**uCapstone Project**

Analytics Anonymous

Scott Bagley

Austin Bertrand

Robert Michell

Dan Plumeri

Executive Summary

Major headwinds in twenty first century United States involve the impact of a shifting population age demographic and the impact on the US labor force and disease prevalence. Our team was provided with data sets related to these topics to initiate our analysis. This report will delve into our process of data exploration, recoding and cleaning, modeling, analysis and conclusions.

With the provided demographic and disease data, we have created a predictive model that looks to disease prevalence in the future. With our scored data, we will describe major findings - such as peak disease years and which diseases are projected to increase the most. As well as discuss what impact future trends may have on disease case.

In this report we will attempt to explain, predict, and plan for predicted changes in population based on projected population data provided. With the aid of further labor force data, we have created a second predictive model aimed at answering concerns related to the future of the US labor force. With our scored data we will answer questions related to industry segments and their relation to overall shifts in population age demographics.

As demographics rapidly shift within the US the problem of Social Security will become apparent. Based on our historic and future data we will compare collected revenues and paid benefits both historically and into the future. We will begin by looking into a historical comparison and then move into a future comparison based on our modeled results. In conclusion, we will use our historic and projected results to formulate potential scenarios to solve the inevitable gap between collections and benefits.

# Exploratory Data Analysis and Data Preparation

## Task 1 – DISEASE DIAGNOSIS

Exploration of the disease scenario began with an analysis of the disease data in Tableau. The data revealed that there were several missing data values as well as how we could “bucket” our data based on age groups. We then took the data out of tableau and transferred it into SAS 9.4.

Our next step was to begin recoding our rows into the age buckets that we determined. Age buckets were broken out into the following: 0 to 17 years, 18 to 24 years, 25 to 44 years, 45 to 64 years, 65 to 84 years, and 85 years and over. We concluded that these age groups would best serve as the different stages of one’s life that would reflect overall health in the most comprehensive way. However, not all the historic and projected data values fit our buckets perfectly. We therefor had to use the SQL lag function in order to create variables based off the existing ones. For instance, in order to get 65 to 84 years for future population, we had to take the 65 years and over and subtract it from the 85 years and over. This gave us 65 to 84 years which would fit in with both our historic and disease tables.

Once recoding and corresponding groupings of the values were completed, it was time to assess the missing values within the disease table. In order to accomplish this, we transposed the disease table, which allowed us to conduct mathematical equations to evaluate our missing data. In order calculate the missing values, we used IF statements with conditional parameters. For example, in gender, if a row had both all, and female, we calculated that all minus female populations equaled male population. If a row had female and male calculations, we calculated that all female plus male populations equaled both populations. If all population was missing, male and female population were missing. However, if we only had all population, we concluded that it would be best to leave female and male missing and impute the data in later steps through SAS Miner. Additionally, we determined that there was no need to include the All or both sexes category within the gender variable since we were looking for disease specific to a gender, not the overall population.

Once the historic, future, and disease case files were done, historic and future were imported into SAS Guide and joined. The code was then extracted and brought back SAS 9.4. Once the input file was complete, the final task was to create the scoring, or future file. To do this, a unique list of diseases was needed. We used the distinct function within the proc SQL step in order to create a Disease\_List table. This table was then cross joined with the scoring file in order to assign diseases to each of the future populations. With the cross join completed, both the scoring file and input file were ready to be brought into miner for further analysis.

## Task 2 - Labor Force

Exploration of labor force population began with uploading census\_population and project\_population into the Tableau software. The beginning Tableau data exploration gave us a general idea of the variables, which included date, gender, age\_group, and population\_in\_thousands. Browsing through the census\_population data we started out by looking at our first variable, date, which we knew had a date range from 1990-2010 and was in increments of one year. Moving on to the next variable, which was gender that was broken out into three categories, male, female, and both sexes. Having the three options for gender spurred the conversation within the group about whether all three will be necessary or not when looking at labor force population. The last variable we discussed was population\_in\_thounsands, since it was shown in thousands, we discussed the option to convert it into just overall population (population\_in\_thousands multipled by one-thousand), however we noticed that population was missing in some rows, so we noted that observation.

Our next step in the process was to begin to “bucket” the data into age categories that would match with our original labor\_force\_population file. We needed to ensure that our buckets could be applicable to both the historic census data and the projected population data. We settled on using the following age buckets: 16 to 24 years, 25 to 44 years, 45 to 64 years, and 65 years and over.

In order to apply these buckets to our data set, the next step in the process was to recode the data using the SAS 9.4 application. This process would prove to be one of our more difficult endeavors as we discovered numerous complications when creating the data sets. For instance, in the historic population data set, there was no easy way to bucket the 16 to 24 age group. Therefore, we had to use the lag function between the 16 and over row and the 18 and over row. The lag function took these two rows, subtracted their population counts and left us with a population of 16 to 17 years. By combining this with 18 to 24 years and recoding and grouping the rows using a proc SQL statement, we were able to create a 16 to 24 variable, where one had not existed before. The overall process for recoding the historic, projected, and labor force table was as follows: recode the corresponding rows that fit our buckets, (i.e. recode 45 to 49 as 45 to 64), group the columns using a sum function of the overall population within a proc SQL statement, and resort the data so that it was easier to interpret. Although this process may have appeared trivial, we ran into several issues when trying to determine the best grouping method for out SQL statements. We executed several iterations of the statements in order to see how grouping the data by different combinations of variables would affect the output. Once we were settled on what we thought was the best output achievable, we ran several proq freq and proq univariate functions to ensure that we did not have any variables that did not belong in the data set as well as to see if we had any missing data.

The next step was to then join the historic data with the labor force data as well as the future data with the industry data from the labor force table. To achieve the first part of this step, joining the labor force and historic data, we turned to Enterprise Guide to assist us without joins. One the join was completed, we extracted the code and placed it within our SAS 9.4 program, noting that the code was written in Guide. However, the next part of this step was not as easy. For the future, or scoring, file, we needed to have a list of the future population as well as industries that could be applied to each year. To do this, we took our recoded labor force file and used the distinct function within our SQL statement to pull out a distinct, non-repeating list of industries that were in the data set. We then took this Industry\_List file and used a cross join function to join it with the recoded future population data.

Additional cleaning was done when we realized that there were issues while attempting to run the file in miner. We took the input and scoring files out the program and adjusted them within SAS 9.4. Such cleaning included ensuring that all variables were of the same type (i.e. both in character and numeric notation), calculating all numeric values into the same format (i.e. keeping numbers in regular digits, not thousands), as well as dropping female and male genders. The decision of whether to drop male and female genders was once that we debated on for a while. However, we concluded that this number was not needed as we were simply trying to analyze the trends of the overall industry populations in the future, not the specifics of the gender breakdowns within those industries.

Once final recoding and cleaning of the files were complete, and we ensured that there was not any data that should be there, we sent the files into miner for further analysis.

## TASK 3 - Taxes and Social Security

In order to conduct our Social Security calculations, excel tables needed to be brought into SAS 9.4. To do this, we used SAS Guide to import the files and extracted the code from the program and brought it into SAS 9.4. Wage Limits did not need further cleaning, however, the contribution tables needed to be adjusted to fit our modeling. We realized that years which had no change in contributions were grouped together. For example, if there was no change in contributions in the years 1985, 1986 and 1987, the table was grouped as 1985-1987. However, this would not work for our modeling. We used DO loops in order to quickly fix issues where this situation existed. Once the tables were cleaned and recoded, they were ready for calculations which would answer our scenarios. Further details on the calculations can be found later in this report.

# Modeling Approach – DISEASE and Labor Force

Predictive modeling is used to generate forecasted results for a specific variable in a data set. Throughout the course of this project two models were built, one predicting projected disease outcomes and the other predicting labor force population.

## Disease Modeling

To begin the disease modeling we used the input file Disease\_Historic\_Join\_1 which had the joined the data tables disease\_case\_final and disease\_historic\_final. The file was assigned to the role of RAW so that we would later be able to use it for predictive training. Cases\_in\_1000s was assigned as the target variable, as we would be using this variable to be modeled and as the predictive variable. The date variable was initially set to rejected, but not dropped so the model could determine its level of importance.

From this point we began to conduct numerous observation steps within StatExplore node to ensure the data was ready to be modeled. The only change that we made to the properties was to include all the values because out data set was so small. The first step in the analysis was looking for missing values in the output window. After identifying there was none present in this set, we moved onto exploring each variable individually to ensure they were all being read in the right way and nothing was being excluded. From this point we examined skewness and kurtosis to identify how much abnormal disruption was affecting the data. We noticed our target variable, Cases\_In\_1000s had a skewness of 2.8 and a kurtosis of 8.2, which exceeds the normal range of –1 to 1 and –2 to 2. To correct this, our initial thought was to normalize the data, by making all values in thousands. This proved to be a valid way to correct some of the issues we were seeing. When this was done, we then attempted to transform the target variable using log and log 10 to make the data appear smaller. However, we ended up reverting back before the transformation because they did not provide any benefit. Once we were confident that nothing was missing and there were not variables being misread, we proceeded to partition the data. Several different iterations of data set allocation weights were attempted, but we settled on 70% train, 30% validation, and 0% test. At this point we were ready to start constructing models and the data was brought into a control point.

The first set of predictive models created were the regression’s, where our team ran every version of regression (Stepwise, Backward, Forward, Default). We had a general idea going into this that was the model type that we wanted to use, due to the simplicity and predictive power. Logistic regressions were considered but ruled out due to liner regressions being better able to model the relationship between multiple input variables, as well as the target variable being non-binary. Throughout different variations of models a variety of manually entered interaction variables were used, however we determined that the best method was to let miner auto select the variables. We also attempted to use polynomial terms in the model, but that only brought the R-square value higher and closer to overfitting. After numerous tweaks to our model we turned on two-factor interactions this allows the model to use all variables to create its own interactions. This ended up giving us the best R-square value of .9189, which we determined is just under the mark for what would be considered an “overfitted model”. We used this as our final model to predict future disease rates by demographic.

After the regression models were completed, a decision tree was created which uses a series of tests on a variable to predict possible outcomes. Very little was changed in term of properties for the decision tree. The main changes were an increase in the leaf size to 8, depth was increases to 10, the number of surrogate rules was set to 4 and the significance level was set at .2. Within the output of the model we saw the most important variable was Population\_in\_thousands, Age\_Group, and then Gender in that order.

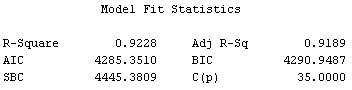
The last two models that we constructed were gradient boosting and neural network. For the gradient boosting we came in with expectations that this would not provide a high R-square value and we were correct, so we decided to move on. Neural Network is the most complex of the models and we decided it would be best if we ran with all the default properties set. We excluded this from our final comparison due to the overly complex results that we didn’t feel we would be able to and provide clear results about.

After all the models had been run, they were brought into a control point which fed into a model comparison. To our surprise miner selected decision tree as the best model, with the regression coming in as the worst. This was very confusing to us as a team as we thought the regression would be our best model, although it did not perform much worse than the decision tree. We decided as a team that we would rather work with the regression and we manually changed this and used the regression as our champion model as it works better with our data set due to population being on a positive incline. To ensure that the regression node that we wanted would be the selected model we disconnected the links from the other models and reran the model comparison node.

The data set disease\_historic\_join\_1 was then brought in to be used as the predictive data table for the regression model. We assigned the data table the role of SCORE as this is the file that would be getting affected by predictive modeling. No variables were dropped from this set and date, which was rejected before, was set to be used. We connected this with the score node, which was also connected to the model comparison. We then went to exported data and explored the scored data file in order to see our results.

After we completed this process, we examined the results and noticed the values repeated, as it seems the model was not correctly reading in the date. This created many problems for our team and stalled the project as no matter which model we used we would get the same results of repeating values. We later were able to identify that the reason for this was because we were dropping the variable date from the input variable, thus confirming our suspicion that date was not read. We were able to correct this and produce projected disease cases more accurately and date was no longer repeating.

At this point our model was predicting values for future disease cases, but we were still running into an issue. We were seeing a very small amount of the negative numbers coming through indicating to us there was a coding problem. We were able to deduce from this that when some of the gender cases got recoded there wasn’t a clear indication of how they should be sorted and they were subsequently dropped, thus giving us strange negative values. We were seeing a negative value for 0 to 17-year old's that had Dementia and Alzheimer's. We were not able to correct this due to timing, so we used context to identify that those values should most likely be zero. We finished with our champion model producing a R-squared value of .9189 as illustrated in the image below.



## Labor Modeling

Our modeling approach to the labor force data began by taking the OG\_Labor\_Join data file, which was the Census\_population and Labor\_force\_population data tables joined, to use as our input data set. The OG\_Labor\_Join dataset was imported into Enterprise Miner with the role of raw, to give us the ability to create predictive models with the data. However, before moving forward we looked through are variables to set out target variable as Sum\_Labor\_Force\_Pop. Scrolling through the remaining variables we noticed that gender was a variable that came into the input data, which we had previously discussed had no important in the modeling of labor. The reason we concluded that it had no importance in our modeling is because in the data cleaning stages of the project, we consolidated gender into just both sexes. The repetitiveness of both sexes for every entry in our import data was not necessary, so we rejected and drop the gender variable. At that moment, the input data roles and types looked good enough for us to connect a StatExplore node to the input file.

Since we were working with a smaller data set, we decide to change only one setting in the StateExplore node, which was setting number of observations as ALL rather than a small sample for generating our summary statistics. Running the node and viewing our output data we noticed that the skewness for population, an input variable and Sum\_Labor\_Force\_pop, the target variable was out of the normal range of –1 to 1. To move forward we noted that when we were on the transformation node to look at methods that would help improve our skewness on those variables. Other than our skewness concerns, the summary statistics confirmed that we had no missing variables and that the correlation between population and our target variables was 0.71. Acknowledging and noting our findings in our StatExplore node we decided to connect a data partition node. In the beginning we decided to leave the default data partition settings of 40% training, 30% validation, and 30% testing, knowing after our initial model creation we would ultimately adjust these allocation settings to 70% training, 30% validation, and 0% testing.

Knowing that we were going to test multiple models we used a control point node that would connect to our decision tree, regression, neural network, and gradient boosting nodes. The first model we attempted was the decision tree based on its simplicity and easy understanding of the process and output. Before running the node, we looked through the settings, examining what needed to be changed. We started out by changing the significance level to 0.05 and the assessment measure for the subtree to be, average squared error because are target variable was an interval. The model showed us in the output that the age\_group and industry variables were the most important variable to our target variable. The results surprisingly stated that the overall population variable was the least important variable, coming in at a level of 0.06. Without spending much time on each model, the first run through we moved on to our next model, which was the regression.

Creating the regression model, we were initially under the impression that this was going to be our best model out of the pack, so naturally we spent more time on changing settings and reviewing the results of the model. However, before connecting the regression node to our control point, we added a transformation node that would ultimately connect to the regression node. The transformation node was insert between our control point and regression node to help adjust our skewness on overall population and Sum\_Labor\_Force\_pop. Based on the both variables be interval variables that were skewed, we decided to use log10 method for transforming the variables. Using the log10 method would give us the base-10 logarithm of the original variable, which would help our skewed data.

Connecting the regression node to our transformation node, we knew we had to change the regression from logistic to linear because our target variable was nonbinary. Second setting that was looked at was the optimization method, giving us options of stepwise, forward, backward, or none. Reading more into the difference between each method we decided on creating four different regression nodes that used each of the four methods. The logic behind this was to compare the changes each method had on the results of the model without having to go back each run of the model to compare the results, this way all the results for each model would show up in our model comparison node.

The last two models we created were the neural network and gradient boosting models. In the creation of the neural network as a good we concluded that the model will create helpful results to compare to other models, but in the end, we wouldn’t use this model as our champion model. The reasoning behind this decision was that even if it was our champion model in the model comparison node, explaining the process and results would be unclear to our audience. Not intending to use the model we went ahead an connected the control point to our neural network node with leaving the default setting. A similar situation happened with the implementation of the gradient boosting model, where we used the default setting of the node.

Connecting all our models to our model comparison node, it was time to see the champion model, which turned out to be the decision tree model. The interesting part of the results of the model comparison node was not only was the decision tree our best model, but all four of the regression we at the bottom of the list. It was interesting to the group because initially we thought they would be the best model of the bunch. Without going back and reviewing ways to improve each models r squared value, we decided to import our Cross\_Join\_Labor\_2 file. The cross\_join\_labor\_2 file was a joined file that included the projected population and labor\_force\_population. After importing the dataset into miner with a role of score we finished our diagram by connecting that dataset and the model comparison node to the score node and ran the diagram.

The output of the score node provided us with repetitive output of the same numbers for each year and age group. For example, 2015 age group 16 to 24 working in the management industry was the same predicted value for 2020 age group 16 to 24 working in management. This issue created much concern and confusion throughout the group, on why this was happening. After many discussions of running through possible reasons for this repetition, we figured out that even though the decision tree was our champion model it wasn’t the appropriate model to use to score our data. The reasoning behind this was that since our target variable was an interval it wasn’t providing us appropriate predictions to score the project population data set.

Dropping the decision tree from being our champion model we believed that the regression was going to be the best model to score our project population data. However, after running just the regression node into the score node we were still confronted with the same repetition in predicted values. Running out of ideas as to why this issue was still occurring in our predict values, we decided to continuously change settings within the original data and the regression model. This exploration led us to drop date from our original dataset and create interaction terms in our regression mode. The date variable we believed was causing the repetition in predicted variables and based on the results of the regression model date provided no predictive value. The interaction terms we initially created by changing two-factor interactions from no to yes, which systematically creates interaction terms. These two changes to our diagram eliminated our repetition issue in our predicted value results.

Even though, our results were coming out positive we still had a concern with our r-squared value being so close to 1, meaning our model was currently overfitted. The first approach we took to reduce are r-squared value was to remove the transformation of population and sum\_labor\_force\_pop, which resulted in a slight reduction in r-squared. Second approach was to create our own interaction terms in the regression model rather than use the system generated interaction terms. Though still not a huge drop in our r-squared value we were able to get it to 0.9922; this turned out to be the best r-square value we could produce.

Concluding the modeling process for labor force population, we proceed with our scored data from the regression model, even though it was overfitted. The reasoning behind proceeding with the regression model was because it gave us reasonable labor force population predictions, in comparison to our other models that provided negative and volatile predictions.

# Revenue and Payout for Social Security 2000-2013

**Introduction**

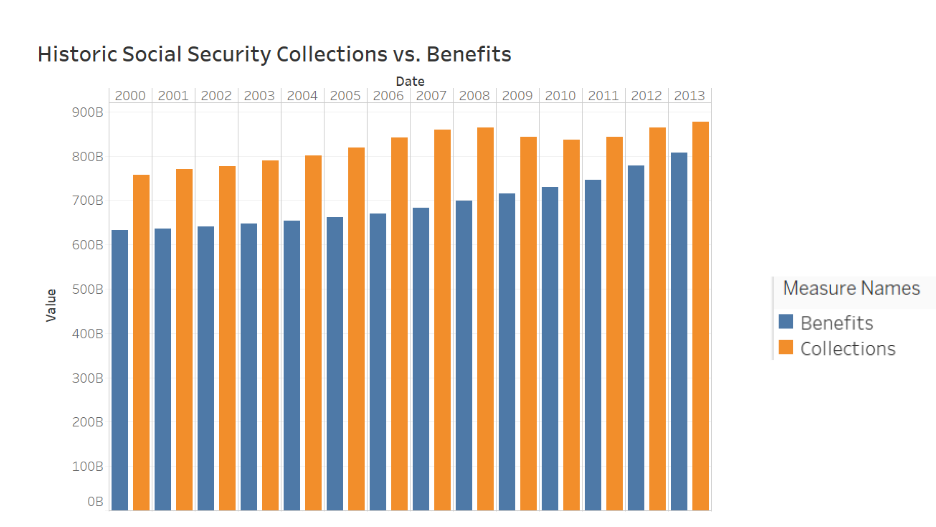
The US population is aging rapidly and with it the cost of social security benefits paid to retired workers, the disabled, widows, and children of the deceased. As the population ages, younger demographics continue to shrink and with them the number of prime earning individuals that pay into social security. The data provided in this case allows us to observe the historic standing of social security collections versus benefits; while the results of our predictive modeling provide the means for looking into the future of social security.

**Historic Taxes and Social Security Process**

We were provided with census population, labor force population, and occupation and pay data, with which we aggregated and joined using base SAS and proc SQL. Numeric values were left in common units. To match each occupation title with each year’s corresponding employee social security rate, employer social security rate, and wage limit, we joined our historic labor table with the provided social security tables. The resulting table (years 2000 - 2013) provided columns labeled: year, age group, industry, occupation title, percent employment, labor force population, overall population, annual mean wage, social security employee rate, social security employer rate, and wage limit. With this table in hand, we began to calculate the total collections and benefits for the years 2000 - 2013. We began by dividing each social security rate by 100 to gain a decimal percentage, while computing an additional column with the name “combined social security rate” (employee rate + employer rate). We then computed another additional column by multiplying each columns percent employment \* the overall population for the corresponding age segment – this resulted in a column named “active contributors” meant to represent the number of workers in each segment eligible to pay social security taxes. The next step was to create an if/else statement that would multiply “combined social security rate” by the corresponding “annual mean wage” IF the annual mean wage was less than the “wage limit” ELSE the calculation would multiply “combined social security rate” by the “wage limit” - all resulting in one column with an accurate annual social security contribution named “total contributions”. We then multiplied “total contributions” by “active contributors” in each column in order to get a numeric value for each column representing the total combined contributions for all the workers in each occupation – this column was labeled total ss contribution. With proc SQL we were able to further aggregate the data resulting in a table with two columns (year and total ss contribution) each year accompanied by that year's total social security collections in dollars.

We then went back to our census population data to isolate all those over retirement age, so that we could compute the amount being paid out in benefits. Based on the data structure provided we were unable to determine population over 67, and instead used all population over 65 to represent those collecting benefits. Using SAS and proc SQL, we isolated the total population over 65 for each year (2000-2013), we then multiplied the population column by the annualized benefits (population \* 1503 \*12) resulting in the total benefits paid for each year. Finally, we used a basic join statement to join the collections and benefits tables resulting in a table named **ss\_calc.historic\_ss\_final** with three columns: date, ss\_payout (benefits), and total\_contributions (collections).

**Figure 1.A**



**Historical Social Security Findings**

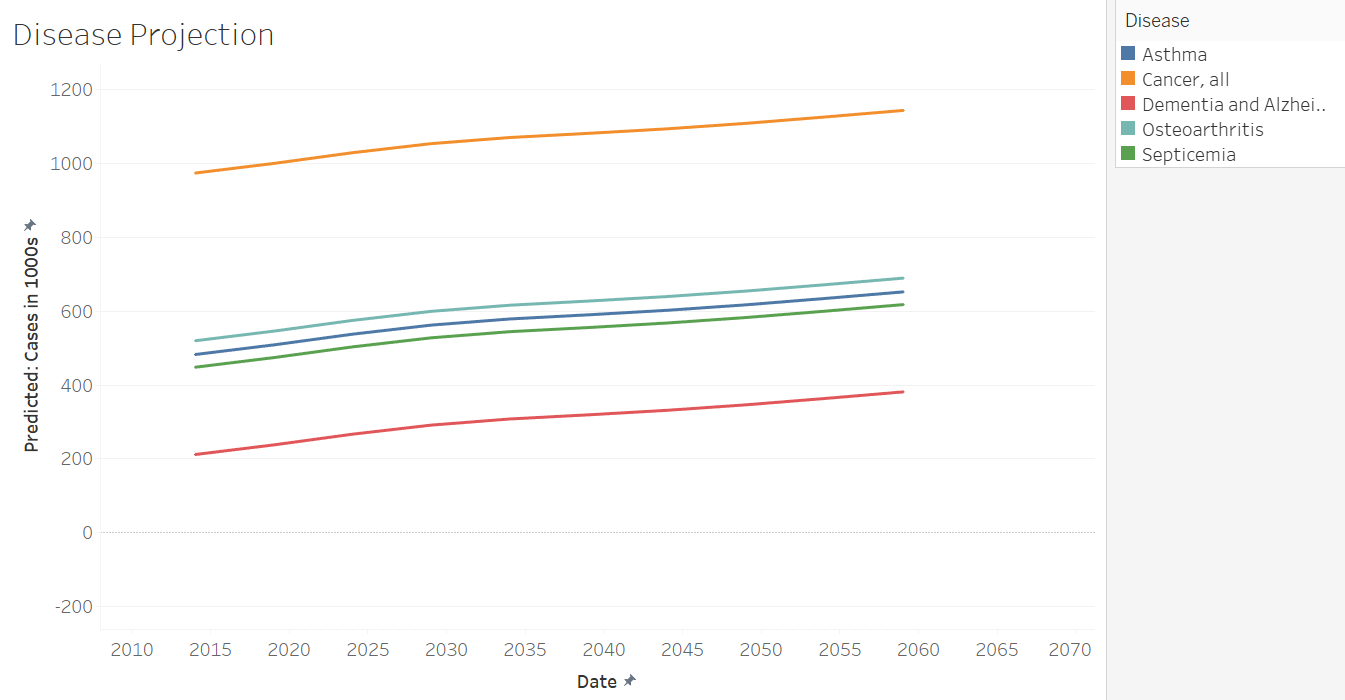
As we can see in **Figure 1.A** the disparity between collected revenues and paid benefits will become an issue even sooner than some may have assumed. According to the historic data provided, collections remained stagnant in the observed periods while benefits picked up momentum and grew steadily into 2013. In 2013 the gap between collections and benefits was only about 50B dollars or about 6%. If this trend were to continue, we would expect benefits to surpass collections earlier than 2020. This is a concerning trend of the United States as the baby boomers' segment is not expected to reach full retirement age until about 2030.

# Generalization of Results

## TASK 1 - Disease DIAGNOSIS

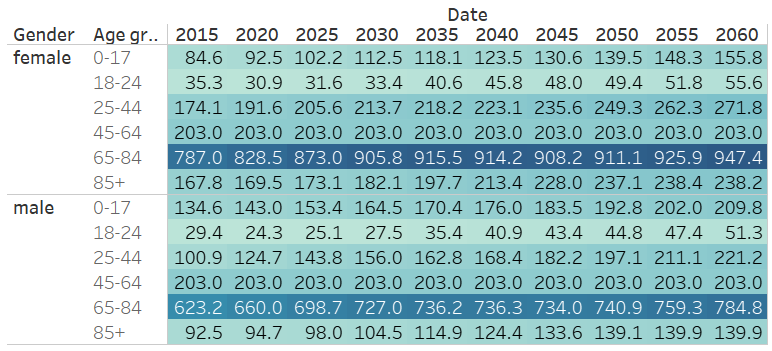
As population is on a steady rise it is no surprise that the amount of people with disease's is also increasing year after year as indicated in figure 2.A. The model that we created depicts the degree to which they are affecting people in various years in various stages of their lives. It was evident to us that current historical disease trends we were observing are no longer the standard as the future values were going to be heavily driven by an aging population as 65 to 84 year old's are causing the data to spike.

**Figure 2.A**

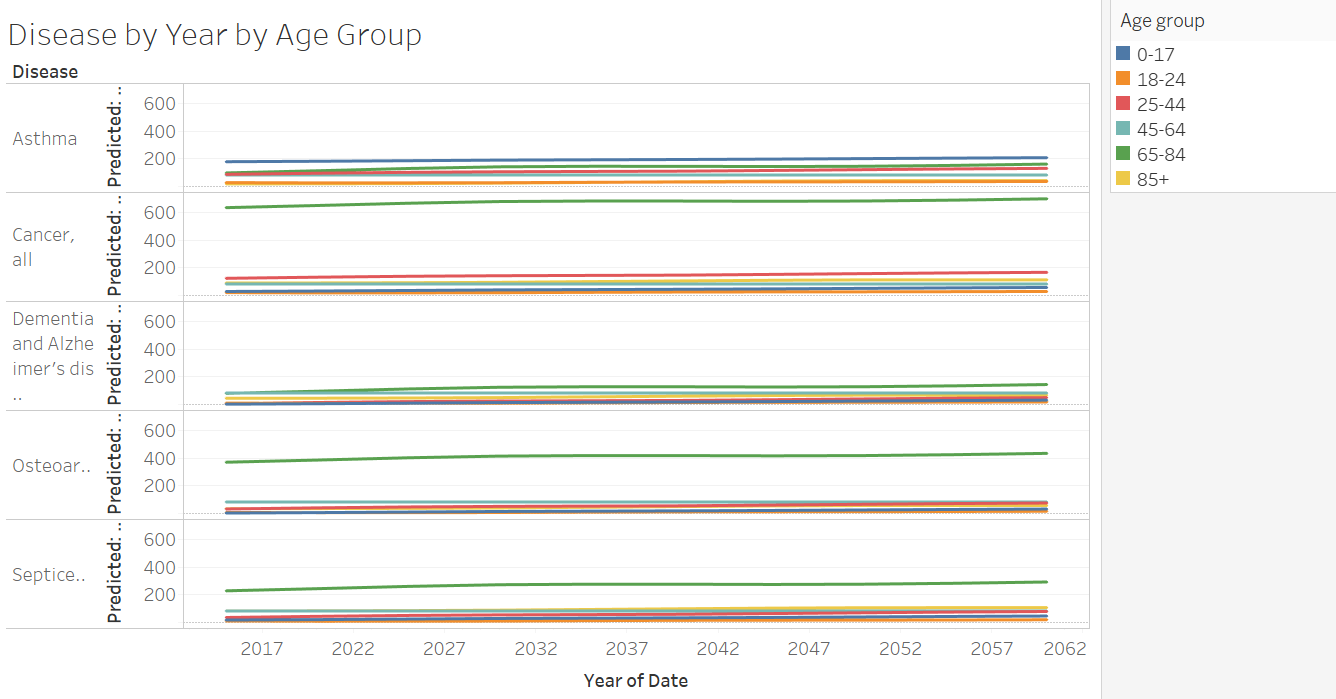


Expanding on that in figure 2.B the data is indicating that there is an increase in asthma, cancer, osteoarthritis, dementia, and septicemia in both males and females from 65-84. This confirmed our initial thoughts that an aging population would make the whole population appear more ill, the younger people remained at a steady growth while it seemed that the older population got infected by these disease at a much higher rate.

**Figure 2.B**

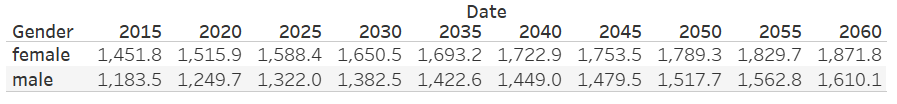


In figure 2.C the disease case by the year by age group, in this chart it is indicating to us that there is a constant growth through all diseases, not matter the age group or the year. After analyzing this chart, we have concluded that this information may be slightly incorrect as our model might have been overfitted with the R-squared value was at .9189. We were able to identify early on that constant growth at this level was not something that was realistic. It was being driven by a model that was trying to perfectly fit each price of data.



Looking at the gender comparison pictured below in figure 2.D, we have concluded that on average woman tend to have more disease cases than men. This is a trend that we have seen all throughout the historical data and well into the projected data. The disease’s that were listed, like osteoarthritis, are more prone in woman thus making woman appear with more diseases.

**Figure 2.D**

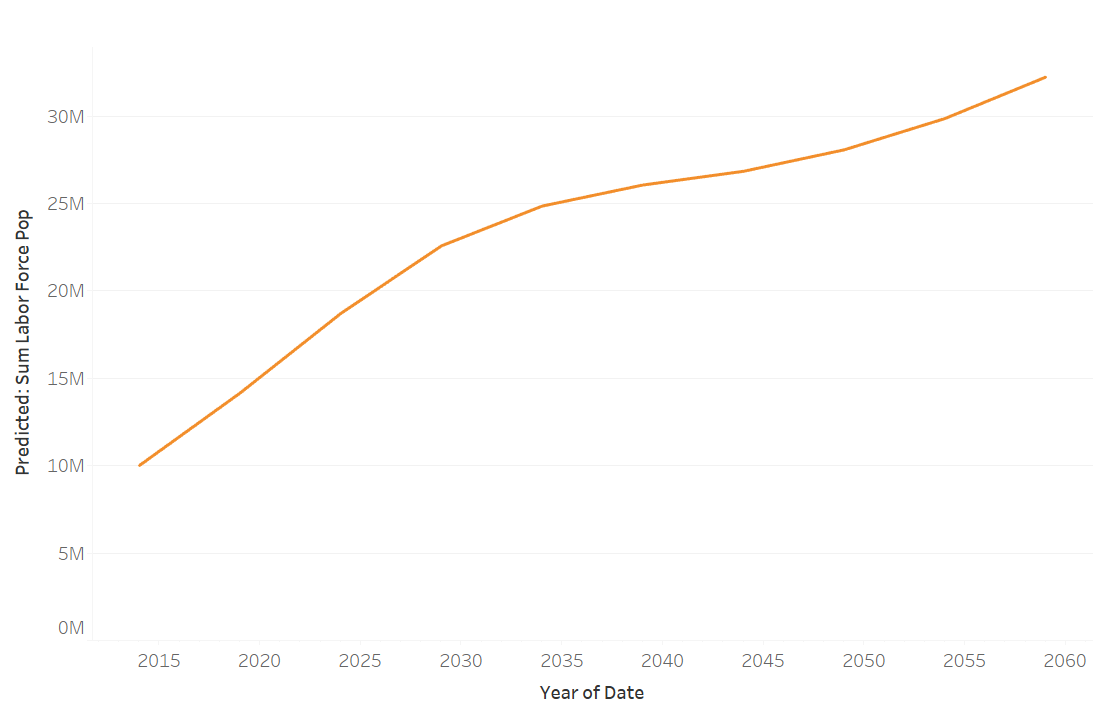


As time progress and technological advancements continue to rise, the amount of people that we see suffering from diseases will decrease. It is likely that we will see disease prevention start to play a larger role in the medical field as technology such as CRISPR begins to enter. CRISPR is a genetic sequencing tool that is used to alter DNA sequences and gene functions. This tool will be essential in prevention of cancer which happens to be the most prevalent disease within our data however with the high costs and experimental phase this will most likely not be implemented for year come. As time progresses CRISPR will become generic and costs will drop as soon as the useful life of the patents run out. In addition, technological advances will continue to happen, making health care not only more affordable, but more effective and better suited to deal with the future.

Task 2 - Labor Force

As we investigated the direction of the labor force population with our scored data, we looked at overall predicted labor force population for each age group. The age group that stuck out was the 65+ years old, who saw a significant year over year increase in labor force population. Refer to the significant increase in the chart below.

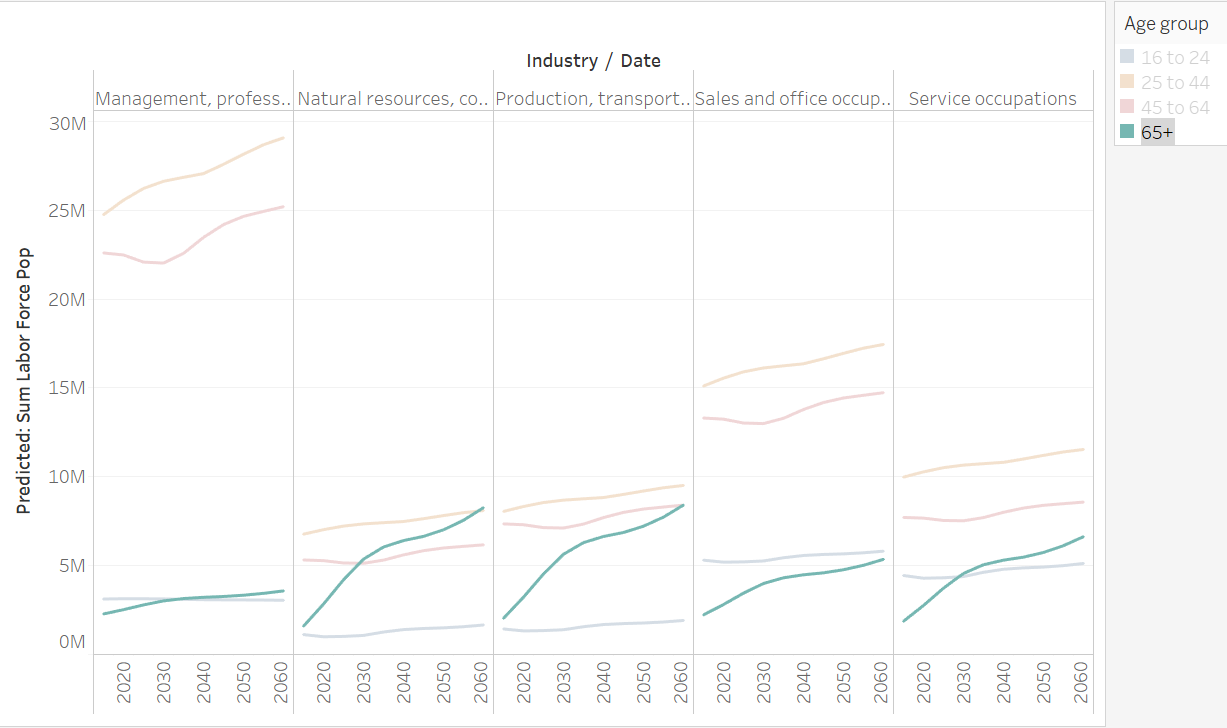
**Figure 2.A: Predicted Labor Force Population by Year and Age Group (65+)**



The theory that was drawn from this year over year increase in labor force population for the 65+ age group was that social security payouts were drying up and cost of living was increasing, meaning the retirement age increased. More workers in the 65+ age bucket were going to spend more time in the workforce. It is assumed later in our analysis that all over 65 are of retirement age but aging population, combined with higher living costs, and potentially lower earning careers will cause more to work later into their lives.

Digging deeper into the 65+ age group increase in labor force population we looked at the increase broken down into industry. Natural resources, construction, production, transportation, and service occupations experienced significant increases in labor force population. Again, this does not take into account the number of 65+ in our analysis that have reached retirement, but it is our assumption that lower earning careers such as production and transportation will see a higher number of 65+ pluses workers as baby boomers are forced to work later into their lives.

**Figure 2.B: Predicted Labor Force Population by Year and Industry**



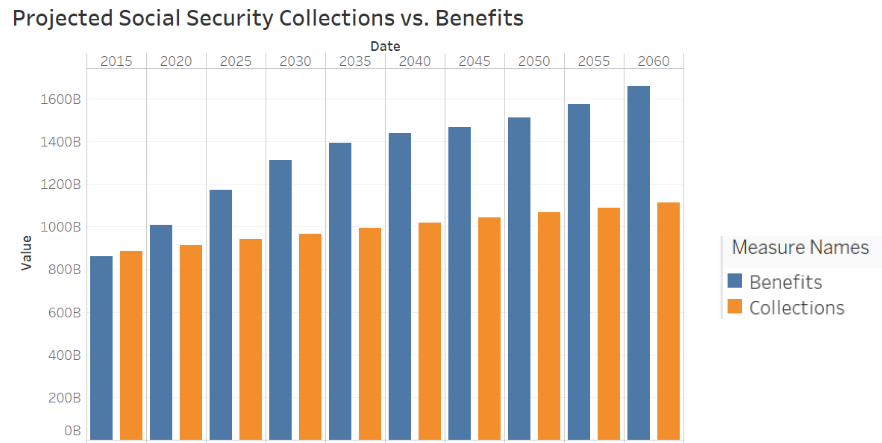
## TASK 3 – Taxes and Social Security

**Projected Taxes and Social Security Process**

After the conclusion of our predictive modeling our scored data was exported to a SAS file. The resulting data set matched the historic labor data used in observing historic social security findings but rather mapped out into the future from years 2015-2060 in five-year increments, and with a predicted “labor force population” resulting from our regression model. We then joined the projected data with the same social security data as in the historic example (employer rate, employee rate, wage cap). In this case, the values were identical for each year until 2060. Employee Rate = 6.2% Employer Rate = 6.2% and Wage Cap = $117,000. The resulting data represented identical columns to that in the historic example. We then calculated a column named “total contributors” by multiplying “percent employment” by “overall population”. The next step, as was before, was to create an if/else statement that would multiply “combined social security rate” by the corresponding “annual mean wage” IF the annual mean wage was less than the “wage limit” ELSE the calculation would multiply “combined social security rate” by the “wage limit” - resulting in one column with an accurate annual social security contribution named “total contributions”. Then with the use of proc SQL we aggregated a final table with two columns (year, total contributions) representing each future year with total collections for that year.

We then went back to the projected population table to isolate the number of individuals over the age of 65 in each future year – in order to compute the total benefits paid each year. Using a combination of Base SAS and proc SQL we were able to create a table with each future year and the complimenting total population over 65. Benefits for each then was computed in the following manner (population \* 1503 \*12) resulting in a table with each future year and the total benefits paid each year. We then took the projected benefits/collections tables and joined them proc SQL, the result is a table named **ss\_calc.projected\_ss\_final** with three columns representing (year, benefits, collections).

**Figure 3.A**



The resulting table **(ss\_calc.projected\_ss\_final)** was exported to Tableau and **Figure 3.A** was created. The resulting data follows the trend observed from 2000 – 2013. Benefits paid continue to accelerate at a rate far exceeding that of collections being collected from the current workforce. This is undoubtably the impact of the baby boomer segment reaching full retirement age, and a younger workforce unable to provide needed contributions to pay for benefits. If the trend observed by our model continues, benefits will have surpassed collections by nearly half a trillion dollars. It is important to note that our model is overfitted, and the trend observed from 2000 – 2013 has been almost perfectly fitted to the future data. Yet, nonetheless we know that social security will be unable to pay for benefits as we move into the future and in certain models predictions by a substantial sum.

## Scenario Task – Social Security crisis

As was observed in the above findings, social security will be unable to pay for expected benefits as population age segments continue to shift. As early as 2020, benefits paid will surpass collections received, and by 2060 this disparity could amount half a trillion dollars per year. Most notably, the population is aging and there are not enough young workers entering and progressing through the workforce to compensate for a growing number of retired workers. It is inevitable that changes will need to be made to the current social security policies in order to ensure future benefits are paid. Suggestions have been made to resolve this issue, including lowering the mean social security benefit, increasing the age to collect benefits, and or increasing the employee/employer payroll tax from its current rate.

**Using our findings, we computed the following scenarios/suggestions:**

Scenario 1.) Lowering the mean monthly benefit paid to achieve above or breakeven collections to benefits through 2060.

Scenario 2.) Increasing employee/employer payroll tax by increments of 1% (2% collectively) to find optimal rate resulting in above or breakeven collections to benefits through 2060.

Scenario 3.) Optimal combination of lowering mean monthly benefit and raising employee/employer payroll tax resulting in above or breakeven collections to benefits through 2060

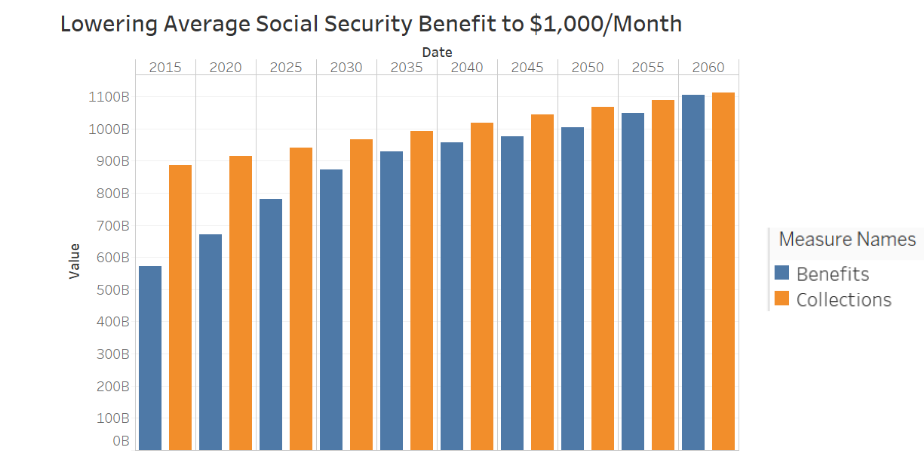
Scenario 4.) Raising collection age to 70 and above.

Scenario 5.) Raising the wage limit from its current level.

**Scenario 1.)**

In our first scenario we wanted to understand what mean monthly benefit payout would result in an above or breakeven collection to benefit ratio through 2060. We determined this by adjusting the total benefits paid for each year through 2060 by adjusting the monthly payout rate in our existing code. The resulting table **(ss\_calc.scenario\_one\_final)** was exported to Tableau and **Figure 4.A** was created.

**Figure 4.A**

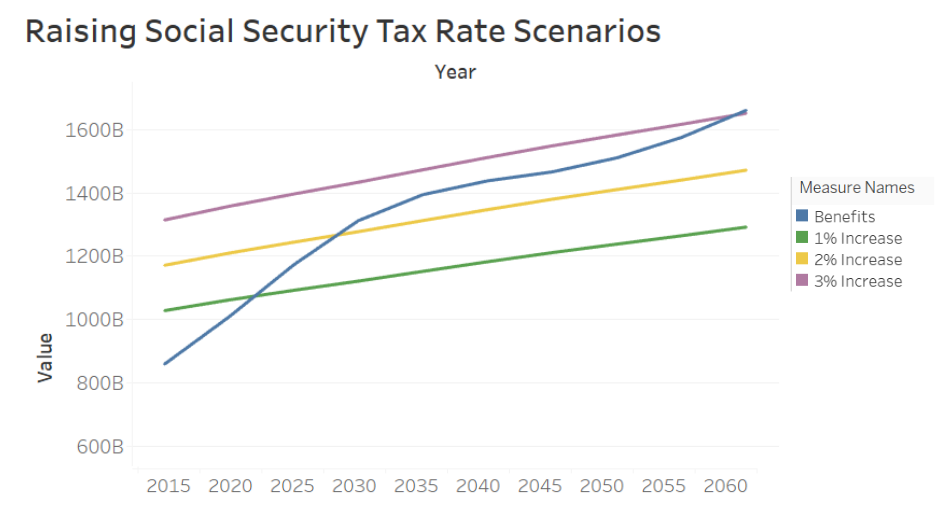


As seen **Figure 4.A,** the optimal number for mean monthly benefit is roughly $1,000. If the benefit is lowered to an average of $1,000 per month per recipient, based on our model collections will number more than benefits until 2060. According to our model and the further assumption, in 2060 collections and benefits will be roughly equal. Yet, is hard to imagine social security being able to reduce average benefits by 50% so further options must be considered.

**Scenario 2.)**

The next variable that could be changed to optimize collection to benefit ratios through 2060 is changing the employee/employer payroll tax to increase the level of collections in future years. Initial changes were made by adjusting each rate in our existing code by 1% increments (2% collectively) resulting in new collection predictions for each year until 2060. Each table was then aggregated and joined into one so that a layered visualization could be created. The resulting table **(ss\_calc.scenario\_comparison)** was exported to Tableau and **Figure 4.B** was created.

**Figure 4.B**

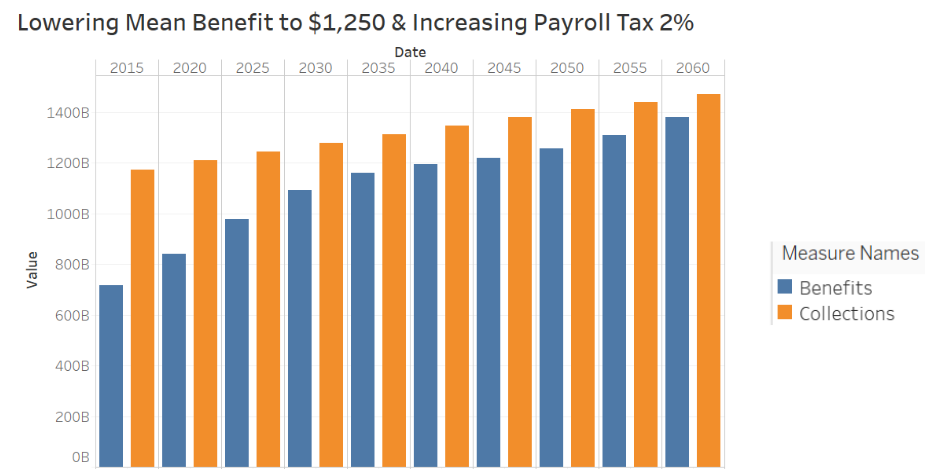


In **Figure 4.B** we see that raising the tax rate in 1% increments (2% collectively) eventually yields an attractive scenario by raising the tax 3% for each the employee and employer rate from the current level of 6.2% to 9.2% or a combined rate of 18.4%. Increasing the rate both 1-2% do not result in enough added collection revenue to provide at least benefit cost coverage through 2060. We believe that increasing the payroll tax is a more viable option than cutting benefits, as cutting benefits will have a direct impact on the elderly population's quality of living, and younger populations will benefit from paying into social security as they grow older. Yet, we thought it important to continue considering other scenarios.

**Scenario 3.)**

Our logic suggested that some optimal combination of the above adjustments could be made to achieve the same result. This was done by adjusting both the mean monthly benefit and the employee/employer rates in our existing code. The resulting table **(ss\_calc.scenario\_five\_final)** was exported to Tableau and **Figure 4.C** was created.

**Figure 4.C**

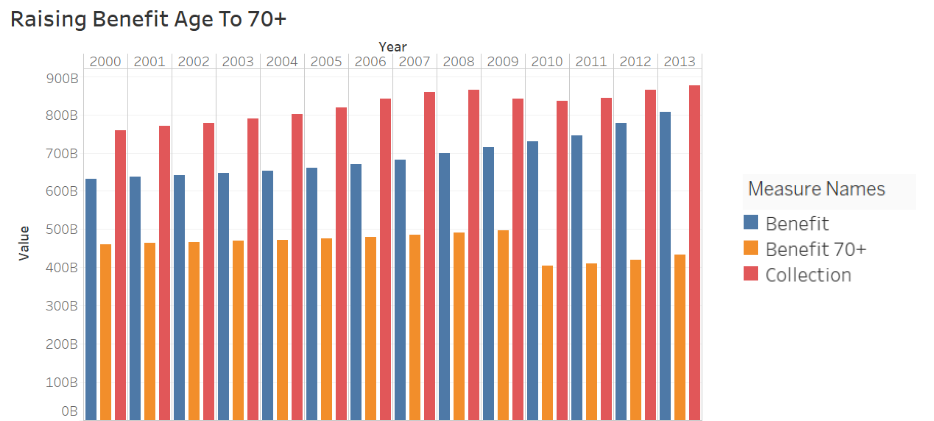


In **Figure 4.C** the impact can be seen of combining both lowering the mean monthly benefit and raising the employee/employer payroll tax in order to ensure collections total more or equal to benefits through 2060. We found a near optimal combination of lowering the benefit to $1250/month and raising the employee/employer rate by 2% (4% collectively). In this scenario collections number more than benefits through 2060. We believe this is a good option for lawmakers moving forward as it combines elements of the above two more extreme options. A 2% rise in payroll tax will not put too much strain on younger demographics will reducing benefits by 8.3%. Some may still argue that this is too much to lower the benefit by, and other optimal combinations can be made and baked into models of future collections/benefits.

**Scenario 4.)**

The next scenario that we considered is one in which the collection age is raised to 70 years old. We did not have the proper data breakdown to do build this into our future model, but we were able to complete a comparison using the years 2000 –2013. We used the census population table to isolate the population 70+ for years 2000 – 2013 and then multiplied them by annual benefit parameters (population \* 1503 \* 12). We then joined this with the historic collection/benefit table created earlier, this shows the impact of raising the benefit age from 65-70 over years observed years. The table **(ss\_calc.adjusted\_age\_payout\_final)** was then expoted to Tableau and **Figure 4.D** was created.

**Figure 4.D**

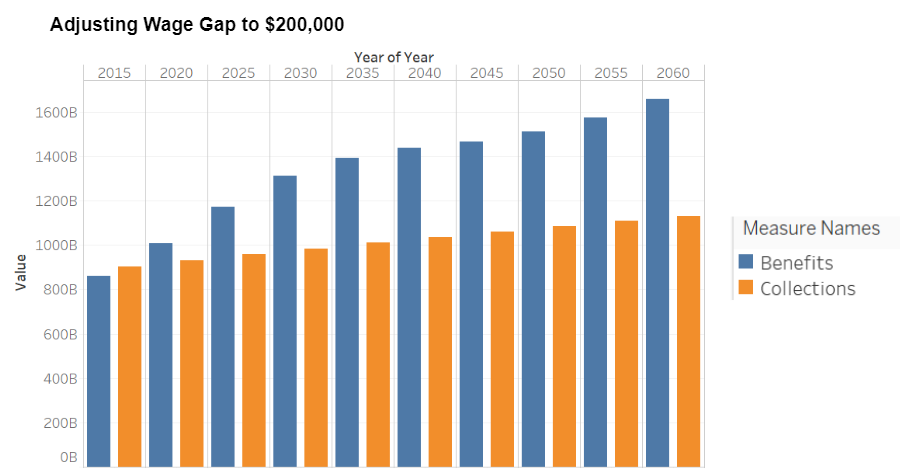


In **Figure 4.D** we see the substantial impact of raising the benefit age to 70 from 65. By raising the eligibility age to 70, based on the data provided, social security benefits are contained from 2000 - 2013. With such a drastic population shift expected from 2013 onward, it is likely that benefits would still surpass collections; although, not as soon as 2020 as seen in our earlier models. As time passes, the impact of raising the benefit age will lesson and eventually further action will need to be made to adjust the payroll tax or monthly benefit.

**Scenario 5.)**

The final scenario we considered was raising the wage cap for social security collections. This was done by using our projected benefits paid, already formulated earlier, and joining it with a new column representing the new collection level if the wage cap is raised. The final table **(ss\_calc.wage\_cap\_adjusted\_final)** was exported to Tableau and **Figure 4.E** was created.

**Figure 4.E**



In creating **Figure 4.E** we experimented with raising the cap as far as we could before there was no more positive impact on collections. We found this level to be $200,000 after this point there is no observable impact of raising the wage cap. Over the next forty years this could lessen the impact of rising benefits but would still need to be combined with one of our earlier scenarios to ensure that benefits are entirely covered by collections.

## Conclusions

The process of bucketing, recoding and cleaning were carried out on the historic census data as well as the future population data. These buckets were created based on the Age variables in the disease and labor tables. Separate bucketing methods were carried out between the labor and disease data sets. Once the respective processes were completed, the scoring and input files were ready for further analysis and were exported to be used in SAS Miner.

By examining the disease modeling we projected that there will be an increase in disease cases as a whole moving forward, due to an aging population. Within the next 45 years we are going to see a shift in the current age demographics as 65+ people will outnumber 0 to 17 year old's. This is leading the charge for the increase in disease cases, as typically we see older individuals need more medical care than younger people. It is no surprise that we will see increase in dementia, Alzheimer's, cancer and septicemia as these a typically associated with older individuals. Moving forward we expect to see this numbers decrease, as technology continues to advance, and treatments and prevention become more cost efficient.

Focusing just on labor force population we were able to back up the statement that retirement for 65+ workers were going to be retiring later and later every five years. This trend was displayed for the whole age demographic but had more of a significant increase in lower paying industries, such as transportation and construction. This observation of the data helped support the findings in the prediction from the social security data.

Based on our modeled labor force predictions social security is going to be a major issue facing our country in the coming decades. In the historic period observed collections remained stagnant while benefits paid steadily grew as more workers retired. We found that if nothing is done to change the current social security policy there could be a deficit in collections to benefits as soon as 2020. If allowed to continue through 2060 our model predicts that this deficit could reach half a trillion dollars per year. Our group attempted to find scenarios in which current social security policy could be adjusted to ensure collections number more than benefits paid through 2060. We computed five scenarios in which social security variables were adjusted to reach this end. We found multiple methods of lessoning the impact of rising benefit payments, including lowering the mean monthly benefit, raising employee/employer payroll taxes, raising the wage cap ceiling, or raising the benefit age all together. We found multiple viable avenues to containing the growth of benefit payouts, yet no one method seems to be a perfect answer. It is our conclusion that some combination of the above scenarios will have to be made in order to ensure retired Americans receive their fair share while younger demographics are not left holding the bill.