Forest Fires: An Analysis of the Initial Spread Index

Nate Josephs[*]Matthew Wiens[†]Aaron Elliot[‡]Kelly Kung[§]Ben Draves[¶]

December 12, 2017

Abstract

Forest fires continue to be a serious ecological issue that endanger human lives and ravage environmental systems. At the time of this report, devastating wild fires are sweeping through Southern California and have destroyed more than 5,700 structures with more than 3,000 acres of land still burning. While human ability to fight forest fires has improved dramatically over recent decades, prevention is still the most effective course to minimizing environmental damage. The *Initial Spread Index (ISI)* is a metric that quantifies the speed at which a fire spreads. Firefighting agencies can use this metric to issue public warnings and deploy resources to most effectively monitor and contain forest fires before they cause substantial harm. This analysis uses linear modeling methodologies to infer key components that affect *ISI* and by extension determine what causes rapid spread of forest fires. Through penalization schemes and iterative model construction, we find that *wind speed*, *temperature*, and a *summer indicator variable* all significantly explained *ISI.* With this result, firefighting groups can leverage existing weather prediction systems to more effectively monitor and detect threats of deadly forest fires.

1 Introduction

Forest fires demolish entire ecosystems and have lasting consequences on the environment beyond their path of destruction. These fires deplete oxygen from the atmosphere, impact the lumber industry, vanquish animal habitats, and damage areas of natural beauty. They also contribute to pollution, carbon emission, soil erosion, flooding, and water contamination. As our planet is increasingly threatened by rising temperatures and dangerous policies, it is imperative that we be more responsible with our resources. Statistics offers many tools that can reveal ways to be more mindful and efficient in addressing complex problems. Here, we will use the theory of linear modeling to analyze the initial spread of forest fires. In particular, we look to pick up where Smokey Bear left-off: the next line of defense after prevention is early-detection. By understanding the factors of initial spread, we hope to contribute to the fight against forest fires.

To accomplish this goal, we examine *Initial Spread Index (ISI)* as our response variable from data obtained in Monteshino Natural Park between 2000 and 2003. It is worth emphasizing that our target is not forest fire occurrences. This subtle difference goes against our intuition. For example, regarding people, we expect that the number of fires increases with the presence of people. On the other hand, more people means earlier detection. Therefore, we have no *a priori* knowledge of the relationship between people and *ISI.* This difficulty extends to other variables such as rain and wind, in which different arguments could be made to explain different possible relationship with *ISI.*

This paper is organized as follows. In Section 2 we give a summary of our reference paper and restate our goal in this context. In Section 3 we outline our analysis, including our initial exploration of the data, all

[*] Introduction, Feature Design, Variable Selection, Appendix, Discussion

[†] Feature Design, Variable Selection, Prediction, Discussion

[‡] Modeling, Bootstrapping, Background, Discussion

[§] Modeling, Data Transformations, Feature Design, Discussion

[¶] Modeling, Data Overview, Variable Selection, Bootstrapping, Appendix, Discussion feature engineering and variable selection, and our final modeling decisions. In Section 5, we apply our final model to our holdout set in order to make unbiased inferences. We conclude our project in Section 6 with a general discussion of our results and possible future directions. Finally, R code for inline plots and the final modeling process may be found in Appendix I and Appendix II, respectively.

2 Background

In the reference paper provided[1], Cortez and Morais explored the potential of machine learning algorithms for predicting area burned by forest fires. They compared support vector machines (SVM), decision trees, multiple linear regression, neural networks, random forests, and a naïve mean prediction with four sets of features. The authors found that SVMs had the overall best success in predicting area burned using a combination of *temperature*, *rain*, *relative humidity* and *wind speed*. Furthermore, their analysis details that in the Fire Weather Index (FWI) system, *ISI* is a complex function of *wind* and *Fire Fuel Moisture Code (FFMC)*. We incorporate this discovery in our own analysis.

We use the same data curated by Cortez and Moraid, which includes meteorological data, as well as burned areas of forest fires within Montesinho Natural Park in the northeast Trás-os-Montes region of Portugal. The data were collected from January 2000 to December 2003 in separate datasets, which were manually integrated by the authors into a single dataset with a total of 517 observations. It should be noted that Cortez and Moraid do not claim this is a totally comprehensive dataset; some of the forest fires in the area may not have been recorded. We keep this in mind with regard to possible sampling bias.

Our analysis takes a standard approach by first splitting the data into training and testing sets (70-30% split), then performing feature engineering and variable selection on our training set, fitting our various models, and finally evaluating these models on our testing set. We acknowledge that due to the relatively small sample size of the dataset, there may be sparsity concerns in our training and testing sets. We address these issues as they arise. While we do make predictions on the test set, our focus is on making statistically significant inferential conclusions. Therefore, we avoid using overly complicated transformations, and when faced with competing models, we select the simpler version. We understand that this may result in a sub-optimal predictive model, but our goal is to obtain interpretable coefficients and generalizable results. We chose this inferential approach because we believe the first step to combating forest fires is understanding how they initially spread, and without an understanding, what good is a highly predictive model? Naturally we want to understand the likely causes of forest fires with a high *ISI*, of which we explore variables related to weather, time, location, and people.

3 Modeling & Analysis

3.1 Data Overview

The *ISI* attempts to quantify the risk associated with the initial spread of a forest fire. Heuristically, we expect variables that describe the weather characteristics of Monteshino Natural Park, as well as the terrain within the park, to affect the initial spread of a newly sparked forest fire. Although we do not know exactly how *ISI* is calculated, we surmise that these characteristics should play a major role in our modeling procedure. While we expect weather and topographic features to be two major explanatory classes, there is a third, far more subtle factor that we need to address. Consider Figure 1. It appears that fires are far more likely to occur in late summer and on the weekends. While the abundance of fires occurring in summer may be attributable to weather patterns[2], we suspect that the imbalance of fires on the weekend is connected in some way to the

[1] P. Cortez and A. Morais. A Data Mining Approach to Predict Forest Fires using Meteorological Data. In J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007 - Portuguese Conference on Artificial Intelligence, December, Guimares, Portugal, pp. 512-523, 2007.

[2] With a large spike in March, however, this behavior could be explained by both human behavior and weather trends.

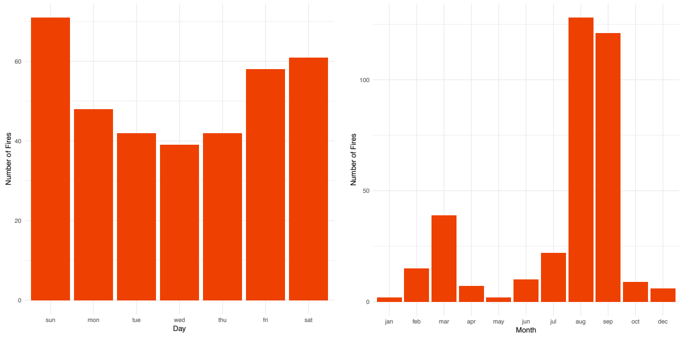


Figure 1: The data suggest more fires occur on the weekend and during the summer months.

presence of more visitors to the park. Although we do not have direct data on the number of individuals in the park on any given day, we expect that this variable greatly affects the way in which *the data were collected*. It seems reasonable to assume park visitors were either responsible for reporting the presence of a fire or starting the fires themselves. Therefore, when more visitors are in the park, we expect more fires to be detected, and more observations to be included in our dataset. So although we expect visitors to have no affect on how fast fires spread, due to the nature in which the data were collected, we include variables that model human presence.

With this in mind, we categorize our data into four types: weather, FWI indices, geo-spatial, and human impact. Weather covariates include rain totals, temperature, and relative humidity, while indices such as *FFMC*, *Duff Moisture Code (DMC)*, and *Drought Code (DC)* attempt to measure characteristics of the terrain. We attempt to capture the effect of geo-spatial components by using the (X, Y) coordinates, as well as designing new variables that identify homogeneous regions of the park. Lastly, we model the human impact inherent in data collection by introducing variables that serve as proxies for the number of visitors to the park on any given day. The construction of these variables will be covered in detail in Section 3.2.

In addition to these features, we also note that during our modeling procedure we found two abnormal instances of the response, *ISI*, in our training set: one point had a value of zero and another had a value 56.1. Considering the boxplot in Figure 2, we see that these two points, especially the outlier for $ISI = 56.1$, were quite abnormal. Upon further analysis, we were unable to identify any specific reason that these values deviated so far from the mean of the distribution. For this reason, we do not remove them from the training set. Instead, we performed sensitivity analysis during each step of the modeling process by repeating our analysis with and without these points, and concluded that the analysis was similar in both cases. Henceforth, we present only our results with these outliers included.

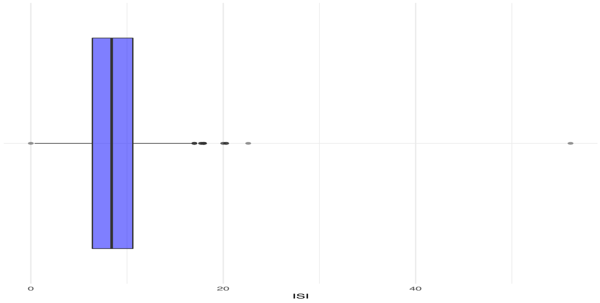


Figure 2: The boxplot of ISI suggests that there are significant outliers in our data.

3.2 Feature Design

Based on our intuition of *ISI*, as well as an initial look at the data, we transformed and created several covariates to obtain more informative predictors. We first found that area burned *(area)* is highly concentrated around zero[3] and extremely right skewed. Following the analysis of Cortez and Morais, we transformed *area* using $areaTrans = \ln(area + 1)$. The rain fall *(rain)* covariate is also concentrated around zero and right skewed, but we note that only 1.7% of the training set was non-zero. Therefore, we created an indicator of whether or not there was rain on the day of the given fire (*rainvnorain*).

In order to better support our intuition about the impact of weather on *ISI*, we also created a wetness metric, *wetness*, that measures the moisture in the air. We expect that this variable is important, because if there is more moisture in the air, then fires should not be able to spread as easily. Furthermore, if there is more moisture in the air, then plants and trees are also more moist, which means that fires lack the dry fuel they need in order to spread. To calculate this metric, we obtained initial values of *wetness* using the LennTech calculator[4] by identifying wetness values for corresponding temperature (*temp*) and relative humidity (*RH*) values. We then approximate this function in order to obtain metric values for each observation in our dataset.

We also created new indicators for the human impact feature class based on our initial analysis as explained in Section 3.1. We created the indicators *summer* and *weekend*, because we observed that more fires occurred during these time periods. In particular, $summer = 1$ corresponds to the months of *June, July, August*, and *September.* These months were chosen based on the climate of Portugal and the number of fires that occurred in this time frame. As for *weekend*, we chose the standard weekend days, *Friday, Saturday,* and *Sunday*, to receive a 1.

Another interesting feature class in this problem is geo-spatial. The raw data contains X and Y coordinates corresponding to a grid that has been overlaid on the map of Monteshino Natural Park. Each coordinate ranges from 1 to 9, therefore there are 81 total boxes in the grid. Of course, the first attempts at capturing any geo-spatial signal involved looking at the raw X and Y coordinates, as well as their interaction. Unfortunately, many of these boxes were sparse and the 81 degrees of freedom necessary for the raw grid were detrimental to the modeling process. Instead, we designed several features based on these values. This resulted in three candidate features. First, we created new coordinates $X2$ and $Y2$ that were created from the following algorithm

1. Set $X2 = X$ and $Y2 = Y$.
2. Sum the first row, last row, first column, and last column of the $X2 - Y2$ matrix separately.
3. Combine the row or column with the lowest sum with its neighboring row or column.
4. Repeat steps 2 and 3 until every entry contains at least 1% of the data.

The algorithm was written under the belief that neighboring spaces should be topographically similar and therefore should be the first candidates when grouping spaces together. Applying this algorithm, we reduced the number of boxes from 81 to 12. We note that $Y2$ only has two levels, which is unsurprising since Monteshino Natural Park is wider than it is tall. The other two engineered features integrated outside information. For these, we found a topographical map of Monteshino Natural Park on Google Maps and overlaid the original $X - Y$ grid. The result may be found in Appendix II. From this, we created *forest_ind*, which is a binary variable that takes the value 1 when the box is mostly covered in trees and 0 otherwise, and *grid_group*, which identifies five major mountain ranges and groups the boxes that cluster around these mountain ranges. Both of these variables are intuitively appealing, since forest fires burn more rapidly when trees are nearby and generally hikers tend to favor certain mountain ranges, which may have an effect on how fires are started.

Lastly, we used a Box-Cox Transformation to identify the appropriate power transformation for our response variable. The result was $\lambda = \frac{1}{2}$, corresponding to a square-root transformation, yielding our new response variable *sqISI.* Since *ISI* is an index, it is unitless. Therefore, we feel comfortable transforming it without loss of interpretation.

[3] Any fire with burn area lower than $100m^2$ was considered zero.

[4] https://www.lenntech.com/calculators/humidity/relative-humidity.htm

3.3 Variable Selection

In order to effectively consider all variables discussed above (along with several interaction terms), while still making inferential statements about *sqISI*, we use penalization schemes in order to identify important variables and classes of variables in our problem.[5] While Ridge Regression has several nice properties, we utilize LASSO for its ability to perform variable selection. By using LASSO, we allow an L_1 penalty to *zero-out* non-informative variables while also identifying variables that are representative of entire classes of covariates, e.g weather, spatial, etc.

After considering penalization coefficients under optimal smoothing parameters, as well as the added variable plots in successive iterations of the modeling procedure, we found that a model including the variables *summer*, *temp*, and *FFMC* were all key components in the mean function of *sqISI*. We note that the three engineered geo-spatial features from Section 3.2 were all considered as covariates, but unfortunately none were selected, as either LASSO zeroed-out their coefficients or they were deemed not significant under further scrutiny. Since we still expect that terrain should play a role in describing *sqISI*, we attempted to include this information in more complex ways in Section 4.1. On the other hand, representatives of weather, human impact, and FWI indices were all included. Lastly we note that all interaction terms were also zeroed-out by LASSO.

With these covariates, we fit a linear regression model to the data and find that this model performed very poorly. The residuals appears highly non-normal with non-constant variance. In addition, it appeared that any covariate except *FFMC* played little role in the model. Investigating the bivariate plot of *sqISI* and *FFMC*, found in Figure 3, we see that *FFMC* and *sqISI* are almost perfectly related, but in a non-linear fashion. Seeing the tight association between these two variables, it follows that any model selection procedure will rightfully include *FFMC* as a predictor. But due to the non-linearity inherent in this plot, any model including *FFMC* fails to satisfy the linear modeling assumptions. Now, if our goal is prediction, then we would simply fit a higher order polynomial of *FFMC*, which would successfully explain the majority of the variance in *sqISI*. This approach, however, would only reveal the relationship between *sqISI* and *FFMC*. Also, we know from Cortez and Morais that *FFMC* is a function of the other covariates found in this model (e.g. *wind*, *temp*, etc.). Therefore, we would only be able to infer the dependence between these other covariates and *sqISI through FFMC*. In this way, *FFMC* effectively masks the explanatory value of the other covariates. To illustrate this, consider the penalization paths with and without *FFMC* in Figure 4. Notice how when we exclude *FFMC*, the other variables' relative importance grows, while their penalization paths maintain a

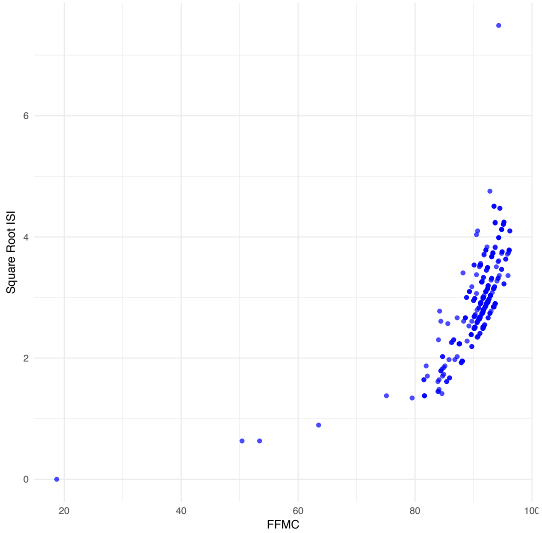


Figure 3: While *FFMC* and *sqISI* are closely related, this relationships is nonlinear.

[5] While we considered forward and backwards model selection procedures, information-theoretic approaches typically require normally-distributed residuals, which we found to be an unreasonable assumption based on our initial fits.

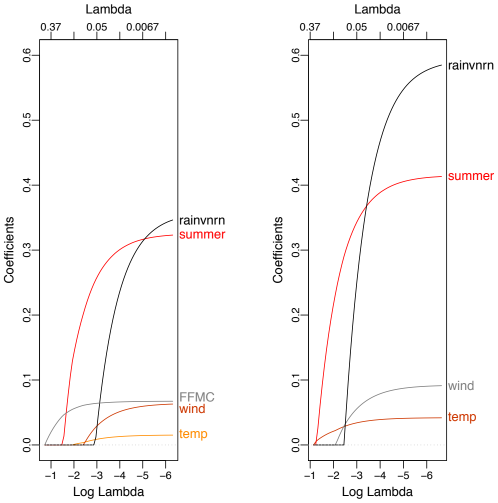


Figure 4: Removing *FFMC* from our modeling procedure suggests that *FFMC* and the weather covariates explain similar variability in *sqISI*.

similar shape. Therefore, we see that *FFMC* and this set of covariates do, in fact, model similar variability in *sqISI*. This was not immediately obvious by looking at correlation values and other collinearity statistics due to the non-linear relationship between *FFMC* and *sqISI*, but by investigating their penalization paths, we see that *FFMC* is skewing our analysis.

Since we chose an inferential approach to this problem, we wish to see the direct relationship between *sqISI*, and hence *ISI*, and the other covariates considered above. Therefore, we wish to simplify the *FFMC* variable and use the information from this variable in a way that does not mask the effects of *wind*, *summer*, *rainvnorain*, and *temp*. While we tried several different transformations of *FFMC* to make this pairwise relationship linear, we could find no interpretable function that resolved this issue. Instead, we note that this complex non-linear relationship is roughly piecewise linear around the point $FFMC = 80$. Therefore, we introduce an indicator variable *(tFFMC)* for $FFMC \leq 80$ into our modeling procedure.

After introducing $tFFMC$ into the modeling procedure, our final penalized coefficient estimates after iterative variable selection are given in Table 1. Using these coefficient values, along with further investigation of added variable plots and initial model fits, we found that the variables *wind*, *summer*, *temp*, *tFFMC*, and *rainvnorain* explain additional variance of *sqISI*. A pairwise scatter plot can also be found in Figure 5. For this reason, we focus our modeling on these weather covariates along with the newly introduced variable *tFFMC*. Seeing $tFFMC$ is the only non-weather covariate included, we attempt to use the information inherent in this variable in multiple ways which we discuss in the next section.

4 Modeling

4.1 Candidate Models

We consider models with *rainvnorain*, *temp*, *summer*, and *wind* as covariates. Therefore, our primary mean function is given by

$$\sqrt{ISI} = \beta_0 + \beta_1 * Summer + \beta_2 * Temperature + \beta_3 * Rain + \beta_4 * Wind + e \tag{1}$$

Upon fitting a regression model with this mean function, we found that the residuals appeared quite random, but were still not normally distributed. In an attempt to remedy this issue, we turned to weighted least squares models and mixed effects models with *tFFMC* and several geo-spatial variables constructed in Section 3.2 as

Variable	$\lvert\beta\rvert$	Variable	$\lvert\beta\rvert$
tFFMC	1.51	X2:7	0.07
rainvnorain	0.66	temp	0.04
summer	0.63	wkd	0.02
Intercept	0.28	X2:2	0.01
X2:5	0.26	areaTrans	0.01
forest_ind	0.14	DMC	0.00
X2:3	0.12	wetness	0.00
X2:4	0.09	DC	0.00
wind	0.07	X2:6	0.00
Y2:5	0.07	RH	0.00

Table 1: LASSO coefficients ranked by relative importance

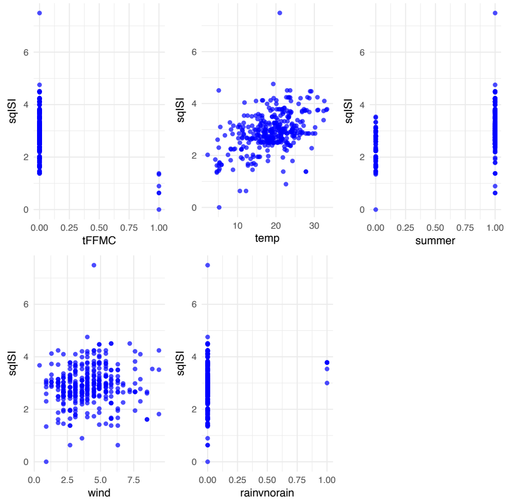


Figure 5: Bivariate plots of *sqISI* and relevant covariates used in modeling

ways to account for the non-normal nature of the residuals. We fit a random intercepts model to each level in *grid_group* and to each coordinate group in (X_2, Y_2). In both cases, we found no change in the distributions of the residuals. Moreover, we attempted weighting by the number of fires in each *grid_group* and each (X_2, Y_2) block. Again, we saw no improvement in the behavior of the residuals. We concluded that without further refinement or more granular data collection, all of the geo-spatial signal was being captured by the class of weather features, and therefore X, Y, and all derivative variables were excluded from our final model.

Next, we consider *tFFMC* both as a weighting variable and as a covariate. By weighting points with $FFMC \leq 80$ differently, we inherently assume that these points have a different variance structure than the remaining data. Referring to Figure 3, we see that while this may be the case, any definitive conclusions about the variance structure are unjustified, with such few observations with $FFMC \leq 80$. However, assuming that this portion of the data is indeed different in some way than those with $FFMC > 80$, we fit a model with mean function given in (1) plus *tFFMC*, as well as a weighted least squares model with the weights corresponding to the number of observations for $tFFMC = 0$ and $tFFMC = 1$. Again, we see that the residuals maintain a similar random pattern, while the Q-Q plots change marginally, which is evident from Figure 6. Here, we scaled these plots to exclude the point where $ISI = 56.1$ (though it was used to train the models), as it skewed the graphics and hindered us from analyzing the normality assumptions of the standardized residuals. We note that there is still odd behavior in the Q-Q plots at the tails of these distributions with the standardized residuals falling under and over the line $y = x$ at the theoretical quantiles greater than ±1. For this reason, we

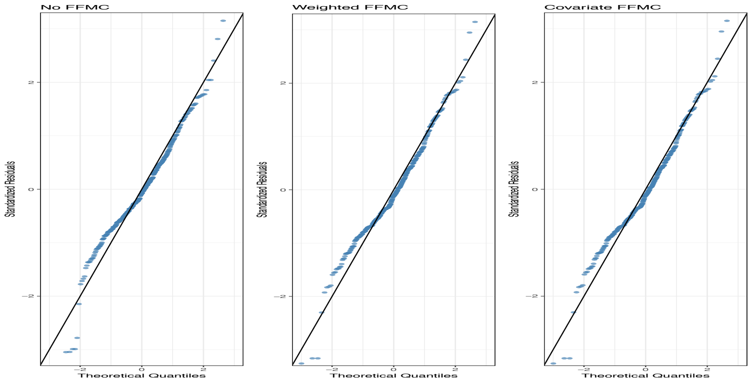


Figure 6: The addition of *FFMC* only marginally improves the linearity of residuals.

cannot justifiably use any model comparison techniques such as ANOVA. Instead, we note that the addition of *tFFMC* adds very little improvement to the normality of our errors when included as a covariate and as weights. Moreover, as we stated above, assuming that the points with $FFMC \leq 80$ have very different structure than those with $FFMC > 80$ may be unreasonable based on this sample size. Therefore, for the sake of simple inferential statements, we do not include *tFFMC* in our final model.[6] The diagnostic plots of the final model described by (1) can be found in Figure 7. We note that the standardized residuals appear

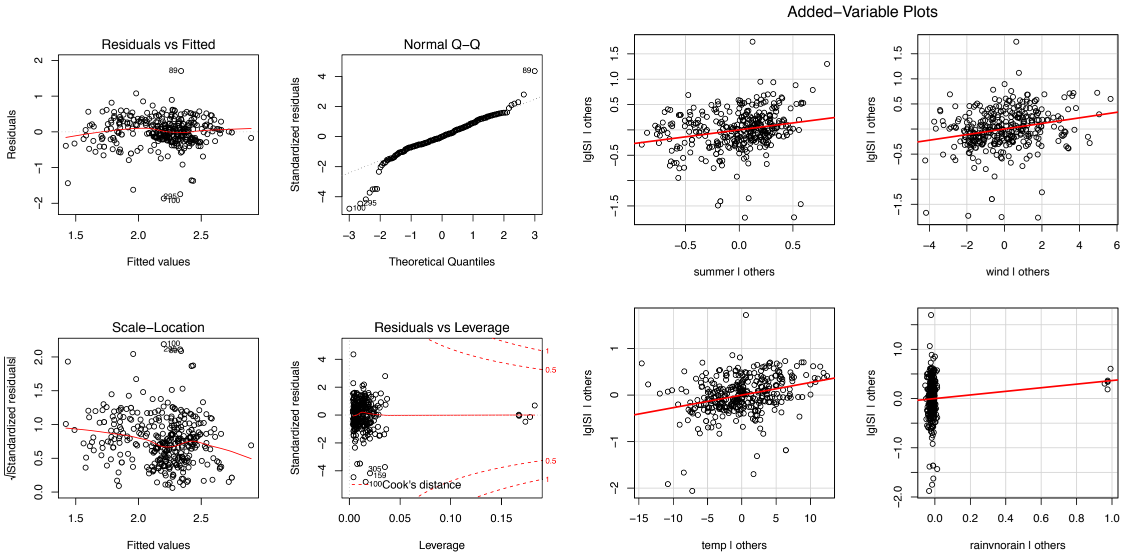


Figure 7: The final model appears to meet most modeling assumptions except that the residuals are normally distributed. The added variable plots suggest that each variable explains additional variance in *sqISI*.

to be random noise around zero, with a few large outliers towards the center of the data. These points correspond to the outliers in *ISI* but do not greatly affect the residual structure seen here. There appears to be a cluster of good leverage points as seen in the Residuals vs Leverage plot and we note that the Q-Q plot suggests that the residuals are not normal. Moreover, we see that the added variable plots suggest that each variable explains additional variance in the response *sqISI*. Before we test for significance of these variables, we note that by having non-normal errors we cannot use classical theory suggesting that our estimates follow a t-distribution. Instead, we turn to a nonparametric technique to construct empirical confidence intervals (CI) for testing significance in our final model.

4.2 Bootstrapping

Seeing that the residuals are non-normal in our final model, we instead use a nonparametric hypothesis testing framework to test the hypothesis $H_0 : \beta_i = 0$ against $H_A : \beta_i \neq 0$. We implement the bootstrap procedure to construct sampling distributions of each β_i and find the corresponding empirical CI. The results of this analysis are shown in Figure 8 and Table 2. We see that all sampling distributions are relatively normal and none of their confidence intervals contain 0. This means that we can reject the null hypotheses and conclude that *summer*, *wind*, *temp*, and *rain* are all significant in explaining *sqISI*.

[6] As we stated in Section 2, *FFMC* relies on the weather covariates found in (1). Therefore, while we exclude *tFFMC* here, we implicitly include its effect via the remaining weather covariates.

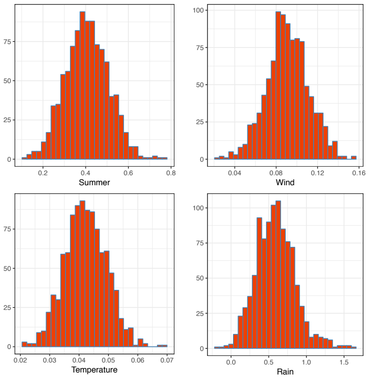


Figure 8: Empirical sampling distributions of β_i

Variable	95% Empirical CI
Wind	(0.06, 0.13)
Temperature	(0.03, 0.06)
Summer	(0.21, 0.61)
Rain	(0.11, 1.14)

Table 2: Empirical CIs

5 Prediction

Finally, we turn our attention to evaluating the weather-only based model given by (1) on the holdout set. We begin by fitting the chosen model on the testing data, which comprises 30% of the original dataset. The results of this model are given in Figure 9. In the Actual vs Predicted plot, we see that our inferential model captures the overall trend of *sqISI*. Considering the predicted *sqISI* values, the Mean Squared Error (MSE) was calculated to be 13.4. In comparison, the MSE for the training set was 17.1. However, as previously noted, there are significant outliers in the the training set, which explains the higher MSE.

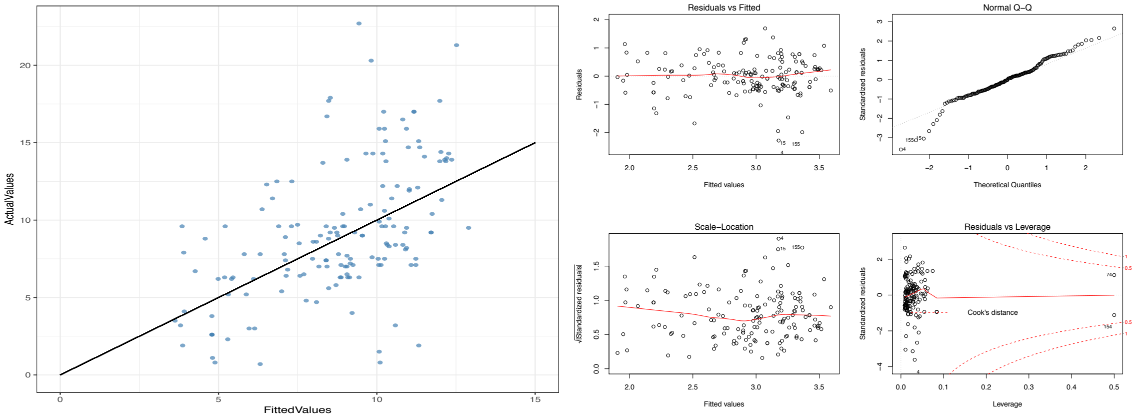


Figure 9: Testing results - predicted vs actual sqISI and model diagnostic plots.

Turning to the diagnostic plots in Figure 9, the model given here matches the behavior of the model fit on the training data. That is, our residuals appear random yet non-normal. This can be seen in the $Q - Q$ plot, which exhibits significant left tail behavior. Therefore, CIs for our coefficient estimates were constructed from a bootstrap sample. We then use these to make conclusions about the significance of the covariates. The results of this analysis are summarized in Table 3 and Table 4.

All the CIs except for the binary *Is Raining* variable in Table 4 do not contain zero, so we conclude that they are all different than zero. As the *Is Raining* CI does contain zero, we do not conclude that the *Is Raining* variable is significantly different than zero. We also note that the CI is quite wide and there is a large standard error, because there were only two data points with rain in the test set. While rain was uncommon in the training set, it was less sparse than in the testing set, so it is difficult to make inferences on such a variable.

The analysis on the testing set mostly confirms the model selection performed on the training set, with

Variable	Estimate	Standard Error
Intercept	1.31	0.24
Is Summer	0.58	0.16
Wind	0.079	0.029
Temperature	0.045	0.012
Is Raining	-0.26	0.47

Table 3: Coefficient estimates on testing set

Variable	95% CI
Intercept	$(0.86,\ 1.76)$
Is Summer	$(0.27,\ 0.91)$
Wind	$(0.02,\ 0.13)$
Temperature	$(0.02,\ 0.07)$
Is Raining	$(-1.30,\ 0.59)$

Table 4: Test set empirical CIs

the key difference being the loss of significance for the *Is Raining* variable. We conclude that all weather variables except for *rain* impact *sqISI*, and fail to conclude that the presence of rain is a predictor for the model. Based on an MSE criterion, the predictions from the model are reasonably close to the actual data.

6 Discussion

After a thorough modeling process, we selected the variables *Wind*, *Temperature*, *Summer*, and *Rain* based on a LASSO-driven approach to variable selection. Higher wind speeds, higher temperatures, and summertime conditions all correspond to higher *ISI* while the presence of rain did not play a role in predicting *ISI*. Wind was found to be the most important predictor of *ISI*, so when considering fire conditions, predicting wind speed and allocating resources based on this quantity should be prioritized. In addition, warm temperatures are an important consideration when predicting fires. Summertime was also predictive of *ISI*, which we suggest captures other underlying weather features such as long stretches of dry weather that are not otherwise captured in the data. These findings match our intuition that dry, windy conditions are conducive to rapid fire spread.

As noted throughout this report, there is a latent observation bias throughout our dataset: each of the observations represents a single fire. As a result, our observations are imbalanced across spatial coordinates, months of year, and even days of the week despite the fact that *ISI* is defined, though not recorded, even when no fire occurs. Therefore, to conduct a complete analysis, one must collect data relating the covariates considered here and *ISI* uniformly over the course of time with disregard to the presence of a fire. This will remove any implicit effect that fires have on *ISI* as well as the human effect discussed throughout the duration of this report. By collecting temporal and spatial data, the effect of weather and topographical attributes may be tested directly.

Throughout our report, we focus primarily on an inferential approach. That is, we only consider simple, interpretable models that provide insight to the *ISI* and, by extension, determine what causes initial rapid spread of fires. In this way, we sacrifice predictive accuracy, because as previously noted, *FFMC* is highly non-linearly predictive, and thus if we simply wanted to predict *ISI*, we could build a complex function of *FFMC* to predict *ISI*, at the cost of interpretability of the model. We choose to keep the interpretable model, because the loss of predictive power is minimal and we prefer to investigate the direct cause of weather impacts on the initial spread of fire. Moreover, this approach empowers firefighting organizations to focus on real-time, easily accessible data when allocating resources to stop fires at their source.

Based on these results, we suggest further investigation into the impact of summer conditions on *ISI* and the impact of the Portuguese Mediterranean climate. In particular, focus could be on the specific climatic differences during the summertime and the impact on fire susceptibility in ways that are not captured by other indicies such as *DC* or *DMC*. Similarly, Portuguese forestry officials could measure changes in the forest through the year, including the amount of dry grass on the forest floor and other flammable vegetation to gather additional data to predict the initial spread of fires.

Appendix I: Figure Code

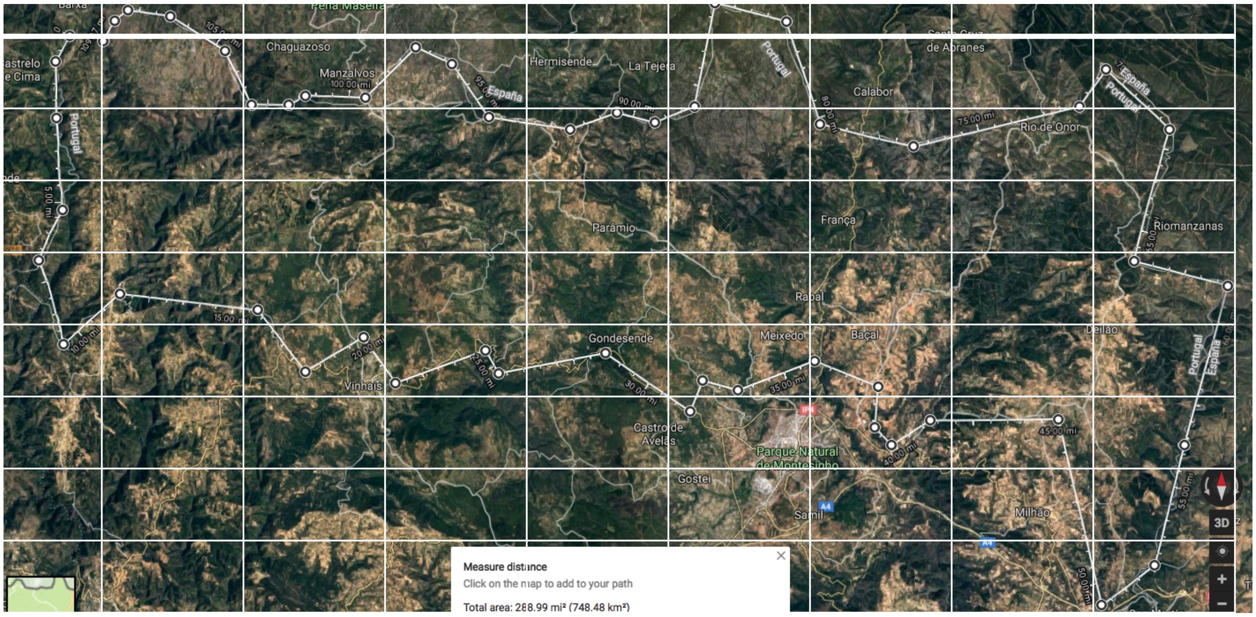
```r
train <- data.frame(read.csv("train.csv"))
test <- data.frame(read.csv("test.csv"))
library(ggplot2)
library(dplyr)
library(glmnet)
library(gridExtra)
library(car)

#-------------------------------------------
#       Figure 1
#-------------------------------------------
train$month <- factor(train$month
                      , levels = c("jan", "feb", 'mar', 'apr', 'may', 'jun'
                                   , 'jul', 'aug','sep', 'oct', 'nov', 'dec'))
train$day <- factor(train$day
                    , levels = c('sun','mon','tue','wed','thu','fri','sat'))
df1 <- data.frame(train %>% group_by(month) %>% summarize(lmonth = n()))
df2 <- data.frame(train %>% group_by(day) %>% summarize(lday = n()))
train$sqISI <- sqrt(train$ISI)

p1 <- ggplot(data = df1, aes(x = month, y = lmonth)) +
  geom_bar(stat = "identity", fill = "orangered2") +
  theme_minimal() +
  labs(y = "Number of Fires", x = "Month", main = "")

pdf("month_bar.pdf")
p1
dev.off()

p2 <- ggplot(data = df2, aes(x = day, y = lday)) +
  geom_bar(stat = "identity", fill = "orangered2") +
  theme_minimal() +
  labs(y = "Number of Fires", x = "Day", main = "")

pdf("day_bar.pdf")
p2
dev.off()

#-------------------------------------------
#       Figure 2
#-------------------------------------------

p3 <- ggplot(data = train, aes(x = "", y = ISI)) +
  geom_boxplot(fill = "blue", alpha = 0.5) +
  theme_minimal() +
  labs(y = "ISI", x = "", main = "") +
  coord_flip()

pdf("ISI_box.pdf")
p3
```

```r
dev.off()

#-------------------------------------------
#       Figure 3
#-------------------------------------------

p4 <- ggplot(data = train, aes(x = FFMC, y = sqISI)) +
  geom_point(color = "blue", alpha = 0.7) +
  theme_minimal() +
  labs(y = "Square Root ISI", x = "FFMC", main = "")

pdf("FFMC_ISI_scatter.pdf")
p4
dev.off()

#-------------------------------------------
#       Figure 5
#-------------------------------------------

f1 <- formula(sqISI ~ temp + wind + summer + rainvnorain)
f2 <- formula(sqISI ~ temp + wind + summer + rainvnorain + FFMC)

X1 <- model.matrix(f1,train)
X2 <- model.matrix(f2,train)
Y <- as.matrix(train$sqISI)

fit1 <- glmnet(X1, Y, alpha = 1)
fit2 <- glmnet(X2, Y, alpha = 1)

pdf("penalization_plots.pdf")
par(mfrow = c(1, 2), mai = c(1, 0.5, 0.1, 0.1))
plot_glmnet(fit2, ylim = c(0, .6))
plot_glmnet(fit1, ylim = c(0, .6))
dev.off()

#-------------------------------------------
#       Figure 6
#-------------------------------------------

train <- data.frame(train %>% group_by(tFFMC) %>% mutate(weight = n()))
m1 <- lm(sqISI ~temp + wind + summer + rainvnorain
         , data = train)
m2 <- lm(sqISI ~temp + wind + summer + rainvnorain
         , weights = weight
         , data = train)
m3 <- lm(sqISI ~ temp + wind + summer + rainvnorain + tFFMC
         , data = train)
df <- data.frame(r1 = as.vector(qqnorm(resid(m1), plot = F))
                 , r2 = as.vector(qqnorm(resid(m2), plot = F))
                 , r3 = as.vector(qqnorm(resid(m3), plot = F)))

p5 <- ggplot(m1, aes(qqnorm(.stdresid)[[1]], .stdresid)) +
  geom_point(na.rm = TRUE, col = "steelblue", alpha = 0.7) +
  geom_abline() +
  xlab("Theoretical Quantiles") +
  ylab("Standardized Residuals") +
  ggtitle("No FFMC") +
  theme_bw() +
  coord_cartesian(ylim = c(-3, 3))

p6 <- ggplot(m2, aes(qqnorm(.stdresid)[[1]], .stdresid)) +
  geom_point(na.rm = TRUE, col = "steelblue", alpha = 0.7) +
  geom_abline()+xlab("Theoretical Quantiles") +
  ylab("Standardized Residuals") +
  ggtitle("Weighted FFMC") +
  theme_bw() +
  coord_cartesian(ylim = c(-3, 3))

p7 <- ggplot(m2, aes(qqnorm(.stdresid)[[1]], .stdresid)) +
  geom_point(na.rm = TRUE, col = "steelblue", alpha = 0.7) +
  geom_abline()+xlab("Theoretical Quantiles") +
  ylab("Standardized Residuals") +
  ggtitle("Covariate FFMC") +
  theme_bw() +
  coord_cartesian(ylim = c(-3, 3))

pdf("FFMC-QQ.pdf")
grid.arrange(p5, p6, p7, nrow = 1, ncol = 3)
dev.off()

#-------------------------------------------
#       Figure 7
#-------------------------------------------

train <- data.frame(train %>% group_by(tFFMC) %>% mutate(weight = n()))
m <- lm(lgISI ~ summer + wind + temp + rainvnorain
        , data = train)

pdf("final_model_diag.pdf")
par(mfrow = c(2, 2))
plot(m)
dev.off()

pdf("final_avp.pdf")
avPlots(m)
dev.off()

#-------------------------------------------
#       Figure 8
#-------------------------------------------

m1 <- lm(sqISI ~ summer + wind + temp + rainvnorain
         , data = train)

B <- 1000
ResidualBootstrapM1 <- t(replicate(B, {
  yb <- fitted(m1) + resid(m1)[sample.int(nrow(train), replace = TRUE)]
  boot <- model.matrix(m1)
  coef(lm(yb ~ boot - 1))
}))

hist(ResidualBootstrapM1[,1], xlab = "summer")
hist(ResidualBootstrapM1[,2], xlab ="wind")
hist(ResidualBootstrapM1[,3], xlab ="temp")
hist(ResidualBootstrapM1[,4], xlab ="rainvnorain")
df <- data.frame(ResidualBootstrapM1)

p8 <- ggplot(df, aes(x = bootsummer1)) +
  geom_histogram(fill = "orangered2", color = "steelblue") +
  theme_bw() +
  labs(x = "Summer", y = "", main = "")

p9 <- ggplot(df, aes(x = bootwind)) +
  geom_histogram(fill = "orangered2", color = "steelblue") +
  theme_bw() +
  labs(x = "Wind", y = "")

p10 = ggplot(df, aes(x = boottemp)) +
  geom_histogram(fill = "orangered2", color = "steelblue") +
  theme_bw() +
  labs(x = "Temperature", y = "")

p11 = ggplot(df, aes(x = bootrainvnorain1)) +
  geom_histogram(fill = "orangered2", color = "steelblue") +
  theme_bw() +
  labs(x = "Rain", y = "")

pdf("boot.pdf")
grid.arrange(p8, p9, p10, p11, nrow = 2, ncol = 2)
dev.off()

#-------------------------------------------
#       Figure 9
#-------------------------------------------

test$sqISI <- sqrt(test$ISI)

test$summer <- as.factor(test$summer)
test$rainvnorain <- as.factor(test$rainvnorain)

test_mod <- lm(sqISI ~ summer + wind + temp + rainvnorain ,data = test)

plotdf <- data.frame(cbind(test$sqISI^2, test_mod$fitted.values^2))
names(plotdf) <- c("ActualValues", "FittedValues")

p12 <- ggplot(plotdf, aes(x= FittedValues, y = ActualValues)) +
  geom_point(col = "steelblue", alpha = .7) +
  geom_segment(aes(x = 0, y = 0, xend = 15, yend = 15), col = "black") +
  theme_bw()
```

```
pdf("pred_plot.pdf")
p12
dev.off()

pdf("TestSetDiagnostics.pdf")
par(mfrow = c(2, 2))
plot(test_mod)
dev.off()
```

Appendix II: Final Modeling Procedure

```r
#-------------------------------------------
#       Section 3.1-3.2
#-------------------------------------------
# create the indicators for weekend and summer
wkd <- rep(0, nrow(fires))
wkd[fires$day %in% c("fri", "sat", "sun")] <- 1
wkdM <- rep(0,nrow(fires))
wkdM[fires$day %in% c("fri", "sat", "sun", "mon")] <- 1
summer <- rep(0, nrow(fires))
summer[fires$month %in% c("jun", "jul", "aug", "sep")] <- 1

fires$wkd <- wkd
fires$wkdM <- wkdM
fires$summer <- summer

# transform the area
areaTrans <- log(fires$area + 1)
fires$areaTrans <- areaTrans

# rain indicator
rainvnorain <- rep(0, nrow(fires))
rainvnorain[fires$rain != 0] <- 1
fires$rainvnorain <- rainvnorain

# wetness metric
rel_humid100_temp <- c(0, 5, 10, 15, 17, 19, 20, 22, 24, 26, 29, 32, 35)
rel_humid100_water <- c(4.2, 5.74, 7.84, 10.7, 12.12, 13.73, 14.62, 16.56
                        , 18.76, 21.25, 25.62, 30.89, 37.24)

water_at_full <- approxfun(x = rel_humid100_temp, y = rel_humid100_water )
est_wetness_metric <- function(Temp, rh){
  return(water_at_full(Temp) * rh)
}
wetness <- est_wetness_metric(fires$temp, fires$RH)

fires$wetness <- wetness
```



Measure distance

Click on the map to add to your path

Total area: 288.99 mi² (748.48 km²)

```
#--------------------------------------------
#        Section 3.2
#--------------------------------------------

# simplified FFMC
fires$tFFMC <- ifelse(fires$FFMC < 80, 0, 1)

# indicator for forest
forest_coords <- c(1, 1, 1, 1, 1, 1, 1, 1, 1
                   , 0, 0, 1, 1, 1, 0, 1, 1, 1
                   , 0, 0, 1, 0, 0, 0, 1, 1, 0
                   , 0, 0, 1, 0, 0, 0, 1, 1, 0
                   , 0, 1, 0, 0, 1, 1, 1, 1, 1
                   , 0, 0, 0, 1, 0, 0, 0, 0, 1
                   , 1, 1, 1, 1, 0, 0, 0, 0, 1
                   , 1, 1, 1, 1, 1, 0, 0, 0, 0
                   , 1, 1, 1, 1, 1, 0, 0, 0, 0)
forest_coords <- matrix(forest_coords, nrow = 9, ncol = 9)
for(i in 1:nrow(fires)){
  fires[i, "forest_ind"] <- forest_coords[fires[i, "X"], fires[i, "Y"]]
}

# geo-spatial grid
fires[, "grid_group"] <- "other"                       # default (other)
```

```
fires[fires$X %in% c(1, 2, 3) &
        fires$Y %in% c(2, 3, 4), "grid_group"] <- "tl" # top left mountain
fires[fires$X %in% c(3, 4, 5) &
        fires$Y %in% c(3, 4, 5), "grid_group"] <- "ml" # middle left mountain
fires[fires$X %in% c(5, 6, 7) &
        fires$Y %in% c(3, 4, 5), "grid_group"] <- "mr" # middle right mountain
fires[fires$X %in% c(7, 8) &
        fires$Y %in% c(6, 7), "grid_group"] <- "br"    # bottom right mountain

# transform response variable (ISI)
fires$sqISI <- sqrt(fires$ISI)

#  create train/test split
set.seed(575)
train.ind <- sample.int(n = nrow(fires), size = floor(nrow(fires) * 0.7), replace = FALSE)
train<- fires[train.ind, ]
test <- fires[-train.ind, ]
```

```
#-------------------------------------------
#       Section 3.2
#-------------------------------------------
# calculate the groupings of FFMC by training quantile
ffmcQuant <- quantile(train$FFMC, probs = seq(0, 1, .1))
FFMCQuantile_train <- rep(0, nrow(train))
FFMCQuantile_test <- rep(0, nrow(test))
for (i in 10) {
  FFMCQuantile_train[ffmcQuant[i] < train$FFMC &
                       train$FFMC <= ffmcQuant[i + 1]] <- i
  FFMCQuantile_test[ffmcQuant[i] < test$FFMC &
                      test$FFMC <= ffmcQuant[i + 1]] <- i
}

train$FFMCQuantile <- FFMCQuantile_train
test$FFMCQuantile <- FFMCQuantile_test

# condense X-Y grid
train$X2 <- train$X
train$Y2 <- train$Y
i <- 0
while (i < 1) {
  m <- as.matrix(table(train$Y2, train$X2))
  top <- sum(m[rownames(m) == min(rownames(m)), ])
  bottom <- sum(m[rownames(m) == max(rownames(m)), ])
  left <- sum(m[, colnames(m) == min(colnames(m))])
  right <- sum(m[, colnames(m) == max(colnames(m))])

  if (top == min(top, bottom, left, right)) {
    train[train$Y2 == min(rownames(m)), "Y2"] <- as.integer(min(rownames(m))) + 1
    if (min(prop.table(table(train$Y2, train$X2))) < .01) {
      i = 0
    } else{
      i = 1
    }
  } else if (bottom == min(top, bottom, left, right)) {
    train[train$Y2 == max(rownames(m)), "Y2"] <- as.integer(max(rownames(m))) - 1
    if (min(prop.table(table(train$Y2, train$X2))) < .01) {
      i = 0
    } else{
      i = 1
    }
  } else if (left == min(top, bottom, left, right)) {
    train[train$X2 == min(colnames(m)), "X2"] <- as.integer(min(colnames(m))) + 1
    if (min(prop.table(table(train$Y2, train$X2))) < .01) {
      i = 0
    } else{
      i = 1
    }
  } else {
    train[train$X2 == max(colnames(m)), "X2"] <- as.integer(max(colnames(m))) - 1
    if (min(prop.table(table(train$Y2, train$X2))) < .01) {
      i = 0
    } else{
      i = 1
    }
  }
}
```

```
#-------------------------------------------
#       Section 3.3
#-------------------------------------------
# cast as factors
vars_factors <- c("wkd", "wkdM", "summer", "FFMCQuantile", "rainvnorain", "grid_group"
                  , "month", "day", "X2", "Y2")
for(var in vars_factors) {
  train[, var] <- as.factor(train[, var])
}

# construct regression equation
f <- formula(sqISI ~ -1
             + tFFMC
             + X2
             + Y2
             + temp
             + RH
             + wind
             + wkd
             + summer
             + rainvnorain
             + forest_ind
             + DMC
             + DC
             + areaTrans
             + wetness
)
# build model matrix
X <- model.matrix(f, train)
Y <- as.matrix(train$sqISI)
a <- 1
```

```
# run lasso
cv = cv.glmnet(X, Y, alpha = a)
lambda_opt = cv$lambda.min

lasso <- glmnet(X, Y, alpha = a, lambda = lambda_opt)
tmp <- sort(abs(coef(lasso)[, 1]), decreasing = TRUE)
varImp <- data.frame(VarNames = names(tmp), Beta = round(as.vector(tmp), 3))
varImp

# fit lm with most important variables
m <- lm(sqISI ~
          tFFMC
        + rainvnorain
        + summer
        + forest_ind
        + wind
        + temp
        + X2
        + wkd
        , data = train)

# diagnostics and added variable plots
par(mfrow = c(2, 2))
plot(m)
```

```
summary(m)
avPlots(m)
```

```
#-------------------------------------------
#       Section 4.1
#-------------------------------------------
# Weight based on tFFMC
train <- data.frame(train %>% group_by(tFFMC) %>% mutate(weight = n()))
m_weight <- lm(sqISI ~
                 rainvnorain
               + summer
               + wind
               + temp
               , weights = weight
               , data = train)
par(mfrow = c(2, 2))
plot(m_weight)
```

```
summary(m_weight)
avPlots(m_weight)
```

```
# Random intercepts with tFFMC
train[train$X2 == 2 & train$Y2 == 4, "region"] = 1
train[train$X2 == 3 & train$Y2 == 4, "region"] = 2
train[train$X2 == 4 & train$Y2 == 4, "region"] = 3
train[train$X2 == 5 & train$Y2 == 4, "region"] = 4
train[train$X2 == 6 & train$Y2 == 4, "region"] = 5
train[train$X2 == 7 & train$Y2 == 4, "region"] = 6
```

```
train[train$X2 == 2 & train$Y2 == 5, "region"] = 7
train[train$X2 == 3 & train$Y2 == 5, "region"] = 8
train[train$X2 == 4 & train$Y2 == 5, "region"] = 9
train[train$X2 == 5 & train$Y2 == 5, "region"] = 10
train[train$X2 == 6 & train$Y2 == 5, "region"] = 11
train[train$X2 == 7 & train$Y2 == 5, "region"] = 12

train$region = as.factor(train$region)

m_rand <- lme(sqISI ~ summer + wind + temp + rainvnorain + tFFMC
              , random = ~1|region
              , data =  train[-89, ]
              , method = "REML")
summary(m_rand)
plot(m_rand)
```

```
par(mfrow = c(1, 1))
qqnorm(m_rand$residuals)
```

```
#--------------------------------------------
#       Section 4.1-4.2
#--------------------------------------------
# build final model
m1 = lm(sqISI ~ summer + wind + temp + rainvnorain
        , data = train)

# MSE for raw ISI
1 / nrow(train) * sum((train$sqISI^2 - m1$fitted.values^2)^2)

# 1000 boostrap samples for each coefficient
B <- 1000
ResidualBootstrapM1 <- t(replicate(B, {
  yb <- fitted(m1) + resid(m1)[sample.int(nrow(train), replace = TRUE)]
  boot <- model.matrix(m1)
  coef(lm(yb ~ boot - 1))
}))

# look at distributions of coefficients
par(mfrow = c(2, 2))
hist(ResidualBootstrapM1[,1], xlab = "summer", main = "")
hist(ResidualBootstrapM1[,2], xlab = "wind", main = "")
hist(ResidualBootstrapM1[,3], xlab = "temp", main = "")
hist(ResidualBootstrapM1[,4], xlab = "rainvnorain", main = "")
```

```
# empirical CI
t(apply(ResidualBootstrapM1, 2, quantile, c(.025, .975)))
```

```
#--------------------------------------------
#       Section 5
#--------------------------------------------
# cast as factors
vars_factors <- c("summer", "rainvnorain")
for(var in vars_factors) {
  test[, var] <- as.factor(test[, var])
}

# fit candidate model on test data
m1 <- lm(sqISI ~ summer + wind + temp + rainvnorain
         , data = test)
par(mfrow = c(2, 2))
summary(m1)
plot(m1)
```

```
avPlots(m1)
```

```
# plot fitted values
par(mfrow = c(1, 1))
plotdf <- data.frame(cbind(test$sqISI^2, m1$fitted.values^2))
names(plotdf) <- c("ActualValues", "FittedValues")
ggplot(plotdf, aes(x = FittedValues, y = ActualValues)) +
  geom_point() +
  geom_segment(aes(x = 0, y = 0, xend = 15, yend = 15), col = "blue")
```

```
# MSE for response ISI
1 / nrow(test) * sum((test$sqISI - m1$fitted.values)^2)
# MSE for raw ISI
1 / nrow(test) * sum((test$sqISI^2 - m1$fitted.values^2)^2)

# 1000 bootstrap samples for each coefficient
bootCoefs  <- t(replicate(B, {
  yb <- fitted(m1) + resid(m1)[sample(nrow(test), replace = TRUE)]
  boot <- model.matrix(m1)
  coef(lm(yb ~ boot - 1))
}))

# empirical CI
t(apply(bootCoefs, 2, quantile, c(.025, .5, .975)))
# empirical p-values
apply(bootCoefs < 0, 2, sum) / B
```