A Study of Forest Fires: Prediction of the Initial Spread Index

Harrison Friedman, Wanxin Qi , Fuqiang Wang, Janet Zhou

Abstract

Forest fires are a significant environmental issue in the world. ISI, a fire behaviour index, could be predicted by some realistic factors related to fires. In this study, we tried to use rain, relative humidity, temperature, wind, and other possible factors provided in the dataset as covariates to set up a multivariable linear regression model to predict ISI. Filtering out the outlier, we divided the collected data into two parts to build the prediction model and verify the validity of the model separately. We created a model based on our chosen covariates to predict the ISI and measured its effectiveness in doing so. In the end, the model we got is acceptable but it could be better. We came up with some further executions to be done to improve the model.

Keywords: Forest fires, ISI, rain, relative humidity, temperature, wind, linear regression

Introduction

Forest fires are a dangerous natural disaster that seriously threaten the forest natural system and even human lives. Meteorological indexes were gathered from Portugal at the time and location of forest fires to provide a basis for predicting forest fires. ISI, the Initial Spread Index indicates fire velocity spread, is a significant factor in the forest fire system. We will try to simulate predicting the spread of a forest fire with the weather indices Cortez and Morais (2007) lists as influences on ISI (Figure 1), as well as a number related to the frequency of forest fires' occurrence each month.

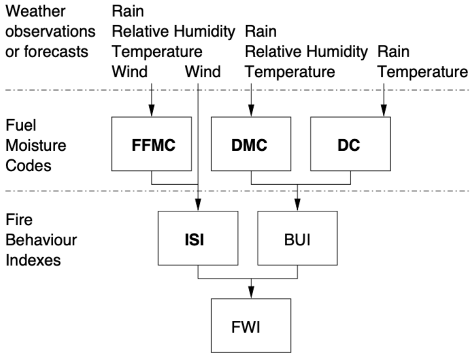


Fig. 1. The Fire Weather Index structure

Background

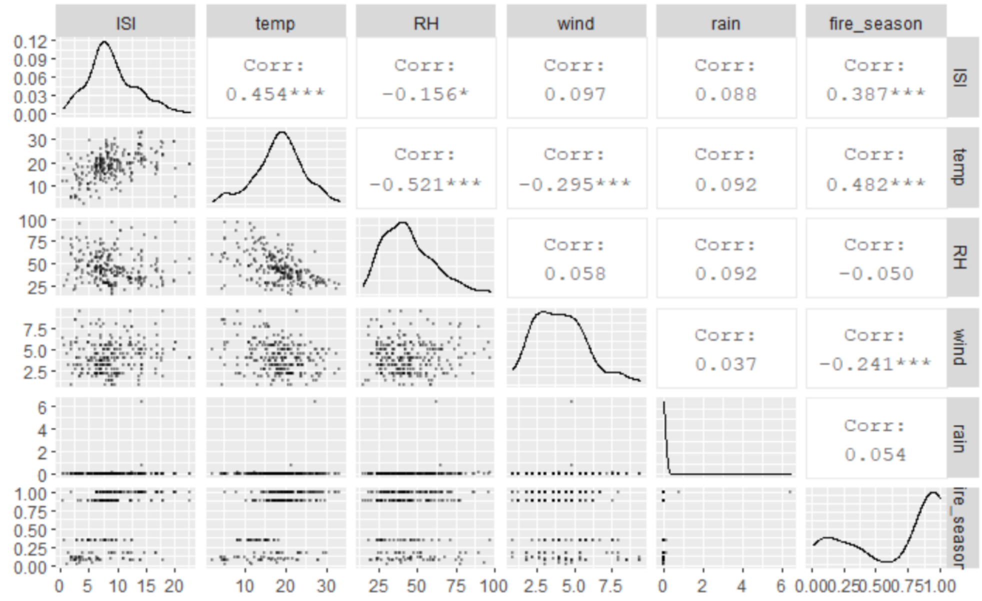
We used a multivariate linear regression model to predict the ISI, the Initial Spread Index, using several weather observations and a calculated index based on the frequency of occurrence of fires each month as covariates. We split the data into two parts with one part used to set up the training model and the other part used to test and verify the prediction conducted by our model. To do this, we randomized the data in the models and then wrote the randomized data into CSVs in order to ensure that we would be working with the same randomized data every time. Our goal is to make the model predict the ISI as accurately as possible.

The 517 data points we used were collected from the Montesinho natural park from the northeast region of Portugal from January 2000 to December 2003. Among the 13 attributes of the forest fires, several features were recorded by the inspector at the time the forest fires occurred, including the six indexes of the FWI system, month of the year, day of the week, location within a 9*9 grid, and the total burned area (in *ha*). The other features were the weather observations collected by the Braganca ̧a Polytechnic Institute, including wind speed (in km/h), temperature (in °C), relative humidity (in %), and accumulated precipitation of rain (in mm/m^2), which were recorded with a 30 minute period by a meteorological station located in the center of the Montesinho park.

Modeling/Analysis

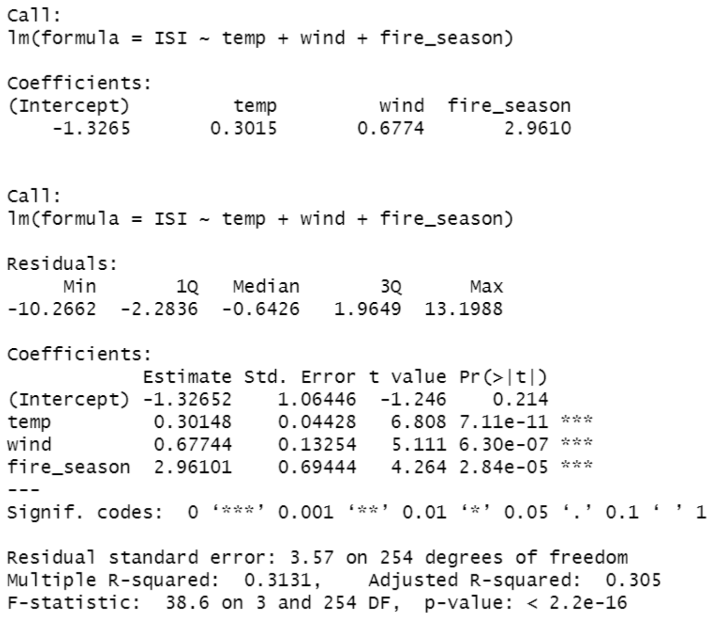
We noticed one large outlier for ISI in the data. All but this point of the ISI fell between 0.0 to 22.7, while the outlier was 56.1. After obtaining the scatter plot and residual plot, it was away from the major group of data, so we **deleted the outlier**[1].

We noticed that about 70% of our observations were in August and September, with August having the highest number of forest fires. It seemed that these two months could be considered the fire season for this area, and therefore the frequency of forest fires may be helpful in predicting its ISI. We accounted for this by making a new covariate called **fire_season**[2], which took the number of forest fires in the month of the data point and divided it by 91, which was the highest number of forest fires (in August) in the training dataset. Although there was technically no need to divide by 91 (since our prediction would be the exact same had we left it as-is) we decided to do it to have all the numbers end up between 0 and 1, which made the numbers easier to digest. The closer a number to 1, the higher the likelihood of a fire occurring in that month versus a different month.



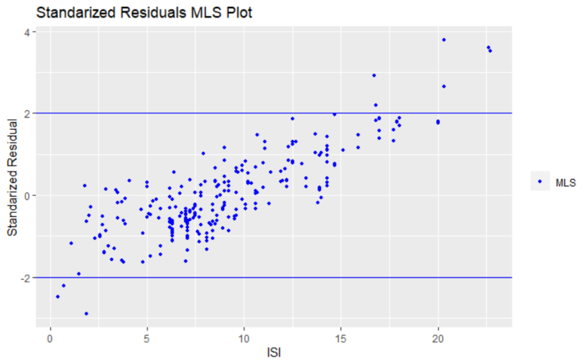
Within the **correlation matrix**[3], we observed a high positive correlation between ISI and temperature as 0.454 and between ISI and fire season index as 0.387. RH is somewhat negatively correlated with ISI with a correlation of -0.156. Wind and rain have a relatively low correlation with ISI. Fortunately wind and temp have normal graphs, and temp especially looks to be mostly normally distributed.

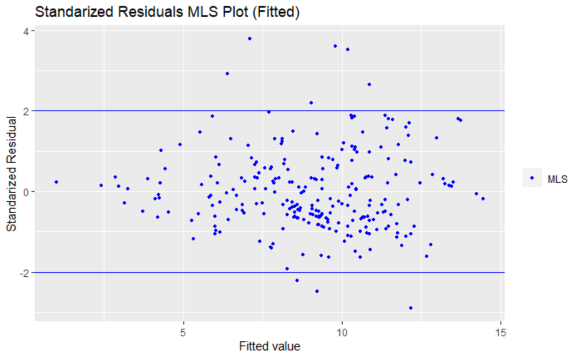
In the process of trying several combinations of covariates, we noticed that rain had very few observations--only 8 out of the 517 were not zero. We tried several transformations of the rain variable such as ln(rain+1). We also tried making it into a categorical variable, either it rained or it didn't, but none of them made rain significant in predicting ISI. Along with having a low correlation with ISI, it seemed that rain would not be helpful to predict ISI. Thus, we left the rain variable out of our prediction. Also, before adding fire season to the model, relative humidity was just significant enough in the model (p-value just below 0.05). However, after adding the fire season covariate to the model, relative humidity ended up having a p-value higher than 0.05, making it insignificant, so we removed it from the model. Finally, the relatively **best** model we could get is: **ISI ~ temp + wind + fire_season**



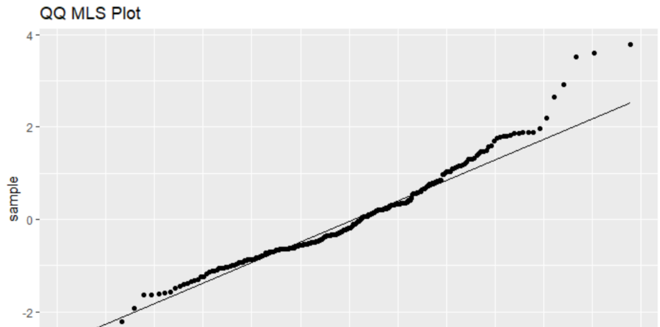
```
Call:
lm(formula = ISI ~ temp + wind + fire_season)

Coefficients:
(Intercept)         temp         wind  fire_season  
    -1.3265       0.3015       0.6774       2.9610  


Call:
lm(formula = ISI ~ temp + wind + fire_season)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.2662  -2.2836  -0.6426   1.9649  13.1988 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.32652    1.06446  -1.246    0.214    
temp         0.30148    0.04428   6.808 7.11e-11 ***
wind         0.67744    0.13254   5.111 6.30e-07 ***
fire_season  2.96101    0.69444   4.264 2.84e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.57 on 254 degrees of freedom
Multiple R-squared:  0.3131,	Adjusted R-squared:  0.305 
F-statistic:  38.6 on 3 and 254 DF,  p-value: < 2.2e-16
```

We see in this **summary of our model**[4] that temp definitely seems to have a significant relationship with ISI, having a p-value far below 0.05. Wind has a significant relationship as well, even though its t-value isn't as high and its p-value isn't as small. The F-statistic for our model is definitely significant, with a value of 38.6, and its p-value is far below 0.05. This implies that the multiple linear relationship is, in fact, significant, and that the effect of these three covariates does explain at least some of ISI. Our R^2 is 0.3131, and while that explains a little less than a third of the variation, it's still pretty good, especially compared to other models that we ran.

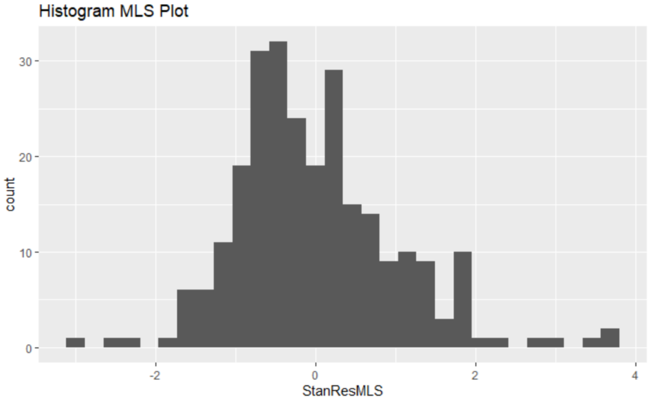

Standarized Residuals MLS Plot


Standarized Residuals MLS Plot (Fitted)

We plotted standardized residuals for our **regression plot**[5]. We have a clear pattern, as it looks like the residuals all fall in a linear fashion. That means that we are not seeing random error. That indicates that we have not captured the entire deterministic component of ISI. However, the **fitted plot**[6] looks far better. While there is a small concentration near the middle that makes us realize that it's not entirely random, they are concentrated on a standard residual of zero, so our predictions seem to be correct on average. Overall, there is no clear pattern to the behavior here, we have very few outliers, and therefore our fitted residuals do seem to be relatively random, indicating that our variables should be predictive and that we have homoscedasticity.


QQ MLS Plot

Our **QQ plot**[7] is about what we'd expect. Our errors are relatively normally distributed for the first standard deviation in both directions, but afterwards, we do have a small amount of variance, but not a significant amount of variance (outside of about 5 points). While our linear relationship is less certain on those points, the plot is nevertheless approximately linear.



We plotted the frequency of the standardized residuals in a **histogram**[8]. The purpose of this was to check that they were approximately normal, which they fortunately appear to be.

Prediction

MSE for training data: **12.54552**[9]

MSE for validation data: **11.50595**[10]

Relative MSE: **0.1239235**[11]

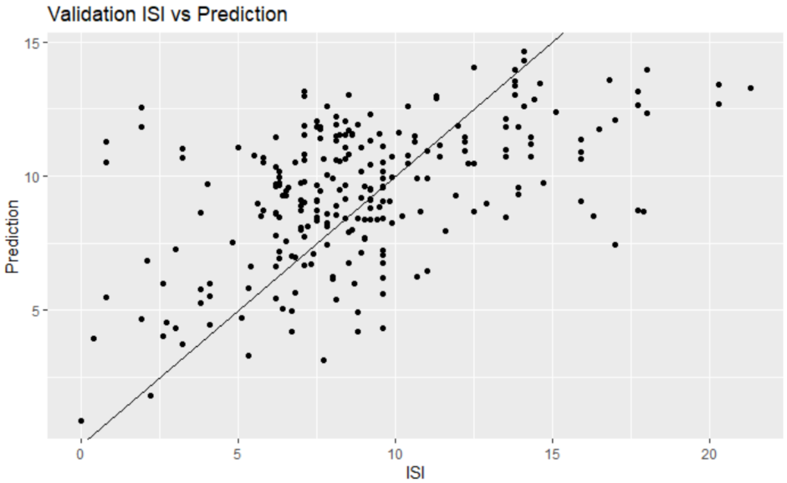
Root Validation MSE: **3.392042**[12]

Data validation ISI variance: **14.74639**[12]

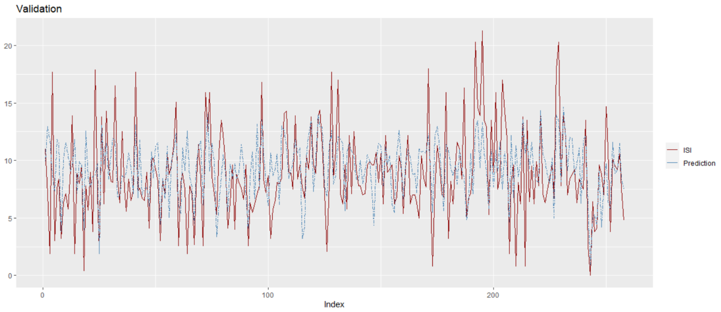
Variance of Predicted ISI: **11.29663**[12]

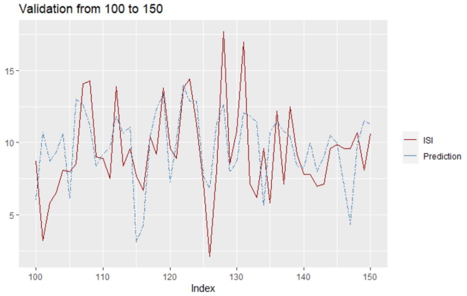
First, we believe that it is unlikely that we overfit the model, since our validation MSE is below our training MSE. Second, with a root validation MSE of 3.39, we know that our average error was a little more than 3 points off the true ISI. Analyzing the reason for this difference, we start by looking at the prediction chart below. We see that we had a tendency to underestimate larger ISI and overestimate smaller ISI. This is backed up by observations about the training data in the QQ plot chart above. However, we were relatively good at predicting ISI in the middle (again backed up by the QQ plot). Also the variance of the true ISI is 3.44 points higher than the variance of our predicted ISI. Taking all this into account, we believe that, especially due to the outliers clearly seen in the QQ plot, our model is a relatively good predictor of ISI in general but relatively poor at predicting extremely high ISI or extremely low ISI. We would have to take further study into the types of fires caused by high ISI, medium ISI, and low ISI to see how viable our model would be in the real world, and whether a blind spot in these specific situations would have extreme consequences. However, since most of the fires in the dataset did have an ISI in the 5 to 10 range, then if these are the norm, we believe that our model would be relatively effective at predicting most fires.

In addition, the model we have is much improved over previous versions. Before we added the fire season covariate, we had a training MSE of 13.165 and a validation MSE of 12.509. We have methodically reduced this MSE to get what we have now by trying different combinations of transforming the weather covariates and adding the fire season covariate. Additionally, we also tried other methods (listed in the appendix under LASSO), but none were quite as good. While we would like a smaller MSE to indicate that our model would be especially predictive, we did the best we could by looking at the data creatively and trying every angle. This model had the lowest MSE, so it is the best predictor of ISI that we found. To see other less-effective methods that we tried, please refer to the Discussion section.



We **validated the model**[13] on the other section of data. Fortunately, the predicted values line seems to successfully capture the trend in the data. There seems to be small groups of variance to the top right of the graph, but otherwise most points fall close to the line. Most of the predictive numbers are, at the very least, close to the slope line.





These **validation graphs**[14] indicate that our model is predictive as some parts are overlapped. Figure # is an example of what our ISI actually looks like for the validation data versus our prediction. Our prediction is actually very good at capturing many of the sharp swings, even though it doesn't do it everywhere. Overall it looks like ISI is predicted relatively well for the data.

Conclusion

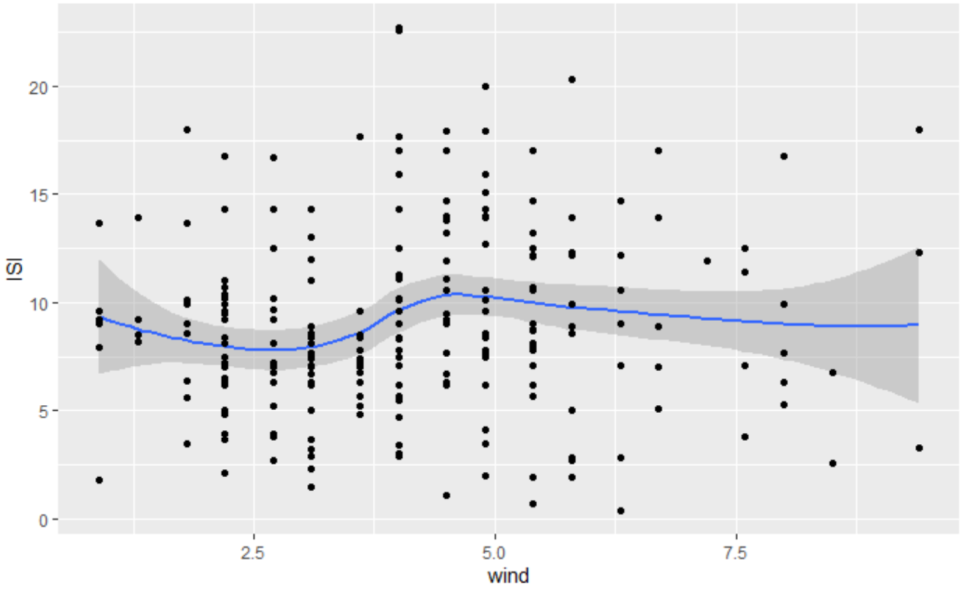
We achieved our goal in some sense by figuring out the influential factors of ISI. The frequency of forest fires each month helped us very much in predicting ISI, especially in the cases that some ISI shared the same values while the weather measurements varied. The three covariates temp, wind, and fire_season were all significant, and the results of R^2 and F-test were relatively good. However, from a real perspective, rain and relative humidity should affect ISI. As we mentioned in the previous part, we tried several ways to include them in our model, but we couldn't. It may be because the data is not very good, or the data volume is not big enough. We explore this and more in our discussion section.

Discussion (for future models and other attempts at finding the right model)

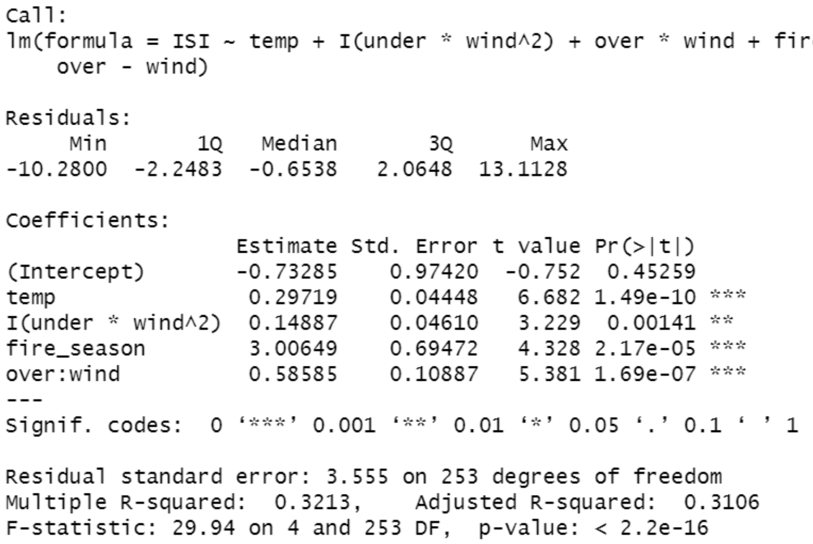
Data Problems

We found something very concerning with the data, which we believe makes up much of the error in the model. As briefly mentioned in the conclusion, we had an issue with the way that the weather covariates were observed. After exploring the Canadian Fire Weather Index, from which the ISI was calculated, we realized that if there was more than one fire on one day, and if they occurred in different X and Y coordinates in the park, then the wind, rain, RH, and temperature would all be measured on that day in that area of the park. However, ISI (along with the other meteorological indices) was calculated based on a complex equation that included previous days, and as such, was calculated only once per day. Because of this, we frequently would see the ISI (and the other indices) be the exact same for several different versions of the weather covariates. This led to duplicates in nearly 400 observations in the data. We hypothesize that this is somewhat responsible for the grouping in the middle.

Wind Separation

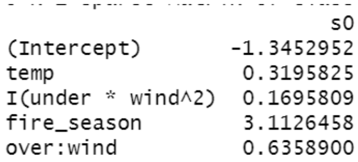


Before we put the data into the regression model, we linearly plotted all the graphs of the relationships between the 3 covariates and ISI. The above graph plots our variable "wind" against our response variable "ISI". After discussing it in the group, we found a better way to modify the model in the future: by separating the variable "wind" into two parts. According to the graph above, the first part of wind (wind less than 4.8) looks like a quadratic relationship, while the second part (wind greater than 4.8) looks like a linear relationship. In this way, we created two indexes for the whole dataset in order to separate the variable wind : the "under" and "over". For the "under", it is equal to 1 when the wind is less than 4.8 and 0 when wind is greater than 4.8. For the "over", it's equal to 1 when the wind is greater than 4.8 and 0 when the wind is less than 4.8 (See the codes for this in Appendix). After creating the indexes mentioned above, we modified our models to ISI = temp + under*wind^2 + over*wind+ fire_season.



```
Call:
lm(formula = ISI ~ temp + I(under * wind^2) + over * wind + fire_season -
    over - wind)

Residuals:
     Min       1Q   Median       3Q      Max
-10.2800  -2.2483  -0.6538   2.0648  13.1128

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -0.73285    0.97420  -0.752  0.45259
temp               0.29719    0.04448   6.682 1.49e-10 ***
I(under * wind^2)  0.14887    0.04610   3.229  0.00141 **
fire_season        3.00649    0.69472   4.328 2.17e-05 ***
over:wind          0.58585    0.10887   5.381 1.69e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.555 on 253 degrees of freedom
Multiple R-squared:  0.3213,    Adjusted R-squared:  0.3106
F-statistic: 29.94 on 4 and 253 DF,  p-value: < 2.2e-16
```

We ran the model and got a training MSE of 12.39628 and a validation MSE of 13.55364. That, unfortunately, implied overfitting--and since the “under” part of the model had a larger p-value than the others, we figured that might be part of the problem. However, removing it wouldn’t make much sense, as it was a key part of the possible solution. We tried to find better coefficients using LASSO (which is discussed for the main model below), and that ended up working better. Here are the LASSO coefficients:



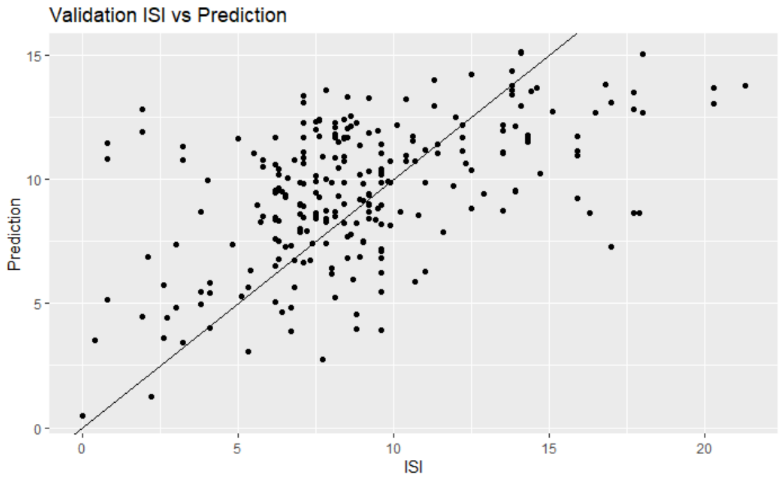
```
                          s0
(Intercept)       -1.3452952
temp               0.3195825
I(under * wind^2)  0.1695809
fire_season        3.1126458
over:wind          0.6358900
```

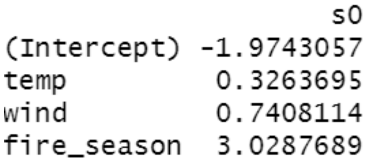
After running the new model, we got a training MSE of 12.4389 and a validation MSE of 11.82203. That was actually one of the best models we got, although not the best. What this does imply is that the idea of separating wind does have some strong merit. Because of that, we’re willing to explore it in the future.

LASSO

One idea we had was to use LASSO to find the best coefficients for our model. We ran LASSO on the full model and got the best results using temp, wind, and fire_season. We used the coefficients for those 3 (as well as the intercept) to create our own model and predict the data. There may have been a simpler method, but we were able to recreate the same charts for LASSO that we had above for the linear model. When we ran the LASSO, the training MSE, validation MSE, and RMSE were very similar to the LM model, but the LM model was about 0.3 more effective at predicting the data. If we had gotten better results, we would have used LASSO as the main model in the project, but since the results were (a little) worse, we decided to only mention it here instead. It should be noted just how similar the prediction of ISI is to the linear model’s prediction, as well as how similar the coefficients are.

The predicted ISI from the LASSO coefficients: Best coefficients from LASSO:



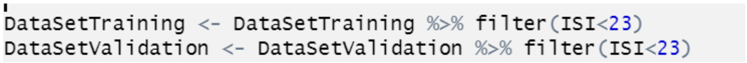


```
                     s0
(Intercept) -1.9743057
temp         0.3263695
wind         0.7408114
fire_season  3.0287689
```

We also ran several other types of cross-validation such as k-folds, random forest, and SVM, but we never got better results than when we used LM. In fact, when we used k-folds with LM as the model, we got the same coefficients that we got from our LM model (which makes sense). Trying all these methods and being unable to improve on the original model made us confident that the model we had was the best one we were going to get.

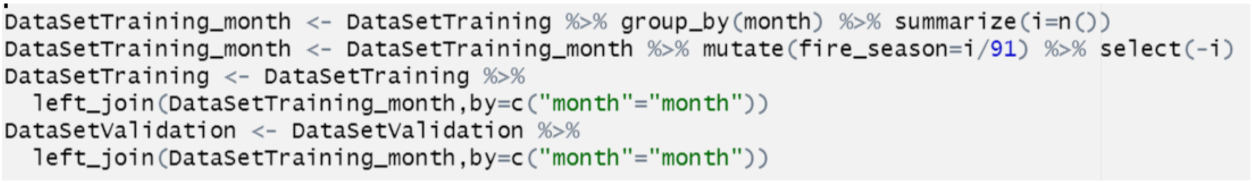
Appendix

1. Removing the outlier:



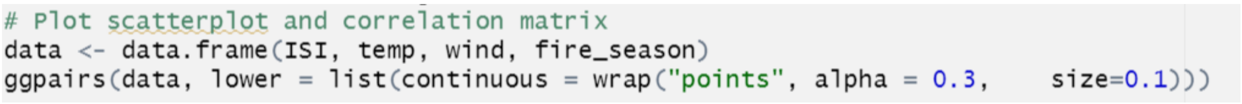
```
DataSetTraining <- DataSetTraining %>% filter(ISI<23)
DataSetValidation <- DataSetValidation %>% filter(ISI<23)
```

2. Fire_season code:



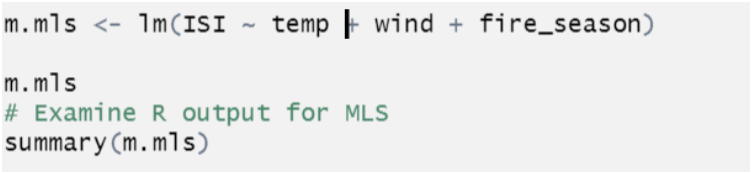
```
DataSetTraining_month <- DataSetTraining %>% group_by(month) %>% summarize(i=n())
DataSetTraining_month <- DataSetTraining_month %>% mutate(fire_season=i/91) %>% select(-i)
DataSetTraining <- DataSetTraining %>%
  left_join(DataSetTraining_month,by=c("month"="month"))
DataSetValidation <- DataSetValidation %>%
  left_join(DataSetTraining_month,by=c("month"="month"))
```

3. Correlation Matrix Code:



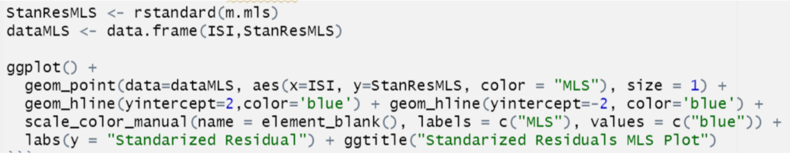
```
# Plot scatterplot and correlation matrix
data <- data.frame(ISI, temp, wind, fire_season)
ggpairs(data, lower = list(continuous = wrap("points", alpha = 0.3,    size=0.1)))
```

4. Summary Code:



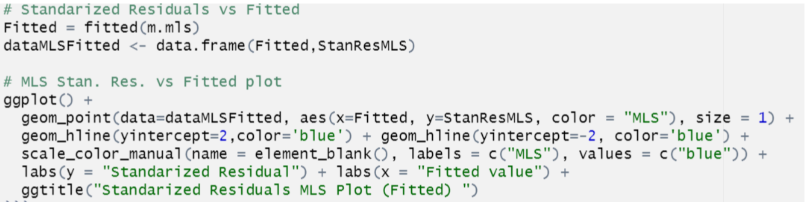
```
m.mls <- lm(ISI ~ temp + wind + fire_season)

m.mls
# Examine R output for MLS
summary(m.mls)
```

5. Standard residual plot code:



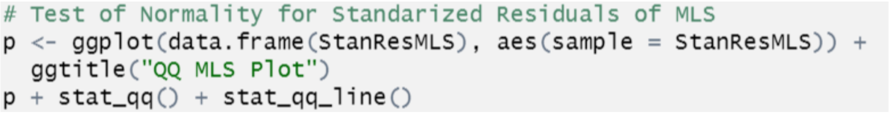
```
StanResMLS <- rstandard(m.mls)
dataMLS <- data.frame(ISI,StanResMLS)

ggplot() +
  geom_point(data=dataMLS, aes(x=ISI, y=StanResMLS, color = "MLS"), size = 1) +
  geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
  scale_color_manual(name = element_blank(), labels = c("MLS"), values = c("blue")) +
  labs(y = "Standarized Residual") + ggtitle("Standarized Residuals MLS Plot")
```

6. Fitted residual plot code:



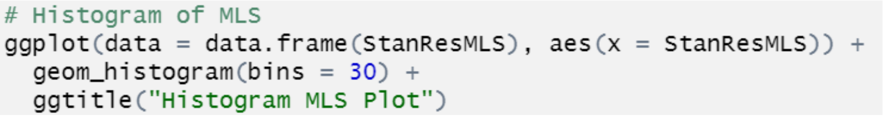
```
# Standarized Residuals vs Fitted
Fitted = fitted(m.mls)
dataMLSFitted <- data.frame(Fitted,StanResMLS)

# MLS Stan. Res. vs Fitted plot
ggplot() +
  geom_point(data=dataMLSFitted, aes(x=Fitted, y=StanResMLS, color = "MLS"), size = 1) +
  geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
  scale_color_manual(name = element_blank(), labels = c("MLS"), values = c("blue")) +
  labs(y = "Standarized Residual") + labs(x = "Fitted value") +
  ggtitle("Standarized Residuals MLS Plot (Fitted) ")
```

7. QQ plot code:



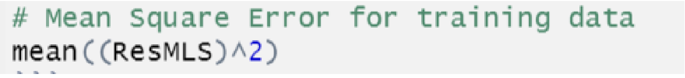
```
# Test of Normality for Standarized Residuals of MLS
p <- ggplot(data.frame(StanResMLS), aes(sample = StanResMLS)) +
  ggtitle("QQ MLS Plot")
p + stat_qq() + stat_qq_line()
```

8. Histogram code:



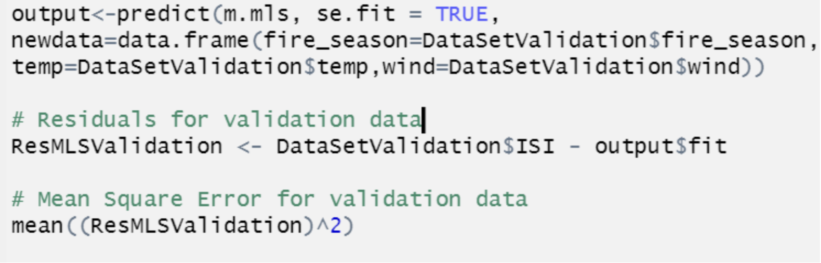
```
# Histogram of MLS
ggplot(data = data.frame(StanResMLS), aes(x = StanResMLS)) +
  geom_histogram(bins = 30) +
  ggtitle("Histogram MLS Plot")
```

9. Training MSE code:



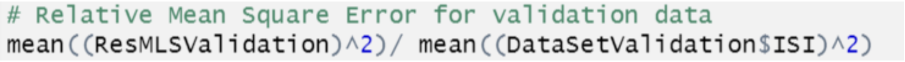
```
# Mean Square Error for training data
mean((ResMLS)^2)
```

10. Validation MSE code:



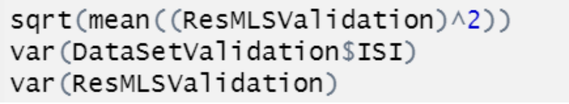
```
output<-predict(m.mls, se.fit = TRUE,
newdata=data.frame(fire_season=DataSetValidation$fire_season,
temp=DataSetValidation$temp,wind=DataSetValidation$wind))

# Residuals for validation data
ResMLSValidation <- DataSetValidation$ISI - output$fit

# Mean Square Error for validation data
mean((ResMLSValidation)^2)
```

11. Relative MSE:



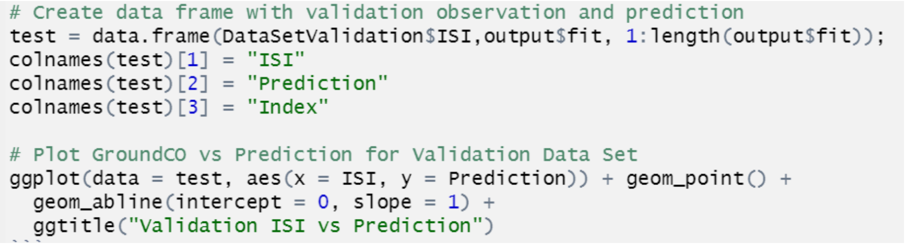
```
# Relative Mean Square Error for validation data
mean((ResMLSValidation)^2)/ mean((DataSetValidation$ISI)^2)
```

12. Root MSE and variance of each ISI:



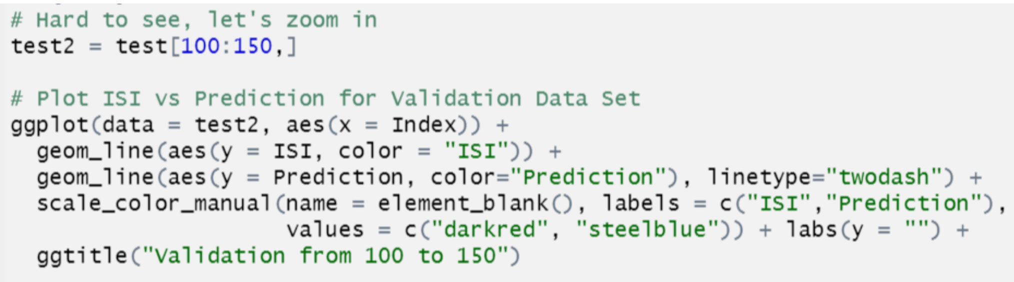
```
sqrt(mean((ResMLSValidation)^2))
var(DataSetValidation$ISI)
var(ResMLSValidation)
```

13. Validation plot code:



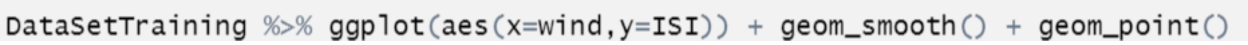
```
# Create data frame with validation observation and prediction
test = data.frame(DataSetValidation$ISI,output$fit, 1:length(output$fit));
colnames(test)[1] = "ISI"
colnames(test)[2] = "Prediction"
colnames(test)[3] = "Index"

# Plot GroundCO vs Prediction for Validation Data Set
ggplot(data = test, aes(x = ISI, y = Prediction)) + geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  ggtitle("Validation ISI vs Prediction")
```

14. Snippet of validation data vs. predicted ISI (from 100:150):



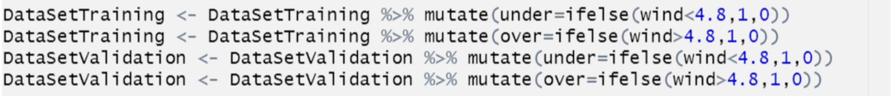
```
# Hard to see, let's zoom in
test2 = test[100:150,]

# Plot ISI vs Prediction for Validation Data Set
ggplot(data = test2, aes(x = Index)) +
  geom_line(aes(y = ISI, color = "ISI")) +
  geom_line(aes(y = Prediction, color="Prediction"), linetype="twodash") +
  scale_color_manual(name = element_blank(), labels = c("ISI","Prediction"),
                     values = c("darkred", "steelblue")) + labs(y = "") +
  ggtitle("Validation from 100 to 150")
```

15. Wind smoothed plot:



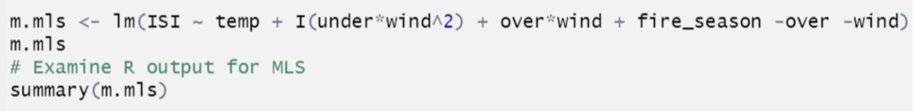
```
DataSetTraining %>% ggplot(aes(x=wind,y=ISI)) + geom_smooth() + geom_point()
```

16. Wind code:



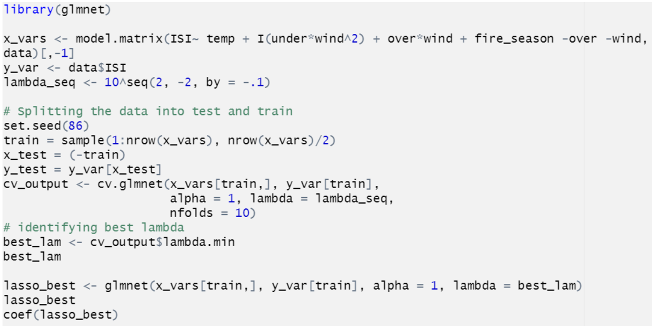
```
DataSetTraining <- DataSetTraining %>% mutate(under=ifelse(wind<4.8,1,0))
DataSetTraining <- DataSetTraining %>% mutate(over=ifelse(wind>4.8,1,0))
DataSetValidation <- DataSetValidation %>% mutate(under=ifelse(wind<4.8,1,0))
DataSetValidation <- DataSetValidation %>% mutate(over=ifelse(wind>4.8,1,0))
```

17. Wind code w/RH:



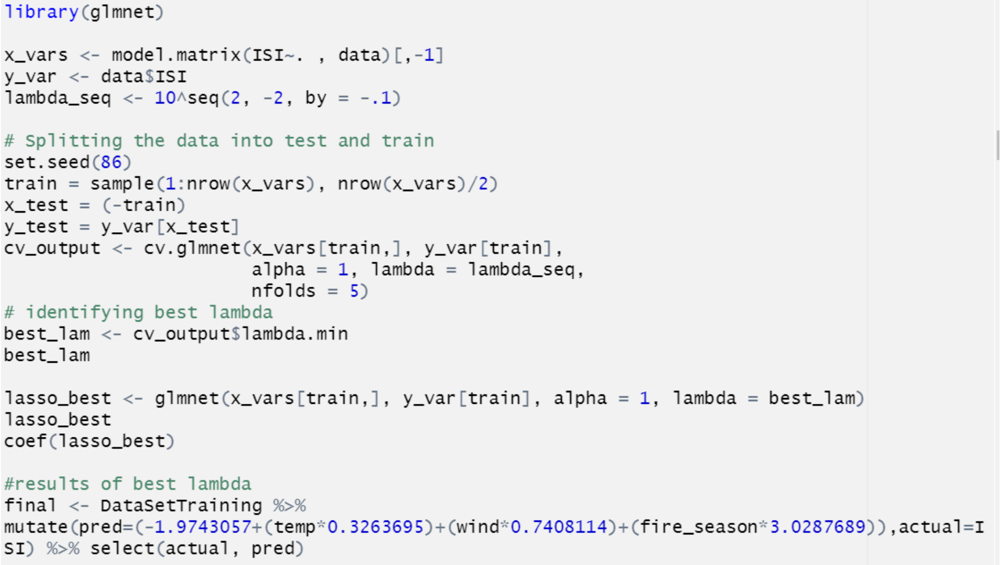
```
m.mls <- lm(ISI ~ temp + I(under*wind^2) + over*wind + fire_season -over -wind)
m.mls
# Examine R output for MLS
summary(m.mls)
```

18. LASSO wind code:



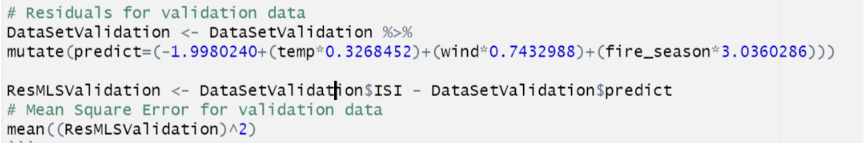
```
library(glmnet)

x_vars <- model.matrix(ISI~ temp + I(under*wind^2) + over*wind + fire_season -over -wind,
data)[,-1]
y_var <- data$ISI
lambda_seq <- 10^seq(2, -2, by = -.1)

# Splitting the data into test and train
set.seed(86)
train = sample(1:nrow(x_vars), nrow(x_vars)/2)
x_test = (-train)
y_test = y_var[x_test]
cv_output <- cv.glmnet(x_vars[train,], y_var[train],
                       alpha = 1, lambda = lambda_seq,
                       nfolds = 10)
# identifying best lambda
best_lam <- cv_output$lambda.min
best_lam

lasso_best <- glmnet(x_vars[train,], y_var[train], alpha = 1, lambda = best_lam)
lasso_best
coef(lasso_best)
```

19. LASSO main model code:



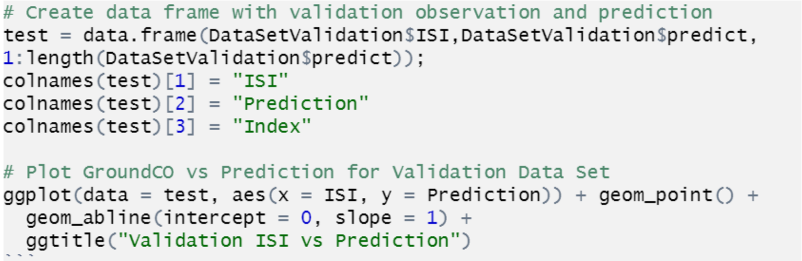
```
library(glmnet)

x_vars <- model.matrix(ISI~. , data)[,-1]
y_var <- data$ISI
lambda_seq <- 10^seq(2, -2, by = -.1)

# Splitting the data into test and train
set.seed(86)
train = sample(1:nrow(x_vars), nrow(x_vars)/2)
x_test = (-train)
y_test = y_var[x_test]
cv_output <- cv.glmnet(x_vars[train,], y_var[train],
                       alpha = 1, lambda = lambda_seq,
                       nfolds = 5)
# identifying best lambda
best_lam <- cv_output$lambda.min
best_lam

lasso_best <- glmnet(x_vars[train,], y_var[train], alpha = 1, lambda = best_lam)
lasso_best
coef(lasso_best)

#results of best lambda
final <- DataSetTraining %>%
mutate(pred=(-1.9743057+(temp*0.3263695)+(wind*0.7408114)+(fire_season*3.0287689)),actual=I
SI) %>% select(actual, pred)
```

20. LASSO main model validation code:



```
# Residuals for validation data
DataSetValidation <- DataSetValidation %>%
mutate(predict=(-1.9980240+(temp*0.3268452)+(wind*0.7432988)+(fire_season*3.0360286)))

ResMLSValidation <- DataSetValidation$ISI - DataSetValidation$predict
# Mean Square Error for validation data
mean((ResMLSValidation)^2)
```

21. LASSO plot code:



```
# Create data frame with validation observation and prediction
test = data.frame(DataSetValidation$ISI,DataSetValidation$predict,
1:length(DataSetValidation$predict));
colnames(test)[1] = "ISI"
colnames(test)[2] = "Prediction"
colnames(test)[3] = "Index"

# Plot GroundCO vs Prediction for Validation Data Set
ggplot(data = test, aes(x = ISI, y = Prediction)) + geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  ggtitle("Validation ISI vs Prediction")
```