Econ 178 - Final Project Writeup

Introduction

One of the questions that is often asked is a question about what causes an increase in wealth for households. Is it education? Is it income? Or is it property ownership? While it is difficult to determine the exact variables that can lead to an increase in wealth and conclude potential causations, it is possible to draw correlations by identifying potential predictors of wealth.

The study that will be analyzed is a sample of the data from the 1991 Survey of Income and Program Participation, specifically at the given 7933 observations that correspond to households with at least one employed person. This study aims to build a prediction model with the chosen predictor variables out of the 17 given predictor variables (or 15 if cutting out unneeded variables). Ultimately, we should be able to accurately predict total wealth using multiple methods when performing statistical analysis in R.

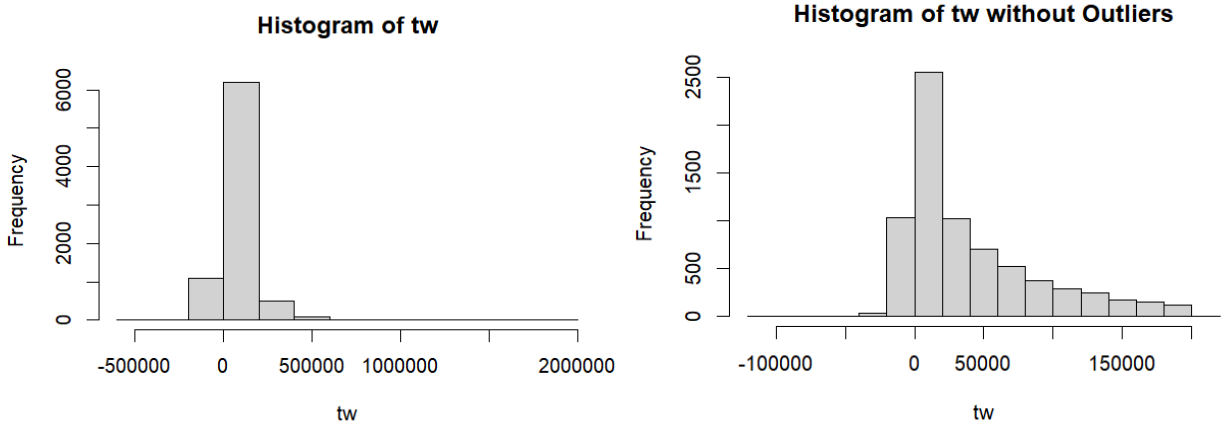
Statistical Analysis

Data Cleaning

Before building the prediction models, some corrections have to be made to the dataset. Initially, 17 dependent variables are stated in the introduction, but this will be cut down to 15, with home equity and no high school variables being excluded from the corrected dataset. Home equity is simply home value subtracted by the home mortgage, which becomes problematic due to perfect collinearity, which violates the OLS assumption about having no perfect collinearity and can be problematic as the predictive models can no longer determine the best fit due to solutions no longer being unique. No high school is a dummy education variable that is also removed since it could interfere with the intercept value when using OLS regression on the intercept.

Outliers can also interfere with the models so outliers have to be removed, especially when it comes to total wealth since numbers can get extremely high, breaking any observable relationships between predictors and total wealth. To remove outliers from the dataset, what constitutes an outlier had to be defined first. Outliers are mathematically determined as any total

wealth values that lie outside of the 25th quartile minus 1.5 times the interquartile range or 75th quartile minus 1.5 times the interquartile range. Using the subset() command, these outliers can be removed from the dataset to make sure that the data isn't too skewed to the upper extremes from the few extremely wealthy individuals. When doing some cursory tests with models with or without outliers, the models with outliers removed showed significant improvements in accuracy. As seen in the histograms below, the distribution of tw without outliers is significantly less skewed than the original distribution, which further shows why removing outliers was necessary.



Model Comparisons - No Predictor Transformations

There are four different methods that can be used in order to build a fitted model for total wealth in USD: OLS, ridge regressions, stepwise selection methods, and lasso. These four will not only be compared to each other through cross-validation, but also be compared across different transformations of plain predictors: polynomial transformations, spline basis representation, and generalized additive models (GAMs).

For the initial prediction model, predictor variables for OLS regression are chosen on an intuitive basis: age, marital status, income, and years of education. Age can be thought of as an accurate predictor of total wealth since older people would have much more wealth accumulated over the years. Marital status could possibly mean that unmarried people would have much more wealth since they do not have children to take care of. Income is also an obvious choice in the sense that higher income would result in higher total wealth. Lastly, years of education could also be used to predict wealth as households with higher education would be better equipped for

highly paid jobs that could result in increased total wealth. Aside from the chosen predictors, OLS regression will also be performed with all of the predictors and For the sake of comparison, OLS will also be performed with no predictors and all predictors. There are other ways, however, to choose predictors that can minimize bias.

Best subset selection could be used to select variables as it is a very thorough method when it comes to selecting predictors. However, it might be too inefficient due to the fact that there are 14 independent variables to sort through, meaning that it might result in the computer having to go through all $2^{14} = 16384$ models containing subsets of the 14 predictors. Ultimately, the best subset selection would be excluded from this study for the sake of efficiency.

Forward and backward stepwise regression are the next methods that are used to eliminate unneeded predictors. Both are much more efficient compared to the best subset selection as they only have to go through $1 + 14(14 + 1) / 2 = 106$ different models, which is so much more efficient that the accuracy loss from having fewer models becomes minimal. Forward stepwise starts with a null model which contains no predictors and adds a variable each step until a model with the lowest RSS is found. On the other hand, backward stepwise starts with a full model that contains all the predictors and removes a variable each step until a model with the lowest RSS is found. While the two methods may look familiar at a glance, the two can have different predictors from each other since they search through different models from each other. Both methods result in the same 10 predictors each, with the education dummy variables, family size, and marital status being eliminated as predictors.

Another way to approach the selection of predictors is through ridge regression. Unlike stepwise regression and lasso, ridge keeps all the predictor variables and tries to correct overfitting by shrinking the coefficients based on the tuning parameter λ. We then use the glmnet function with zero for alpha to run ridge regression. As for choosing the tuning parameter, it is done so by calculating the λ that gives us the smallest MSPE through cross-validation. In theory, with the bias-variance tradeoff, ridge regression could have a lower MSPE than OLS.

Finally, lasso regression will also be performed to build a model that can predict total wealth. While lasso is very similar to ridge regression in that the purpose is to correct the overfitting of the model, it's also able to exclude unnecessary coefficients, unlike ridge (hence why ridge isn't a feature selection method). Again, the steps to perform ridge are very similar with the only difference being that alpha is set to 1 when running the glmnet function on R. Unlike the stepwise methods where most of the predictors were kept, lasso only kept six predictor variables: IRA, non-401k assets, income, home mortgage, home value, and age.

After running a 5-fold cross-validation, and comparing the MSPEs of all the methods without the variable transformations, it was the OLS method with all the predictors and the stepwise methods that resulted in the lowest MSPE. OLS is only marginally higher than stepwise methods so there is no significant difference in using either when it comes to accuracy. When it comes to stepwise, there is no difference whatsoever between forward and backward directions since both eliminate the same predictors. It can be reasonably inferred that both the stepwise models and the OLS model (with all predictors) are the most accurate before applying any variable transformations. There may be an explanation as to why stepwise performed slightly worse. Since the stepwise methods used in R go forward and backward, they are not thorough enough when it comes to selecting the best predictors (as explained before, best subset selection goes through a lot more models compared to forward and backward, but it can be time-consuming) and thus would perform worse than OLS. Lasso and Ridge were also marginally higher in terms of MSPE compared to OLS, but not significantly so. On the other hand, the OLS models that used only the intercept and the chosen predictors had the highest MSPEs, so they will not be used in the models going forward when using cross-validations after transforming the predictors.

```
## Check the best
c(mspe.ols1, mspe.ols2, mspe.ols3, mspe_step_backward, mspe_step_forward, mspe.Lasso, mspe.ridge)
# 1739601805 2455317801  317342442  317357485  317387315  317383348  326766943
# OLS with all the predictors gives the best result when comparing just the linear models.
```

Transformation of Predictors - Intro/Troubleshooting

After comparing the four methods for the linear relationships between the predictors and total wealth, there is a way to ensure better model accuracy in terms of predictive power. When

using scatterplots to show the relationships between the predictors and the response variable, it's clear that not all relationships are linear. These non-linear relationships cannot always be shown accurately with linear regressions, so this is where transformations of predictor variables come into place. The first transformation that will be looked at is polynomials. In R, the command poly() will be used to transform variables into polynomials. For example, using poly(ira, 3) would transform the predictor ira into a polynomial of degree 3, which can help with prediction accuracy when it comes to non-linear relationships as seen in scatterplots.

Unlike in-class examples where the focus of flexible linear models is on only one predictor, this study will start with generalized additive models (GAMs) for the sake of accuracy. This is due to how if the models start with single predictors, many variables can change as more and more predictors get added throughout the project. To illustrate the problem with this single predictor approach, let's start with polynomials. We would need to choose the degree of polynomial transformation by running cross-validation to achieve minimal MSPE. However, this same cross-validation would likely give a different result if cross-validation is run with the regression that includes other predictors. Let's look at the polynomial of nifa (more in the next section, this is just for troubleshooting). When choosing the degree for nifa, the code in R would say that the polynomial of nifa with the degree of 10 is the best when running a regression that only includes the nifa polynomial as the sole predictor for total wealth. On the other hand, when running a regression that includes nifa polynomials alongside all the other predictors, this no longer becomes the case and the best degree becomes 5.

```
which.min(mspe_degree) # 5 is the best degree for nifa
which.min(mspe_degree2) # 4 is the best degree for inc
which.min(mspe_degree_simple) # 10 is the best degree for nifa if the nifa polynomial is the only predictor
```

Furthermore, when running prediction tests with single transformed predictors, the MSPEs are magnitudes higher than prediction tests that include other predictors. The accuracy loss is simply not worth using single predictor models for transformations. To reiterate, the reason why the models will start with all the predictors at once is due to accuracy issues and degree changes. As for finalizing the model, transformed predictors will be added to the existing linear regression while removing the previous non-transformed variable (e.g. with poly(nifa,5) around, the predictor nifa has to be removed from the regression or there may be conflicts).

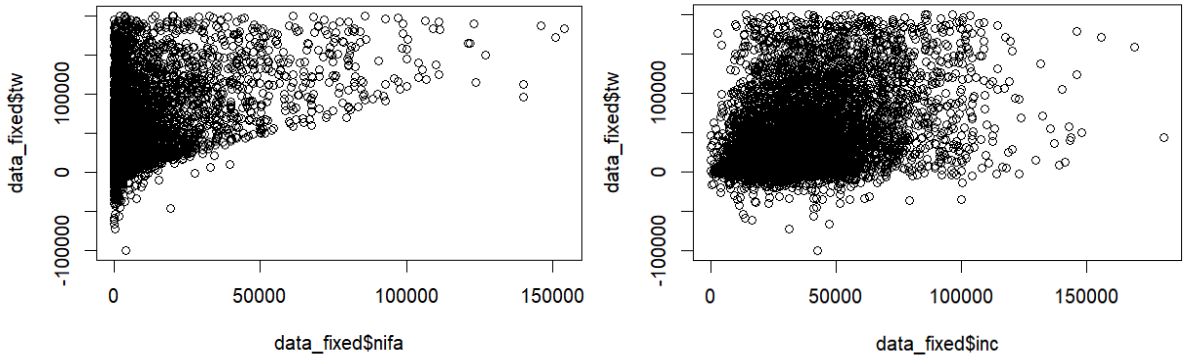
Furthermore, for the sake of simplicity, cross-validations for predictor transformations will only include OLS models before and after transformations to determine whether or not there is an improvement in the lowering of MSPE (of course, for the final model, all four methods will still be included to determine which results in the lowest MSPE). This means that we will use OLS models without transformations as a baseline model of sorts for the sake of comparison to determine whether the transformation helps with accuracy.

Transformation of Predictors - Polynomials

After comparing the four methods for the linear relationships between the predictors and total wealth, there is a way to ensure better model accuracy in terms of predictive power. For the sake of simplicity, cross-validations for predictor transformations will only OLS models before and after transformations with only one selected predictor (e.g income vs total wealth) to determine whether or not there is an improvement in the lowering of MSPE (of course, for the final model, all four methods will still be included to determine which comes out on top). This means that we will use OLS models without transformations as a baseline model of sorts for the sake of comparison to determine whether the transformation helps with accuracy. When using scatterplots to show the relationships between the predictors and the response variable, it's clear that not all relationships are linear. These non-linear relationships cannot always be shown accurately with linear regressions, so this is where transformations of predictor variables come into place. The first transformation that will be looked at is polynomials. In R, the command poly() will be used to transform variables into polynomials. For example, using poly(ira, 3) would transform the predictor ira into a polynomial of degree 3, which can help with prediction accuracy when it comes to non-linear relationships as seen in scatterplots.

For this particular model, the use of polynomial regression is rather limited due to the presence of categorical variables such as marital status and 401k eligibility. This is because polynomial regressions are better suited for numerical variables since the regression explores non-linear relationships between variables by raising predictor variables to numerous powers. Thus, only numerical variables will be looked at for this particular model.

Intuitively, non-401k assets and income are chosen for polynomial transformation since it's likely that they won't have a linear model with total wealth. For non-401k assets, a person can have really diverse investments and thus have a lot of total wealth despite having a few assets. With income, a person with higher income will most likely also pursue other activities that can increase total wealth that do not correlate linearly with income, such as buying property or investing. Furthermore, when plotting them against total wealth in the scatterplot, the relationship is visually nonlinear. Degree 5 is chosen for the polynomial transformation of nifa and degree 4 for inc since they yielded the lowest MSPE after cross-validation.



A 5-fold cross-validation test is then run to determine the accuracy of the new models, with a regular linear regression of predictors against total wealth as a control. The new model with all predictors alongside the polynomial transformations yielded great improvements in terms of accuracy, with the OLS model resulting in a reduced MSPE in comparison to the previous model that did not include polynomials. Ultimately, due to the improvements from the polynomial transformation of nifa and inc, this transformation would be kept for the final predictive model.

```
mean(MSPE.ols_poly) # 313854010 ## With transformed predictors
mean(MSPE.ols) # 317339936 ## This is the control when regressing with just plain predictors
```

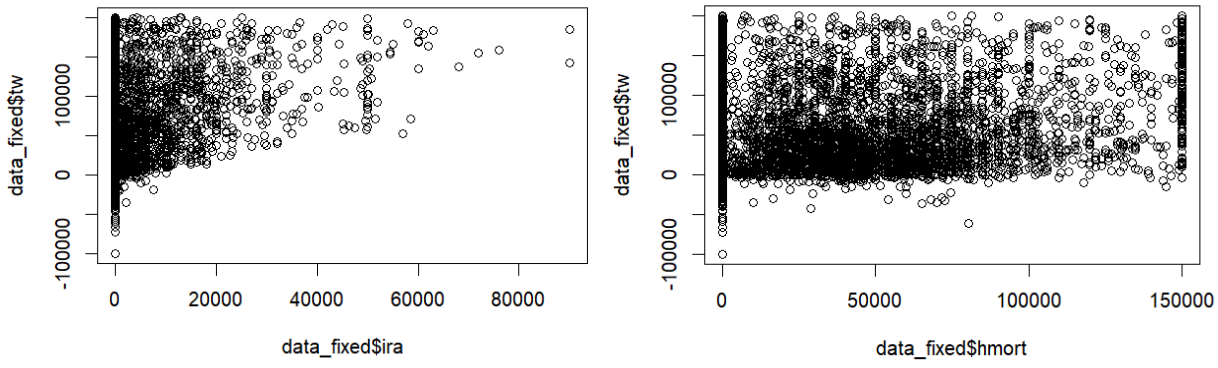
Note that there isn't any cross-validation between the feature selection methods and ridge yet. To reiterate, these cross-validations will be saved for the final method.

Transformation of Predictors - Splines

A spline is a piecewise polynomial and is even more flexible compared to the polynomial transformation that was done for the predictor variable non-401k assets. Splines are useful when polynomials aren't enough to accurately access nonlinear relationships. For example, a

scatterplot that shows a trend that seems to curve up and down multiple times can be better portrayed with splines compared to polynomials. In R, the splines package would be used as it contains useful functions for generating splines.

Home mortgage (hmort) and individual retirement account (ira) variables will be considered for spline transformation. The intuitive reasoning for this is that the relationship between these predictor variables and total wealth can be very complex and won't give a clear pattern that can be portrayed with polynomial regressions. For hmort, it is possible that a person with a high mortgage may have either high total wealth due to being able to afford a huge down payment or have low total wealth due to housing being financed by debt. For ira, a high ira could indicate a very fiscally responsible person who invested for retirement age and thus would have high total wealth. On the other hand, it could also mean that the person doesn't have much since said person would be putting all their wealth into retirement and thus not have much in the present, resulting in lower total wealth. Much like polynomials, scatterplots would be used to determine whether the selected predictors fit the criteria for transformation. In this case, the hmort was perfect for spline transformation since the relationship between home mortgage and total wealth may look linear, but it has many bumps that can be hard to portray with the usual polynomial transformation and linear models. Another variable that was also looked at for spline transformation is the individual retirement account variable (ira) due to its unusual scatter plot that also could not be accurately represented by increasing the power of the ira coefficient.



Afterward, the bs command can help generate basic splines that can be used for further testing. For example, running bs(ira, 3) would create a B-spline with 3 degrees of freedom for ira. Note that all the B-splines produced in this model have 3 knots, so they're all cubic splines. Back to the model, after running a test with multiple degrees of freedom from 1 to 20, 10 and 12

degrees of freedom for splines on hmort and ira results in the lowest MSPE for the model respectively. When the optimal knot is found, a cross-validation can be run to determine if there are any significant improvements.

Unfortunately, there appears to be no significant change whatsoever when using spline transformation on hmort. The OLS model with spline transformation has a marginally higher MSPE. It's possible that since hmort already looks somewhat linear to begin with, a linear model may work better for this case. Thus, splines on hmort will not be used for the final predictive model due to accuracy loss.

On the other hand, using spline transformation on ira was a great success as it resulted in a significant accuracy improvement, with the MSPE greatly decreasing. This could be because of the non-linear relationship between ira and tw that is better suited for splines than hmort and tw. Going forward, spline transformation on ira will also be included alongside any other transformations for the final model.

```
mean(mspe_OLS) # 317729111 ## Using B-spline on hmort worsens the accuracy
mean(mspe_OLS2) # 316370133 ## Using B-spline on ira increases accuracy
mean(mspe_OLS_control) # 317547401 ## Control model without any splines
```

Transformation of Predictors - Natural Cubic Splines

With the failure of the spline transformation on hmort, could the model have been improved if natural cubic splines were used instead? After all, there could have been overfitting issues when using splines on hmort, resulting in higher variance. The solution to this issue would be to use natural cubic splines instead. While conceptually similar to splines, natural cubic splines help free up degrees of freedom.

Again, the steps for running natural cubic splines are very similar to those of the previous splines, except that the command ns() is used instead of bs(). After running cross-validation, it was found that 2 degrees of freedom are the best for hmort transformation. After testing against the baseline model, it is shown that there is a small improvement in accuracy as the MSPE gets lower. Due to this, natural spline transformation will be used for hmort for the final predictive model.

```
mean(mspe_OLS) # 317514110 ## Using natural spline on hmort improves the accuracy
mean(mspe_OLS_control) # 317547401 ## Control
```

Transformation of Predictors - Final Model with GAMs

As noted in the section "Transformation of Predictors - Intro/Troubleshooting", the model already starts off with GAMs. This section will simply cover the final prediction model that combines all the transformation methods used previously. As stated before, the final model will also be cross-validated using all four methods: OLS, stepwise, ridge, and lasso to determine which model has the highest accuracy. For the final choice of predictors, all the predictors on the dataset will be included along with the transformed predictors: the degree 4 polynomial transformation of the income variable concerning total wealth: the degree 5 polynomial transformation of non-401k assets, the B-spline transformation of ira with 12 degrees of freedom, and the natural spline transformation of home mortgage with 2 degrees of freedom (the reason behind these transformations of selected predictors are explained in their respective sections). There will also be a control predictor model that only contains the plain predictors to see if the transformations help with accuracy.

After running a 5-fold cross-validation test, the feature selection methods had the lowest MSPEs and performed far better than the baseline model, with the two stepwise methods performing marginally better than lasso. The OLS model that included all the transformations also had significant improvements. On the other hand, the ridge methods performed worse than the baseline method. It's entirely possible that the reason ridge performed worse was due to the bias-variance tradeoff. Since the method relies on introducing bias to the models to reduce the variance, the bias ended up becoming higher than the reduced variance.

Here are the final results of the MSPEs of all the methods performed on the final model. Note that MSPE_control refers to the baseline model without any of the transformations that use the OLS method.

```
mean(mspe_OLS) ## 312393646
mean((pr.stepwise_backward-y)^2) ## 312320914
mean((pr.stepwise_forward-y)^2) ## 312320914
mean((pr.lasso-y)^2) ## 312580608
mean((pr.ridge-y)^2) ## 322128696
mean(mspe_control) ## 317547401
```

Conclusions

Summary

Initially, this project only started off with linear models since it was one of the easier models to start with. One of the early lessons that was learned is that intuition might not always be the best solution when it comes to picking out predictors, as seen with the model with chosen predictors having the highest MSPE overall, even higher than a model that only includes the intercept without any of the other predictors. When it comes to prediction models with a lot of predictor variables, it's very difficult to intuitively tell how each of these variables interact with each other, so this is where R comes to play as there are many functions and commands in R that can help with selecting the predictors: stepwise, lasso, and ridge (although ridge is strictly not a feature selection method).

With the plain predictor models, it was also surprising to learn that the OLS model with all the variables did better than stepwise, lasso, and ridge since the algorithms for the latter method would help select predictors (or reduce the coefficient of predictors in the case of ridge) for the model that can give the best results (lowest MSPE). While the project could have stopped here and the OLS model could have been used for the final predictions, the scatterplots between individual predictors and total wealth show that not all relationships are linear. This is where the transformation of variables became necessary in order to improve accuracy. These transformations may also end up changing which methods would become the best.

After running multiple tests, polynomial transformations of income and non-401k assets turned out to be very helpful when it comes to predictive accuracy. This is due to how the scatterplots for these two variables in relation to total wealth can be more accurately portrayed with a non-linear regression rather than a linear one. The next transformations will be spline transformations of ira and home mortgage. In all cases, the degrees for polynomials and degrees of freedom for splines are chosen through cross-validations.

With the transformations of plain predictors done, it was time to combine all the transformations into one model, which would include all the unchanged predictors alongside the transformed predictors. Much like the cross-validations done for plain predictor models earlier,

cross-validations will be used again for the new predictors for all four methods. Ultimately, the stepwise methods will be used for the final out-of-sample predictions.

Caveats

While the MSPEs have been lowered as much as possible, this is not to say that the predictions from the models will always be accurate. The removal of outliers was done to ensure that the model isn't too skewed due to possible measurement errors, but carelessly removing outliers can be detrimental in that the new dataset might end up being a poor representative of the original data, especially if these outliers tend to be closer to the norm for the overall population.

More tests could have been also run to determine the best degrees for the polynomials and cubic splines by using cross-validations for every step of the process whenever a transformation of a variable occurs. Cross-validations for degrees could have also been done across multiple methods but it turned out to be overly time-consuming. For example, a cross-validation for income polynomial degree involving forward and backward stepwise methods had to be removed since running the code would freeze up R and take 5-10 minutes just to give results, which would be an issue if the code is tweaked multiple times afterward. While the degrees and degrees of freedom used in the final product do improve accuracy, there was still room for improvements with the cut methods mentioned.