**Exercise-05**

Marissa Wiener

11/17/2016

1. **Dataset**

Link: <https://www.healthdata.gov/dataset/community-health-status-indicators-chsi-combat-obesity-heart-disease-and-cancer>

The link provided above allows access to a zipfile which contains datasets related to community health status indicators (CHSI). The information provided with these indicators are used to fight obesity, heart disease and cancer by tracking community health in all states. One of the reasons I chose this dataset was because it was very well documented. There are multiple sheets with explanations of the indicators and organization of the datasets. The datafile I focused on from the zipfile was the RISK\_FACTORS\_AND\_ACCESS\_TO\_CARE dataset. This dataset contains identifies the data elements and values in the Risk Factors and Access to Care indicator domain. The data contains 3141 observations and 31 columns. It shows percentages of certain health issues by county, the confidence interval associated with that percentage, and also raw numbers of individuals who meet certain attributes such as insurance status/type. In this analysis, I focus on the percentage of obesity by state and also number of Disabled and Elderly Medicare beneficiaries.

1. **Transformations**

I completed two types of transformations in this exercise. This data was very nicely formatted, and already had -1111.11 encoded for null values. This is convenient for analysis, but takes away from seeing the distribution of each column in trifecta, since for most columns there is a very large distance between -1111.11 and the valid ranges for that particular column. I used “set” statements in Trifecta to remove -1111.11 from the cells which contained that value, and just keep it as a missing value. The nice thing about the software is that it tells you how much of your data is missing, therefore this information could still be found when removing the imputed value for nulls. After this, I noticed that the Obesity column had the least amount of null values, so I decided to use these columns for transformations. It is difficult to interpret what these values mean without having a baseline to compare to. Therefore, I created a derived column which would take the obesity percentage relative to the mean of the column. If we had population distribution information, that would have made the relativities a bit more accurate, but this is the best option with the dataset provided. If this value is greater than 1, it means that the obesity percentage for that particular county is greater than the mean obesity percentage. If it is less than 1, it means that that particular county has a less than average obesity percentage. The next column that was derived was a counter for counties having a higher than average obesity percentage. If a particular county had a relativity higher than 1, this derived column would take 1, and if it was less than average it would take 0. Since we also have the confidence interval for the obesity percentage, I derived a column which measures the span of the confidence interval, so the difference between the CI Maximum Obesity value and the CI Minimum Obesity value. The larger the value of the span of the confidence interval, the less confident we are in the accuracy of the obesity measure. The relative CI length takes this calculated CI length and creates a relativity for CI length with the mean of all CI lengths. Moving away from the obesity percentage, I was also interested in comparing the amount of elderly people with Medicare and the amount of disabled people with Medicare in a particular county. Since the columns provided in the dataset only had raw numbers, it is difficult to make comparisons across counties, since that would not make sense given our lack of knowledge of the total population of each county. Therefore, I decided to create a ratio between the elderly people on Medicare and the disabled people on Medicare for each county. If the ratio is greater than 1, there are more elderly people on Medicare, and if the ratio is less than 1, there are more disabled Medicare policyholders. Using a set statement, I also replaced the encoded -2222.22 value for nulls with a blank cell.

1. **Script**

The script used to achieve the above description is shown below. The reason for the length is because the null encoding was removed for all of the columns. The derived columns are highlighted in yellow.

splitrows col: column1 on: '\r\n'

split col: column1 on: ',' limit: 30 quote: '\"'

header

set col: No\_Exercise value: null() row: ((-1150) <= No\_Exercise) && (No\_Exercise < (-1100))

set col: CI\_Min\_No\_Exercise value: null() row: ((-1150) <= CI\_Min\_No\_Exercise) && (CI\_Min\_No\_Exercise < (-1100))

set col: CI\_Max\_No\_Exercise value: null() row: ((-1150) <= CI\_Max\_No\_Exercise) && (CI\_Max\_No\_Exercise < (-1100))

set col: Few\_Fruit\_Veg value: null() row: ((-1150) <= Few\_Fruit\_Veg) && (Few\_Fruit\_Veg < (-1100))

set col: CI\_Min\_Fruit\_Veg value: null() row: ((-1150) <= CI\_Min\_Fruit\_Veg) && (CI\_Min\_Fruit\_Veg < (-1100))

set col: CI\_Max\_Fruit\_Veg value: null() row: ((-1150) <= CI\_Max\_Fruit\_Veg) && (CI\_Max\_Fruit\_Veg < (-1100))

set col: Obesity value: null() row: ((-1150) <= Obesity) && (Obesity < (-1100))

set col: CI\_Min\_Obesity value: null() row: ((-1150) <= CI\_Min\_Obesity) && (CI\_Min\_Obesity < (-1100))

set col: CI\_Max\_Obesity value: null() row: ((-1150) <= CI\_Max\_Obesity) && (CI\_Max\_Obesity < (-1100))

set col: High\_Blood\_Pres value: null() row: ((-1150) <= High\_Blood\_Pres) && (High\_Blood\_Pres < (-1100))

set col: CI\_Min\_High\_Blood\_Pres value: null() row: ((-1120) <= CI\_Min\_High\_Blood\_Pres) && (CI\_Min\_High\_Blood\_Pres < (-1100))

set col: CI\_Max\_High\_Blood\_Pres value: null() row: ((-1150) <= CI\_Max\_High\_Blood\_Pres) && (CI\_Max\_High\_Blood\_Pres < (-1100))

set col: Smoker value: null() row: ((-1150) <= Smoker) && (Smoker < (-1100))

set col: CI\_Min\_Smoker value: null() row: ((-1150) <= CI\_Min\_Smoker) && (CI\_Min\_Smoker < (-1100))

set col: CI\_Max\_Smoker value: null() row: ((-1150) <= CI\_Max\_Smoker) && (CI\_Max\_Smoker < (-1100))

set col: Diabetes value: null() row: ((-1150) <= Diabetes) && (Diabetes < (-1100))

set col: CI\_Min\_Diabetes value: null() row: ((-1150) <= CI\_Min\_Diabetes) && (CI\_Min\_Diabetes < (-1100))

set col: CI\_Max\_Diabetes value: null() row: ((-1150) <= CI\_Max\_Diabetes) && (CI\_Max\_Diabetes < (-1100))

derive value: CI\_Max\_Obesity - CI\_Min\_Obesity as: 'CI\_Length\_Obesity'

derive value: CI\_Length\_Obesity / mean(CI\_Length\_Obesity) as: 'Relative\_CI\_Length\_Obesity'

derive value: Obesity / mean(Obesity) as: 'Relative\_Obesity'

derive value: if(Relative\_Obesity >= 1, 1, 0) as: 'Obesity\_Indicator'

derive value: Prim\_Care\_Phys\_Rate / 1000 as: 'Prim\_Care\_Phys\_PCT'

set col: Elderly\_Medicare value: null() row: 0 > Elderly\_Medicare

set col: Uninsured value: null() row: Uninsured < 0

set col: Disabled\_Medicare value: null() row: Disabled\_Medicare < 0

derive value: Elderly\_Medicare / Disabled\_Medicare as: 'Elderly\_Disabled\_Medicare\_Ratio'

1. Evaluation

The results did what I expected them to. An example of how these new variables can be used in analysis and can be shown in the transformed\_data\_pivots.xlsx file. The data generated from the “Generate Data” step can be found in the RISKFACTORSANDACCESSTOCARE.csv file.