**Getting the Data**

I used the handy {spotifyr} package to access the Spotify API. I defined a helper function to get artists related to Aphex Twin. This function wrapped around the spotifyr::get\_related\_artists() function to do the following:

1. Initialize a list of related artists by getting the 20 artists related most to Aphex Twin (as indicated by listener behavior).
2. For each of those 20 artists, get their 20 related artists. Bind these data to the initialized data in Step 1. Deduplicate so that no artists are listed twice.
3. Get the 20 related artists for each of the artists in the remaining dataset resulting after Step 2. Dedupe.
4. Do it one last time: Get the 20 related artists for each of the artists after Step 3. Dedupe.

The function looks like:

get\_related\_recursive <- function(id, iter) {

out <- get\_related\_artists(id)

done\_iter <- 0

while (done\_iter < iter) {

tmp <- unique(do.call(bind\_rows, lapply(out$id, get\_related\_artists)))

out <- unique(bind\_rows(out, tmp))

done\_iter <- done\_iter + 1

}

return(out)

}

Where id is the Spotify artist ID and iter is how many iterations one wants to do after the initial grab of related artists. Full code for getting all of the data can be found at the end of this post. This returned thousands of artists, and I only considered those that had more than 80,000 followers. This resulted in 329 artists. For each artist, I grabbed the audio features for every one of their songs. Three artists didn’t parse, resulting in 326 artists to look at.

**Choosing a Metric**

My hypothesis is that Aphex Twin has an especially eclectic discography. In statistical terms, we could characterize this as *the characteristics of his discography vary more than other artists’.* So we want to look at each artist’s discography and see how those variables mentioned above—danceability, energy, and valence—vary. We need a measure of multivariate spread or dispersion.

When looking at just one variable, the answer is straightforward: We can look at the variance for each artist to characterize how much spread there is in the distribution of songs. This would be a good enough metric for our uses, since all songs are measured on the same 0 to 1 scale.

But we want to look at *three* variables. Summing up the variances of each of the three variables sounds appealing, but there’s a problem: What if two of the variables are highly correlated? Imagine two situations where two variables are plotted against one another, one situation where the variables are completely uncorrelated (left panel) and one where they correlate highly (right panel).

library(tidyverse)

library(mvtnorm)

set.seed(1839)

rmvnorm(100, sigma = matrix(c(1, .8, .8, 1), 2)) %>%

rbind(rmvnorm(100, sigma = matrix(c(1, .0, .0, 1), 2))) %>%

as.data.frame() %>%

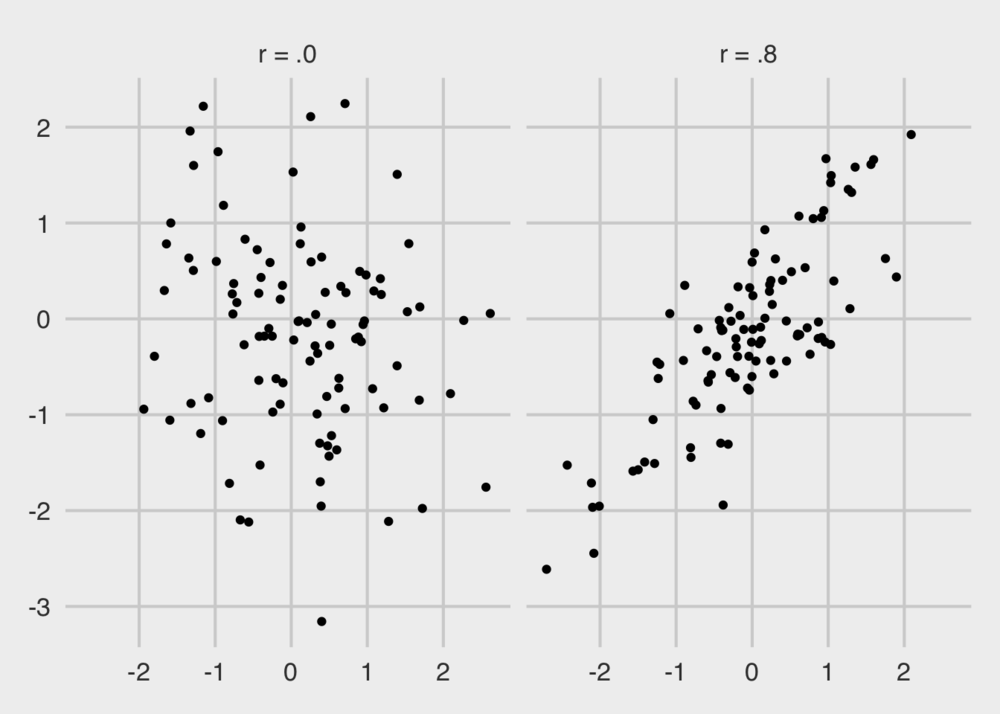
mutate(r = rep(c("r = .8", "r = .0"), each = 100)) %>%

ggplot(aes(x = V1, y = V2)) +

geom\_point() +

facet\_wrap(~ r) +

ggthemes::theme\_fivethirtyeight(16)



We can see that the uncorrelated points take up more space—they are more dispersed. We want a metric that appreciates this. Think of it this way: Since each of the three metrics from Spotify are on a 0 to 1 scale, imagine a cube where each of the sides has a length of 1. In one corner of the box is a point representing a song that is high in energy, valence, and danceability. Another corner has songs where the danceability is high, valence is high, but energy is low. And so on.

I am considering an eclectic catalog of songs to be one that has points all over this cube. The determinant of the covariance matrix can do that for us, which is sometimes known as “generalized variance,” see [Wilks (1932)](https://doi.org/10.1093/biomet/24.3-4.471) and [Sen Gupta (2006).](https://doi.org/10.1002/0471667196.ess6053.pub2)

**Analysis**

A highly-eclectic, varied discography of songs is thus one with a large generalized variance. For each of the artists, I got the generalized variance across all of their songs on Spotify, using the following code:

dat <- read\_csv("aphex\_twin\_related\_features.csv")

gen\_var <- sapply(unique(dat$artist\_name), function(x) {

dat %>%

filter(artist\_name == x) %>%

select(danceability, energy, valence) %>%

cov() %>%

det()

})

And we can plot the top 30 artists, similar-ish to Aphex Twin, that have the most generalized variance.

tibble(artist\_name = unique(dat$artist\_name), gen\_var) %>%

top\_n(30, gen\_var) %>%

arrange(gen\_var) %>%

mutate(artist\_name = factor(artist\_name, artist\_name)) %>%

ggplot(aes(x = artist\_name, y = gen\_var)) +

geom\_bar(stat = "identity", fill = "#4B0082") +

coord\_flip() +

labs(title = "Most Varied Discographies",

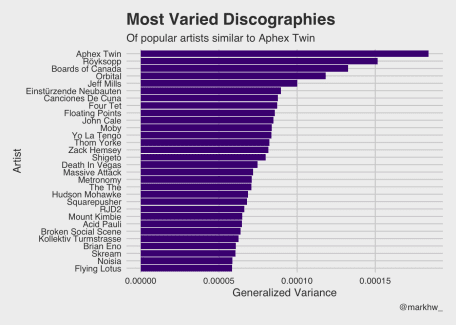
subtitle = "Of popular artists similar to Aphex Twin",

x = "Artist", y = "Generalized Variance",

caption = "@markhw\_") +

ggthemes::theme\_fivethirtyeight() +

theme(axis.title = element\_text())



Aphex Twin, by a wide margin, has the most varied Spotify discography in terms of danceability, energy, and valence. This is of a sample of popular, comparable artists—those that people who listen to Aphex Twin also listen to. Some other great artists are on here, as well: Boards of Canada, Four Tet, Thom Yorke, RJD2, Broken Social Scene, Flying Lotus.

**Code for Grabbing Data**

# get data ---------------------------------------------------------------------

library(tidyverse)

library(spotifyr)

get\_related\_recursive <- function(id, iter) {

out <- get\_related\_artists(id)

done\_iter <- 0

while (done\_iter < iter) {

tmp <- unique(do.call(bind\_rows, lapply(out$id, get\_related\_artists)))

out <- unique(bind\_rows(out, tmp))

done\_iter <- done\_iter + 1

}

return(out)

}

dat <- get\_related\_recursive("6kBDZFXuLrZgHnvmPu9NsG", 3)

dat <- dat %>%

filter(followers.total >= 80000) %>%

select\_if(function(x) !is.list(x))

write\_csv(dat, "aphex\_twin\_related.csv")

dat2 <- lapply(dat$name, function(x) {

cat(x, "\n")

tryCatch(

get\_artist\_audio\_features(x),

error = function(x) return(NA)

)

})

dat2 <- lapply(dat2, function(x) {

if (is.data.frame(x)) {

x %>%

select\_if(function(x) !is.list(x))

} else {

NA

}

})

dat2 <- dat2[![is.na](http://is.na)(dat2)] %>% # 3 artists failed to parse

do.call(bind\_rows, .)

write\_csv(dat2, "aphex\_twin\_related\_features.csv")