Population Prediction Machine

Jeremy Waibel, Michael Amberg, Ian Webster, London Kasper

CS 5394/7394

https://github.com/londonkasper/mlP1.git

Summary

For this project, we were tasked with predicting the population of the Earth in 2122. In order to predict this, we used pre-existing data rates of population growth for the last 70 years. The three major datasets we use are; total population, deaths per year, and births per year, all ranging from 1950 to 2021. After experimenting with a variety of different ways to solve this problem, we eventually settled upon using a method called autoregression, a model primarily used for predicting stock prices. Using this model, we came to a conclusion of 10.5 billion people in 2122.

Research

In order to find a way to approach this problem, we first researched how modern population predictions are made. One way to predict the future is to use regression, most commonly linear, to build a “best fit” line to describe the data. We would assume that this “best fit” line would continue the trend for future years. It is important to distinguish between interpolation and extrapolation. Whereas interpolation relies on previously existing data to come to a conclusion, extrapolation is the process of creating new data from patterns found in old. We had to make sure our models were able to extrapolate data, not interpolate.

Data

Our data was gathered from Macrotrends.com, having many different metrics that affect population dating back to 1960. We chose to download and use the .csv files on world population, birth rate per 1000 people, and death rate per 1000 people. None of the data included missing values or outliers, so no cleanup was needed in order to use the information.

Prediction Process

We decided that while the population is affected by numerous factors (such as food production rates, natural disasters, healthcare expansions), there are only two factors that directly impact the population. These factors, the birth and death rate of each year, are the only direct causes of changes in population and therefore should form the basis of our initial model.

Initially, we formed a regression based on the net change (birth rate minus death rate) of the world’s population alone. Our result was much larger than we expected— we predicted 15.8 billion people, more than twice the current world’s population. Since this seemed unreasonable, we decided to

take a different approach and create separate predictions of both a birth and death rate for each year until 2122.

In addition to our previous approach, we attempted a few other methods to see if we could achieve a more reasonable prediction. One of which was a brute force method of simply creating a trendline based on the rate of change of the population throughout the years. With this we came up with the trendline: y = -0.0142x + 2.1332, with x and y being associated with the year and predicted rate of change respectively. For each year between 2022 and 2122, the program used the trendline above to predict the rate of change for that year then apply it to the present-day population. Through this method, the program calculated roughly 11.9 billion people in 2122. While being very satisfyingly close to actual scientific predictions, a critical hole lies in the middle of this solution. The R^2 value of a trendline describes the accuracy of the line, in terms of 0-1 (1 being the most accurate). The trendline we used only has an R^2 value of 0.96, meaning for the data above it's fairly accurate but only for short term prediction. Contributing to this, the equation is linear and has a negative slope, meaning the further away in time we attempt to use it to make a prediction of the population rate of change, the closer the R^2 value will move to 0 and the more inaccurate it will become. Therefore, although this method did get a satisfying and semi-reasonable answer, we determined that for our purposes this method is unreliable.

The other small approach we looked into was using sklearns linear regression calculator through a model shaped by the pandas database analysis. The method simply just shapes our total population data with its corresponding years into a new dataset, which then fits a linear regression model on top of it. This method gave us an answer of roughly 13.6 billion. This result was greater than the previous model and thus more inaccurate, but it theoretically should be more accurate. Unfortunately, this is also shown to be false through means of our projected data. Unfortunately there is no R^2 for this regression, so to test its accuracy we cross-referenced its predicted data with the projected data we had previously neglected to feed the program since we wanted it only to have access to factual, accurate data. After this cross-reference we saw the model consistently predicting either too high or too low. Due to this wide range of error, we were unable to consider this solution as reliable enough to provide accurate data.

Final Model

We finally settled on using an autoregressive model. Autoregression is a time series based model that uses the previous value(s) of Y to predict future values of Y, which when applied to our datasets led to some pretty promising results. Additionally, since an autoregressive model relies on previous data, events (such as war, famine and disease) that affected population rates in the past are automatically incorporated into the prediction through the data itself. Autoregression assumes that the future will resemble the trends of the past, and is often used for analyzing natural processes of the Earth and in stock movement prediction.

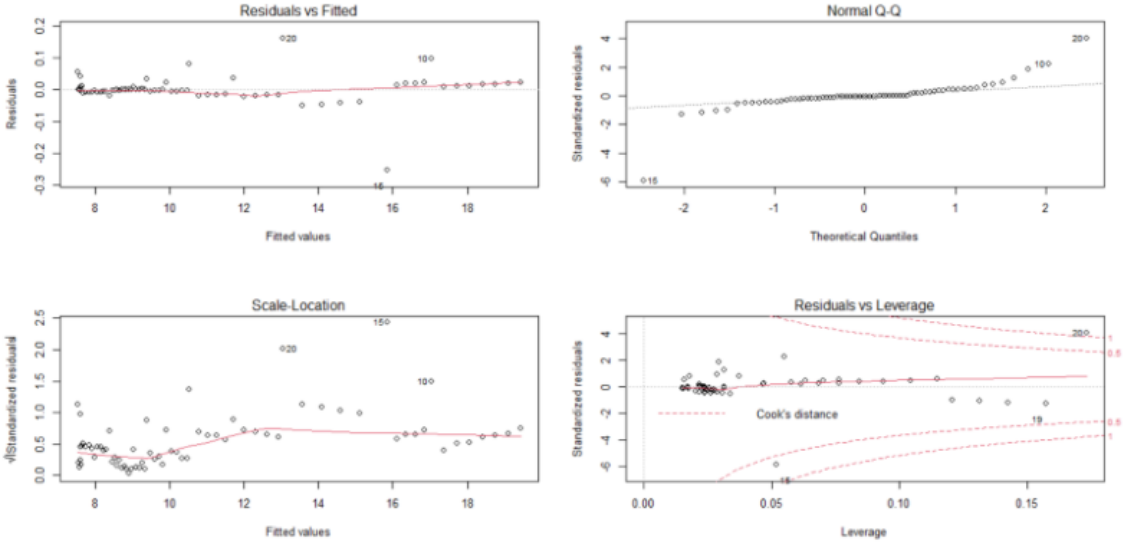
We applied autoregression, AR(p), to model both deaths_per_1000_people and births_per_1000_people. In order to determine the order of the autoregressive model, we calculated the BIC (Bayesian Information Criterion) for an order of 1 through 6. The lowest BIC would indicate

which order to use. Coincidently both models were best fit with an autoregression to the second order, AR(2).

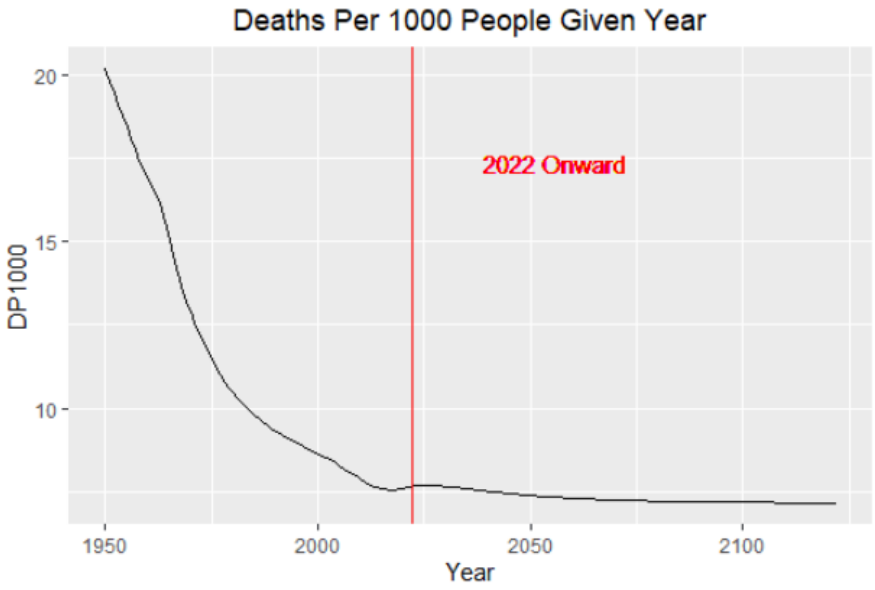
Modeling Deaths Per 1000 People (dp1000):

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          0.04955    0.02060   2.406   0.0189 *
L(ts(deathparam))    1.80361    0.06555  27.513   <2e-16 ***
L(ts(deathparam), 2) -0.81052    0.06349 -12.765   <2e-16 ***
```

$$dp1000_{year} = 0.04955 + (1.80361 * dp1000_{year-1}) - (0.81052 * dp1000_{year-2})$$



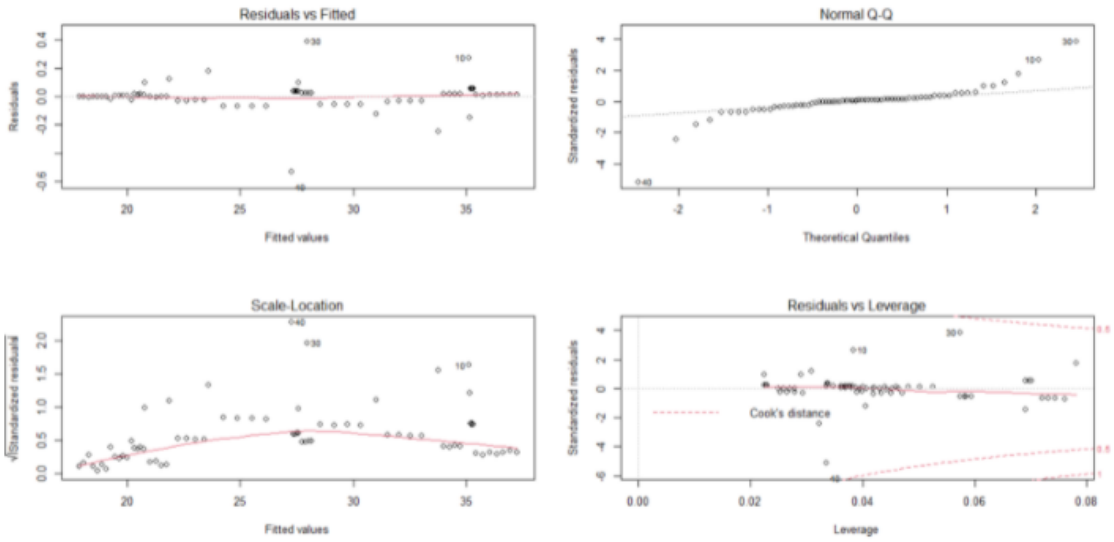
Our normal Q-Q plot shows some derivation from normality, however since n=72 the central limit theorem holds and we don't have to worry about this slight error. The other diagnostic plots show that there are a few points of concern (10,15,20). Despite these concerns, we decided to work with this model since we do not know any work-arounds and there are only a few points that have a higher weight than others. We did not want to remove these points as they are part of a time-series and removing them may have negative implications on the outcome of our results.



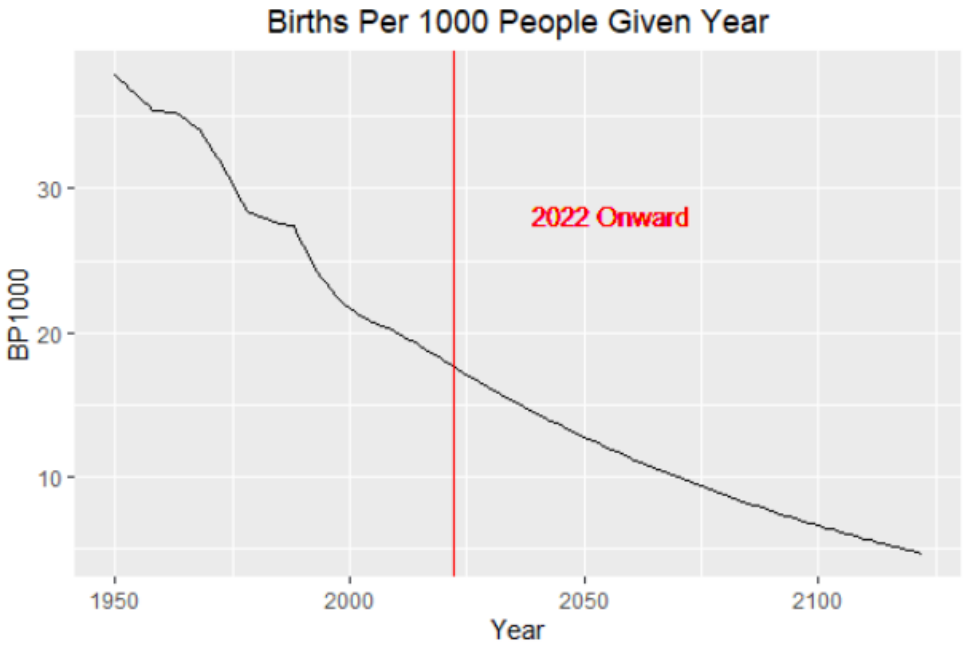
Modeling Births Per 1000 People (bp1000):

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.006267   0.057702  -0.109    0.914
L(ts(birthparam))   1.815524   0.069958  25.952   <2e-16 ***
L(ts(birthparam), 2) -0.817133   0.069706 -11.723   <2e-16 ***
```

$$bp1000_{year} = -0.006267 + (1.815524 * bp1000_{year-1}) - (0.817133 * bp1000_{year-2})$$



Similarly to the diagnostic plot of the autoregression on death, there are some points that have higher leverage than others. The CLT holds again as the sample size is still n=72.

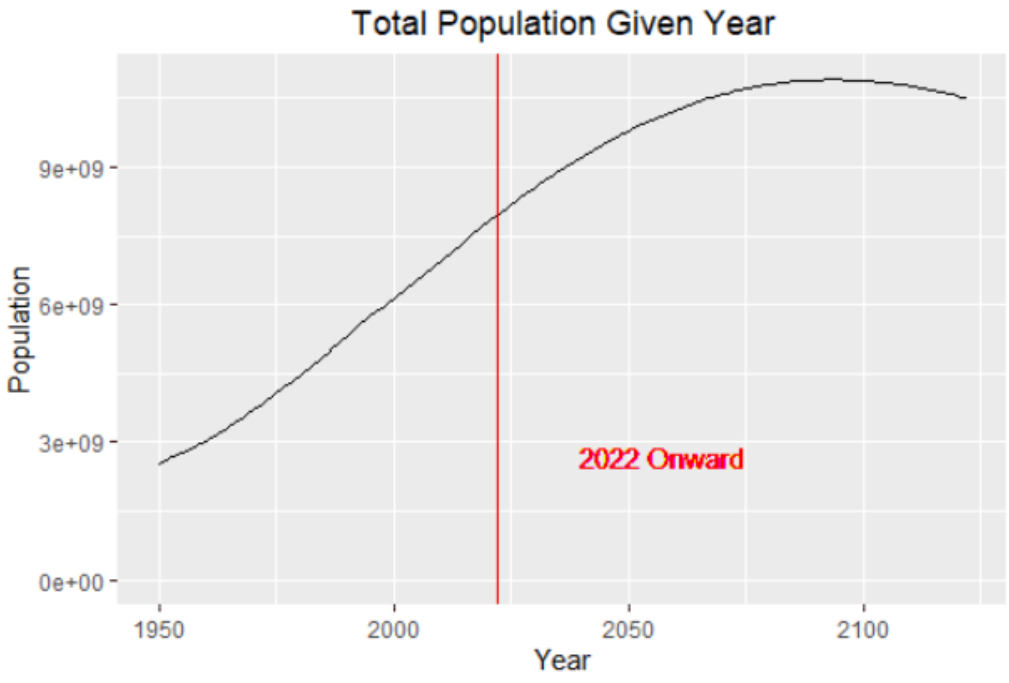


Population Prediction

After extrapolating bp1000 and dp1000 until 2122, we had to convert the numbers back into total population.

$$Change_{year} = (bp1000_{year} - dp1000_{year}) * Population_{year-1}/1000$$

$$Population_{year} = Change_{year} + Population_{year-1}$$



Hindsight is 2122

In reflecting, we realized we had a difficult time getting started with this project due to our lack of understanding of the prompt. Together we brainstormed countless factors that could affect the human population at any point in the next century. As you might imagine, the list we came up with was far out of the scope of our knowledge, deadline, and distracted us from the far simpler problem at hand. It took us a while to realize that factors such as war and advancements in technology are already reflected in our data, and that it was out of the scope of our project to include predictions for major events. Essentially, we started too large-scale, and eventually had to limit our ambitions for the sake of time.

Since we were unable to include factors such as famine, disease, and population density, we felt that our prediction was missing worldly context. Given more time we would incorporate more factors into our prediction. For example, we would want to check for correlations between food production and fertility, or see if there are correlations between the number of births per mother and the GDP of the country.

Additionally, between the members of our group we had very different levels of experience with statistical models, ranging from those with no prior statistics experience to graduate-level statisticians. Communication between these groups was very difficult due to this learning curve, and led to a bit of time-consuming frustration. In order to avoid this sort of conflict in the future, we will seek to allocate tasks more efficiently at the beginning of the project.

Similarly, we attempted to approach this problem solely using Python. While most (if not all) of the functions we ended up using are available in Python, we found R to be more effective in terms of preliminary data inspection. As this was the first time for any of using Anaconda Navigator, Python and (for some) Jupyter Lab, there was a bit of a learning curve involved in simply finishing the project to fit the rubric.

"All good things must come to an end."

As for our final model, we were satisfied with our final answer. 10.5 billion people is a reasonable prediction for the population. However, since autoregression is based on previous data, it doesn't take into account the possibility of technological advancement in the future. The model only acts on known concrete values and implicitly assumes the future will resemble past trends. As we know, a lot can change in a century. If we could restart this project, we would consider other autoregressive models that incorporate additional variables such as ARIMA to incorporate different parameters into our prediction.

In conclusion, each member of the group learned quite a bit throughout the course of working on the project. Some were introduced to statistical methods for the first time, and all were introduced to the powerful language that is Python. As a group, we grew closer as friends and coworkers, and everybody learned a little more about time management and how to utilize their own strengths within a team environment.