# MSc in Data Analytics – CA1

An Investigation into Irish Emigration trends in the 21st Century

Author: Stephen Burke

e-mail: sba23027[@student.cct.ie](about:blank)

Student ID: 23027

Github: https://github.com/step-hen-burke/ca1

Accompanying notebook:

https://github.com/step-hen-burke/ca1/blob/master/population\_and\_emigration\_analysis.ipynb

## Abstract

*An exploration of data regarding the population of Ireland over time was carried out. In order to account for population shifts, further analysis of migration trends was performed. It was shown that these can be modelled using various statistical distributions predicted using Machine Learning when framed as a supervised regression problem.*

## **Introduction**

This analysis was carried out using python, primarily in an Ipython notebook using Jupyter. This format allows for mixing code, markdown, and visualizations, in order to effectively convey the reasoning and insight generated throughout a data project alongside the output of code itself.

The code itself is primarily imperative; as interactive data analysis primarily uses the REPL (or notebook output), it makes sense to step through small chunks of code in order to inspect their outputs rather than storing state in user-defined objects as would be the case under an OO paradigm. That is not to say that Objects are not used extensively throughout the analysis, as there is extensive use of the pandas DataFrame (McKinney, 2010), and several classes exposed by the scikit-learn, scipy, etc., packages. Some functions are defined for structure and to avoid code re-use, however the project could not be considered functional as these are typically not pure.

The principals outlined in Clean Code (Martin, 2012) were followed; names are descriptive, repetition is minimised within reason, and a code style guide is adhered to (PEP-8).

The seaborn library was used to produce the graphics in this report and accompanying notebook. Its style was set to “darkgrid”, which is similar to default themes used in other graphics libraries such as R’s ggplot2, and conforms to style directives outlined by Tufte (p. 112, 116); the grid background is a muted grey so as to not be distracting, and gridlines use minimal ink. Indeed, the gridlines are given as negative space, so do not use ink at all. Tufte also argues in favour of using a font family with serifs in order to create more “friendly” graphics.

Additionally, the default linewidth was set to slightly thinner than seaborn’s default. Tufte argues that thinner lines result in more aesthetically pleasing graphics (p. 184-185).

## **Population Data Exploration**

The first dataset explored concerned population estimates in April from 1950 to 2023. The data was loaded and preliminary exploration was carried out (Section 1 of the accompanying notebook). The data was summarised and missing values were identified - these were not missing at random and attempts to infer their values were carried out (Section 1.1). However, the candidate imputation values did not equal the reported values in 47/108 cases.

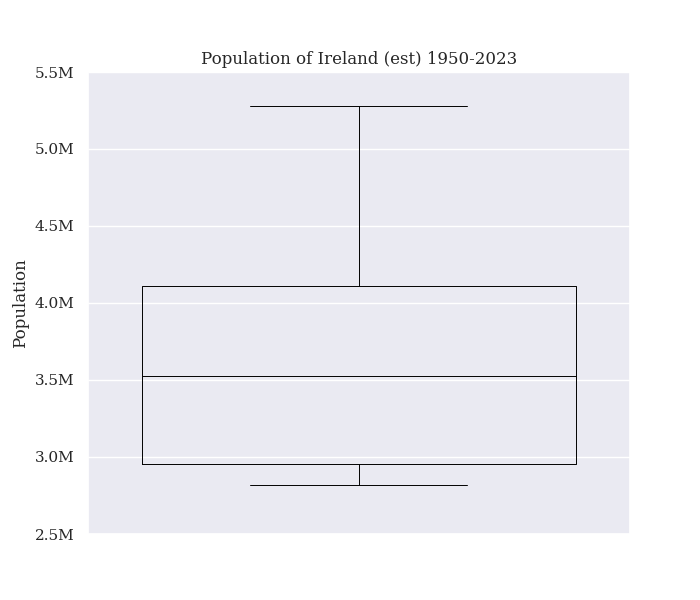
Further investigation revealed that in certain cases, the discrepancies between the additions of age groups and their supposed combined groups were abnormally large. Census data was imported for comparison, and it was determined that the 1-4 year old data for Females was reported erroneously in the population estimates table. The CSO was contacted about this, and it was confirmed that the data for 15-24 year old Females had overwritten the data for 1-4 year olds when it was originally ingested.

These values were also treated as missing and were imputed. As the Female figures could be thought of as a univariate time series, methods such as those outlined by Moritz et. al. in 2015 were considered. However, due the structure of this particular dataset, the missing values could be calculated directly. Full details can be found in sections 1.1-1.3 of the notebook.

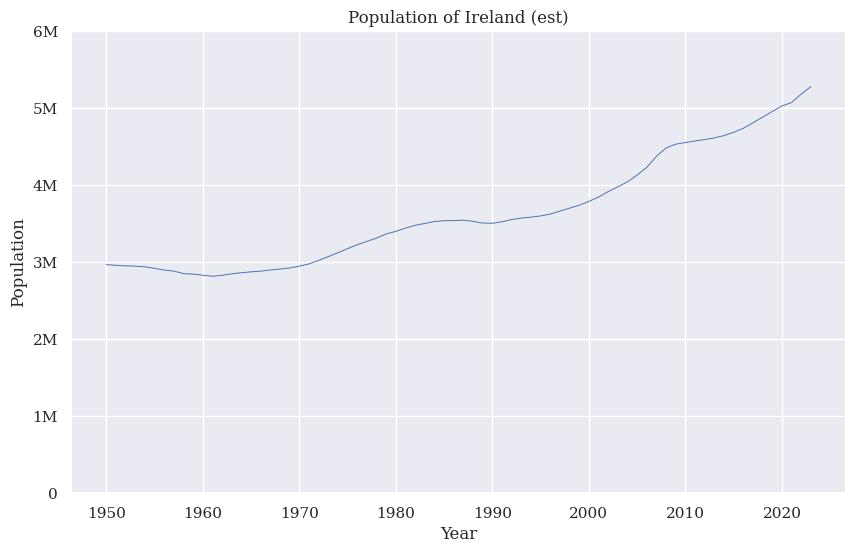
Wickham (2011) argues that when data is collected at multiple levels, the data should be split into tables containing individual observational units. This was done, resulting in two clean datasets – one split by sex, and one containing the data summed across both sexes. The data was then summarised visually – focusing initially on the combined case in order to give an idea of high level trends in population over time.

It was seen that, across all 74 years within the data, the mean population count was about 3.6M, with a standard deviation of 718K. We can also see that the data is right-skewed, as the mean is greater than the median and the upper tail of the variable’s boxplot is longer.

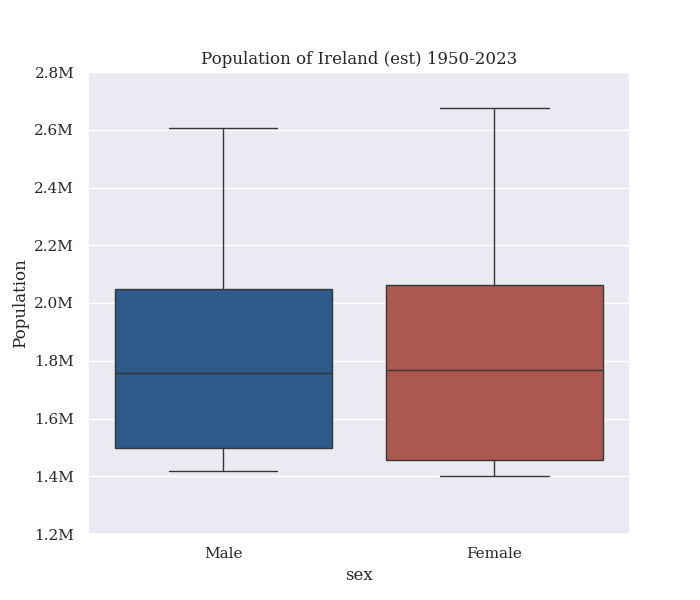
|  |  |  |
| --- | --- | --- |
|  | year | value |
| count | 74 | 74 |
| mean | 1986.5 | 3641.32 |
| std | 21.51 | 718.94 |
| min | 1950 | 2818.3 |
| 25% | 1968.25 | 2954.83 |
| 50% | 1986.5 | 3527.35 |
| 75% | 2004.75 | 4111.65 |
| max | 2023 | 5281.6 |



The year and population variables are both continuous, so a line chart is a natural choice of visualisation to highlight the time-dependencies between each successive data point. We see that the overall population trend is increasing, however periods of decline/stability are seen in the 50s - early 70s, the late 80s - mid-90s, and the late 2000s - mid 2010s. The Y-axis was formatted with M suffixes denoting millions to aid readability.



The same visualisations split by sex were produced. A colour palette that has high contrast was chosen. It is also colourblind friendly as it was retrieved from David Nichol’s “Coloring for Colorblindness” tool. Tufte advocates for colourblind-conscious choices of palette in aid of creating more “friendly” graphics (p. 183).

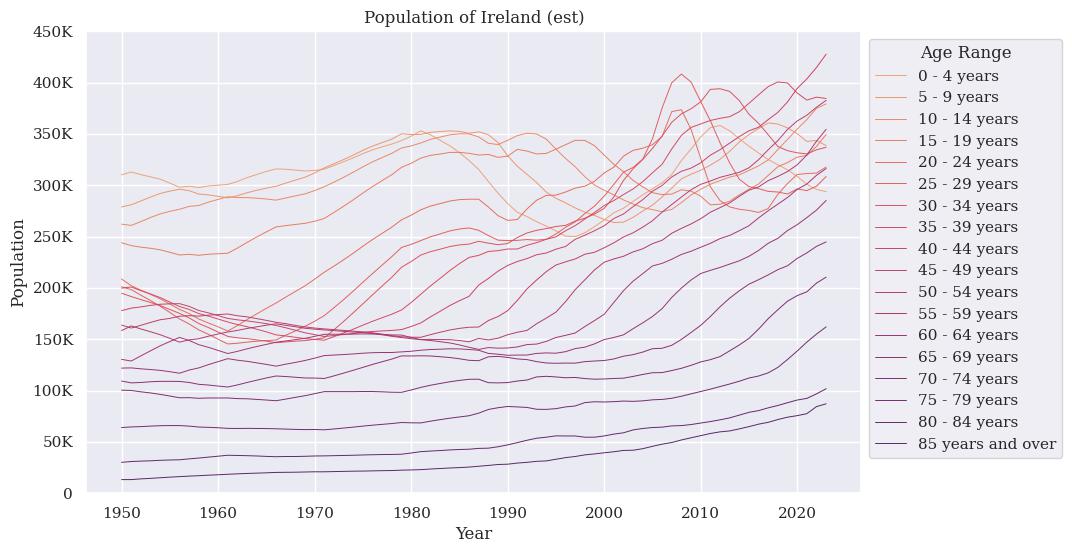
It can be seen that the mean population when split by gender is roughly equal, however the distribution of the female population is slightly more spread and right-skewed than the male equivalent.



It be seen that in the 1950s there were slightly more males than females, and since the late 2000s there have been more females than males.

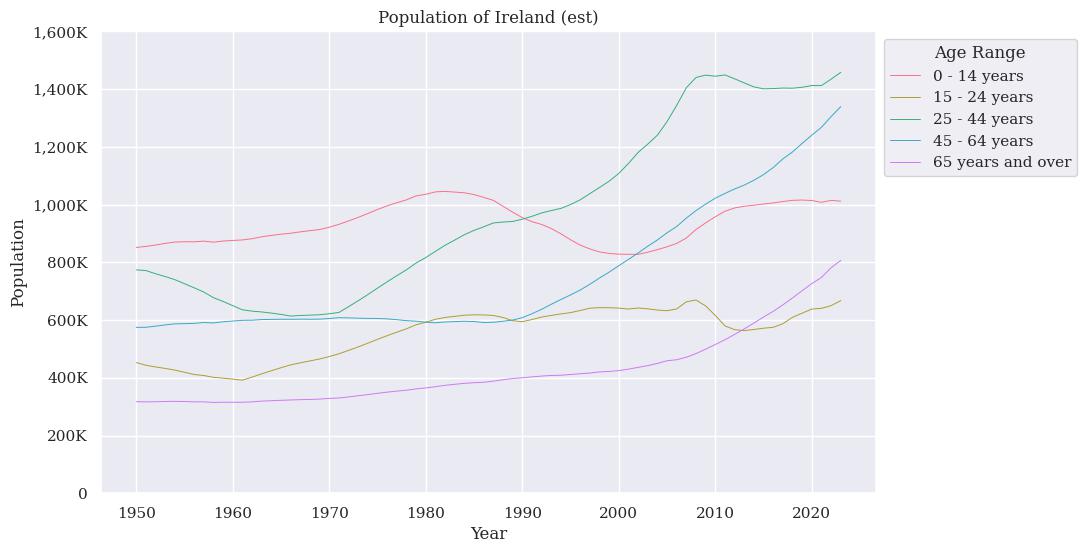
Like the sex variable, there are overlapping age bounds within the age\_group variable. A covering of ages comprised of non-overlapping ranges was selected to avoid double counting. Multiple choices exist, and the covering giving the highest level of granularity and having each of the age ranges (except 85+) be equal width was selected (Section 1.4.3 of the notebook).

Population per age group can be visualised as a line chart as before, splitting on age range. However, the colour palette previously used does not have enough levels to represent this many age groups. The age levels are ordered, so it makes sense to use a sequential colour palette as opposed to a categorical one.



The resulting graphic is very busy, or something that Tufte might call “a puzzle”, and is in general diﬀicult to follow – therefore a coarser age covering was chosen in order to simplify this visualisation. One observation that can be noted from this chart however, is that the youngest age categories transition from being the most populous in the 1950s, to being overtaken by working ages in the 2000s. This is indicative of a shift in demographic makeup.

An alternate covering of the span of ages can be used to make the above visualization less cluttered. With this visualisation, we can clearly see the number of 0-14 year olds declining in the 90s until the mid 2000s, and the number of 25-44 year olds increases dramatically over this timeframe.

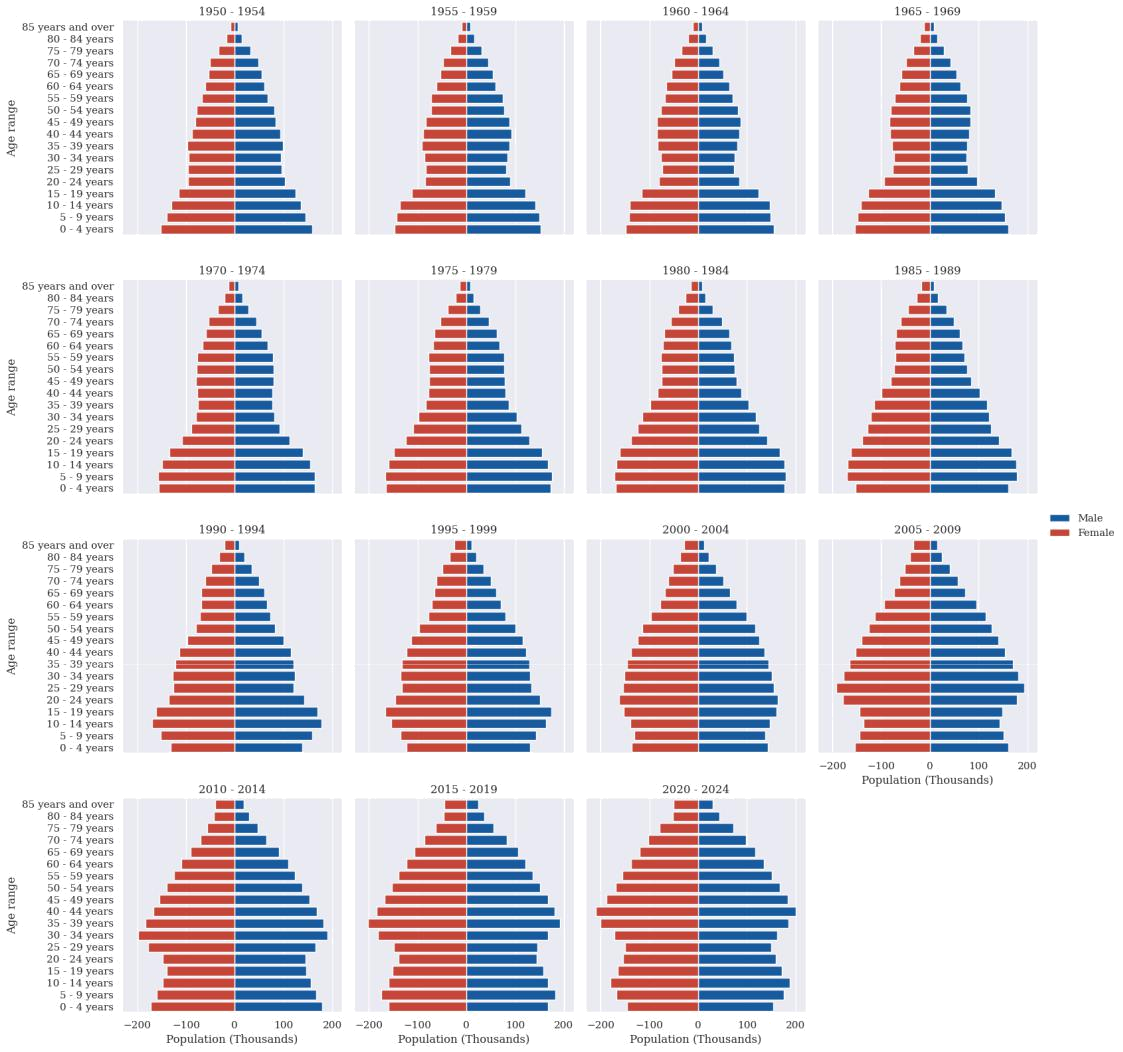


As we are only working with 5 levels in this chart, we switch the palette to one with more variation so that individual trends can be followed more easily, rather than having the hue change as before.

One thing to note here though, is that the age ranges should not be compared to one another, as the number of ages in each bucket is not equal.

We would also like to look at population age composition separated by year, so we group the years together into a range. Grouping by decade would be the obvious choice, however we can see from the above chart that some of the rapid changes in demographic occur over a shorter period than that, so buckets of 5 years were chosen.

We can also reintroduce gender at this point without creating too much visual clutter.



We visualise the changing population across genders and ages using multiple population pyramids (age structure diagrams), which are paired bar plots typically used for this purpose (Wilson 2016). Tufte argues (p. 170-175) in favour of “small multiple” visualizations in order to observe a change over time.

This shows that the population in Ireland has shifted from having a relatively wide base - many births and young people - to having a wide middle and comparatively narrower base, indicative of a lower birth rate. A notable change in the width of the middle ages can be seen in 2000-2010 - expected given the above line charts.

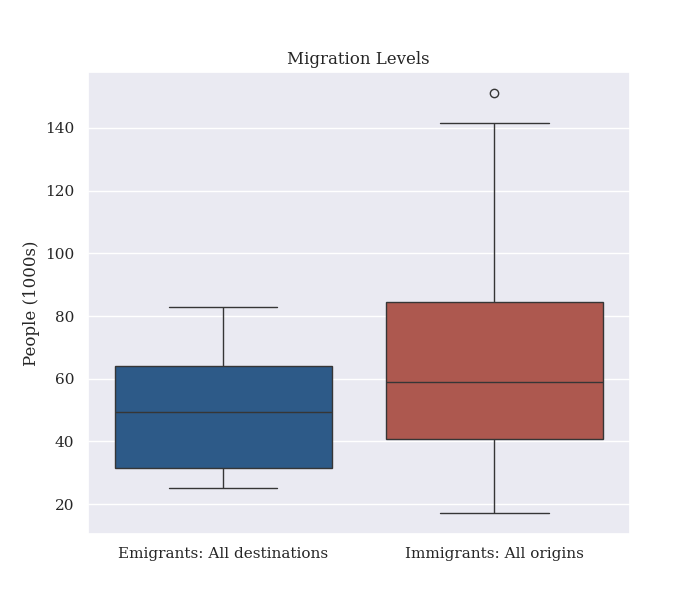
This can explain the higher percentage of females in recent years we saw previously - As women have a longer life expectancy than men (Gryclewska, 2016), an aging population would naturally be comprised of more women - the gender difference in the older cohorts can be seen clearly in the plots from 2000 onwards.

## Migration Data Exploration

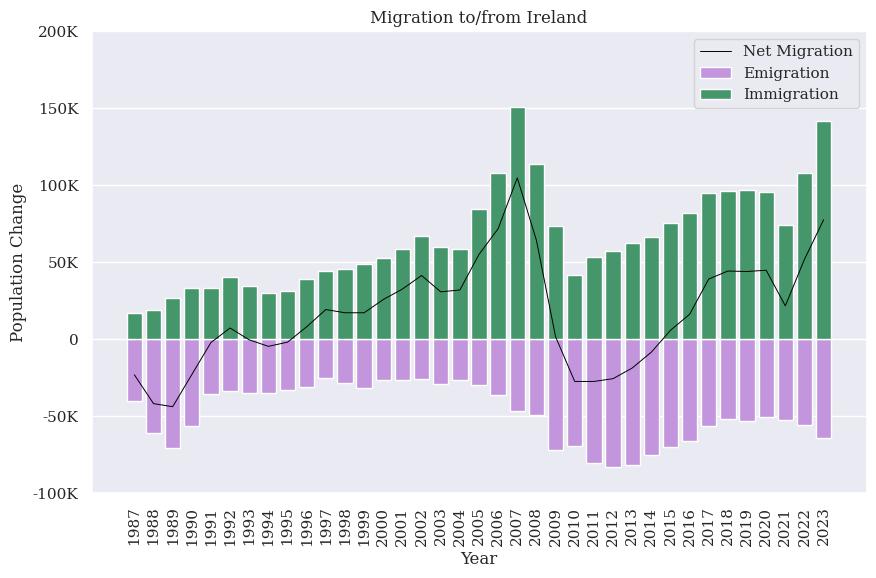
Bringing in migration data will help us gain some understanding of what is driving some of the population changes seen above. On loading the estimated migration data we see that again, we have missing values in the value column. These primarily impact the net migration and All ages datapoints, so can be repaired using combinations of other values as was done for the population data above. The net migration values are not missing in more recent data, so it’s possible that this was not originally calculated when records began.

We first pivot with age group across the columns, repair All ages by adding the figures for the other ages together, melt and repivot with inward\_or\_outward\_flow across the columns, then finally repair Net Migration. After this procedure we have once again obtained a cleaned dataset – this is demonsrated in section 2.1 of the accompanying notebook.

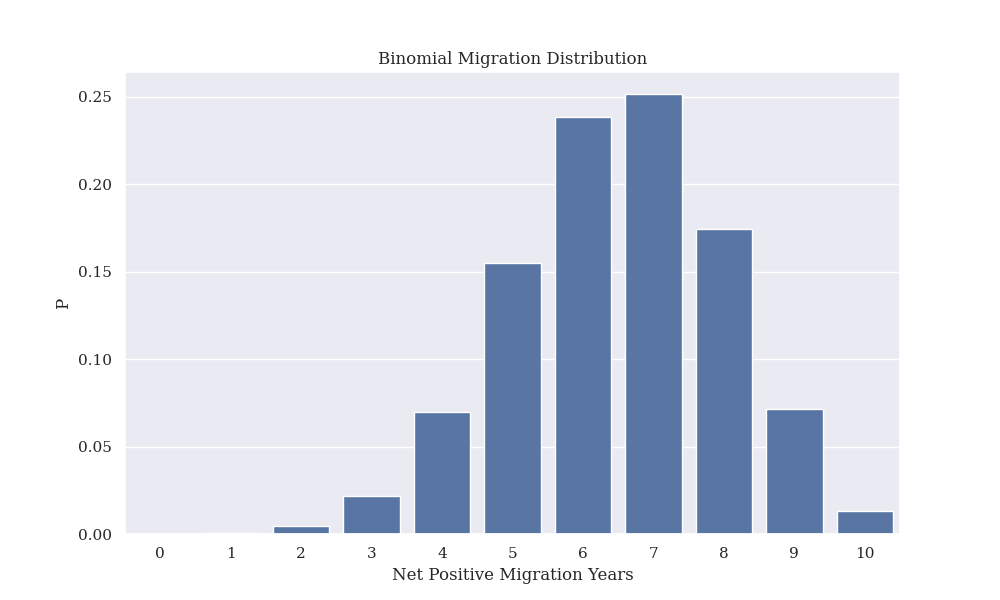
We can observe the distribution of Emigration and Immigration levels across the years using another pair of boxplots – this shows that the mean immigration level is higher than the mean emigration level, and the distribution of immigration levels is right-skewed, with one outlier point.



We can display emigration and immigration as bars, with an accompanying line denoting net migration. Although emigration results in a reduction in the population and immigration results in an increase in the population, I have elected not to use typical increasing / decreasing colour choices (blue for increasing, red for decreasing, etc.) as these may also be interpreted as ascribing goodness or badness to either phenomenon, which I want to avoid, while still choosing colours that are visually distinct.



We can model the relationship between emigration and immigration using a binomial distribution, as is done in Section 2.3 of the accompanying notebook; where each year is an independent Bernoulli trial with success being defined as immigration > emigration, or that net migration contributes to population growth. If we denote 1 as a success and 0 as a failure, we can estimate the success probability over all years by taking the mean of our success column.

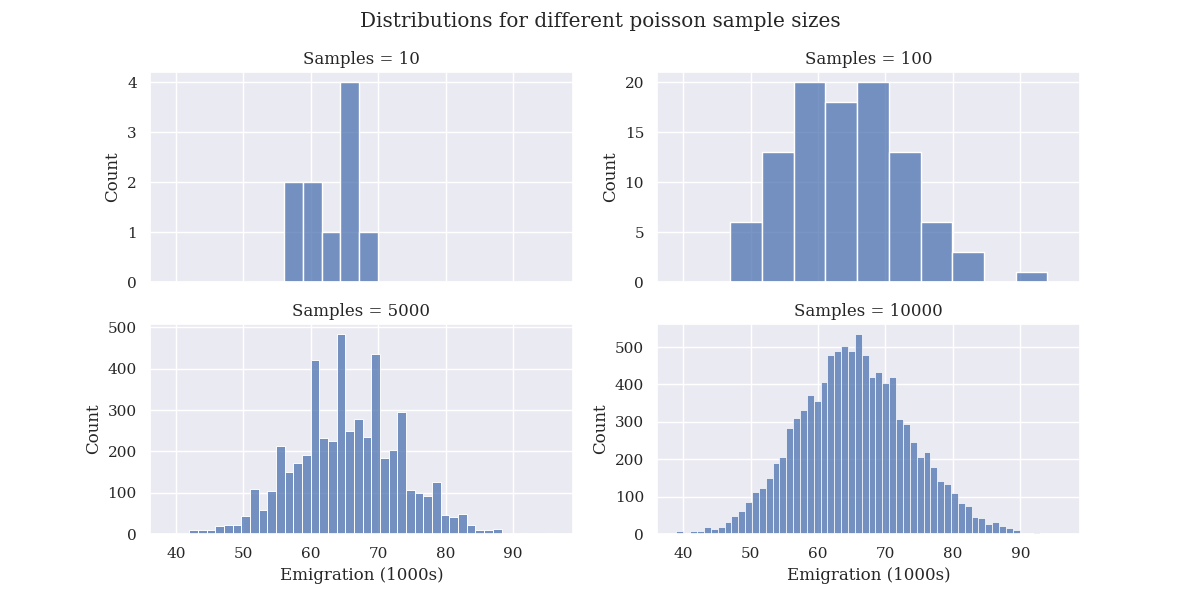
We can then draw up the binomial distribution for the number of successes in a decade using the binom.pmf() function by setting our number of trials to 10. As our estimated P(X=x) is 0.648 (> 0.5) our distribution is left-skewed (Weiss, 2017).

Looking at the decades before and after the millenium, we see that the 90s had 5 successes, which according to our model had a probability of 0.155 of occurring, wheras the 2000s had 10 successes, which had only a probability of 0.013 of occurring. This indicates that the distribution of successes may have shifted over time, as the probability of the post 2000s data occurring according to our model is low.

We can also model migration using a poisson distribution (section 2.4 of the accompanying notebook) by treating one immigration event as a success. We can calculate lambda as the mean number of immigrants per year.

Using this, we can estimate the probability that immigration is greater than the mean emigration level, which over the whole dataset has a probability of 0.487

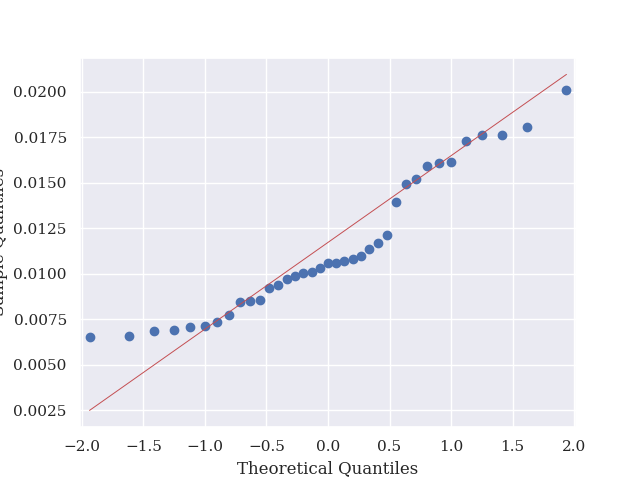
Additionally, As the number of “successes” are discrete variables in both of the above cases it is appropriate to model them with discrete distributions rather than continuous ones. However, as the number of samples grows large, the use of a continuous distribution becomes justified. By drawing successively larger samples from the above poisson distribution we can observe it converging to a normal distribution in the limit.



### Comparing pre & post 2008 Emigration

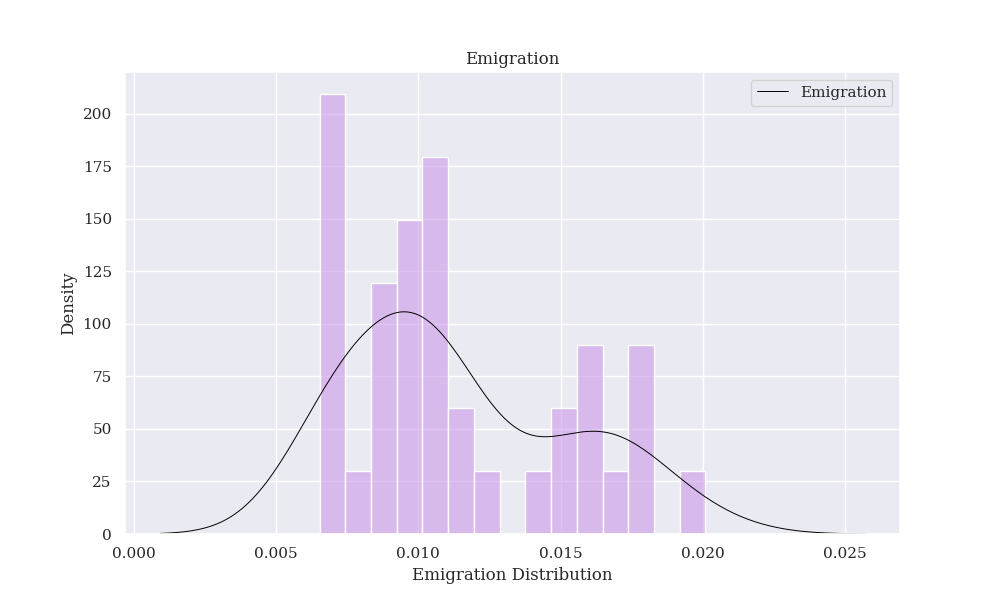
Looking at the line and bar graph there was an inflection point in 2008 and a period of net emigration afterward. Before and after this our emigration figures can be thought of as coming from different distributions. The comparison between these two time periods can be found in sections 2.5 and 2.6 of the notebook.

We can use a qqplot to check the emigration variable’s distribution against a theoretical normal distribution. We expect emigration to increase as the overall population increases, so in this case we can look at emigration as a percentage of total population to control for this.

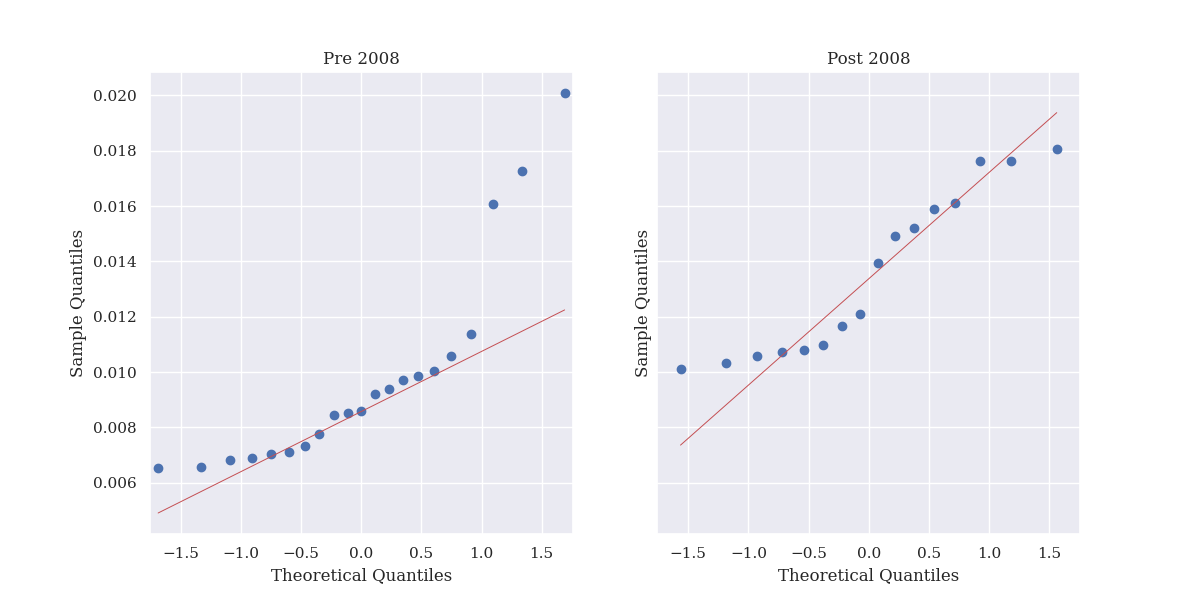


This is a poor fit - implying a nonnormal distribution for the overall series.

The inflection point at 0.5 indicates bimodality - something we can see with a histogram.



So we look at separate qq plots pre and post 2008 to see if these partitoned series are distributed normally.



Neither of these are convincing fits, so this partition does not cleanly split our data into two normally distributed subsets.

### Bootstrap Comparison

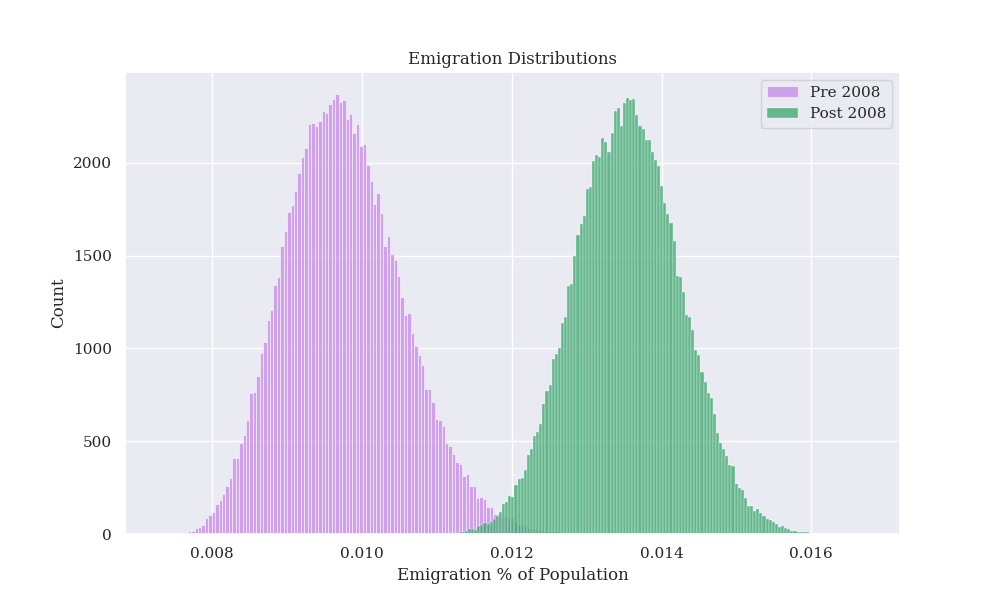
Even without knowing the exact distributions of these subsets, we can investigate whether they are drawn from the same distribution numerically using a non-parametric bootstrap (Efron, 1994). This allows us to construct confidence intervals for any statistic by the following procedure.

- A bootstrap sample is chosen by drawing samples from the data with replacement

- The chosen statistic (in this case the mean) is calculated on each bootstrap sample

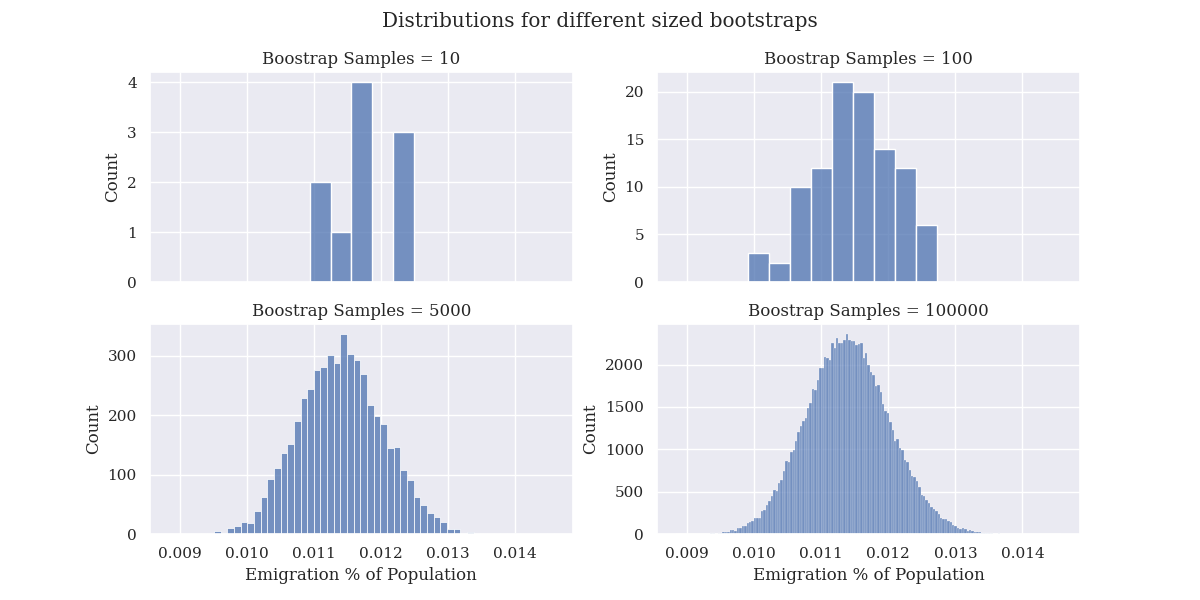
- Quantiles are derived for the computed statistics, and we can obtain a confidence interval from this (95% by convention)

Different distributions have different moments, so if two datasets result in non-overlapping bootstrapped confidence intervals, it follows that they are drawn from different distributions, as illustrated below.



The 95% confidence intervals for the means do not overlap, so we can reject the hypothesis that the data were drawn from the same distribution, and conclude that the mean emigration rate was significantly greater post-2008.

Additionally, we can observe that the distribution of means of our bootstrap samples are normally distributed thanks to the central limit theorem, with the distribution converging to normality as the number of samples increases. This occurs even when starting with a nonnormal distribution, such as the combined pre and post '08 emigration\_pop\_pct variable below.



## Predicting future Emigration Levels

We would like to be able to forecast emigration levels into the future based on past data. We can frame this as a supervised learning application, as we know the value of our target variable - the emigration level. We are making a prediction about a continuously valued target variable, so can be thought of as a regression problem. *Note: Encoding the next migration value as a boolean depending on whether it is positive allows us to frame our analysis as a binary classification problem, this is shown in section 4 of the notebook.*

As we have few features to work with, we can use the lags of variables to augment our data. This can be thought of as analagous to an autoregressive model, which are commonly seen in time series analysis (Chatfield, 2003).

It is also important that when we split our data into train and test sets, we set a temporal cutoff, and don't randomly sample data points as would be typical. If we were to randomly draw data points from the full history, data leakage could occur and the model could become informed of values in the test set based on information in the training set. This procedure of withholding the tail of a time series is commonly used in forecasting applications (Bergmeir & Benitez, 2012). The data preparation specific to ML applications can be found in section 3.1 of the notebook.

We compare multiple algorithms - OLS, linear regression with L1 and L2 regularization (Lasso & Ridge), ElasticNet, and Random Forest Regression, all made available through the scikit-learn package. (Sections 3.2-3.7 of the notebook)

Finally, we compare the performance of these algorithms to that of a method formulated specifically for time series analysis - ARIMA (Autoregressive Integrated Moving Average models), which can conveniently be tuned using the pmdarima package, which ports R's auto.arima() functionality to python. (Section 3.9 of the notebook)

### CRISP-DM

We are following the CRISP-DM process outlined by Wirth & Hipp (2000), which gives an iterative process for the progression of data mining projects. We have already gained an understanding of our data and done some preliminary data preparation in the previous sections. We will now do additional data preparation in order to frame our problem in a way compatible with the machine learning method we wish to apply, before actually applying those methods, evaluating them, and finally either deploying (or in this case accepting) the finalised model, or iterating through the previous steps in order to gain a deeper understanding and improve performance (Section 3.10 of the notebook).

### **Data Preparation**

Some additional data processing is required in order to get our dataframe into a format where machine learning algorithms can be applied. We first need to get our independent variables into their own dataframe; the continuous variables of year, population, migration, and emigration levels, and the categorical variables of sex and age group. We generate lags for each of population, emigration and migration, and also generate 1 lead term for emigration to serve as our target variable – the next value in the series. The number of lags is parameterised and will inform our train/test split criteria later, in this case 3 lag terms was selected.

As the first few years of the data don't have data in their lag terms, these come out to be null. We will ignore these rows and just deal with the rows with complete data - we could alternatively back-fill the earliest seen value, but in order to not bias the models towards older values this was not done and the rows were removed instead. The last year has no lead variable for the target, as that is in the future, so these rows are discarded as well.

We also choose a temporal cutoff point - before this is be our training set, and after this (plus our number of lags) is be our test set. The temporal gap between the two sets ensures that none of the values of the lags of the test set are present in the training set, so no data leakage should occur.

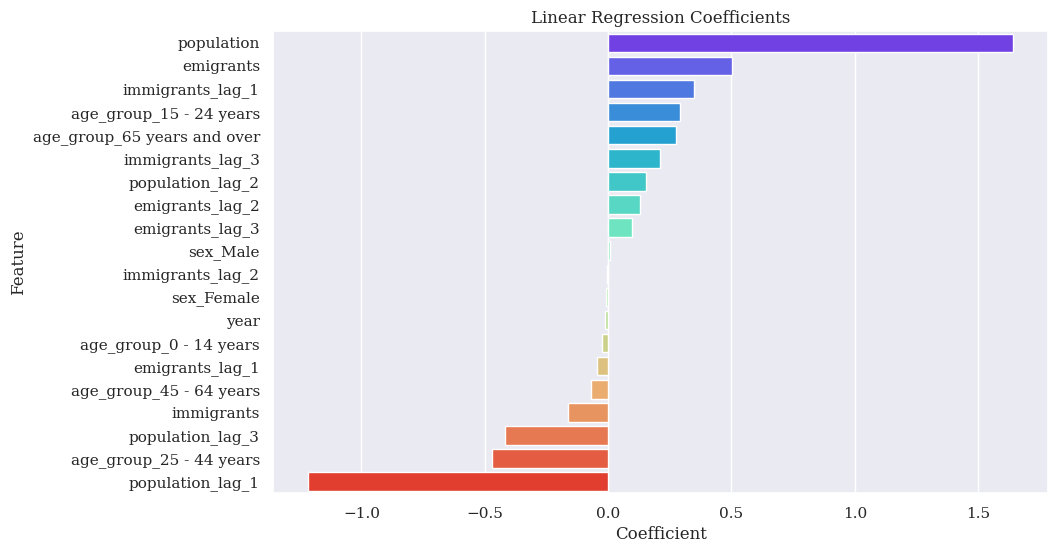
Multiple temporal cutoffs were tried and ultimately 2010 was chosen in order to strike a balance betwen having enough training data to effectively train the models, and enough test data to effectively evaluate them. A caveat to this though is that we have previously shown that the generating process behing emigration figures changed significantly post-2008, so our training data is unfortunately not necessarily representative of our test data, however this is a necessesity given the relatively small amount of data available.

We chose to scale the population, migration, and emigration values and their lags with a selective ColumnTransformer. We use StandardScaler to subtract the mean and divide by the standard deviation of each variable - centering and scaling the columns. This was fitted only to the training set, and the same columns in the test set were scaled using the transformations learned from the training set, again to minimise data leakage - If we applied the scaling to the whole dataset, information about the training set would be available to the test set as the scalers would use the mean and standard deviations of the combined train and test set.

We also one-hot-encode the sex and age group variable, as these are encoded as strings (an argument could be made for the age group to be encoded as ordered integers instead however).

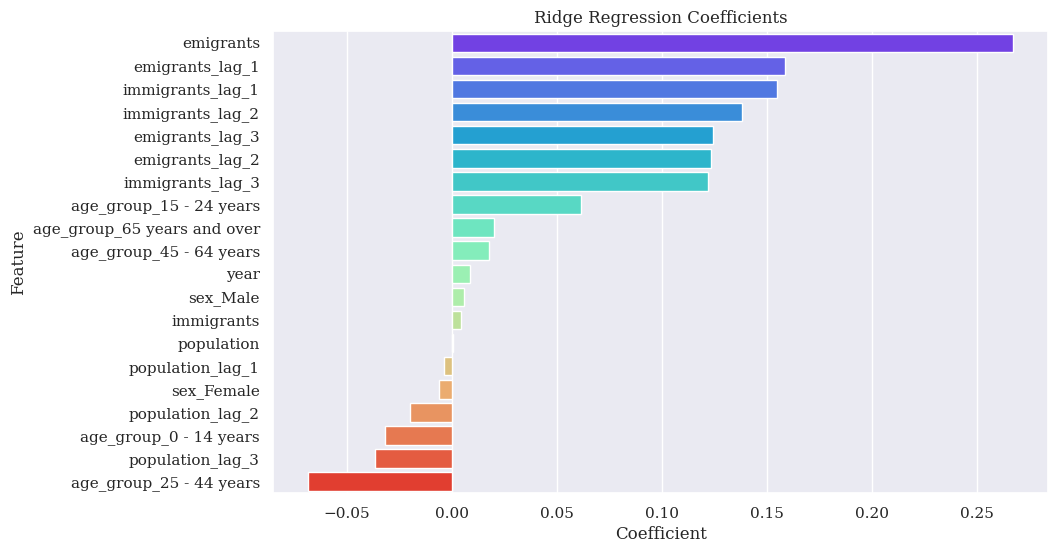
## Regression Methods Comparison

When we fit a simple linear regression model, we obtained a training R^2 value of 0.96, and a test R^2 of 0.87. As the test R^2 is quite a bit less than the training R^2, this indicates that the model has not generalised as well as we would like, and is at risk of being overfit (Frost, 2020).

We can see that the features the model treats as most important (high absolute value of coefficient), are the current and previous population values, the current emigration value, the previous immigration values, and the 15-24 and 25-44 age groups. The sex varibles, year, etc. are treated as less important.

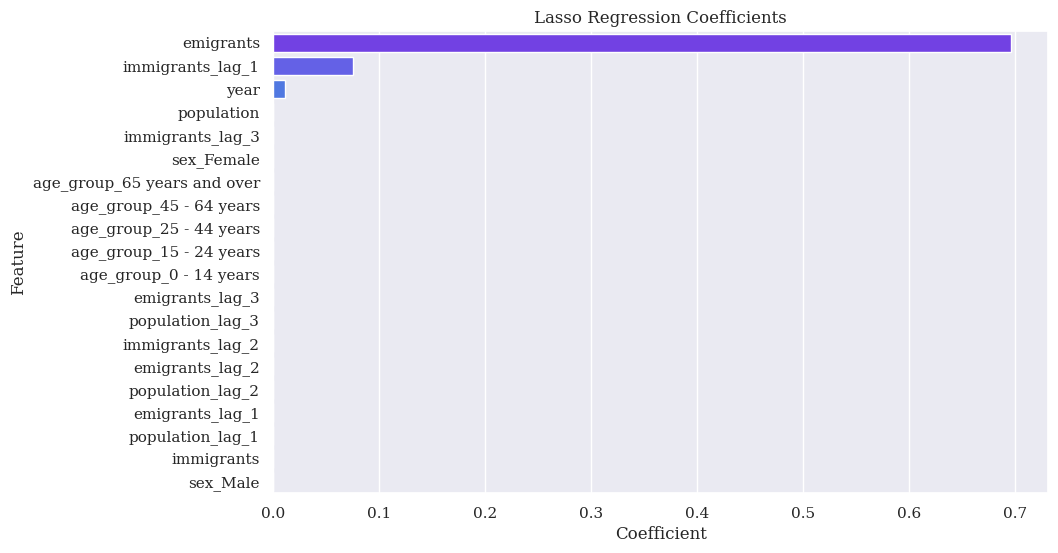
Most of our variables have high coefficient values, so we can now look at regularization in order to reduce the complexity of our models (Burkov, 2019) - this will reduce the absolute values of the coefficients and effectively prune the features considered.

Firstly L2, or Ridge regression. This takes a hyperparameter alpha which we can select automatically using a random search cross validator.



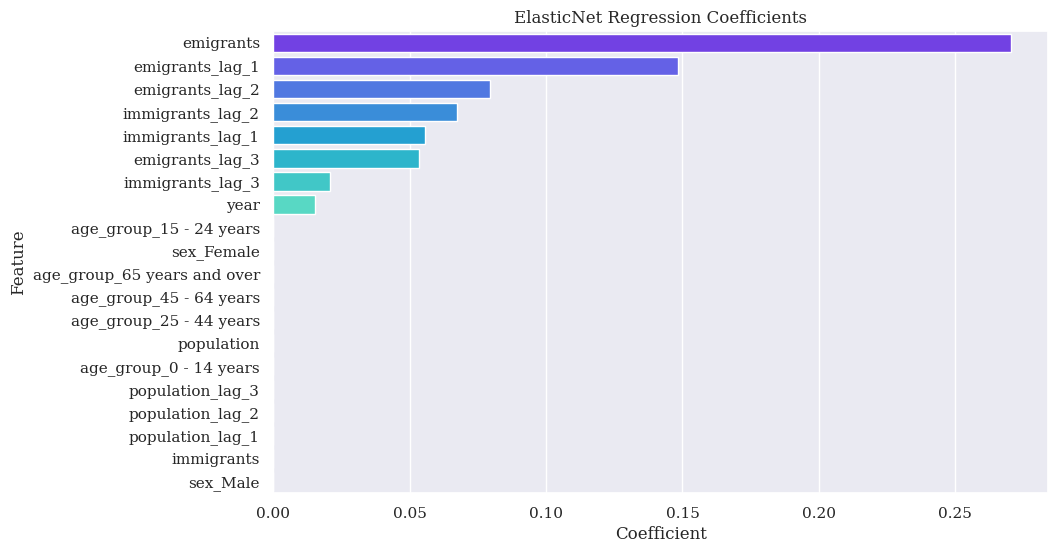
Although there is still a large difference between the train and test R^2 scores, the absolute value of our coefficients have been much reduced thanks to the regularization.

Lasso regression allows the absolute values of coefficients to go to zero - so in effect prunes the variables that the model uses.



We can see that the L1 regularization has allowed most of the coefficients to go to 0, and we are left with the current number of emigrants, the previous number of immigrants, and the year as variables that our model considers.

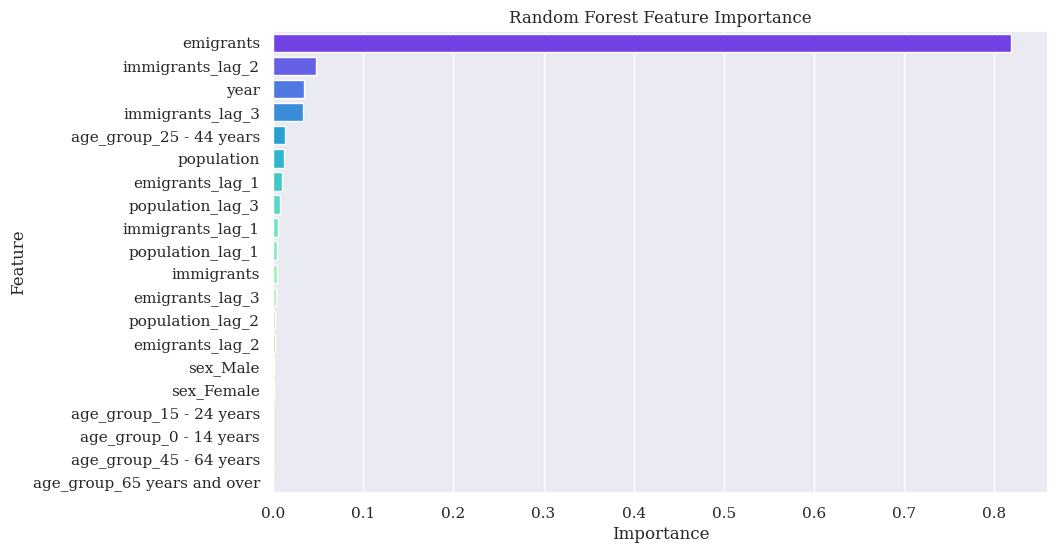
Our final linear regression technique is ElasticNet, which uses a mix of L1 and L2 regularization.

Migration Exploration & imputation

We can see that our elastic net model has not pruned the features as aggressively as Lasso, but has retained the significant features found previously via Lasso.

We can also use tree-based ensemble methods for regression - in this case, Random Forest, where the predictions of many individual trees are combined to product a final estimate (Burkov, 2019).

Random forest can give a sense of feature importance using the information on how much each individual tree improves with a split in each feature, averaged over each tree (Hastie et. al. 2009).



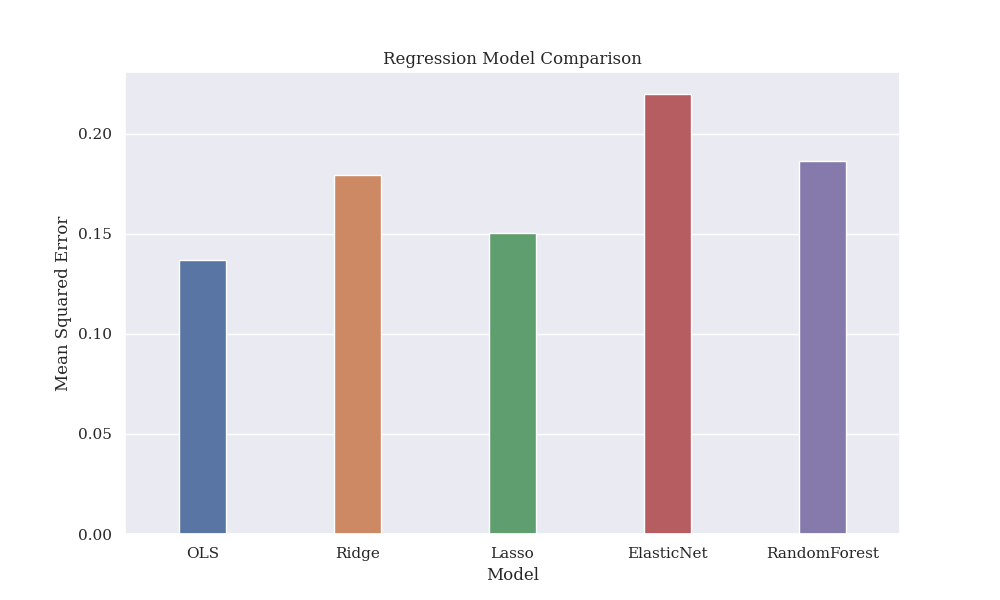
We can see that the random forest also selects the current emigration level as the most important feature.

In order to select hyperparameters for each of our models used two classes; RandomizedSearchCV and GridSearchCV - which fit models using many combinations of hyperparameters, score them, and provide the best model found within a given search.

RandomizedSearchCV takes distributions from which it randomly draws hyperparameter values (or randomly selects in the case of a list). In this case, we provide uniform distributions using the scipy library, however if we had more of an intuition of what values our parameters are likely to take, we could pass more complex arguments. GridSearchCV takes discrete values for the hyperparameters, and searches over combinations of these.

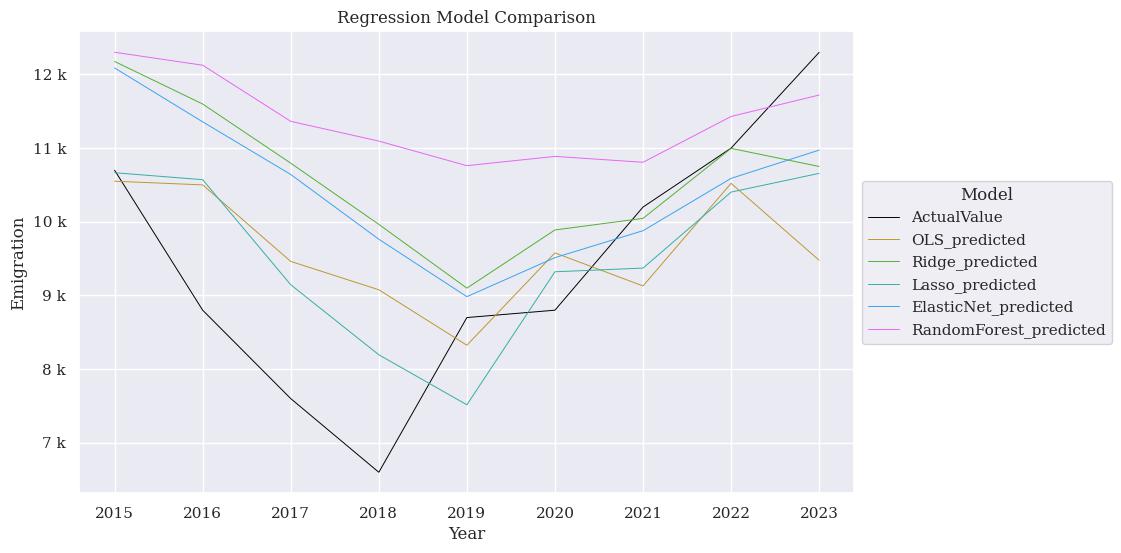
As we did some rough experiments with the regression models above, we were able to refine our search space for their values of alpha. In the case of Random Forest, we defined a grid of reasonable values to search through. Training many RandomForests is computationally intensive, especially with high numbers of very deep trees, so our grid is relatively small, but we could expand our search arbitrarily in order to further refine our model.

By taking the best estimator from each search object, we can compare the best model of each technique using their mean squared errors on the test set.



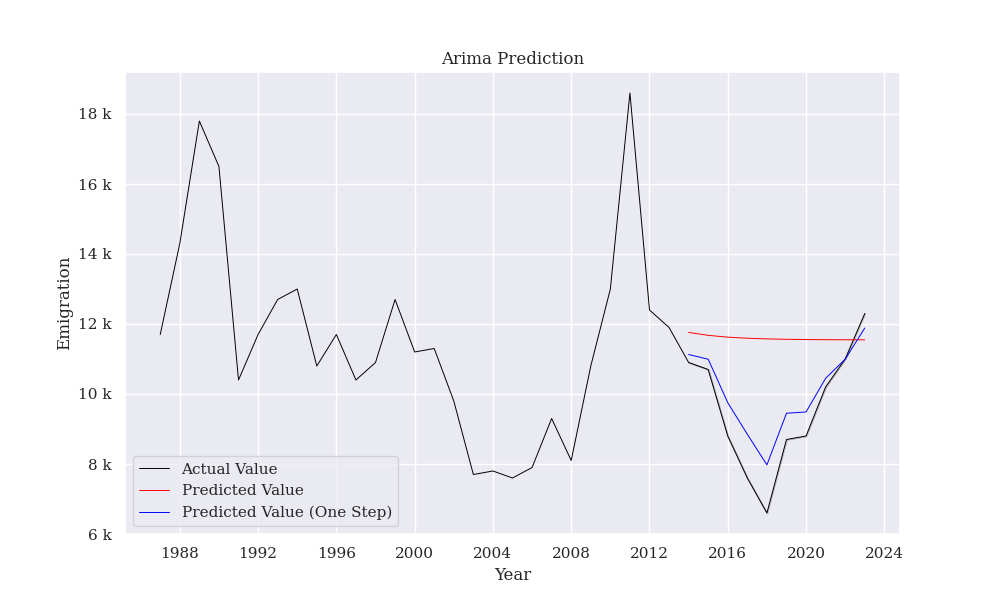
We see that OLS performed best, with Lasso in second. By parsimony, we might elect to choose the Lasso model over the OLS model despite its worse performance. This balance of complexity and accuracy is not reflected in the MSE, but other measures such as the AIC (often used in time series analysis) do penalise complexity. Given that the OLS AIC is less than the Lasso AIC (-44.78 < 13.39), we conclude that although the Lasso model is simpler, the performance decrease is too great to accept this model over OLS (Burnham et. al., 2002).

Zooming in on a single time series - where sex is Female and age group is 15-24 - we can see that, although we have trained 5 different models using hyperparameter tuning, none of them give consistently good predictions for the next value of emigration. It appears as though each trend is vaguely following the shape of the acutal values, but offset by 1 year. This observation makes sense, as we are using lag data, and the most important variable was consistently found to be the value of the year before.



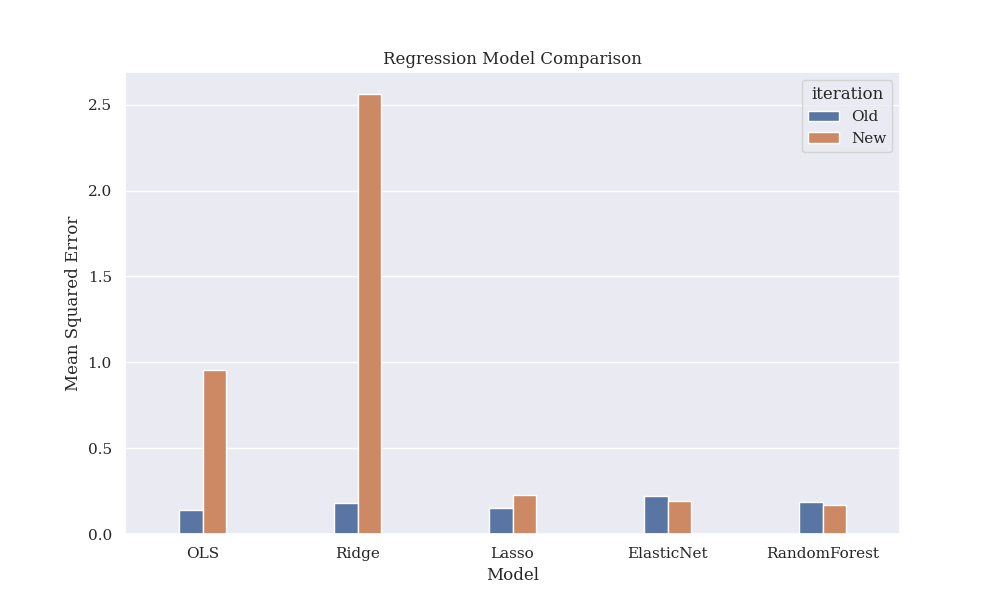
We can use a time-series-specific model (ARIMA) as a comparison. In this case, we are considering the time series from the above plot. We can see that auto arima has selected an AR(1), or Markov model (Chatfield, 2003), which is consistent with the results of the previous techniques - the emigration value immediately previous is the most important.

ARIMA can predict many steps ahead at once, however we can also progress one step at a time as our previous models have done. This changes the predicted results; as the AR(1) model only sees its previous value, the all at once prediction stays constant across the predicted years, but the one step prediction matches the true values more closely - resulting in a lower mean squared error.

****

After we have predictions and evaluated their performance, can go back and get a better understanding of our problem and data in order to iterate on our modelling techniques. We can use a PolynomialFeatures transformer to capture interaction terms and perform a polynomial regression (James et. al., 2013).

Repeating the above analysis with the newly transformed data, we see that OLS, Lasso, and Ridge regression perform worse with the new dataset. ElasticNet and Random Forest perform better, however the improvement may not be considered sufficiently better to justify the additional model complexity that our new transformer brings.



## Conclusion

We have cleaned, explored, visualised, and understood several aspects of population and migration trends in Ireland. We have shown that migration trends can be modelled using statistical distributions and we can both estimate the probability of certain observed trends in our data, and we can compare different timeframes and their underlying distributions. Finally, we have shown that the problem of predicting emigration levels can be framed as a supervised regression problem which is analagous to an autoregressive time series model.

**References**

* Bergmeir, C., Benítez, J.M., 2012. On the use of cross-validation for time series predictor evaluation. Information Sciences 191, 192–213. <https://doi.org/10.1016/j.ins.2011.12.028>
* Burkov, A., 2019. The hundred-page machine learning book. Andriy Burkov, Polen.
* Burnham, K.P., Anderson, D.R., Burnham, K.P., 2002. Model selection and multimodel inference: a practical information-theoretic approach, 2nd ed. ed. Springer, New York.
* Chatfield, C., 2003. The Analysis of Time Series, 0 ed. Chapman and Hall/CRC. <https://doi.org/10.4324/9780203491683>
* CSO, n.d. 2016 Census.
* Efron, B., Tibshirani, R.J., 1994. An Introduction to the Bootstrap, 0 ed. Chapman and Hall/CRC. <https://doi.org/10.1201/9780429246593>
* Frost, Jim, 2020. Regression Analysis: An Intuitive Guide for Using and Interpreting Linear Models.
* Gryclewska, B., 2016. Why women live longer than men? Review of biological and non-biological factors. Przeglad lekarski 73, 392–394.
* Hastie, T., Tibshirani, R., Friedman, J.H., 2017. The elements of statistical learning: data mining, inference, and prediction, Second edition, corrected at 12th printing 2017. ed, Springer series in statistics. Springer, New York, NY. <https://doi.org/10.1007/b94608>
* James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An Introduction to Statistical Learning, Springer Texts in Statistics. Springer New York, New York, NY. <https://doi.org/10.1007/978-1-4614-7138-7>
* Martin, R.C., 2012. Clean code: a handbook of agile software craftsmanship, Repr. ed, Robert C. Martin series. Prentice Hall, Upper Saddle River, NJ Munich.
* Mckinney, W., 2010. Data structures for statistical computing in python. Proceedings of the 9th Python in Science Conference.
* Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M., Stork, J., 2015. Comparison of different Methods for Univariate Time Series Imputation in R.
* Nichol, D., n.d. Coloring for Colorblindness.
* Tufte, E., 2001. The visual display of quantitative information / E.R. tufte. American Journal of Physics 31. <https://doi.org/10.1109/MPER.1988.587534>
* Weiss, N.A., 2017. Introductory statistics, 10th edition, global edition. ed. Pearson, Boston; Columbus; Indianapolis New York.
* Wickham, H., 2014. Tidy Data. J. Stat. Soft. 59. <https://doi.org/10.18637/jss.v059.i10>
* Wilson, T., 2016. Visualising the demographic factors which shape population age structure. Demographic Research 35, 867–890.
* Wirth, R., Hipp, J., 2000. Crisp-dm: towards a standard process modell for data mining.
* Yee, J.L., Niemeier, D.A., 1996. Advantages and disadvantages : longitudinal vs. repeated cross-section surveys.