Weigh-In-Motion Stations

Benjamin Zevin

STAT 515

I. Project Description

This final project will showcase the skills that I've developed while taking this class. I will be exploring a dataset made by the New York State Department of Transportation that contains traffic patterns of significant roads and highways in New York. I will be attempting to answer two research questions with the skills I've developed. One of those questions requires logistic analysis, and the methods I will be using to answer this question are Logistic Regression and a Classification Tree. The second question involves regression analysis, and the techniques I will be using to answer this question are Linear Regression and a Regression Tree. I will be using a variety of different visualizations and outputs to represent my process in answering these questions and displaying my results.

II. Data set

The data set I have decided to use for this project is the Weigh-In-Motion Station Vehicle Traffic Counts: 2013[1], and I found this dataset on the data.gov website, but it can also be found on the data.ny.gov website. The dataset contains 21 variables, thirteen of which are counts of the different classes of vehicles that pass through the Station in a day (the thirteen classes of vehicles are shown in an image in Appendix A). It includes counts for each direction traveled on the highway along with the latitude and longitude of each Station's location. The dataset contains counts for the year 2013. Some variables are characteristics of the WIM stations, and another variable is the collection date. There are 7909 observations in this data set, and it appears that the data is complete (there is no missing data in the observations). There are two limitations set on the dataset, though; the first is that data is only collected on weekdays, so there is no data collection on Saturdays and Sundays. The second one is that the collection time period is only during certain months for each Station. That time period and duration can be different from Station to Station. Other than these limitations, the data set is already pretty clean, but a few things need to be changed to answer the research questions I've established.

In preparation for the data analysis, I need to inspect and make some changes to the data set to make it more appropriate for research. I did all analysis and preparation for this project in R[3]. The first thing I did once I loaded in the data set was turn all categorical variables into factored variables so they would be appropriate for all analysis methods. The five variables I changed to factored variables were the number of lanes, the direction of the traffic, the route the Station is on, the station number, and the months that the Station is active. After completing that task, I thought it wasn't essential to have the collection date for each observation because I wasn't doing any time series analysis. Instead of removing the collection date altogether, I thought it might be helpful to know the month of collection. I created a new variable, took the month from the collection date variable, and made a new column. I then turned this newly constructed collection date month variable into a factored variable.

I looked over the data set and discovered that some of the variables in the dataset were unnecessary for the analysis I had planned to do. Variables like collection date were now unnecessary because I had taken the information I wanted to use for that variable and created a new variable. Other variables that I decided to remove were Latitude, Longitude, and the combination of both latitude and longitude because the Weigh-In-Motion stations do not move from their location, so I felt it was unnecessary to keep these variables in the data set. Instead, I created a new variable, a sum of every vehicle that passed through the Station on the collection date. I could use this variable to find the percentages of each vehicle class that passed through the Station on any given day instead of the induvial counts.

When looking over the dataset in excel, I discovered that two entries didn't have any cars pass through the Station that day. I figured this must have been some error, or the highway was closed down, so I decided to remove them because I didn't think it was inappropriate for inaccurate data to alter the results. The final thing I had to do was create a new response variable for my first research question. I wanted to create a model that could predict the direction traffic traveled based on the other variables in the data set. Still, to do this, I had to create a new variable based on the direction variable that was binary. If the road were a north or south road, the value for the response variable would be yes, and if the direction of the road were an east or west road, the value of the response variable would be no. Once I created this variable, I also made it a factored variable. Now that this is completed, I could conduct my exploratory analysis and attempt to answer my research questions.

III. Research Questions and Exploratory Analysis

I developed two research questions that I wanted to answer using this dataset. The first one is can you create a model that can accurately predict the direction of the highway or road using the variables available other than the ones that have a direct relationship to the direction of the road like route number and station number. The response variable for this question is binary. If the direction of the road is north or south, the value will be yes, and if the road is east or west, it would be no. Since the response variable is binary, linear regression and a regression tree wouldn't be appropriate. To answer this question, I will be using Logistic Regression and a Classification Tree. The second research question is can you make a model that predicts the number of class 2 vehicles (passenger vehicles) that pass through a station on any given day using the variables in the dataset. Since the response variable for this question is a count and isn't a factor variable, linear regression can answer this question, and I will be using Linear

Regression and Regression Tree. Before we attempt to answer the two questions, I must do some exploratory analysis of the data set.

The first form of exploratory analysis I did was create a summary for all of the viable variables in this data set. The output from R for the summary is located in Appendix B. When you look at the summary of the variables, you notice some interesting things. First off, most of the data entries belong to either a two-lane road or a four-lane road. This shows that the streets that either five, six, or eight lanes didn't have as many stations as active as the other stations.

Another interesting thing I noticed is that class 2 vehicles are the most used vehicles. The average count for class 2 vehicles was 8959. The next closest average is class 3, with 1591. The rest of the classes have relatively low counts compared to these two classes. Finally, the binary variable for question 2 has a pretty even split, with 56% of the observations being north or south and 44% of the results being east or west. This shows that our response variable is suitable for analysis because it isn't unbalanced.

The subsequent exploratory analysis I did was create a correlation plot of the count variables to see if they had high correlation or multicollinearity. If some of the variables have multicollinearity, they could be removed for the logistic regression and classification tree because of redundancy. The correlation plot is located in Appendix C. When looking at the correlation plot, you can see some significant correlation between classes 2, 3, 4, 5, and 6. The correlation between class 2 and the other classes stated previously shows a high possibility that these classes could develop an excellent model when predicting the number of class 2 vehicles. There is also some correlation between classes 9, 11, and 12. Some of these variables will be removed when the analysis comes for the first research question to avoid redundancy.

Another form of exploratory analysis I did was a box plot for the total number of cars on the road for each month located in Appendix D. There is no importance to the colors; it shows you that each box plot is of a different month. The averages and quartiles are similar for each month, but the summer months and holiday months at the end of the year have more days with outliers. I assume this is because people like to go on vacation in the summer and travel for the holidays. These two events would require more vehicles to be on the road.

The final form of exploratory analysis was a grouped bar plot of the total number of vehicles seen in each route grouped by each direction in Appendix E. The one thing I noticed is that each route had equal numbers for each of their perspective direction. For example, if one road travels east and west, the number of cars that traveled east is almost the same as the number of people that traveled west. I assume this is because people are creatures of habit, and they will use the same road they used to get one place to return from that location. The two most popular routes in New York are I-495, with around 85,000 total vehicles traveling in each direction during the collection period. The second most is NY590, with about 55,000 vehicles traveling in each direction.

IV. Data Analysis

A. Can you create a model that can accurately predict the direction of the highway or road?

The first method I used to answer this question is logistic regression since the response variable is binary. Before I ran the glm[5] on the data, I removed the multicollinear variables and variables that directly relate to the direction of the highway. The variables I removed were Class 3, Class 5, Class 11, Route, and Station. My initial model uses all of the remaining variables on the training set of data. The summary for the model is located in Appendix F, along with the pseudo-R-squared value and the predictive accuracy from using the test set. After reading the summary for the initial model, I noticed that not many of the month factors were statistically significant. The pseudo-R-squared value for this initial model is only 0.17, but the model's accuracy was 0.707 using the test set. So I decided to remove the month variable and remake the model to see if I could receive similar results. I did this because it is always better to have a model with fewer predictors is if you receive similar results. The summary of the next model is located in Appendix G, along with its pseudo-R-squared value and predictive accuracy. When I reviewed the summary, I noticed that two of the lane factors weren't significant, so I decided to remove the lanes variable to see if I could get similar results again. The pseudo-R-squared value for the third model is only 0.16, but the model's accuracy was 0.703 using the test set. With the accuracy of this model only decreasing by 0.004, it is clear that this model is better than the initial model because you know 12 fewer predictors with the same accuracy.

I removed the lanes variable for the third model, and the summary, pseudo-R-squared, and the test set accuracy are located in Appendix H. Nearly all of the variables now are significant for this model except for one, and that variable is class 2 count. The pseudo-R-squared value for this second model is only 0.10, but the model's accuracy was 0.654 using the test set. This is a decent-sized decrease, so you might want to stick with the second model because its accuracy is 5% higher. Still, if you're going to have a model with as few predictors as possible and are willing to sacrifice this accuracy, this might be a better model. I will create one last model with all of the current variables in this model except for class 2 count to see if the model's accuracy stays the same without class 2 count. This final logistic regression model has a pseudo-R-squared of 0.10, and the model's accuracy was 0.654 using the test set and is located in Appendix I. These are the same exact values as the third model, so if you want a model that sacrifices accuracy to lose predictors, this model is better than model 2. Still, if you want a model with high precision, then model 2 is the better model. Even though these four models provide pretty high accuracy, I wanted to see if a Classification Tree could provide high accuracy.

When creating the classification tree, I used all of the same variables for the logistic regression. The results from the unpruned tree are in Appendix J. Using the graph and summary of the initial tree, and I decided to prune the tree two different ways, once when the relative error is below 0.1 and once when the x error is less the 0.1. The image of the relative error, along with the summary results and test accuracy, is located in Appendix K, and the image, summary, and test set accuracy for x error tree is located in Appendix L. In order to get the relative error tree, I pruned the initial tree with a cp value of 0.0047. The resulting tree has 15 splits and a relative error below 0.1. The root node error is 0.43282, and the accuracy of the test set is 0.951, which is much higher than the accuracy of any logistic regression model. In order to get the x error tree, I pruned the initial tree with a cp value of 0.0022. The resulting tree has 24 splits and a relative error below 0.1. The root node error is the same as the relative error tree, and the accuracy of the test set is 0.959. Since the two trees have almost the same precision, the relative error tree has nine fewer splits, and the accuracy is only lower than the x error tree by .9%. Therefore, I believe that the relative error tree is the best model for this data.

B. Can you make a model that predicts the number of class 2 vehicles (passenger vehicles) that pass through a station on any given day?

The first method I used to answer this question was logistic regression since class 2 count is the response variable and it's a continuous variable. Before I did that, I had to remove the response variable for research question one. In Appendix M, my initial model used all of the remaining variables and returned an Adjusted R-squared value of 0.9957, which clearly shows that the model is overfit. The variables Route, Direction, and the number of lanes returned a value of NA which means that these variables weren't used. There is a one-to-one relationship between the Station and those three variables. I removed these three variables and collection month because this variable had one significant p-value out of twelve. I created another model, located in Appendix N and the Adjusted R-squared value is 0.9956, so this model is clearly overfit again. There are still too many predictors for this model to be usable, but the other remaining variables are statistically significant.

The next course of action is to use the exhaustive form of Regsubsets[4] for best subset selection. The list of Adjusted R-squared values is located in Appendix O. From this list, you can see that with one predictor, the Adjusted R-squared value is already .965. Because of how high the R-squared value is, I decided to make three models, the one-predictor model, the two-predictor model, and the five-predictor model. The summary and test set R-squared value for the one-predictor model are located in Appendix P. The predictor used in this model is class 3 count. The Adjusted R-squared value is 0.9651, which means it is an excellent model for one predictor. The R-squared value for the test set with this model is 0.968. The next model is the two-predictor model located in Appendix Q, and the predictors for this model are class 3 count and class 6 count. The adjusted R-squared value for this model is 0.9742, which is .9% better, but it does require one more predictor. The R-squared value for the test set with this model is 0.978, which is 1% better than the one predictor model's results. The five-predictor model is located in Appendix R, and the predictors for this model are class 3, class 9, station 580, station 4342, and station 8280. The Adjusted R-squared value for this model and the R-squared value for the test set with this model is 0.993.

The five-predictor model performs the best with the training set and test set but only marginally, and it requires four more predictors than the one-predictor set. This is why I believe that the one-predicter model is the best model for predicting the number of class 2 vehicles. In order for this to be valid, it must pass all of the assumptions. To see if this model passes all of the assumptions, I will be using the diagplots123[7] and diagplot45[7] methods given to us for a previous homework assignment. The resulting plots are located in Appendix S. When looking at the Residuals Vs. Leverage graph, it is clear that the chosen model doesn't pass the normality assumptions because a couple of hundred points are over the 3p/n line, which means that the data in this model isn't normal. I looked at the residuals of the other models, and none of them passed the normality assumption. Because of this, I attempt to answer the second research question with a Regression Tree instead.

When creating the Regression Tree, I used the same variables I used for Regsubsets[4], but I added back in the collection date month variable because this variable could now be significant with the new method. I had the same process for creating this tree as the Classification tree. I created an initial tree with all and then pruned it down to make two trees, one where the relative error is less than 0.1 and one where the x error is less than one. Looking through the results of the unpruned tree located in Appendix T, the required cp to achieve a tree with 0.1 relative error is 0.001, and the cp needed for a tree to achieve 0.1 x error is 0.0007. The results of the relative error pruned tree are located in Appendix U, and the results for the x error pruned tree are located in Appendix V. The square root mean square error of the relative error tree is 1443 with nine splits while the square root mean square error is 1379 with ten splits. Both trees used the same predictors: class 3 counts, class 4 counts, and Station. When you look at the graphs or their fitted values vs. actual values located in Appendix W, they appear to be very similar. The only difference is that one of the sets of values is split in two. It all depends on if you want the extra accuracy for one extra split.

V. Conclusion / Challenges / Further Analysis

1. *Can you create a model that can accurately predict the direction of the highway or road?*

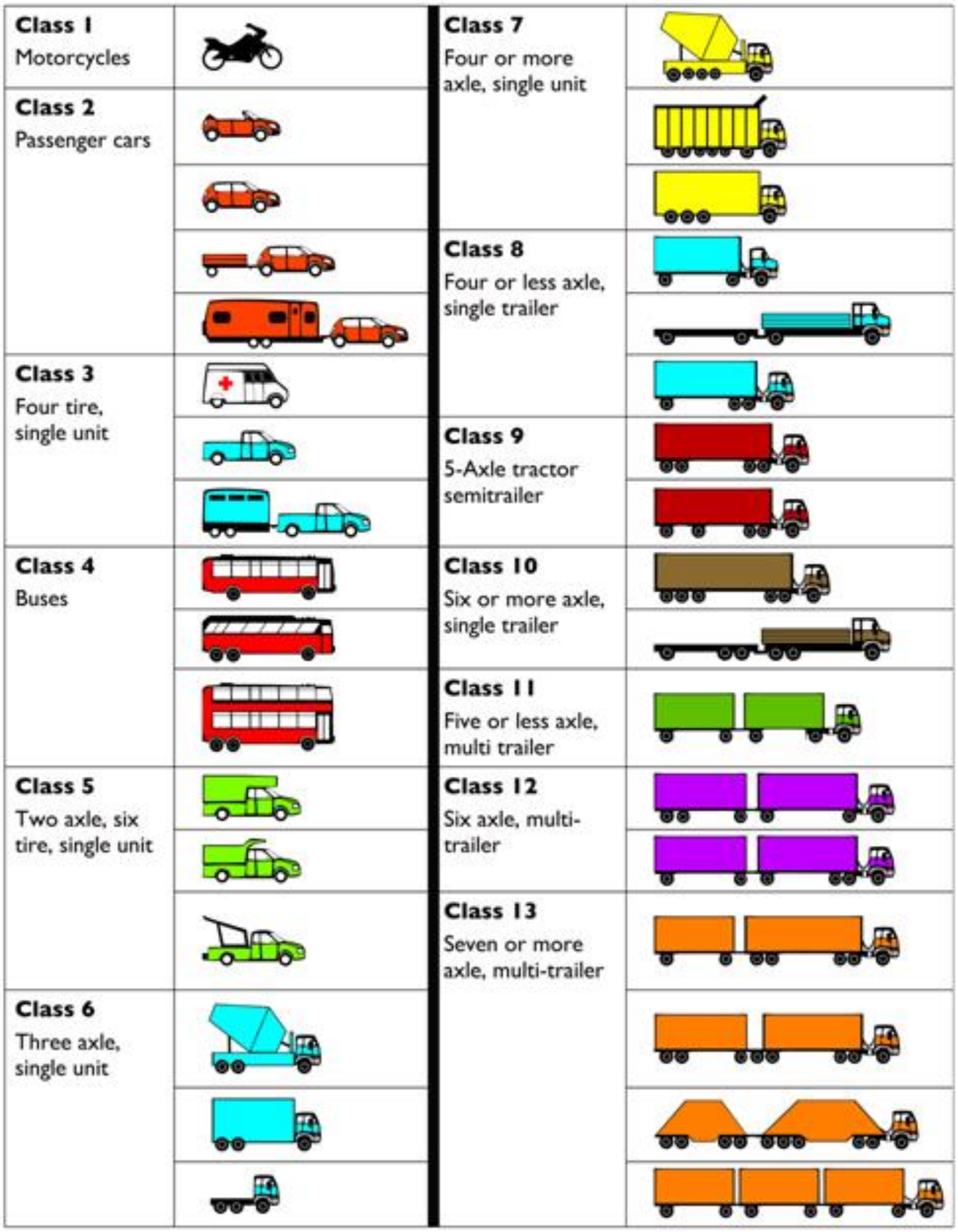
When creating a model that can predict the direction of the highway, I used two different methods to complete this task. The first was logistic regression, and the second was a classification tree. The best model I achieved was a model with 0.703 accuracy and 13 predictors. I could achieve higher accuracy with a classification tree, and I was correct. The best model for a classification tree had an accuracy of 0.95, and it had 15 splits using eight predictors. A challenge I came across when completing this research question was deciding what variables I needed to remove due to multicollinearity. It was clear that there was multicollinearity in a series of variables, but I had to determine which variables to remove. I remembered that when you have a high correlation, either one can be removed and have the same results. If I were to do future analysis on this research question, I would attempt to answer this question using a random forest to see if it could achieve higher accuracy than the classification tree.

2. *Can you make a model that predicts the number of class 2 vehicles (passenger vehicles) that pass through a station on any given day?*

When creating a model that can predict the number of class two cars that pass through a station on any given day, I used two different methods when attempting to answer this question. The first method was using linear regression. Even though the models created using linear regression have extremely high Adjusted R-squared scores, the models didn't pass the assumptions needed to be a linear model. None of the models passed the normality assumption. Because of this, I moved to a regression tree, and the regression tree created better results. When you compare the two trees created to answer this question, both are very similar, but I believe the best one to be is the tree based on X error because it only had one more split and provided better accuracy. One challenging thing that occurred when trying to answer this question was determining if it was worth removing all the outliers in the data to make it normal. I decided that it wasn't appropriate because there aren't just a few outliers and these observations weren't errors; they are a part of the data set for a reason. If I were to do further analysis with this question, I would do a clustering analysis and associate the longitude and latitude variables into the data so I can see where class two cars travel to the most and which roads and areas are less frequently traveled.

Overall, I thought this project was an excellent task to display the skills we have learned and developed in this class. It gave me the ability to see a problem all the way through with just my thought without any guidance from a professor, and I believed I answered these two research questions very well. I think my answers to these questions can help people understand the traffic patterns of the major roads in New York.

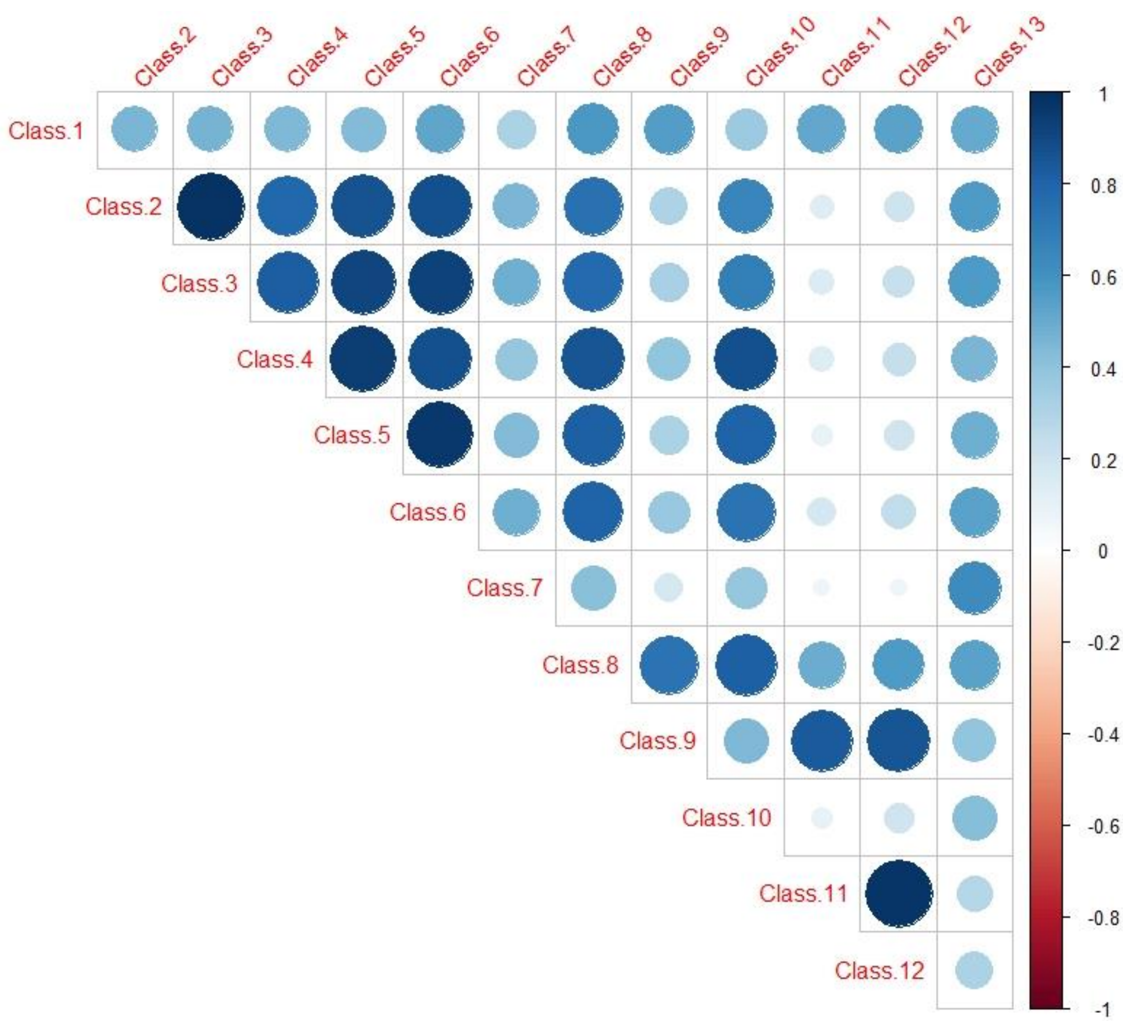
Appendix A[2]


Class 1
Motorcycles
Class 2
Passenger cars
Class 3
Four tire, single unit
Class 4
Buses
Class 5
Two axle, six tire, single unit
Class 6
Three axle, single unit
Class 7
Four or more axle, single unit
Class 8
Four or less axle, single trailer
Class 9
5-Axle tractor semitrailer
Class 10
Six or more axle, single trailer
Class 11
Five or less axle, multi trailer
Class 12
Six axle, multi-trailer
Class 13
Seven or more axle, multi-trailer

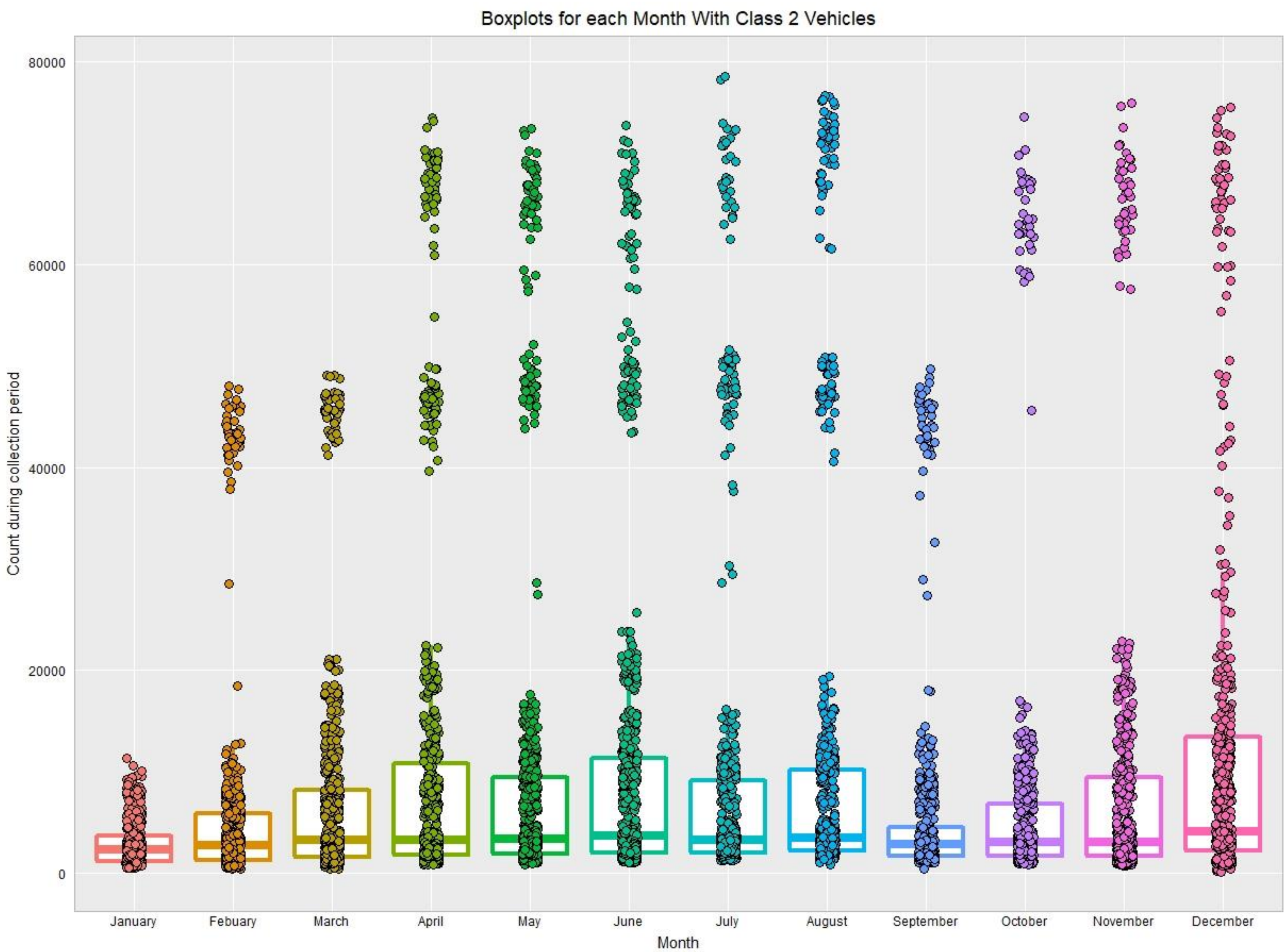
Appendix B

```
> summary(WIM)
    Station         Route          Lanes              Months.Used      Direction        Class.1              Class.2           Class.3
 5183    : 522   I-81    :1272   2:2555   Jan-Dec            :3601   EAST :1711   Min.    :   0.00   Min.   :   81   Min.   :   15
 6282    : 522   I-86    : 870   4:4608   Jan-Nov            :1396   NORTH:2239   1st Qu.:   2.00   1st Qu.: 1740   1st Qu.:  408
 6340    : 522   I-84    : 666   5:  43   Jan-Aug, Oct-Dec: 470   SOUTH:2244   Median :  10.00   Median : 3104   Median :  909
 7181    : 522   NY14    : 522   6: 686   Feb-Dec            : 444   WEST :1710   Mean   :  36.27   Mean   : 8959   Mean   : 1591
 1281    : 517   NY328   : 522   8:  12   Jan-Oct            : 386                3rd Qu.:  45.00   3rd Qu.: 8238   3rd Qu.: 1731
 1800    : 516   US 2    : 522            Feb-Sep, Dec       : 364                Max.   :1736.00   Max.   :78629   Max.   :11807
 (Other):4783   (Other):3530             (Other)            :1243
    Class.4          Class.5         Class.6          Class.7           Class.8           Class.9           Class.10          Class.11
 Min.   :  0.0   Min.   :   0   Min.   :  0.0   Min.   :  0.00   Min.   :  0.00   Min.   :   0.0   Min.   :  0.00   Min.   :  0.00
 1st Qu.:  1.0   1st Qu.:  24   1st Qu.: 15.0   1st Qu.:  2.00   1st Qu.: 16.00   1st Qu.: 146.0   1st Qu.: 16.00   1st Qu.:  0.00
 Median :  6.0   Median :  58   Median : 33.0   Median :  8.00   Median : 34.00   Median : 346.0   Median : 37.00   Median :  3.00
 Mean   : 25.6   Mean   : 169   Mean   : 71.3   Mean   : 16.61   Mean   : 70.72   Mean   : 641.0   Mean   : 53.92   Mean   : 26.07
 3rd Qu.: 21.0   3rd Qu.: 139   3rd Qu.: 81.0   3rd Qu.: 22.00   3rd Qu.: 79.00   3rd Qu.: 714.2   3rd Qu.: 62.00   3rd Qu.: 21.00
 Max.   :447.0   Max.   :2112   Max.   :624.0   Max.   :683.00   Max.   :524.00   Max.   :3023.0   Max.   :441.00   Max.   :272.00

    Class.12          Class.13       Count.Date.Month       sums        North.South
 Min.   : 0.000   Min.   : 0.000   May     : 776     Min.   :  142   No :3421
 1st Qu.: 0.000   1st Qu.: 0.000   April   : 771     1st Qu.: 2492   Yes:4483
 Median : 1.000   Median : 2.000   August  : 752     Median : 4508
 Mean   : 8.962   Mean   : 3.469   June    : 706     Mean   :11673
 3rd Qu.: 7.000   3rd Qu.: 5.000   July    : 690     3rd Qu.:12560
 Max.   :95.000   Max.   :94.000   October: 672     Max.   :93850
                                   (Other):3537
```

Appendix C

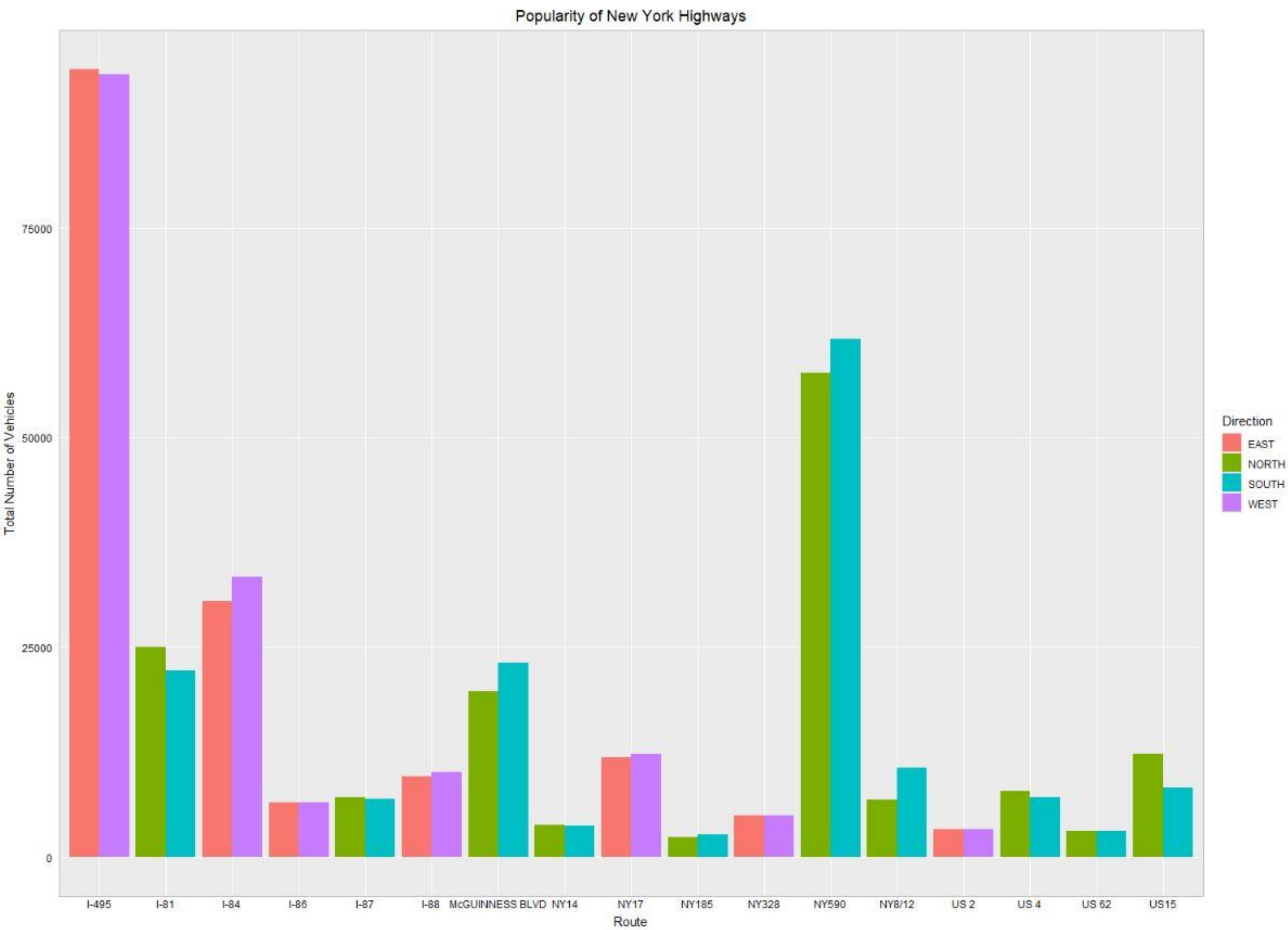


Appendix D



Boxplots for each Month With Class 2 Vehicles
Count during collection period
80000
60000
40000
20000
0
January
Febuary
March
April
May
June
July
August
September
October
November
December
Month

Appendix E



Popularity of New York Highways
Total Number of Vehicles
75000
50000
25000
0
I-495
I-81
I-84
I-86
I-87
I-88
McGUINNESS BLVD
NY14
NY17
NY185
NY328
NY590
NY8/12
US 2
US 4
US 62
US15
Route
Direction
EAST
NORTH
SOUTH
WEST

Appendix F

```
> glm_1 <- glm(North.South ~.,family = binomial, data = WIM_logistic[train,])
> summary(glm_1)

Call:
glm(formula = North.South ~ ., family = binomial, data = WIM_logistic[train,
    ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3050  -1.0306   0.5828   0.8784   2.6930

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)               1.035e+00  1.213e-01   8.533  < 2e-16 ***
Lanes4                   -1.575e+00  8.043e-02 -19.588  < 2e-16 ***
Lanes5                   -1.726e+01  2.339e+02  -0.074 0.941169
Lanes6                   -3.274e+00  5.393e-01  -6.070 1.28e-09 ***
Lanes8                    1.343e+01  5.055e+02   0.027 0.978810
Class.1                   1.376e-03  6.609e-04   2.082 0.037335 *
Class.2                   2.901e-04  1.225e-04   2.369 0.017851 *
Class.4                  -3.239e-02  1.947e-03 -16.637  < 2e-16 ***
Class.6                   7.435e-03  1.293e-03   5.750 8.94e-09 ***
Class.7                  -6.090e-03  1.804e-03  -3.376 0.000735 ***
Class.8                   7.536e-03  1.568e-03   4.806 1.54e-06 ***
Class.9                   7.406e-04  1.497e-04   4.946 7.58e-07 ***
Class.10                  6.140e-03  1.134e-03   5.415 6.12e-08 ***
Class.12                 -6.049e-02  4.424e-03 -13.672  < 2e-16 ***
Class.13                 -5.898e-03  8.948e-03  -0.659 0.509807
Count.Date.MonthFebuary   2.146e-01  1.543e-01   1.391 0.164184
Count.Date.MonthMarch     2.047e-01  1.489e-01   1.374 0.169316
Count.Date.MonthApril    -3.329e-01  1.417e-01  -2.350 0.018786 *
Count.Date.MonthMay      -1.112e-01  1.434e-01  -0.775 0.438180
Count.Date.MonthJune     -2.400e-01  1.453e-01  -1.652 0.098537 .
Count.Date.MonthJuly     -3.010e-01  1.473e-01  -2.044 0.041002 *
Count.Date.MonthAugust   -5.362e-02  1.459e-01  -0.367 0.713298
Count.Date.MonthSeptember  1.794e-01  1.502e-01   1.194 0.232437
Count.Date.MonthOctober  -1.075e-01  1.462e-01  -0.736 0.461939
Count.Date.MonthNovember -5.106e-01  1.509e-01  -3.383 0.000717 ***
Count.Date.MonthDecember -3.800e-01  1.569e-01  -2.422 0.015445 *
sums                     -2.080e-04  1.101e-04  -1.889 0.058850 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8630.8  on 6299  degrees of freedom
Residual deviance: 7154.4  on 6273  degrees of freedom
AIC: 7208.4

Number of Fisher Scoring iterations: 14
```

```
> cat("Model 1 R-squared =",1-
Model 1 R-squared = 0.1710565
> print(table(Predict_1, WIM_l

Predict_1  No Yes
      No  426 223
      Yes 247 708
> |
```

Appendix G

```
Call:
glm(formula = North.South ~ Lanes + Class.1 + Class.2 + Class.4 +
    Class.6 + Class.7 + Class.8 + Class.9 + Class.10 + Class.12,
    family = binomial, data = WIM_logistic[train, ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1639  -1.0349   0.6146   0.8457   2.7200

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  8.842e-01  6.581e-02  13.436  < 2e-16 ***
Lanes4      -1.582e+00  7.674e-02 -20.619  < 2e-16 ***
Lanes5      -1.725e+01  2.368e+02  -0.073   0.9419
Lanes6      -2.919e+00  5.260e-01  -5.549 2.88e-08 ***
Lanes8       1.354e+01  5.057e+02   0.027   0.9786
Class.1      1.366e-03  6.641e-04   2.057   0.0397 *
Class.2      5.311e-05  1.343e-05   3.954 7.69e-05 ***
Class.4     -3.197e-02  1.931e-03 -16.561  < 2e-16 ***
Class.6      5.189e-03  8.710e-04   5.958 2.55e-09 ***
Class.7     -7.163e-03  1.548e-03  -4.628 3.70e-06 ***
Class.8      6.460e-03  1.468e-03   4.400 1.08e-05 ***
Class.9      5.944e-04  1.240e-04   4.793 1.64e-06 ***
Class.10     6.373e-03  1.121e-03   5.687 1.29e-08 ***
Class.12    -6.078e-02  4.336e-03 -14.019  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8630.8  on 6299  degrees of freedom
Residual deviance: 7215.8  on 6286  degrees of freedom
AIC: 7243.8

Number of Fisher Scoring iterations: 14
```

```
> cat("Model 2 R-squared =",1-
Model 2 R-squared = 0.1639466
> print(table(Predict_2, WIM_l

Predict_2  No Yes
      No  425 228
      Yes 248 703
> |
```

Appendix H

```
Call:
glm(formula = North.South ~ Class.1 + Class.2 + Class.4 + Class.6 +
    Class.7 + Class.8 + Class.9 + Class.10 + Class.12, family = binomial,
    data = WIM_logistic[train, ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0158  -1.2736   0.7617   1.0048   2.8296

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.851e-01  4.872e-02   3.798 0.000146 ***
Class.1      1.303e-03  6.434e-04   2.025 0.042837 *
Class.2     -1.826e-06  4.526e-06  -0.404 0.686545
Class.4     -3.537e-02  1.945e-03 -18.186  < 2e-16 ***
Class.6      5.132e-03  8.078e-04   6.353 2.12e-10 ***
Class.7     -8.750e-03  1.482e-03  -5.903 3.57e-09 ***
Class.8      1.543e-02  1.419e-03  10.873  < 2e-16 ***
Class.9     -5.862e-04  1.101e-04  -5.324 1.01e-07 ***
Class.10     7.205e-03  1.076e-03   6.695 2.16e-11 ***
Class.12    -4.543e-02  4.100e-03 -11.081  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8630.8  on 6299  degrees of freedom
Residual deviance: 7752.7  on 6290  degrees of freedom
AIC: 7772.7

Number of Fisher Scoring iterations: 4
```

```
> cat("Model 3 R-squared =",1-
Model 3 R-squared = 0.1017315
> print(table(Predict_3, WIM_l

Predict_3  No Yes
      No  216  98
      Yes 457 833
> |
```

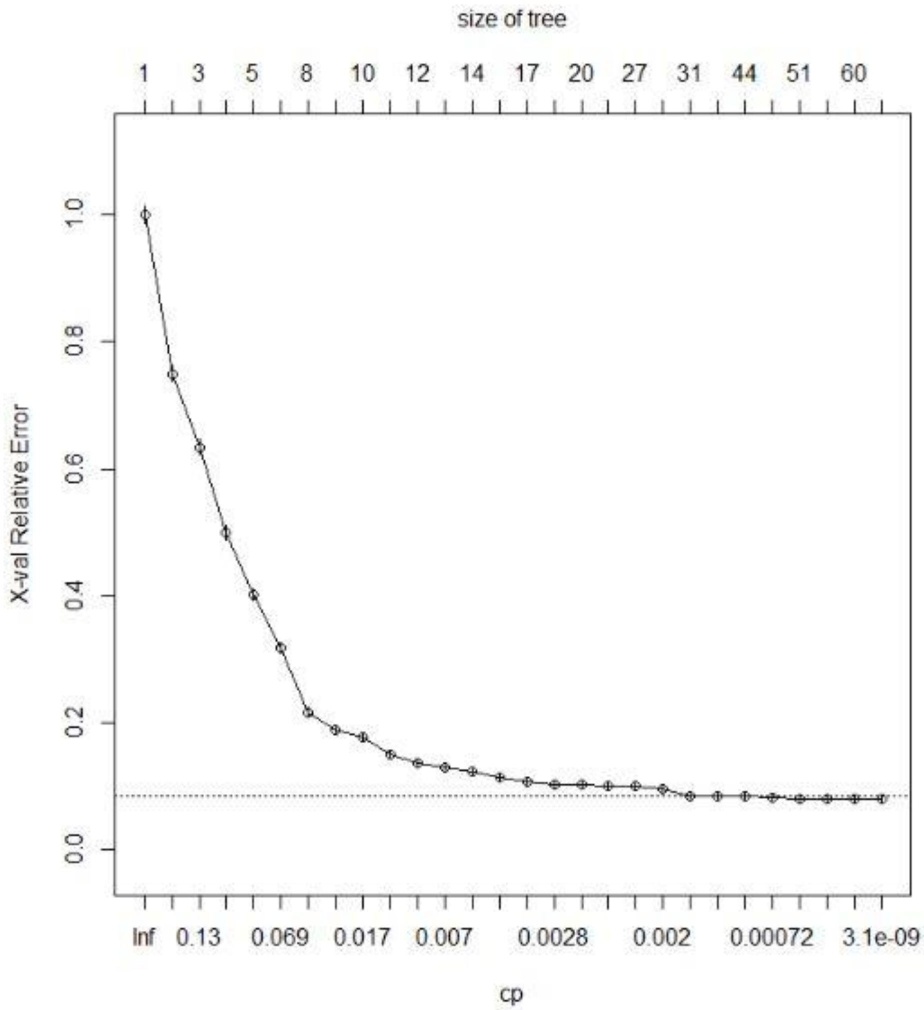
Appendix I

```
Call:
glm(formula = North.South ~ Class.1 + Class.4 + Class.6 + Class.7 +
    Class.8 + Class.9 + Class.10 + Class.12, family = binomial,
    data = WIM_logistic[train, ])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0018  -1.2736   0.7569   1.0059   2.8263

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.1836536  0.0485770   3.781 0.000156 ***
Class.1      0.0013031  0.0006421   2.030 0.042403 *
Class.4     -0.0353251  0.0019410 -18.200  < 2e-16 ***
Class.6      0.0049785  0.0007106   7.006 2.45e-12 ***
Class.7     -0.0087635  0.0014829  -5.910 3.43e-09 ***
Class.8      0.0152438  0.0013396  11.379  < 2e-16 ***
Class.9     -0.0005747  0.0001062  -5.409 6.34e-08 ***
Class.10     0.0072355  0.0010727   6.745 1.53e-11 ***
Class.12    -0.0454211  0.0040969 -11.087  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8630.8  on 6299  degrees of freedom
Residual deviance: 7752.9  on 6291  degrees of freedom
AIC: 7770.9

Number of Fisher Scoring iterations: 4
```

```
> cat("Model 4 R-squared =",1-
Model 4 R-squared = 0.1017127
> print(table(Predict_4, WIM_l

Predict_4  No Yes
      No  216  98
      Yes 457 833
> |
```

Appendix J

```
Classification tree:
rpart(formula = North.South ~ ., data = WIM_logistic, cp = 1e-13)

Variables actually used in tree construction:
 [1] Class.1           Class.10          Class.12
 [4] Class.2           Class.4           Class.6
 [7] Class.7           Class.8           Class.9
[10] Count.Date.Month Lanes              sums

Root node error: 3421/7904 = 0.43282

n= 7904

           CP nsplit rel error   xerror      xstd
1  2.5168e-01      0  1.000000 1.000000 0.0128761
2  1.2569e-01      1  0.748319 0.749488 0.0121661
3  1.2511e-01      2  0.622625 0.634025 0.0115963
4  9.8802e-02      3  0.497515 0.499854 0.0107006
5  8.7109e-02      4  0.398714 0.401929 0.0098514
6  5.4078e-02      5  0.311605 0.317451 0.0089468
7  3.2447e-02      7  0.203449 0.216896 0.0075795
8  1.7539e-02      8  0.171003 0.189418 0.0071295
9  1.7246e-02      9  0.153464 0.177141 0.0069145
10 1.3154e-02     10  0.136217 0.150833 0.0064197
11 7.3078e-03     11  0.123063 0.137679 0.0061520
12 6.7232e-03     12  0.115756 0.129787 0.0059839
13 4.9693e-03     13  0.109032 0.123648 0.0058489
14 4.6770e-03     15  0.099094 0.113709 0.0056216
15 3.2154e-03     16  0.094417 0.106986 0.0054613
16 2.4847e-03     17  0.091201 0.102894 0.0053608
17 2.3385e-03     19  0.086232 0.102602 0.0053535
18 2.1923e-03     24  0.074540 0.099678 0.0052802
19 2.0462e-03     26  0.070155 0.099678 0.0052802
20 1.9000e-03     28  0.066063 0.097048 0.0052131
21 1.1327e-03     30  0.062262 0.085063 0.0048938
22 1.0718e-03     39  0.051739 0.084186 0.0048695
23 8.7694e-04     43  0.047062 0.085063 0.0048938
24 5.8462e-04     47  0.043555 0.083017 0.0048368
25 2.9231e-04     50  0.041801 0.080094 0.0047540
26 1.4616e-04     57  0.039754 0.079801 0.0047456
27 9.7437e-05     59  0.039462 0.080386 0.0047624
28 1.0000e-13     62  0.039170 0.079509 0.0047373
```

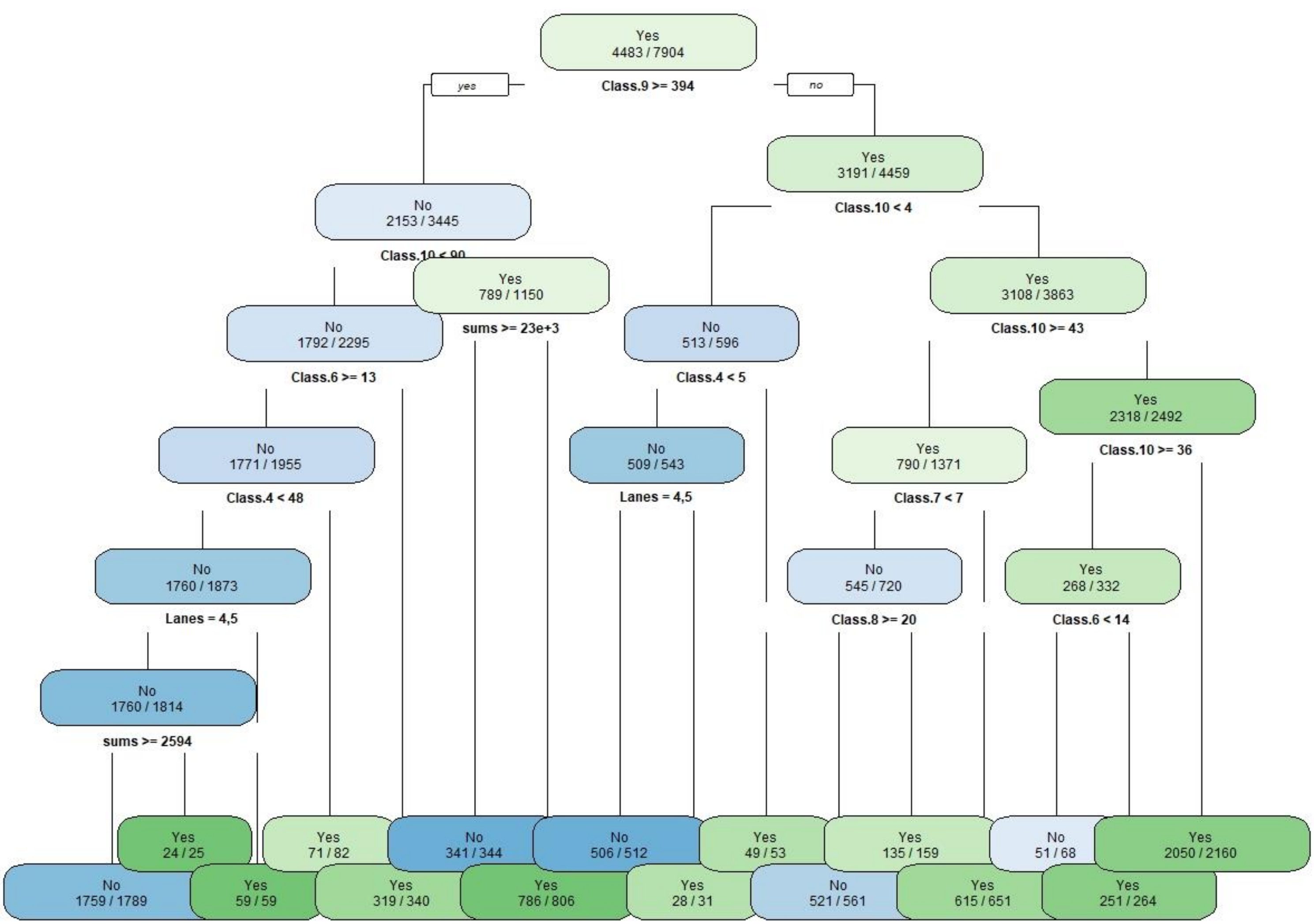


Appendix K

```
Classification tree:
rpart(formula = North.South ~ ., data = WIM_logistic, cp = 1e-13)

Variables actually used in tree construction:
[1] Class.10 Class.4  Class.6  Class.7  Class.8  Class.9  Lanes    sums

Root node error: 3421/7904 = 0.43282

n= 7904

          CP nsplit rel error  xerror      xstd
1  0.2516808      0  1.000000 1.00000 0.0128761
2  0.1256942      1  0.748319 0.74949 0.0121661
3  0.1251096      2  0.622625 0.63403 0.0115963
4  0.0988015      3  0.497515 0.49985 0.0107006
5  0.0871090      4  0.398714 0.40193 0.0098514
6  0.0540778      5  0.311605 0.31745 0.0089468
7  0.0324467      7  0.203449 0.21690 0.0075795
8  0.0175387      8  0.171003 0.18942 0.0071295
9  0.0172464      9  0.153464 0.17714 0.0069145
10 0.0131540     10  0.136217 0.15083 0.0064197
11 0.0073078     11  0.123063 0.13768 0.0061520
12 0.0067232     12  0.115756 0.12979 0.0059839
13 0.0049693     13  0.109032 0.12365 0.0058489
14 0.0047000     15  0.099094 0.11371 0.0056216
```

```
rpart.pred.rel.1  No Yes
             No  620  25
             Yes  53 906
> mean(rpart.pred.rel.1==W
[1] 0.9513716
```

WIM 0.1 rel error classification tree

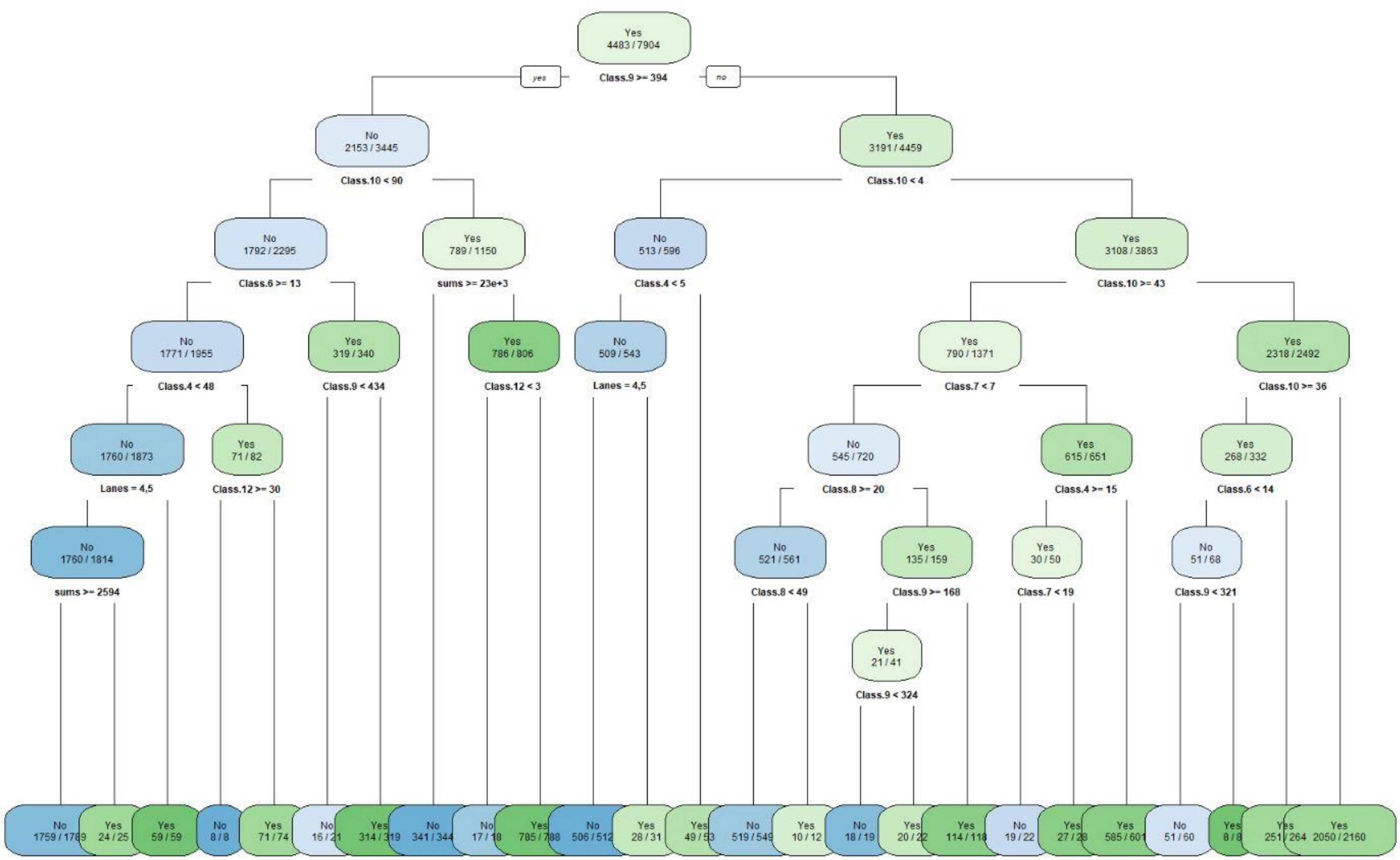


Appendix L

```
Classification tree:
rpart(formula = North.South ~ ., data = WIM_logistic, cp = 1e-13)

Variables actually used in tree construction:
[1] Class.10 Class.12 Class.4  Class.6  Class.7  Class.8  Class.9  Lanes    sums

Root node error: 3421/7904 = 0.43282

n= 7904

          CP nsplit rel error   xerror      xstd
1  0.2516808      0  1.000000 1.000000 0.0128761
2  0.1256942      1  0.748319 0.749488 0.0121661
3  0.1251096      2  0.622625 0.634025 0.0115963
4  0.0988015      3  0.497515 0.499854 0.0107006
5  0.0871090      4  0.398714 0.401929 0.0098514
6  0.0540778      5  0.311605 0.317451 0.0089468
7  0.0324467      7  0.203449 0.216896 0.0075795
8  0.0175387      8  0.171003 0.189418 0.0071295
9  0.0172464      9  0.153464 0.177141 0.0069145
10 0.0131540     10  0.136217 0.150833 0.0064197
11 0.0073078     11  0.123063 0.137679 0.0061520
12 0.0067232     12  0.115756 0.129787 0.0059839
13 0.0049693     13  0.109032 0.123648 0.0058489
14 0.0046770     15  0.099094 0.113709 0.0056216
15 0.0032154     16  0.094417 0.106986 0.0054613
16 0.0024847     17  0.091201 0.102894 0.0053608
17 0.0023385     19  0.086232 0.102602 0.0053535
18 0.0022000     24  0.074540 0.099678 0.0052802
```

```
rpart.pred.xerror.1  No Yes
                No  632  24
                Yes  41 907
> mean(rpart.pred.xerror.1==
[1] 0.9594763
> |
```

WIM 0.1 xerror classification tree



Appendix M

```
Call:
lm(formula = Class.2 ~ ., data = WIM_lm[train, ])

Residuals:
     Min       1Q   Median       3Q      Max
-11751.9   -275.5     22.5    303.6  10466.0

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.645e+03  1.720e+02  27.005  < 2e-16 ***
Station580   3.558e+04  3.433e+02 103.644  < 2e-16 ***
Station1281 -4.962e+03  1.779e+02 -27.891  < 2e-16 ***
Station1800 -5.248e+03  1.897e+02 -27.668  < 2e-16 ***
Station2680 -6.029e+03  1.727e+02 -34.903  < 2e-16 ***
Station3311  4.047e+02  2.439e+02   1.659 0.097077 .
Station4342  1.066e+04  2.519e+02  42.296  < 2e-16 ***
Station5183 -4.629e+03  1.875e+02 -24.684  < 2e-16 ***
Station5281 -4.496e+03  1.871e+02 -24.035  < 2e-16 ***
Station6100 -4.293e+03  1.953e+02 -21.975  < 2e-16 ***
Station6282 -7.020e+03  1.643e+02 -42.718  < 2e-16 ***
Station6340 -4.781e+03  1.772e+02 -26.981  < 2e-16 ***
Station6482 -5.740e+03  4.107e+02 -13.977  < 2e-16 ***
Station7100 -1.725e+03  2.676e+02  -6.446 1.23e-10 ***
Station7181 -4.482e+03  1.973e+02 -22.718  < 2e-16 ***
Station7381 -3.331e+03  2.029e+02 -16.415  < 2e-16 ***
Station8280  6.434e+03  2.811e+02  22.886  < 2e-16 ***
Station8382 -2.165e+03  2.586e+02  -8.371  < 2e-16 ***
Station9121 -4.097e+02  2.564e+02  -1.598 0.110091
Station9580 -2.963e+03  2.451e+02 -12.090  < 2e-16 ***
Station9631 -3.981e+03  1.958e+02 -20.330  < 2e-16 ***
Class.1      1.149e+00  3.253e-01   3.533 0.000414 ***
Class.3      6.283e+00  5.637e-02 111.462  < 2e-16 ***
Class.4      1.363e+01  1.300e+00  10.482  < 2e-16 ***
Class.5     -1.804e+01  3.667e-01 -49.195  < 2e-16 ***
Class.6     -1.696e+00  6.969e-01  -2.434 0.014964 *
Class.7     -3.744e+00  9.249e-01  -4.048 5.22e-05 ***
Class.8     -5.937e+00  8.817e-01  -6.733 1.81e-11 ***
Class.9     -1.290e+00  1.112e-01 -11.597  < 2e-16 ***
Class.10    -6.163e+00  7.812e-01  -7.889 3.57e-15 ***
Class.11     1.001e+01  1.583e+00   6.322 2.77e-10 ***
Class.12    -1.002e+01  3.941e+00  -2.544 0.010994 *
Class.13    -1.588e+01  3.999e+00  -3.972 7.22e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1031 on 6267 degrees of freedom
Multiple R-squared:  0.9956,    Adjusted R-squared:  0.9956
F-statistic: 4.481e+04 on 32 and 6267 DF,  p-value: < 2.2e-16
```

Appendix N

```
Call:
lm(formula = Class.2 ~ ., data = WIM_lm[train, ])

Residuals:
     Min       1Q   Median       3Q      Max
-11751.9   -275.5     22.5    303.6  10466.0

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.645e+03  1.720e+02  27.005  < 2e-16 ***
Station580   3.558e+04  3.433e+02 103.644  < 2e-16 ***
Station1281 -4.962e+03  1.779e+02 -27.891  < 2e-16 ***
Station1800 -5.248e+03  1.897e+02 -27.668  < 2e-16 ***
Station2680 -6.029e+03  1.727e+02 -34.903  < 2e-16 ***
Station3311  4.047e+02  2.439e+02   1.659 0.097077 .
Station4342  1.066e+04  2.519e+02  42.296  < 2e-16 ***
Station5183 -4.629e+03  1.875e+02 -24.684  < 2e-16 ***
Station5281 -4.496e+03  1.871e+02 -24.035  < 2e-16 ***
Station6100 -4.293e+03  1.953e+02 -21.975  < 2e-16 ***
Station6282 -7.020e+03  1.643e+02 -42.718  < 2e-16 ***
Station6340 -4.781e+03  1.772e+02 -26.981  < 2e-16 ***
Station6482 -5.740e+03  4.107e+02 -13.977  < 2e-16 ***
Station7100 -1.725e+03  2.676e+02  -6.446 1.23e-10 ***
Station7181 -4.482e+03  1.973e+02 -22.718  < 2e-16 ***
Station7381 -3.331e+03  2.029e+02 -16.415  < 2e-16 ***
Station8280  6.434e+03  2.811e+02  22.886  < 2e-16 ***
Station8382 -2.165e+03  2.586e+02  -8.371  < 2e-16 ***
Station9121 -4.097e+02  2.564e+02  -1.598 0.110091
Station9580 -2.963e+03  2.451e+02 -12.090  < 2e-16 ***
Station9631 -3.981e+03  1.958e+02 -20.330  < 2e-16 ***
Class.1      1.149e+00  3.253e-01   3.533 0.000414 ***
Class.3      6.283e+00  5.637e-02 111.462  < 2e-16 ***
Class.4      1.363e+01  1.300e+00  10.482  < 2e-16 ***
Class.5     -1.804e+01  3.667e-01 -49.195  < 2e-16 ***
Class.6     -1.696e+00  6.969e-01  -2.434 0.014964 *
Class.7     -3.744e+00  9.249e-01  -4.048 5.22e-05 ***
Class.8     -5.937e+00  8.817e-01  -6.733 1.81e-11 ***
Class.9     -1.290e+00  1.112e-01 -11.597  < 2e-16 ***
Class.10    -6.163e+00  7.812e-01  -7.889 3.57e-15 ***
Class.11     1.001e+01  1.583e+00   6.322 2.77e-10 ***
Class.12    -1.002e+01  3.941e+00  -2.544 0.010994 *
Class.13    -1.588e+01  3.999e+00  -3.972 7.22e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1031 on 6267 degrees of freedom
Multiple R-squared:  0.9956,    Adjusted R-squared:  0.9956
F-statistic: 4.481e+04 on 32 and 6267 DF,  p-value: < 2.2e-16
```

Appendix O

```
> print(sum_models$adjr2)
 [1] 0.9651496 0.9741710 0.9835738 0.9883456 0.9898213 0.9909265 0.9916530 0.9924972
 [9] 0.9929348 0.9931396 0.9934373 0.9937669 0.9941062 0.9943284 0.9944931 0.9946675
[17] 0.9948653 0.9949997 0.9951666 0.9953303 0.9953911 0.9954315 0.9954823 0.9955223
[25] 0.9955517 0.9955817 0.9956056 0.9956129 0.9956185 0.9956224
> 
```

Appendix P

```
Call:
lm(formula = Class.2 ~ Class.3, data = WIM_lm[train, ])

Residuals:
   Min     1Q Median     3Q    Max 
-12947  -1553     87   1196  47139 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.236e+03  4.561e+01  -49.02   <2e-16 ***
Class.3      7.050e+00  1.688e-02  417.67   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2911 on 6298 degrees of freedom
Multiple R-squared:  0.9652,	Adjusted R-squared:  0.9651 
F-statistic: 1.744e+05 on 1 and 6298 DF,  p-value: < 2.2e-16
```

```
p=1 R-squared:  0.9682727
```

Appendix Q

```
Call:
lm(formula = Class.2 ~ Class.3 + Class.6, data = WIM_lm[train, 
    ])

Residuals:
   Min     1Q Median     3Q    Max 
-17582  -1350    425   1283  42467 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.350e+03  3.934e+01  -59.74   <2e-16 ***
Class.3      8.756e+00  3.915e-02  223.64   <2e-16 ***
Class.6     -3.650e+01  7.780e-01  -46.91   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2506 on 6297 degrees of freedom
Multiple R-squared:  0.9742,	Adjusted R-squared:  0.9742 
F-statistic: 1.188e+05 on 2 and 6297 DF,  p-value: < 2.2e-16
```

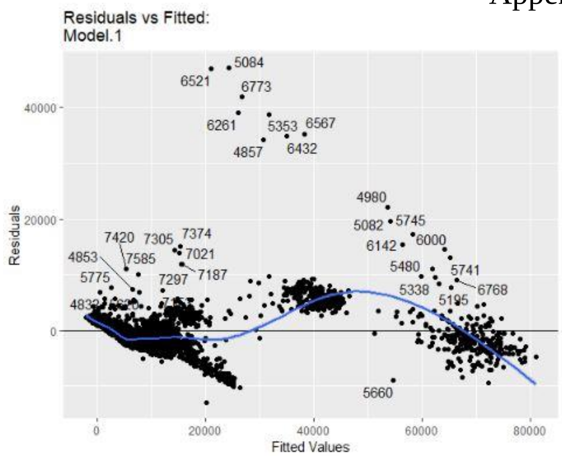
```
p=2 R-squared:  0.9777595
```

Appendix R

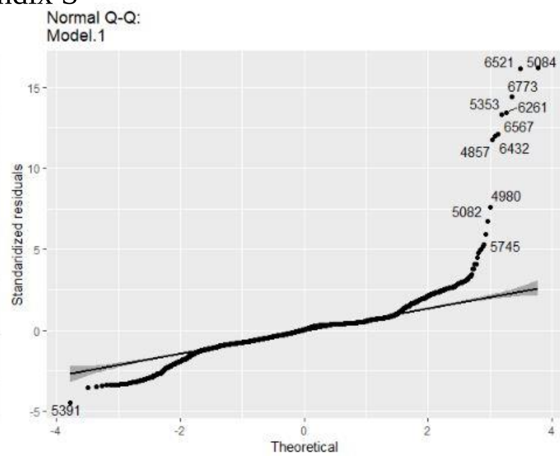
```
Call:
lm(formula = Class.2 ~ Class.3 + Class.9 + Station580 + Station4342 + 
    Station8280, data = WIM_lm[train, ])

Residuals:
     Min       1Q   Median       3Q      Max 
-15558.0   -683.5    -23.5    285.6  29803.1 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.907e+02  3.388e+01  -8.579   <2e-16 ***
Class.3      3.996e+00  3.122e-02 127.971   <2e-16 ***
Class.9      1.038e+00  3.646e-02  28.475   <2e-16 ***
Station580   2.673e+04  2.830e+02  94.438   <2e-16 ***
Station4342  2.157e+04  1.850e+02 116.565   <2e-16 ***
Station8280  6.153e+03  1.445e+02  42.570   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1584 on 6294 degrees of freedom
Multiple R-squared:  0.9897,	Adjusted R-squared:  0.9897 
F-statistic: 1.207e+05 on 5 and 6294 DF,  p-value: < 2.2e-16
```

```
p=5 R-squared:  0.9930083
```

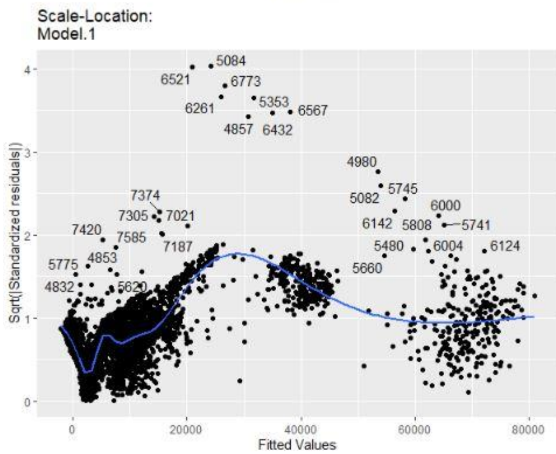
Appendix S



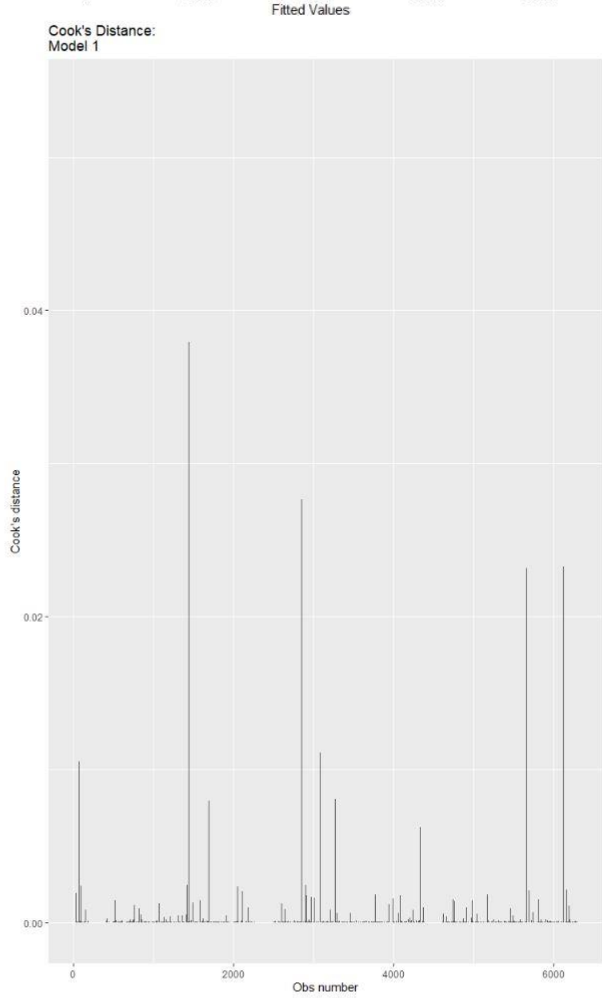
Residuals vs Fitted:
Model.1
Residuals
Fitted Values



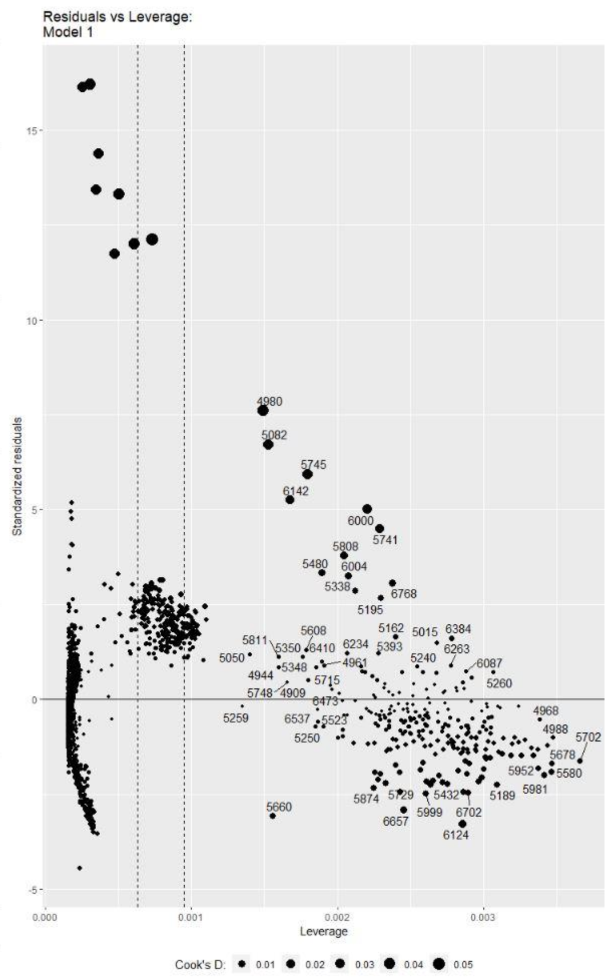
Normal Q-Q:
Model.1
Standardized residuals
Theoretical



Scale-Location:
Model.1
Sqrt(|Standardized residuals|)
Fitted Values



Cook's Distance:
Model 1
Cook's distance
Obs number



Residuals vs Leverage:
Model 1
Standardized residuals
Leverage
Cook's D: 0.01 0.02 0.03 0.04 0.05

Appendix T

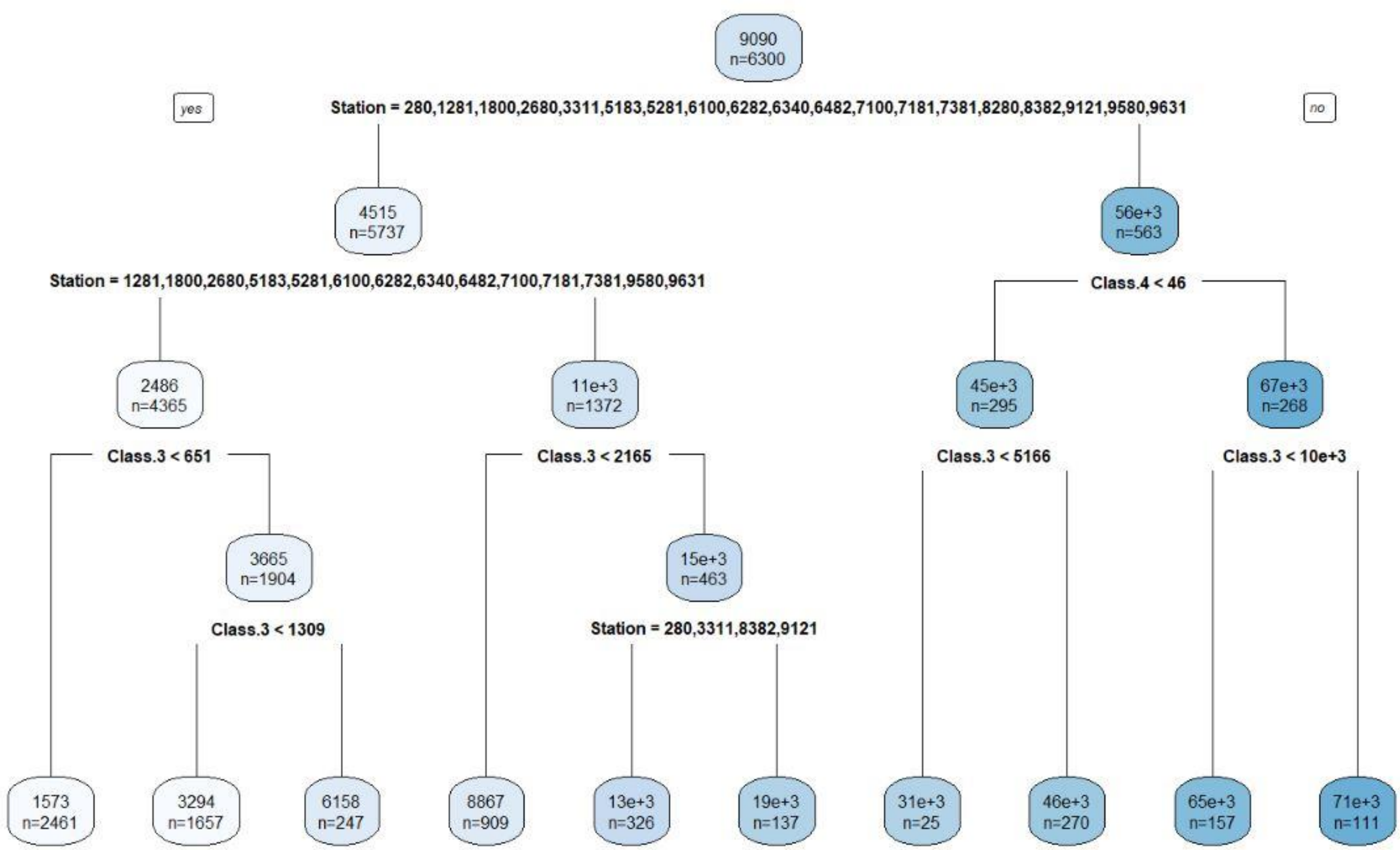
```
Regression tree:
rpart(formula = Class.2 ~ ., data = WIM_rt[train, ], method = "anova",
    cp = 1e-04)

Variables actually used in tree construction:
[1] Class.13 Class.3  Class.4  Class.9  Station

Root node error: 1.5311e+12/6300 = 243029935

n= 6300

           CP nsplit rel error    xerror       xstd
1  0.87769534      0 1.0000000 1.0001265 0.03809268
2  0.04910461      1 0.1223047 0.1224100 0.00359907
3  0.04519622      2 0.0732001 0.0733292 0.00322225
4  0.00779477      3 0.0280038 0.0284686 0.00128722
5  0.00334049      4 0.0202091 0.0208990 0.00116425
6  0.00306955      5 0.0168686 0.0204123 0.00136090
7  0.00223581      6 0.0137990 0.0151709 0.00109773
8  0.00133364      7 0.0115632 0.0128587 0.00108074
9  0.00115187      8 0.0102296 0.0118911 0.00104647
10 0.00085797      9 0.0090777 0.0104445 0.00103853
11 0.00062110     10 0.0082197 0.0096543 0.00103391
12 0.00057763     11 0.0075986 0.0091773 0.00102944
13 0.00050151     12 0.0070210 0.0086644 0.00101697
14 0.00033368     13 0.0065195 0.0079542 0.00097308
15 0.00022757     14 0.0061858 0.0078419 0.00120029
16 0.00022082     15 0.0059582 0.0076909 0.00119962
17 0.00021948     16 0.0057374 0.0075114 0.00119912
18 0.00018715     17 0.0055179 0.0071740 0.00119782
19 0.00014430     18 0.0053308 0.0070711 0.00119952
20 0.00013827     19 0.0051865 0.0069946 0.00119901
21 0.00012578     20 0.0050482 0.0069306 0.00119915
22 0.00011736     21 0.0049224 0.0067044 0.00119806
23 0.00011626     22 0.0048051 0.0065766 0.00119782
24 0.00010859     23 0.0046888 0.0064934 0.00119729
25 0.00010341     24 0.0045802 0.0064344 0.00119740
26 0.00010000     25 0.0044768 0.0064044 0.00119726
>
```

Appendix U

```
Regression tree:
rpart(formula = Class.2 ~ ., data = WIM_rt[train, ], method = "anova",
    cp = 1e-04)

Variables actually used in tree construction:
[1] Class.3 Class.4 Station

Root node error: 1.5311e+12/6300 = 243029935

n= 6300

          CP nsplit rel error   xerror      xstd
1  0.8776953      0 1.0000000 1.000127 0.0380927
2  0.0491046      1 0.1223047 0.122410 0.0035991
3  0.0451962      2 0.0732001 0.073329 0.0032222
4  0.0077948      3 0.0280038 0.028469 0.0012872
5  0.0033405      4 0.0202091 0.020899 0.0011643
6  0.0030695      5 0.0168686 0.020412 0.0013609
7  0.0022358      6 0.0137990 0.015171 0.0010977
8  0.0013336      7 0.0115632 0.012859 0.0010807
9  0.0011519      8 0.0102296 0.011891 0.0010465
10 0.0010000      9 0.0090777 0.010445 0.0010385
```

```
Rel Error 0.01 MSE= 2083481
> cat("Rel Error 0.01 sqrt(MSE)=",
Rel Error 0.01 sqrt(MSE)= 1443.427
>
```

WIM 0.1 rel error regression tree

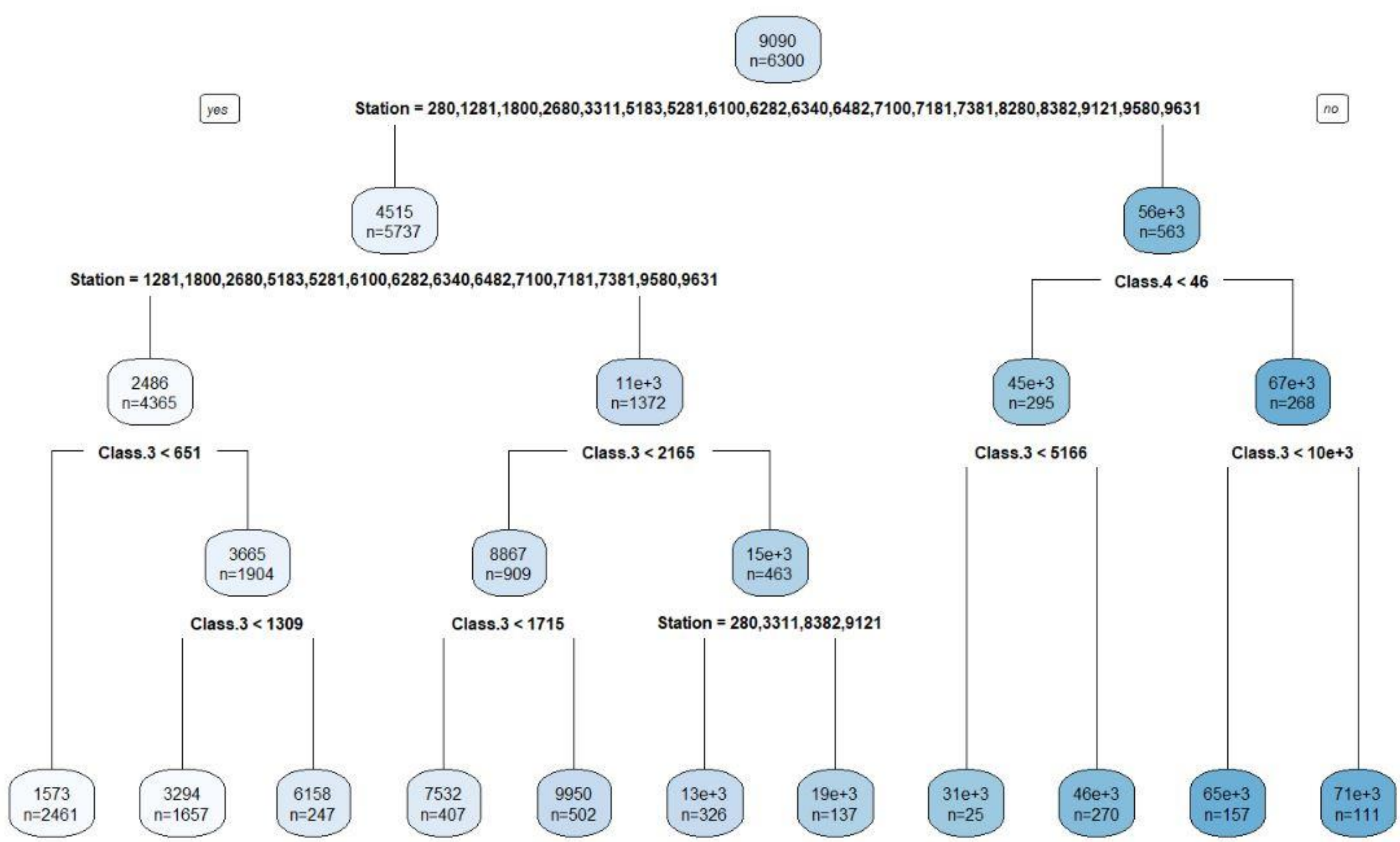


Appendix V

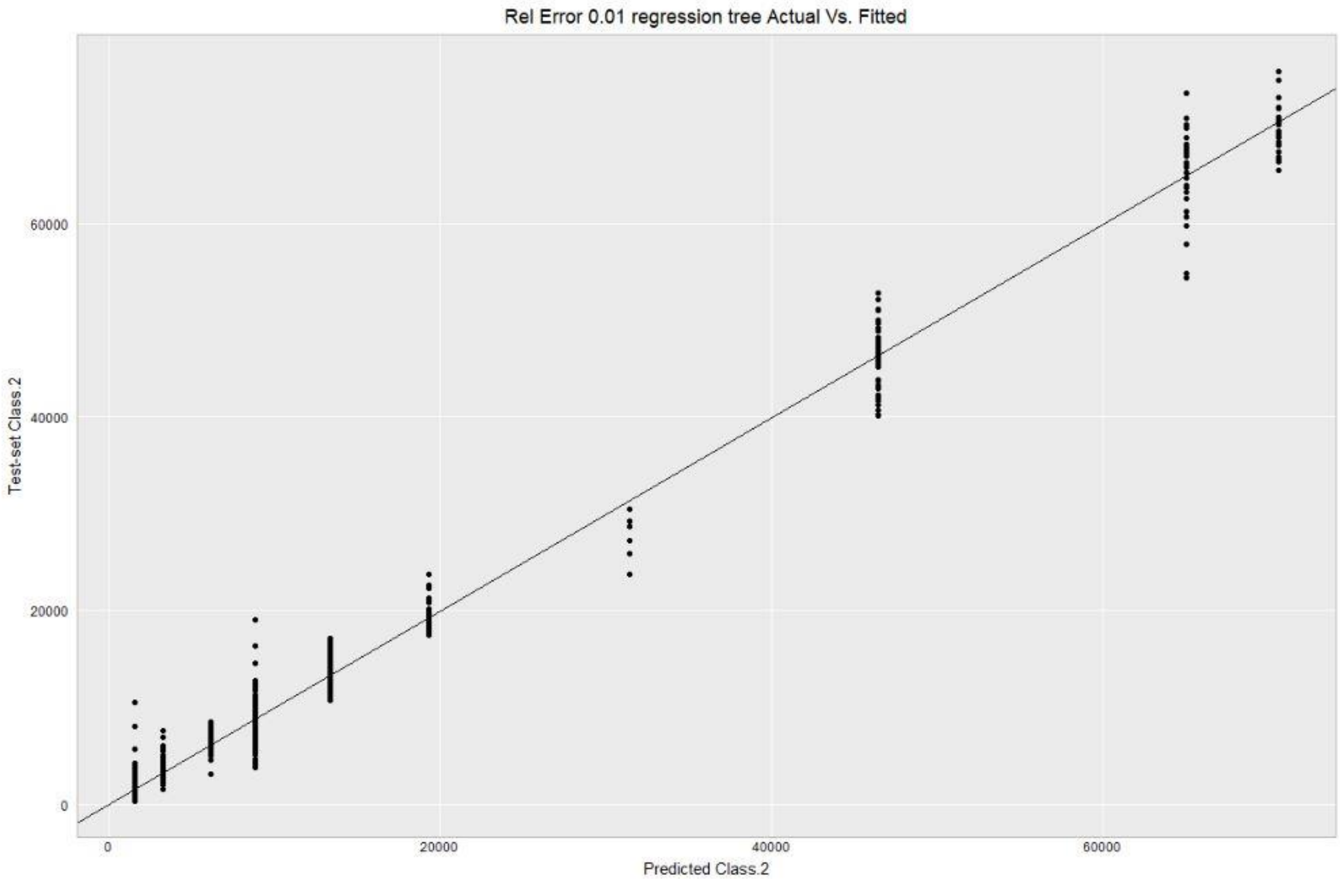
```
Regression tree:
rpart(formula = Class.2 ~ ., data = WIM_rt[train, ], method = "anova",
    cp = 1e-04)

Variables actually used in tree construction:
[1] Class.3 Class.4 Station

Root node error: 1.5311e+12/6300 = 243029935

n= 6300

           CP nsplit rel error    xerror      xstd
1  0.87769534      0 1.0000000 1.0001265 0.0380927
2  0.04910461      1 0.1223047 0.1224100 0.0035991
3  0.04519622      2 0.0732001 0.0733292 0.0032222
4  0.00779477      3 0.0280038 0.0284686 0.0012872
5  0.00334049      4 0.0202091 0.0208990 0.0011643
6  0.00306955      5 0.0168686 0.0204123 0.0013609
7  0.00223581      6 0.0137990 0.0151709 0.0010977
8  0.00133364      7 0.0115632 0.0128587 0.0010807
9  0.00115187      8 0.0102296 0.0118911 0.0010465
10 0.00085797      9 0.0090777 0.0104445 0.0010385
11 0.00070000     10 0.0082197 0.0096543 0.0010339
```

```
> cat("Xerror 0.01 MSE= ", (MSEx =
Xerror 0.01 MSE=  1902610
> cat("Xerror 0.01 sqrt(MSE)= ", (
Xerror 0.01 sqrt(MSE)=  1379.351
> |
```

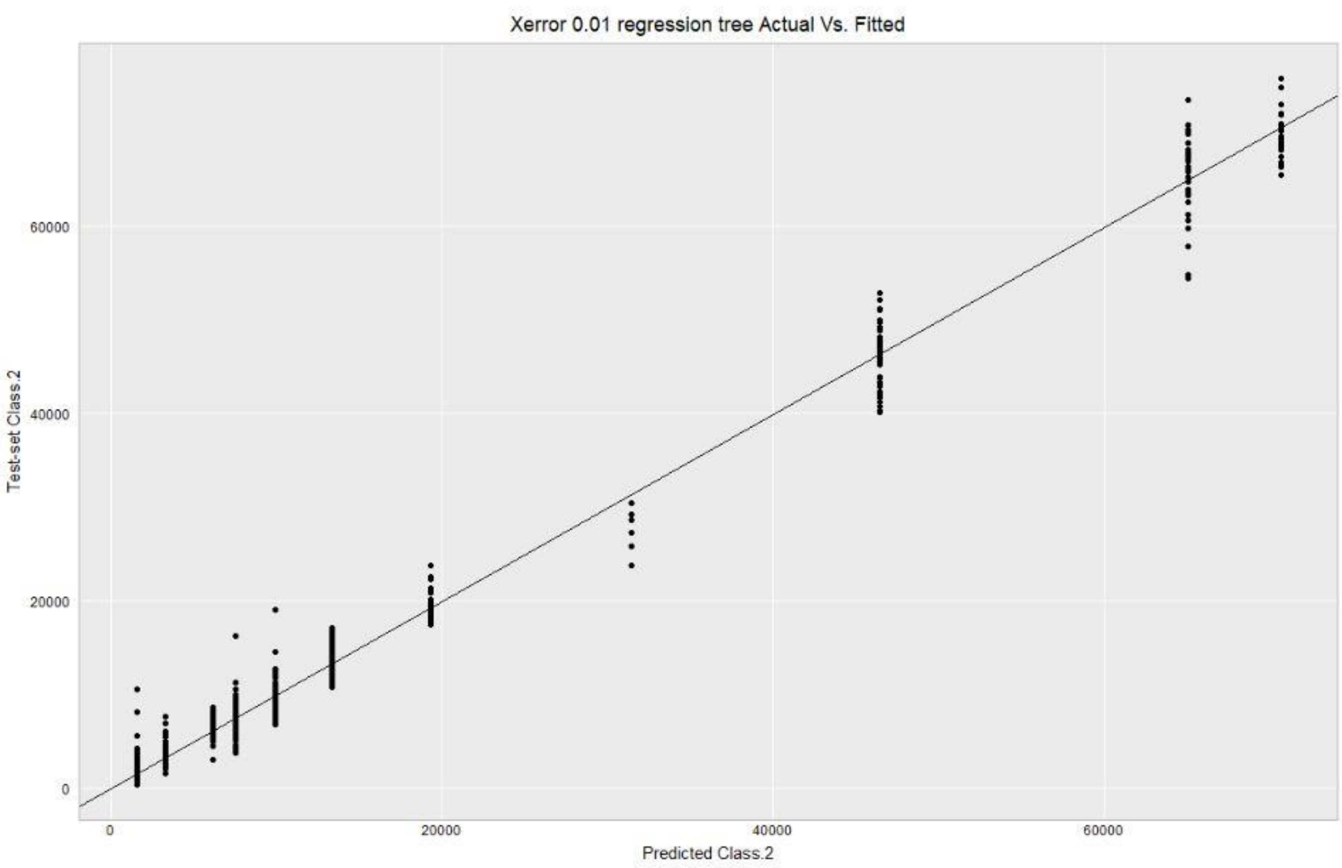
WIM 0.1 xerror regression tree



Appendix W



Rel Error 0.01 regression tree Actual Vs. Fitted
Test-set Class.2
0
20000
40000
60000
Predicted Class.2



Xerror 0.01 regression tree Actual Vs. Fitted
Test-set Class.2
0
20000
40000
60000
Predicted Class.2

References

[1] Data.NY.Gov. (June 13, 2019). Weigh-In-Motion Station Vehicle Traffic Counts: 2013, Weigh-In-Motion_Station_Vehicle_Traffic_Counts__2013.csv [data file]. New York State Department of Transportation (DOT)[producer]. Data.Gov [distributor]. https://catalog.data.gov/dataset/weigh-in-motion-station-vehicle-traffic-counts-2013

[2] Office of Highway Policy Information. (November 7, 2014). Traffic Monitoring Guide Appendix C. VEHICLE TYPES. U.S. Department of Transportation Federal Highway Administration. https://www.fhwa.dot.gov/policyinformation/tmguide/tmg_2013/vehicle-types.cfm

[3]R Core Team. (2016). R: A Language and Environment for Statistical Computing. Vienna, Austria. https://www.R-project.org/

[4] Lumley, Thomas. (January 16, 2020). Functions in leaps (3.1). RDocumentation. https://www.rdocumentation.org/packages/leaps/versions/3.1

[5] R-core@R-projecct.org. (December 31, 1969). Functions in stats (3.6.2). RDocumentation https://www.rdocumentation.org/packages/stats/versions/3.6.2

[6] Atkinson, Beth. (April 12, 2019). The ‘rpart’ package. RDocumentation https://www.rdocumentation.org/packages/rpart/versions/4.1-15

[7] Sigman, Richard (March, 10, 2021). Rss_regress_funcs_v2.R. George Mason University Volgenau School of Engineering. https://mymasonportal.gmu.edu/ultra/courses/_432480_1/cl/outline