

INF552: Data Visualization: Project Report

Alexander Hägele

December 17, 2022

The visualizations are available on my homepage: <https://haeggee.github.io/posts/folktables>. Note that the text between visualizations might be missing at the time of handing in this project, yet the visualizations are all finalized and working. The goal is to present it in the theme of a blog post for other researchers; the descriptions in this report are comprehensive and most text will just be ported over to the website.

1 Introduction: *Visualizing folktables*

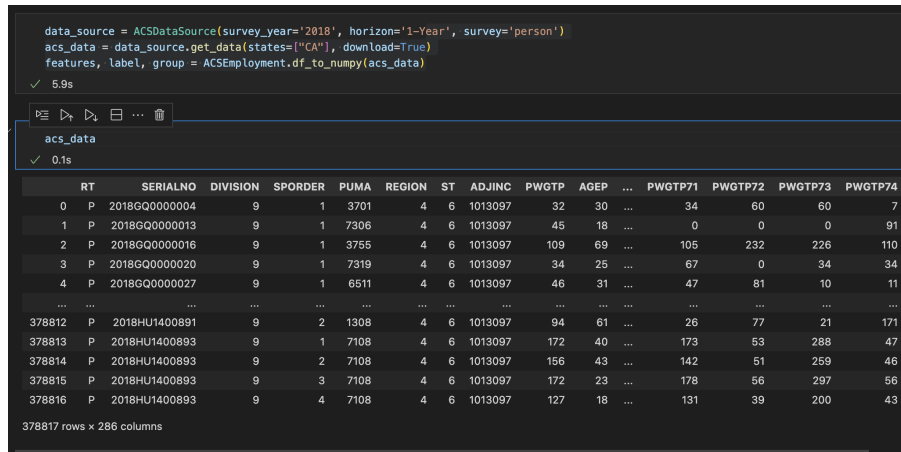
Datasets are central to the machine learning (ML) ecosystem. Besides providing training and testing data for model builders, datasets formulate problems, organize communities, and interface between academia and industry. This holds particularly true for real-world data of individuals, e.g. in the realm of classical tabular data. In this project, we focus on *folktables*, a new dataset for facilitating the benchmarking of machine learning algorithms.

Why the name? Folktables is a neologism describing tabular data about individuals. It emphasizes that data has the power to create and shape narratives about populations and challenges us to think carefully about the data we collect and use.

1.1 The dataset

The folktables dataset has only been recently introduced by Ding et al. [2021] with the goal of extending the existing data ecosystem for research on fair machine learning. In essence, folktables comprise multiple binary prediction tasks extracted from a single major corpus of US Census data derived by the US Census Bureau. It uses the American Community Survey (ACS) Public Use Microdata Sample (PUMS) files managed by the US Census Bureau, involving millions of US households each year. The datasets span multiple years and all states of the United States. This allows for studying temporal shifts and geographic variations as is demonstrated by this project. The dataset is accessible via a Python package called **folktables**, which interfaces with Census data sources and allows users to both access and manipulate data. An example view of this tabular data is shown in Fig. 1.

The complete description of the original dataset, including motivation, can be found in the paper [Ding et al., 2021]. More information is available on the GitHub page accompanying the package. The tasks are summarized in the following; an overview of the most important features (columns), including their possible values, is found



```
data_source = ACSDataSource(survey_year='2018', horizon='1-Year', survey='person')
acs_data = data_source.get_data(states=["CA"], download=True)
features, label, group = ACSEmployment.df_to_numpy(acs_data)
```

✓ 5.9s

acs_data

✓ 0.1s

	RT	SERIALNO	DIVISION	SPORDER	PUMA	REGION	ST	ADJINC	PWGRP	AGEP	...	PWGRP71	PWGRP72	PWGRP73	PWGRP74
0	P	2018GQ0000004	9	1	3701	4	6	1013097	32	30	...	34	60	60	7
1	P	2018GQ0000013	9	1	7306	4	6	1013097	45	18	...	0	0	0	91
2	P	2018GQ0000016	9	1	3755	4	6	1013097	109	69	...	105	232	226	110
3	P	2018GQ0000020	9	1	7319	4	6	1013097	34	25	...	67	0	34	34
4	P	2018GQ0000027	9	1	6511	4	6	1013097	46	31	...	47	81	10	11
...
378812	P	2018HU1400891	9	2	1308	4	6	1013097	94	61	...	26	77	21	171
378813	P	2018HU1400893	9	1	7108	4	6	1013097	172	40	...	173	53	288	47
378814	P	2018HU1400893	9	2	7108	4	6	1013097	156	43	...	142	51	259	46
378815	P	2018HU1400893	9	3	7108	4	6	1013097	172	23	...	178	56	297	56
378816	P	2018HU1400893	9	4	7108	4	6	1013097	127	18	...	131	39	200	43

378817 rows x 286 columns

Figure 1: Example view of the ACS data after downloading it via the folktables API.

in Appx. A. For more information about all features or the ACS US census data, please see the official ACS PUMS documentation.

ACSIIncome (Target feature: PINCP): predict whether an individual's income is above \$50,000, after filtering the ACS PUMS data sample to only include individuals above the age of 16, who reported usual working hours of at least 1 hour per week in the past year, and an income of at least \$100.

ACSPublicCoverage (Target feature: PUBCOV): predict whether an individual is covered by public health insurance, after filtering the ACS PUMS data sample to only include individuals under the age of 65 and those with an income of less than \$30,000. This filtering focuses the prediction problem on low-income individuals who are not eligible for Medicare.

ACSMobility (Target feature: MIG): predict whether an individual had the same residential address one year ago, after filtering the ACS PUMS data sample to only include individuals between the ages of 18 and 35. This filtering increases the difficulty of the prediction task, as the base rate of staying at the same address is above 90% for the general population.

ACSEmployment (Target feature: ESR): predict whether an individual is employed, after filtering the ACS PUMS data sample to only include individuals between the ages of 16 and 90.

ACSTravelTime (Target feature: JWMNP): predict whether an individual has a commute to work longer than 20 minutes, after filtering the ACS PUMS data sample to only include employed individuals above the age of 16. The threshold of 20 minutes was chosen as it is the US-wide median travel time to work in the 2018 ACS PUMS data release.

1.2 This project

Motivation. Census data is often used by social scientists or economists to study aspects like inequality or sociodemographic properties. This is *not* the motivation of this project. Folktables has been introduced for the empirical study of ML algorithms – and as it is a novel benchmark, the visualizations created in this project can serve as an entry point to working with the datasets and their prediction tasks. For instance, researchers can find key statistics at a glance and see important features together with their signals which ML algorithms could pick up; it can help them understand the distribution shifts and provide visual insight into the complexities of the data, such as under- or misrepresentation of certain characteristics.

Notable challenges. Investigating the folktables Python package and its source code, one can find that the full dataset (before applying task-specific filters) is simply the *full* ACS census data, accessed through the official government website and its API. Over the span of all years 2014-2018, this covers multiple million individuals each year, as well as hundreds of features; in total, it contains roughly 3GB of data when trying to combine it all. Even when filtering out the features of interest for the tasks, more than 30 columns remain. It is thus infeasible to consider the whole dataset for all visualizations. At the same time, there are endless ways to visualize parts of the data.

Consequently, I have tried to extract smaller datasets and critical characteristics that are interesting to visualize. In particular, I tried to create a certain storyline, from a coarse scale looking at general trends, to distribution shifts, and then statistics dividing the population into groups. All datasets were extracted in Python notebooks interacting with the folktables package, then stored in CSV or JSON files to load in JavaScript. I link notebooks containing the code for reproducibility at the appropriate locations¹.

Outline of visualizations & this report. In a short summary, the following visualizations were created.

- **Overview map:** A single choropleth map (for a selectable task), together with key numbers for the 2018 prediction tasks. The color mapping shows the deviation from the US-wide mean percentage of targets per task. Done in Vega-Lite.
- **Feature importance:** To pin down a subset of interesting features to visualize, I trained a simple classifier and extracted the most important features for each task, which were plotted in a barplot. Done in Python and Matplotlib.
- **Scatterplot/Binning matrix:** Visualizing a subset of the data in a matrix to see different relations at hand, using the most interesting features identified before. Depending on the type of features (numerical or categorical), I sometimes use binning histograms instead of scatterplots. Done with Vega-Lite.
- **Ridgeline Plot & Bar chart:** Focusing on the geographical and temporal distribution shifts, a ridgeline plot that shows the distribution of certain selectable features per state. A slider allows selecting the year.

¹Please note that running on Colab most often leads to exceeding the RAM limit; it is easier to download the notebooks and run them locally with more RAM.

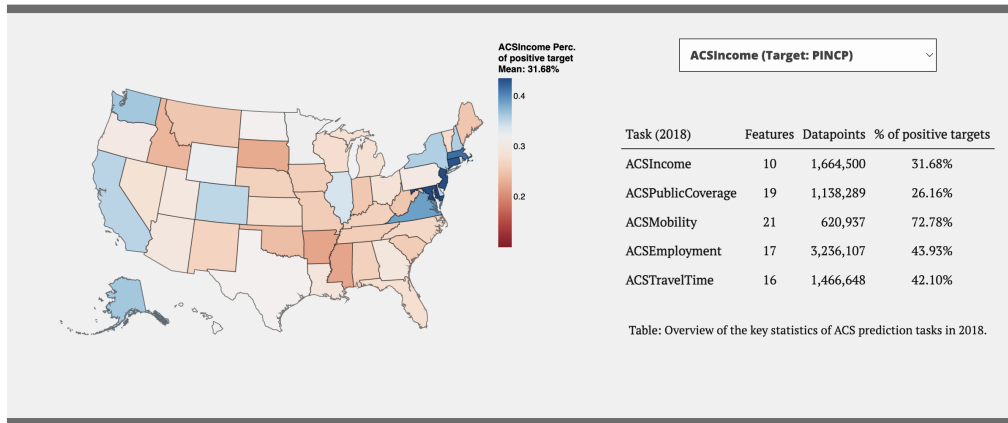


Figure 2: Overview map

Side-by-side, there is a barplot in order to keep in mind the sample size for each state. Done in D3 (ridgeline) and Vega-Lite.

- **Choropleth matrix grid:** Similar to what we have seen in the lectures, the last visualization is a matrix grid split by year (rows) and ethnical groups (columns). It again shows the percentage of positive targets (where the task is selectable) with a diverging color scheme that has the US-wide mean at its center. Additionally, I added a matrix that shows the population count (number of samples). Done in Vega-Lite.

In the next section, I describe the motivation and reasoning behind each visualization, including its design and features (interaction).

2 Visualizations

2.1 Overview

As a simple entry point to the datasets and prediction tasks, I created a choropleth map that visualizes the percentage of positive targets for a prediction task per US state for the year 2018. Figure 2 shows a screenshot of the result.

Technical details. The summary statistics were collected from the full ACS data in a JSON file, as outlined in the notebook here. This JSON file is then loaded into JavaScript, where I use Vega-Lite to create the Choropleth based on the Albers USA mapping. The midpoint of the color scheme is chosen as the US-wide positive target average.

Description. As outlined above, the main idea here was to start with a coarse perspective on the task. This is put next to a table that summarizes the key numbers (no. of features, data points, mean percentage of positive targets). The task is selectable via a simple dropdown menu. Hovering over a state shows the state's description, i.e. state code, target percentage, and count of people (samples). The choice of diverging color scheme was deliberately chosen to upfront show certain geographical contrasts between states. This is a theme throughout the other visualizations.

2.2 Visualizing a subset of the data

Diving deeper into the dataset and prediction tasks, the goal was to get some understanding of the tasks, in particular of the connection between features and outcomes as well as possible groupings. As outlined in Sec. 1.2, the full dataset contains too many features and aspects that could be visualized. As a consequence, I have tried to identify a subset of the data (features) that are of independent interest or can be of interest to other researchers using the dataset.

2.2.1 Feature importance.

To have a first investigation of the data – and connect the investigation to the theme of machine learning – I trained a decision tree on all prediction tasks to sort features by their importance for predictive power. The resulting bar plots are shown in Fig. 3.

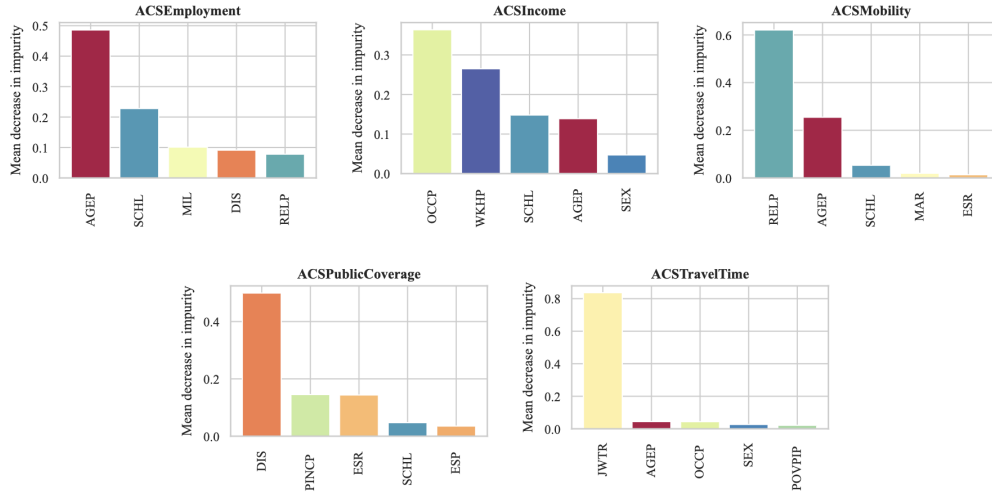


Figure: Plotting the feature importance for all 5 prediction tasks using a Gradient Boosting Classifier and the mean decrease in impurity.

Figure 3: Feature importances for a simple gradient-boosted decision tree and the mean decrease in impurity (MDI).

Description. This notebook here contains the code to run and reproduce the experiments. The training was done on US-wide data from 2014. I used the gradient-boosted decision tree provided by `scikit-learn` with exponential loss, `num_estimators` 5, `max_depth` 5, with all other hyperparameters set to the default. This is exactly the same model as the authors Ding et al. [2021] used in their experiments, denoted as the gradient-boosted decision tree (GBM). The feature importance is extracted using the concept of mean decrease in impurity (descriptions available here)².

The bar plots were created using Matplotlib within the same notebook. The colors were randomly chosen and set coherently to be the same for each feature across all plots. The figures are imported as SVG files and are thus not interactive; the foremost point was to have a relative ordering of features in order to select them for further visualization. The concrete values do not matter. Some interesting observations can be made: for instance, the relationship status (RELP) seems to be a good indicator (for this model) for the mobility task, or the means of transportation to work (JWTR) is of great importance (for this model) for predicting the travel time to work.

2.2.2 Scatterplot matrix.

Having identified (possible) important features, I chose to compare their relationships with each other and the target variable per task. This is done in a scatterplot matrix visible in Fig. 4.

Technical details. In the same notebook used for the feature importance, I extracted 1000 sample individuals and their corresponding features (the top three identified by MDI + the target) for each prediction task to store them in CSV files. These features are then put together in a Vega-Lite scatterplot matrix.

Description. Scatterplot matrices are a great analytical tool, especially for the first investigation of data as they can show larger quantities. Here, the matrix serves as a first inspector of correlation in the data, and possible clustering effects. For instance, one can see a clear linear correlation between education level (SCHL) and personal income (PINCP). The color coding was chosen to discriminate between positive and negative targets. Moreover, holding down the shift key enables brushing in order to select certain points in one cell and highlight them across the matrix.

As a downside to the scatterplot, it doesn't work well with categorical data which is inherent to this tabular data. Therefore, I decided to switch to a *binned* scatter matrix for tasks that were identified to have important categorical features (ACSMobility, ACSPublicCoverage). The size of the circles indicates the bin count. These plots do not have brushing capability. Similarly to before, it enables the observation of clustering effects, e.g. if the bins tend to be larger for positive targets for certain feature categories.

²*Disclaimer:* Impurity-based feature importances have multiple drawbacks and are by no means perfect. Here, they just serve as a simple tool for the first investigation of the data.

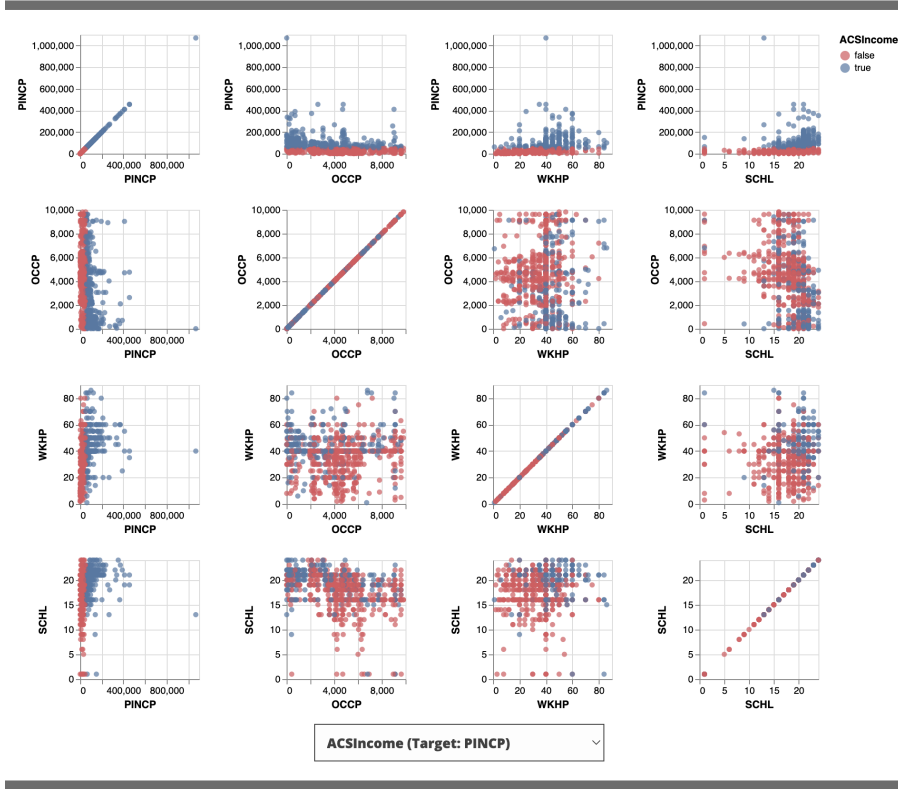


Figure 4: Scatterplot Matrix

2.3 A look at the distribution shifts

Let's dive deeper into the distribution of the features. Of particular interest to this dataset and benchmarking of ML algorithms is the concept of *distribution shifts*. To that end, I created a ridgeline visualization of selected features, shown in Fig. 5, side-by-side to a barplot that takes into account the population size additionally separated by gender.

Technical details. The Python code that was used to extract samples for the ridgeline plots can be found [here](#). This was done for the general ACS data (not task-specific) and 10'000 samples per state were taken (or less, if the initial count is lower). The ridgeline plot is then created with D3 and a kernel density estimator (starting code taken from [here](#)) with different hyperparameters for each feature. The diverging color scale midpoint is set to the US-wide mean for the selected feature.

Description. Ridgeline plots are of great value when the number of groups to represent is large, whereas a classic plot per group would take too much space; allowing the densities to overlap uses space more efficiently. This dataset is thus a clear use case for such plots, using important features such as personal income or age distribution per state which are a large number (> 50) of groups. They work well when there is a clear pattern in the result (e.g. if there is a clear separation or shifts); if there is not, the plot brings this to light.

This is the most interactive of the visualizations and it has multiple interesting channels. First, the diverging color scheme (as before) allows us to quickly grasp the gap and shift between states (characterized by the means). Hovering over the densities reveals the state code and mean and highlights the selected density. One can choose between two orderings of the states: the first one orders them by the mean (and thus also the color), whereas the second keeps them in alphabetical order. The first is insightful to e.g. follow the trend of a particular state when moving the year slider and following the transitions up/down. The second is insightful when looking at the transition of the density of a particular state over time, for example, if a density moves to the right or left.

Focusing our attention to the right, we have the barplot that shows the sample size split per gender and state. In particular, this is ordered in the same way as the left in order to take into account the sample size and representation when comparing ridgeline distributions. I initially had a diverging barplot (y-axis in the middle) to compare the gender counts more direct; however, I found it to clutter the visualization and it was not very insightful (the numbers do not differ largely). Instead, I stuck to the simpler barplot which shows the actual counts when hovering over the bars. I also removed color red/blue color coding that could lead to confusion if connecting the left diverging scheme to the gender separation, which is not related. Similar to the ridgeline plot, the bars change with respect to the selected year. I had planned to include a possible third order which

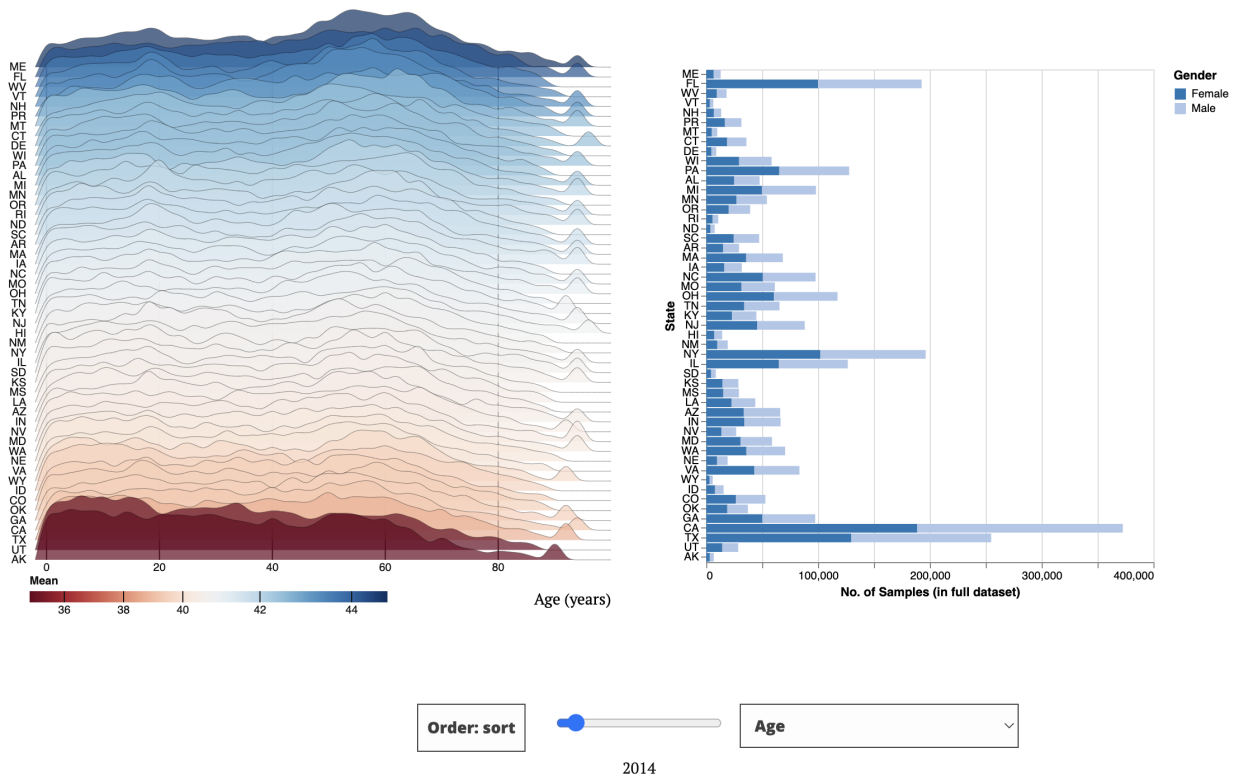


Figure 5: Ridgeline Plot

sorts by the sample size but unfortunately could not find the time to implement it.

2.4 Matrix map view by separating groups

Having looked at both correlation and grouping effects, as well as distribution shifts, the last focus of the visualizations is the contrast between different population subgroups as well as their representation in the dataset. The two resulting matrices are put next to each other in Fig. 6 and Fig. 7.

Technical details. The notebook that extracts the target percentages per state and group, together with a sample count, can be found [here](#). The extracted data is stored in a JSON file, which is then loaded into JavaScript. The choropleth matrices were then created using Vega-Lite and row/column separation.

Description. Coming back to the main motivation of folktables and, by extension, this visualization, is the understanding of ML algorithms and their societal impact. In particular, we would like to understand the complexities of the data that is used to train and evaluate models, and what impact it can have on individuals.

I thus decided to use a choropleth map in a matrix view (similar to what we have seen in the lecture) to compactly put groups side-by-side and visually capture the differences with respect to the prediction tasks. As before, I have focused on a diverging color scheme that captures the differences to the US-wide mean. This allows an understanding of the divergence between specific subgroups and the effects e.g. locally trained classifiers could have. Hovering over a state reveals the name, sample size, and exact percentage.

As the authors of folktables emphasize, there may be very few individuals with particular characteristics (e.g. ethnicity) in certain states, and generalizing conclusions from these few individuals may be highly inaccurate. This is also very apparent when hovering and looking at certain sample counts which are very low (often below 50). Further, benchmarking fair machine learning algorithms on datasets with few representatives of certain subgroups may provide the illusion of “checking a box” for fairness, without substantive merit. Therefore, I have added another matrix that captures the sample size in the same choropleth format to highlight such contrast. Here, I chose a linear scheme on a logarithmic scale. Other representations were not insightful, for example, a linear scale puts most states and groups to non-distinguishable colors.

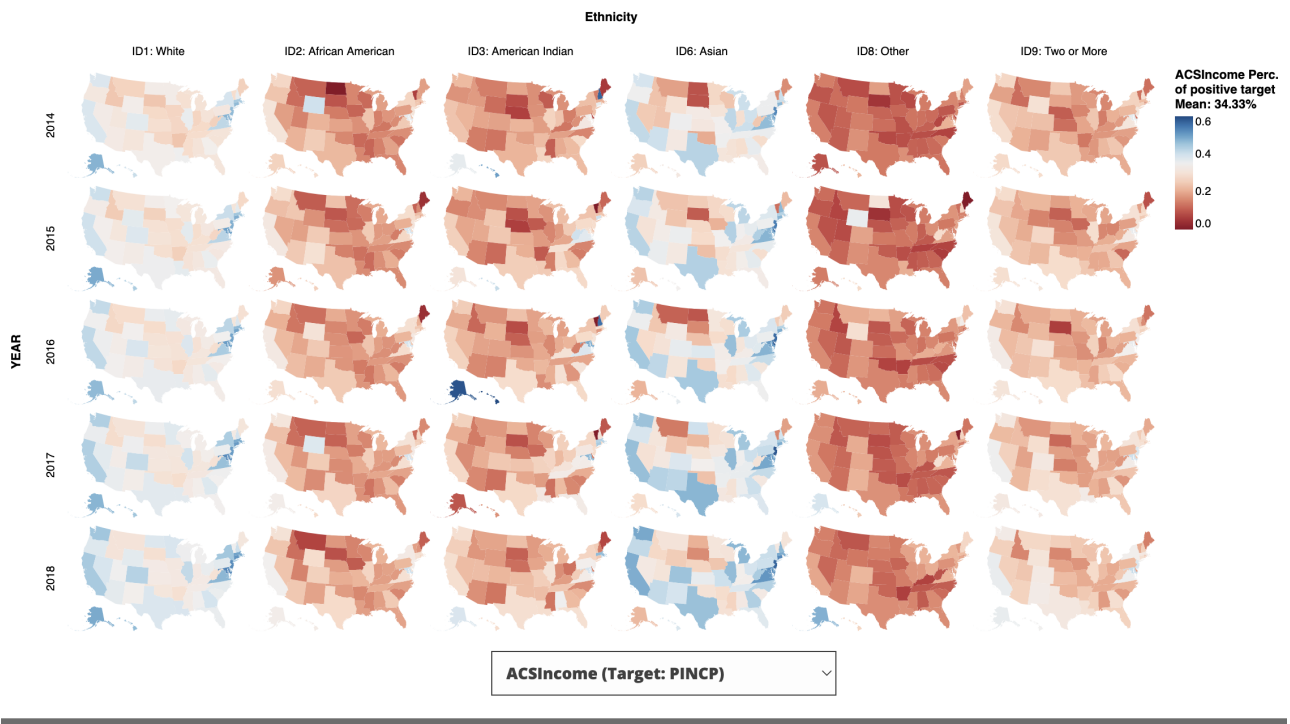


Figure 6: Choropleth matrix with diverging color scales for target percentages

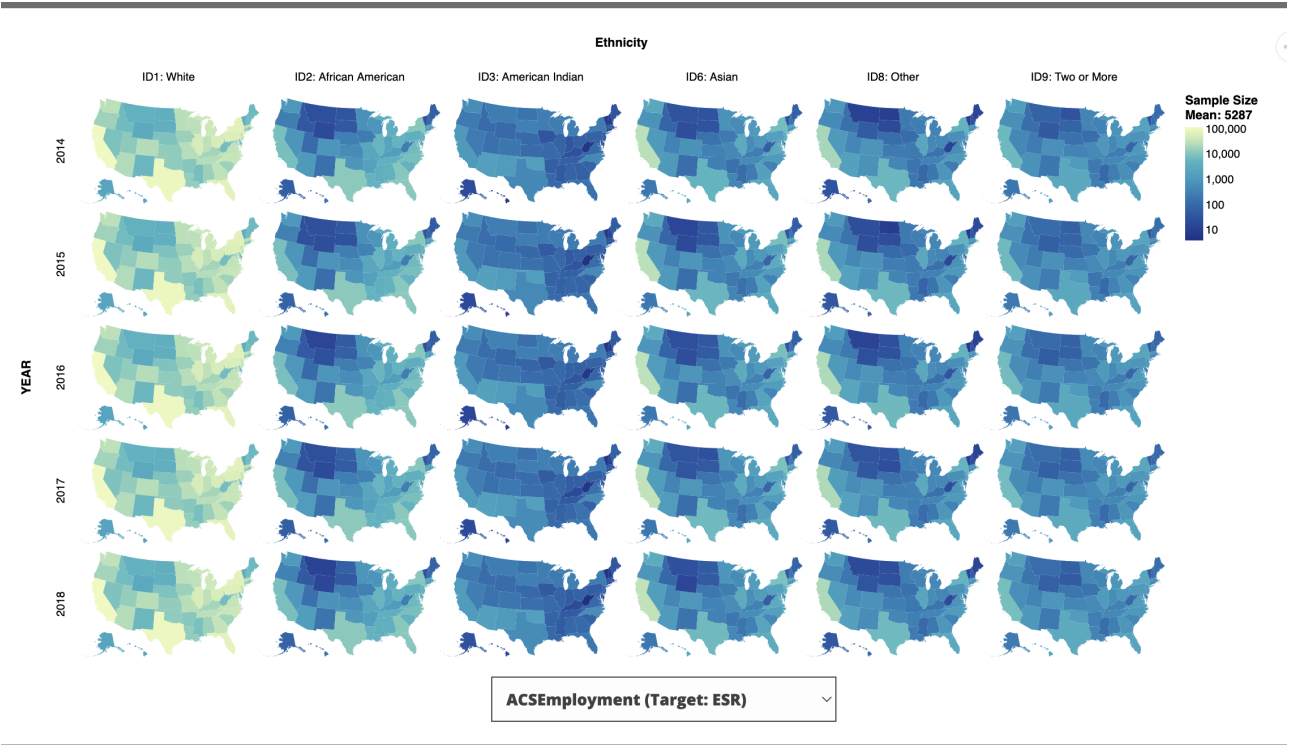


Figure 7: Choropleth matrix indicating sample size

References

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34:6478–6490, 2021.

Appendix A Feature descriptions

The most important features used in the visualizations are summarized here for comprehension. A full list is also available in the original FolkTables paper in the Appendix.

AGEP (Age): Range of values:

- 0 – 99 (integers)
- 0 indicates less than 1 year old.

DIS (Disability recode): Range of values:

- 1: With a disability
- 2: Without a disability

ESR (Employment status recode): Range of values:

- N/A (less than 16 years old)
- 1: Civilian employed, at work
- 2: Civilian employed, with a job but not at work
- 3: Unemployed
- 4: Armed forces, at work
- 5: Armed forces, with a job but not at work
- 6: Not in labor force

JWMNP (Travel time to work): Range of values:

- N/A (not a worker or a worker that worked at home)
- integers 1 - 200 for minutes to get to work
- top-coded at 200 so values above 200 are coded as 200

MIG (Mobility status (lived here 1 year ago): Range of values:

- N/A (less than 1 year old)
- 1: Yes, same house (nonmovers)
- 2: No, outside US and Puerto Rico
- 3: No, different house in US or Puerto Rico

MIL (Military service): Range of values:

- N/A (less than 17 years old)
- 1: Now on active duty
- 2: On active duty in the past, but not now
- 3: Only on active duty for training in Reserves/National Guard
- 4: Never served in the military

OCCP (Occupation): Please see ACS PUMS documentation for the full list of occupation codes

PINCP (Total person’s income): Range of values:

- integers between -19997 and 4209995 to indicate income in US dollars
- loss of \$19998 or more is coded as -19998.
- income of \$4209995 or more is coded as 4209995.

RAC1P (Recoded detailed race code): Range of values:

- 1: White alone
- 2: Black or African American alone
- 3: American Indian alone
- 4: Alaska Native alone
- 5: American Indian and Alaska Native tribes specified, or American Indian or Alaska Native, not specified and no other races
- 6: Asian alone
- 7: Native Hawaiian and Other Pacific Islander alone
- 8: Some Other Race alone
- 9: Two or More Races

RELDP (Relationship): Range of values:

- 0: Reference person
- 1: Husband/wife
- 2: Biological son or daughter
- 3: Adopted son or daughter
- 4: Stepson or stepdaughter
- 5: Brother or sister
- 6: Father or mother
- 7: Grandchild
- 8: Parent-in-law
- 9: Son-in-law or daughter-in-law
- 10: Other relative
- 11: Roomer or boarder
- 12: Housemate or roommate
- 13: Unmarried partner
- 14: Foster child
- 15: Other nonrelative
- 16: Institutionalized group quarters population
- 17: Noninstitutionalized group quarters population

SCHL (Educational attainment): Range of values:

- N/A (less than 3 years old)
- 1: No schooling completed
- 2: Nursery school/preschool
- 3: Kindergarten
- 4: Grade 1
- 5: Grade 2
- 6: Grade 3
- 7: Grade 4
- 8: Grade 5
- 9: Grade 6

- 10: Grade 7
- 11: Grade 8
- 12: Grade 9
- 13: Grade 10
- 14: Grade 11
- 15: 12th Grade - no diploma
- 16: Regular high school diploma
- 17: GED or alternative credential
- 18: Some college but less than 1 year
- 19: 1 or more years of college credit but no degree
- 20: Associate's degree
- 21: Bachelor's degree
- 22: Master's degree
- 23: Professional degree beyond a bachelor's degree
- 24: Doctorate degree

SEX (Sex): Range of values:

- 1: Male
- 2: Female

WKHP (Usual hours worked per week past 12 months): Range of values:

- N/A (less than 16 years old / did not work during the past 12 months)
- 1 - 98 integer valued: usual hours worked
- 99: 99 or more usual hours