STATS/CSE 780
Homework Assignment 3

Konrad Swierczek - 001423065

3/26/23

Introduction

Music contains many patterns and statistical regularities, some of which are abstract yet salient to the average encultured listener. Indeed, the remarkable human capacity for implicit knowledge and expertise of structure in music provokes an explanation. Numerous mechanistic models (Lerdahl et al. 2001; Krumhansl 2001; Woolhouse 2009) of heirarchical relations in musical pitch have been proposed based on empirical evidence from human behaviour experiments. However, the unique features involved in the perception of pitch of western music may provide an opportunity to use unsupervised statistical learning to approximate these abstract patterns. Pearce et al. (2010) applied unsupervised statistical learning to modelling human learning and development of expectation in western music. This study uses Pitch Class Distibutions (PCD) with clustering and dimensionality reduction to explore the heirarchical organization of pitch in the music of J.S. Bach. Pitch Class Distributions are 12 dimensional vectors that describe the concentration of each pitch-class (unique pitch or note) in the 12-tone equal temperment system (the pitch tuning system which is dominant in western music for at least the past hundred years) (Duffin et al. 2007). PCDs have previously been used to understand heriarchical organization of pitch (Krumhansl 2001; Lieck and Rohrmeier 2020; Lieck, Moss, and Rohrmeier 2020) and seem to account for much of the information the brain uses to compute these heirachies. PCDs are suitable for unsupervised learning techniques as they are high dimensional (ideal for dimensionality reduction) and are relatively abstract to interpret on their own. However, they are also generally thought to belong to discrete groups (Krumhansl 2001) and therefore ideal for clustering and dimensionality reduction to improve interpretability. [1]

Methods

A collection of 228 Musical Instrument Digital Interface (MIDI) files representing the works of J.S. Bach were accessed from the "Bach Central" database (bachcentral.com). These files were processed through a bespoke algorithm for extracting PCDs(Van Rossum and Drake 2009; Cuthbert and Ariza 2010).[2] 6 files were removed due to failure to complete the PCD (missing values). Three unsupervised statisical learning methods were applied to this dataset: heirarchical clustering, k-

[1] All materials and reproducible code used are available at https://github.com/konradswierczek/STATS780
[2] see github repository for details

means clustering, and heirachrical clustering using principal component analysis. Scaling of the features was not performed since these PCDs were already normalized to range from [0-1]. Average linkage was used for both heirarchical clusterings since the data cotains outliers that may influence a single or complete linkage approach (see Figure 5). Rand index analysis was performed between each of the models to determine compabitibility of the clusterings. The value of k=12 for al clusterings were determined using silhouette analysis as seen in Figure 6, Figure 8, and Figure 9. This value also corresponds with the amount of "keys" or tonal heirachies in this system. Two principal components were used to maximize visual interpreatability: since the first two components account for over 70% of the variance, additional components were not deemed necessary.

Results

The results of K-means clustering are summarized in Figure 3. Since key labels were unavailable for this dataset, comparisons were not possible. Figure 6 shows a peak average silhouette value at 12 clusters, which corresponds to the amount of pitches in the tuning system, or heirachires available. Figure 7 indicates a good fit with k=12, with few values below zero and all clusters partially above the average. Since the dataset has not been selected with a balanced amount of observations from each pitch heirarchy or key, smaller clusters may lack sufficient sample size. However, performance in silhouette plots worsens for both heirarchical clusterings. Although clusters remain generally above the average, increasing amounts of observations dip below zero. A more balanced and larger sample may be necessary to accuratley represent all clusters. Figure 10 and Figure 11 are dendrograms of the heirarchical clusterings. Due to the relatively large sample, interpreting these at the lowest level is difficult. Principle component analysis revealed only a few components are necessary to account for the majority of the variance in the data (Figure 12). In Figure 1, clusters represented in two-dimensional space conform to a circular shape, which may approximate previous theories such as the circle of fifths and the Tonnetz (Lieck, Moss, and Rohrmeier (2020), Lerdahl et al. (2001)). Finally, Figure 2 show Rand scores between the three clustering methods. The three clustering methods are highly compatible with each other, likely due to similar cluster sizes.

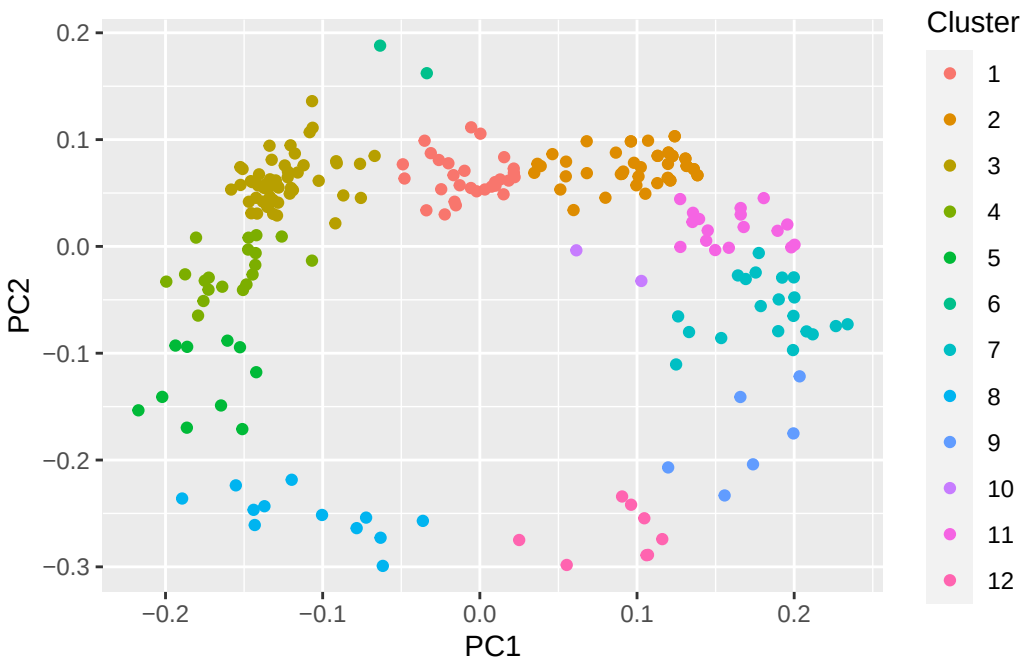

Figure 1: Results of principal component anaysis and heirachrical clustering. Each point represents a piece of music (observation). Axes correspond to principal components. Colour of point corresponds to computed cluster of the observation.

Conclusions

Unsupervised statisitcal learning techniques such as clustering and principle component analysis may prove useful for better understanding musical heirarchy and tonality, as well as making PCDs more interpretable and comparable. While this study shows the potential of PCDs for this application, datasets with key labels would reveal if the clusters formed in here indeed correspond with the human percept of key. If so, future work may investigate what patterns are responsbile for this clustering. These methods show that unsupervised learning can be suitable for determining not only the tonal pitch groupings of music, but also to determine the size of the tonal space.

Model Comparison	Rand Index
k-means - Heirarchical Clustering	0.9250596
k-means - Principal Components	0.9275239
Heirarchical Clustering - Principal Components	0.8727023

Figure 2: Rand score comparisons between the three clustering methods.

References

Cuthbert, Michael Scott, and Christopher Ariza. 2010. "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data."

Duffin, Ross W et al. 2007. *How Equal Temperament Ruined Harmony (and Why You Should Care)*. WW Norton & Company.

Krumhansl, Carol L. 2001. *Cognitive Foundations of Musical Pitch*. Vol. 17. Oxford University Press.

Lerdahl, Fred et al. 2001. *Tonal Pitch Space*. Oxford University Press, USA.

Lieck, Robert, Fabian C Moss, and Martin Rohrmeier. 2020. "The Tonal Diffusion Model." *Transactions of the International Society for Music Information Retrieval* 3 (1).

Lieck, Robert, and Martin Alois Rohrmeier. 2020. "Modelling Hierarchical Key Structure with Pitch Scapes." In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 811–18. CONF.

Pearce, Marcus T, María Herrojo Ruiz, Selina Kapasi, Geraint A Wiggins, and Joydeep Bhattacharya. 2010. "Unsupervised Statistical Learning Underpins Computational, Behavioural, and Neural Manifestations of Musical Expectation." *NeuroImage* 50 (1): 302–13.

Van Rossum, Guido, and Fred L. Drake. 2009. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Woolhouse, Matthew. 2009. "Modelling Tonal Attraction Between Adjacent Musical Elements." *Journal of New Music Research* 38 (4): 357–79.

Supplementary Materials

Cluster	Count
1	20
2	11
3	11
4	33
5	13
6	19
7	12
8	28
9	11
10	14
11	1
12	50

Figure 3: Count of observation in each cluster for k-means clustering.

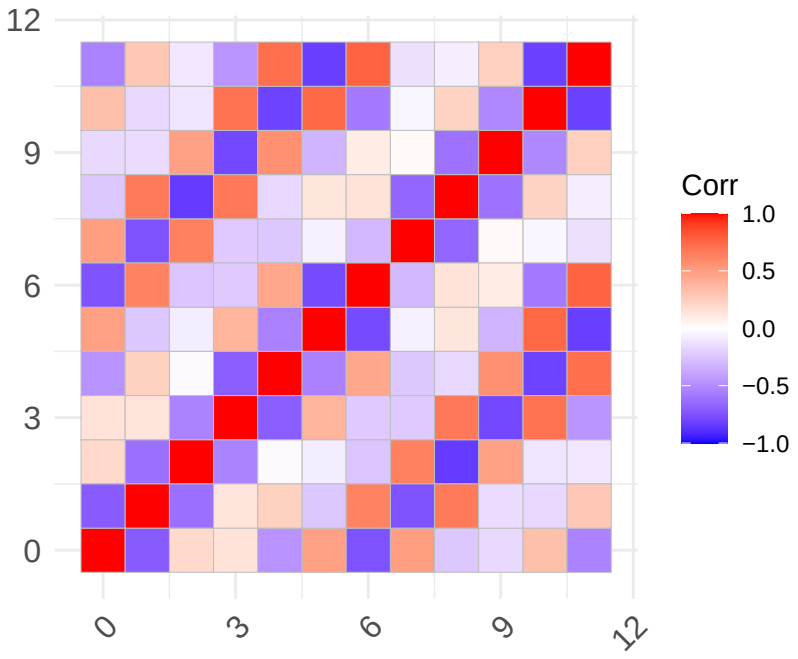

Figure 4: Correlation matrix of pitch-class distibutions

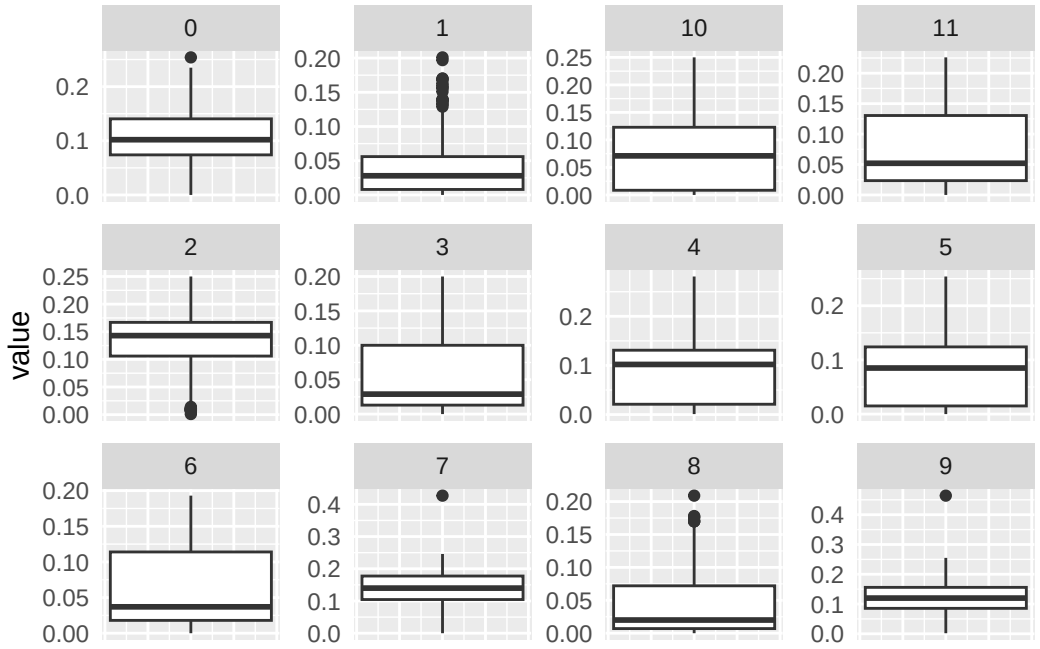

Figure 5: Boxplots for each pitch-class feature. Each plot corresponds to a pitch class, between 0 and 11.

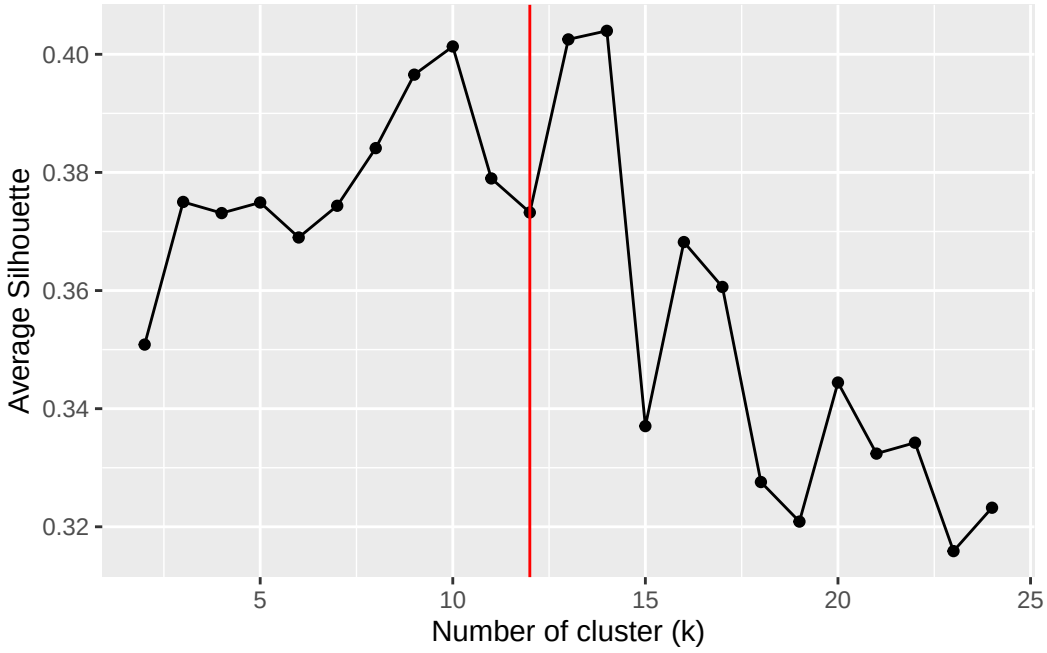

Figure 6: Average silhouette value for cluster size in k-means. k=12 was selected for clustering. NOTE: This plot is rendering differently in PDF than in R.

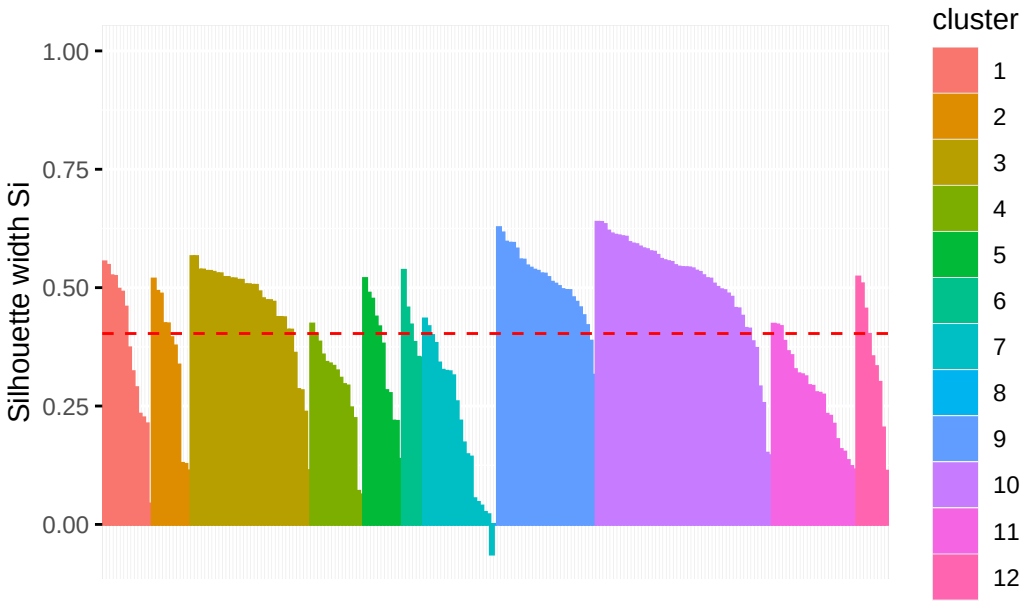

Figure 7: Silhouette plot for k-means clustering, k=12. Red line indicates average value.

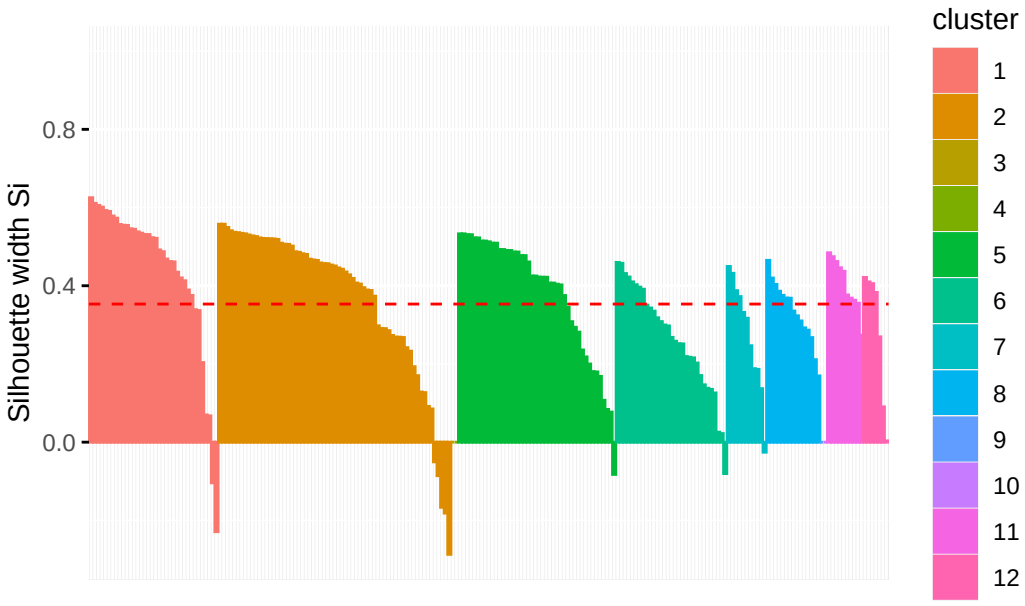

Figure 8: Silhouette plot for average linkage heirarchical clustering, k=12s. Red line indicates average value.

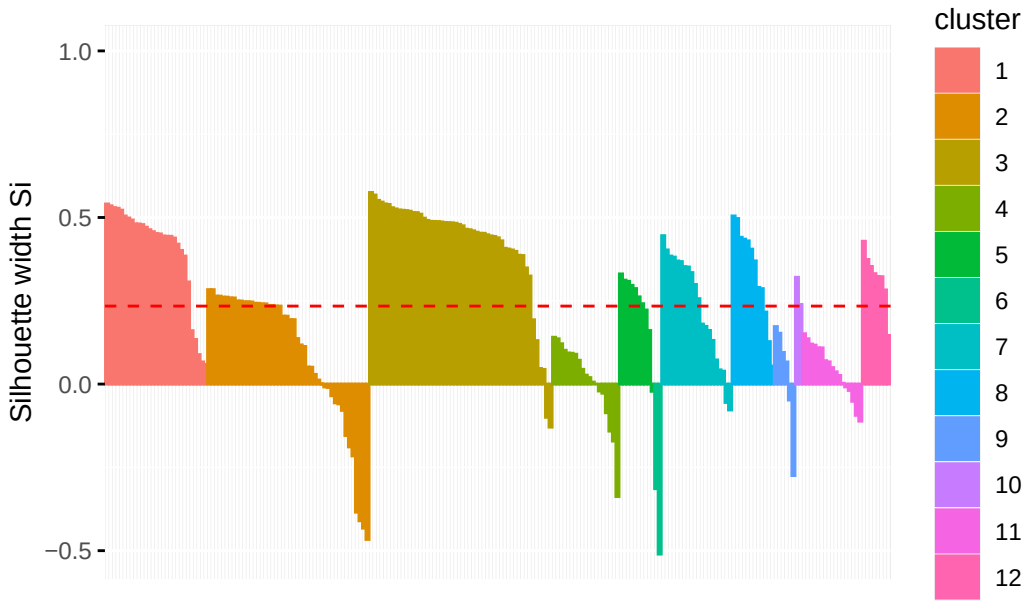

Figure 9: Silhouette plot for average linkage heirarchical clustering, k=12 on principle components. Red line indicates average value.

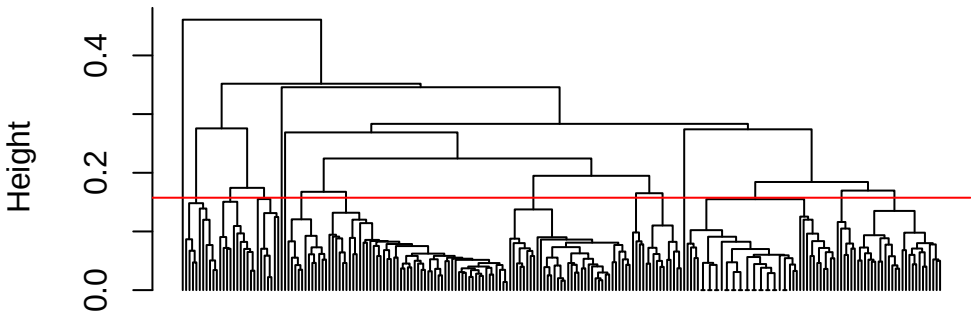

Figure 10: Dendrogram of heirarchical clustering. Labels have been removed to maintain legibility. Red line indicates cutting point for clustering

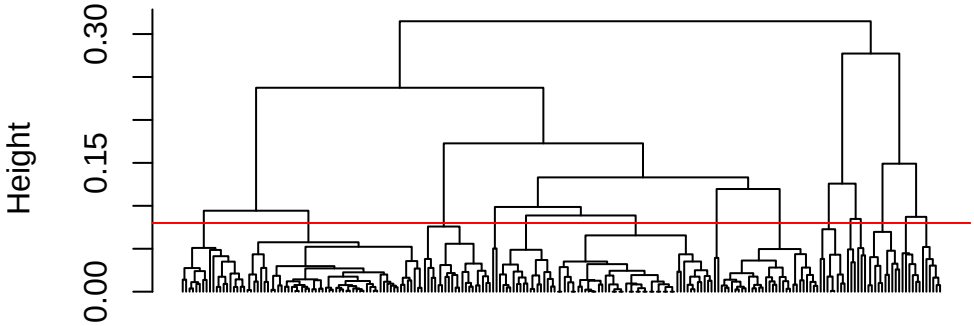

Figure 11: Dendrogram of heirarchical clustering pf principal components. Labels have been removed to maintain legibility. Red line indicates cutting point for clustering

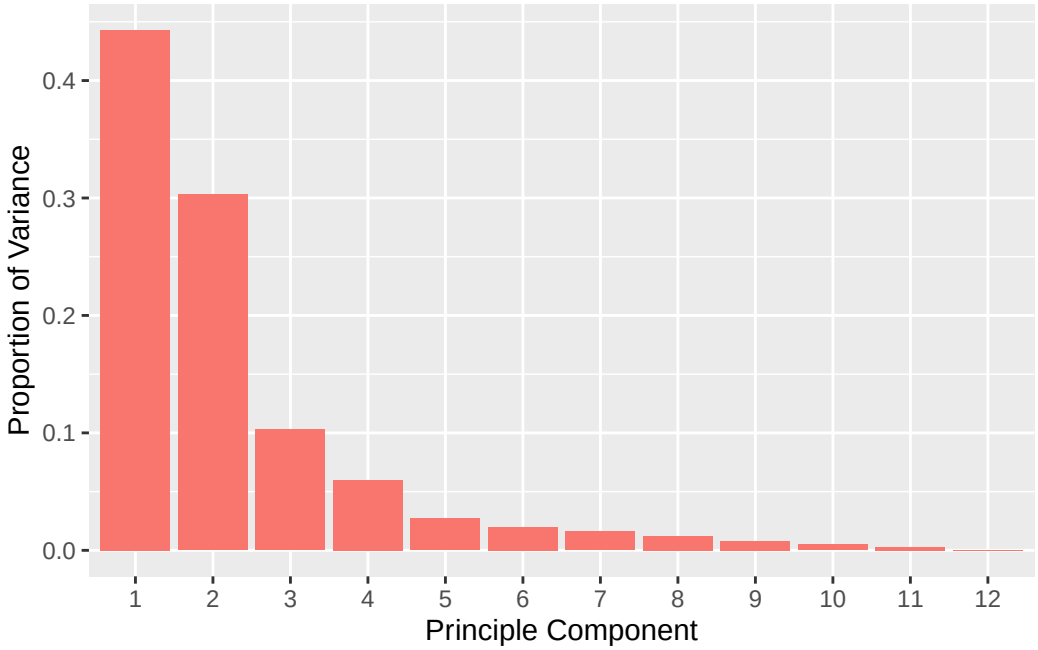

Figure 12: Proportion of variance explained by each principle component. The first two components are used in subsequent analyses and account for ~75% of the variance in the data.

::: {.cell}

```r
# knitr setup
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)
knitr::opts_chunk$set(fig.pos = "H", out.extra = "")
################################################################################
# Imports
packages <- c("tidyverse", "ggcorrplot", "cluster", "factoextra", "reticulate",
              "ggfortify", "knitr")
lapply(packages, library, character.only = TRUE)
################################################################################
set.seed(75)
# Pull data
# Retrieved from bachcentral.com, a source for MIDI files.
temp <- paste(tempfile(), ".zip", sep = "")
options(timeout = 60 * 10)
download.file("https://www.bachcentral.com/bach.zip", temp)
unzip(temp, exdir = "assignment3/data/midi")
# Performing PCD extraction on dataset and generating csv for analysis in R.
# See pcd_extract.py for details.
from pcd_extract import *
pcd_dataframe('assignment3/data/midi/bach',
              outpath = "assignment3/data/data.csv")
# Import Data
data <- read_csv("data/data.csv") %>%
# Remove NA values using first column.
drop_na("0") %>%
# Renaming index column.
rename(`piece` = `...1`) %>%
# Changing filepath to a readable format: just the filename without extension.
separate(piece, c('a', 'b', 'c', 'd', 'e', 'piece')) %>%
select(6:18)
# k-means clustering
```

```r
km.out <- kmeans(data[, 2:13], 12, nstart = 20)
km.clusters <- km.out$cluster
# Heirarchical clustering
data.dist <- dist(data[, 2:13])
hc1.clusters <- cutree(hclust(data.dist, method = "average"), 12)
# Principle component analysis
pca <- prcomp(data[, 2:13])
# Heirarchical clustering on first two principal components.
hc.out <- hclust(dist(pca$x[, 1:2]), method = "average")
pcahc_clusters <- cutree(hc.out, 12)
# Plotting principal components with clusters as colours.
as_tibble(pca$x[, 1:2]) %>%
  add_column(`cluster` = pcahc_clusters) %>%
    ggplot(aes(x = PC1, y = PC2)) +
      geom_point(aes(colour = as.factor(cluster))) +
      labs(colour = "Cluster")
kable(
tibble(`Model Comparison` = c("k-means - Heirarchical Clustering",
                              "k-means - Principal Components",
                              "Heirarchical Clustering - Principal Components"),
       `Rand Index` = c(fossil::rand.index(km.clusters, hc1.clusters),
                        fossil::rand.index(km.clusters, pcahc_clusters),
                        fossil::rand.index(hc1.clusters, pcahc_clusters))))
kable(table(km.out$cluster), col.names = c("Cluster", "Count"))
cor_mat <- round(cor(data[, 2:13]), 3)
ggcorrplot(cor_mat, hc.order = TRUE)
ggplot(gather(data[, 2:13]), aes(y = value)) +
  geom_boxplot() +
  facet_wrap(~key, scales = "free") +
  theme(axis.text.x = element_blank(), axis.ticks = element_blank())
avg_sil <- function(k) {
```

```
  x_k <- kmeans(data[, 2:13], k, nstart = 20)
  si <- silhouette(x_k$cluster, dist(data[, 2:13]))
  mean(si[, 3])
}

k.values <- 2:24
avg_sil_values <- map_dbl(k.values, avg_sil)

tibble(`val` = avg_sil_values, `clusters` = c(2:24)) %>%
  ggplot(aes(x = clusters, y = val)) +
    geom_point() +
    geom_line() +
    geom_vline(xintercept = 12, colour = "red") +
    xlab("Number of cluster (k)") +
    ylab("Average Silhouette")
kmeans_sil <- kmeans(data[, 2:13], 12, nstart = 20)
sil1 <- silhouette(kmeans_sil$cluster, dist(data[, 2:13]))
fviz_silhouette(sil1, print.summary = FALSE) +
  ggtitle("")
sil2 <- silhouette(hc1.clusters, dist(data[, 2:13]))
fviz_silhouette(sil2, print.summary = FALSE) +
  ggtitle("")
hc.out <- hclust(dist(pca$x[, 1:2]), method = "average")
pcahc_clusters <- cutree(hc.out, 12)
sil3 <- silhouette(pcahc_clusters, dist(data[, 2:13]))
fviz_silhouette(sil3, print.summary = FALSE) +
  ggtitle("")
plot(hclust(data.dist, method = "average"), labels = FALSE, hang = -1,
     cex = 1, main = "", sub="", xlab="")
abline(h = 0.1575, col = "red")
plot(hc.out, hang = -1, cex = 1, labels = FALSE, main = "", sub=NA, xlab="")
```

```
abline(h = 0.08, col = "red")
tibble(`prop` = round(pca$sdev^2/sum(pca$sdev^2),3), `pc` = c(1:12)) %>%
  ggplot(aes(x = as.factor(pc), y = prop)) +
    geom_bar(stat="identity", aes(fill = "#6495ed")) +
    xlab("Principle Component") +
    ylab("Proportion of Variance") +
    theme(legend.position = "none")
```

:::