IMPROVING REFUGEE RESETTLEMENT IN THE UNITED STATES THROUGH ANALYTICS

A Major Qualifying Project submitted to the faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the degree of

Bachelor of Science

BY:

Roberto Esquivel

Toni Joy

Jean Philippe Miralda

Jenna Vandervort

ADVISOR:

Andrew C. Trapp, Ph.D.

SPONSOR:

HIAS (Hebrew Immigrant Aid Society)

Abstract

HIAS is a refugee resettlement agency that works with 19 affiliate locations to resettle refugee populations throughout the United States. Our project provided alternative viewpoints to augment the way HIAS resettles refugees, including a proof of concept for optimizing refugee placement by analyzing refugee and location-based data. We identified complementary cities that could be added to HIAS's network by using clustering algorithms. Finally, we improved the accuracy of *Annie*, a refugee placement model, by developing scripts to extract location-specific data.

Acknowledgements

Our team would like to thank the following people for contributing to the development, progress, and eventual success of this project:

- **Professor Andrew Trapp** for his constant guidance and support and for challenging us to think critically, step outside our comfort zone, and ultimately achieve all of our goals
- **Narges Ahani** for providing us with important information about refugee resettlement and the functionality of *Annie*
- **Professor Gbetonmasse Blaise Somasse** for helping us to understand how to incorporate economic concepts into our refugee placement optimization model
- **Professor Alessandro Martinello of Lund University** for helping us to understand the algorithm behind *Annie* and how it could be enhanced
- **HIAS** for giving us the opportunity to work on a meaningful project and to present our recommendations to an empowering community of people dedicated to improving the lives of refugees in the United States

Without the support and guidance from each of these individuals and organizations, we would not have been able to complete our project. Thank you all for your unwavering dedication, guidance, and support, and for contributing to the success of our project.

Authorship

All four team members contributed equally to our project. We each contributed to the scoping of our initial project, the execution of our goals, and the writing and editing of our final report.

Executive Summary

The world faces a major challenge in resettling the vast number of refugees seeking relocation from their homeland. As of 2018, the UNHCR estimated that there are 19.9 million people in the world classified as refugees (UNHCR, 2018a). However, during this same year only 55,700 refugees were resettled (UNHCR, 2018b). The United States (US) remains one of the leading contributors to the growing gap between refugees requiring resettlement and refugees who are actually resettled. Under the new US administration, the cap on the number of refugees permitted to enter the country has sharply declined to only 30,000 in 2019 (Cepla, 2019).

As the US administration continues to reduce the number of refugees allowed to resettle in the US, many US citizens have demonstrated their opposition to these policies by supporting resettlement agencies (Ahmed, 2017). The resettlement process is delegated to the network of resettlement agencies that allocate refugees throughout their affiliate locations and who, along with volunteers, provide medical, financial, and housing support (Refugee Council USA, 2018). These agencies are responsible for making the important decision of placing refugees in a location that will ensure their long term economic success and social integration, a task that can prove to be very complex (Ahmed, 2017; Trapp et al., 2018).

We expanded on current research that aims to improve how placement and matching is done by refugee resettlement agencies. In particular, our research capitalizes on the use of publicly available location-specific data analytics to explore concepts in the optimization of refugee placement. We also conducted an in-depth analysis of the US resettlement agency HIAS (founded as the Hebrew Immigrant Aid Society) and their network of 19 affiliate resettlement agencies.

To complete our analysis, we established three objectives. First, we performed a proof of concept for optimizing refugee placement through an analysis of refugee-specific and

location-specific data. We also analyzed and identified areas of improvement for HIAS's agency network by evaluating location-specific data relative to the other locations in the US. Finally, we improved the accuracy of the predictive modeling within *Annie*, a current optimization model for refugee resettlement, by developing scripts that extract location-specific economic data.

We achieved our objectives through industrial engineering and data science techniques, including optimization, data analysis, and statistical analysis. By considering both individual and location-based factors, we developed an optimization model that maximizes a refugee's projected income and places them accordingly in one of the 19 affiliate locations. Through this exercise, we identified which factors affect the potential income of a refugee, and how this income differs by location and the characteristics of each individual refugee. We used refugee-specific factors such as English proficiency, previous employment and experiences, and education level, and location-specific factors such as percent of non-English speaking people and number of physicians per 100,000 people.

Our optimization model placed 40 families in a suitable location, while using up full capacities in each affiliate location. We identified an optimal objective function value of $1,378,069, which is the combined income of placing 40 families in optimal affiliate locations. These results enabled us to identify key features of each location, such as whether the location is suitable for a refugee with multiple health issues or a refugee who has minimal English proficiency.

While this optimization model explored the possibilities of resettling refugees within HIAS's current network of affiliates, we employed analytics to describe the agency's current state and explore possible expansions to the HIAS network. We used analytics to compare the 19 affiliate locations to other locations in the US based on key socioeconomic metrics. Using an online resource, DataUSA, we analyzed the 3,142 counties and census areas in the US based on specific socioeconomic indicators available. We compiled 25 different metrics that cover all of the 10 domains of

integration and grouped them into six areas: economy, health and safety, education, living and housing facilitators, social integration facilitators, and Jewish population density. This enabled us to understand the performance of each of the 19 counties within HIAS's network compared to all counties in the US.

By using clustering techniques and principal components analysis, we identified complementary counties for each of the 19 affiliate locations based on socioeconomic indicators and percent of the population that is Jewish. The top five counties we selected that complement HIAS's network the best are Howard county in Maryland, Loudoun county in Virginia, Collin County in Texas, DuPage County in Illinois, and Adams County in Colorado. We summarized our results using dashboards, which enable each of the affiliate locations to be compared to each other and to other counties in the US.

Additionally, we employed data science techniques to expand *Annie* by incorporating characteristics of each location to inform the algorithm that predicts employment probability. We communicated with Professor Alessandro Martinello of Lund University, who is the architect of the predictive algorithm, and received his input on which location characteristics would be most beneficial for improving the algorithm's accuracy. He also provided insight into how these characteristics could be incorporated into the code that powers *Annie*.

We also developed python scripts that pull data from different websites and structure the data to be merged with the dataset used by the current predictive algorithm. These metrics increase *Annie*'s accuracy by incorporating the economic environment in each location to inform its employment probability estimates. Regardless of how favorable a person's likelihood for employment is, if there is not a successful economic environment to facilitate employment, it will be more difficult for a person to be employed.

As a final step to this project, we developed a set of recommendations for HIAS to use to enhance their refugee resettlement process. We first recommend that HIAS

considers collecting better and more information, particularly about affiliate locations and refugee employment. Data regarding medical conditions and refugee preferences for an affiliate location may be beneficial to this process. Increased information would have a significant impact on future data-driven decision making tools and refugee integration.

We also recommend that if HIAS considers expanding its network to include additional affiliate locations, then it should use the results of our clustering algorithm to aid in its decision-making. Our clustering algorithm output allows HIAS to choose which metric has the most importance for them. They can then identify clusters of counties that outperform in the decided metrics, enabling HIAS to explore additional options for each affiliate location.

Our final recommendations are to draw upon publicly available sources of data to aid placement and expansion decision making, and to use and continue to develop tools that increase the accuracy of *Annie*. Public sources of information such as the US Census and the Bureau of Labor Statistics can enable HIAS to take additional factors into consideration when placing refugees. Access to more data would improve the accuracy of resettlement optimization models such as *Annie*. If the resettlement decisions are more informed and a more data-driven approach is used, the success of refugee integration can be greatly enhanced.

While the present may look bleak as the resettlement cap is consistently declining, it is important to remain optimistic about the future of refugee resettlement, as with future administrations comes the potential for the cap to increase again. Our research makes a compelling argument for how the US and specifically HIAS can improve refugee placement decisions, resulting in more successful resettlement results.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

As of 2018, the United Nations High Commissioner of Refugees (UNHCR) estimated that at least 19.9 million people around the world lie under their mandate as *refugees*, the highest number in recorded history (UNHCR, 2018a). Of these, the UNHCR estimates that 1.4 million are in need of resettlement from an asylum country to a third and permanent host country (UNHCR, 2019). Despite the high demand for resettlement, the UNHCR was able to support only around 163,000 individuals for resettlement in 2016, representing about 14% of the total demand. Moreover, due to fluctuating global quotas for host nations that admit refugees, the UNHCR was able to support only around 55,700 individuals in 2017 (UNHCR, 2018b).

In spite of being the nation that has traditionally helped to resettle the most refugees (UNHCR, 2018b), the United States (US) is a major contributor for the drop in global numbers of refugees resettled. The country's administration reduced the cap for intake of refugees from a total of 110,000 individuals in FY 2017 down to just 45,000 in FY 2018 ("RPC - Refugee Processing Center", 2018). As the Trump administration continues to push back to reduce the intake of refugees into US borders, the panorama is not set to improve, with the cap for refugee intake decreasing even further to 30,000 for FY 2019 (Cepla, 2019).

Yet as the US administration continues to reduce the number of refugees allowed to resettle in the US, a significant portion of the population presents opposition to the administration by increasing their support towards resettlement agencies that lead resettlement initiatives in the US (Ahmed, 2017). For many years now, the resettlement process has been delegated to the network of resettlement agencies that allocate refugees throughout their affiliate locations and who, along with volunteers, provide medical, financial, and housing support (Refugee Council USA, 2018). The US administration's policies have led to an imbalance in supply and demand, where some agencies claim to have significantly more volunteer capacity than refugees to resettle

(Ahmed, 2017). In addition to providing support for the refugees, resettlement agencies are responsible for making the important decision of placing refugees in a location that will ensure their long term economic success and social integration, a task that can prove to be very complex and difficult to get just right (Ahmed, 2017; Trapp et al., 2018).

Several longitudinal studies demonstrate the complexities associated with studying and keeping track of refugee populations over the years, particularly in measuring social integration (Nibbs, 2017; Lichtenstein, Puma, Engelman, & Miller, 2016). Nonetheless, empirical evidence suggests that the initial placement of refugees is crucial for the lifetime outcomes of refugees in their new countries (Åslund and Rooth 2007, Åslund and Fredriksson 2009). Given that the initial conditions for placement are crucial to a refugee's success suggests that a greater emphasis should be placed on understanding the factors that could have an impact on finding the right placement for refugees. In recent years, academics have focused on understanding the conditions that improve the "match" between a refugee and their host community, and most importantly developing methods to operationalize the process of improving placement and matching at the level of resettlement agencies (Trapp et al., 2018; Fernandez & Rapoport, 2014; Jones & Teytelboym, 2016; Jones & Teytelboym, 2018).

The following report expands on current research that aims to improve how placement and matching is done by resettlement agencies for the refugees under their oversight. In particular, our research capitalizes on the use of publicly available location-specific data analytics to explore concepts in the optimization of refugee placement and to conduct an in-depth analysis of the US resettlement agency HIAS (founded as the Hebrew Immigrant Aid Society) and their network of affiliate resettlement agencies.

2. Background

The US has long been involved in the resettlement of refugees from various countries throughout the world. Since 1975, the US has resettled over three million refugees into the country, providing these people with an opportunity for freedom from threat and persecution, and integration into the US society (Refugee Council USA, 2018). Even before 1975, the US has played a key role in offering protection for those fleeing persecution and violence. A post-World War II era brought with it the resettlement of thousands of European people, leading to the Displaced Persons Act of 1948, the first act of legislation allowing refugees to become a major contributor to US immigration ("Displaced Persons Act of 1948", 2015). The Cold War era also brought with it an expansion in the bandwidth for the number and type of refugees allowed into the country, as people from Southeast Asia, Russia, Cuba, and various parts of Europe were now accepted into the US and allowed to resettle. With 25.4 million refugees in the world, a number that is consistently rising, the US has remained a major contributor in the resettlement of these populations (Kerwin, 2018). A recent shift in US administration, however, has caused a significant decrease in the number of refugees allowed into the country. Figure 1 shows the trend in the number of refugees admitted into the US between 1975 and 2018.

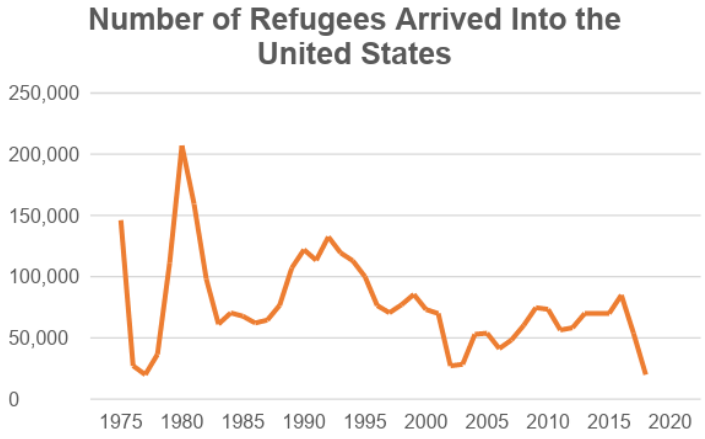


Figure 1: Trend for Refugee Arrivals in the United States

The number of refugee arrivals per year is impacted by the US administration, which sets an annual capacity for the number of refugees allowed to resettle in the US. As of 2018, the annual cap designated by the Executive Office was set to a low of 30,000, more than a 50% decrease from the cap during the Obama administration just a few years prior (Cepla, 2019; Kerwin, 2018).

2.1 Current Process of US Refugee Admission and Resettlement

The process of refugee resettlement in the US is quite complex, as it involves various stakeholders working together on behalf of the refugee and their families. A “refugee” is defined as someone who under US law holds the following qualifications prior to resettlement (Bray, 2016; USCIS, 2018):

- Is located outside of the US
- Is of special humanitarian concern to the US
- Demonstrates that they were persecuted due to race, religion, nationality, political opinion, or membership in a particular social group
- Is not firmly resettled in another country
- Does not violate any major grounds of inadmissibility such as having improper vaccinations or a communicable disease

Refugees are not to be confused with “asylum seekers,” those who seek protection with an official claim that has not been fully processed by the place in which they hope to relocate. An asylum seeker can be someone who is already in the US or someone seeking admission from the outside.

The process of refugee admission into the US begins with a verification that the above “refugee” qualifications are met. To be eligible for the admission process, a refugee must be referred to the US Refugee Admission Program (USRAP) for consideration. Referrals are made to the USRAP by one of the nine Resettlement Support Centers (RSC) around the world. In certain cases, when refugees have relatives in the US, the

referral process can be disregarded. For the majority of people, however, an RSC must create the referral and obtain background information on the refugee to prepare for a security screening. Referrals are designated under three different priorities defined as follows ("USRAP: Application and Case Processing", 2018):

- **Priority 1 (P1):** Cases referred by UNHCR, the US embassy, or an NGO
- **Priority 2 (P2):** Cases involving special humanitarian concern
- **Priority 3 (P3):** Cases involving family reunification

Once a referral and application has been submitted, the refugee is interviewed in their present host country by a member of the United States Citizenship and Immigration Services (USCIS) who determines if the refugee meets all of the necessary requirements. If approved, the RSC then requests a sponsorship assurance from one of the nine resettlement agencies in the US. The nine resettlement agencies in the US are the Church World Service (CWS), Ethiopian Community Development Council (ECDC), Episcopal Migration Ministries (EMM), Hebrew Immigrant Aid Society (HIAS), International Rescue Committee (IRC), Lutheran Immigration and Refugee Service (LIRS), US Committee for Refugees and Immigrants (USCRI), United States Conference of Catholic Bishops (USCCB), and World Relief (WR) (Refugee Council USA, 2018). The resettlement agencies act in coordination with over 300 US affiliates to resettle refugees into locations to facilitate integration specific to each individual or family. Upon approval of the sponsorship assurance, the refugee undergoes a medical examination to evaluate any existing medical conditions and screen for diseases, such as tuberculosis, that need to be treated before entering the US. If medically approved, the refugee is then brought to an RSC where they undergo a cultural orientation to equip them with knowledge, skills, and important information for their first few months in the US (Cultural Orientation Resource, R&P Orientation Curriculum, 2018). Following the cultural orientation, the International Organization for Migration arranges travel to the US and provides the refugee with an interest-free loan to cover for travel expenses. Most refugees pay off their travel loan in installments within 5 years of resettlement in

the US (Kerwin, 2018). Upon entrance into the US, the refugee undergoes a final background check at the port of admission into the country and is left under the responsibility of the designated resettlement agency to help the refugee assimilate socially and economically into the US. Figure 2 below shows a detailed process map of the refugee resettlement process in the US and the various stakeholders involved at each stage.

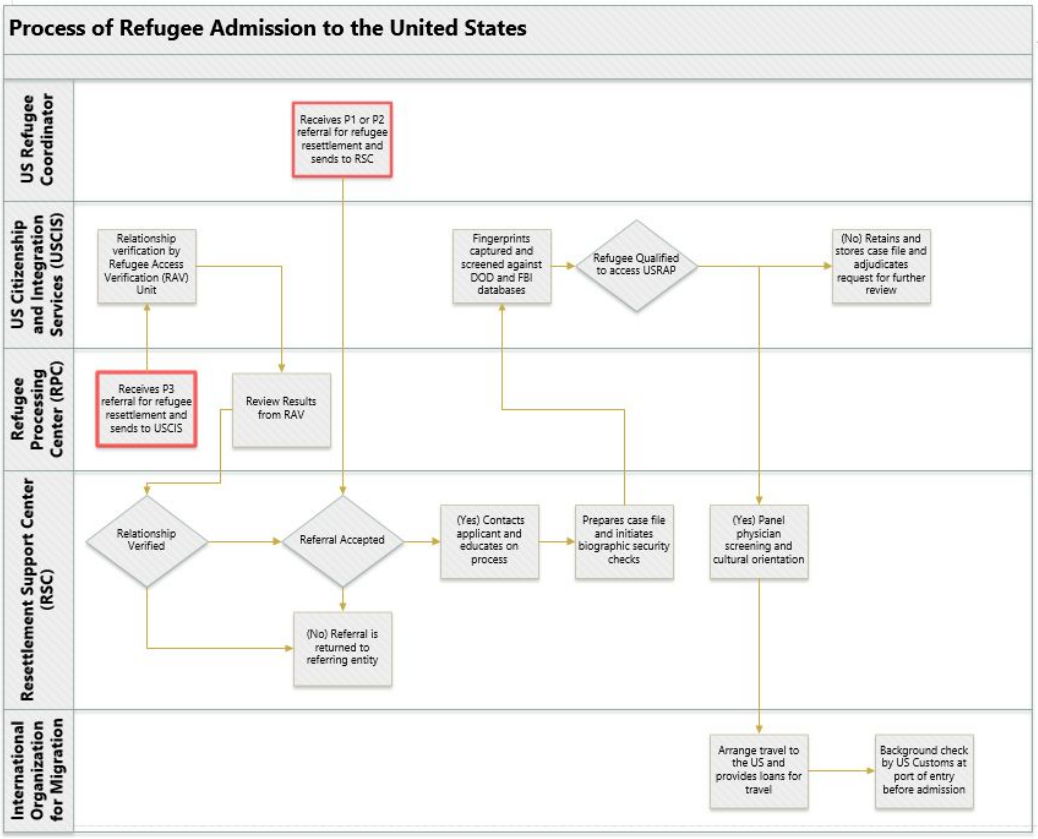


Figure 2: Process Map for Refugee Admission into the US

2.2 HIAS and Affiliated Locations

HIAS is one of nine resettlement agencies in the US that resettles vulnerable populations. HIAS is a non-profit organization founded in 1881 to aid Jewish folk fleeing Russia and Eastern Europe. However, in the early 2000s, HIAS expanded its work to include the resettlement of other persecuted populations such as those from Bosnia, Vietnam, Afghanistan, Iran, and other countries throughout the world. HIAS is the oldest

resettlement agency in the US, celebrating 130 years of “helping refugees escape persecution and resettle safely, reuniting families who have been separated, and helping them build new lives in safety and freedom” (“Our History”, 2018). At the time of this writing, HIAS works in collaboration with 19 affiliate locations to resettle refugee families. These 19 locations include the following cities:

1. San Diego, CA
2. Los Gatos, CA
3. Walnut Creek, CA
4. Wilmington, DE
5. Clearwater, FL
6. Framingham, MA
7. Springfield, MA
8. Ann Arbor, MI
9. Charlotte, NC
10. Buffalo, NY
11. New York, NY
12. White Plains, NY
13. Cleveland Heights, OH
14. Columbus, OH
15. Toledo, OH
16. Philadelphia, PA
17. Pittsburgh, PA
18. Kent, WA
19. Madison, WI

HIAS uses a manual process to place refugees in locations throughout the US. Each week, the HIAS resettlement team meets together to discuss, one by one, each refugee case at hand. A case represents a family of one or more refugees that is in need of relocation in the US (Trapp et al., 2018). Given brief information about each case and the capacities for each affiliate location, the HIAS resettlement team manually selects where each family should be placed (locations are limited to the 19 affiliates in HIAS’s network). While the process by which HIAS does this is somewhat efficient, there is room for improvement in incorporating family-specific and location-specific factors. With the large number of refugee families that need to be considered through this process, it may also be difficult to reconsider a placement decision once it has been made. HIAS may benefit from additional methods to optimize their process of refugee relocation in

the US. Analytical methods used in industrial engineering and operations research may be able to help with this.

2.3 Addressing Refugee Resettlement from Multiple Perspectives

The challenges associated with resettling refugees are neither new to the scene for society at large, nor for academic circles. Naturally, different disciplines of the social sciences have addressed the issue from a variety of perspectives.

Perhaps one of the greatest challenges around the resettlement of refugees is understanding what drives social integration and how to measure it as an indicator of a refugee's successful adaptation to a new community. Social integration is not only crucial for the refugees themselves, but has also been demonstrated to be just as important for local community members who consider it one of the highest priorities in their outlook towards refugees (Tent Foundation, 2017). To propose a normative framework for the concept of social integration, social researchers Alastair Ager and Alison Strang (2008) introduced 10 Domains of Integration[1] as detailed in Figure 3 where domains are categorized into markers and means, social connection, facilitators, and finally the foundation.

[1] The question of what constitutes social integration has been a subject of constant discussion, but many of the themes remain consistent. See for instance the Council of Europe's Measurement and Indicators of Integration Report (1997) or the OECD's Indicators of Immigrant Integration Report (2015).

Figure 3: Domains of Social Integration Reproduced from Ager and Strang (2008)

Understanding what constitutes social integration is a challenge of its own, but perhaps more challenging is understanding how to measure it. In 2016, researchers from the independent firm Quality Evaluation Design (QED) published a longitudinal study in collaboration with the Colorado Office of Economic Security entitled *Refugee Integration Survey and Evaluation* (RISE) Report that explored the practical implications of Ager and Strang's (2008) framework for integration. QED designed a survey that tracked a sample of refugees in the state of Colorado over five years, and in the process, helped operationalize Ager and Strang's Domains of Integration through a variety of metrics, ultimately determining an "Overall Integration Score" (Lichtenstein, Puma, Engelman, & Miller, 2016). The RISE report boasts a 70% retention rate of information about refugees over five years, a number that reflects success but also illustrates the complexity associated with keeping track of refugees once they have arrived in the US. The success achieved by the team behind RISE is largely attributed to a team of community members who lived among refugee populations known as *Community Connectors*. The Community Connectors were crucial in collecting the qualitative information that came directly from the refugees themselves and helped shape the RISE report.

> "*Collecting surveys using a professional model—scheduling appointments, for example, simply wouldn't have worked. The fluidity and mild to extreme chaos of refugees' lives would quickly frustrate the schedules of 9-5 professionals*"
> (Lichtenstein, Puma, Engelman, & Miller, 2016).

The complexity and cost of operationalizing the assessment of social integration after refugees have arrived in their destination country suggests that more emphasis can be placed on an earlier stage of the process of resettlement. To that end, some economists have focused on approaching the problem through the lens of market design and matching theory. Matching markets suggest that both the supply and demand side of the market (our case assumes refugees and hosts) will be matched, but there are conditions that make that match better or worse. In other words, not all matches are equal (Rysman, 2009).

Market design is meant to address actual and potential market failures or improper allocation of goods, which in the context of refugee resettlement refers to the proper allocation of refugees among host communities (Duke Kominers, Teytelboym, & Crawford, 2017). Particularly for the European continent, some academics address the possibilities for a Tradable Immigration Quotas system, where the number of refugees taken in by European nations can be traded in a controlled market between said countries (Fernandez & Rapoport, 2014). Some early concerns about the quota system suggested that the market would be flawed if preferences of refugees were excluded from the model or if ultimately the market ended up negatively affecting smaller, less powerful countries (Kousmanen, 2012). Nonetheless, the possibility of expanding the market through matching theory has also been addressed, where the preferences of both refugees and localities alike could be taken into account (Jones & Teytelboym, 2016; Jones & Teytelboym, 2018).

Once a refugee has been cleared to resettle, the case passes through a governing organization that makes the decision of where to resettle the refugee. In some European countries, the placement of refugees to specific localities is vetted by

individual countries' local governments. In the US, however, the process of locality selection is left at the discretion of resettlement agencies with limited oversight by the state or federal government (van Selm, 2003). Every week the nine national resettlement agencies in the US meet to determine which agencies will resettle which refugees (Holder, 2018). Employees at the resettlement agencies will generally make a decision of where to place a refugee among their agencies' network locations through a heuristic approach represented in Figure 4, which is based on three main components of information about the refugee and the agency itself.

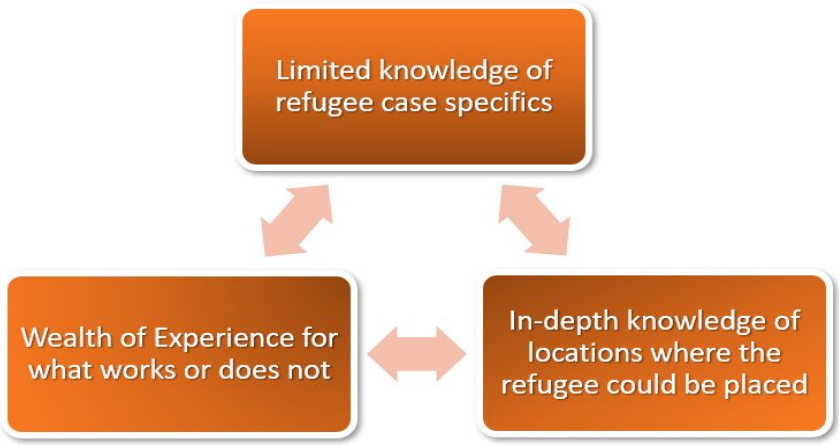


Figure 4: The Heuristic for Placement Decision at US Resettlement Agencies.

Many other factors are taken into account when deciding on placement, including nationality, gender, and languages spoken. The resettlement agency workers also gain a comprehensive understanding of the refugees' needs, as well as the resources available in each city such as hospitals, housing, and mental health support. Though the process is a natural space for human decision, the complexity of information also means that there is room for improvement to fill the gaps where human decision leads to inefficiencies (Trapp et al., 2018). For example, one of the inefficiencies presented by the current method of refugee placement is the sequential nature of decisions, versus a holistic consideration to make a decision. In other words, employees at resettlement agencies will consider the incoming refugees in a case-by-case basis as they arrive, as opposed to making simultaneous decisions on the entire pool of refugees coming

through the agency over a period of time. The process also leads to inefficiencies by leaving the decision exposed to biases in cognition and decision making on behalf of the decision maker, a subject that has been widely studied over the years by social scientists, particularly in the domain of psychology[2].

This section explored some of the perspectives that academic research has provided into the study of improving refugee resettlement particularly from the areas of the social sciences. The overarching subjects of social integration, market design, matching markets, and decision making are explored in more detail within the context of HIAS in following sections.

2.4 Factors Affecting Refugee Employment and Income

One's ability to make a living can be affected by a variety of individual, economic, and governmental factors. These factors include an individual's education level and previous employment experience, gender, relevant skills and proficiencies, government legislation, and even current market conditions ("Utah State Board of Education", 2008). Since the US resettles refugees from various countries throughout the world, it is likely that refugees differ in their background, skill sets, and overall ability to succeed in the US workforce. These differences between refugees make it both interesting and important to understand the factors that contribute to whether or not one can easily obtain employment, and thus maintain a steady income, in the US.

Previous Employment and Experiences

One of the greatest determinants of future employment, and ultimately income, is whether or not an individual has relevant employment experiences that are suitable for the current market. According to authors Gina Dokko, Steffanie Wilk, and Nancy Rothbard, "Employers hire on the basis of work experience because they expect experienced workers to perform better. Prior experience is often used by employers as

[2] See for instance the Availability Bias proposed by Amos Tversky and Daniel Kahneman.

an expedient proxy for the knowledge and skill that contributes to performance" (Dokko, Wilk & Rothbard, 2009). Although the previous employment experiences of refugees differ depending on where they are from, Table 1 below depicts some that are more common before entrance into the US (Ahani, 2018).

Table 1: Some Examples of Previous Occupations and Skills

Previous Employment & Skills of Refugees before Entering US	
Accounting	Mechanist
Baker/Cook	Nurse
Cashier	Pathologist
Construction	Project Manager
Electrician/Engineer	Salesperson
Farming/Fishing	Secretary
Housekeeping	Teacher
Linguist	Trader
Interpreter	Web developer
Manicurist	Welder

It is evident that there is a wide range of variability between the experiences and skills of refugees before entering the US. In some cases, refugees have adequate experiences in more complex subjects such as engineering and management, while in other cases, refugees have experiences with more basic skills. For a complete list of previous refugee employment and skills, reference Appendix G.

Education

Education level also impacts one's employment status and income. According to the Organization for Economic Cooperation and Development (OECD), "In general, people

with higher levels of education have better job prospects; the difference is particularly marked between those who have attained upper secondary education and those who have not" (OECD, 2012). Upon entering the US, refugees are currently classified into nine levels: kindergarten, primary, intermediate, secondary, technical school, pre-university, university, professional, and graduate school (Ahani, 2018). In general, refugees have very limited access to education prior to coming to the US. For example, roughly half of the refugees who are of schooling-age, approximately 3.5 million people, do not receive any education in their home country. Overall, only about 61% of refugee children attend primary school. Additionally, only about 23% of refugee adolescents and 9% of refugee adolescents in low-income countries attend secondary school. Only about 1% enroll in a college or university ("USA for UNHCR: The UN Refugee Agency", 2018). This demonstrates how limited access to education is a commonality amongst refugee populations.

English Proficiency

The ability to speak English is something US natives often take for granted when applying for domestic jobs. However, as a refugee, one's ability to speak English significantly impacts the ability to obtain employment in the US. According to a paper written for the US Census Bureau, "Employers may avoid hiring otherwise qualified individuals who have difficulty communicating effectively. People who have difficulty with English may [even] feel uncomfortable applying for some jobs that require proficiency" (Day & Shin, 2005). A study completed on behalf of the US Census Bureau also analyzed whether or not English-speaking ability actually affects employment status, work status, and earnings. The results showed that people who do not speak English at home have a smaller chance of obtaining employment, a smaller chance of finding full-time work when employed, and experience lower median earnings than English-speaking people. The largest gap in earnings existed between the "very well"

and “well” speakers, with “very well” speakers earning on average around $7,000 more than the latter (Day & Shin, 2005).

There are currently 133 languages represented amongst the refugee population in the US. When a refugee enters the US, language proficiency levels are measured in three dimensions; reading, speaking, and writing. For each dimension, a refugee is assigned a score of either “Good”, “Some”, “None”, and “Unknown”. If an individual in a family can speak “Good” or “Some” in a language, it is considered that they know this language. Furthermore, ability to speak English specifically is measured on a binary scale of 1 or 0, with 1 representing “Yes” and 0 representing “No” (Ahani, 2018).

Gender

Although hiring on the basis of gender is still an ongoing gender discrimination issue, gender continues to prominently play a role in the gap between men and women salaries. The gender pay gap is a statistical indicator used to determine the status of a woman’s earnings relative to a man’s. It is calculated by dividing the median annual earnings of women by the median annual earnings of men (“National Women's Law Center”, 2018). Although this pay gap has been decreasing at a very slow rate in the last 30 years, it has always been prevalent in our society. In as recent as 2017, women earned on average around 82% of what men earned (Graf, Brown, & Patten, 2018). Although in some cases women have higher salaries than men, it is estimated that women as a whole will not receive equal pay until the year 2059 (“National Committee on Pay Equity”, 2018).

2.5 – Using Analytics in Refugee Resettlement

Analytics have recently been used for refugee resettlement in the US, specifically by HIAS as well as other resettlement agencies.

Annie

When HIAS receives resettlement cases, resettlement staff review the cases one at a time and make the resettlement decisions (Trapp et al., 2018). However, as stated in previous sections, there may be better ways for placing refugees. A new tool has been developed, entitled *Annie*, to aid with the refugee matching problem by creating a predictive model to estimate the likelihood of employment for each employable refugee. *Annie* then works to determine the optimal match for which cities to place refugees in by optimizing the total expected number of employed refugees. It uses integer optimization methods and looks at the whole picture by considering all cases at once. *Annie* considers the same factors such as language and nationality, and also accounts for the available capacity of each city. The solution output from *Annie* is then used as a recommendation and reviewed by HIAS resettlement staff, who can visually interact with the recommendation to arrive at the ultimate decision.

Although this software has been effective in improving refugee-location matches in the US, it still has various shortcomings, including that it does not take into consideration location factors when resettling.

Other Optimization Models Used in Refugee Resettlement

Fernandez-Huertas Moraga and Rapoport (2014) present an optimization model that minimizes total cost for a given number of refugees or maximizes the number of refugees for a given budget constraint. In addition to cost minimization, the model provides a framework for considering refugee preferences and host preferences, such as language, skills, and country of origin, over refugees' legal status, such as asylum seekers waiting for a decision, refugees whose asylum requests have been accepted, or internationally resettled refugees. The optimal solution to the model equalizes the marginal net cost of hosting one additional refugee or asylum seeker across hosts. Refugees would be hosted where it is 'cheapest' to host them from a receiving host's point of view.

Under the tradable refugee quota market proposed, each host is assigned an initial quota that can be filled with both refugees and asylum seekers (Fernandez-Huertas Moraga & Rapoport, 2014). These quotas can be traded in a market where there is a price received for accepting one additional refugee or asylum seeker in excess to the assigned quota, and a price paid for accepting fewer refugees or asylum seekers than the quota. The hosts would theoretically reach an equilibrium quota price where relative cost between hosts is minimized.

Bansak et al. (2018) used machine learning to create a data-driven algorithm to allocate refugees across affiliate locations to optimize integration for one of the largest resettlement agencies in the US. The algorithm combines supervised machine learning with optimal matching to identify relationships between refugee data and location-based data, and ultimately increase expected employment for refugees. The results of their algorithm increased the expected probability of employment for refugees in the US by about 40%, and also lead to overall higher employment rates in almost every resettlement location.

Trapp et al. (2018) has extended upon the work of Bansak et al. by using an approach that seeks to maximize economic benefit. This research proposes training a regression model that outputs the probability that a refugee will be employed within 90 days of arrival. The model's output is then used in an optimization model which matches groups of refugees with locations that maximize the refugees' combined probability of employment. This research has resulted in a tool that is currently in operational use by resettlement agencies.

These studies show the promising potential that analytics have on improving refugee resettlement. They serve as a basis for using optimization and machine learning to improve refugee resettlement, either by maximizing the number or refugees resettled or maximizing refugee employment outcomes. There are several other analytical

techniques that have the potential to be used to improve refugee resettlement, which are outlined below.

Unsupervised Learning

Unsupervised learning is a branch of machine learning that is concerned with discovering relationships within data (Gareth et al., 2013). It is often performed as part of exploratory data analysis and its goal is to find patterns in data that lead to better ways of visualizing and grouping the data. To perform an in-depth analysis of the HIAS network and identify a recommended set of locations that would complement this network, our team focused on two unsupervised learning algorithms, Clustering and Principal Components Analysis.

Clustering

Clustering is a set of different techniques that are used to discover subgroups within data (Schubert, 2017). The goal is for every group to be composed of data that is more similar to each other than the data that is in a different group. The similarity measure depends both on the knowledge domain that the data belongs to and the algorithm that is being used to cluster the data.

Principal Components Analysis

Principal Components Analysis is another type of unsupervised learning that tries to simplify data by reducing the set of variables that “explain” the data (Jolliffe, 2002). It achieves this by creating linear combinations (components) of the variables that collectively explain the most variability in the data. Once the linear components are computed, the reduced set of components that have the highest sum of Proportion Variance Explained (PVE) are selected as the new set of representative variables (Lorenzo-Seva,2013).

3. Methodology

Our goal was to assist HIAS with improving refugee resettlement in the US through the use of analytics. To accomplish this, we focused on three main objectives:

1. Perform a proof of concept for optimizing refugee placement through an analysis of refugee-specific and location-specific data.

2. Analyze and find areas of improvement for HIAS's network through an analysis of location-specific data relative to other locations in the US.

3. Improve the accuracy of the predictive modeling within *Annie* by developing scripts that extract location-specific economic data.

We employed various industrial engineering and data science techniques, including optimization, data analysis, and statistical analysis to evaluate refugee resettlement in the US, specifically with HIAS.

3.1 Proof of Concept: Optimization in Refugee Resettlement

An area of improvement in data-driven refugee resettlement is the inclusion of both location-based factors and refugee-specific factors to determine where a refugee should be placed. By considering both individual and location-based factors, we developed an optimization model that maximizes a refugee's projected income and places them accordingly in one of the 19 affiliate locations. Before discussing the details of our model, we first outline some important assumptions and considerations.

Assumptions

- Refugee data provides accurate information on gender, English proficiency, education level, and medical conditions. Every refugee has a known skill or previous employment experience that is reflected in our refugee data.

- The capacities of each affiliate location can be adjusted. Current values are based on the number of 2017 cases resettled in HIAS affiliate locations.
- Factors such as education, English proficiency, and gender independently affect income.
- Social adjustment can be quantified, and plays a role in placement decisions.
- Because we are discounting our income by various individual and location-specific factors in our model, we multiplied the expected income and social adjustment score in our objective function. It is also reasonable to multiply in this case, since we are discounting our baseline income by various factors represented by ratios.
- Factors such as the percent of non-English speaking people in a location and the number of physicians per 100,000 people of a location are adequate measures to determine whether or not a refugee will adjust appropriately to a specific location.
- Each skill or previous employment experience falls under an appropriate industry category outlined from DataUSA.io, a source used to pull data from locations throughout the US.
- When determining tax brackets, the marital status of all refugees was defined to be “single.”

We created this list of assumptions to work around our limited access to refugee information and data pertaining to refugee resettlement. We reached out to Narges Ahani, presently a data science PhD student at Worcester Polytechnic Institute, to obtain a list of common employment experiences and skills of refugees before resettlement. We also obtained a list of example attributes that are documented about each refugee, including information such as education level, proficiency level, case number, and gender. While it is realistic to have this information, it may not be realistic to have zero gaps in data. For example, we assumed all refugees had a previous employment experience or skill, which might not be the case in the real world. Because our model was based off of income levels (and thus, employment), it was important to assume various attributes about each person were known fully. Additionally, it was

justifiable to use the percent of non-English speaking people and physicians per 100,000 people as two measures of a location's adequacy for refugee resettlement, as medical conditions and language barriers are common refugee challenges.

Optimization Model

Our optimization model served as an exercise to understand what factors affect the potential income of a refugee, and how this income differs by location and the characteristics of each individual refugee. To build our optimization model, we used Microsoft Excel and an add-in called OpenSolver.

OpenSolver can be used to solve both linear and non-linear optimization models ("OpenSolver for Excel", 2019). OpenSolver is ideal for its ease of use, as well as its practicality in solving models with a large number of decision variables. Like any optimization solver, OpenSolver requires the input of decision variables, constraints, and an objective function. Other notable features of OpenSolver include a built-in visualizer that highlights the model's decision variables, objective function and constraints, a QuickSolve mode that makes it faster to re-solve the model after making changes, and OpenSolver's ability to solve a model of unlimited size. We decided to use this software primarily because of its ease of use, as well as its practicality in solving models with a large number of decision variables.

To begin, we generated our own set of test data that is reflective of the information that is collected about each refugee. Descriptions of the information available is shown in Table 2 (Ahani, 2018).

Table 2: Available Features of Refugees

Refugee Information	Description
Case Number	An anonymized, unique identifier for each family
Sequence	The sequence of numbers represented in a family, with "1" meaning person #1, "2" meaning person #2, etc.
Case Size	Family size
Relationship Code	The relationship to the principal applicant for each individual in a family; these include Principal Applicant (PA), Husband (HU), Wife (WI), Daughter (DA), Son (SO), Stepdaughter (SD), and Stepson (SN)
Gender Code	Gender of the refugee (Male or Female)
Nationality Code	Code for each nationality (there are 33 nationalities represented)
English Speaking	Whether or not the refugee speaks English (1 or 0)
DOB	Date of Birth
Education Level	Levels include kindergarten, primary, intermediate, secondary, technical school, pre-university, university, professional, and graduate school
Medical Condition	Whether or not the refugee has a medical condition (1 or 0)
Urgency Code	How fast the case must be assured by the resettlement agency. Values include both normal and expedited
Language	Proficiency levels for reading, speaking and writing (levels include Good, Some, None, Unknown). There are 133 languages represented. If an individual in a family is classified as "Good" or "Some" in one language, then we consider them to know that language.
Case Number To	Case number of a family that it is cross-referenced to
Affiliate To	Affiliate that the cross-referenced family is already assigned to
Previous Employment/Skills	Known skills, interests, or previous employment experiences of a refugee

We then converted each category into a quantifiable number if possible. For example, we changed DOB to represent Age, education level to be assigned a value between 1 and 9, with 1 being kindergarten and 9 being graduate school. We represented proficiency levels on a scale of 1 to 4 with 1 representing "None" and 4 representing the "Great".

We also gathered location-specific information from a data source called DataUSA.io. We retrieved numerous location-based indicators for the 19 affiliate locations. Descriptions of a subset of these indicators is shown below in Table 3. A full list of indicators is given in Appendix A.

Table 3: Location-Based Indicators (Adapted from DataUSA.io)

Indicator	Description
Median Income of Employment Industry by Gender	The median income for both males and females in each employment industry within each affiliate location
Physicians per 100,000 people	The number of physicians per 100,000 people in an affiliate location
Percent of Non-English Speaking People	Percent of total population in an affiliate location that does not speak English
Tax Rates	Tax rate per affiliate location
Median Salary per English Speaking Level per Age	The median income for each level of English proficiency broken down by age in each affiliate location

We then determined which refugee-specific and location-based factors might influence income the most. Given our prior research on aspects affecting income, we identified that previous employment or experiences, gender, English-proficiency, and education level were four of the most important factors that affect employment probability and yearly earnings. Our next task was to determine the extent to which each of these factors impact income and develop a method to quantify these relationships.

Previous Employment and Experiences

From DataUSA.io we gathered data on the median income for both males and females in 34 different industries across the 19 affiliate locations. These 34 industries included those such as administrative work, community and social service, management, material moving and transportation (see Appendix H for full list). We then matched a refugee's previous experience or skill set to one of the 34 industries. For example, a

refugee who has “driving skills” would be listed under the transportation industry. We were then able to determine a value for this refugee’s median income in each of the 19 locations based on their industry match and gender. For example, a male refugee who falls under the transportation industry would make $32,192 in Buffalo, NY. This value acted as the refugee’s baseline income in a specific affiliate location before any additional factors were taken into account.

Gender

To determine the gender gap per industry, we calculated the ratio of the female to male salary within each employment industry for every affiliate location. For example, in Buffalo, NY a male in the transportation industry makes on average $32,192. A female from Buffalo, NY in the transportation industry makes on average $21,323. Therefore, the gender gap ratio is as follows:

$$\frac{\$21,323}{\$32,192} = 0.662$$

We established a rule that if the refugee is male, his baseline income would not be affected by his gender. Thus, every male refugee was assigned a score of 1 to represent how gender is not a discount factor for the income of men. However, if the refugee was female, the baseline income would be multiplied by the calculated gender gap to represent the reality of women being paid less than men. In certain industries, women had a higher income than men so our gender gap ratio was greater than one. In these cases, we would again multiply the baseline income by the gender gap ratio.

English Proficiency

To determine English proficiency, we calculated the average score across an individual’s proficiency in reading, writing, and speaking. From “The Earnings of Immigrants in the US: The Effect of English-Speaking Ability” written by Park in The American Journal of Economics and Sociology, we were able to understand how

English proficiency impacts income on a quantifiable scale (Park, 1999). This source included a breakdown of income per age bracket for people of various English proficiencies, which we used to calculate the English proficiency ratio. This ratio is calculated by dividing the median salary based on the refugee's age and level of English proficiency by the median salary for a person with the maximum English proficiency in the same age bracket. From this information, we developed the following table of ratios that would also be multiplied by a refugee's baseline income.

Table 4: English Proficiency Scoring (Reprinted from The Earnings of Immigrants in the United States: The Effect of English Speaking Ability, by Jin Heum Park, 1999)

Score	25-34 years	35-44 years	45-54 years	55-64 years	65 or older
Great	1	1	1	1	1
Good	0.811	0.782	0.793	0.796	0.815
Some	0.637	0.590	0.060	0.596	0.626
None/Unknown	0.519	0.457	0.447	0.446	0.490

If the average of the reading, writing, and speaking score was less than 1, the refugee was assigned the ratio in the "None/Unknown" category of their age group. If the average was at least 1 but less than 2, the refugee was assigned the ratio in the "Some" category of their age group. If the average was at least 2 but less than 3, the refugee was assigned the ratio in the "Good" category of their age group. If the average was at least 3, the refugee was assigned the ratio in the "Great" category of their age group. This ratio would be the same in every location and would be multiplied by the baseline income to represent how anything below a "Great" English proficiency can affect yearly earnings. For example, a person age 32 with an average proficiency of 1.4 would qualify as having "Some" proficiency and his or her income would be impacted by about a 36% decrease (1-0.635).

Education Level

To measure how education level impacts income, we referenced the "Current Population Survey (CPS): Historical Time Series Tables" filled with educational attainment data from the US Census Bureau (US Census Bureau, 2019). The latest available information was from 2016. This data included information on the mean earnings of workers 18 years and older by educational attainment, race, Hispanic origin, and sex. It defined the mean earnings overall for both men and women and showed the percentage of this mean that a person would make depending on his or her education level. The education levels represented in this data included below a high school level, completing only a high school diploma, completing some college/associate's degree, completing a bachelor's degree, and completing an advanced degree. Table 5 below summarizes these percentages as ratios, and represents how much an income would be affected depending on one's level of education. For example, a person that has below a high school education level makes, on average, 53.6% of the mean salary within their industry.

Table 5: Education Level Earnings (Reprinted from CPS: Historical Time Series Tables, by the US Census Bureau, 2019)

Education Level	Percent of Mean Earnings (2016)
Not a High School Graduate	53.6%
High School Graduate	70.7%
Some College / Associate's Degree	77.5%
Bachelor's Degree	129.6%
Advanced Degree	183.5%

Since these education levels are not the same as the ones listed in our refugee data, we reorganized these numbers depending on how the nine levels matched up to the levels above. See Table 6.

Table 6: Refuge Data to Education Level (Adapted from CPS: Historical Time Series Tables, by the US Census Bureau, 2019)

Refugee Data: Education Level	Education Level (Bureau of Labor Statistics)	Percent of Mean Earnings (2016)
(1)Kindergarten	Not a High School Graduate	53.6%
(2)Primary	Not a High School Graduate	53.6%
(3) Intermediate	Not a High School Graduate	53.6%
(4) Secondary	High School Graduate	70.7%
(5) Technical School	High School Graduate	70.7%
(6) Pre-university	Some College	77.5%
(7) University	Bachelor's Degree	129.6%
(8) Professional	Advanced Degree	183.5%
(9) Graduate School	Advanced Degree	183.5%

We assigned a refugee a ratio based on their level of education and multiplied this ratio by the baseline income to represent how their income is impacted by their education level.

Location-Specific Indicators

Although refugee-specific factors such as gender, English-proficiency, education level, and previous employment experiences have a tremendous impact on yearly earnings, it is also important to consider how location-specific indicators play a role. Our model incorporated a social adjustment score which represented location-specific indicators and how these affect one's income. We first identified what aspects of a given location would make it difficult to live there. For example, we recognized that individuals with medical conditions would likely have a more difficult time resettling in locations that do not have sufficient medical resources. We also recognized that people with lower English-proficiency levels might have an easier time resettling in locations with a higher percentage of non-English speaking people. Therefore, the social adjustment score was

based on two location-based indicators; physicians per 100,000 people in an affiliate location, and the percent of non-English speaking people per affiliate location.

Physicians per 100,000 People

From DataUSA.io, we obtained the number of physicians per 100,000 people in an affiliate location. See Table 7 below for the full list of information we gathered.

Table 7: Physician Data (Adapted from DataUSA.io)

Affiliate Location	Physicians per 100,000 people	Percent Rank	Discount Ratio
Springfield, MA	70	0.95	0.7507
Philadelphia, PA	70	0.9	0.7784
Los Gatos, CA	77	0.85	0.8061
San Diego, CA	79	0.8	0.8338
Buffalo, NY	80	0.75	0.8615
Wilmington, DE	83	0.7	0.8892
Charlotte, NC	86	0.65	0.9169
Clearwater, FL	90	0.60	0.9446
Toledo, OH	90	0.55	0.9723
Walnut Creek, CA	98	0.50	1
Pittsburgh, PA	109	0.4	1.0277
Cleveland, OH	111	0.4	1.0554
Kent, WA	119	0.35	1.0831
Framingham, MA	122	0.3	1.1108
Madison, WI	126	0.25	1.1385
Columbus, OH	130	0.2	1.1662
New York, NY	138	0.15	1.1939
Fairview, Westchester County, NY	139	0.05	1.2216

Ann Arbor, MI	173	0.05	1.2493

We calculated the minimum value, maximum value, and 50th percentile to be 70, 173, and 98 respectively. We then calculated the percent rank for each location, ranking the location with the highest number of physicians as the best case because a location with more physicians is more ideal for a person with medical needs. We decided to score each affiliate location between 0.75 and 1.25, with 1.25 being the best case scenario. We chose this range because we did not want our discount factor to be so low that it would be too close to 0. We also did not want it to be so large that it would impact the baseline income dramatically. Values on the right hand side of Table 7 show these calculated discount values. We then assigned a score to each individual in each affiliate location by first referencing the test data to determine whether or not the refugee had a medical condition. If the refugee had a medical condition (a score of 1) and the affiliate location had a number of physicians below the 50th percentile, we assigned the discount ratio of that location. If the refugee had no medical problems, they automatically received a score of 1 for each affiliate location, implying that refugee's income would not be impacted if placed there.

Percent of Non-English Speaking People

From DataUSA we also obtained the percent of the population that does not speak English in each affiliate location. See Table 8 below for these percentages. Locations are ranked from smallest ratio to highest ratio.

Table 8: Non-English Speaking Data (Adapted from DataUSA.io)

Affiliate Location	Percent of the population that does not speak English	Percent Rank	Discount Ratio
Columbus, OH	0.005	0.95	0.7507
Toledo, OH	0.008	0.9	0.7784
Pittsburgh, PA	0.008	0.85	0.8061
Ann Arbor, MI	0.014	0.8	0.8338

Buffalo, NY	0.015	0.75	0.8615
Cleveland, OH	0.018	0.7	0.8892
Madison, WI	0.02	0.65	0.9169
Wilmington, DE	0.024	0.60	0.9446
Clearwater, FL	0.026	0.55	0.9723
Framingham, MA	0.04	0.50	1
Kent, WA	0.048	0.4	1.0277
Charlotte, NC	0.048	0.4	1.0554
Springfield, MA	0.055	0.35	1.0831
Philadelphia, PA	0.056	0.3	1.1108
Fairview, Westchester County, NY	0.064	0.25	1.1385
Walnut Creek, CA	0.065	0.2	1.1662
San Diego, CA	0.079	0.15	1.1939
New York, NY	0.096	0.1	1.2216
Los Gatos, CA	0.118	0.05	1.2493

From here, we calculated the minimum value, maximum value, and 50th percentile to be 0.005, 0.04, and 0.118 respectively, and then calculated the percent rank for each location. The location with the highest percentage of non-English speaking people was ranked as the best case scenario since it is more ideal for people who are not proficient in English to be placed in a location with other non-English speaking people. To stay consistent, our score for each affiliate location fell between 0.75 and 1.25 because we did not want our discount factor to be too low or too high. Values on the right hand side of Table 8 show these calculated discount values. We then assigned a score to each individual in each location by first referencing the test data to determine the average proficiency score of each refugee across English reading, writing, and speaking. If the refugee had an average proficiency score below 2 and the affiliate location was above

the 50th percentile for the percent of non-English speakers, we assigned the discount ratio for that location. If the refugee had an average proficiency above 2, he or she automatically received a score of 1.

We multiplied the scores for the percent of non-English speaking people and the population per physician together to create one social adjustment score per person per affiliate location. We then multiplied this value by the baseline income. To summarize our model, we multiplied scores from the refugee-specific factors by our baseline income to represent how income can be discounted by various person-specific factors. We then multiplied this newfound income by the social adjustment score to represent how location-based factors can also discount one's income.

This method only accounted for individual refugees, not families. Therefore, to account for family cases, we found the average of the final incomes for each member of the family. We compressed our 100 individuals into 40 families, and placed each family in a location that maximized their average income.

Capacities

We obtained estimates from Narges on the capacities of each affiliate location to resettle both individuals and families. The raw data we obtained was for a total of 329 families. Since our model accounted for only 40 families, we scaled down the given capacities to meet our needs by calculating the percentage of the total capacity for each affiliate location, and then multiplying by 40. We then rounded up to the nearest whole number. See Table 9 below for reference.

Table 9: Capacities

Affiliate Location	Initial Case Capacity	Percent of Total (%)	Scaled down Capacity
Los Gatos, CA	1	0.304 %	0.12
San Diego, CA	7	2.13 %	0.85
Walnut Creek, CA	14	4.26 %	1.70

Wilmington, DE	6	1.82 %	0.73
Clearwater, FL	45	13.68 %	5.47
Springfield, MA	21	6.38 %	2.55
Ann Arbor, MI	37	11.25 %	4.50
Charlotte, NC	33	10.03 %	4.01
Buffalo, NY	28	8.51 %	3.40
New York, NY	1	0.304 %	0.12
Cleveland, OH	38	11.55 %	4.62
Columbus, OH	15	4.56 %	1.82
Toledo, OH	15	4.56 %	1.82
Philadelphia, PA	21	6.38 %	2.55
Pittsburgh, PA	21	6.38 %	2.55
Kent, WA	5	1.52 %	0.61
Madison, WI	7	2.13 %	0.85
Framingham, MA	7	2.13 %	0.85
Westchester County, NY	1	0.304 %	0.12
Sum	329		

Algebraic Formulation of Model

Each refugee-specific and location-based metric discussed above is represented by a matrix in our model. These matrices have affiliate locations represented on the rows and refugee individuals represented on the columns. Our optimization function considers two main areas: the income and the social adjustment score. The income per family per location is reduced by the location's tax rate. Maximizing the sum product of income subtracted by tax, the social adjustment score, and the decision variables is the objective function of our model. Our constraints include the fact that each family can be relocated to only one affiliate, and each location has a certain capacity that can be filled. The algebraic formulation of our model is presented on the following page.

Set definition:

$$Let\ I\ be\ the\ set\ of\ families\ with\ index\ i$$

$$Let\ J\ be\ the\ set\ of\ locations\ with\ index\ j$$

Parameter definition:

$$Let\ s_{ij}\ be\ the\ social\ adjustment\ score\ of\ family\ i\ in\ location\ j$$

$$Let\ m_{ij}\ be\ the\ expected\ income\ of\ a\ family\ i\ in\ location\ j$$

$$Let\ c_j\ be\ the\ capacity\ of\ a\ location\ j\ to\ accommodate\ refugee\ families$$

Variable definition:

$$Let\ x_{ij}\ be\ the\ decision\ of\ whether\ or\ not\ to\ place\ family\ i\ in\ location\ j$$

With this notation, we formulate the problem of allocating refugee families to locations, as follows:

$$max \quad \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} (s_{ij} * m_{ij})\, x_{ij}$$

$$s.t.$$

$$\sum_{j=1}^{|J|} x_{ij} \leq 1 \qquad \forall\ i \in I$$

$$\sum_{i=1}^{|I|} x_{ij} \leq c_j \qquad \forall\ j \in J$$

$$x_{ij} \in \{0, 1\} \quad \forall\ i \in I,\ \forall\ j \in J$$

3.2 Perform an Analysis of the HIAS Network

While the optimization model presented in the previous section evaluates the possibilities of resettling refugees within HIAS's current network of affiliates, we explored the use of analytics to describe its current state and identify possible expansions to the HIAS network. Our analysis is driven by comparing the 19 affiliate

locations with the greater context of the US at a county level of key socioeconomic metrics.

DataUSA.io is a co-development between Deloitte, Datawheel, and the Collective Learning Group that serves as a data collection and visualization engine of up to date, publicly available US Government Data. As an online source that compiles information from the Bureau of Labor Statistics, the United States Census Bureau, and other official sources of information, DataUSA served as the primary source of data that informed our analysis of HIAS's network and beyond. Using DataUSA's API, we extracted as much data as needed from its database, which would later be restructured using python libraries to accommodate our analysis needs.

Our analysis is informed by the data extracted at a county level from the information available through DataUSA. We selected specific socioeconomic indicators available and collected throughout the 3,142 counties and census areas in the US. The complete data set of counties in the entire country would help us to understand the performance of each of the 19 counties within the HIAS network compared to all counties in the US. For our network analysis exercise, we selected a set of indicators falling within five main categories: economic, health and safety, education, diversity, and housing and living. The full list of indicators and their descriptions extracted from the DataUSA database are summarized in Table 10.

Table 10: Full list of 20 socioeconomic indicators extracted at a county level (Adapted from DataUSA)

Category	Indicator	Description
Economy	Median Household Income	The median income of household in the area in USD ($)
	% Unemployment	Percent of the population in an area that is unemployed
	% Income Below Poverty	Percent of the population whose income lies below the poverty line
Health and Safety	% Uninsured	Percent of the population without healthcare insurance coverage
	Primary Care Physicians	Ratio of primary care physicians for every 100,000 residents
	Mental Health Providers	Ratio of mental health providers for every 100,000 residents
	Other Primary Care Providers	Ratio of other primary care providers for every 100,000 residents
	Homicide Rate	Ratio of deaths by homicide for every 100,000 residents
	Violent Crime	Ratio of violent crimes for every 100,000 people
Education	% High School Graduation	Percent of the population that graduated from High School level education
	% with Some College Education	Percent of the population that has attained at least some level of college education
Diversity	Age	Average age of residents in the area
	% Non-US Citizens	Percent of residents who are not American citizens
	% Pop. Not Proficient in English	Percent of the population not proficient in English
Housing and Living	Mean Commute Minutes	The average commute time for residents in the area
	% Home Ownership	Percent of the population that owns their current home
	Median Property Value	The median value of residential property in USD ($)
	Income Inequality Ratio	The ratio comparison of the income perceived by the lower 80th percentile compared to that of the top 20th percentile
	Food Insecurity	Percent of the population without a stable and reliable source of food
	Social Associations	Ratio of social associations for every 100,000 residents

It is important to note that we chose to extract only the twenty indicators outlined in Table 10 as we believe that these are the most relevant to the integration of refugees informed by the Ten Domains of Integration framework proposed by Ager and Strang (2008) and introduced earlier in Figure 3. Additionally, for our analysis we considered data between the years 2013 and 2017, and for some indicators, data was not available for each of the five years.

After considering all twenty indicators, we were able to derive a set of six scores for each of the 3,142 counties in the US: a score for each of the five categories based on the attributes within them and one overarching score that considers all twenty

indicators. The score is representative of the percentile rank for each indicator within the larger set of the US.

Table 11: Sample for County Score Calculation

Location: Middlesex County, MA

Year: 2015

Category: Education

Indicator	Raw Data	% Rank
% of High School Graduates	89.50%	72.30%
% of Pop. With Some College	78.20%	97.40%
County Category Score		**84.85%**

Consider the example in Table 11. In 2015, Middlesex County in Massachusetts reported an 89.5% high school graduation rate and 78.2% of the population had some college education, which placed it at the 72.30% percentile rank and 97.40% percentile rank, respectively, among all counties in the US for that year. By taking the average of Middlesex county's rank in both indicators we conclude that in 2015, it received a score of 84.85% in the education category.

Similar to the categories score, each county received an overall score by year, which is an average of the percentile rank that the county received for all twenty indicators. The yearly score for each county was pivotal to our data analysis and presentation described in our results and analysis chapter. The twenty indicators served as a good measure to present an overall score that encompasses a comprehensive set of indicators, which are arguably crucial to a refugee's integration. It is important to note the assumption that we used our own judgment to determine what merited a high ranking or a low ranking. For instance, with indicators such as median property value or percent of population not proficient in English, we assumed that higher values merit a higher rank. A higher percent of the population that is not proficient in English might signify higher diversity, which could be favorable for refugees, but that might not always be the case. We also treated indicators independently and ignored any potential correlations that might exist between them. For instance, higher property values might

contribute to a lower percent of home ownership, but our results do not reflect any potential correlation that could exist between both.

In addition to conducting the current state analysis of all of HIAS's affiliates by their respective county's national rank, we identified areas where HIAS's network could be improved by considering all affiliates at once. This analysis of the whole network was completed by using unsupervised learning algorithms to group and compare specific attributes of the network in relation to the full county data set.

Analytics Through Unsupervised Learning Algorithms

To decide what clustering techniques to use and whether principal components analysis would be necessary, it was necessary to first gain a better understanding of the metrics that we needed to use to identify optimal counties to resettle refugees. The ideal metrics are the ones that are most representative of the previously mentioned 10 Domains of Integration (Ager and Strang, 2008). We used Datausa.io and the Bureau of Economic Analysis to compile 25 different metrics that cover all of the 10 domains of integration and grouped them into six areas: economy, health and safety, education, living and housing facilitators, social integration facilitators, and Jewish population density. The full list of metrics and their descriptions are summarized in Table 12.

Table 12: List of Indicators Used for Unsupervised Learning Analysis

Category	Indicator	Description
Economy	Unemployment	Average monthly unemployment
	County GDP per Capita for 2015	A county's gross domestic product per capita for the year 2015
	County Real GDP for 2015	A county's gross domestic product for the year 2015
	State GDP per Capita for Last Quarter	The county's state gross domestic product per capita for Q4 2018
	State Real GDP for Last Quarter	The county's state gross domestic product for Q4 2018
	Income Below Poverty	Percent of population in the area whose income lies below the poverty line
Health and Safety	Primary Care Physicians	Ratio of the population to primary care physicians
	Mental Health Providers	Ratio of the population to mental health providers
	Other Healthcare Providers	Ratio of the population to other health providers
	Uninsured	Percentage of population with no health insurance
	Violent Crime	Ratio of violent crimes for every 100,000 residents
Education	High School Graduation Rate	Percent of highschool 9th grade cohort that graduates
	Some College	Percent of population with at least some level of post high school education
Living and Housing Facilitators	County Median houshold Income	Median income of households in the county
	State Median Income Adjusted for Price Parity and Taxes	Median income of households in the state adjusted for relative price differences compared to the norm and taxes
	Mean Commute Minutes	Average commute time for residents
	Percent of Non-driving Commuters	Percent of work force that commute via means other than a personal vehicle
	Public Transit Stations per Square Mile	The number of public transit stations per square mile
	Food Insecurity	The percent of the population in the area that did not have a reliable food source
	Income Inequality	Ratio of household income at the 80th percentile to that at the 20th percentile
Social Integration Facilitators	Social Associations	Number of social associations per 10,000 population, including membership organizations such as civic organizations, bowling centers, golf clubs, fitness centers, sports organizations, religious organizations, political organizations, labor organizations, business organizations, and professional organizations.
	Population that is not Proficient in English	Percent of the population in the county that is not proficient in English
	Non Profit Organizations per person	Number of non profit organizations per person
	Non-US Citizens	Percent of population that are not US citizens
Jewish Population	Estimate of the Percent of Jewish Population in the County	The annual estimate of the percent of population in a county that are associated with the Jewish religion

Before using the algorithms and analyzing their output, it was necessary to prepare the data for the algorithms to work effectively. Preparing the data consisted of deciding what to do with null values and inconsistent formatting. The null values were only 0.6% of the entire dataset, and to preserve the information that was held in the rest of the sample's metrics that were not null, we decided on replacing the null value with the most common value of that metric in the dataset. The formatting inconsistencies were handled by converting all of the numeric metrics to the same format. After this, we were able to begin the modelling.

It was then necessary to understand the characteristics of the data to decide which unsupervised learning techniques to use. The dataset that we used contained 3,142 rows and 25 columns. Each row represented a county and each column was one of the metrics that we compiled to describe the county. The dataset had a low amount of samples relative to its amount of dimensions (columns). Because of this, we deemed it appropriate to either use algorithms that are less sensitive to high dimensions, or use algorithms to reduce dimensionality before using an algorithm that could be sensitive to high dimensionality. Aside from this, the dataset had a number of outliers, which we needed to take into account in the clustering algorithm because they could contain the optimal counties that we were looking for. Therefore, we also needed to use an algorithm with low sensitivity to outliers.

The clustering algorithm we used is an algorithm known as HDBSCAN. It was developed by Campello, Moulavi, and Sander in 2013 and combines two types of clustering: hierarchical and density based. It uses hierarchical based clustering, which joins points based on their proximity to each other, to form all of the potential clusters in the data. It then uses density-based clustering to filter for the potential clusters that are made out of sparse data. The result is a set of clusters that are not sensitive to outliers.

The exploratory analysis consisted of an iterative process of determining the amount of clusters that the algorithm found, using principal components analysis to visualize the clusters on the data, analyzing the clusters of counties as a collective to compare each

cluster to another, and finding ways to manipulate the metrics so that the clustering algorithms found better separations within the data and were more consistent. The most consistent and explainable clusters that we found were given by three different approaches to analyzing the data.

The first approach consisted of converting the metrics to their percentile rank, dividing all percentiles that were below the 70th by 0.5, and using the HDBSCAN algorithm to find clusters in the data.

The second approach consisted of computing the average percentile rank for a county under each of the six groups of metrics, inflating the high performing percentiles by dividing all percentiles below the 70th by 0.5, and using HDBSCAN to cluster the counties based on their average percentile rank for the six areas.

The third approach consisted of using principal components analysis to reduce the dataset's dimensionality to 15 while retaining the PVE at 90%, and using HDBSCAN to find clusters among the reduced dataset. This allowed HDBSCAN to perform better by reducing the amount of densities across different dimensions that it would have to calculate, and identify a better separation between dense and sparse data.

3.3 Increasing the Accuracy of *Annie*

While *Annie* achieves great placements for refugee cases by maximizing the total expected number of employed refugees, it does not explicitly use the characteristics of each location, other than capacity, to inform the algorithm that predicts employment probability.

Our team communicated with Professor Alessandro Martinello of Lund University, who is the architect of the predictive algorithm behind *Annie*. We received his input on which location characteristics would help the algorithm's accuracy the most, and how they could be incorporated into the code that powers *Annie*.

At his request, we developed python scripts that pull data from different websites and structured the data to enable it to merge with the dataset used by the predictive algorithm to output employment probabilities. The metrics compiled by the tools, along with their descriptions and sources, are in the Table 13.

Table 13: List of Indicators Compiled by the Python Scripts

Indicator	Description	Source	Website
Unemployment Rate	A county's monthly unemployment rate	Bureau of Labor Statistics	https://www.bls.gov/
Employment Ratio	A county's monthly percentage of the working age population that is employed	Bureau of Labor Statistics	https://www.bls.gov/
County Yearly GDP Growth for 2013-2015	A county's Gross Domestic Product yearly growth for the years 2013-2015	Bureau of Economic Analysis	https://www.bea.gov/
State Quarterly GDP Growth	A county's state Gross Domestic Product quarterly growth	Bureau of Economic Analysis	https://www.bea.gov/
Share of Unskilled Labor	The USA's monthly percent of people in the work force that have an educational attainment of highschool graduation or less	Bureau of Labor Statistics, and US Census	https://www.bls.gov/, http://www.census.gov

4. Results and Analysis

In this chapter we present our findings and outcomes from each objective.

4.1 Proof of Concept: Optimization in Refugee Resettlement

From our optimization model, we identified an optimal objective function value of $1,378,069, which is the combined income of placing 40 families in optimal affiliate locations. We placed all 40 families in a suitable location, while using up full capacities in each affiliate location. Table 14 below presents a breakdown of notable features about each family and where the family was placed according to our model.

Table 14: Results from Solving Optimization Model

Family Number	Case Size	Relocation Site	Family Income in Affiliate Location	Avg. Family English Prof.	Avg. Family Education Level	Number of Medical Conditions
1	3	Charlotte, NC	$30,581	1.78	4.33	2
2	2	Philadelphia, PA	$33,421	2.67	8.5	1
3	4	Ann Arbor, MI	$42,972	3	4	3
4	2	Pittsburgh, PA	$35,123	3	5	0
5	1	Clearwater, FL	$19,194	2	1	0
6	5	Walnut Creek, CA	$59,042	2.33	4.8	3
7	2	Buffalo, NY	$31,581	3.33	6	1
8	4	Cleveland, OH	$27,510	2.5	5.75	1
9	5	Cleveland, OH	$26,302	2.6	4.4	3
10	4	Clearwater, FL	$27,076	2.92	5	3
11	2	Cleveland, OH	$16,749	3.16	5.5	1
12	2	Madison, WI	$39,644	2.16	5.5	2

13	3	Springfield, MA	$29,237	2.66	3	3
14	2	Philadelphia, PA	$32,180	3.33	3.5	1
15	1	Pittsburgh, PA	$15,967	2.67	2	1
16	4	Charlotte, NC	$21,086	1.5	4.75	1
17	2	Columbus, OH	$78,241	2.5	4	1
18	3	Charlotte, NC	$35,832	3.56	4	0
19	2	Clearwater, FL	$32,118	2.83	7	2
20	2	Toledo, OH	$20,230	2.83	5	1
21	3	Cleveland, OH	$24,363	2.78	3.33	3
22	1	Cleveland, OH	$10,315	1	4	0
23	3	Charlotte, NC	$45,140	2.22	5	1
24	2	Toledo, OH	$23,321	3	7	0
25	1	Columbus, OH	$47,818	1.33	2	1
26	2	Pittsburgh, PA	$24,652	1.67	3	0
27	3	Clearwater, FL	$19,485	1.67	4.67	0
28	2	Philadelphia, PA	$23,002	2.33	4.5	1
29	4	Buffalo, NY	$25,695	2.92	6	2
30	4	Ann Arbor, MI	$48,285	2.33	7.5	3
31	2	San Diego, CA	$36,952	1.67	7.5	1
32	3	Springfield, MA	$23,353	2.33	2.67	2
33	2	Ann Arbor, MI	$35,361	1.67	6.5	2
34	1	Springfield, MA	$18,318	1	2	0
35	2	Kent, WA	$47,682	1.67	2	2
36	2	Walnut Creek, CA	$56,397	2.83	6	1

37	3	Buffalo, NY	$21,742	2.56	1.67	2
38	1	Wilmington, DE	$56,599	2.67	3	1
39	1	Framingham, MA	$71,575	3.33	6	0
40	3	Ann Arbor, MI	$47,695	2.11	4.33	3

These results enabled us to identify key features of each location. For example, Ann Arbor is the affiliate location with the largest number of physicians per 100,000 people, making it an ideal location for a family with multiple medical conditions. Our model resettled three families in Ann Arbor. Each of these families had either two or three people with medical conditions, demonstrating that this is the ideal location for families with various medical conditions. See Appendix B for the complete output from the optimization model.

To compare the results of our optimization model, we tried an alternative approach to placing refugee families in affiliate locations, which, while admittedly we are less experienced in resettling than the staff at HIAS, we thought might be a reasonable exercise. We decided to place refugee families via a round-table discussion, a similar process to how HIAS places refugee families. We briefly compared family-specific factors and location-based factors and made a judgment call on where we believed the family should be relocated, taking affiliate capacities into consideration. The following table displays the results of our deliberation. Whether or not a person has medical conditions is a big factor in how HIAS decides where to place people. Based off of this, we placed people who have medical conditions first. We placed them in locations that have the best medical resources according to what we know about each location. The remaining people were placed on a somewhat ad-hoc basis. We then pulled the family income data to see how the new placement decisions affected income. We looked at whether or not this new placement decision had an income that was greater than or less than our model's placement decision. We found that 85% of the time, the optimization model did better than our round-table placement when using income as a metric for

comparison. This shows that our model has some validity, at least over an attempt to replicate what a manual placement process might look like.

Table 15: Results from Simulating Manual Placement

Family Number	Case Size	Relocation Site	Family Income in Affiliate Location
1	3	Pittsburgh, PA	$31,279
2	2	Wilmington, DE	$29,254
3	4	Madison, WI	$38,828
4	2	Charlotte, NC	$32,827
5	1	San Diego, CA	$10,190
6	5	Ann Arbor, MI	$43,649
7	2	Cleveland, OH	$24,792
8	4	Cleveland, OH	$27,511
9	5	Ann Arbor, MI	$25,414
10	4	Ann Arbor, MI	$21,027
11	2	Cleveland, OH	$16,749
12	2	Kent, WA	$38,562
13	3	Ann Arbor, MI	$27,236
14	2	Cleveland, OH	$23,748
15	1	Cleveland, OH	$14,713
16	4	Columbus, OH	$24,299
17	2	Columbus, OH	$78,241
18	3	Clearwater, FL	$30,574
19	2	Springfield, MA	$24,300
20	2	Toledo, OH	$20,230
21	3	Framingham, MA	$29,926
22	1	Clearwater, FL	$9,259
23	3	Toledo, OH	$36,938

24	2	Clearwater, FL	$23,521
25	1	Charlotte, NC	$17,078
26	2	Clearwater, FL	$19,537
27	3	Philadelphia, PA	$17,048
28	2	Walnut Creek, CA	$27,204
29	4	Springfield, MA	$20,816
30	4	Pittsburgh, PA	$47,921
31	2	Charlotte, NC	$28,748
32	3	Springfield, MA	$23,353
33	2	Buffalo, NY	$31,538
34	1	Philadelphia, PA	$17,428
35	2	Buffalo, NY	$36,902
36	2	Walnut Creek, CA	$56,397
37	3	Buffalo, NY	$21,742
38	1	Charlotte, NC	$44,864
39	1	Philadelphia, PA	$47,246
40	3	Pittsburgh, PA	$41,704

4.2 Perform an Analysis of the HIAS Network

To capitalize on the opportunities that data presents, it is important to consider that proper presentation and data visualization is just as important as the analytics behind the data. In our methodology, we introduced the derivation process for county specific scores, an overall measure of a county's performance in 20 socioeconomic metrics extracted from the DataUSA database, and ranked nationally by year among all counties in the US. Using Microsoft Excel, we were able to develop a set of two dashboards and one lookup to be used by HIAS to visualize our findings from the national counties analysis.

Figure 5 is a sample output of Dashboard 1 - County Analysis by National Rank for Middlesex County in Massachusetts. In the sample output, Middlesex County receives an overall score of 71.88% when considering all five years of data (2013 to 2017) and all twenty indicators. We considered it important to allow the user to decide which indicators are relevant and which ones are not by enabling them to determine whether an indicator should be considered in the score. The scores in each of the five categories do not affect the overall score, but the individual indicators do.

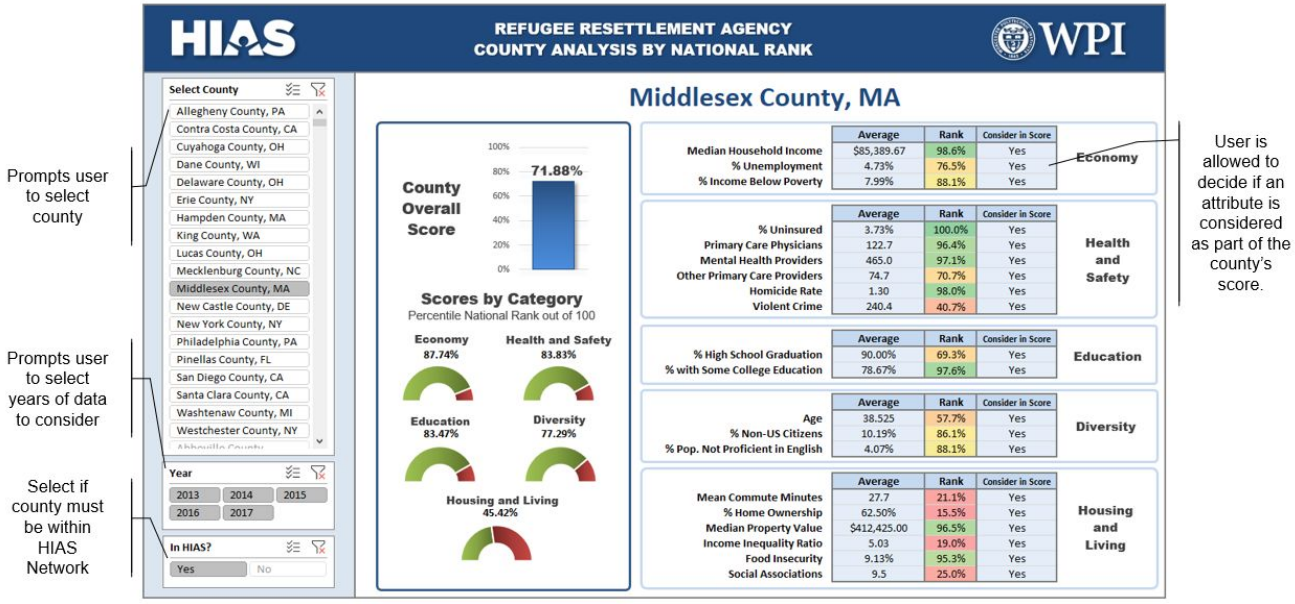


Figure 5: Dashboard 1 - County Analysis by National Rank

The user can also compare the county selected in Dashboard 1 with any other county in the US. The output of the comparison is available through Dashboard 2 - County Comparison by National Rank in Key Metrics as seen in Figure 6. The results of Dashboard 2 also reflect the selection criteria for metrics to consider that the user was provided with in Dashboard 1.

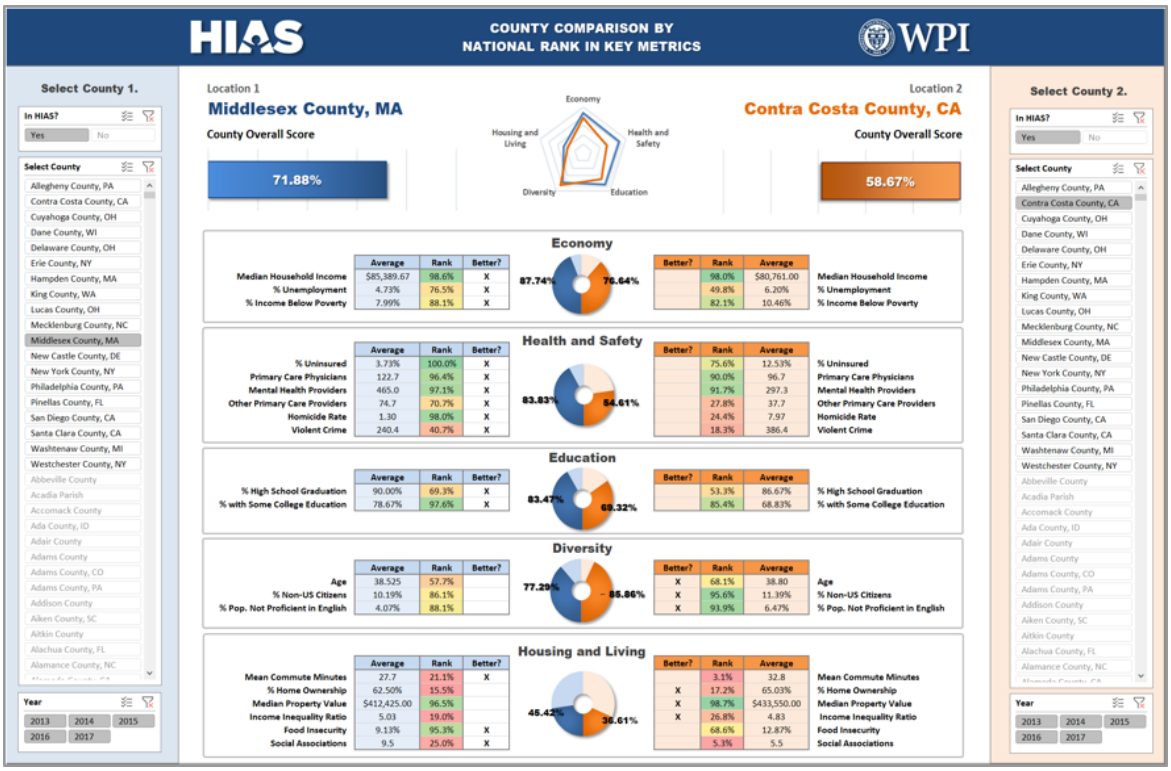


Figure 6: Dashboard 2 - County Comparison by National Rank in Key Metrics

Figure 6 is the sample output of the comparison between Middlesex County in Massachusetts and Contra Costa County in California. The user can visualize the comparison between overall scores for the county, as well as compare the five categories along with the indicators and a respective signal for which county ranks higher in each indicator. Ultimately, Dashboard 2 is meant to be a tool to discover and compare the current HIAS affiliate locations with locations outside the network for future expansion opportunities.

As part of our data analysis summary, we provided HIAS with a lookup tool to summarize potential new locations based on their county overall score and percent of Jewish population in the area. We extracted data updated to as late as 2015 for the Jewish population throughout the US by state from the Steinhardt Social Research Institute at Brandeis University (“American Jewish Population Project”, 2015). The user

is provided with a data summary for HIAS as seen in Figure 7 to help inform and guide the lookup process for new cities.

Summary of HIAS Network Counties
By Jewish Population and County Score
Accounts for 2015 Information Only

State	County	% Jewish Population*	Location Score
California	Contra Costa County	1.75%	63.37%
California	San Diego County	1.45%	64.63%
California	Santa Clara County	1.21%	71.32%
Delaware	New Castle County	1.62%	65.50%
Florida	Pinellas County	1.89%	48.23%
Massachusetts	Middlesex County	6.03%	77.08%
Massachusetts	Hampden County	2.10%	54.12%
Michigan	Washtenaw County	1.66%	70.17%
New York	New York County	14.62%	62.48%
New York	Westchester County	11.15%	70.06%
New York	Erie County	0.80%	58.86%
North Carolina	Mecklenburg County	1.76%	58.68%
Ohio	Cuyahoga County	2.57%	48.14%
Ohio	Delaware County	1.04%	78.93%
Ohio	Lucas County	0.91%	42.04%
Pennsylvania	Philadelphia County	4.24%	39.99%
Pennsylvania	Allegheny County	2.86%	62.70%
Washington	King County	1.11%	70.18%
Wisconsin	Dane County	0.37%	77.10%
	HIAS Average	**3.11%**	**62.29%**
	National Average	**0.51%**	**49.91%**

*Source: Steinhardt Social Research Institute - Brandeis University

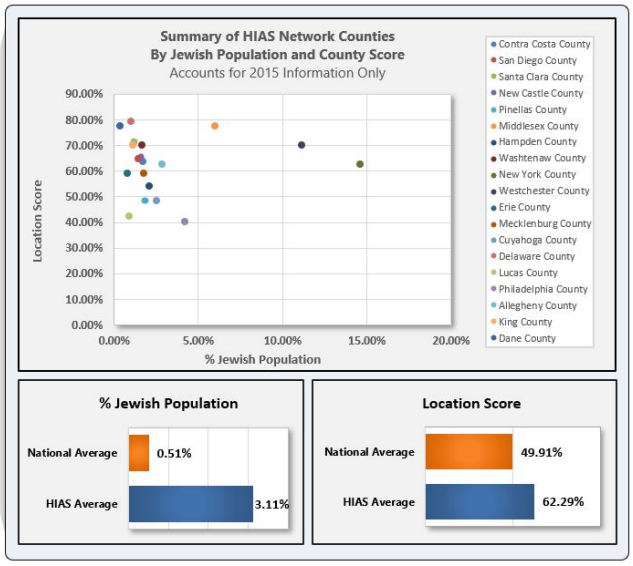
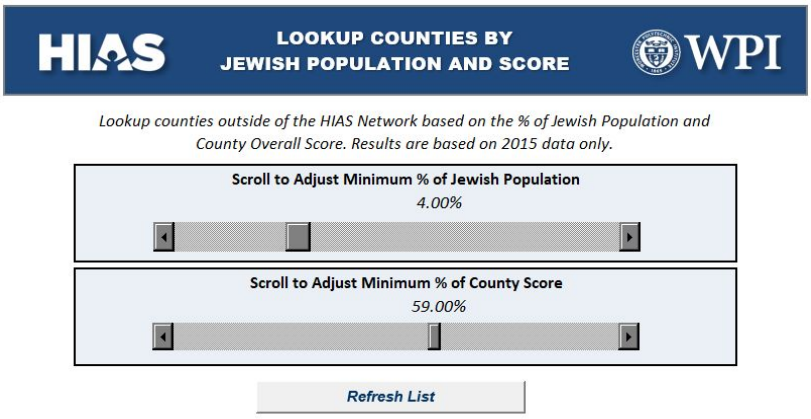


Figure 7: Summary Data - HIAS Network by County and Jewish Population

While HIAS was originally an organization that exclusively resettled Jewish refugees, it now resettles refugees from diverse backgrounds and has operations in countries other than the US. Nonetheless, it is not surprising that eighteen out of the 19 counties in HIAS's network have a higher concentration of Jewish population than the national average of 0.51% at a county level, with Dane County in Wisconsin being the only county below the national average. The average Jewish population by county in the HIAS network is 3.11%, with New York County, Westchester County, and Middlesex County having the highest concentration of Jewish population.

Our data analysis is enhanced through a lookup form that allows the user to find other counties in the US based on their county score and Jewish population as seen in Figure 8. As in the other dashboards, the user is allowed to define the minimum thresholds for Jewish population and county score, a selection process that is likely informed by the information provided in the HIAS Jewish Population Summary. Finally, it is important to note that our findings with Jewish population are limited to the latest data available at a

county level as of 2015. Future versions of this analysis could consider more recent information about Jewish population and county scores.



Counties by State	% of Jewish Population	County Score
District of Columbia		
District of Columbia, DC	6.99%	59.72%
Maryland		
Baltimore County, MD	4.55%	67.05%
Montgomery County, MD	6.99%	77.23%
Massachusetts		
Middlesex County, MA	6.03%	77.08%
New Jersey		
Bergen County, NJ	6.43%	74.26%

Figure 8: Location Lookup by County Score and Jewish Population

Clustering Results

The first approach for discovering groups of counties grouped them according to the disparity in their performances across different metrics. Each cluster represented counties that excelled in one specific category of metrics, but were average or below average in other metrics. This output gives HIAS the liberty of choosing which metric has the most importance for them. They can look to the clusters of counties that outperform in these metrics, while setting a performance threshold for the other metrics.

The second approach, which generalized performance to six overarching areas, output clusters that were divided by the overall rank of counties across the six areas. It allowed us to identify counties that were above-average performers relative to the entire US in

all six categories. This output enables HIAS to sort through the counties that have the best overall combination of economic, health, quality of life, and diversity metrics.

The last approach used principal components analysis to reduce the amount of variables and retain the most information. It then used HDBSCAN to cluster the data and output clusters that were similar to the first approach. The only difference was that it output fewer clusters and generalized county performance to more than one or two specific metrics.

The full list of the distinctively “best” counties found by each cluster is included in Appendix C. The visualization plots of the clusters for each algorithm are found in Appendix F.

An additional dataset that we pulled from DataUSA.io allows the data to be segmented by predominant and weak industries or occupations in each county. This allows for additional analysis of the clusters of counties created by the algorithms. Upon further exploration, we found that HIAS is strong in the healthcare, educational services, retail trade, and scientific services industries. We also found that it does not have affiliates with a higher prevalence of the utilities, mining, oil, agriculture, transportation, and warehousing industries. When we compared these characteristics to the cluster of high performing counties across the US, we found that the strong industries in HIAS’s network are similar to most of the strong industries in the high performing counties.

After analyzing the clusters in the context of the locations and industries that HIAS has a prevalence in, we selected five top performing locations that would be prime candidates for new locations. These counties are Howard county in Maryland, Loudoun county in Virginia, Collin County in Texas, DuPage County in Illinois, and Adams County in Colorado. All of the counties have proportions of Jewish populations above the 70th percentile, as well as above average scores in the five overarching socioeconomic indicators that we used to group our data. These recommended locations are in new

areas of the US that are far from HIAS's current affiliates, and they also increase the diversity of the predominant industries across the affiliates.

4.3 Increasing the Accuracy of *Annie*

The metrics compiled by the tool increase *Annie*'s accuracy and validity by allowing the predictive algorithm behind *Annie* to incorporate the economic environment in each location into its employment probability estimates. It increases accuracy because ultimately, no matter how favorable a person's likelihood for employment is, if there is not a favorable economic environment to facilitate employment, it will be harder for a person to be employed. It is important to note that, due to time limitations, it was not possible to test the accuracy.

5. Recommendations and Conclusions

The research and analysis we performed throughout the duration of our MQP allowed us to make multiple recommendations for HIAS. We aimed to produce insightful recommendations for how HIAS can improve their refugee resettlement process. This section will reference the current method of data and information collection, as well as a potential for expanding the current affiliate locations to include additional complementary cities. It will also touch upon current optimization tools used to aid refugee resettlement and recommendations for how to improve these methods for the future.

We recommend that HIAS considers the potential impact of collecting more information about affiliate locations and refugees, as it might have a large impact on future data-driven decision making tools and refugee integration. There is currently little information collected on refugees. Specifically, there is no easily accessible information collected on the type of medical conditions a person has or a refugee's preferences for an affiliate location. Furthermore, limited information about each affiliate location is taken into consideration when making a placement decision. Through our analysis we were able to determine the strengths and weaknesses of each affiliate location amongst six domains: education, community & social diversity, health & safety, facilitators, economy & employment, and Jewish population. It would be beneficial for HIAS to decide their own methods or domains, such as these, for assessing a location because it could help HIAS understand how each affiliate location compares to one another. Knowing at a deeper level what each location's strengths and weaknesses are may be beneficial for placing families with certain characteristics. This information could also aid future decision-making software, as there will be a greater understanding for how to incorporate both location-specific and refugee-specific indicators.

Additionally, we recommend that if HIAS considers expanding its network to include more affiliate locations, then it should use the results of our clustering algorithm to aid in its decision-making. As mentioned previously, our clustering algorithm output enables HIAS to choose which metric has the most importance to them. This allows HIAS to look at the clusters of counties that outperform in these decided metrics. For example, using our current clustering algorithm, HIAS can sort through the counties that have the best overall combination of economic, health, quality of life, and diversity metrics. This could be beneficial for HIAS if they choose to explore additional options for affiliate locations.

Furthermore, we recommend to draw upon publicly available sources of data to aid placement and expansion decision making. Using DataUSA enabled us to gain information that was beneficial to our project. Public sources of information such as this can enable HIAS to have more factors to take into consideration when placing refugees, instead of only using refugee information. Sources such as the United States Census and the Bureau of Labor Statistics have vital location and employment statistics. Access to more data such as this would make resettlement optimization models such as *Annie* more accurate, which will improve placement decisions and subsequently the results.

Our last recommendation is to use and continue to develop tools that increase the accuracy of *Annie*. Increasing the accuracy and validity of *Annie*'s decision making has a direct impact on refugee integration because it influences placement decisions. If those decisions are more informed and a more data-driven approach is taken to reach placement decisions, the success of refugee integration can be greatly enhanced.

Ultimately, it is important to remain optimistic about the future. The present may look bleak as the resettlement cap is consistently declining, but with future administrations comes the potential for the cap to increase again. Our initiatives make a compelling argument for how the US can improve refugee placement decisions, leading to more successful resettlement outcomes.

6. Project Reflection

6.1 Designing the Project

We applied the engineering design process to our MQP through scoping our project. Initially, our project was going to be with UMass Memorial Medical Center in their cataract surgery department. The objective in this project was clear as we were to improve throughput in this department. However, after our first meeting with the doctors was cancelled at the last minute due to an emergency surgery, we realized this project may not have the available resources that we would need, including the doctors' time. This left us to quickly scope out on a new, practical project. Furthermore, we applied engineering design to our project through the development of various tools and recommendations that could be used by HIAS to aid in refugee resettlement. For example, we designed a clustering algorithm that seeks to identify complementary cities to the affiliate locations that are in HIAS's network. We also designed a dashboard, which is a user interface that can aid the members of HIAS in visualizing important information about various cities in the US. This improves the refugee resettlement process by incorporating location-based factors into the decision.

6.2 Constraints and Limitations

Throughout the process of completing our project, we encountered a number of constraints that limited our ability to accomplish certain tasks. For example, we had limited access to information on refugee resettlement. This was due to the fact that there is a large amount of classified information on this topic. We were not given access to actual refugee data, thus, we were required to make up our own based off of actual information collected. We also had limited access to information on the capacities of affiliate locations.

For the first semester of working on this project, we used the website DataUSA to pull online location information. Our optimization model, statistical techniques, and data

pulling tool were all based on information from DataUSA. After this semester, we learned that DataUSA will no longer be updated, so all of the information may become obsolete. This was a major limitation we encountered, as we had to figure out how to complete our project with this new obstacle.

Another major limitation we encountered during this project was that the majority of the refugee resettlement process happens at a government level. This meant that there was only so much we could do to implement our MQP, as we could not change much of the process nor interact with the people who are in charge of changing the process. For example, it would not be feasible for us to change the way refugee resettlement data is collected since this is a standard governmental procedure.

Additionally, time was a limitation for us during this project. Because refugee resettlement is a broad topic, it required us to do an extensive amount of research to determine the impact we wanted to make. By the time we surpassed the learning curve and pinpointed the goals of our project, we only had about seven to ten weeks to develop our model, build the clustering algorithm, and perform data analysis. We wish we had even more time to work on this project because six months is a limited time to make an impact.

6.3 Acquiring and Applying New Knowledge

Our project required a combination of knowledge about industrial engineering, economics, social science, computer science and data science. One of the biggest learning curves for us was with regards to the computer/data science portion of our project; creating a clustering algorithm, data pulling tool, and a complex Excel dashboard to display the results of our clustering algorithm. There is a small computer science requirement for industrial engineers, thus, we had to figure out how to appropriately use specific software in a more advanced matter. For example, we learned how to use Python to run a clustering algorithm that could identify complementary cities similar to those already in HIAS's network.

Even more so, our project had a major social science and economics component as we had to research factors that affect a refugee's ability to integrate into US society. To gain knowledge on these topics, we not only performed individual research online, but we also reached out to experts at WPI. For example, to learn more about how to quantify a refugee's economic impact, we reached out to Professor Somasse, an economics professor at WPI, and met with him to discuss our questions.

From an industrial engineering perspective, we learned more about how to scope a project appropriately, understand the types of constraints affecting a model, and how to provide meaningful recommendations to a sponsor. Additionally, we learned that sometimes data is not as readily available in the real world as we would like to assume. Data is often incomplete, hard to access, or incorrect, and can sometimes require manual manipulation to fix.

6.4 Project Teamwork

Each member of our MQP team made valuable contributions to our project. As individuals, we each have strengths in different areas of the industrial engineering field. A few members of our team are more skilled in the data analysis, computer science, and mathematics side of industrial engineering, while others are more comfortable with economics, social science, and business. In this way, we each brought something special to the table, as our project was a culmination of all of these areas.

Although the physical MQP work was spread out amongst the four of us, the administrative side changed frequently. We each took turns leading meetings with our advisor, being the point of contact for emails, creating and sending agendas, and writing the paper. This gave each of us valuable experience with leadership and developing strong interpersonal skills that are necessary for the workforce. In this way, we became a tremendously well-rounded team and learned to trust in one another and our abilities to excel in any dimension of the project. This itself worked to our

advantage and kept us each accountable for our actions, especially with regards to meeting weekly goals.

Furthermore, we met frequently outside of our standard meetings with our advisor. In addition to working independently, each week we met about four times for about two hours each. Recurrent team meetings were beneficial to our team dynamic because it allowed us to stay up to date with our individual tasks, as well as identify how we could work together as a team to help each other. There was never a question about whether any team member was committed to this project, and that speaks volumes about the dedication, work ethic, and passion that was put into our MQP.

7. References

Ager, Alastair and Strang, Alison (2008). Understanding Integration: A Conceptual Framework. Journal of Refugee Studies, 166-191.

Ahani, N. (2018, November 10). Files to test NLP scripts for Medical Information in Refugee Resettlement [E-mail].

Ahmed, Afreen (2017, August 29). The Matching Project: A Systematic Approach to Refugee Placement. Unpublished Report presented to the Hebrew Immigrant Aid Society (HIAS).

American Jewish Population Project. (2015). Retrieved February 1, 2019 from http://ajpp.brandeis.edu/

Åslund, Olof, Per-Anders Edin, Peter Fredriksson, and Hans Grönqvist (2011). "Peers, Neighborhoods, and Immigrant Student Achievement: Evidence from a Placement Policy." American Economic Journal: Applied Economics, 3 (2): 67-95.

Åslund, O., and D.-O. Rooth (2007). “Do when and where matter? Initial labour market conditions and immigrant earnings”. The Economic Journal 117 (518): 422–448.

Bray, I. (2016, January 27). Inadmissibility: When the U.S. Can Keep You Out. Retrieved February 20, 2019, from https://www.nolo.com/legal-encyclopedia/us-deny-entry-inadmissibility-reasons-29715.html

Bansak, K., Ferwerda, J., Hainmueller, J., Dillon, A., Hangartner, D., Lawrence, D., & Weinstein, J. (2018). Improving refugee integration through data-driven algorithmic assignment. *Science*, *359*(6373), 325-329. doi: 10.1126/science.aao4408

Campello R.J.G.B., Moulavi D., Sander J. (2013) Density-Based Clustering Based on Hierarchical Density Estimates. In: Pei J., Tseng V.S., Cao L., Motoda H., Xu G. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2013. Lecture Notes in Computer Science, vol 7819. Springer, Berlin, Heidelberg

Cepla, Z. (2019, January 25). Fact Sheet: U.S. Refugee Resettlement. Retrieved February 25, 2019, from https://immigrationforum.org/article/fact-sheet-u-s-refugee-resettlement/

Cultural Orientation Resource, R&P Orientation Curriculum. (2018). Retrieved October 7, 2018 from

http://www.culturalorientation.net/providing-orientation/toolkit/r-p-orientation-curriculum

Day, J. C., & Shin, H. B. (2005). How Does Ability To Speak English Affect Earnings?(Report). US Census Bureau.

Displaced Persons Act of 1948. (2015). Retrieved October 5, 2018, from http://immigrationtounitedstates.org/464-displaced-persons-act-of-1948.html

Dokko, G., Wilk, S. L., & Rothbard, N. P. (2009). Unpacking Prior Experience: How Career History Affects Job Performance. *Organization Science,20*(1), 51-68. doi:10.1287/orsc.1080.0357

Duke Kominers, Scott & Teytelboym, Alexander & P Crawford, Vincent. (2017). An invitation to market design. Oxford Review of Economic Policy. 33. 541-571. 10.1093/oxrep/grx063.

Gareth et al., 2013. An Introduction to Statistical Learning. New York: Springer.

Graf, N., Brown, A., & Patten, E. (2018, April 09). The Narrowing, But Persistent, Gender Gap in Pay. Retrieved February 10, 2019 from http://www.pewresearch.org/

Holder, Sarah. The Algorithm That Can Resettle Refugees (2018). *Citylab*. Accessed September 30, 2018 from https://www.citylab.com.

Jolliffe, I. (2002). Principal Component Analysis, Second Edition. *Springer Series In Statistics*, 518. Retrieved from http://cda.psych.uiuc.edu/statistical_learning_course/Jolliffe%20I.%20Principal%20Component%20A

Jones, Will & Teytelboym, Alexander. (January 2016). Choices, preferences and priorities in a matching system for refugees. Forced Migration Review. Retrieved October 2, 2018, from https://www.fmreview.org/destination-europe/jones-teytelboym

Jones, Will & Teytelboym, Alexander. (2018). The Local Refugee Match: Aligning Refugees' Preferences with the Capacities and Priorities of Localities. Journal of Refugee Studies. 31. 152-178. 10.1093/jrs/fex022.

Kerwin, D. (2018). The US Refugee Resettlement Program — A Return to First Principles. *Journal on Migration and Human Security,*233150241878778. doi:10.1177/2331502418787787

Lichtenstein, G., Puma, J., Engelman, A., & Miller, M. (2016) The Refugee Integration Survey & Evaluation (RISE): Year 5 Report. *Technical report by Quality Evaluation Designs.* Denver, CO: Colorado Office of Economic Security.

Lorenzo-Seva, U. (2013). How to report the percentage of explained common variance in exploratory factor analysis. Technical Report. Department of Psychology, Universitat Rovira i Virgili, Tarragona.

L. McInnes, J. Healy, S. Astels, HDBSCAN: Hierarchical density based clustering In: Journal of Open Source Software, The Open Journal, volume 2, number 11. 2017

Moraga, J. (2014, December 14). Tradable Refugee-admission Quotas and EU Asylum Policy *. Retrieved October 1, 2018, from https://doi.org/10.1093/cesifo/ifu037

National Committee on Pay Equity. (2018). The Wage Gap Over Time: In Real Dollars, Women See a Continuing Gap. Retrieved February 10, 2019, from https://www.pay-equity.org/

National Women's Law Center. (2018). The Wage Gap. Retrieved February 10, 2019, from https://www.infoplease.com/us/gender-sexuality/wage-gap

Nibbs, F. (2017, March 02). Belonging: The Resettlement Experiences of Hmong Refugees in Texas and Germany. Retrieved on October 3, 2018, from https://www.migrationpolicy.org/article/belonging-resettlement-experiences-hmong-refugees-texas-and-germany

OECD (2012), “How does education affect employment rates?”, in Education at a Glance 2012: Highlights, OECD Publishing, Paris.

OpenSolver for Excel. (2019). Retrieved February 10, 2019, from https://opensolver.org/

Our History (2019). HIAS. Retrieved September 30, 2018 from https://www.hias.org.

Park, J. H. (1999). The Earnings of Immigrants in the United States: The Effect of English-Speaking Ability. *American Journal of Economics and Sociology, Inc.* Retrieved December 13, 2018.

Python Software Foundation. Python Language Reference, version 3.7. Available at http://www.python.org

Refugee Council USA. History, Legislative Authority, & Major Administrative Agencies. (n.d.). Retrieved October 5, 2018, from http://www.rcusa.org/history/

Schubert, E. (2017). *Knowledge Discovery in Databases, Part III: Clustering*. Presentation, Universitat Heidelberg.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

Statistical Analysis - What is it?. (2019). Retrieved February 15, 2019 from https://www.sas.com/en_us/insights/analytics/statistical-analysis.html

RPC - Refugee Processing Center (2018). “Refugee Admissions Report September 30, 2018”. Retrieved on October 2, 2018, from http://www.wrapsnet.org/admissions-and-arrivals/

Rysman, M. (2009). The economics of two-sided markets. *Journal of Economic Perspectives*, *23*(3), 125–143. doi:10.1257/jep.23.3.125

Tent Foundation (2017). Tent Tracker: Public Perceptions of the Refugee Crisis. Global Report.

Trapp, A., Teytelboym, A., Martinello, A., Andersson, T., & Ahani, N. (2018). Placement Optimization in Refugee Resettlement. *Working Papers 2018:23, Lund University, Department Of Economics*. Retrieved February 20, 2019 from https://ideas.repec.org/p/hhs/lunewp/2018_023.html

UNHCR. (2017, June). “Global Resettlement Needs 2018”. Technical report, United Nations High Commissioner for Refugees.

UNHCR. (2018a). UNHCR Statistical Yearbooks. Retrieved October 2, 2018, from http://www.unhcr.org/en-us/statistical-yearbooks.html

UNHCR. (2018b, Mar). “Mid Year Trends 2017”. Technical report, United Nations High Commissioner for Refugees.

UNHCR. (2019). *UNHCR Projected Global Resettlement Needs*. Geneva.

US Census Bureau. (2019, February 19). Data: CPS Historical Time Series Tables. Retrieved February 24, 2019, from https://www.census.gov/data/tables/time-series/demo/educational-attainment/cps-historical-time-series.html

USCIS: Learn About the Refugee Application Process. (n.d.). Retrieved October 5, 2018, from https://www.uscis.gov/humanitarian/refugees-asylum/refugees

USA for UNHCR: The UN Refugee Agency. (2018). Refugee Statistics. Retrieved February 10, 2019, from https://www.unrefugees.org/refugee-facts/statistics/

USRAP: Application and Case Processing. (2018). Retrieved October 5, 2018, from https://www.state.gov/j/prm/ra/admissions/

Utah State Board of Education, & Utah Education Network. (2008). Finance in the Classroom: What Factors Affect Your Income? Retrieved from https://financeintheclassroom.org/

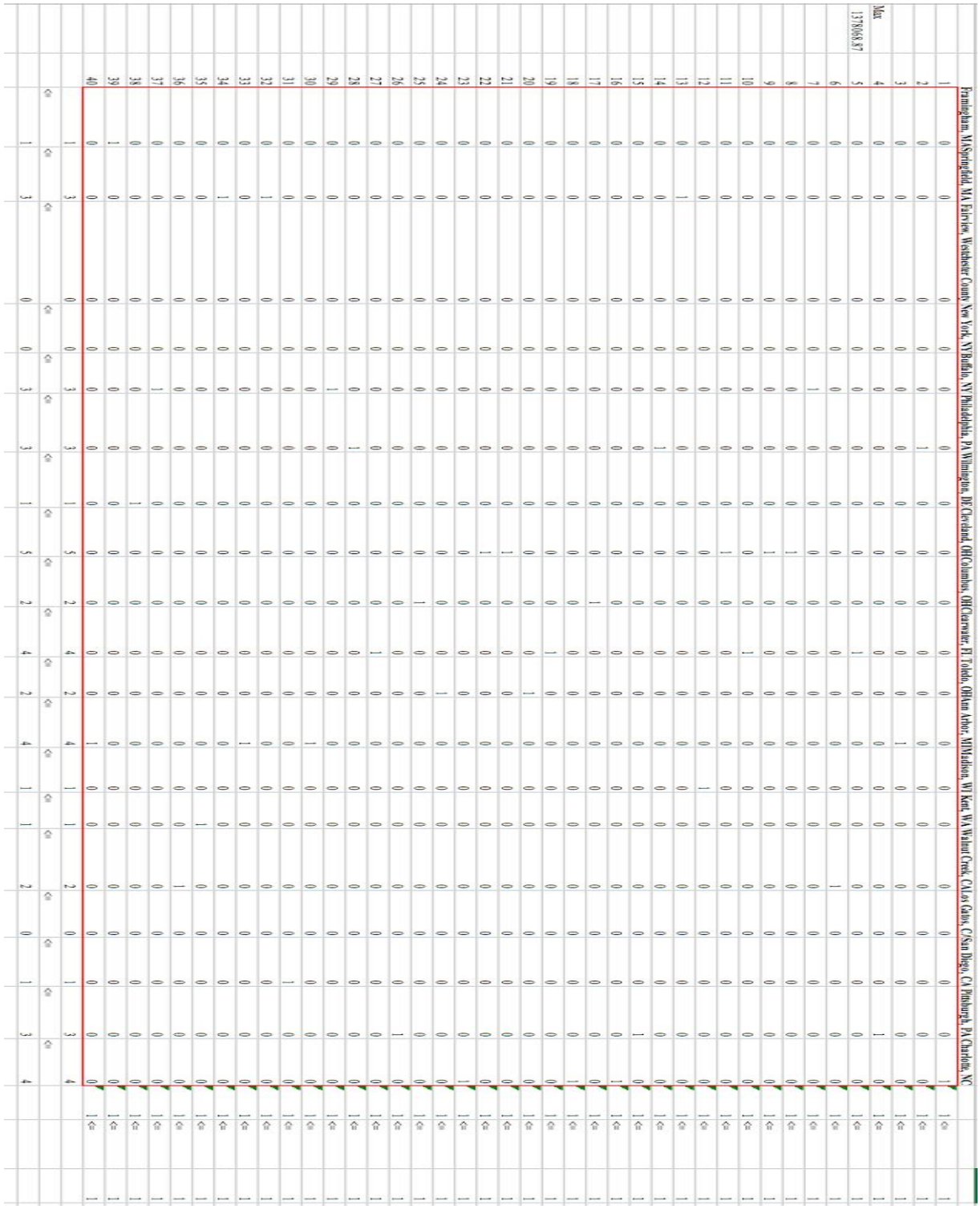
van Selm, Joanne. "Public-Private Partnerships in Refugee Resettlement: Europe and the US." *Journal of International Migration and Integration* 4.2 (2003): 157-75. *ProQuest.* Web. 27 Sep. 2018.

Appendix A: Data Factors

Food Insecurity	The percentage of the population without reliable access to an acceptable quantity of food in each affiliate location
Unemployment Rate	The percentage of the civilian labor force, age 16 and older, that is unemployed but seeking work in each affiliate location
Income Inequality	The ratio of household income at the 80th percentile to that at the 20th percentile in each affiliate location
Poverty	Population living below the poverty line, for whom poverty status is determined
High School Graduation Rate	The percentage of the ninth-grade cohort in public schools that graduates from high school in four years in each affiliate location
Percentage of Population with Some College	Percentage of the population ages 25-44 with some post-secondary education
Violent Crimes	Number of reported violent crime offenses per 100,000 population
Health Care Costs	The amount of price-adjusted Medicare reimbursements per enrollee in each affiliate location
Physicians per 100,000 people	The number of physicians per 100,000 people in an affiliate location
Percent of Non-English Speaking People	Percent of total population in an affiliate location that does not speak English
Median Household Income	Income at which half the households earn more and half the households earn less
Mental Health Providers	Ratio of the county population to the number of mental health providers
Other Primary Care Providers	Number of other primary care providers per the population of a county, which include nurse practitioners, physician assistants, and clinical nurse specialists
Uninsured	Percentage of the population under age 65 that has no

	health insurance coverage
Social Associations	Number of social associations per 10,000 population
Nonprofits per Person	The number of non-profit organizations per person
Transit Stations per Square Mile	The number of transit stations per square mile
Percent Jewish in State	Percentage of the population that is Jewish per each state
Percent Non-US Citizens	Percentage of population that are Non-US Citizens
Mean Commute Minutes	Mean Commute Time in Minutes
Percent of Commuters that do not Drive to Work	The percentage of the total commuter population that does not drive to work
GDP per Capita for County (2015)	Growth domestic product per capita
Average County GDP Growth (2013-2015)	Average growth domestic product growth per country
Average State Quarterly GDP Growth (2015-2017)	Average quarterly growth domestic product growth per state
State per Capita GDP (2017)	State per capita growth domestic product
State Income Adjusted for price Parity and taxes	The income per state, adjusted to include price and parity taxes
Homicide Rate	Number of deaths due to homicide per 100,000 population
Owner Occupied Housing Units	Percentage of housing units that are Owner occupied
Median Property Value	Median property value
Tax Rates	Tax rate per affiliate location
Median Income of Employment Industry by Gender	The median income for both males and females in each employment industry within each affiliate location
Median Salary per English Speaking Level per Age	The median income for each level of English proficiency broken down by age in each affiliate location

Appendix B: Optimization Model Results



Appendix C: Table of Complementary Counties by Cluster

Top 20 overall Performers	Top 20 Economy	Top 20 Health and Safety	Top 20 Education	Top 20 Living and Housing Faciliators	Top 20 Social Integration Faciliators	Top 20 by Jewish Population
Norfolk, MA	Howard, MD	Chittenden, VT	Borden, TX	Dukes, MA	Sioux, IA	New York, NY
Somerset, NJ	Collin, TX	Warren, NY	Kent, TX	San Juan, WA	Stevens, KS	Palm Beach, FL
Dukes, MA	Williamson, TX	Dickinson, MI	Dallas, IA	Olmsted, MN	Marshall, SD	Kings, NY
Montgomery, PA	Morris, NJ	Washington, VT	Pitkin, CO	Adams, WA	Meade, KS	Westchester, NY
Morris, NJ	San Mateo, CA	Olmsted, MN	Oldham, TX	Middlesex, MA	Saline, NE	Nassau, NY
DuPage, IL	Loudoun, VA	Wood, WI	Williamson, TN	Roseau, MN	Crawford, IA	Broward, FL
Nantucket, MA	Midland, TX	Windsor, VT	Collin, TX	Hampshire, MA	Grant, SD	Montgomery, MD
Bergen, NJ	Hartley, TX	Rowan, KY	Delaware, OH	Divide, ND	Greeley, KS	Suffolk, NY
Nassau, NY	Dakota, MN	La Crosse, WI	Armstrong, TX	Rice, MN	Osceola, IA	Bergen, NJ
Loudoun, VA	Anne Arundel, MD	Cerro Gordo, IA	Teton, WY	Norfolk, MA	Madison, NE	Morris, NJ
Hunterdon, NJ	Saratoga, NY	Perry, KY	Ozaukee, WI	Nicollet, MN	Aurora, SD	Essex, NJ
Howard, MD	Broomfield, CO	Winneshiek, IA	Boone, IN	McMullen, TX	Alexandria	Passaic, NJ
Marin, CA	Nassau, NY	Boyd, KY	St. Croix, WI	Nobles, MN	Dixon, NE	Montgomery, PA
San Mateo, CA	Douglas, CO	Cumberland, ME	Morris, NJ	Lyon, MN	Cuming, NE	Bucks, PA
Montgomery, MD	DuPage, IL	Orange, VT	Hunterdon, NJ	Essex, MA	Emmet, IA	Rockland, NY
Summit, CO	Somerset, NJ	Whitley, KY	Kane, UT	Grant, WA	Kearny, KS	Sullivan, NY
Monmouth, NJ	Suffolk, NY	Waukesha, WI	Hendricks, IN	Douglas, IL	Stafford, KS	Orange, NY
Grafton, NH	Arlington, VA	Ozaukee, WI	Winneshiek, IA	Bergen, NJ	Mower, MN	Putnam, NY
Hampshire, MA	Eagle, CO	Wadena, MN	Randall, TX	Kittson, MN	Kandiyohi, MN	Dutchess, NY
Middlesex, CT	Guadalupe, TX	Eau Claire, WI	Middlesex, CT	Sargent, ND	Hall, NE	Ulster, NY

Appendix D: Python Code for Data Pull and Industries in Counties

```
## Necessary libraries
import requests
import pandas as pd

## Creating dictionary to link counties to affiliates
county_names = ['Middlesex County, MA','Hampden County, MA','Westchester County,
NY','New York County, NY','Erie County, NY',
'Philadelphia County, PA','New Castle County, DE','Cuyahoga County, OH','Delaware
County, OH','Pinellas County, FL',
'Lucas County, OH','Washtenaw County, MI','Dane County, WI','King County,
WA','Contra Costa County, CA',
'Santa Barbara County, CA','San Diego County, CA','Allegheny County,
PA','Mecklenburg County, NC','Broward County, Fl',
'DeKalb County, GA','Fulton County, GA','Cook County, IL', 'DuPage County, IL',
'Montgomery County, MD',
'Essex County, NJ']

affiliate_names = ['Framingham, MA','Springfield, MA','Fairview, Westchester
County','New York, NY','Buffalo, NY',
'Philadelphia, PA','Wilmington, DE','Cleveland, OH','Columbus, OH','Clearwater,
FL','Toledo, OH','Ann Arbor, MI',
'Madison, WI','Kent, WA','Walnut Creek, CA','Los Gatos, CA','San Diego,
CA','Pittsburgh, PA','Charlotte, NC',
'Lauderdale Lakes City, FL','Atlanta City, GA','Atlanta City, GA','Chicago City, IL',
'Chicago City, IL',
'Rockville City, MD','East Orange City, NJ']

county_code =
['CN2501700000000','CN2501300000000','CN3611900000000','CN3606100000000','C
N3602900000000',
'CN4210100000000','CN1000300000000','CN3903500000000','CN3904100000000','CN
1210300000000','CN3909500000000',
'CN2616300000000','CN5502700000000','CN5303500000000','CN0601500000000','CN
0608500000000','CN0607500000000',
'CN4200500000000','CN3712100000000','CN1201300000000','CN1309100000000','CN
1312300000000','CN1703500000000',
'CN1704700000000','CN2403500000000','CN3401700000000']

membership_dictionary = {'countyname': county_names,'area_code':
county_code,'affiliate_name': affiliate_names}
```

```
membership_df = pd.DataFrame.from_dict(membership_dictionary)

## get monthly information
def get_data(start_year, end_year):
    ## reading csvs
    census_estimates = pd.read_csv(...)
    us_employment_ratio_df = pd.read_csv(...)
    Us_unemployment_df = pd.read_csv(...)
    hs_diploma_lf = pd.read_csv(...)
    no_hs_dip_lf = pd.read_csv(...)
    US_lf = pd.read_csv(...)
    county_gdp_growth = pd.read_excel(...,
                sheet_name = 'Real GDP Growth')
    state_gdp = pd.read_csv(...)

    df_areas = pd.read_table('https://download.bls.gov/pub/time.series/la/la.area')

    # Only keep county information
    df_areas = df_areas.loc[df_areas['area_type_code'].str.contains('F')]
    df_areas.reset_index(drop=True, inplace=True)

    # Get county and state information
    df_areas['countyname'] = df_areas['area_text']

    # Remove whitespace
    df_areas['area_code'] = df_areas['area_code'].map(lambda x: x.strip())
    df_areas['countyname'] = df_areas['countyname'].map(lambda x: x.strip())

    # Remove unnecessary columns
    df_areas = df_areas[['area_code', 'countyname']]

    #-----------------------------------------------------

    def get_BLS_county_data(BLS_data_path, df_areas):
        '''
        BLS_data_path : path for the text file containing the BLS data
        df_areas      : dataframe containing BLS information about counties/areas
        '''
        # Import area information
        col_types = {'series_id': str, 'year': int, 'period': str, 'value': str, 'footnote_codes': str}
        df_bls_county = pd.read_table(BLS_data_path, dtype=col_types)

        # Remove white space from code..
        df_bls_county['series_id'] = df_bls_county['series_id'].map(lambda x: x.strip())
```

```
    # Convert 'value' to numeric (kind of slow...)
    df_bls_county['value'] = df_bls_county['value'].apply(pd.to_numeric,
errors='coerce')

    # Get variable code
    df_bls_county['var_code'] = df_bls_county['series_id'].str[-2:]

    # Get area code
    df_bls_county['series_id'] = df_bls_county['series_id'].astype(str).str[3:].str[:-2]

    # Get FIPS code (as string to preserve initial zeros)
    df_bls_county['FIPS'] = df_bls_county['series_id'].str[2:7]

    #-------------------------------------------------------
    # Only keep rows corresponding to counties
    df_bls_county = df_bls_county.loc[df_bls_county['series_id'].str.contains('CN')]

    # Drop columns, reset index
    df_bls_county = df_bls_county[['series_id','year','period','value','var_code','FIPS']]
    df_bls_county.reset_index(drop=True, inplace=True)

    # Rename codes with variable names, rename columns
    df_bls_county['var_code'] = df_bls_county['var_code'].map({'03':
'Unemployment_Rate', '04': 'Unemployment',
                                                '05': 'Employment', '06': 'Labor_Force'})
    df_bls_county.columns = ['area_code', 'year', 'month', 'value','variable_name',
'FIPS']
    df_bls_county = df_bls_county.loc[df_bls_county['month']!='M13']

    # Convert month to numeric values
    df_bls_county['month'] = pd.to_numeric(df_bls_county['month'].str[1:])

    #-------------------------------------------------------
    # Merge area names and data
    df_bls_county = pd.merge(df_bls_county, df_areas, how='inner', on='area_code')

    # Convert to wide-format table
    df_bls_county = df_bls_county.pivot_table(values='value', index=['area_code',
'FIPS', 'countyname',
                                            'year', 'month'], columns='variable_name')
    df_bls_county.reset_index(inplace=True)
    df_bls_county.columns.name = None

    #-------------------------------------------------------
```

```
        return df_bls_county


    df_unemp_10_14 =
get_BLS_county_data('https://download.bls.gov/pub/time.series/la/la.data.0.CurrentU10
-14', df_areas)
    df_unemp_15_19 =
get_BLS_county_data('https://download.bls.gov/pub/time.series/la/la.data.0.CurrentU15
-19', df_areas)

    df_unemp_county = df_unemp_10_14
    df_unemp_county = df_unemp_county.append(df_unemp_15_19)

    df_unemp_county = df_unemp_county.sort_values(by=['area_code', 'year', 'month'],
axis=0)
    df_unemp_county = df_unemp_county[(df_unemp_county['year']>= int(start_year)) &
(df_unemp_county['year']<= int(end_year))]
    df_unemp_county = df_unemp_county[['area_code', 'countyname', 'year', 'month',
'Employment','Labor_Force', 'Unemployment_Rate']]

    census_columns = []
    for i in census_estimates.iloc[0]:
        census_columns.append(i)

    census_estimates = census_estimates.iloc[1:,:]
    census_estimates.columns = census_columns

    state_abb_dict ={'Alabama': 'AL','Alaska': 'AK','Arizona': 'AZ','Arkansas':
'AR','California': 'CA','Colorado': 'CO',
    'Connecticut': 'CT','Delaware': 'DE','District of Columbia': 'DC','Florida': 'FL','Georgia':
'GA','Hawaii': 'HI',
    'Idaho': 'ID','Illinois': 'IL','Indiana': 'IN','Iowa': 'IA','Kansas': 'KS','Kentucky':
'KY','Louisiana': 'LA',
    'Maine': 'ME','Maryland': 'MD','Massachusetts': 'MA','Michigan': 'MI','Minnesota':
'MN','Mississippi': 'MS',
    'Missouri': 'MO','Montana': 'MT','Nebraska': 'NE','Nevada': 'NV','New Hampshire':
'NH','New Jersey': 'NJ',
    'New Mexico': 'NM','New York': 'NY','North Carolina': 'NC','North Dakota': 'ND','Ohio':
'OH','Oklahoma': 'OK',
    'Oregon': 'OR','Pennsylvania': 'PA','Rhode Island': 'RI','South Carolina': 'SC','South
Dakota': 'SD','Tennessee': 'TN',
    'Texas': 'TX','Utah': 'UT','Vermont': 'VT','Virginia': 'VA','Washington': 'WA','West
Virginia': 'WV','Wisconsin': 'WI',
    'Wyoming': 'WY','Puerto Rico': 'PR'}
```

```
    census_estimates['state'] = census_estimates['Geography'].str.split(",").str[1]
    census_estimates['state'] =
census_estimates['state'].str.strip().replace(state_abb_dict)
    census_estimates['new geo'] = census_estimates['Geography'].str.split(",").str[0] + ", "
+ census_estimates['state']

    data_ = pd.merge(df_unemp_county, census_estimates, left_on = 'countyname',
right_on= 'new geo')

    data_['Employment to Population ratio'] = data_['Labor_Force']/data_['Population
Estimate (as of July 1) - 2017'].astype(float)

    # Total US country Employment-Population ratio
    #('https://data.bls.gov/timeseries/lns12300000')

    us_employment_ratio_df = pd.melt(us_employment_ratio_df, id_vars =
'Year',value_name = 'nation_wide monthly employment ratio', var_name = 'month',
value_vars= ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep','Oct', 'Nov', 'Dec'])

    # whole us unemployment
    #source: 'https://data.bls.gov/timeseries/lns14000000')
    # the name of the csv file is: Unemployment whole US.csv
    Us_unemployment_df = pd.melt(Us_unemployment_df, id_vars = 'Year',value_name
= 'nation_wide monthly unemployment rate', var_name = 'month', value_vars= ['Jan',
'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep','Oct', 'Nov', 'Dec'])

    # Employment level for people with only a highschool diploma or less
    #('https://data.bls.gov/timeseries/lns12000048')
    ## with a hs diploma
    hs_diploma_lf = pd.melt(hs_diploma_lf, id_vars = 'Year',value_name = 'lf_hs_dip',
var_name = 'month', value_vars= ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
'Sep','Oct', 'Nov', 'Dec'])

    # with NO hs diploma
    no_hs_dip_lf = pd.melt(no_hs_dip_lf, id_vars = 'Year',value_name = 'lf_nohs_dip',
var_name = 'month', value_vars= ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
'Sep','Oct', 'Nov', 'Dec'])

    Us_unskilled_labor_df = hs_diploma_lf
    Us_unskilled_labor_df['Unskilled_laborforce'] = hs_diploma_lf['lf_hs_dip'] +
no_hs_dip_lf['lf_nohs_dip']
    Us_unskilled_labor_df = Us_unskilled_labor_df[['Year', 'month',
'Unskilled_laborforce']]

    # getting data on US labor force to derive nationwide level of unskilled labor
```

```
    # https://www.bls.gov/lau/home.htm#cntyaa

    US_lf = pd.melt(US_lf, id_vars = 'Year',value_name = 'lf', var_name = 'month',
value_vars= ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep','Oct', 'Nov', 'Dec'])

    Us_unskilled_labor_df['share_unskilled_labor_US'] =
Us_unskilled_labor_df['Unskilled_laborforce']/US_lf['lf']
    Us_unskilled_labor_df = Us_unskilled_labor_df[['Year', 'month',
'share_unskilled_labor_US']]

    dict_month =
{'Jan':1,'Feb':2,'Mar':3,'Apr':4,'May':5,'Jun':6,'Jul':7,'Aug':8,'Sep':9,'Oct':10,'Nov':11,'Dec':
12}

    Us_unskilled_labor_df['month'] =
Us_unskilled_labor_df['month'].str.strip().replace(dict_month)
    us_employment_ratio_df['month'] =
us_employment_ratio_df['month'].str.strip().replace(dict_month)
    Us_unemployment_df['month'] =
Us_unemployment_df['month'].str.strip().replace(dict_month)

    data_['new index'] = data_['year'].astype('int').apply(str) +
data_['month'].astype('int').apply(str)
    Us_unskilled_labor_df['new index'] =
Us_unskilled_labor_df['Year'].astype('int').apply(str) +
Us_unskilled_labor_df['month'].astype('int').apply(str)
    us_employment_ratio_df['new index'] =
us_employment_ratio_df['Year'].astype('int').apply(str) +
us_employment_ratio_df['month'].astype('int').apply(str)
    Us_unemployment_df['new index'] =
Us_unemployment_df['Year'].astype('int').apply(str) +
Us_unemployment_df['month'].astype('int').apply(str)

    data_ = pd.merge(data_, Us_unskilled_labor_df, on = 'new index')
    data_ = pd.merge(data_, us_employment_ratio_df, on = 'new index')
    data_ = pd.merge(data_, Us_unemployment_df, on = 'new index')

    data_ = data_[['area_code','countyname', 'year', 'month_x', 'Unemployment_Rate',
'state','Employment to Population ratio', 'new index',
          'nation_wide monthly employment ratio','nation_wide monthly unemployment
rate', 'share_unskilled_labor_US']]

    data_['nation_wide monthly share_unskilled_labor']=
data_['share_unskilled_labor_US']
    data_['county monthly unemployment_rate']= data_['Unemployment_Rate']
```

```
    data_['county monthly employment ratio'] = data_['Employment to Population ratio']
    data_['month'] = data_['month_x'].iloc[:,0]

    data_ = data_[['area_code','countyname', 'state', 'year','month', 'county monthly
unemployment_rate', 'county monthly employment ratio', 'nation_wide monthly
employment ratio', 'nation_wide monthly unemployment rate',
                'nation_wide monthly share_unskilled_labor']]

    ## get county gdp yearly information from 2013-2015

    years = county_gdp_growth.iloc[1,5:]
    new_columns = ['FIPS', 'Countyname', 'Postal', 'LineCode', 'IndustryName', '2013',
'2014', '2015']
    county_gdp_growth.columns = new_columns
    county_gdp_growth = county_gdp_growth.iloc[2:,:][['FIPS','Countyname', 'Postal',
'IndustryName', '2013', '2014', '2015']]

    county_gdp_growth['2013'] = pd.to_numeric(county_gdp_growth['2013'], errors =
'coerce')
    county_gdp_growth['2014'] = pd.to_numeric(county_gdp_growth['2014'], errors =
'coerce')
    county_gdp_growth['2015'] = pd.to_numeric(county_gdp_growth['2015'], errors =
'coerce')

    county_gdp_growth =
county_gdp_growth[county_gdp_growth['Countyname'].isnull()==False]
    county_gdp_growth['FIPS'] = county_gdp_growth['FIPS'].astype('int')
    county_gdp_growth = county_gdp_growth.fillna(0.0)

    pivot_industries = pd.pivot_table(county_gdp_growth, index = 'FIPS', columns =
'IndustryName')

    renamed_columns = []
    j = 0
    for i in range(0, len(pivot_industries.columns.get_level_values(0))):
        level0 = str.strip(pivot_industries.columns.get_level_values(0)[i])
        level1 = str.strip(pivot_industries.columns.get_level_values(1)[i])

        new_column = "county " + level1 +" growth for " + level0
        renamed_columns.append(new_column)
    renamed_columns.append('FIPS')

    pivot_industries['FIPS'] = pivot_industries.index
```

```
    gdp_growth_df = pd.DataFrame(pivot_industries.values, columns =
renamed_columns)

    ## merging datasets on FIPS code (Federal Information Processing Standards code)
    gdp_growth_df['FIPS'] = gdp_growth_df['FIPS'].astype('int')

    data_['fips'] = data_['area_code'].str[2:7]
    data_['fips'] = data_['fips'].astype('int')

    data_ = pd.merge(data_, gdp_growth_df, left_on = 'fips', right_on = 'FIPS')
    data_['nation_wide monthly employment ratio'] = data_['nation_wide monthly
employment ratio']/100.0

    data_ = data_[['FIPS', 'countyname', 'state', 'year', 'month','county monthly
unemployment_rate', 'county monthly employment ratio',
          'nation_wide monthly employment ratio', 'nation_wide monthly unemployment
rate','nation_wide monthly share_unskilled_labor',
          'county Government and government enterprises growth for 2013','county Private
goods-producing industries growth for 2013',
          'county Private services-providing industries growth for 2013','county All
Industries growth for 2013',
          'county Government and government enterprises growth for 2014','county Private
goods-producing industries growth for 2014',
          'county Private services-providing industries growth for 2014','county All
Industries growth for 2014',
          'county Government and government enterprises growth for 2015','county Private
goods-producing industries growth for 2015',
          'county Private services-providing industries growth for 2015','county All
Industries growth for 2015']]

    ## Get state quarterly gdp information
    ## state gdp
    ##
https://apps.bea.gov/itable/iTable.cfm?ReqID=70&step=1#reqid=70&step=1&isuri=1

    state_gdp = state_gdp[['GeoFips', 'GeoName', 'Description',
'2012:Q4-2013:Q1','2013:Q1-:Q2', '2013:Q2-:Q3',
                  '2013:Q3-:Q4', '2013:Q4-2014:Q1','2014:Q1-:Q2', '2014:Q2-:Q3',
'2014:Q3-:Q4', '2014:Q4-2015:Q1',
                  '2015:Q1-:Q2', '2015:Q2-:Q3', '2015:Q3-:Q4',
'2015:Q4-2016:Q1','2016:Q1-:Q2', '2016:Q2-:Q3',
                  '2016:Q3-:Q4', '2016:Q4-2017:Q1','2017:Q1-:Q2', '2017:Q2-:Q3',
'2017:Q3-:Q4',
                  '2017:Q4-2018:Q1','2018:Q1-:Q2']]
```

```
    state_gdp['Postal'] = state_gdp['GeoName'].str.strip().replace(state_abb_dict)
    pivot_state_gdp = pd.pivot_table(state_gdp, index = 'Postal', columns =
['Description'])

    renamed_columns = []
    for i in range(0, len(pivot_state_gdp.columns.get_level_values(0))):
        level0 = str.strip(pivot_state_gdp.columns.get_level_values(0)[i])
        level1 = str.strip(pivot_state_gdp.columns.get_level_values(1)[i])

        new_column = "state "+ level1 +" growth for " + level0
        renamed_columns.append(new_column)

    pivot_state_gdp['state'] = pivot_state_gdp.index
    renamed_columns.append('state')
    State_gdp_df = pd.DataFrame(pivot_state_gdp.values, columns =renamed_columns)
    data_ = pd.merge(data_, State_gdp_df, on ='state')

    return data_
```

Appendix E: Python Code for Data Preparation, Clustering, and Visualization

```
#Data Preparation and Joining Datasets

import numpy as np
import requests
import pandas as pd
import re
import json

def get_yearly_data():
    # Getting the data from different urls within datausa.io api
    link1 =
'https://api.datausa.io/api/?show=geo&sumlevel=county&required=income_below_poverty'
    link2 =
'https://api.datausa.io/api/?show=geo&sumlevel=county&required=high_school_graduation,som
e_college'
    link3 =
'https://api.datausa.io/api/?show=geo&sumlevel=county&required=food_insecurity,median_hous
ehold_income,uninsured,other_primary_care_providers,mental_health_providers,violent_crime,
primary_care_physicians,food_insecurity,unemployment,income_inequality,population_that_is_
not_proficient_in_english,homicide_rate,violent_crime'
    link4 = 'https://api.datausa.io/api/?show=geo&sumlevel=county&required=pop,age'
    link5 = 'https://api.datausa.io/api/?show=geo&sumlevel=county&required=non_us_citizens'
    link6 =
'https://api.datausa.io/api/?show=geo&sumlevel=county&required=mean_commute_minutes,ow
ner_occupied_housing_units,median_property_value'
    link7 =
'https://api.datausa.io/api/?show=geo&sumlevel=county&required=social_associations'
    link8 =
'https://api.datausa.io/api/?show=geo&sumlevel=county&required=transport_bicycle,transport_c
arpooled,transport_drove,transport_motorcycle,transport_other,transport_publictrans,transport_t
axi,transport_walked,transport_home,workers'

    # Using json and requests to convert the website data to json
    json_link1 = requests.get(link1).json()
    json_link2 = requests.get(link2).json()
    json_link3 = requests.get(link3).json()
    json_link4 = requests.get(link4).json()
    json_link5 = requests.get(link5).json()
    json_link6 = requests.get(link6).json()
    json_link7 = requests.get(link7).json()
    json_link8 = requests.get(link8).json()
```

```
# zipping the json format to dictionaries
fcl = [dict(zip(json_link1["headers"], d)) for d in json_link1["data"]]
fc2 = [dict(zip(json_link2["headers"], d)) for d in json_link2["data"]]
fc3 = [dict(zip(json_link3["headers"], d)) for d in json_link3["data"]]
fc4 = [dict(zip(json_link4["headers"], d)) for d in json_link4["data"]]
fc5 = [dict(zip(json_link5["headers"], d)) for d in json_link5["data"]]
fc6 = [dict(zip(json_link6["headers"], d)) for d in json_link6["data"]]
fc7 = [dict(zip(json_link7["headers"], d)) for d in json_link7["data"]]
fc8 = [dict(zip(json_link8["headers"], d)) for d in json_link8["data"]]

# making dataframes from the dictionaries
fcl_df =pd.DataFrame.from_dict(fcl)
fc2_df = pd.DataFrame.from_dict(fc2)
fc3_df = pd.DataFrame.from_dict(fc3)
fc4_df = pd.DataFrame.from_dict(fc4)
fc5_df = pd.DataFrame.from_dict(fc5)
fc6_df = pd.DataFrame.from_dict(fc6)
fc7_df = pd.DataFrame.from_dict(fc7)
fc8_df = pd.DataFrame.from_dict(fc8)

# Getting the dataframes ready for merging
all_dfs = [fcl_df, fc2_df, fc3_df, fc4_df, fc5_df, fc6_df, fc7_df, fc8_df]

fcl_df = fcl_df.reset_index()
fc2_df = fc2_df.reset_index()
fc3_df = fc3_df.reset_index()
fc4_df = fc4_df.reset_index()
fc5_df = fc5_df.reset_index()
fc6_df = fc6_df.reset_index()
fc7_df = fc7_df.reset_index()
fc8_df = fc8_df.reset_index()

fcl_df['new index'] = fcl_df['year'].astype('int').apply(str) + fcl_df['geo'].apply(str)
fc2_df['new index'] = fc2_df['year'].astype('int').apply(str) + fc2_df['geo'].apply(str)
fc3_df['new index'] = fc3_df['year'].astype('int').apply(str) + fc3_df['geo'].apply(str)
fc4_df['new index'] = fc4_df['year'].astype('int').apply(str) + fc4_df['geo'].apply(str)
fc5_df['new index'] = fc5_df['year'].astype('int').apply(str) + fc5_df['geo'].apply(str)
fc6_df['new index'] = fc6_df['year'].astype('int').apply(str) + fc6_df['geo'].apply(str)
fc7_df['new index'] = fc7_df['year'].astype('int').apply(str) + fc7_df['geo'].apply(str)
fc8_df['new index'] = fc8_df['year'].astype('int').apply(str) + fc8_df['geo'].apply(str)


years = [2013,2014,2015,2016,2017]
new_index = []
geo = []
year = []
```

```
    for i, j in fc8_df[fc8_df['year']==2016]['geo'].iteritems():
        for k in years:
            new_index.append(str(k)+j)
            geo.append(j)
            year.append(k)

    dict_ids = {'new index': new_index,
                'id': geo,
                'year': year}

    id_df = pd.DataFrame.from_dict(dict_ids)

    ## merging to create a final dataset
    big_df = pd.merge(fcl_df, fc2_df, on ='new index', how = 'outer' )
    big_df = pd.merge(big_df, fc3_df, on ='new index', how = 'outer' )
    big_df = pd.merge(big_df, fc4_df, on ='new index', how = 'outer')
    big_df = pd.merge(big_df, fc5_df, on ='new index', how = 'outer')
    big_df = pd.merge(big_df, fc6_df, on ='new index', how = 'outer')
    big_df = pd.merge(big_df, fc7_df, on ='new index', how = 'outer')
    big_df = pd.merge(big_df, fc8_df, on ='new index', how = 'outer')
    big_df = pd.merge(id_df, big_df, on = 'new index', how = 'left')
    big_df['county_id'] = big_df['id']

    big_df['income_below_poverty'] = big_df['income_below_poverty']/big_df['pop']
    # cleaning unnecessary columns
    unnecessary_ = []
    for column in big_df.columns:
        if column.endswith('_x') or column.endswith('_y') or column == 'index' or column == 'year':
            unnecessary_.append(column)

    necessary_ =[]
    for column in big_df.columns:
        if not column in unnecessary_:
            necessary_.append(column)

    big_df = big_df[necessary_]
    big_df['year'] = big_df['new index'].str[:4]

    return big_df

## Getting FIPS identifier for counties

df = get_yearly_data()
df['FIPS'] = df['id'].str[7:]
```

```
county_ids =
['05000US25017','05000US25013','05000US36119','05000US36061','05000US36029','05000U
S42101','05000US10003',
'05000US39035','05000US39041','05000US12103','05000US39095','05000US26161','05000US
55025','05000US53033','05000US06013',
'05000US06085','05000US06073','05000US42003','05000US37119']

county_names = ['Middlesex County, MA','Hampden County, MA','Westchester County,
NY','New York County, NY','Erie County, NY',
'Philadelphia County, PA','New Castle County, DE','Cuyahoga County, OH','Delaware County,
OH','Pinellas County, FL',
'Lucas County, OH','Washtenaw County, MI','Dane County, WI','King County, WA','Contra Costa
County, CA','Santa Clara County, CA',
'San Diego County, CA','Allegheny County, PA','Mecklenburg County, NC']

affiliate_names = ['Framingham, MA','Springfield, MA','Fairview, Westchester County','New
York, NY','Buffalo, NY',
'Philadelphia, PA','Wilmington, DE','Cleveland, OH','Columbus, OH','Clearwater, FL','Toledo,
OH','Ann Arbor, MI',
'Madison, WI','Kent, WA','Walnut Creek, CA','Los Gatos, CA','San Diego, CA','Pittsburgh,
PA','Charlotte, NC']

membership_dictionary = {'affiliate_name': affiliate_names,'county_name':
county_names,'county_id': county_ids}
membership_df = pd.DataFrame.from_dict(membership_dictionary)

state_abb_dict ={'Alabama': 'AL','Alaska': 'AK','Arizona': 'AZ','Arkansas': 'AR','California':
'CA','Colorado': 'CO',
   'Connecticut': 'CT','Delaware': 'DE','District of Columbia': 'DC','Florida': 'FL','Georgia':
'GA','Hawaii': 'HI',
   'Idaho': 'ID','Illinois': 'IL','Indiana': 'IN','Iowa': 'IA','Kansas': 'KS','Kentucky': 'KY','Louisiana': 'LA',
   'Maine': 'ME','Maryland': 'MD','Massachusetts': 'MA','Michigan': 'MI','Minnesota':
'MN','Mississippi': 'MS',
   'Missouri': 'MO','Montana': 'MT','Nebraska': 'NE','Nevada': 'NV','New Hampshire': 'NH','New
Jersey': 'NJ',
   'New Mexico': 'NM','New York': 'NY','North Carolina': 'NC','North Dakota': 'ND','Ohio':
'OH','Oklahoma': 'OK',
   'Oregon': 'OR','Pennsylvania': 'PA','Rhode Island': 'RI','South Carolina': 'SC','South Dakota':
'SD','Tennessee': 'TN',
   'Texas': 'TX','Utah': 'UT','Vermont': 'VT','Virginia': 'VA','Washington': 'WA','West Virginia':
'WV','Wisconsin': 'WI',
   'Wyoming': 'WY','Puerto Rico': 'PR'}

## reading gdp data

county_gdp_growth = pd.read_excel(...)
state_gdp = pd.read_csv(...)
```

```
fips_postal_df= county_gdp_growth.drop_duplicates(subset = 'FIPS')[['FIPS','Postal']]

"""## county gdp dataset restructure
years = county_gdp_growth.iloc[1,5:]
new_columns = ['FIPS', 'Countyname', 'Postal', 'LineCode', 'IndustryName', '2013', '2014',
'2015']
county_gdp_growth.columns = new_columns
county_gdp_growth = county_gdp_growth.iloc[2:,:][['FIPS','Countyname', 'Postal',
'IndustryName', '2013', '2014', '2015']]
print(county_gdp_growth['FIPS'].iloc[0])
county_gdp_growth['2013'] = pd.to_numeric(county_gdp_growth['2013'], errors = 'coerce')
county_gdp_growth['2014'] = pd.to_numeric(county_gdp_growth['2014'], errors = 'coerce')
county_gdp_growth['2015'] = pd.to_numeric(county_gdp_growth['2015'], errors = 'coerce')

county_gdp_growth = county_gdp_growth[county_gdp_growth['Countyname'].isnull()==False]
print(county_gdp_growth['FIPS'].iloc[0])
county_gdp_growth = county_gdp_growth.fillna(0.0)

#pivot_industries = pd.pivot_table(county_gdp_growth, index = 'FIPS', columns =
'IndustryName')
county_gdp_growth = county_gdp_growth[county_gdp_growth['IndustryName']=='All Industries']
new_columns = ['FIPS', 'Countyname', 'Postal', 'LineCode', 'IndustryName', '2013', '2014',
'2015']
county_gdp_growth.columns = new_columns
county_gdp_growth = county_gdp_growth[['FIPS','Countyname', 'Postal', '2013', '2014', '2015']]

county_gdp_growth.columns = ['FIPS','Countyname','Postal', '2013 county gdp growth','2014
county gdp growth','2015 county gdp growth']


renamed_columns = []
j = 0
for i in range(0, len(pivot_industries.columns.get_level_values(0))):
    level0 = str.strip(pivot_industries.columns.get_level_values(0)[i])
    level1 = str.strip(pivot_industries.columns.get_level_values(1)[i])

    new_column = "county " + level1 +" growth for " + level0
    renamed_columns.append(new_column)

renamed_columns.append('FIPS')
pivot_industries['FIPS'] = pivot_industries.index
gdp_growth_df = pd.DataFrame(pivot_industries.values, columns = renamed_columns)

## Restructure state gdp dataset
state_gdp = state_gdp.loc[:, state_gdp.columns!='LineCode']
state_gdp = state_gdp[state_gdp['Description']=='All industry total']
```

```
state_gdp['Postal'] = state_gdp['GeoName'].str.strip().replace(state_abb_dict)
state_gdp = state_gdp.iloc[1:,:]
new_state_columns = ['GeoFips', 'GeoName', 'Description']
for i in range(3,len(state_gdp.columns)-1):
    new_state_columns.append(state_gdp.columns[i] + ' state gdp growth')

new_state_columns.append('Postal')
state_gdp.columns = new_state_columns

state_gdp = state_gdp[['GeoFips', 'GeoName', 'Description', '2012:Q4-2013:Q1','2013:Q1-:Q2',
'2013:Q2-:Q3',
                       '2013:Q3-:Q4', '2013:Q4-2014:Q1','2014:Q1-:Q2', '2014:Q2-:Q3', '2014:Q3-:Q4',
'2014:Q4-2015:Q1',
                       '2015:Q1-:Q2', '2015:Q2-:Q3', '2015:Q3-:Q4', '2015:Q4-2016:Q1','2016:Q1-:Q2',
'2016:Q2-:Q3',
                       '2016:Q3-:Q4', '2016:Q4-2017:Q1','2017:Q1-:Q2', '2017:Q2-:Q3', '2017:Q3-:Q4',
                       '2017:Q4-2018:Q1','2018:Q1-:Q2']]

state_gdp['Postal'] = state_gdp['GeoName'].str.strip().replace(state_abb_dict)
pivot_state_gdp = pd.pivot_table(state_gdp, index = 'Postal', columns = ['Description'])

renamed_columns = []
for i in range(0, len(pivot_state_gdp.columns.get_level_values(0))):
    level0 = str.strip(pivot_state_gdp.columns.get_level_values(0)[i])
    level1 = str.strip(pivot_state_gdp.columns.get_level_values(1)[i])

    new_column = "state "+ level1 +" growth for " + level0
    renamed_columns.append(new_column)

pivot_state_gdp['state'] = pivot_state_gdp.index
renamed_columns.append('state')
State_gdp_df = pd.DataFrame(pivot_state_gdp.values, columns = renamed_columns)
State_gdp_df.head(3)"""

# merging to create data frame of gdps
gdp_df = pd.merge(county_gdp_growth,state_gdp, on = 'Postal')

gdp_df['2013 county gdp growth']=gdp_df['2013 county gdp growth'].astype('float')
gdp_df['2014 county gdp growth']=gdp_df['2014 county gdp growth'].astype('float')
gdp_df['2015 county gdp growth']=gdp_df['2015 county gdp growth'].astype('float')

## reading non profit density

np_dens_df = pd.read_csv(...)

np_dens_df = pd.DataFrame(np_dens_df['STATE'].value_counts(), index =
np_dens_df['STATE'].value_counts().index)
```

```
np_dens_df.columns = ['nonprofits']
np_dens_df['state']=np_dens_df.index
np_dens_df['state']=np_dens_df['state'].str.strip()

## reading state jew population

jew_pop_df = pd.read_excel(...)
jew_pop_df['state'] = jew_pop_df['state'].str.strip().replace(state_abb_dict)
forgotten = pd.DataFrame.from_dict({'state':['AK','HI'],'pct jew':[0.8,0.5]})
jew_pop_df = jew_pop_df.append(forgotten)

## reading public transit stations per state
state_stations_df = pd.read_csv(...)
pivot_state_stations = pd.pivot_table(state_stations_df, index = 'State', values = 'Total Stations',
aggfunc = 'sum')
pivot_state_stations['state'] = pivot_state_stations.index
## reading square miles per state

sq_mi_state_df = pd.read_csv(...)

sq_mi_state_df['state'] = sq_mi_state_df['state'].str.strip().replace(state_abb_dict)
sq_mi_state_df = sq_mi_state_df[['state', 'sq mi']]

## Deriving public transit stations per square mile

pivot_state_stations = pd.merge(pivot_state_stations, sq_mi_state_df, on = 'state')
pivot_state_stations['square miles per station'] = pivot_state_stations['sq
mi']/pivot_state_stations['Total Stations']
# reading state population
state_pop_df = pd.read_csv(...)
state_pop_df['state'] = state_pop_df['state'].str.strip().replace(state_abb_dict)
state_pop_df['population'] = state_pop_df['population'].astype('float')

## Deriving non profits per person
np_density_df = pd.merge(np_dens_df,state_pop_df,on = 'state')
np_density_df['np_per person'] = np_density_df['nonprofits']/np_density_df['population']
np_density_df['persons per np']= np_density_df['population']/np_density_df['nonprofits']

## merging all dataframes together

df_wgdp = pd.merge(df, gdp_df, on = 'FIPS')
df_wgdp_np = pd.merge(df_wgdp, np_density_df, left_on = 'Postal', right_on = 'state')
df_wgdp_np_td = pd.merge(df_wgdp_np, pivot_state_stations, left_on = 'Postal', right_on =
df_wgdp_np_td_jp = pd.merge(df_wgdp_np_td,jew_pop_df, left_on='Postal',right_on = 'state')

## Getting most recent data available from the created data frame
columns_2016 = []
```

```
columns_2017 = []
for i in df_wgdp_np_td_jp.columns:
    a =
(df_wgdp_np_td_jp[df_wgdp_np_td_jp['year']=='2017'][i].isnull().sum())/len(df_wgdp_np_td_jp[df
_wgdp_np_td_jp['year']=='2017'])
    if a > 0.05:
        print(i, a)
    if a > 0.9:
        columns_2016.append(i)
    if a <1.0:
        columns_2017.append(i)
columns_2016.append('FIPS')

df_2017 = df_wgdp_np_td_jp[df_wgdp_np_td_jp['year']=='2017'][columns_2017]
df_2016 = df_wgdp_np_td_jp[df_wgdp_np_td_jp['year']=='2016'][columns_2016]
df_almost_ready = pd.merge(df_2017,df_2016, on = 'FIPS')




## Cleaning data
columns_w_null = []
for i in df_almost_ready.columns:
    a = df_almost_ready[i].isnull().sum()/len(df_almost_ready)
    if a> 0.0:
        columns_w_null.append(i)
        print(i, a)
numeric_columns = []
for i in df_almost_ready.columns:
    if df_almost_ready[i].dtype == 'float64':
        numeric_columns.append(i)

from sklearn.preprocessing import Imputer

# Taking care of missing data
imputer = Imputer(missing_values = 'NaN', strategy = 'most_frequent', axis = 0)

# since, the distributions for the data columns are not normal, in order to preserve the behavior
of the data,
# null values are filled in with the most frequent value of the data

imputer = imputer.fit(df_almost_ready[columns_w_null].values)
df_almost_ready[columns_w_null] =
imputer.transform(df_almost_ready[columns_w_null].values)

## reading state income adjusted for taxes and price parity
```

```
real_state_income = pd.read_csv(...)
real_state_income['state'].str.strip().replace(state_abb_dict)
real_state_income = real_state_income[['Postal','income adjusted for price parity and taxes']]
## real gdp per capita state
per_capita_gdp_state = pd.read_excel(...)
per_capita_gdp_state['State'].str.strip().replace(state_abb_dict)
per_capita_gdp_state = per_capita_gdp_state[['Postal', 'Per capita Real GDP in chained 2009
U.S. dollars']]
real_gdp_state = pd.read_excel(...)
real_gdp_state['Postal'] = real_gdp_state['State '].str.strip().replace(state_abb_dict)
real_gdp_state = real_gdp_state[['Postal', 'Real GDP in billion chained (2009) U.S. dollars']]

# merging datasets

df_almost_ready = pd.merge(df_almost_ready, real_gdp_state, left_on = 'Postal_x', right_on =
'Postal')
df_almost_ready = pd.merge(df_almost_ready,real_state_income, on = 'Postal')

df_almost_ready = pd.merge(df_almost_ready,per_capita_gdp_state, on = 'Postal')

## Deriving percent of non-driving commuters
df_almost_ready['pct non_driving_commuters'] = (df_almost_ready['transport_bicycle'] +
df_almost_ready['transport_carpooled']+df_almost_ready['transport_home']+df_almost_ready['tr
ansport_other']+
df_almost_ready['transport_publictrans']+df_almost_ready['transport_taxi']+
                                df_almost_ready['transport_taxi'])/df_almost_ready['workers']

## reading GDP total for counties
county_total_gdp = pd.read_excel(...)
county_total_gdp = county_total_gdp[county_total_gdp['IndustryName']=='All Industries']
new_columns = ['FIPS', 'Countyname', 'Postal', 'LineCode', 'IndustryName', 'gdp 2012','gdp
2013', 'gdp 2014', 'gdp 2015']
county_total_gdp.columns = new_columns
county_total_gdp = county_total_gdp[['FIPS','Countyname', 'Postal', 'gdp 2012','gdp 2013', 'gdp
2014', 'gdp 2015']]

## merge with original data frame
df_almost_ready = pd.merge(df_almost_ready,county_total_gdp, on = 'FIPS')
df_almost_ready['county per capita gdp 2015'] = df_almost_ready['gdp
2015']/df_almost_ready['county population']
df_almost_ready['county real gdp 2015'] = df_almost_ready['gdp 2015']

df_almost_ready['avg state quarterly gdp growth 2017-2018'] = (df_almost_ready['2017:Q1-:Q2
state gdp growth'] +
                                        df_almost_ready['2017:Q2-:Q3 state gdp growth'] +
                                        df_almost_ready['2017:Q3-:Q4 state gdp growth'] +
                                        df_almost_ready['2017:Q4-2018:Q1 state gdp growth'] +
```

```
                                        df_almost_ready['2018:Q1-:Q2 state gdp growth'])/5

non_numeric = []
for i in df_almost_ready.columns:
    if df_almost_ready[i].dtype != 'float64':
        non_numeric.append(i)

## reading pct Jewish population
pct_jew_county = pd.read_excel(...)
pct_jew_county = pct_jew_county[['ID','Jewish Population']]

# merge w original data frame
df_almost_ready = pd.merge(df_almost_ready, pct_jew_county, left_on = 'id', right_on = 'ID')
df_almost_ready['county pct jew']  = df_almost_ready['Jewish Population']

# selecting relevant columns from the original data frame
useful_columns = ['id', 'high_school_graduation', 'some_college',
       'food_insecurity', 'income_inequality', 'median_household_income',
       'mental_health_providers', 'other_primary_care_providers',
       'population_that_is_not_proficient_in_english',
       'primary_care_physicians', 'unemployment', 'uninsured', 'violent_crime',
       'social_associations', 'county_id', 'year', 'FIPS','2015 county gdp growth','np_per
person','stations per mile',
       'state pct jew','income_below_poverty','non_us_citizens',
       'mean_commute_minutes', 'median_property_value','income adjusted for price parity and
taxes',
       'pct non_driving_commuters','county per capita gdp 2015',
       'avg county gdp growth 2013-2015',
       'avg state quarterly gdp growth 2017-2018',
       'state Per capita Real GDP in chained 2009 U.S. dollars',
       'state income adjusted for price parity and taxes',
       'county real gdp 2015', 'Postal',
       'state Real GDP in billion chained (2009) U.S. dollars',
       'state pct jew', 'county pct jew']

df_almost_ready_2 = df_almost_ready[useful_columns]

# standardizing formatting
df_almost_ready_2['county per capita gdp 2015'] = df_almost_ready_2['county per capita gdp
2015'].astype('float64')
df_almost_ready_2['county real gdp 2015'] = df_almost_ready_2['county real gdp
2015'].astype('float64')

numeric_columns =[]
for i in df_almost_ready_2.columns:
    if df_almost_ready_2[i].values.dtype == 'float64':
        numeric_columns.append(i)
```

```
# calculating percentile ranks
df_almost_ready_pctle = df_almost_ready_2
for i in df_almost_ready_pctle[numeric_columns].columns:
   df_almost_ready_pctle[i] = df_almost_ready_pctle[i].rank(pct = 'True')

## converting the lower is better columns
lower_is_better_columns =
['food_insecurity','income_inequality','unemployment','uninsured','violent_crime',
                    'mean_commute_minutes','income_below_poverty','median_property_value']
for i in lower_is_better_columns:
   df_almost_ready_pctle[i] = 1 - df_almost_ready_pctle[i]

numeric_minus_jewpop = []
for i in numeric_columns:
   if (i!='state pct jew') and (i!='county pct jew'):
      numeric_minus_jewpop.append(i)

# remove duplicate columns created when merging
 df_almost_ready_2 = df_almost_ready_2.loc[:,~df_almost_ready_2.columns.duplicated()]

## Clustering and Visualizations

import matplotlib.pyplot as plt

df_raw = pd.read_csv(...)
df_pctle = pd.read_csv(...)

## affiliates
county_ids =
['05000US25017','05000US25013','05000US36119','05000US36061','05000US36029','05000U
S42101','05000US10003',
'05000US39035','05000US39041','05000US12103','05000US39095','05000US26161','05000US
55025','05000US53033','05000US06013',
'05000US06085','05000US06073','05000US42003','05000US37119']

county_names = ['Middlesex County, MA','Hampden County, MA','Westchester County,
NY','New York County, NY','Erie County, NY',
'Philadelphia County, PA','New Castle County, DE','Cuyahoga County, OH','Delaware County,
OH','Pinellas County, FL',
'Lucas County, OH','Washtenaw County, MI','Dane County, WI','King County, WA','Contra Costa
County, CA','Santa Clara County, CA',
'San Diego County, CA','Allegheny County, PA','Mecklenburg County, NC']

affiliate_names = ['Framingham, MA','Springfield, MA','Fairview, Westchester County','New
York, NY','Buffalo, NY',
'Philadelphia, PA','Wilmington, DE','Cleveland, OH','Columbus, OH','Clearwater, FL','Toledo,
OH','Ann Arbor, MI',
```

```
'Madison, WI','Kent, WA','Walnut Creek, CA','Los Gatos, CA','San Diego, CA','Pittsburgh,
PA','Charlotte, NC']

membership_dictionary = {'affiliate_name': affiliate_names,'county_name':
county_names,'county_id': county_ids}
membership_df = pd.DataFrame.from_dict(membership_dictionary)

# aggregating indicators under 6 categories
economy_employment = ['unemployment','county per capita gdp 2015','state Per capita Real
GDP in chained 2009 U.S. dollars',
                      'county real gdp 2015','state Real GDP in billion chained (2009) U.S.
dollars','median_household_income',
                      'state income adjusted for price parity and
taxes','income_below_poverty','income_inequality']

health_safety =
['primary_care_physicians','mental_health_providers','other_primary_care_providers','uninsured'
,
                'violent_crime']
education = ['high_school_graduation', 'some_college']
housing_living_facilitators = ['median_household_income',
'income_below_poverty','food_insecurity','mean_commute_minutes','pct
non_driving_commuters','stations per mile']
community_social_diversity = ['social_associations',
                              'np_per person','non_us_citizens']
jewish_population = 'county pct jewish'

columns_cluster = economy_employment + health_safety + education +
housing_living_facilitators + community_social_diversity

columns_cluster.append('county pct jew')

#Creating indicators for visualization summaries
df_pctle['average economy and employment rank'] = df_pctle[economy_employment].mean(axis
= 1)
df_pctle['max category economy and employment rank'] =
df_pctle[economy_employment].idxmax(axis=1)
df_pctle['min category economy and employment rank'] =
df_pctle[economy_employment].idxmin(axis=1)
df_pctle['average safety'] = 'violent_crime'
df_pctle['average housing and living'] = df_pctle[housing_living].mean(axis = 1)
df_pctle['max category housing and living rank'] = df_pctle[housing_living].idxmax(axis=1)
df_pctle['min category housing and living rank'] = df_pctle[housing_living].idxmin(axis=1)
df_pctle['average health_safety rank'] = df_pctle[health_safety].mean(axis = 1)
df_pctle['average education rank'] = df_pctle[education].mean(axis = 1)
df_pctle['average facilitators rank'] = df_pctle[facilitators].mean(axis = 1)
```

```
df_pctle['average community_social_diversity rank'] =
df_pctle[community_social_diversity].mean(axis = 1)
df_pctle['max category health_safety rank'] = df_pctle[health_safety].idxmax(axis=1)
df_pctle['min category health_safety rank'] = df_pctle[health_safety].idxmin(axis=1)
df_pctle['max category education rank'] = df_pctle[education].idxmax(axis=1)
df_pctle['min category education rank'] = df_pctle[education].idxmin(axis=1)
df_pctle['max category facilitators rank'] = df_pctle[facilitators].idxmax(axis=1)
df_pctle['min category facilitators rank'] = df_pctle[facilitators].idxmin(axis=1)
df_pctle['max category community_social_diversity rank'] =
df_pctle[community_social_diversity].idxmax(axis=1)
df_pctle['min category community_social_diversity rank'] =
df_pctle[community_social_diversity].idxmin(axis=1)




##Importing clustering algorithm library
import hdbscan

##Importing PCA library
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import seaborn as sns

#------------------------------------------------------------------------------

## Different iterations of feature engineering, clustering, and visualization to discover subgroups
of counties

#sc = StandardScaler()
X_trans = df_pctle[columns_cluster].values
clusterer = hdbscan.HDBSCAN(min_cluster_size=5, gen_min_span_tree=True)
clusterer.fit(X_trans)
clusterer.condensed_tree_.plot()

summary_columns = ['average economy and employment rank','violent_crime','average housing
and living',
            'average health_safety rank','average education rank','average facilitators rank',
            'average community_social_diversity rank','county pct jew']

#sc = StandardScaler()
X_trans = df_pctle[summary_columns].values
clusterer = hdbscan.HDBSCAN(min_cluster_size=5, gen_min_span_tree=True)
clusterer.fit(X_trans)
clusterer.condensed_tree_.plot()
clusterer.condensed_tree_.plot(select_clusters=True, selection_palette=sns.color_palette())
```

```
y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 (summary pctle)'] = y_hdbscan
df_raw['hdbscan feb20 (summary pctle)'] = y_hdbscan

y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 more bias (summary pctle)'] = y_hdbscan
df_raw['hdbscan feb20 more bias (summary pctle)'] = y_hdbscan

#sc = StandardScaler()
X_trans = df_raw[columns_cluster].values
clusterer = hdbscan.HDBSCAN(min_cluster_size=5, gen_min_span_tree=True)
clusterer.fit(X_trans)

clusterer.condensed_tree_.plot(select_clusters=True, selection_palette=sns.color_palette())

y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 raw'] = y_hdbscan
df_raw['hdbscan feb20 raw'] = y_hdbscan

#sc = StandardScaler()
X_trans = df_raw[columns_cluster].values
clusterer = hdbscan.HDBSCAN(min_cluster_size=100, gen_min_span_tree=True)
clusterer.fit(X_trans)
clusterer.condensed_tree_.plot()

clusterer.condensed_tree_.plot(select_clusters=True, selection_palette=sns.color_palette())

y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 more bias raw'] = y_hdbscan
df_raw['hdbscan feb20 more bias raw'] = y_hdbscan

#sc = StandardScaler()
X_trans = df_raw[columns_cluster].values
clusterer = hdbscan.HDBSCAN(min_cluster_size=50, gen_min_span_tree=True)
clusterer.fit(X_trans)
clusterer.condensed_tree_.plot()

clusterer.condensed_tree_.plot(select_clusters=True, selection_palette=sns.color_palette())

y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 intermediate bias raw'] = y_hdbscan
df_raw['hdbscan feb20 intermediate bias raw'] = y_hdbscan

sc = StandardScaler()
X_trans = sc.fit_transform(df_raw[columns_cluster].values)
clusterer = hdbscan.HDBSCAN(min_cluster_size=5, gen_min_span_tree=True)
clusterer.fit_predict(X_trans)
```

```
df_pctle['hdbscan feb20 scale raw'] = y_hdbscan
df_raw['hdbscan feb20 scale raw'] = y_hdbscan

clusterer = hdbscan.HDBSCAN(min_cluster_size=50, gen_min_span_tree=True)
clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 scale intermediate bias raw'] = y_hdbscan
df_raw['hdbscan feb20 scale intermediate bias raw'] = y_hdbscan

clusterer = hdbscan.HDBSCAN(min_cluster_size=100, gen_min_span_tree=True)
clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 scale more bias raw'] = y_hdbscan
df_raw['hdbscan feb20 scale more bias raw'] = y_hdbscan

sc = StandardScaler()
X_trans = sc.fit_transform(df_raw[columns_cluster].values)
pca = PCA()
pca2 = pca.fit_transform(X_trans)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance')
sum(pca.explained_variance_ratio_[:10])

sc = StandardScaler()
X_trans = sc.fit_transform(df_raw[columns_cluster].values)
pca = PCA(n_components = 10)
pca2 = pca.fit_transform(X_trans)
X_trans = pca2

clusterer = hdbscan.HDBSCAN(min_cluster_size=5, gen_min_span_tree=True)
y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 pca raw'] = y_hdbscan
df_raw['hdbscan feb20 pca raw'] = y_hdbscan

clusterer = hdbscan.HDBSCAN(min_cluster_size=50, gen_min_span_tree=True)
y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 pca intermediate bias raw'] = y_hdbscan
df_raw['hdbscan feb20 pca intermediate bias raw'] = y_hdbscan

clusterer = hdbscan.HDBSCAN(min_cluster_size=100, gen_min_span_tree=True)
y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 pca more bias raw'] = y_hdbscan
df_raw['hdbscan feb20 pca more bias raw'] = y_hdbscan

crank_columns = []
for i in columns_cluster:
    crank_columns.append(i + '_crank')
```

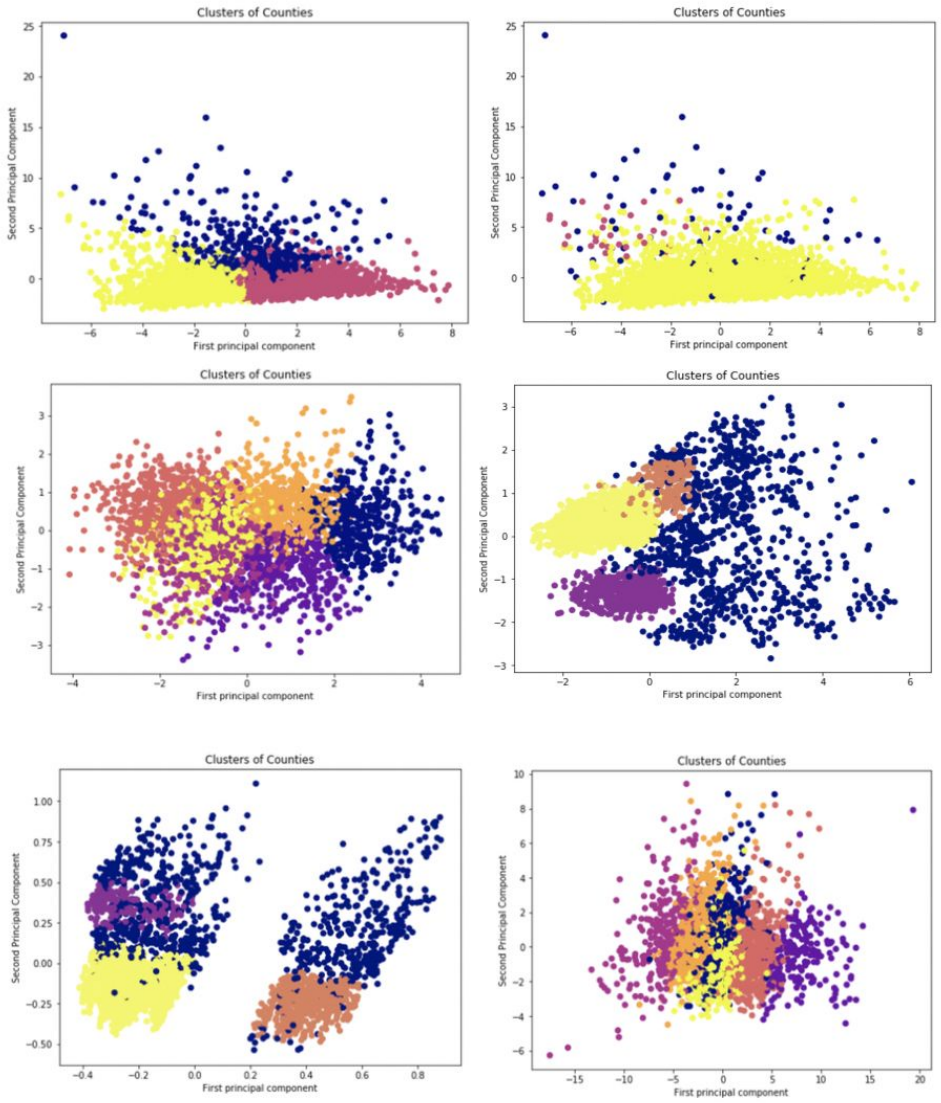
```
for i in columns_cluster:
    df_pctle[i + ''] = df_raw[i].rank()
columns_rank = []
for i in columns_cluster:
    columns_rank.append(i + 'rank')

sc = StandardScaler()
X_trans = sc.fit_transform(df_raw[columns_cluster].values)
pca = PCA()
pca2 = pca.fit_transform(X_trans)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance')

sum(pca.explained_variance_ratio_[:15])

sc = StandardScaler()
X_trans = sc.fit_transform(df_raw[columns_cluster].values)
pca = PCA(n_components = 15)
pca_raw_and_percentile = pca.fit_transform(X_trans)

# visualize clusters in pca plot
plt.figure(figsize=(8,6))
plt.scatter(pca_raw_and_percentile[:,0],pca_raw_and_percentile[:,1],c = df_pctle['is_affiliate'],
cmap='cool')
plt.xlabel('First principal component')
plt.ylabel('Second Principal Component')
plt.title('Clusters of Counties')

X_trans = pca_raw_and_percentile

clusterer = hdbscan.HDBSCAN(min_cluster_size=5, gen_min_span_tree=True)
y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 pca pct+raw'] = y_hdbscan
df_raw['hdbscan feb20 pca pct+raw'] = y_hdbscan

clusterer = hdbscan.HDBSCAN(min_cluster_size=50, gen_min_span_tree=True)
y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 pca intermediate bias pct+raw'] = y_hdbscan
df_raw['hdbscan feb20 pca intermediate bias pct+raw'] = y_hdbscan

clusterer = hdbscan.HDBSCAN(min_cluster_size=100, gen_min_span_tree=True)
y_hdbscan = clusterer.fit_predict(X_trans)
df_pctle['hdbscan feb20 pca more bias pct+raw'] = y_hdbscan
df_raw['hdbscan feb20 pca more bias pct+raw'] = y_hdbscan

X_trans = pca_raw_and_percentile
```

```
clusterer = hdbscan.HDBSCAN(min_cluster_size=5, gen_min_span_tree=True)
y_hdbscan = clusterer.fit_predict(X_trans)

clusterer.minimum_spanning_tree_.plot(edge_cmap='viridis',
                                      edge_alpha=0.6,
                                      node_size=80,
                                      edge_linewidth=2)
clusterer.single_linkage_tree_.plot(cmap='viridis', colorbar=True)

## display summary statistics for clusters
df_pctle['hdbscan feb20 pca pct+raw'].value_counts()
dicts_cluster_numeric = {}
#dicts_cluster_non_numeric = {}
for i in df_pctle['hdbscan feb20 pca intermediate bias pct+raw'].value_counts().index:
    k = df_pctle[df_pctle['hdbscan feb20 pca intermediate bias
pct+raw']==i][summary_columns].describe()
    #j = df_pctle[df_pctle['hdbscan feb20 (pctle)']==i][summary_non_numeric_columns].describe()

    dicts_cluster_numeric[i] = k
    #dicts_cluster_non_numeric[i] = j
dicts_cluster_numeric[2]
dicts_cluster_numeric[1]
dicts_cluster_numeric[0]
dicts_cluster_numeric[-1]

dicts_cluster_numeric = {}
#dicts_cluster_non_numeric = {}
for i in df_pctle['hdbscan feb20 (summary pctle)'].value_counts().index:
    k = df_pctle[df_pctle['hdbscan feb20 (summary pctle)']==i][summary_columns].describe()
    #j = df_pctle[df_pctle['hdbscan feb20 (pctle)']==i][summary_non_numeric_columns].describe()

    dicts_cluster_numeric[i] = k
    #dicts_cluster_non_numeric[i] = j

dicts_cluster_numeric[1]
dicts_cluster_numeric[-1]
dicts_cluster_numeric[0]

columns_cluster = columns_cluster + columns_rank

column_constructed_rank = []
for i in columns_cluster:
    column_constructed_rank.append(i+ '_crank')

columns_pctle_and_constructedr = columns_cluster + column_constructed_rank
```

```
for i in columns_cluster:
    df_pctle.loc[df_pctle[i]>=0.75, i+ '_crank'] = 4
    df_pctle.loc[(df_pctle[i]<0.75)&(df_pctle[i]>=0.50), i+ '_crank']=3
    df_pctle.loc[(df_pctle[i]<0.50)&(df_pctle[i]>=0.25), i+ '_crank']=2
    df_pctle.loc[df_pctle[i]<0.25, i+ '_crank']=1

for i in column_constructed_rank:
    df_raw[i] = df_pctle[i].values

# pca of constructed rank and pctle
sc = StandardScaler()
X_trans = sc.fit_transform(df_raw[columns_pctle_and_constructedr].values)
pca = PCA()
pca2 = pca.fit_transform(X_trans)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance')

# pca of constructed rank and pctle
sc = StandardScaler()
X_trans = sc.fit_transform(df_raw[column_constructed_rank].values)
pca = PCA()
pca2 = pca.fit_transform(X_trans)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('number of components')
plt.ylabel('cumulative explained variance')

## aggregating feature engineered ranks under the overarching indicators
economy_employment = ['unemployment_crank','county per capita gdp 2015_crank']

health_safety =
['primary_care_physicians_crank','mental_health_providers_crank','other_primary_care_provide
rs_crank','uninsured_crank']
safety = 'violent_crime_crank'
education = ['high_school_graduation_crank', 'some_college_crank']
housing_living = ['median_household_income_crank',
'income_below_poverty_crank','food_insecurity_crank']
facilitators = ['mean_commute_minutes_crank','pct non_driving_commuters_crank','stations per
mile_crank']
community_social_diversity = ['social_associations_crank',
                              'np_per person_crank','non_us_citizens_crank']
jewish_population = 'county pct jew_crank'
df_pctle['county pct jew_crank'] = 0
df_pctle.loc[df_pctle['county pct jew']>=0.75,'county pct jew_crank'] = 4
df_pctle.loc[(df_pctle['county pct jew']<0.75)&(df_pctle[i]>=0.50), 'county pct jew_crank']=3
df_pctle.loc[(df_pctle['county pct jew']<0.50)&(df_pctle[i]>=0.25), 'county pct jew_crank']=2
df_pctle.loc[df_pctle['county pct jew']<0.25,'county pct jew_crank']=1
```

```
i = 'violent_crime'
df_pctle.loc[df_pctle[i]>=0.75, i+ '_crank'] = 4
df_pctle.loc[(df_pctle[i]<0.75)&(df_pctle[i]>=0.50), i+ '_crank']=3
df_pctle.loc[(df_pctle[i]<0.50)&(df_pctle[i]>=0.25), i+ '_crank']=2
df_pctle.loc[df_pctle[i]<0.25, i+ '_crank']=1

df_raw['county pct jew_crank']=df_pctle['county pct jew_crank']
column_constructed_rank.append('county pct jew_crank')
columns_pctle_and_constructedr.append('county pct jew_crank')

df_pctle['average economy and employment rank'] = df_pctle[economy_employment].mean(axis
= 1)
df_pctle['max category economy and employment rank'] =
df_pctle[economy_employment].idxmax(axis=1)
df_pctle['min category economy and employment rank'] =
df_pctle[economy_employment].idxmin(axis=1)
df_pctle['average safety'] = df_pctle['violent_crime_crank']
df_pctle['average housing and living'] = df_pctle[housing_living].mean(axis = 1)
df_pctle['max category housing and living rank'] = df_pctle[housing_living].idxmax(axis=1)
df_pctle['min category housing and living rank'] = df_pctle[housing_living].idxmin(axis=1)
df_pctle['average health_safety rank'] = df_pctle[health_safety].mean(axis = 1)
df_pctle['average education rank'] = df_pctle[education].mean(axis = 1)
df_pctle['average facilitators rank'] = df_pctle[facilitators].mean(axis = 1)
df_pctle['average community_social_diversity rank'] =
df_pctle[community_social_diversity].mean(axis = 1)
df_pctle['max category health_safety rank'] = df_pctle[health_safety].idxmax(axis=1)
df_pctle['min category health_safety rank'] = df_pctle[health_safety].idxmin(axis=1)
df_pctle['max category education rank'] = df_pctle[education].idxmax(axis=1)
df_pctle['min category education rank'] = df_pctle[education].idxmin(axis=1)
df_pctle['max category facilitators rank'] = df_pctle[facilitators].idxmax(axis=1)
df_pctle['min category facilitators rank'] = df_pctle[facilitators].idxmin(axis=1)
df_pctle['max category community_social_diversity rank'] =
df_pctle[community_social_diversity].idxmax(axis=1)
df_pctle['min category community_social_diversity rank'] =
df_pctle[community_social_diversity].idxmin(axis=1)
```

Appendix F: Clustering Visualizations



Above are sample plots to visualize a couple of the clusters we found from the data. The axes are the two principal components of the data after manipulation (feature engineering) done before using the clustering algorithms to find subgroups. Each color is a different cluster found by the algorithm. The clusters in these plots separate counties by their performances in different indicators.

Appendix G: Refugee Employment Experiences/Skills

Accountant	Cook	Grass Cutter	Owner	Specialist physician
Accounting and Finance Director	Cooking	Grocer	Owner and Manager	Street vendor
Administrator	Counsellor	Gunner	Owner of the factory "Inter wood"	Student
An embroidering worker	Daily Labor	Homemaker	Paddy Field Worker	Tailors, dressmakers and hatters
Assistant in Laundry service	Daily laborer	Household and related work	Pastoralist	Taking Pictures (Cameraman)
Babysitter & Housewife	Daily Worker	Housekeeping	Pathologist	Taxi driver
Baker	Delivery Boy	Housewife	Plantation Worker	Teacher
Barber	Director of the Company "Algoritm"	Human Resources	Playing soccer	Teashop
Builder	Drinks Maker and General Worker in a car wash	Importing and selling used cars	Porter	Technical director
Building Care taker	Driver	Interpreter	Private Civil Engineer	Technical Manager
Building Services	Education Officer	Interpreter/ Translator for UNHCR and JVA	Project Manager	Trader
Bus driver	Electrician	Ironing	PSD	Transportation
Business woman	English Professor	Kitchen assistant	Ration Staff	Vendor
Car and Truck Driver	Event Planner	Kitchen Helper	Sales manager	Vet doctor
Car mechanic	Farmer	Laborer	Salesperson (clothes)	Waiter
Cashier	Fish Seller	Linguist	Salesperson: Consumable Commodities	Warehouse and Distribution Coordinator
Cleaner	Flight Attendant	Livestock keeping.	Secretary	Wash Cloths for a pay
Cleaners Supervisor	Football	Logistics and Store Assistant	Security Guard	Watch Mechanic (Owns a shop)
Cleaning and household work	Footballer	Manicurist	Security- sallay poort	Web developer
Clothes Merchant	Former Student	Mason	Senior House officer (internship)	Weldor
Co-Owner	Furniture	Mechanist	Senoir specialist	Windows glass installer
Company Owner	Furniture fixer	Miscellaneous Construction Worker	Septech lead sales	Witch Doctor (Dhami)
Construction Engineer	General Restaurant Worker	Needleworks	Sewing	
Construction labour	General Worker (Building construction)	Nurse	Shopkeeper	
Construction worker	General worker in restaurant	Nutshell cleaning	Social worker (NGO)	
Contractor	Geography Teacher	Odd jobs (Cut tree, cut flower, cut grass)	Sold fabrics	

Appendix H: Industry Categories

Administrative	Installation, Maintenance, & Repair
Architecture & Engineering	Law Enforcement Supervisors
Arts & Recreation	Life, Physical, & Social Science
Business & Financial Operations	Management
Cleaning & Maintenance	Management, Business, & Financial
Community & Social Service	Management, Business, Science, & Arts
Computer & Mathematical	Material Moving
Computer, Engineering, & Science	Natural Resources, Construction, & Maintenance
Construction & Extraction	Personal Care & Service
Education, Legal, Community Service, Arts, & Media	Production
Education, Training, & Library	Production & Transportation
Farming, Fishing, & Forestry	Protective Service
Fire Fighting Supervisors	Sales
Food & Serving	Sales & Office
Health Practitioners	Service
Health Technicians	Transportation
Healthcare Practitioners & Technical	
Healthcare Support	