Google fiber

Identifying Surrogate Geographic Research Regions with Advanced Exact Test Statistics

Steven Ellis | Elliss@Google.com

Challenge

In anticipating a regional product launch, you would ideally like your consumer research instrument to spell out every detail of the question you are trying to answer, e.g.,

> *"We are going to introduce this product in your market soon, would you buy this exact product, at this exact price, from a company just like ours?"*

but business needs force you to anonymize and abstract away important details of the choice.

What if...

You could test your instrument on populations that have very similar demographics - not just a "similar" city (e.g., Ann Arbor and Columbus), but populations matched at the individual- or household-level, using high-dimensional attributes, and at a zip code level, picking up areas you might never think of?

Turned into a Win-Win

By applying our open-source R package to publicly available data, we find surrogate regions with demographics that robustly match our region of interest (research needs satisfied) and are physically removed from the region of interest (business need satisfied).

If your product targets households, not individuals, our algorithm allows you to specifically look at that level of granularity, a methodological advance that every stakeholder can appreciate.

What we do...

Match your region of interest to other regions on key metrics of interest, such as income and household size - not at an aggregate level (e.g., t-test), but by computing pairwise matches of individuals from your region of interest to regions across the country.

Prep and Run

Prep: Download publicly available data from the Census Bureau and CDC.

Run: In 9 lines of R (our code is open source):

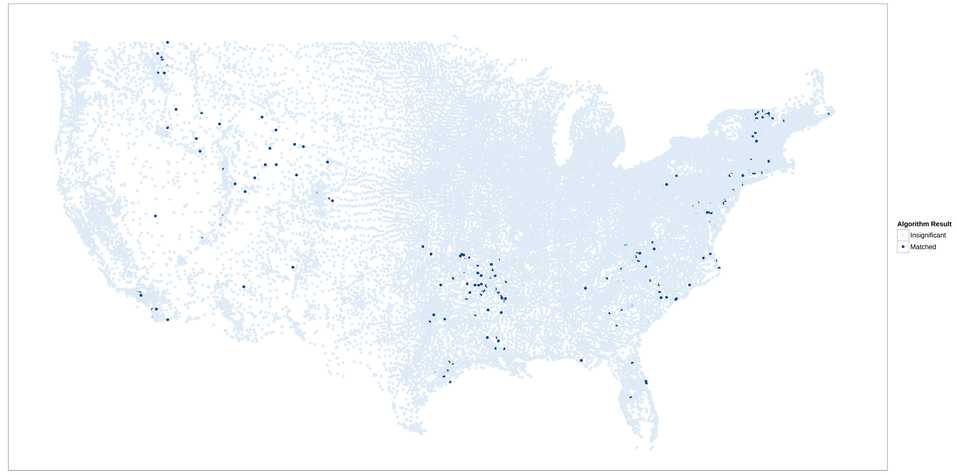
```
ZipDir ← "~/census_data/zip_files"
StatesDir ← "~/census_data/by_state_hh/"
StateFipsFile ← "~/state_fips.csv"
PumaCodeMapFile ← "~/new_puma.csv"
RegionFilenames ← c("ss10hil.csv") # you may
    specify multiple states
RegionPumas ← seq(03501, 03519, 1) # puma
    codes are a superset of zip codes
MatchVars ← c("HINCP","NOC","TEN","BLD")
Matches ← RunModel(100) # tune figure low
    for speed, high for accuracy
ZipCodes ← MatchesToZips(Matches)
```

Traditional Methods...

Are to match cities simply:

1. Based on comparisons of sample means at an aggregate level (e.g., mean income, mean age)
2. Based on regional similarity (e.g., DC - Baltimore, Boston - Cambridge)
3. Based on fiat (e.g., instinct that Ann Arbor and Columbus are similar)

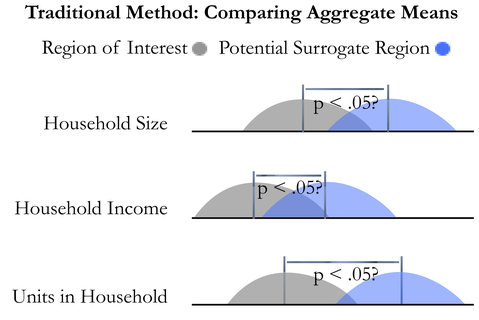
Results



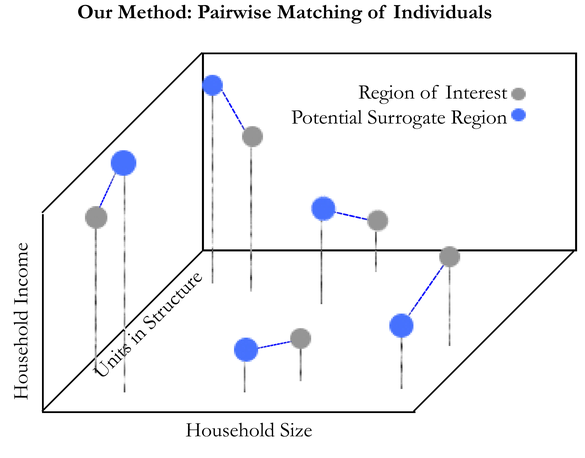
The above map represents the results of our algorithm's hunt for surrogate regions similar to Chicago, IL. Our computational approach shows surrogate regions where preservation of anonymity will not be a problem.

We chose to compare areas based on Household Income, Household Size, Household Ownership, and Units in Structure. The demographics chosen for comparison are entirely up to the researcher, limited only by the data collected by the Census. You may compare as many as you like.

Traditional Method versus Our Method



Traditional methods involve aggregate statistics (above) or fiat selection, our method (right) matches at an individual- or household-level, in high-dimensional space [1].



References

[1] Rosenbaum, Paul (2005), "An exact distribution-free test comparing two multivariate distributions based on adjacency," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 515-530.

Limitations

1. This package is designed to work with US Census data - it therefore is currently only applicable for US research.
2. The next step is to include OECD data.

Source Code

The documented R source code is available from the author.

Get in touch via `elliss@google.com` with questions and comments!