1-2021

Machine learning models for LoDI indices.

Lucas A. Bruns
University of Louisville

MACHINE LEARNING MODELS FOR LODI INDICES

By

Lucas Alexander Bruns
B.S., University of Louisville, 2020

A Thesis
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
In Partial Fulfillment of the Requirements
for the Degree of

Master of Engineering
In Industrial Engineering

Department of Industrial Engineering
University of Louisville
Louisville, Kentucky

August 2021

MACHINE LEARNING MODELS FOR LODI INDICES

By

Lucas Alexander Bruns
B.S., University of Louisville, 2020

A Thesis Approved on

July 22, 2021

by the following Thesis Committee:

Lihui Bai, Thesis Co-Advisor

Monica Gentili

Monica Gentili, Thesis Co-Advisor

Nick Hawkins

Erin Gerber

ACKNOWLEDGEMENTS

I would like to thank Dr. Lihui Bai and Dr. Monica Gentili for suggesting this topic, as well as for their support and guidance throughout all of the different iterations of this thesis. I appreciate your patience and kindness.

Thanks also to Dr. Xiaoyu Chen, who kindly agreed to help guide me through this machine learning research and is still willing to be my PhD advisor, Dr. Nick Hawkins, who offered his help when I was in great need, and special thanks to Dr. Erin Gerber, without whom this research would not have been possible, both literally and metaphorically.

Finally, thank you to my friends and chosen family who have supported me throughout this process and through all of my time at Speed School. Extra thanks to Sydney for loving and encouraging me. I couldn’t have done this without you.

ABSTRACT

MACHINE LEARNING MODELS FOR LODI INDICES

Lucas A. Bruns

July 22, 2021

Two indices published monthly by the Logistics and Distribution Institute (LoDI) predict changes in logistics and distribution activity levels nationally and regionally and are useful for organizations when planning projects and expenses. This research validates the current linear regression model, updates the index conversion method, and introduces machine learning models.

New source data are introduced to the models to validate the current linear regression model and a comparative analysis verifies that the current source data are robust. A rolling average is used for index conversion in place of a fixed reference month to reflect recent changes in employment levels.

Three linear machine learning models are tested on the data. Patterns among the residuals indicate non-linearity. Four non-linear models are tested on the data and compared to the linear models. The non-linear models are found to be more accurate than the linear models for both the national index and regional index.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The two indices published monthly by the Logistics and Distribution Institute (LoDI) were developed in 2012 at the University of Louisville (Gerber, 2013). One is a regional index for the Greater Louisville area and the other is a national index for the United States. Both are designed to predict changes in employment levels for the logistics and distribution industry and can be used by many organizations in their strategic planning of such activities.

After the creation of the indices in 2012, only one update to the regional index model has been completed (Eskridge & Gerber, 2016) and the national model has remained the same since its inception. Although the models are only nine years old, the constantly changing nature of the logistics industry necessitates regular revision so they don't become obsolete. This update seeks to address some of the problems that arise due to outdated data, such as the index conversion that relies on data from more than five years prior.

Though the current index models for the national LoDI index and the Greater Louisville area index are useful for predicting employment levels under normal circumstances, they are prone to overcorrection and due to the use of a linear rolling regression model, tend to lag behind the trends in the actual data.

This research attempts to account for the seasonal trends, as well as the long-term trends that occur due to the growth of the logistics and distribution industries over time, by using machine learning models, both linear and non-linear.

BACKGROUND

LoDI Indices

Rolling Regression Model

For both of the current indices, national and regional, many variables are considered in the regression model. The regional index uses data collected by the Ports of Indiana regarding barge (“River”) and railway (“Rail”) transit (Gerber, 2013). Barge data is reported as tonnage shipped, but railway data is reported in “carloads” shipped, requiring the use of a multiplier to calculate the estimated tonnage shipped. For the regional index, “Air” data is reported by the Louisville Regional Airport. Trucking tonnage data is not tracked in the same way for the region, so the American Trucking Associations (ATA) Truck Tonnage Index (TTI), which is seasonally adjusted, is used in place of a raw number.

Upon completion of the update in 2016, two new factors were added to the regional index model (Eskridge & Gerber, 2016). The Purchasing Manager’s Index (PMI) is reported monthly by the Institute for Supply Management and published on ycharts.com. It is based on a survey of managers in the area of supply management and purchasing. It is seasonally adjusted. The Kentucky Crude Oil First Purchase Price (KY Oil) is reported monthly by the U.S. Energy Information Administration (EIA). It represents the price of the initial removal of crude oil from a property.

For a response variable, the regional index model uses employment data published monthly by the Bureau of Labor Statistics (Gerber, 2013). The specific numbers used are for “Manufacturing” and “Trade, Transportation, and Utilities” employment, which are reported in thousands (i.e., 84.3 = 84,300 employed). Those numbers are not seasonally adjusted.

The national index uses Rail data from the Association of American Railroads (AAR), which, like the regional data, is reported in carloads shipped and must be converted into estimated tonnage (Gerber, 2013). Air data for the national index is published by the Bureau of Transportation Statistics and reported as total tonnage shipped. The ATA index is also used for the national model because raw tonnage data is unavailable. Total tonnage shipped by barge in the U.S. is published by the Federal Reserve Bank of St. Louis online database. It is collected and provided by the Army Corps of Engineers.

The national index also uses several economic indicators (Gerber, 2013). One is the estimated monthly Gross Domestic Product (GDP), which is published on ycharts.com. That value is calculated by a group called the Macroeconomic Advisors using the actual GDP value that is only published quarterly and yearly. Monthly crude oil prices are reported by the U.S. EIA. The PMI is also used for the national index.

The response variable for the national index is employment data, specifically “Transportation and Warehousing”, which is published monthly by the Bureau of Labor Statistics (Gerber, 2013). This data is also reported in thousands and is seasonally adjusted.

The original model for the regional index used the following variables: Air (lag = 3), Rail (lag=3), River (lag=3), ATA Index (lag=3), Air (lag = 4), Rail (lag=4), River (lag=4), ATA Index (lag=4), Air (lag = 5), Rail (lag=5), ATA Index (lag=5), and seasonal indicators (M2-M12) (Gerber, 2013). The original regional index model is shown in Equation 1.

$$\begin{aligned}
&\text{Total Employment} = 508 + 0.000501\ \text{Air} - 0.000104\ \text{Rail} + \\
&0.000031\ \text{River} + 0.745\ \text{ATA} - 1.90\ \text{M2} + 1.95\ \text{M3} + 12.3\ \text{M4} + \\
&26.4\ \text{M5} + 35.3\ \text{M6} + 15.1\ \text{M7} + 23.5\ \text{M8} + 25.0\ \text{M9} + \\
&20.4\ \text{M10} + 21.9\ \text{M11} + 19.9\ \text{M12} + 0.000515\ \text{Air}\ (\text{Lag} = 1) + \\
&0.000075\ \text{Rail}\ (\text{Lag} = 1) + 0.000025\ \text{River}\ (\text{Lag} = 1) - \\
&0.456\ \text{ATA}\ (\text{Lag} = 1) + 0.000340\ \text{Air}\ (\text{Lag} = 2) - \\
&0.000091\ \text{Rail}\ (\text{Lag} = 2) - 0.479\ \text{ATA}\ (\text{Lag} = 2)
\end{aligned} \tag{1}$$

The regional index model was updated in 2016 to a 36-month rolling regression model that better accounted for changes in the industry and to the data year over year (Eskridge & Gerber, 2016). Thus, there is no set model for each index calculation because the coefficients change each time a new month is added.

Changes were also made to the data used in the model (Eskridge & Gerber, 2016). The current regional model uses Air (lag = 3), Rail (lag=3), River (lag=3), ATA Index (lag=3), Purchasing Manager's Index (PMI), KY Oil, Air (lag = 4), Rail (lag=4), ATA Index (lag=4), River (lag=5), and ATA Index (lag=5).

The original model for the national index used the following variables: Air (lag=4), Rail (lag=4), River (lag=4), ATA Index (lag=4), Air (lag=5), Rail (lag=5), Air (lag=6), Estimated Monthly Gross Domestic Product (GDP), PMI, and Monthly Crude Oil Prices. No changes have been made to the model since its creation. The model for the national index is shown in Equation 2.

$$\begin{aligned} \text{T\&W Employment} = {} & 2708.79 - 1.64519e-005 \text{ Air } (L = 4) + \\ & 1.76987e-005 \text{ Rail } (L = 4) + 0.655678 \text{ River } (L = 4) - \\ & 0.230156 \text{ ATA Index } (L = 4) + 5.53457e-005 \text{ Air } (L = 5) + \\ & 3.11192e-006 \text{ Rail } (L = 5) + 3.09447e-005 \text{ Air } (L = 6) + \\ & 132.957 \text{ GDP for US } - 0.5871 \text{ PMI } - 1.20091 \text{ Oil} \end{aligned} \qquad (2)$$

Index Conversion

The existing formula for converting the national index is shown in Equation 3. It finds the ratio of the predicted employment to the base year employment, multiplies by one hundred, and then divides by two. This is done because the index is presented as a number from 1 to 100, with 50 indicating no growth or decline (Gerber, 2013). An index value between 1 and 50 indicates a projected decline of the logistics industry in the area the index represents, while an index value between 50 and 100 indicates a projected growth in the logistics industry.

$$\frac{\left({}^{Predicted\ employment}/_{Base\ yr\ employment}\right) * 100}{2} \qquad (3)$$

The existing formula for the Louisville index is shown in Equation 4. This formula was originally the same as that of the national index, but was changed in 2016 (Eskridge & Gerber, 2016). This updated formula uses the difference between the predicted employment and the base year employment instead of the ratio. The result is halved and then 50 is added to produce an index value between 1 and 100.

$$\frac{(Predicted\ employment\ -\ Base\ year\ employment)}{2} + 50 \tag{4}$$

The reason for this change of the conversion method is to amplify the effects of employment changes on the index (Eskridge & Gerber, 2016). Because the ratio of the predicted employment and the base year employment is often close to 1, the index values often fall within a very small window that makes it seem like there is very little change between months. To mitigate that issue and to provide an index value that the author felt better represented the actual changes to the employment numbers, the above index conversion method was created.

Another consideration when revising the existing model was that the "base year", on which all of the index calculations are based, had not been updated since 2013 for the national index (Gerber, 2013) and 2016 for the regional index (Eskridge & Gerber, 2016). This is problematic, considering that overall, the logistics industry has been growing consistently for at least the last decade. Therefore, employment data from 2015 cannot be accurately used to indicate the health of the industry. For example, the COVID-19 pandemic that began in 2020 caused a significant decrease in logistics employment for several months. Compared with the base year data from 5 years prior, however, the index showed that employment was still growing, albeit at a slower rate than months at the end of 2019.

In previous updates to the model, the solution to this problem has been to intermittently change the base year to a more recent one, thereby lessening the disparity between that data and the current month's predicted employment number (Eskridge & Gerber, 2016). However, that method doesn't prevent the accuracy of the index prediction from degrading over time.

Regularization

Ridge regression, introduced in 1970 (Hoerl & Kennard, 1970) uses an L_2 penalty to control the fitting of a regression model to a set of data. The purpose of this is to reduce overfitting of the model to the training data, a phenomenon that can occur when model complexity is increased and reduces a model's ability to "generalize" or predict values that are not in the training set (Hastie et al., 2017).

To reduce overfitting, ridge regression minimizes the residual sum of squares by penalizing large coefficients, which are thereby reduced. These penalties introduce bias into the model, which must be balanced with the variance of the coefficients to ensure that neither overfitting nor underfitting (when a model is not complex enough to account for correlation among data and does not generalize well) occurs (Hastie et al., 2017).

The most obvious drawback of the ridge regression method is that while it reduces the coefficients toward zero, it never results in any of them being exactly zero. This can cause resulting models to be difficult to interpret if the models are complex (has a large number of variables).

Lasso (Least Absolute Shrinkage and Selection Operator) was introduced in 1996 (Tibshirani, 1996) and, like ridge, can perform regularization of a regression model. A key difference between the two, however, is that lasso can also perform model selection by setting some of the coefficients equal to zero, due to its use of an L_1 penalty (Tibshirani, 1996). In this way, lasso can achieve a similar effect as best subset selection by removing some of the variables to create a sparse model (García-Nieto et al., 2021).

The difference between the penalties imposed by ridge and lasso is based on the L_p norm of a vector, specifically a coefficient vector in this case, which can be seen in Equation 5 (García-Nieto et al., 2021).

$$\|\beta\|_p = \left(\sum_{i=1}^{n} |\beta_i|^p\right)^{1/p} \tag{5}$$

Thus, the L_1 penalty for ridge regression is $\lambda \sum_{j=1}^{p} \beta_j^2$ and the L_2 penalty for Lasso is $\lambda \sum_{j=1}^{p} |\beta_j|$. In those penalty terms, λ represents the tuning parameter used to adjust the impact of the penalty terms on the coefficients (García-Nieto et al., 2021).

Though the introduction of bias to a regression model is intended to improve the ability of the model to generalize, one study has shown that ridge and lasso models performed similarly to other linear regression models (Chen et al., 2019). That research also concludes that the circumstances of the study could cause the results to change, so multiple models and algorithms should be tested to make the most informed decision about which one is best for a particular study.

Another study demonstrated that ridge and lasso could perform worse than an Ordinary Least Squares (OLS) model under certain circumstances (Melkumova & Shatskikh, 2017). That research concludes that the penalties imposed on the coefficients can cause the model to be less accurate unless the appropriate hyperparameter is chosen to optimize the shrinkage. This is supported by García-Nieto et al. (2021) in a study that compares several models, including lasso and ridge, which further explains that to find the best value for the hyperparameter, cross-validation should be used.

In addition to cross-validation, different algorithms can be used for model-selection to optimize the fitting of the model to the data as well as optimize the computational effort. One of those algorithms, Least Angle Regression (LARS), is a modified version of the Forward Stagewise method, which constructs a regression model by making tiny steps based on the largest correlation between a covariate and the current residual vector (Efron et al., 2004). The LARS algorithm uses the same parameters to select the next step, but takes much larger steps, reducing the computational effort. Another algorithm, coordinate descent (CD), constructs a model by iteratively selecting a direction and then minimizing a function in that direction (Wright, 2015).

Gaussian Process Regression

One way of accounting for non-linearity in a regression model is to use a Gaussian process, which uses the Gaussian probability distribution to generalize predictive functions

(Rasmussen & Williams, 2006). To create a predictive model, a set of functions are selected from the "prior" (a set of smooth functions that doesn't consider the data points), which are usually smooth for Gaussian processes, and are fitted to a set of observations. A mean function, called the "posterior," is then derived from the prior and weights of the model and is used for prediction.

For Gaussian processes, a kernel needs to be chosen. A kernel is the covariance function that determines the shape of the prior and the posterior, the parameters for which can be chosen automatically or manually (Duvenaud, 2014). Gaussian processes are non-parametric, meaning that there are not a fixed number of parameters, but rather the number of parameters depends on the size of the dataset, which means that the parameters, and therefore the prior and posterior change if new data is added to the model (Wang, 2021).

The following four kernel types are available in the scikit-learn library for Python. They can be used individually or in combination to create kernels of different shapes.

The constant kernel uses the equation seen in Equation 6 to construct a kernel.

$$k(x_1, x_2) = constant\_value \; \forall \; x_1, x_2 \tag{6}$$

A constant kernel is used to scale the other kernel when used as part of a product-kernel and when used as part of a sum-kernel, it is like adding a constant to the Gaussian process, which changes the mean.

The Exp-Sine-Squared kernel uses the equation seen below in Equation 7 to construct a kernel.

$$k(x_i, x_j) = exp\left(-\frac{2sin^2(\pi d(x_i, x_j)/p)}{l^2}\right) \tag{7}$$

where "l is the length scale of the kernel, p is the periodicity of the kernel, and $d(x_i,x_j)$ is the Euclidean distance" (Pedregosa et al., 2011). The Exp-Sine-Squared kernel is a periodic kernel that is used to model functions that repeat exactly.

The radial-basis function kernel (or squared-exponential kernel) uses the equation seen in Equation 8 to construct a kernel.

$$k(x_i, x_j) = exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right) \tag{8}$$

where "l is the length scale of the kernel and $d(x_i,x_j)$ is the Euclidean distance" (Pedregosa et al., 2011). The RBF kernel is "infinitely differentiable, which implies that GPs with this kernel as covariance have mean square derivative of all orders, and are thus very smooth." The RBF kernel can be used to model functions with local variation (Duvenaud, 2014).

The Rational Quadratic kernel uses the formula in Equation 9 to construct a kernel.

$$k(x_i, x_j) = \left(1 + \frac{d(x_i, x_j)^2}{2\alpha l^2}\right)^{-\alpha} \tag{9}$$

where "α is the scale mixture parameter, l is the length scale of the kernel and $d(x_i,x_j)$ is the Euclidean distance" (Pedregosa et al., 2011). The Rational Quadratic kernel is a "scale mixture of RBF kernels with different characteristic length scales."

If a single kernel doesn't fit the shape of the function being modeled, a combination of multiple kernels can be tailored more precisely. According to Duvenaud (2014), there are numerous ways that kernels can be combined, both through multiplication and addition. For example, if the RBF kernel and a periodic kernel like the Exp-Sine-Squared kernel are multiplied, a kernel is produced that can model local periodicity. If a linear kernel is combined additively with a periodic kernel, the resulting function will be periodic with a linear trend. Combining an RBF kernel additively adds variation.

METHODOLOGY

Updating Rolling Regression Model

Since the indices were created, there have been many changes in the logistics industry. To account for that, the first consideration when revising the model was whether new variables or source data should be included in the regression model.

The first potential source data update was to use the Trucking Tonnage Index (TTI) calculated by the Bureau of Transportation Statistics (BTS) based on the TTI calculated by the American Trucking Associations (ATA) instead of the ATA index, which is currently used in the model.

To determine if the change would be necessary or beneficial, the national index regression model was run using the BTS index values in place of the ATA index values. With the regression equations that were generated for each iteration of the rolling regression, the predicted employment values were calculated for each. Then, using the actual employment data, the residuals for each predicted month were calculated. The residuals for the original predictions (using the ATA index) were also calculated.

The second potential source data update was to add infrastructure spending to the national model. No specific data could be found that represented the Greater Louisville area exactly, therefore the use of this spending data was limited to being tested for the national model only. The infrastructure spending data is provided by BTS and is one field of a larger report called "Monthly Transportation Statistics".

Following the same method as for the ATA index vs. BTS index, to evaluate the change, the national index regression model was run with the existing variables and then with the added infrastructure spending with a lag of 3, 4, and 5 months. With the regression equations that were generated for each iteration of the rolling regression, the predicted employment values were calculated for each. Then, using the actual employment data, the residuals for each predicted month were calculated for both models. Once the predictions were made, the residuals could be calculated for each model.

Index Conversion

To fully address the issue of outdated data being used to covert the predicted employment value to an index value, a new method is proposed that uses a moving average of the actual employment data for the 12 months prior to the month being predicted instead of a fixed month. The formula for the proposed index conversion method is shown in Equation 10. As described, it uses the ratio of the predicted employment number of month *i* and the actual employment data for the most recent 12 months. Otherwise, the formula is the same as the existing formula for converting the national index.

$$\frac{\left(Pred_i \Big/ \frac{\sum_{j=i-1}^{i-1} Act_j}{12} \right) * 100}{2} \tag{10}$$

A limitation of the existing Louisville index conversion formula (see Equation 4) is that it requires the predicted employment to be within ±100 units of the base year employment, otherwise the index output would fall outside the specified 0 to 100 range.

Though the 2016 update to that formula was made to emphasize any growth or decline more obviously, the use of the difference between two employment values instead of the ratio means that the results are less representative of the true relationship between the two values.

The benefit of the proposed method over the existing index conversion method is that the resulting index value more accurately represents the changes in logistics employment in the last year as opposed to over the last five or ten years. Due to the lags in the variables themselves, there will still be a delay in any drastic changes to the pattern.

Machine Learning Models

The first step in preparing the data for use in a machine learning model was to determine whether or not it needed to be normalized. For the national index, the rail tonnage, river tonnage, ATA index, PMI, and crude oil price all followed an approximately normal distribution. The employment data, air tonnage, and GDP did not. Those three variables were log-transformed using natural log to normalize the distributions. For the regional index, only the ATA index had to be log-transformed.

Then, once the data had been normalized, it needed to be standardized so that the scale of each variable would be the same. This was accomplished using z-score normalization, which resulted in each variable having a mean of 0 and standard deviation of 1.

Additionally, to ensure that all of the models created would produce results that were replicable and comparable, the "train_test_split" module from the scikit-learn library for Python was used to split the data into replicable subsets of training and testing data. For the training data, 70% of the total data set was used each time, and 30% was used for testing. The exact subsets that were selected for training data and testing data were determined by passing an integer for the train_test_split parameter "random_state", which controls the way the data is shuffled before the split is applied (Pedregosa et al., 2011).

To further ensure that the results of each iteration of each model would be comparable, the training and test sets were compared to determine if the distributions were similar. To do so, histograms were created for each variable for both the training and test sets and the histograms were visually inspected for any significant differences.

Once the data was prepared, an ordinary least squares (OLS) linear regression model was created using Python to mimic the current regression model, though the Python model did not use a 36-month rolling regression. Instead, it used all of the available data for the local index (12 years or 144 months) and for the national index (14 years or 168 months).

To create this model, the "fit" method of the LinearRegression class in scikit-learn was used to fit the OLS regression to the training data (Pedregosa et al., 2011). Then the "predict" method was used with the test data to get a set of predicted values. Those predicted values were then compared to the test set of response data to find the root mean square error. This process was repeated for each run of the model with different random subsets of training and test data.

The first two machine learning models that were created were linear and implemented Lasso. The first Lasso model used a coordinate descent solver (LassoCV from scikit-learn) to cross-validate and select the best model (Pedregosa et al., 2011). The resulting alpha value is then used to fit the first Lasso model to the training data set and to predict values for the test set. Like the OLS model, the root mean square error was found for each set of training and testing data. The second Lasso model used the LARS algorithm (LassoLarsCV) for cross-validation and model selection (Pedregosa et al., 2011). The alpha value of that model was used to fit the second Lasso model to the training data and predict values for the test data. Like the previous models, the root mean square error was calculated for each run.

The last linear machine learning model used Ridge Regression. RidgeCV was used with a list of alpha values to determine the best model via cross-validation (Pedregosa et al., 2011). That model was then fit to the training data and used to predict values for the test data. The root mean square error was calculated for each run.

Once the linear models were created, they were also cross-validated using the leave-one-out method. By that method, each data point was predicted individually using all of the other data points as training data. The resulting values were used with the actual values to calculate the residuals and root mean square error for each data point. The residuals for each model were then examined to determine whether or not they indicated normality.

When residuals were analyzed for the linear regression models and a determination was made that there was autocorrelation among the data that wasn't accounted for, indicating that a non-linear model might be a better fit for the data, a Gaussian model was chosen to test that theory.

Another scikit-learn module, GaussianProcessRegressor, was used to create four additional models to test (Pedregosa et al., 2011). The difference between the four models was the kernel used. Though the data differs between the local and the national indices, the Gaussian models tested, like the linear models, are the same for both.

The GaussianProcessRegressor class accepts “kernel”, “alpha”, “optimizer”, “n_restarts_optimizer”, “normalize_y”, “copy_X_train”, and “random_state” parameters (Pedregosa et al., 2011). For the four models created for the LoDI indices, input is only passed for kernel, alpha, n_restarts_optimizer, and random_state.

Per the scikit-learn documentation (Pedregosa et al., 2011)., the alpha parameter is the value added to the diagonal of the kernel matrix to prevent an error during fitting. The n_restarts_optimizer indicates how many times the optimizer that finds the parameters that maximize the log-marginal likelihood of the kernel restarts. On the first run, the initial parameters set for the kernel are used, but after that, each restart uses different, randomly sampled values within the allowed range. The random_state parameter is used to initialize the centers, so if an integer is passed for it, the results can be reproduced for multiple iterations.

The first model used a product-kernel of the Constant (C) kernel and Exp-Sine-Squared (Exp) kernel. The parameters “length_scale” (which determines the effect of distance on correlation) and "periodicity" (which determines how often the pattern repeats) of the Exp-Sine-Squared kernel were set to 24 and 1, respectively.

The second model used a product-kernel of a Constant kernel and a Rational Quadratic (RQ) kernel. The parameters "length_scale" and "alpha" (scale mixture) of the Rational Quadratic kernel were set to 24 and 1, respectively.

For the third kernel, a product-kernel of a Constant kernel, Exp-Sine-Squared kernel, and Rational Quadratic was used. The parameters "length_scale" and "periodicity" of the Exp-Sine-Squared kernel were set to 24 and 1, respectively. The parameters "length_scale" and "alpha" of the Rational Quadratic kernel were set to 24 and 0.5, respectively, with the "length_scale_bounds" values set to 1e-5 and 2 and the "alpha_bounds" values set to 1e-5 and 100,000.0.

For the last kernel, the product-kernel of a Constant kernel, Radial-basis function (RBF) kernel, and Rational Quadratic kernel was used as part of a sum-kernel with a Exp-Sine-Squared kernel (i.e. (C * RBF * RQ) + Exp). The "length_scale" parameter of the RBF kernel was set to 24 with the "length_scale_bounds" values set to 1e-5 and 2. The parameters "length_scale" and "alpha" of the Rational Quadratic kernel were set to 1 and 0.5, respectively, with the "length_scale_bounds" values set to 1e-5 and 2 and the "alpha_bounds" values set to 1e-5 and 100,000.0. The parameters "length_scale" and "periodicity" of the Exp-Sine-Squared kernel were set to 24 and 1, respectively.

Once the kernel was set, the model was assigned the GaussianProcessRegressor class function with the kernel, alpha, n_restarts_optimizer, and random_state parameters specified. The kernel is assigned the respective kernel determined above, the alpha is set to 1e-1, the restart optimizer is set to 4, and the random_state is set to 0. The alpha, restart optimizer, and random state parameters have the same value for all four of the models.

Each non-linear model was run on the same sets of training and test data that the linear models were tested on to maintain replicability and comparability of results. For each run of each model, the root mean square error was calculated and recorded and, like the linear models, each non-linear model was cross-validated using the leave-one-out method. The average and standard deviation RMSE values for each model using that method were also calculated and recorded, as well as the residuals. The residuals were graphed and plotted to determine whether or not the non-linear models had accounted for the autocorrelation observed in the linear model residuals.

The RMSE values were normalized by the range of response values and compared for all of the models, as well as the normalized average and final RMSE values for the Leave-one-out validation.

ROLLING REGRESSION UPDATE ANALYSIS AND RESULTS

The residuals for the rolling regression model that used the ATA index and the model that used the BTS index were compared using a t test in Minitab to determine if there was a significant different between the means of the residuals. The results of that t test are shown in Figures 1 and 2. Based on the p-value, it is clear that the means of the two results are significantly different.

Because the difference is statistically significant, other factors were used to determine which index would be used. It can be seen in Figure 3 that the ATA index model residuals had a mean much closer to zero, as well as a smaller standard deviation than the residuals from the BTS index model. Therefore, the ATA index should not be replaced by the BTS index.

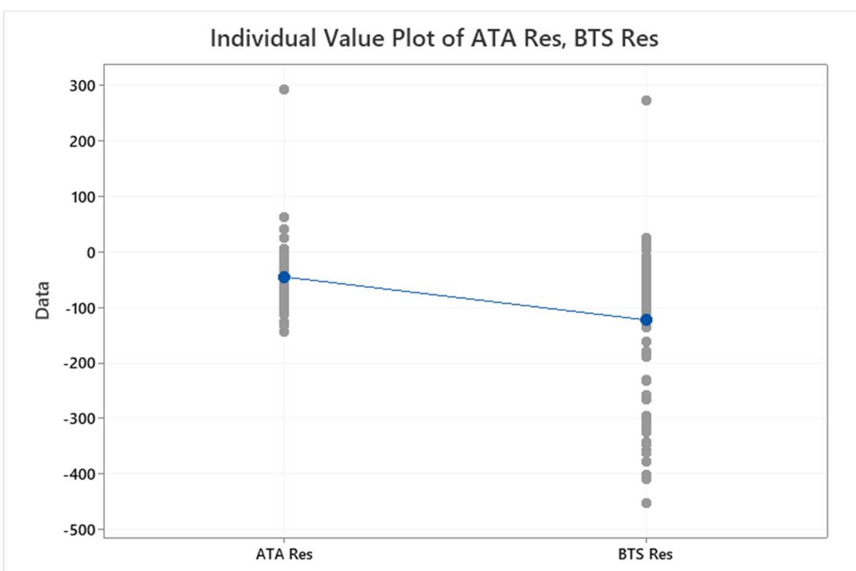

Figure 1. Plot of residuals for ATA and BTS indices from Minitab

Test

Null hypothesis H_0: $\mu_1 - \mu_2 = 0$

Alternative hypothesis H_1: $\mu_1 - \mu_2 \neq 0$

T-Value	DF	P-Value
5.90	137	0.000

Figure 2. Results of t test of ATA and BTS index residuals from Minitab

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
ATA Res	109	-45.9	47.6	4.6
BTS Res	109	-123	129	12

Figure 3. Descriptive statistics of ATA and BTS index residuals from Minitab

Another t test was used to compare the residuals of the national rolling regression model that used the ATA index without adding infrastructure spending and the model that added infrastructure spending to determine if there was a significant difference between the means. The results are shown in Figures 4 and 5 and based on the p-value (0.859), there is no significant difference between the residuals of the two models. Therefore, infrastructure spending should not be added to the model.

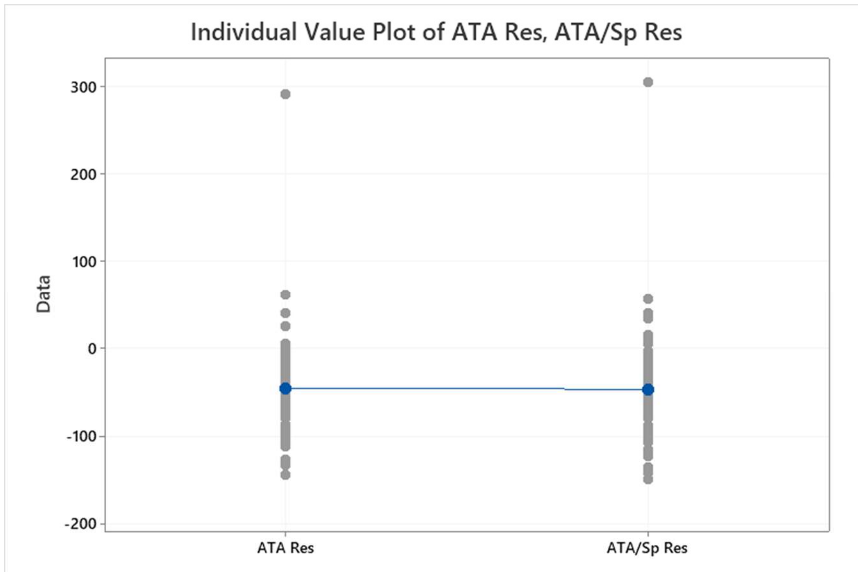

Figure 4. Plot of residuals for ATA index without infrastructure spending and ATA index with infrastructure spending from Minitab

Test

Null hypothesis $H_0: \mu_1 - \mu_2 = 0$

Alternative hypothesis $H_1: \mu_1 - \mu_2 \neq 0$

T-Value	DF	P-Value
0.18	215	0.859

Figure 5. Results of t test of ATA index without infrastructure spending residuals and ATA index with infrastructure spending residuals from Minitab

Descriptive Statistics

Sample	N	Mean	StDev	SE Mean
ATA Res	109	-45.9	47.6	4.6
ATA/Sp Res	109	-47.0	50.8	4.9

Figure 6. Descriptive statistics of ATA index without infrastructure spending residuals and ATA index with infrastructure spending residuals from Minitab

MACHINE LEARNING MODELS ANALYSIS AND RESULTS

Once all of the models are trained and tested and the resulting root mean square error values are collected, the models are compared. Table 1 shows the mean and standard deviation for the national index RMSE values across all runs. To see a table of the individual run RMSE data, see Appendix E. Table 2 shows the mean and standard deviation for the Greater Louisville area index RMSE values across all runs. To see a table of the individual run RMSE data, see Appendix F.

Table 1. Model comparison for national LoDI Index

Model		RMSE Mean	RMSE Standard Deviation
Linear	OLS	0.10044	0.05725
	CD Lasso	0.08069	0.03130
	Lars Lasso	0.10044	0.05725
	Ridge	0.08437	0.00768
Non-Linear (Gaussian)	C * Exp	0.05569	0.01281
	C * RQ	0.05555	0.01245
	C * Exp * RQ	0.05316	0.00744
	(C * RBF * RQ) + Exp	0.04500	0.00445

Based on the results for the national index, there is no significant difference between any of the linear models, nor between any of the non-linear models. However, the means of any linear model and non-linear model are significantly different. The means of the RMSE values for the non-linear models are all lower than the means of the linear models. Additionally, with the exception of the Ridge Regression model, all of the standard deviation values are lower for the non-linear models.

Table 2. Model comparison for regional LoDI index

Model		RMSE Mean	RMSE Standard Deviation
Linear	OLS	0.22150	0.01621
	CD Lasso	0.22156	0.01614
	Lars Lasso	0.22150	0.01621
	Ridge	0.22150	0.01621
Non-Linear (Gaussian)	C * Exp	0.14572	0.01729
	C * RQ	0.14517	0.01722
	C * Exp * RQ	0.14907	0.01714
	(C * RBF * RQ) + Exp	0.14579	0.01723

The results for the regional index model are similar to the national index model, but with even less difference between the means of all of the linear models and between the means of the non-linear models. Again, the means are all lower for the non-linear models than for the linear models, but the standard deviations for the linear models are slightly lower than those of the non-linear models.

The mean and standard deviation of the RMSE values from the Leave-one-out cross-validation are shown in Table 3 for the national index models and in Table 4 for the regional index models.

Table 3. Model comparison for national index leave-one-out validation

Model		RMSE Mean	RMSE Standard Deviation
Linear	OLS	0.05867	0.06770
	CD Lasso	0.05449	0.04307
	Lars Lasso	0.05867	0.06770
	Ridge	0.06326	0.04991
Non-Linear (Gaussian)	C * Exp	0.03548	0.03573
	C * RQ	0.03552	0.03618
	C * Exp * RQ	0.03698	0.03711
	(C * RBF * RQ) + Exp	0.03311	0.03045

For the national index, there is no significant difference between the means of any of the linear models or between the means of any of the non-linear models at $\alpha=0.05$. There is, however, a significant difference between the mean of any of the linear models and the mean of any of the non-linear models. The standard deviations are lower for all of the non-linear models than for any of the linear models.

Table 4. Model comparison for regional index leave-one-out validation

Model		RMSE Mean	RMSE Standard Deviation
Linear	OLS	0.17235	0.11804
	CD Lasso	0.17244	0.11750
	Lars Lasso	0.17235	0.11803
	Ridge	0.17235	0.11804
Non-Linear (Gaussian)	C * Exp	0.08920	0.08955
	C * RQ	0.08911	0.08931
	C * Exp * RQ	0.09004	0.09151
	(C * RBF * RQ) + Exp	0.08982	0.08869

For the regional index, there is no significant difference between the means of any of the linear models or between the means of any of the non-linear models at $\alpha=0.05$, but there is a significant difference between the mean of any of the linear models and the mean of any of the non-linear models. The standard deviations are also lower for all of the non-linear models than for any of the linear models.

CONCLUSIONS AND FUTURE RESEARCH

The results of the comparative analysis of the rolling regression models indicate that, if the rolling regression model is retained as the primary method of prediction for the LoDI indices, no data needs to be added or changed. The data is robust for the purposes of the rolling regression.

The index conversion equations should be changed to use a moving average of the pervious 12 months of employment data instead of a fixed month because it would better reflect recent trends in employment growth or decrease.

The results of the root mean square error analysis for the national index show that a non-linear model is more accurate than a linear model, which is also true for the regional index. However, this research does not conclusively select a specific non-linear model for predicting the LoDI indices. It only concludes that non-linearity is present in the data and that a non-linear Gaussian model is better suited for it than a linear regression model. Several kernel combinations were tested in this research, but there are more available to test and other parameters that can be adjusted to optimize a Gaussian model.

The results of the leave-one-out validation are also important because, while using the same set of data to break down into training and testing data facilitated replicability and comparability, the current rolling regression model uses 36 months of historical data to predict a single, future value. The RMSE values for all of the runs of the models are found by using 70% of the data to fit a model and predict the other 30%. However, for the leave-one-out validation, one data point is predicted using all of the other data. That the RMSE values for the leave-one-out method also indicate that a non-linear model is more suited for the data than a linear model and predicts more accurately makes it even more evident that a non-linear model should be implemented for predicting the LoDI index.

While this research evaluates the source data of the current model and proposes new models based on analysis of current data, it is possible that different source data would make the predictive power of the proposed non-linear models even better. More research and data validation should be done with Gaussian models and other non-linear models to determine which is the best fit.

REFERENCES

Chen, J., de Hoogh, K., Gulliver, J., Hoffmann, B., Hertel, O., Ketzel, M., Bauwelinck, M., van Donkelaar, A., Hvidtfeldt, U. A., Katsouyanni, K., Janssen, N. A. H., Martin, R. V., Samoli, E., Schwartz, P. E., Stafoggia, M., Bellander, T., Strak, M., Wolf, K., Vienneau, D., … Hoek, G. (2019). A comparison of linear Regression, regularization, and machine learning algorithms to Develop europe-wide spatial models of fine particles and nitrogen dioxide. *Environment International*, *130*, 104934. https://doi.org/10.1016/j.envint.2019.104934

Duvenaud, D. K. (2014). *Automatic model construction with Gaussian processes* (dissertation). University of Cambridge, Cambridge, England.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, *32*(2). https://doi.org/10.1214/009053604000000067

Eskridge, C., Gerber, E. (2016, May 21-24*). An Improved Index for Logistics Activity in Louisville* [Conference session]. IIE Annual Conference and Expo 2016, Anaheim, CA, United States.

García-Nieto, P. J., García-Gonzalo, E., & Paredes-Sánchez, J. P. (2021). Prediction of the critical temperature of a superconductor by using the WOA/MARS, Ridge, Lasso and Elastic-net machine learning techniques. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-021-06304-z

Gerber, Erin Lynn, "Development of a predictive index for the logistics and distribution industry." (2013). Electronic Theses and Dissertations. Paper 488. https://doi.org/10.18297/etd/488

Hastie, T., Friedman, J., & Tisbshirani, R. (2017). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, *12*(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

Melkumova, L. E., & Shatskikh, S. Y. (2017). Comparing ridge and LASSO estimators for data analysis. *Procedia Engineering*, *201*, 746–755. https://doi.org/10.1016/j.proeng.2017.09.615

Pedregosa, F., Profile, V., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., Contributor MetricsExpand All Fabian Pedregosa Higher Normal School - PSL Publication Years2011 - 2018Publ, Fabian Pedregosa Higher Normal School - PSL Publication Years2011 - 2018Publication counts10Available for Download5Citation count2, & Authors: Fabian Pedregosa View Profile. (2011, February 1). *Scikit-learn: Machine Learning in Python*. The Journal of Machine Learning Research. https://dl.acm.org/doi/10.5555/1953048.2078195.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Wang, J. (2021, February 3). *An Intuitive Tutorial to Gaussian Processes Regression*. arXiv.org. https://arxiv.org/pdf/2009.10862.pdf.

Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming*, *151*(1), 3–34. https://doi.org/10.1007/s10107-015-0892-3

APPENDICES

Appendix A

Table of Standardized National Data

Employment	Air (Lag=4)	Rail (Lag=4)	River (Lag=4)	ATA (Lag=4)	Air (Lag=5)	Rail (Lag=5)	Air (Lag=6)	GDP for US	PMI	Crude Oil Price
-0.6186	0.729984	0.004051	-1.01385	0.792207	2.298329	-0.07138	1.911919	-1.55301	0.421375	-0.47854
-0.59444	0.197386	0.000746	-1.2564	0.580579	0.712015	0.001379	2.26195	-1.54283	0.521617	-0.62352
-0.59176	2.390499	0.351827	0.785023	0.242019	0.186589	-0.00193	0.693078	-1.44686	0.36123	-0.51526
-0.55162	1.471544	0.026638	0.198872	0.535006	2.350169	0.349432	0.173429	-1.46192	0.581763	-0.20136
-0.51428	1.413241	0.320575	0.56269	0.49258	1.443589	0.023985	2.31322	-1.42683	0.120649	-0.12026
-0.49831	1.979051	0.048733	0.400993	0.492437	1.386071	0.318155	1.416608	-1.41185	-0.09988	-0.15635
-0.46114	1.29537	0.057525	0.320145	0.560976	1.944261	0.046097	1.359723	-1.41185	0.060503	0.017427
-0.45849	2.263818	12.74159	-0.06388	-0.03898	1.269786	0.054896	1.911776	-1.38198	-0.01969	-0.05662
-0.43995	1.900536	0.024534	0.097812	0.396931	2.225194	12.74901	1.244717	-1.35225	-0.22017	-0.4679
-0.41616	2.050438	0.026916	-0.02346	0.264188	1.866803	0.021879	2.189619	-1.33743	-0.4808	-0.66093
-0.38187	2.195998	0.244088	-0.24579	-0.06694	2.014687	0.024263	1.83517	-1.28828	-0.50085	-0.66135
-0.32932	2.39079	-0.04499	-0.40749	0.425836	2.158287	0.241607	1.981427	-1.28338	-0.18008	-0.56324
-0.3267	1.147482	0.187564	-0.89258	0.179561	2.350456	-0.0477	2.123448	-1.25408	-0.74143	-0.81762
-0.3267	0.642336	-0.04756	-2.00424	0.339799	1.12389	0.185038	2.313504	-1.18139	-0.20013	-0.63773
-0.3136	2.187925	-0.00976	0.360569	0.4426	0.625547	-0.05027	1.100425	-1.21037	-0.34047	-0.53336
-0.30574	1.300527	0.269229	0.400993	0.255896	2.150323	-0.01244	0.607561	-1.12381	0.000358	-0.36188
-0.2979	1.879425	-0.0111	0.785023	0.058635	1.274874	0.266768	2.115572	-1.10472	-0.05979	-0.32188
-0.30574	1.83764	0.017996	0.461629	-0.0095	1.845977	-0.01378	1.249749	-1.09996	0.060503	-0.16667
-0.30313	1.427219	-0.01698	-0.00325	0.02778	1.804754	0.015336	1.814572	-1.0952	-0.16003	0.074274
-0.3136	2.196574	0.019947	-0.81173	0.043491	1.399861	-0.01967	1.773803	-1.04307	-0.4808	-0.04267
-0.24832	1.639605	0.014833	-1.13512	0.116413	2.158855	0.017288	1.373361	-0.9773	-0.50085	0.124594
-0.26134	2.168283	0.296401	0.198872	0.126202	1.609386	0.01217	2.12401	-1.0101	-0.6011	0.381046
-0.27178	1.979817	0.000827	-0.20537	0.258261	2.130945	0.293962	1.580582	-0.98666	-0.44071	0.667542
-0.27961	1.827697	0.155084	-0.50855	0.613183	1.945017	-0.00185	2.096407	-0.93536	-0.84168	0.553613
-0.27178	0.992223	0.198479	-0.95322	0.850156	1.794945	0.152533	1.912523	-0.92144	-0.4808	0.801888
-0.26917	0.513651	-0.01966	-1.86276	0.754542	0.970722	0.195962	1.764102	-0.99603	-0.84168	0.90959
-0.25613	1.241602	-0.01095	-1.64043	0.623086	0.498595	-0.02235	0.948941	-0.95863	-0.76148	1.325252
-0.24571	1.497067	0.281398	-0.95322	0.499504	1.216743	-0.01363	0.482005	-0.93072	-0.80158	1.663559
-0.28483	1.329033	-0.00533	-0.26601	0.521094	1.468768	0.278946	1.192257	-0.91218	-0.66124	2.163881
-0.30574	0.668334	-0.02177	-1.41809	0.722713	1.302996	-0.00801	1.44151	-0.83394	-0.58105	2.537044

-0.33194	0.937892	0.231458	-0.20537	0.487959	0.651195	-0.02446	1.277562	-0.8431	-0.66124	2.565488
-0.37134	0.814985	0.014974	-0.16494	0.284025	0.917123	0.228967	0.632927	-0.90292	-0.74143	1.845649
-0.45849	0.500469	-0.03556	-1.9234	0.216329	0.795871	0.012311	0.895931	-0.91681	-1.94434	1.191579
-0.493	1.11893	0.257994	-0.38728	-0.12148	0.48559	-0.03826	0.776012	-0.96796	-2.90667	0.097785
-0.63204	-0.09131	-0.10762	-0.67025	-0.04306	1.095723	0.255524	0.469143	-0.98198	-3.24749	-0.82532
-0.68055	0.503578	-0.01927	-1.88297	-0.54145	-0.09822	-0.11038	1.072567	-1.11903	-3.92914	-1.42078
-0.74554	-0.93369	-0.20961	-2.54997	-0.72643	0.488658	-0.02196	-0.10825	-1.08094	-3.44797	-1.36821
-0.82728	-1.53188	-0.17528	-2.4287	-0.51738	-0.92926	-0.21245	0.472177	-1.08569	-3.38783	-1.2966
-0.92061	-0.81386	-0.18368	-1.80212	-1.04644	-1.51939	-0.17809	-0.93015	-1.10472	-3.26754	-0.9604
-1.03692	-0.75143	-0.2471	-1.82234	-1.18675	-0.81104	-0.1865	-1.51379	-1.0952	-2.60594	-0.79903
-1.0955	-0.73424	-0.26451	-1.7617	-1.03514	-0.74945	-0.24997	-0.81323	-1.08569	-2.20497	-0.47813
-1.14595	-0.68529	-0.0345	-0.99364	-1.10721	-0.73249	-0.26739	-0.75231	-1.18139	-1.64361	-0.06991
-1.21355	-0.26509	-0.19813	-0.79152	-0.9773	-0.6842	-0.0372	-0.73554	-1.18622	-0.78153	-0.19919
-1.23616	-0.43263	-0.16715	-0.69046	-0.61392	-0.26966	-0.20096	-0.68778	-1.14295	-0.30037	-0.00226
-1.25599	0.013522	0.047833	-2.59039	-0.77057	-0.43494	-0.16996	-0.2778	-1.12859	0.060503	-0.02768
-1.29006	0.65464	-0.17972	-1.94361	-0.76921	0.0052	0.045196	-0.44127	-1.04779	0.581763	0.158945
-1.32137	0.114503	-0.18949	-0.81173	-0.49351	0.637685	-0.18253	-0.00596	-1.0762	0.36123	0.251738
-1.28437	1.366198	-0.01831	-1.7617	-0.32938	0.104822	-0.19231	0.619566	-1.0952	0.702053	0.173577
-1.35848	-0.6945	-0.2155	-2.69145	-0.22216	1.339662	-0.02099	0.092561	-1.01951	0.762199	0.277485
-1.38424	-1.06951	-0.18833	-2.1053	-0.18776	-0.69329	-0.21834	1.313824	-0.98666	0.581763	0.221409
-1.35848	0.249484	0.099116	-1.11491	-0.20961	-1.06325	-0.19116	-0.69677	-0.94466	1.283458	0.354667
-1.33278	0.273683	-0.11611	-1.33725	-0.08571	0.237985	0.09652	-1.06267	-0.8983	1.223313	0.497913
-1.30997	0.099799	-0.13649	-0.28622	-0.1153	0.261859	-0.11888	0.22426	-0.88907	1.183216	0.107231
-1.2645	0.271189	0.076096	-0.24579	-0.22159	0.090316	-0.13927	0.247872	-0.90292	0.62186	0.142157
-1.22202	0.25158	-0.16219	-0.56919	-0.07695	0.259398	0.073482	0.078215	-0.86145	0.561714	0.196253
-1.23051	0.189292	-0.11577	-0.67025	-0.1881	0.240053	-0.16499	0.245438	-0.85686	0.902538	0.202552
-1.18815	0.28713	0.134489	-0.99364	-0.10955	0.178603	-0.11854	0.226305	-0.81566	0.702053	0.181045
-1.16563	0.612577	-0.10195	-0.30643	-0.0328	0.275124	0.131921	0.165531	-0.7838	0.822344	0.334317
-1.13192	0.188475	-0.1499	-0.79152	-0.0391	0.596189	-0.1047	0.260991	-0.79289	1.022828	0.457888
-1.12631	1.371566	0.02915	-0.89258	0.2856	0.177797	-0.15269	0.578525	-0.73854	0.882489	0.690917
-1.1235	-0.6594	-0.14481	-1.78191	0.745094	1.344957	0.026499	0.164734	-0.77926	1.403749	0.776453
-1.05363	-1.191	-0.15061	-1.84255	0.371506	-0.65866	-0.1476	1.319061	-0.77019	1.3837	0.929575
-1.02857	0.747501	-0.12365	-1.15534	0.625207	-1.1831	-0.15341	-0.66253	-0.65784	1.363652	1.3845
-1.0008	0.00059	0.131229	-0.14473	0.539622	0.729295	-0.12643	-1.1812	-0.6356	1.363652	1.818913
-0.98139	-0.34116	0.090009	-0.65003	0.26873	-0.00756	0.128659	0.710168	-0.6356	0.260988	1.587617
-0.9482	0.132244	-0.13527	0.360569	0.612419	-0.3447	0.087406	-0.01858	-0.68464	0.581763	1.476244
-0.95925	-0.30598	0.067236	0.764811	0.506605	0.122324	-0.13805	-0.35202	-0.59575	-0.30037	1.488373
-0.93992	0.018103	0.123654	0.966931	0.435894	-0.31	0.064615	0.109871	-0.35297	-0.07984	1.178568
-0.9344	-0.14614	-0.10265	0.320145	0.641932	0.00972	0.121078	-0.3177	-0.39997	-0.07984	1.256552

-0.92612	-0.05087	-0.08638	1.209476	0.701162	-0.15231	-0.10541	-0.00149	-0.27677	-0.22017	1.293575
-0.89859	-0.00353	0.125654	0.17866	0.746029	-0.05833	-0.08912	-0.16174	-0.31053	-0.13998	1.526363
-0.88209	1.012248	-0.15222	-0.04367	1.658445	-0.01162	0.123079	-0.06879	-0.31053	0.040455	1.476541
-0.83522	-1.12232	-0.14391	-0.0841	0.985836	0.990477	-0.15501	-0.0226	-0.25576	0.240939	1.410248
-0.79615	-0.73501	0.087502	-0.50855	1.059599	-1.11535	-0.1467	0.968479	-0.16408	-0.09988	1.519509
-0.77953	0.379815	-0.16203	0.198872	1.14202	-0.73325	0.084897	-1.11419	-0.19727	0.020406	1.632062
-0.82947	-1.50547	-0.16966	0.158448	0.988953	0.366561	-0.16483	-0.7363	-0.17236	0.160746	1.520761
-0.74174	-0.80635	0.057186	0.441417	0.847661	-1.49333	-0.17247	0.351423	-0.15994	0.060503	1.3097
-0.75287	-0.9642	-0.14797	0.037176	1.000728	-0.80363	0.054557	-1.48802	-0.16822	-0.38056	0.846873
-0.71435	-1.38791	-0.17728	0.097812	1.071374	-0.95936	-0.15076	-0.8059	-0.10639	-0.46076	0.88154
-0.6946	-0.59627	-0.04945	0.299933	0.941856	-1.37736	-0.18009	-0.95992	-0.13929	-0.36051	1.027879
-0.67515	-1.10426	-0.16754	-0.16494	0.988953	-0.59638	-0.05216	-1.37332	-0.0859	-0.13998	1.207037
-0.66921	-0.6748	0.081802	0.906295	0.412011	-1.09752	-0.17035	-0.60093	-0.10639	-0.34047	1.157384
-0.51428	-0.48796	-0.15531	0.219084	0.941856	-0.67386	0.079192	-1.09657	-0.09819	-0.68129	1.045556
-0.40164	-0.26134	-0.19088	0.198872	1.330409	-0.48952	-0.15811	-0.67756	-0.04919	-0.50085	0.863703
-0.43916	-1.02163	0.014335	-0.38728	1.49525	-0.26596	-0.1937	-0.49525	-0.00863	-0.11993	1.032506
-0.63634	-1.97191	-0.16906	-0.65003	1.565896	-1.01601	0.011672	-0.27415	-0.03294	0.040455	1.063232
-0.61135	-0.99362	-0.16615	0.441417	1.554122	-1.95349	-0.17187	-1.01595	-0.00459	-0.28032	1.070474
-0.58667	-1.41627	-0.17322	-0.30643	1.518799	-0.99897	-0.16896	-1.94312	0.015587	-0.58105	0.968093
-0.48157	-0.91881	-0.15406	-0.04367	1.84848	-1.40534	-0.17604	-0.98862	-0.03294	-0.58105	1.052548
-0.49725	-1.20199	0.068269	0.461629	1.836706	-0.91458	-0.15686	-1.40099	0.039717	-0.07984	0.989295
-0.53427	-1.27594	0.051916	0.259509	1.754285	-1.19394	0.065648	-0.91563	0.059759	0.401327	1.164144
-0.5023	-0.63045	-0.11644	0.603114	1.954449	-1.2669	0.049283	-1.19192	0.095684	0.682005	1.262479
-0.40296	-1.44348	-0.13174	-0.14473	2.025095	-0.6301	-0.1192	-1.26408	0.135374	0.62186	1.227631
-0.40111	-0.35344	0.098825	0.845659	1.742511	-1.43219	-0.13452	-0.63428	0.186622	0.74215	0.992053
-0.29502	-0.86328	-0.14347	0.097812	2.437197	-0.35682	0.096229	-1.42755	0.202312	0.822344	0.677434
-0.28326	0.099984	-0.19791	-0.24579	2.319454	-0.8598	-0.14626	-0.36401	0.206229	0.722102	0.675968
-0.24285	-0.97808	0.018868	-0.56919	-2.21935	0.090498	-0.20074	-0.86145	0.163018	-0.32042	0.631501
-0.25561	-1.93022	-0.17961	-0.83194	-2.01111	-0.97305	0.016209	0.078395	0.190548	0.060503	0.900083
-0.21945	-0.93892	-0.13425	0.603114	-1.95037	-1.91236	-0.18243	-0.97346	0.221873	0.160746	0.939844
-0.18831	-0.76937	-0.11195	0.421205	-1.85493	-0.93441	-0.13703	-1.90244	0.249162	0.401327	0.935242
-0.14538	-0.81404	0.131431	0.643538	-1.74213	-0.76715	-0.11471	-0.93525	0.303408	0.501569	0.979147
-0.11084	-1.21382	-0.11723	0.724386	-1.8289	-0.81121	0.12886	-0.76983	0.326522	0.481521	1.055461
-0.05665	-0.64944	0.121735	0.764811	-1.69007	-1.20561	-0.11999	-0.8134	0.372511	0.842392	0.982987
-0.02748	-0.78122	-0.08909	1.431809	-1.48183	-0.64884	0.119157	-1.20347	0.429557	1.223313	0.764186
-0.0096	-1.17257	-0.10685	0.704174	-1.48183	-0.77883	-0.09184	-0.65281	0.406797	0.74215	0.615296
0.024299	-0.27443	0.151066	1.73499	-1.52521	-1.16492	-0.10961	-0.78138	0.429557	1.223313	0.35613
0.111019	-0.85024	-0.13009	1.128628	-1.1174	-0.27887	0.148512	-1.16322	0.467318	1.163168	0.022775
0.207639	0.543826	0.099508	0.764811	-1.1174	-0.84693	-0.13287	-0.28691	0.452239	0.441424	-0.66107

0.174766	-0.94659	-0.13088	0.400993	-1.21285	0.528364	0.096913	-0.84872	0.452239	0.120649	-1.16813
0.198364	-1.46361	-0.18908	-0.4277	-1.56859	-0.94198	-0.13366	0.511446	0.512351	0.000358	-1.06276
0.203128	-0.91679	-0.16648	-0.32664	-1.40374	-1.45204	-0.1919	-0.94273	0.486119	-0.28032	-1.06358
0.231922	-0.74494	-0.1641	0.724386	-1.47315	-0.91258	-0.16928	-1.44718	0.531026	-0.28032	-0.87501
0.264888	-1.10162	0.028772	1.411597	-1.52521	-0.74304	-0.16691	-0.91366	0.560798	-0.01969	-0.62693
0.296518	-1.09558	-0.19156	0.785023	-0.99593	-1.09493	0.026121	-0.74598	0.601518	0.120649	-0.57363
0.331537	-0.71247	0.044236	0.582902	-1.24755	-1.08896	-0.19438	-1.094	0.623625	-0.03974	-0.84781
0.344425	-1.11189	-0.13459	1.108416	-1.35168	-0.71102	0.041596	-1.0881	0.638322	-0.36051	-1.22928
0.329057	-0.67685	-0.15799	0.138236	-1.43844	-1.10506	-0.13737	-0.71431	0.682221	-0.54095	-1.31095
0.334512	-0.20298	0.075377	1.492445	-1.2302	-0.67587	-0.16079	-1.10401	0.66762	-0.561	-1.26961
0.53829	-0.91896	-0.22775	0.441417	-1.33432	-0.20839	0.072762	-0.67955	0.663965	-0.86172	-1.40077
0.578434	0.741571	-0.08328	-0.04367	-1.29961	-0.91473	-0.2306	-0.21721	0.682221	-0.98202	-1.6477
0.527561	-1.00467	-0.28751	-0.30643	-1.33432	0.723446	-0.08602	-0.91578	0.671273	-0.94192	-1.88475
0.528537	-1.62184	-0.27857	-0.67025	-0.49268	-0.99928	-0.29041	0.704383	0.652987	-0.68129	-1.91418
0.542432	-0.10441	-0.10219	0.259509	-1.04799	-1.60814	-0.28147	-0.9994	0.729455	-0.22017	-1.67904
0.571389	-0.60098	-0.30674	0.17866	-1.24755	-0.11115	-0.10494	-1.60156	0.758357	-0.42066	-1.52332
0.58232	-0.81162	-0.29195	0.603114	-0.90048	-0.60103	-0.30966	-0.12103	0.747534	-0.32042	-1.31095
0.566527	-0.28142	-0.0625	0.360569	-1.09137	-0.80883	-0.29486	-0.60553	0.772761	0.060503	-1.18194
0.596152	-0.73306	-0.2413	0.239297	-1.343	-0.28577	-0.06522	-0.81104	0.808637	-0.05979	-1.29017
0.641902	-0.27263	-0.20005	0.502054	-0.77033	-0.73133	-0.24417	-0.29373	0.862091	-0.70134	-1.30263
0.634168	-0.35158	0.016087	0.138236	-1.54256	-0.2771	-0.20288	-0.73439	0.893959	-0.28032	-1.3172
0.665555	-0.46021	-0.20712	1.290325	-1.56859	-0.35498	0.013426	-0.28516	0.87273	-0.20013	-1.1559
0.707916	-0.16144	-0.22525	0.259509	-0.60548	-0.46215	-0.20996	-0.36219	0.932704	0.060503	-1.27702
0.958955	0.727034	-0.28296	0.158448	-1.28226	-0.16741	-0.2281	-0.46818	0.950241	0.321133	-1.07423
0.928571	-0.86707	-0.26433	0.643538	-0.93519	0.709104	-0.28586	-0.17668	0.957243	0.62186	-1.01781
0.946481	-1.44131	-0.22577	1.714778	-0.94387	-0.86353	-0.26721	0.690199	0.971225	0.962683	-0.97007
0.955661	0.217613	-0.03125	0.764811	-1.06534	-1.43004	-0.22862	-0.86514	1.016466	0.862441	-1.09948
0.97259	-0.58862	-0.24262	0.016964	-1.24755	0.206543	-0.03395	-1.42543	0.999102	0.381278	-1.06323
1.004502	-0.27307	-0.23456	1.916899	-0.43194	-0.58883	-0.24549	0.193164	1.061403	0.401327	-1.06801
1.017618	-0.14401	-0.00254	0.603114	-0.97858	-0.27753	-0.23742	-0.59347	1.092336	0.982732	-1.20259
1.035627	-0.67878	-0.24592	1.452021	-0.91784	-0.15021	-0.00522	-0.28559	1.102616	0.682005	-1.1733
1.051276	0.248035	0.016662	1.270112	-0.34517	-0.67778	-0.24879	-0.15967	1.13336	1.183216	-1.06206
1.093913	-0.26731	-0.22538	0.66375	-0.58812	0.236556	0.014	-0.68144	1.177519	1.584185	-0.93804
1.102516	0.184007	-0.20833	1.573294	-0.1109	-0.27185	-0.22823	0.222847	1.194426	1.163168	-0.89381
1.373861	0.451453	-0.01173	0.158448	0.166754	0.173389	-0.21117	-0.27997	1.251592	1.062925	-0.69377
1.393607	0.890438	-0.26304	0.340357	-0.10606	0.437235	-0.01441	0.160375	1.27499	1.283458	-0.64851
1.428253	-0.34438	-0.08493	-0.22558	0.211847	0.870307	-0.26592	0.421319	1.301629	1.243361	-0.56376
1.468434	-0.93952	-0.25217	-0.44791	0.105878	-0.34788	-0.08768	0.84963	1.328164	1.584185	-0.63972
1.512745	0.225774	-0.06672	0.744598	-0.02364	-0.93501	-0.25505	-0.35516	1.338087	1.283458	-0.68914

1.518132	-0.58807	-0.2201	1.290325	0.317816	0.214594	-0.06945	-0.93583	1.390763	0.882489	-0.51195
1.553536	0.166502	-0.00206	1.63393	0.364914	-0.58829	-0.22294	0.201127	1.462521	1.163168	-0.32756
1.587051	-0.07166	-0.19593	1.553082	0.294268	0.15612	-0.00474	-0.59293	1.456029	1.463894	-0.29518
1.605332	-0.27323	-0.22232	0.744598	0.553303	-0.07883	-0.19876	0.143296	1.507787	1.042877	-0.30671
1.65668	0.487709	0.052047	1.189264	0.282493	-0.27769	-0.22517	-0.08908	1.507787	1.684427	-0.35795
1.711341	-0.13864	-0.20726	0.845659	0.294268	0.473002	0.049413	-0.28575	1.523879	1.3837	-0.31188
1.748155	0.45644	0.013062	2.038171	0.95363	-0.14491	-0.2101	0.456694	1.575116	0.962683	-0.27299
1.811596	0.299499	-0.23629	0.825447	1.012502	0.442154	0.010398	-0.15443	1.552748	1.283458	-0.81809
2.000987	0.687521	-0.24369	0.825447	0.51798	0.287327	-0.23915	0.426184	1.555948	0.281036	-1.29135
2.065881	-0.14734	-0.06781	0.421205	0.859436	0.670123	-0.24656	0.27306	1.638614	0.74215	-1.00856
2.052372	-1.10758	-0.26156	-1.33725	0.647498	-0.1535	-0.07054	0.651647	1.613287	0.260988	-0.73061
2.045291	0.111187	-0.29287	-0.02346	0.364914	-1.10081	-0.26445	-0.16292	1.622796	0.481521	-0.55714
2.059664	-0.14867	-0.2278	0.987143	1.306861	0.101551	-0.29578	-1.09981	1.682706	-0.01969	-0.39231
2.059236	0.354699	-0.0246	1.270112	0.729918	-0.1548	-0.23065	0.089326	1.704644	-0.16003	-0.41039
2.113786	-0.14637	-0.24254	0.805235	0.565077	0.341783	-0.0273	-0.16421	1.717147	-0.24022	-0.69013
2.113145	0.101468	-0.047	0.704174	1.401055	-0.15254	-0.24541	0.326917	1.757623	-0.34047	-0.65534
2.096058	0.268858	-0.21655	1.654142	0.824113	0.091963	-0.04971	-0.16197	1.757623	-0.76148	-0.80892
2.114213	-0.48158	-0.26761	0.643538	0.965405	0.257098	-0.2194	0.079844	1.779316	-1.02211	-0.73517
2.120187	0.606803	-0.07919	1.63393	0.859436	-0.48323	-0.27049	0.243163	1.816341	-0.92187	-0.85589
2.145545	0.077283	-0.29763	0.865871	0.376688	0.590493	-0.08193	-0.48903	1.819417	-0.96197	-0.82774
2.309167	0.941154	-0.51759	0.785023	0.824113	0.068103	-0.30054	0.572892	1.847039	-1.1424	-0.77002

Appendix B

Table of Standardized Regional Data

Employment	Air (Lag=3)	Rail (Lag=3)	River (Lag=3)	ATA (Lag=3)	PMI	Ky Oil	Air (Lag=4)	Rail (Lag=4)	ATA (Lag=4)	River (Lag=5)	ATA (Lag=5)
-0.21077	0.253317	0.665299	0.442519	0.533786	-0.89266	1.260237	0.003722	0.236698	-0.04048	1.508695	0.425034
-0.30364	0.570882	0.65836	0.339448	0.77957	-0.7982	1.595697	0.247083	0.671459	0.542368	0.898994	-0.03807
-0.08308	0.629871	-0.7502	-0.56416	0.895692	-0.7982	2.168143	0.56581	0.664502	0.788925	0.429737	0.545432
-0.19336	0.136518	-0.35122	-1.02703	1.001109	-0.60928	2.584133	0.625014	-0.7476	0.905413	0.327391	0.79227
-0.53001	0.537578	-0.39633	-1.34485	0.924539	-0.57149	2.599778	0.129856	-0.34762	1.011161	-0.56986	0.90889
-0.36169	0.426984	-1.50305	-1.11469	0.503932	-0.72263	1.823939	0.532384	-0.39284	0.93435	-1.02948	1.014758
-0.58806	0.376557	-0.77449	-0.87816	0.642584	-1.55389	1.254715	0.421385	-1.50235	0.512421	-1.34505	0.93786
-0.6461	0.839355	-0.13612	-0.73894	1.010644	-2.66853	0.023312	0.370774	-0.77195	0.651508	-1.11652	0.515451
-0.34427	-0.21694	-0.94795	-1.0024	-0.94998	-3.12195	-0.8487	0.835265	-0.13198	1.020726	-0.88165	0.654697
-0.43714	0.517235	-0.33041	0.239463	-0.85979	-3.76428	-1.59601	-0.2249	-0.94585	-0.94606	-0.74341	1.024334
-1.10465	-0.40184	-0.63918	0.624215	-1.36489	-3.42422	-1.59003	0.511966	-0.32675	-0.85559	-1.00501	-0.94469
-1.16849	-0.70087	-1.79101	1.141251	-1.62608	-3.31087	-1.70415	-0.41047	-0.6363	-1.36228	0.22811	-0.85411
-1.23234	-0.1406	-2.34958	-0.16179	-0.51689	-3.21641	-1.29552	-0.7106	-1.79103	-1.62429	0.610154	-1.36138
-1.23815	0.095359	-1.44407	-0.4178	-0.85979	-2.55518	-1.19429	-0.14827	-2.351	-0.51161	1.123552	-1.62369
-1.26136	0.008015	-2.82488	-0.36173	-0.81496	-2.13955	-0.78842	0.088547	-1.44322	-0.85559	-0.17032	-0.50974
-1.26717	0.164795	-1.8257	-0.91706	-0.23703	-1.36497	-0.27442	0.000884	-2.8275	-0.81062	-0.42453	-0.85411
-1.4413	0.511108	-2.54733	-1.77729	-0.34386	-0.59038	-0.49483	0.158237	-1.82581	-0.23087	-0.36885	-0.80909
-1.38906	0.233052	-1.89162	-1.89399	-0.39765	0.089738	-0.22012	0.505817	-2.54925	-0.33803	-0.92028	-0.22869
-1.4355	0.761515	-1.75285	-0.65589	-0.18399	0.259769	-0.26245	0.226744	-1.89189	-0.39199	-1.77445	-0.33597
-1.42389	1.11427	-1.46489	-0.02108	0.004937	0.562045	0.01779	0.75714	-1.75277	-0.17767	-1.89033	-0.38999
-1.27297	0.241535	-2.53345	0.400979	-0.97264	0.259769	0.136973	1.111185	-1.46409	0.011856	-0.66095	-0.17542
-1.16849	1.599454	-0.72938	0.239966	-0.71473	0.429799	-0.00706	0.235258	-2.53534	-0.96879	-0.0306	0.014318
-1.569	0.457068	0.540402	1.973984	-1.2365	0.788752	0.12823	1.598144	-0.72673	-0.71008	0.388489	-0.96744
-1.85921	0.257559	0.304485	0.829819	-1.29468	0.52426	0.009967	0.451579	0.546247	-1.23349	0.228609	-0.70844
-1.56319	1.229969	0.706931	1.865954	0.691697	1.091028	0.216121	0.25134	0.309737	-1.29185	1.950425	-1.23244
-1.50515	1.115526	0.214282	0.748784	0.190787	0.958782	0.373958	1.227308	0.713196	0.700775	0.814312	-1.29087
-1.39486	0.921908	-1.23591	-0.45535	-0.07865	0.996566	-0.05446	1.112446	0.219307	0.19829	1.843156	0.70402
-1.34843	1.236803	-0.71898	-0.73715	0.622879	0.637614	-0.03237	0.91812	-1.23453	-0.07199	0.733847	0.200964
-1.49934	1.214103	-0.96183	-0.85706	0.036123	0.637614	0.004445	1.234166	-0.7163	0.631742	-0.46181	-0.06962
-1.30199	1.077588	-1.33305	-1.67228	0.393798	0.93989	0.026993	1.211383	-0.95976	0.04314	-0.74163	0.634908
-1.25556	1.282281	-0.68428	-0.55625	0.313026	0.618721	-0.02823	1.074369	-1.33192	0.401939	-0.8607	0.045637
-1.13367	1.421938	-0.41714	0.213648	0.353483	0.883213	0.255235	1.279811	-0.68152	0.320914	-1.67018	0.404844

-0.93632	1.038236	-0.93755	0.286795	-0.1312	0.864321	0.368896	1.419978	-0.4137	0.361498	-0.56201	0.323727
-0.72736	1.648624	-0.34775	0.760626	-0.11015	0.845428	0.581953	1.034873	-0.93542	-0.12471	0.202476	0.364358
-1.27297	0.436724	-0.13265	1.019976	-0.48424	1.147705	0.620146	1.647494	-0.34414	-0.10359	0.275108	-0.1224
-1.30199	0.290313	-0.37898	0.723125	-0.80378	1.242166	0.608182	0.431161	-0.1285	-0.47886	0.745605	-0.10126
-1.25556	1.745944	1.529169	1.524078	1.367201	1.166597	1.197654	0.284214	-0.37545	-0.7994	1.00313	-0.47696
-1.19171	0.925993	0.869991	-0.00097	0.42394	1.185489	1.524372	1.74517	1.537502	1.378404	0.708368	-0.79786
-1.1801	0.815713	0.033875	-0.45408	0.672077	0.165307	1.141514	0.922219	0.876666	0.432176	1.503685	1.382419
-1.11045	1.41102	-1.02081	-1.186	1.227783	0.732075	0.916493	0.811536	0.038447	0.681094	-0.01063	0.435116
-1.34262	0.84886	-0.88897	-1.70505	0.211251	-0.0614	0.944563	1.40902	-1.01889	1.238548	-0.46056	0.684316
-1.12786	1.304667	-1.11448	-1.31816	1.34871	-0.04251	0.457248	0.844804	-0.88672	0.218819	-1.18732	1.242403
-1.12786	1.226513	-0.85081	-1.41947	0.982015	-0.0614	0.428718	1.302279	-1.1128	1.359855	-1.70272	0.221516
-1.08723	1.006817	0.242037	-0.18985	0.895692	-0.307	0.454487	1.223839	-0.84847	0.992006	-1.31856	1.363848
-0.84925	1.372061	0.609789	-0.0861	0.593259	-0.25032	0.9372	1.00334	0.247132	0.905413	-1.41915	0.995582
-0.7854	2.443126	0.949786	0.200506	0.672077	-0.04251	1.068348	1.369919	0.615809	0.602028	-0.19818	0.90889
-1.11045	0.332492	1.19958	1.343908	0.914932	-0.02362	1.152558	2.444902	0.956662	0.681094	-0.09516	0.60516
-1.1801	0.17367	0.706931	0.796583	0.972456	0.014169	1.249653	0.326548	1.207084	0.924712	0.189427	0.684316
-0.99436	1.417539	1.716515	1.081541	1.039202	0.051954	1.427277	0.167145	0.713196	0.982417	1.324783	0.928211
-0.91891	0.454004	1.952431	0.025277	0.914932	0.221984	1.304873	1.415564	1.725319	1.049374	0.78131	0.985982
-0.7738	1.437412	-0.48306	-0.49243	0.799006	0.070846	1.053622	0.448504	1.961829	0.924712	1.064262	1.053014
-0.67512	1.110028	-1.0902	-1.4389	0.924539	-0.32589	0.374878	1.435509	-0.47979	0.808422	0.015431	0.928211
-0.67512	0.607092	-0.96183	-1.29009	0.924539	-0.5526	0.587475	1.106928	-1.08845	0.93435	-0.49864	0.811789
-0.61708	1.590814	-0.69816	-1.67292	0.876421	-0.43924	0.892104	0.602152	-0.95976	0.93435	-1.43844	0.93786
-0.57645	0.943901	0.009589	-1.44185	0.914932	-0.21254	0.942722	1.589508	-0.69543	0.88608	-1.29068	0.93786
-0.5126	1.364677	0.734686	-0.04875	0.43397	-0.43924	0.715401	0.940232	0.0141	0.924712	-1.67082	0.889535
-0.28042	1.609115	-0.24714	-0.38331	0.876421	-0.77931	0.592537	1.362514	0.74102	0.442238	-1.44137	0.928211
-0.12371	1.918511	1.629781	-0.05302	1.190316	-0.57149	0.65696	1.607803	-0.24328	0.88608	-0.05808	0.445189
-0.52421	-1.28542	3.104259	1.407085	1.320918	-0.08029	0.945944	1.9184	1.638367	1.200962	-0.39029	0.889535
-0.54743	-1.98935	0.401627	0.710954	1.376435	0.203092	0.99058	-1.29727	3.116554	1.331975	-0.06231	1.204775
-0.5126	-1.53842	1.602026	0.539452	1.367201	-0.19365	0.913732	-2.00376	0.407124	1.387667	1.387516	1.335937
-0.5126	-1.68271	1.050398	-0.00988	1.339454	-0.51481	0.885202	-1.55123	1.610542	1.378404	0.696276	1.391692
-0.46616	-1.12094	-0.29224	-0.56274	1.595844	-0.5526	1.001624	-1.69603	1.057526	1.350569	0.525994	1.382419
-0.40812	-1.54525	-0.88204	-1.3419	1.586786	-0.04251	0.957448	-1.13221	-0.28849	1.607766	-0.01948	1.354552
-0.60547	-1.55876	-0.63918	-1.50972	1.523176	0.259769	1.378959	-1.55809	-0.87977	1.598679	-0.56845	1.612041
-0.33266	-0.97304	0.235098	-1.4307	1.677046	0.486476	1.389543	-1.57158	-0.6363	1.534869	-1.34213	1.602944
-0.32105	-1.48579	0.078977	-0.92897	1.730862	0.486476	1.447524	-0.98376	0.240176	1.689224	-1.50877	1.539061
-0.31525	-0.69639	0.630605	1.676335	1.514059	0.52426	1.192592	-1.4984	0.083662	1.743209	-1.43031	1.693591
-0.08888	-1.06156	-0.13612	0.780841	2.039826	0.562045	0.960209	-0.70613	0.636678	1.525724	-0.9321	1.747637
0.085252	0.836056	-0.00082	1.788768	1.952401	0.580937	1.033835	-1.07262	-0.13198	2.053144	1.654866	1.529906
-0.36749	-0.81696	0.599381	2.290027	-2.06037	-0.23143	0.911431	0.831959	0.003666	1.965444	0.765683	2.057925

-0.39071	-1.50975	0.328771	1.074849	-1.84184	0.240876	1.200415	-0.82711	0.605375	-2.05995	1.766515	1.970125
-0.30945	-1.26602	1.397334	2.467753	-1.77888	0.259769	1.21468	-1.52244	0.334084	-1.84073	2.264249	-2.05379
-0.26301	-1.07059	1.57774	2.076673	-1.68064	0.429799	1.347668	-1.27778	1.405335	-1.77757	1.057617	-1.84287
-0.18175	-0.8875	0.859583	0.392223	-1.56562	0.486476	1.3265	-1.08163	1.586196	-1.67902	2.440724	-1.78156
-0.12951	-1.40606	0.894276	-0.37938	-1.65399	0.505368	1.481116	-0.89789	0.866231	-1.56364	2.05239	-1.68415
-0.12371	-0.97798	1.366109	0.06205	-1.51292	0.637614	1.376198	-1.4184	0.901012	-1.65229	0.379796	-1.56354
-0.02503	-0.99644	0.908154	0.15907	-1.30454	0.958782	1.099639	-0.98875	1.374032	-1.51078	-0.38638	-1.64784
-0.01923	-0.96149	1.84835	-0.58202	-1.30454	0.580937	0.945023	-1.00725	0.914925	-1.30174	0.051945	-1.51565
0.085252	-0.28222	1.053867	0.672794	-1.34764	0.920997	0.576431	-0.97218	1.857486	-1.30174	0.148282	-1.30259
0.33484	-0.98317	-0.36163	1.364383	-0.94877	0.864321	0.183449	-0.29043	1.061004	-1.34497	-0.58759	-1.30259
0.578624	1.516037	0.599381	2.460228	-0.94877	0.392014	-0.56248	-0.99395	-0.35806	-0.94486	0.658394	-1.3496
0.183926	-1.06596	0.606319	2.370906	-1.04088	0.089738	-1.06314	1.514428	0.605375	-0.94486	1.345114	-0.94469
0.108469	-1.4367	1.747739	1.12218	-1.3909	-0.02362	-0.94212	-1.07699	0.612331	-1.03725	2.433248	-0.94469
0.154904	-1.2386	1.14754	1.238791	-1.22736	-0.28811	-1.07188	-1.4491	1.756622	-1.38837	2.34456	-1.03598
0.033012	-1.06855	1.185703	2.49655	-1.29594	-0.28811	-0.79164	-1.25026	1.154913	-1.22432	1.104621	-1.38498
0.305818	-1.07138	-0.13612	0.437439	-1.34764	-0.04251	-0.55742	-1.07962	1.193172	-1.29311	1.220406	-1.22079
0.398688	-1.02079	-0.28184	0.342378	-0.83261	0.089738	-0.54085	-1.08246	-0.13198	-1.34497	2.469309	-1.29087
0.160709	-0.70221	0.037344	-0.2796	-1.07456	-0.0614	-0.87769	-1.03165	-0.27806	-0.82833	0.424693	-1.3496
0.508971	-1.00618	-0.97224	-0.80836	-1.1762	-0.36368	-1.26101	-0.71196	0.041925	-1.07104	0.330301	-0.83158
0.508971	-0.48872	0.165711	-0.56177	-1.2616	-0.53371	-1.17496	-1.01701	-0.9702	-1.173	-0.2873	-1.06981
0.636668	-0.34953	1.539577	1.119163	-1.05771	-0.5526	-1.1294	-0.49767	0.170614	-1.25866	-0.81234	-1.17188
0.810799	-1.00995	0.266322	0.950158	-1.1592	-0.83598	-1.29736	-0.35801	1.547937	-1.05413	-0.56748	-1.25764
0.990735	1.882144	0.318363	0.629295	-1.12527	-0.94934	-1.54815	-1.02082	0.271479	-1.15594	1.10162	-1.05288
0.642472	-1.04137	0.661829	0.929544	-1.1592	-0.91155	-1.77409	1.88189	0.32365	-1.12191	0.933804	-1.15481
0.578624	-1.42492	1.331416	0.107717	-0.36358	-0.66595	-1.82333	-1.05231	0.66798	-1.15594	0.615198	-1.12073
0.694711	-0.5005	1.591618	-0.05413	-0.88225	-0.23143	-1.52653	-1.4373	1.339251	-0.35782	0.913335	-1.15481
0.735342	-0.93306	1.227335	0.642506	-1.07456	-0.42035	-1.38617	-0.50951	1.600108	-0.87812	0.09729	-0.35577
0.793386	-0.94547	-0.65653	-0.34798	-0.74216	-0.32589	-1.12572	-0.94363	1.234909	-1.07104	-0.06342	-0.87667
0.75856	-0.42187	-0.7502	-0.61719	-0.92378	0.033061	-1.00838	-0.95612	-0.65369	-0.73759	0.628316	-1.06981
0.764364	-0.78979	-0.84734	-0.34484	-1.1677	-0.08029	-1.19199	-0.43059	-0.7476	-0.91978	-0.3552	-0.73598
0.752755	-0.17115	-0.83347	-1.05292	-0.61997	-0.68484	-1.1883	-0.79987	-0.84499	-1.16447	-0.62252	-0.91838
1.025561	-0.2083	0.186527	-0.91791	-1.36492	-0.28811	-1.1791	-0.17895	-0.83108	-0.61501	-0.35208	-1.16334
0.979126	-0.70095	-0.71551	1.369134	-1.3909	-0.21254	-0.9504	-0.2162	0.191483	-1.36231	-1.05518	-0.61326
1.217105	-0.25685	-0.19163	-0.40221	-0.46704	0.033061	-1.10685	-0.7107	-0.71282	-1.38837	-0.92112	-1.36141
1.600194	0.985374	-0.09449	-0.22811	-1.10834	0.316445	-0.92647	-0.26493	-0.18763	-0.46161	1.349831	-1.3875
1.234518	-1.1244	0.460606	1.14139	-0.77497	0.562045	-0.8372	0.981826	-0.09024	-1.10492	-0.40905	-0.45968
1.106822	-1.45657	0.60285	-0.09641	-0.78319	0.883213	-0.79302	-1.13567	0.466251	-0.7705	-0.23617	-1.10373
1.118431	-0.16526	0.987949	0.585414	-0.89885	0.788752	-0.97065	-1.46902	0.608853	-0.77875	1.12369	-0.76893
1.176475	-0.9358	-0.37204	-1.09373	-1.07456	0.335338	-0.91128	-0.17303	0.994921	-0.89477	-0.1054	-0.77718

1.193888	-0.09818	0.543871	-0.24562	-0.30825	0.35423	-1.02448	-0.94636	-0.36849	-1.07104	0.571626	-0.89334
1.26354	0.101329	-0.17082	-0.97076	-0.81611	0.902105	-1.26469	-0.10569	0.549726	-0.30232	-1.0957	-1.06981
0.880451	-0.55595	0.065099	-0.67633	-0.75855	0.618721	-1.09167	0.094506	-0.16676	-0.81178	-0.25356	-0.30021
1.269345	0.345217	-0.81612	-1.23349	-0.22968	1.091028	-1.03921	-0.56512	0.069749	-0.75404	-0.97359	-0.81025
1.217105	-0.07124	0.682645	-1.03402	-0.45106	1.468873	-0.97249	0.339303	-0.81368	-0.2235	-0.68124	-0.75244
1.292562	-0.11758	-0.15	-0.72283	-0.02019	1.072136	-0.88966	-0.07868	0.688849	-0.44558	-1.23448	-0.2213
1.536346	0.534908	-0.51355	0.879595	0.223208	0.977674	-0.64577	-0.12519	-0.14589	-0.01335	-1.03641	-0.44363
1.582781	1.227063	-0.71551	-0.15999	-0.0159	1.185489	-0.57674	0.529716	-0.51035	0.230814	-0.72741	-0.01092
1.17067	-0.42077	0.561218	1.035467	0.262252	1.147705	-0.31905	1.224411	-0.71282	-0.00905	0.863737	0.233524
1.054583	-1.20428	0.967133	0.005651	0.170286	1.468873	-0.38854	-0.42944	0.567116	0.26998	-0.16853	-0.00661
1.124235	-0.39658	1.289783	0.265643	0.056866	1.185489	-0.34988	-1.21583	0.974052	0.177725	1.018512	0.272736
1.141648	-1.41887	0.427647	-0.22561	0.353483	0.807644	-0.20815	-0.4052	1.297514	0.063948	-0.00406	0.180375
1.106822	-0.55344	-0.05633	-0.2968	0.393798	1.072136	-0.03927	-1.43122	0.43321	0.361498	0.254105	0.06647
1.275149	-0.81775	-0.46051	-0.79398	0.333273	1.355519	-0.15477	-0.56265	-0.05198	0.401939	-0.23369	0.364358
1.048778	-0.94814	-0.39112	-0.51059	0.553644	0.958782	0.014109	-0.82788	-0.45718	0.341224	-0.30438	0.404844
1.280953	-0.39909	-0.82479	-1.1432	0.323154	1.563334	-0.12394	-0.9588	-0.38762	0.562289	-0.79806	0.34406
1.106822	-0.88608	0.434586	-0.97597	0.333273	1.27995	-0.02316	-0.40773	-0.82238	0.331074	-0.51666	0.565376
1.0778	-0.4346	-0.43275	0.323151	0.886061	0.883213	0.007666	-0.89652	0.440166	0.341224	-1.14483	0.333898
1.40865	-0.20163	-0.35259	0.238701	0.934139	1.185489	-0.58779	-0.44333	-0.42936	0.89575	-0.97877	0.34406
1.553759	0.814299	-0.405	-0.19404	0.523843	0.203092	-1.00792	-0.20951	-0.34899	0.94398	0.311209	0.899217
1.217105	-0.62586	0.510912	1.088424	0.808712	0.675398	-0.87677	0.810084	-0.40153	0.532394	0.227344	0.947501
1.228714	-1.42947	0.784991	-0.04537	0.632736	0.221984	-0.7194	-0.63533	0.516684	0.818159	-0.20235	0.535447
1.292562	-0.67746	1.138866	0.425528	0.393798	0.429799	-0.57582	-1.44189	0.791453	0.641629	1.071101	0.821537
1.304171	-0.38857	0.027804	-0.65967	1.171536	-0.04251	-0.31583	-0.6871	1.146217	0.401939	-0.05473	0.644807
1.478302	0.236272	0.243771	-0.27121	0.701494	-0.17475	-0.44698	-0.39712	0.03236	1.182123	0.412866	0.404844
1.478302	0.236194	-0.31566	-0.88237	0.563561	-0.25032	-0.63841	0.229986	0.248871	0.710603	-0.6647	1.185915
1.269345	1.073975	-0.16301	-0.59346	1.246471	-0.34478	-0.61678	0.22992	-0.31197	0.572237	-0.27897	0.713859
1.402845	0.88287	-0.82045	-1.18834	0.77957	-0.74152	-0.72308	1.070739	-0.15893	1.257294	-0.88583	0.575335
1.246127	-0.24059	0.558616	-1.005	0.895692	-0.98712	-0.68258	0.878924	-0.81803	0.788925	-0.59895	1.261171
1.211301	0.75319	-0.29138	-0.19983	0.808712	-0.89266	-0.76127	-0.24865	0.564507	0.905413	-1.18965	0.79227
1.286758	0.209645	-0.43307	0.559148	0.403854	-0.93044	-0.70053	0.748778	-0.28762	0.818159	-1.00759	0.90889
1.437672	1.573298	-0.56025	-0.17702	0.77957	-1.10047	-0.56984	0.203261	-0.42967	0.412027	-0.2081	0.821537
1.135844	0.079728	0.536065	1.06195	0.740599	-0.40146	-0.60896	1.571875	-0.55718	0.788925	0.545541	0.414944
1.071996	-1.01663	0.876062	-0.01987	0.94373	-0.5526	-0.60896	0.072842	0.5419	0.749831	-0.18544	0.79227

Appendix C

Regional Index Residuals

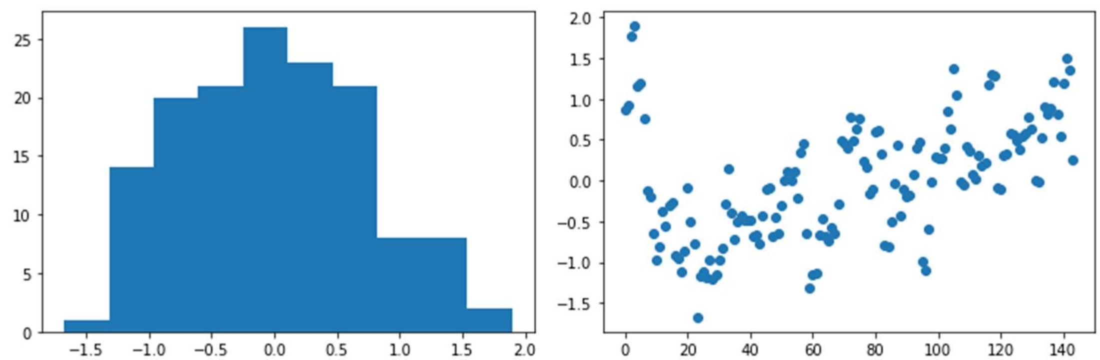

Figure C1. Histogram and scatterplot of residuals from the regional OLS regression model

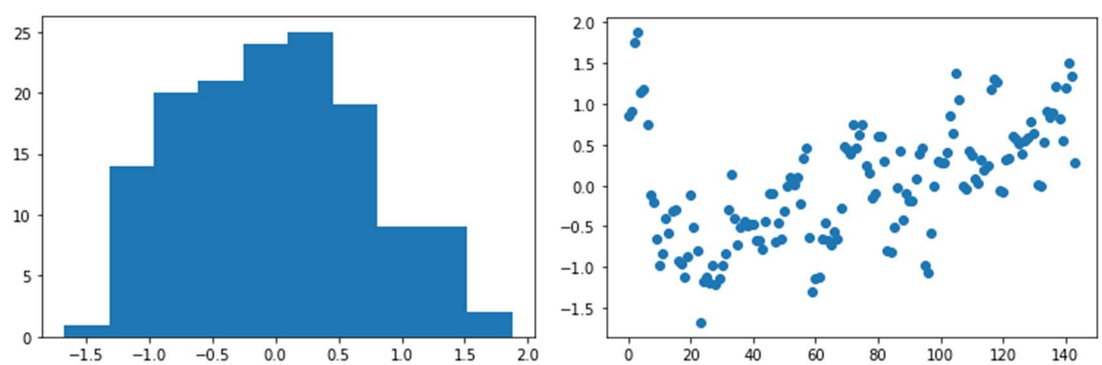

Figure C2. Histogram and scatterplot of residuals from the regional coordinate descent Lasso Regression model

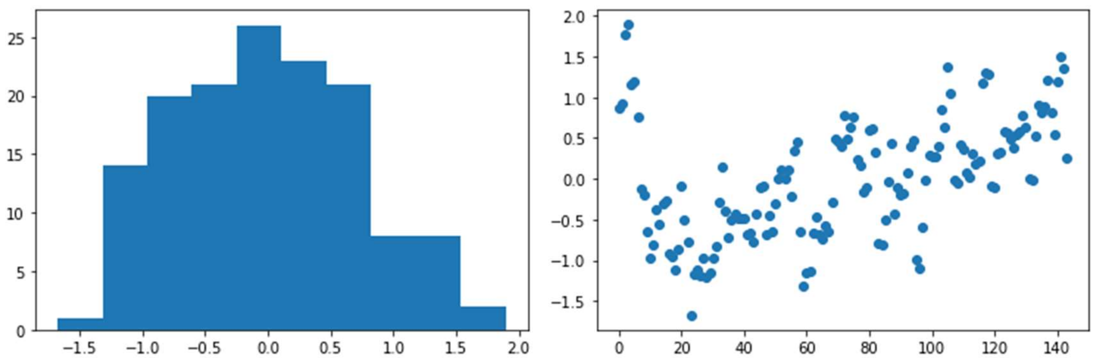

Figure C3. Histogram and scatterplot of residuals from the regional Lars Lasso Regression model

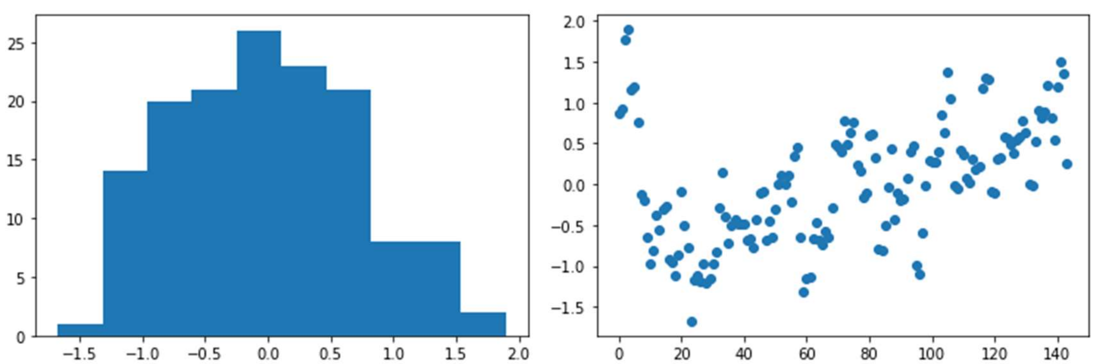

Figure C4. Histogram and scatterplot of residuals from the regional Ridge Regression model

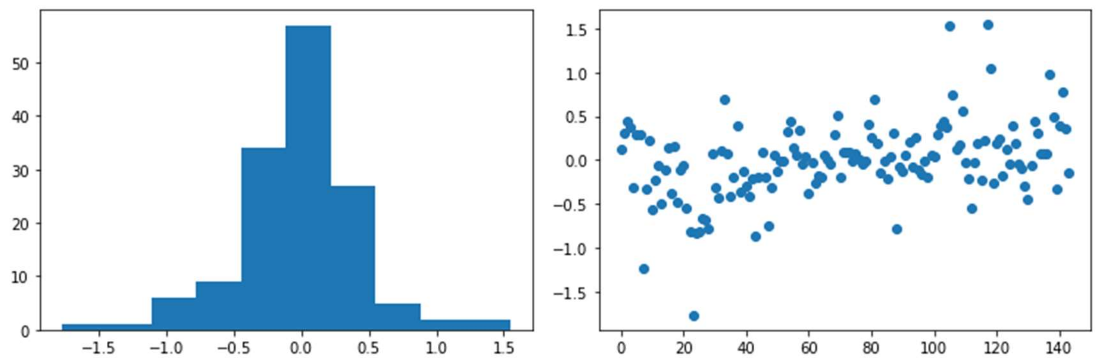

Figure C5. Histogram and scatterplot of residuals from regional Gaussian Regression model using C * Exp kernel

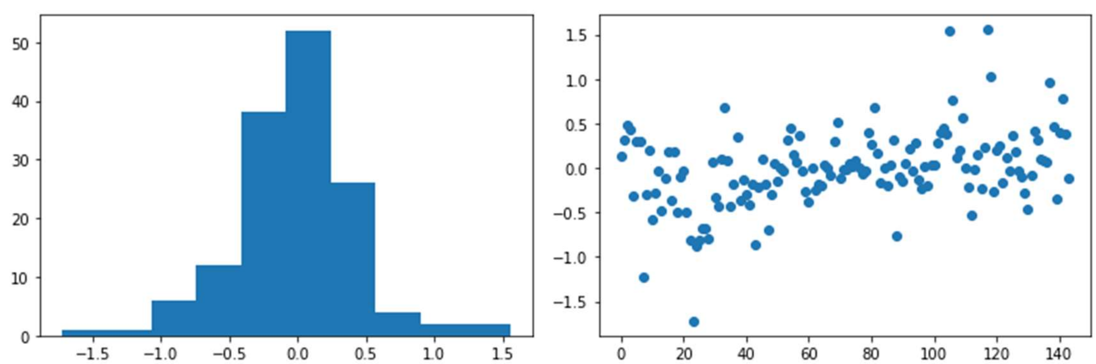

Figure C6. Histogram and scatterplot of residuals from regional Gaussian Regression model using C * RQ kernel

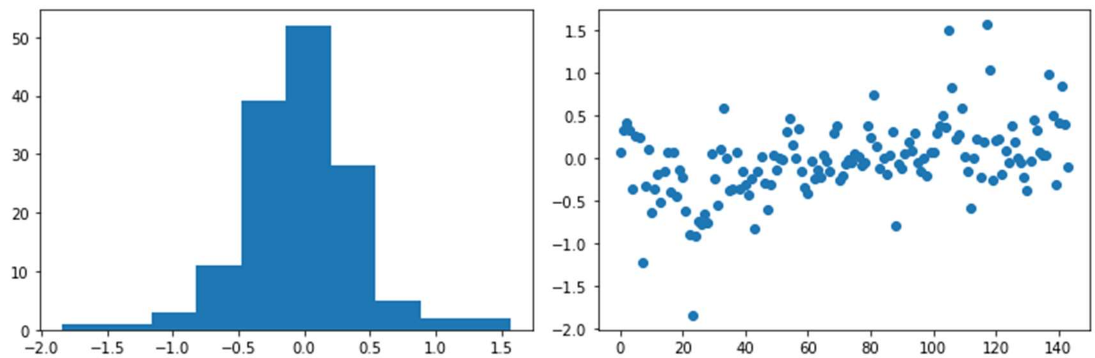

Figure C7. Histogram and scatterplot of residuals from regional Gaussian Regression model using C * Exp * RQ kernel

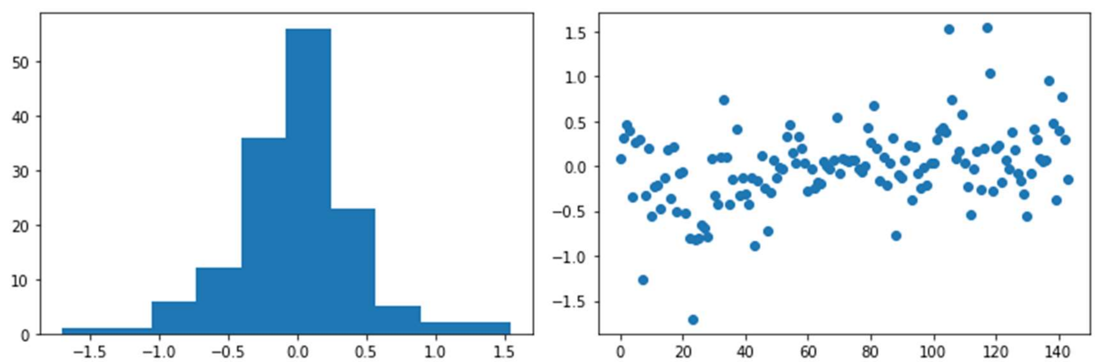

Figure C8. Histogram and scatterplot of residuals from regional Gaussian Regression model using (C * RBF *RQ) + Exp kernel

Appendix D

National Index Residuals

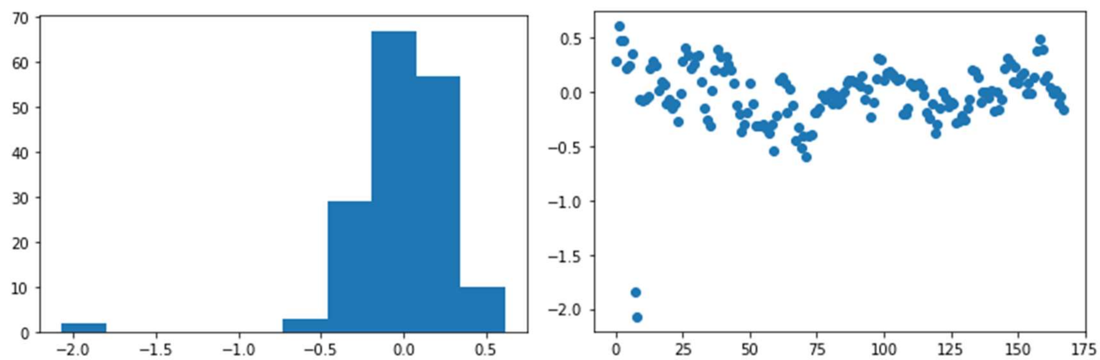

Figure D1. Histogram and scatterplot of residuals from the regional OLS regression model

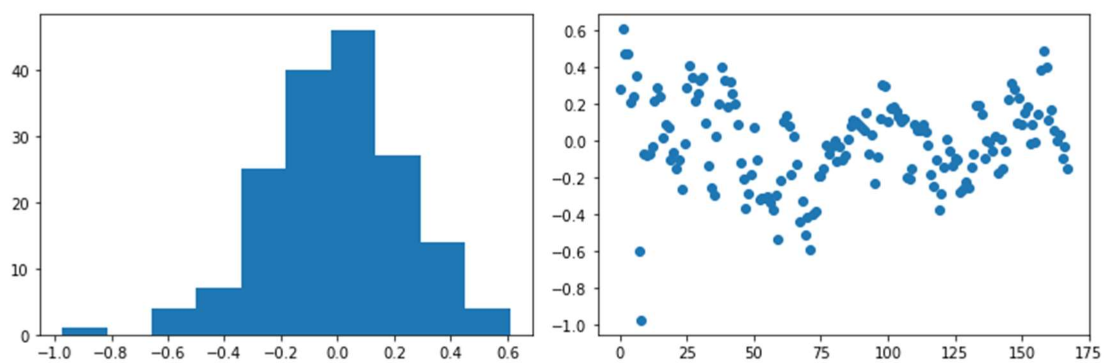

Figure D2. Histogram and scatterplot of residuals from the regional coordinate descent Lasso Regression model

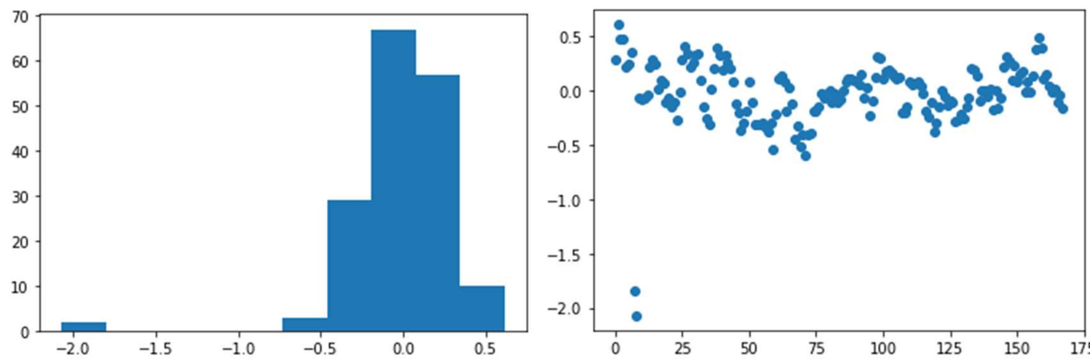

Figure D3. Histogram and scatterplot of residuals from the regional Lars Lasso Regression model

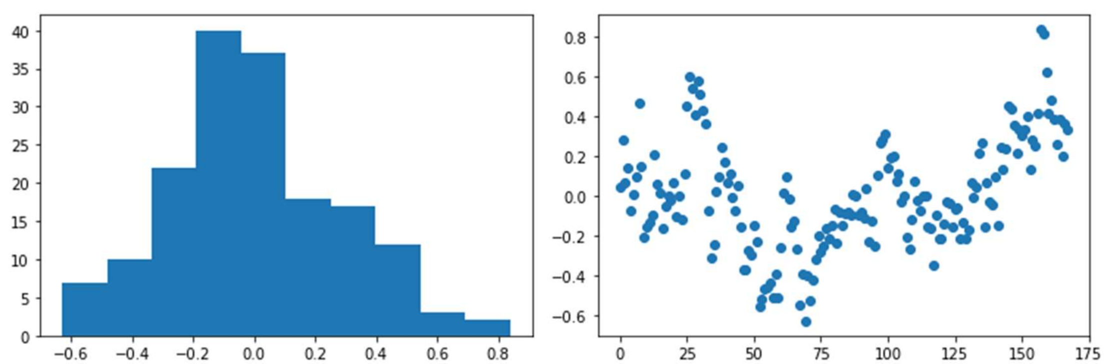

Figure D4. Histogram and scatterplot of residuals from the regional Ridge Regression model

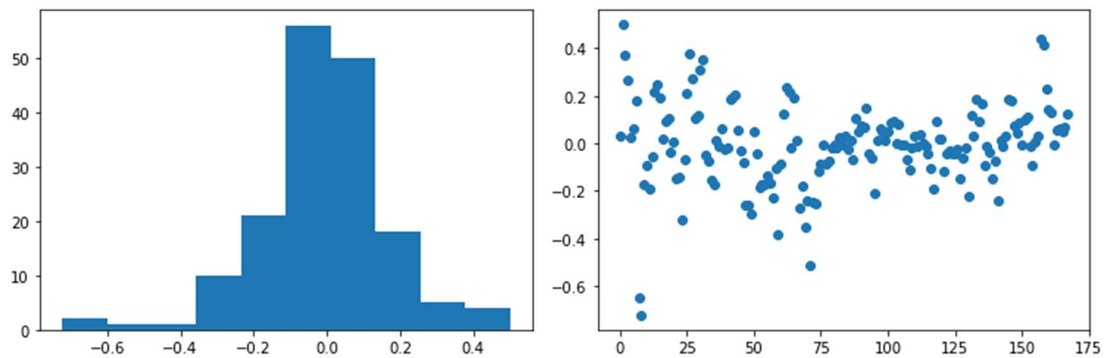

Figure D5. Histogram and scatterplot of residuals from regional Gaussian Regression model using C * Exp kernel

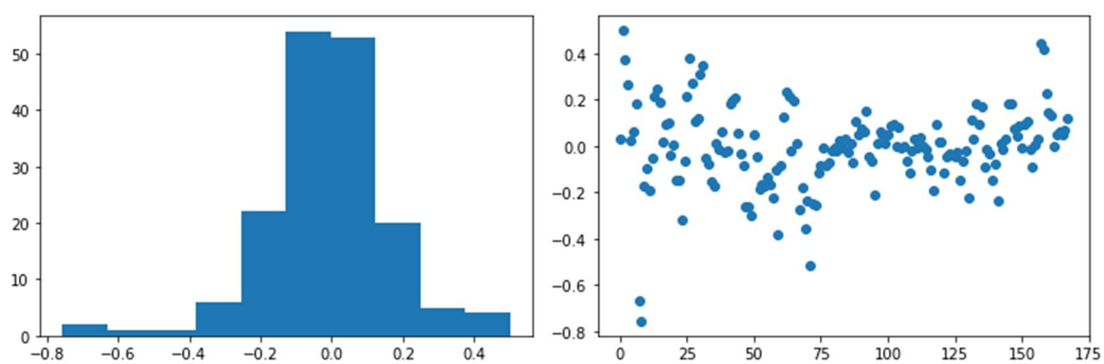

Figure D6. Histogram and scatterplot of residuals from regional Gaussian Regression model using C * RQ kernel

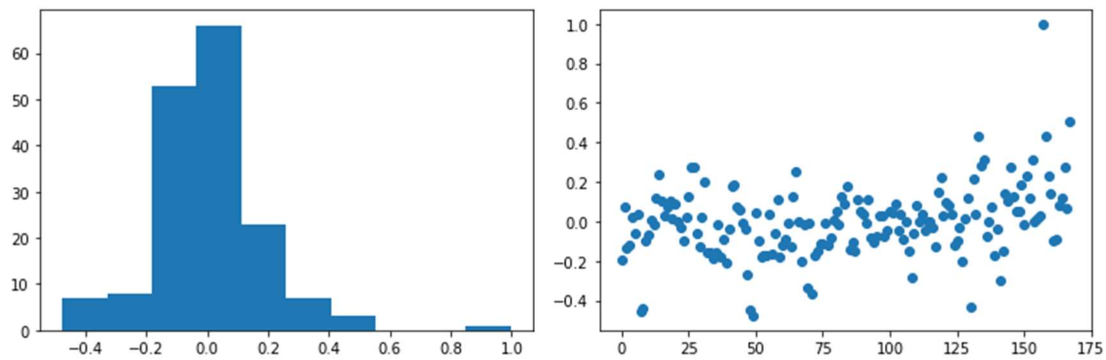

Figure D7. Histogram and scatterplot of residuals from regional Gaussian Regression model using C * Exp * RQ kernel

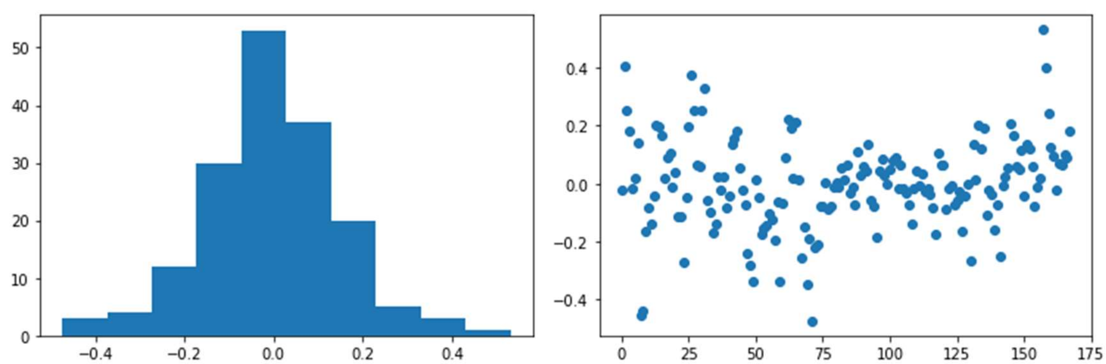

Figure D8. Histogram and scatterplot of residuals from regional Gaussian Regression model using (C * RBF *RQ) + Exp kernel

Appendix E

Root Mean Square Error for National Index Models

RMSE	Rand State										
	0	1	2	3	4	5	10	25	50	100	250
OLS	0.62493	0.22454	0.24497	0.34933	0.21443	0.23559	0.25299	0.79416	0.22958	0.26173	0.64852
Lasso (CD)	0.39667	0.22346	0.24366	0.25867	0.21376	0.23440	0.25319	0.55953	0.23025	0.21975	0.44492
Lasso (LARS)	0.62493	0.22454	0.24497	0.34933	0.21443	0.23559	0.25299	0.79416	0.22958	0.26173	0.64852
Ridge	0.31068	0.29555	0.33408	0.28078	0.25411	0.32932	0.30561	0.30970	0.35770	0.33484	0.31526
Exp	0.22242	0.15743	0.19919	0.21204	0.16393	0.17280	0.19031	0.30078	0.15972	0.20417	0.27969
RQ	0.22215	0.15742	0.19935	0.21268	0.16387	0.17280	0.19031	0.29566	0.15970	0.20504	0.27774
Exp*RQ	0.22908	0.14337	0.19526	0.16396	0.16312	0.19854	0.20844	0.21875	0.21510	0.20945	0.21482
RBF*RQ+Exp	0.18777	0.13628	0.16698	0.16366	0.14670	0.15778	0.17095	0.18183	0.15649	0.17179	0.18821
	Normalized										
OLS	0.16920	0.06079	0.06633	0.09458	0.05806	0.06379	0.06850	0.21502	0.06216	0.07086	0.17559
Lasso (CD)	0.10740	0.06050	0.06597	0.07004	0.05787	0.06346	0.06855	0.15150	0.06234	0.05950	0.12046
Lasso (LARS)	0.16920	0.06079	0.06633	0.09458	0.05806	0.06379	0.06850	0.21502	0.06216	0.07086	0.17559
Ridge	0.08412	0.08002	0.09045	0.07602	0.06880	0.08916	0.08274	0.08385	0.09685	0.09066	0.08536
Exp	0.06022	0.04263	0.05393	0.05741	0.04438	0.04679	0.05153	0.08144	0.04324	0.05528	0.07573
RQ	0.06015	0.04262	0.05397	0.05758	0.04437	0.04678	0.05153	0.08005	0.04324	0.05552	0.07520
Exp*RQ	0.06202	0.03882	0.05287	0.04439	0.04416	0.05376	0.05644	0.05923	0.05824	0.05671	0.05816
RBF*RQ+Exp	0.05084	0.03690	0.04521	0.04431	0.03972	0.04272	0.04629	0.04923	0.04237	0.04651	0.05096

Appendix F

Root Mean Square Error for Regional Index Models

RMSE	Rand State										
	0	1	2	3	4	5	10	25	50	100	250
OLS	0.764589	0.700254	0.837318	0.729522	0.815971	0.709397	0.879523	0.72892	0.74145	0.754428	0.76747
Lasso (CD)	0.763597	0.700439	0.835766	0.730531	0.820843	0.708297	0.876716	0.727242	0.74218	0.756667	0.768945
Lasso (LARS)	0.764574	0.700248	0.837292	0.729536	0.816042	0.709379	0.879457	0.728887	0.741459	0.75446	0.767492
Ridge	0.764588	0.700251	0.83729	0.729511	0.815985	0.709393	0.87951	0.728915	0.741449	0.754439	0.767485
Exp	0.481091	0.373393	0.579876	0.560302	0.509315	0.497595	0.471464	0.579049	0.459629	0.527321	0.506235
RQ	0.481091	0.371243	0.582759	0.543342	0.50908	0.494354	0.480691	0.579048	0.451212	0.527322	0.50425
Exp*RQ	0.477515	0.389338	0.586919	0.550488	0.527237	0.517747	0.508833	0.598568	0.459257	0.549667	0.506893
RBF*RQ+Exp	0.483588	0.378153	0.583155	0.559956	0.504702	0.49512	0.471284	0.577279	0.455251	0.535437	0.504065
	Normalized										
OLS	0.221017	0.20242	0.242041	0.21088	0.23587	0.205063	0.254241	0.210706	0.214329	0.21808	0.22185
Lasso (CD)	0.220731	0.202474	0.241592	0.211172	0.237278	0.204745	0.253429	0.210222	0.21454	0.218727	0.222276
Lasso (LARS)	0.221013	0.202418	0.242033	0.210885	0.235891	0.205058	0.254222	0.210697	0.214331	0.218089	0.221856
Ridge	0.221017	0.202419	0.242033	0.210877	0.235874	0.205062	0.254237	0.210705	0.214328	0.218083	0.221854
Exp	0.139067	0.107935	0.167623	0.161965	0.147226	0.143838	0.136285	0.167384	0.132864	0.152431	0.146336
RQ	0.139067	0.107314	0.168456	0.157062	0.147158	0.142901	0.138952	0.167384	0.13043	0.152431	0.145762
Exp*RQ	0.138034	0.112545	0.169659	0.159128	0.152407	0.149663	0.147087	0.173026	0.132756	0.15889	0.146526
RBF*RQ+Exp	0.139789	0.109311	0.168571	0.161865	0.145893	0.143123	0.136232	0.166872	0.131598	0.154777	0.145708