Метод главных компонент

Алла Тамбовцева

Загрузка библиотек и данных

Загрузим данные из файла `genres_v2.csv` по различным музыкальным трекам с платформы Spotify, а также библиотеку `tidyverse`, она нам понадобится для удобного выбора столбцов по названию:

```
library(tidyverse)
music <- read.csv("genres_v2.csv")
```

Описание данных

Описание основных количественных переменных в файле:

- `danceability`: «танцевабельность», показатель от 0 до 1, индикатор того, насколько под трек удобно танцевать;
- `energy`: энергичность, показатель от 0 до 1, индикатор того, насколько трек интенсивный и активный с точки зрения восприятия;
- `loudness`: общая громкость трека, в децибелах;
- `speechiness`: обилие текста, показатель от 0 до 1;
- `acousticness`: акустичность, степень уверенности от 0 до 1, с которой можно считать трек акустическим;
- `instrumentalness`: инструментальность, показатель от 0 до 1, индикатор того, насколько трек инструментальный;
- `valence`: валентность, показатель от 0 до 1, индикатор того, насколько трек позитивный;
- `tempo`: темп трека в битах в минуту (BPM).

Более подробное описание можно найти здесь (оно лучше, чем на странице файла на Kaggle).

Отберём несколько количественных характеристик треков:

```
small <- music %>% dplyr::select(danceability, energy,
                                 loudness, speechiness, acousticness,
                                 instrumentalness, valence, tempo)
```

Метод главных компонент: реализация

Реализуем метод главных компонент с целью снизить размерность — перейти от выбранных 8 характеристик треков к более общим 3-4 характеристикам (сколько именно выбрать, решим позже), то есть некоторым интегральным индикаторам, отвечающим за различные качества треков.

Для этого воспользуемся функцией `prcomp()` (от *principal components analysis*) и не забудем центрировать и нормировать наши исходные данные с помощью соответствующих аргументов — показатели в датафрейме измерены в разных шкалах измерения:

```
pca <- prcomp(small, center = TRUE, scale = TRUE)
pca
```

```
## Standard deviations (1, .., p=8):
## [1] 1.5049497 1.3020588 1.1223244 0.8831993 0.8567074 0.7717853 0.6797895
## [8] 0.4565109
##
## Rotation (n x k) = (8 x 8):
##                            PC1         PC2         PC3        PC4         PC5
## danceability     -0.34850043  0.06622294 -0.60970302  0.1971300  0.16489826
## energy            0.54272745 -0.26236076 -0.18939783 -0.0394195 -0.05692188
## loudness          0.33115665 -0.55133513 -0.05804468 -0.2415440 -0.17296927
## speechiness      -0.31227254 -0.39388897 -0.02329042 -0.2020204  0.69107724
## acousticness     -0.43980271  0.07950023  0.32048639 -0.2011501 -0.47095185
## instrumentalness  0.34525380  0.45390797 -0.16865366  0.3636612  0.11836360
## valence          -0.24770219 -0.33592611 -0.50592894  0.2630921 -0.47150644
## tempo            -0.05322133 -0.37744314  0.44873086  0.7864489  0.06275693
##                           PC6         PC7         PC8
## danceability     -0.2515680  0.57377580 -0.20944545
## energy            0.3278728  0.07883363 -0.69442990
## loudness         -0.1161111  0.47084956  0.50946314
## speechiness       0.4668772 -0.06461012  0.07903218
## acousticness      0.4813914  0.43049372 -0.13498977
## instrumentalness  0.5488154  0.20443411  0.39632340
## valence           0.2495651 -0.43507871  0.16346612
## tempo            -0.0503791  0.14926886 -0.07800823
```

Что нам показывает выдача `pca`? Во-первых, стандартные отклонения главных компонент (квадратные корни из собственных значений ковариационной матрицы), которые отвечают за их информативность. Как мы и обсуждали, сначала идут наиболее информативные компоненты, с наибольшими дисперсиями (и наибольшими стандартными отклонениями), далее — менее информативные.

Во-вторых, выдача содержит матрицу поворота (`Rotation`), которая отвечает за поворот исходной системы координат. Коэффициенты в столбцах `PC1`-`PC8` показывают, с какими весами необходимо взять исходные показатели, чтобы получить их линейные комбинации, то есть главные компоненты.

Рассмотрим первую главную компоненту `PC1`. Она формируется следующим образом:

$$\text{PC}_1 = -0.35 \times \text{danceability} + 0.54 \times \text{energy} + 0.33 \times \text{loudness} - 0.31 \times \text{speechiness} - 0.44 \times \text{acousticness} +$$

$$+0.35 \times \text{instrumentalness} - 0.25 \times \text{valence} - 0.05 \times \text{tempo}.$$

Давайте подумаем, как проинтерпретировать полученный индикатор. В него с наибольшими положительными весами входят энергичность, громкость и инструментальность. Назовём этот индикатор показателем «эмоциональной заряженности» трека. По аналогии попробуем проинтерпретировать вторую главную компоненту `PC2`. В неё с положительными весами входят инструментальность, акустичность и «танцевабельность», те характеристики, которые отвечают за технические качества звука. Назовём этот индикатор показателем качества звука. Начиная с третьей главной компоненты, интерпретируемость немного снижается, уже сложнее понять, чем эта главная компонента отличается от предыдущей (но при желании можно подумать).

Соответственно, если мы захотим узнать значение нашего нового индекса «эмоциональной заряженности» для первого трека, нам нужно будет подставить в выражение для `PC1` центрированно-нормированные значения исходных показателей `danceability`, `energy`, `loudness` и др. для этого трека. К счастью, R умеет выполнять подобные операции самостоятельно, причём сразу для всех наблюдений в датафрейме, далее мы в этом убедимся.

Метод главных компонент: выбор количества главных компонент

Итак, мы поняли логику, которая используется при построении и интерпретации интегральных индикаторов, теперь давайте поговорим о том, сколько главных компонент нужно выбирать. Критерии следующие:

- выбрать столько ГК, сколько можем содержательно проинтерпретировать (особенно актуально при построении новых содержательных интегральных индексов);
- выбрать столько ГК, чтобы они объясняли не менее 75-80% дисперсии исходных данных;
- правило Кайзера: выбрать столько ГК, сколько собственных значений больше 1 (возводим стандартные отклонения компонент в квадрат и смотрим);
- правило Кэттела: выбрать столько ГК, сколько наблюдается до излома на графике «каменистой осыпи» (визуальный способ).

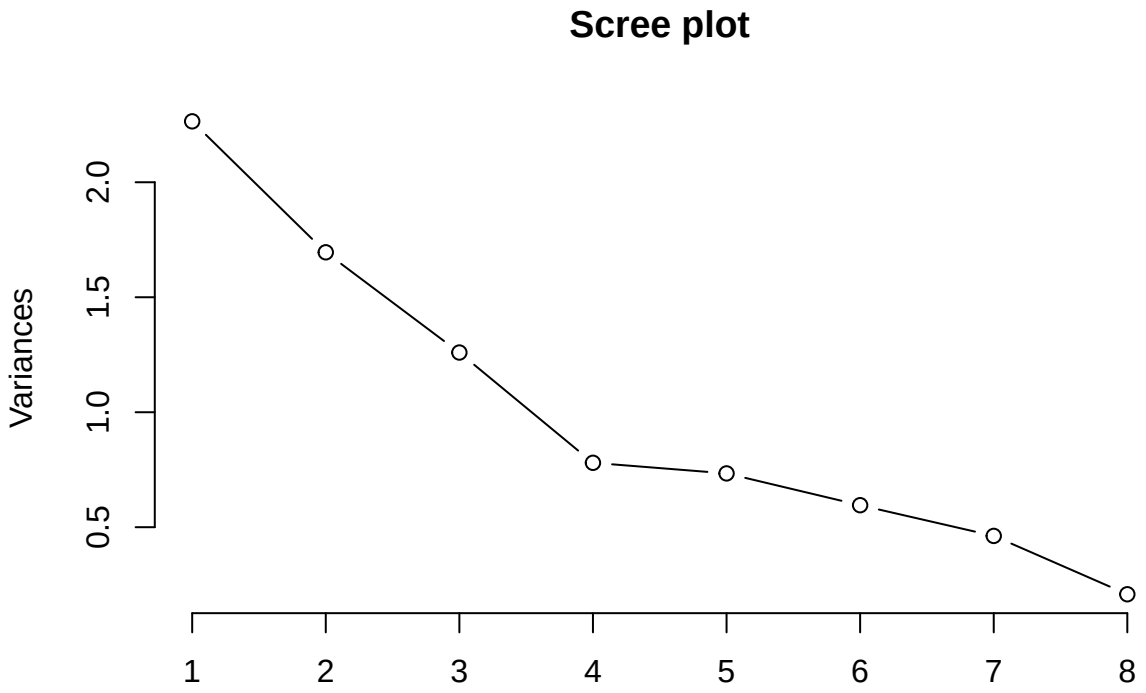
Содержательно мы смогли пояснить только две главные компоненты, перейдём к другим более формальным критериям. Стандартных отклонений главных компонент больше 1 у нас три, следовательно, собственных значений больше 1 у нас тоже три. По правилу Кайзера, нам следует извлечь три главных компоненты, если мы хотим «схлопнуть» наше исходное восьмимерное пространство (8 показателей) в пространство меньшей размерности. Посмотрим на доли дисперсии исходных данных, которые объясняют главные компоненты:

```
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.5049 1.3021 1.1223 0.88320 0.85671 0.77179 0.67979
## Proportion of Variance 0.2831 0.2119 0.1575 0.09751 0.09174 0.07446 0.05776
## Cumulative Proportion  0.2831 0.4950 0.6525 0.74999 0.84173 0.91619 0.97395
##                           PC8
## Standard deviation     0.45651
## Proportion of Variance 0.02605
## Cumulative Proportion  1.00000
```

Обратим внимание на накопленные доли объяснённой дисперсии (`Cumulative Proportion`). Согласно этому критерию, стоит выбрать четыре главных компоненты, так как вместе они объясняют примерно 75% дисперсии исходных данных. Осталось задействовать графический способ для выбора количества главных компонент. Построим график каменистой осыпи:

```
# type = "l" for both dots and lines
plot(pca, type = "l", main = "Scree plot")
```



Судя по графику, излом наблюдается при количестве главных компонент, равном четырём. Соответственно, увеличение числа главных компонент до пяти и выше не приведёт к существенному увеличению общей объяснённой дисперсии исходного массива данных.

Итак, выбор за нами: взвешиваем полученные результаты и решаем, какое количество главных компонент выбрать. Это количество зависит от задачи. Если мы хотели построить единый индикатор качества музыки, мы просто возьмём одну первую главную компоненту как наиболее информативную. Если мы хотели «схлопнуть» пространство, пожалуй, стоит найти компромисс между методами Кайзера, Кэттела и оценкой доли объяснённой дисперсии и выбрать, например, четыре главных компоненты. Тогда мы в итоге получим четыре индикатора качества музыки вместо исходных восьми, задача снижения размерности выполнена!

Метод главных компонент: вычисление значений компонент

Напоследок вычислим значения каждой главной компоненты для каждого трека в датафрейме с использованием исходного массива данных (вспомните запись для `PC1` выше и представьте, что такую операцию мы выполняем для всех компонент с соответствующими весами):

```
small2 <- predict(pca, newdata = small)
```

NB: функция `predict()` в данном случае не выполняет никакого предсказания в терминах статистических моделей, она просто использует веса из матрицы `Rotation` и вычисляет значения главных компонент (линейных комбинаций) для конкретных наблюдений.

Посмотрим, что получилось — запросим первые три строки новой таблицы `small2`:

```
head(small2, 3)
```

```
##             PC1       PC2        PC3         PC4        PC5        PC6
## [1,] -1.2896934 -1.252409 -0.6694496 -0.02185065  1.7658933  0.3808046
## [2,] -1.6460213  1.384039  0.5900536 -1.65561867 -0.6455453 -0.5308879
## [3,]  0.4185207 -1.224618  1.0183646  2.00624451  0.6410871 -1.5872771
##            PC7        PC8
## [1,] 0.1948755 -0.6971055
```

```
## [2,] 0.9273028  0.1425671
## [3,] 1.8064965 -1.2209551
```

Получили координаты первых трёх наблюдений в новой системе координат или, что и хотели, значения новых интегральных индексов для первых трёх треков.

Проверим, что корреляция между главными компонентами, действительно, равна 0:

```
cor(small2)
```

```
##               PC1           PC2           PC3           PC4           PC5
## PC1  1.000000e+00  5.229284e-16 -3.722319e-15  2.899766e-15  5.440201e-15
## PC2  5.229284e-16  1.000000e+00  1.266879e-15 -5.749794e-16  1.283088e-15
## PC3 -3.722319e-15  1.266879e-15  1.000000e+00  9.939825e-15  1.012259e-14
## PC4  2.899766e-15 -5.749794e-16  9.939825e-15  1.000000e+00 -1.531434e-15
## PC5  5.440201e-15  1.283088e-15  1.012259e-14 -1.531434e-15  1.000000e+00
## PC6 -4.420793e-15  5.648589e-15 -1.702919e-14  8.229745e-15  3.648314e-15
## PC7  2.810437e-14 -6.134770e-15  3.347198e-14 -8.228258e-15 -8.273002e-15
## PC8 -1.798897e-14  9.261441e-15 -1.599618e-14  6.535501e-15  7.259539e-15
##               PC6           PC7           PC8
## PC1 -4.420793e-15  2.810437e-14 -1.798897e-14
## PC2  5.648589e-15 -6.134770e-15  9.261441e-15
## PC3 -1.702919e-14  3.347198e-14 -1.599618e-14
## PC4  8.229745e-15 -8.228258e-15  6.535501e-15
## PC5  3.648314e-15 -8.273002e-15  7.259539e-15
## PC6  1.000000e+00  1.944634e-14 -5.313909e-15
## PC7  1.944634e-14  1.000000e+00  1.714814e-14
## PC8 -5.313909e-15  1.714814e-14  1.000000e+00
```

Всё так, не во всех случаях строго 0, но это нормально, ведь здесь у нас не теоретический коэффициент Пирсона, а выборочный!

Наконец, склеим старый датафрейм с исходными данными и новый датафрейм с главными компонентами, чтобы всё было вместе для удобства:

```
with_pc <- cbind(small, small2[, 1:2])
head(with_pc, 3)
```

```
##   danceability energy loudness speechiness acousticness instrumentalness
## 1        0.831  0.814   -7.364      0.4200       0.0598         1.34e-02
## 2        0.719  0.493   -7.230      0.0794       0.4010         0.00e+00
## 3        0.850  0.893   -4.783      0.0623       0.0138         4.14e-06
##   valence   tempo        PC1       PC2
## 1  0.3890 156.985 -1.2896934 -1.252409
## 2  0.1240 115.080 -1.6460213  1.384039
## 3  0.0391 218.050  0.4185207 -1.224618
```

Отлично!