CSE 564
Visualization and Visual Analytics

Final Project Report

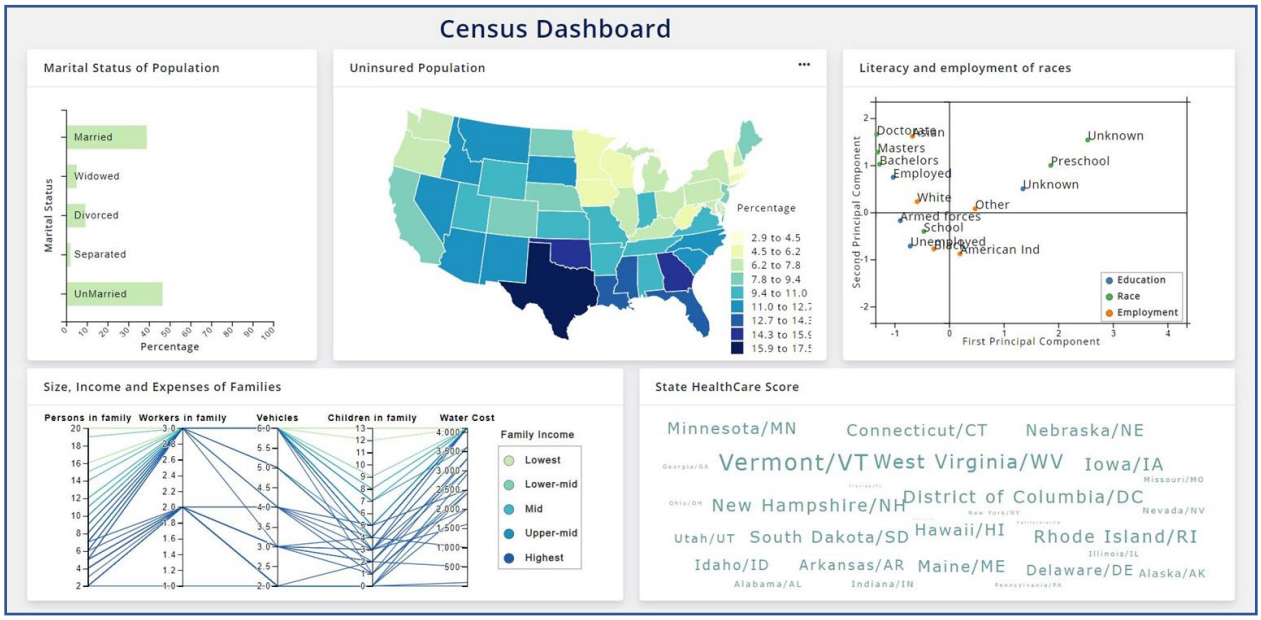
Adnan Vasanwalla (112674911)
Pratik Mukund Velhal (112675099)

Demo Video: Link

Introduction:

We have implemented a visual dashboard for the US census data to identify and elegantly visualize key trends and information across multiple domains. Census provides a snapshot of our nation- who we are, where we live, and so much more. It gets even more complicated when two domains are interrelated and data needs to be filtered out in order to understand and gain useful information. In such scenarios, visualizations can greatly simplify the task by presenting all the required information in an intuitive and simple manner. Data cleaning activities such as imputing and dropping null values where applicable, standardisation, normalisation, binning/smoothing were all carried out to make the data suitable for visualisation.

Snapshot of our dashboard:



Objectives:
The following are key objectives which we achieved through our visualization.

Marital Status of Population

Approach

For the marital status categorical column, we grouped the entire population by **Married** attribute and computed the population percentages of each attribute. Filters such as selection by state, race, employment and education are applied. Below code snippets show how we are processing the data to get the percentage values of each marital status column.

```
    cat_data = persondatadf[['ST','MAR','PWGTP', 'SCHL', 'RAC1P',  'ESR']]
    ## Filters applied here
    cat_data = cat_data[cat_data.MAR <6]
    cat_data.drop(columns=['ST', 'SCHL', 'RAC1P',  'ESR'], inplace=True)
    cat_data.dropna(inplace=True)
    sumvalue = cat_data['PWGTP'].sum()
    cat_data = cat_data.groupby(['MAR'], as_index=False).sum()
    cat_data["Percent"] =  (cat_data["PWGTP"]*100) / sumvalue
    cat_data.Percent = cat_data.Percent.round(2)
    cat_data['MAR'] = cat_data['MAR'].apply(lambda race: married.get(race))
    cat_data.drop(columns="PWGTP", inplace=True)
```

Insights

1. The percentage of the unmarried population is more than the married population for the entire US region.
2. Higher per capita income states such as California have equal percentages of married and unmarried populations.
3. Nebraska has the least percent of the separated population.

HealthCare, Economy and Immigration

Approach

We have plotted choropleth maps to depict the overall percentage of uninsured, unemployed and immigrant population of different states of US. For this, we grouped the data by states and the respective map attribute and calculated the percentages of such attributes for each state. Below Python code snippet shows how we are getting the economic data for our maps. Similar steps are performed health-care and immigration domains.

```
employmentdf = persondatadf.groupby(['ST','ESR'],
as_index=False)['PWGTP'].sum()
    employmentdf["Sum"] =
employmentdf.groupby('ST')['PWGTP'].transform('sum')
    employmentdf["Percent"] =  (employmentdf["PWGTP"]*100) /
employmentdf["Sum"]
    employmentdf = employmentdf[employmentdf.ESR == 3]
    employmentdf.drop(['ESR', 'PWGTP', 'Sum'], axis=1, inplace=True)
    employmentdf['ST'] = employmentdf['ST']*1000
    employmentdf.Percent = employmentdf.Percent.round(2)
```

Insights

1. The percentage of the uninsured population keeps increasing as we move from the northern region of the US to southern states.
2. Interior states have a higher unemployment rate compared to the coastal states.
3. Immigrant population is more in southern states as compared to northern states of the US.

Literacy and employment of races

Approach

For deriving relationships between races, employment and education, we used an MCA plot. For that we sampled 10% of data to reduce the processing time. These categorical columns were then fitted to the Multiple Correspondence Analysis (MCA) algorithm. A scatterplot is then used to show the high-dimensional data on 2-D screen. Brushing and linking on scatterplots unravels interesting insights discussed below. Below code snippet demonstrates the steps carried out to perform the algorithm and the optimization parameters used. Basically we first filter the data by state, perform random sampling to get 10% of data (stratified sampling would require manual intervention when state changes, hence not used), apply the MCA algorithm and convert to JSON format.

```
    cat_data = persondatadf[['SCHL', 'RAC1P',  'ESR', 'ST']]

    if state > 0:
        cat_data = cat_data[cat_data['ST'] == state]

    cat_data = cat_data.sample(frac=0.1)
    cat_data.drop(columns="ST", inplace=True)

    cat_data['RAC1P'] = cat_data['RAC1P'].apply(lambda race: races.get(race))
    cat_data['SCHL'] = cat_data['SCHL'].apply(lambda deg: degrees.get(deg))
    cat_data['ESR'] = cat_data['ESR'].apply(lambda emp: employment.get(emp))
    cat_data = cat_data.fillna("Unknown")
    mca = prince.MCA(
         n_components=2,
         n_iter=3,
         copy=True,
         check_input=True,
         engine='auto',
         random_state=42
    )
    mca = mca.fit(cat_data)
    mca_res = mca.column_coordinates(cat_data)
    mca_res.reset_index(inplace=True)
    types = {
        "SCHL": 1,
        "RAC1P": 2,
        "ESR": 3
    }
    mca_res['type'] = mca_res['index'].apply(lambda idx:
types.get(idx.split('_')[0]))
    mca_res['index'] = mca_res['index'].apply(lambda idx: idx.split('_')[1])
    mca_res.rename(columns={0: 'x', 1: 'y', index: 'cat'}, inplace=True)
```

Insights

For the entire US region:

1. **Asians** are highly educated (Doctorate, Masters, Bachelors) and employed.
2. **Blacks** and American Indians are more unemployed and have the highest educational attainment only till 10th grade.
3. **Whites** are generally employed and serve in the Armed forces.
4. Selecting just the **Asian** race or **White** race, we see equal percentage of married and unmarried people with percentages of widow, divorced are very less. While just selecting **Black** race or **American Indian** race, percentages of unmarried are high.
5. Selecting just the ones **employed**, have higher percentages of married than unmarried and vice versa for just unemployed.
6. Populations with educational attainment till school are 99% unmarried as expected. The 1% comprising married, widowed or separated could be the ones who may have completed their 10th grade after first marriage.
7. Selecting American Indian, Blacks, Unemployed, we see they are all unmarried (100%)
8. Unemployed Blacks have 20% married population whereas Unemployed Blacks with educational attainment till school level have no married population.

Size, Income and Expenses of Families

Approach

We used the housing dataset to group the population by their family income, binned it in 50 groups and plotted the data Number of family members, Number of workers in family, number of vehicles owned, and number of children in family and water costs incurred. Family income is colored by contrast to visualise it.

```
    para = housing[['NPF','WIF', 'VEH', 'NOC', 'FINCP', 'WATP', 'ST']].copy()
    if state > 0:
        para = para[para['ST'] == state]
    para = para[para['FINCP'].notna()]
    para['FINCP_grp'] = pd.cut(para['FINCP'], bins=50)
    grouped = para.groupby('FINCP_grp').agg({'NPF': 'max', 'WIF':
'max','VEH': 'max', 'NOC': 'max', 'WATP': 'max'})
    grouped.dropna(inplace=True)
    grouped = applyIncomeCategory(grouped)
    grouped = grouped[['NPF','WIF', 'VEH', 'NOC', 'WATP', 'Income_cat']]
    grouped.rename(columns={'NPF': 'Persons in family', 'WIF': 'Workers in
family', 'VEH': 'Vehicles', 'NOC': 'Children in family', 'WATP': 'Water
Cost'}, inplace=True)
```

Insights

Points 1-4 are for entire US region:

1. Number of persons and number of workers in a family are positively correlated. More the people, more the workers.
2. Number of Vehicles and number of children are positively correlated.
3. Number of children and Water cost are positively correlated.
4. Using conditioning and bracketing:

a. Conditioning on lesser Number of persons in the family: These people comparatively have a greater number of workers, own many vehicles, have fewer children and pay more water costs. Also their family income is higher. These represent the **elite class of population.**
b. Conditioning on lesser Number of persons in the family (> 14 people): These people have a greater number of workers, own many vehicles, have many children and pay more water costs. However their family income is very low. These could represent a **class of population ignorant to family planning.**
c. Conditioning on a higher number of vehicles shows positive correlation with children in family while for lower number of vehicles shows no correlation**.**

5. States with low employment levels generally have low family incomes.
6. **Outlier detected:** For the state of Nebraska, for lower water cost and lesser workers in family, people still own many vehicles (some even 6). Such states may not be having efficient public transport systems.
7. Comparison between states: Nebraska has a maximum of **9** people in family, **7** children in family and maximum water cost as **$2800**. While California has a maximum of **20** people in family, **12** children in family and maximum water cost as **$4000**

State HealthCare Score

Approach

To know the true extent of how well a state is performed on the three domains used in the map, we calculated a cumulative score taking into account various attributes for each domain. For each domain, we took a weighted sum of attributes and are displaying the score as a word cloud visualisation for states. Higher the score, bigger will be that state's word size. For economic data, this is how the economic score is generated. Steps remain the same for other two domains.

```
wordclouddf_economy =
persondatadf.groupby(['ST','COW','ESR','PERNP','POVPIP'],
as_index=False)['PWGTP'].sum()
    wordclouddf_economy['SCORE'] = 0
    wordclouddf_economy['SCORE'] = wordclouddf_economy.apply(lambda row:
((0.2 if row['COW'] < 8 else 0) + (0 if row['ESR'] == 3 else 0.2) + (0.25 if
row['PERNP'] > 50000 else 0) + (0.25 if row['POVPIP'] > 50 else 0)), axis=1)
    wordclouddf_economy = wordclouddf_economy.groupby(['ST'],
as_index=False)['SCORE'].mean()
    scaler = MinMaxScaler()
    wordclouddf_economy['SCORE'] =
scaler.fit_transform(wordclouddf_economy['SCORE'].values.reshape(-1,1))
```

Insights

1. States larger in size have a lower immigration, health-care and employment scores and vice versa.
2. The state - District of Columbia does have a high score for all the three categories.

Conclusion:

We were able to derive meaningful insights utilizing 24 attributes of the dataset (i.e., visualising 24-D data on 2-D screen). Brushing and linking on the plots will help the user to navigate the high dimensional data with ease. The UI theme colors are carefully chosen to help them relate data effectively and convey the desired information. This can go a long way in mining insights that may not have been covered above.