the number of episodes, while the third is a mix of both positive for the ratings and negative for the members.

```
#patchwork of variance and screeplot
scree_components <- ggplot(components, aes(value, terms, fill = value)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ component, scales = "free_y") +
  scale_y_reordered() +
  labs(y = NULL) +
  theme(axis.text = element_text(size = 7)) +
  labs(title="Scree Plot of Principal Components")

variance_components <- ggplot(variances %>% arrange(desc(value))) +
  geom_col(aes(component, value,fill=component)) +
  labs(title = "Variance Explained by Principal Components",
       x = "Principal Component",
       y = "Proportion of Variance Explained") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1))


scree_components / variance_components
```