Lab 4: PCA and Clustering

Getting started

Click the Github Assignment link on Canvas, then open it in Posit Cloud as a new project from a GitHub repository.

Warm up

Let's warm up with some simple exercises. Open the `lab4.qmd` Quarto file. Update the YAML of your Quarto file with your information, render, commit, and push your changes. Make sure to commit with a meaningful commit message. Then, go to your repo on GitHub and confirm that your changes are visible in your `.qmd` file. If anything is missing, commit and push again.

Packages

We'll use the `tidyverse` package for data wrangling and visualization. The PCA and clustering functions we will use are in the `stats` package, which is included with base R, so we don't need to load it.

```
library(tidyverse)
```

Lab Goal

In economics, political science, and sociology, we often are very interested in how wealth and income are related to a variety of important economic, political, and social factors. One of the best ways to get all of these variables is through surveys. However, it turns out that people do not like answering income questions, so they see a lot of *non-response*, which is a problem. In addition, in low-income contexts, individuals may genuinely not know how much they generally earn (or it could fluctuate widely). In such contexts, researchers ofen use unsupervised learning techniques like PCA and clustering to see if the can get a sense of how affluent or wealthy individuals are. There are a variety of ways to do this; one important one that you will be exploring in this lab is by using possession questions - we ask how much of different types of possessions – such as chickens, basic cell phones, houses, etc – someone owns. Individuals in low income contexts in particular usually have a better sense of this than their actual income as a monetary amount.

Data

The data for this lab, contained in `data/vendor_data.RData`,[1] come from a project that Dr. Hoellerbauer is working on that analyzes a randomized control trial (RCT) focused on tax compliance carried out in Malawi. In Malawi, market vendors have to pay a fee (a flat tax) in order to be able to sell goods or services in a market. This fee is collected by fee collectors, who make the rounds each day.

Compliance is not optimal for a variety of reasons, a main one being that markets are underfunded and so are often in poor conditions. This causes market vendors to not want to pay the market tax, which in turn makes it harder for the government to improve the markets (this is called a *vicious cycle*). The RCT ran from October 2017 to March 2019 in 128 markets in 8 districts in Malawi. The goal of the RCT was to investigate different ways to break out of this vicious cycle in Malawian markets. [^If you are interested in this topic, I am happy to talk more about it during office hours!]

The data you will be using was collected via an in-person survey of market vendors carried out between October 2018 and January 2019. In total, 12,370 market vendors completed the survey. 2,531 of these vendors were asked a longer form of the survey, with more questions. We will be working with the responses from these individuals for this lab.

Please see the README page of the `data` folder of the GitHub repository for a complete codebook of the data.

The variables we will be using in PCA and clustering are `houses` through `smart_phones`. In this survey, respondents did answer a question about household income – `hh_income_trim_99`. We will also use this variable to compare how our clustering and dimensionality reduction seems to be doing – does it reflect wealth?

[1] This is a `.RData` file, which is a file format native to R that allows us to save one or more R objects in their entirety and then load them into our environment later on. We use the `load()` function to do so in the following way: `load("folder/file_name.RData")`

Note that because saving an R object in an `.RData` file saves the name of the object (usually called a variable in other languages), we do **not** need to assign the `load()` function a return value.

Exercises

Data Wrangling

Q1

First, let's fix a small issue and create a new, categorical version of income.

1. One particular respondent reported owning 80,000 houses. This is most likely untrue (and seriously affects the standard deviation, which, as you know from the reading and the example code, plays a key role in PCA in particular). Let's replace this value with `NA`.
2. Use the `cut_number()` function on the income variable to create a 10-level version of this variable. Call this `hh_income_cat10`.

PCA

Q2

What are the mean and variance of the possessions variables? Does it seem like we should scale them before doing PCA?

Q3

Use the `prcomp()` function to do PCA. Create two scree plots like the ones in Figure 12.3 of the textbook. Is there an elbow? How many components does it seem are sufficient?

Q4

Plot each of these components (the optimal number based on Q3) against household income. Arrange the plots using the `patchwork` package as demonstrated in this module's example code. Which component(s) seem to proxy for income, if any?

Clustering

It can sometimes be easier to try to discretize income instead. We will use clustering to do that.

Q5

Use `kmeans()` with $K = 10$. Set the seed to 67 beforehand and use 30 different starting points.

Then do the following:

1. Report the average possession variables in each of the ten groups.
2. Calculate the average household income within each category.
3. Visualize or show in some way the concordance between the `hh_income_cat10` variable and the 10 clusters (such as a cross tab or mosaic plot).

Finally, report on how well the clustering worked to approximate income groups. Put the 10 clusters into order from least wealthy to most wealth and explain why you chose that order.

Q6

Repeat the previous, but this time use hierarchical clustering with *average* linkage. Cut the tree a 10 clusters. Answer the same three questions.

Finally: is there any overlap between the clusters created through hierarchical clustering and the clusters created using K-means?

Note: If focusing on prediction, we could use cross-validation to try to see which linkage approach works best, as the section on clustering in ISLR points out. We'll talk more about that during the homework for this module.

Final Steps

When you have made your final edits, remember to render, stage the changed files, then commit, and push to the repo.

Don't forget to submit your pdf on Canvas.

Grading

Please put your names on your answer document and add the date. You won't get points for doing this, but you will *lose* points for not doing so.

Remember to comment your code. You don't get points for commenting, but you will *lose* points for not commenting your code.

For clarity's sake, you should label your code chunks (this helps make debugging easier). You will *lose* points for not naming chunks.

If an exercise asks a question, you should answer it in narrative form and not just rely on the code. You will be *penalized* otherwise.

Please suppress warnings and messages. You will *lose* points if you do not suppress warnings and messages.

If you are asked to make a visualization, make sure that it is presentable, professional in appearance, has a title, and has properly labeled axes. You will *lose* points if you do not do this.

Points Breakdown

The different components of this lab are worth the following points:

- Q1: 2 pts
- Q2: 2 pts
- Q3: 6 pts (2 pts for each scree plot, 2 pts for narrative response)
- Q4: 6 pts (5 pts for plots, 1 pts for interpretation)
- Q6: 5 pts (1 pt for 1., 2., 3., 2 pts for ordering and narrative response)
- Q7: 6 pts (Same as Q6, 1 pt for comparing two clustering approaches)
- Workflow (includes making sufficient commits): 4 pts

Total: 31 pts