Predicting Fire Behavior Using Weather & Meteorological Data

Darshil Desai, Edwin Ramirez

November 27, 2018

2.1 Abstract

2.2 Introduction

Our regressional model will focus on utilizing previous forest fire and weather data sourced from the UCI Machine Learning Repository to predict the Initial Spread Index (ISI) of fires in Portugal and any countries using the Canadian Forest Wildfire Index System (CFWIS). We found this project to be of relevance considering the recent devastating droughts and fires in Northern Europe throughout Greece, Portugal, and Spain within the last year. Additionally, the relevance of the recent California fires also give support to the necessity of predictive analytics in this area. Hence, by utilizing a regressional model, we can predict the early behavior of a fire. Portugal utilizies the Canadian Forest Fire Weather Index System in order to track fuel moisture and wind speed to determine the intensity of a fire. Thus, by predicting the initial spread index of potential fires, we'd be able to predict the danger of a fire based on how quickly it would spread. The ISI scale begins at 0, where a 10 indicates a high rate of spread after ignition, and 16 or higher indicates an extreme rapid rate of spread.

The research question can be hypothesized as follows:

H_{0}: None of the predictor variables in the dataset are useful in making predictions about the Initial Spread Index.

H_{a}: At least one of predictor variables in the dataset is useful in making predictions about the Initial Spread Index.

2.3 Dataset

View Dataset

The UCI dataset comprises of the following 13 variables:

1 & 2. X,Y : coordinates within the Montesinho park. Ranges from 1 to 9
3. month: month when the fire frst occured
4: day: day of the given month when the fire occured
5: FFMC (Fine Fuel Moisture Code) : a numeric rating of the moisture content of litter and other cured fine fuels. This code is an indicator of the relative ease of ignition and the flammability of fine fuel.
6: DMC (Duff Moisture Code): A numeric rating of the average moisture content of loosely compacted organic layers of moderate depth. This code gives an indication of fuel consumption in moderate duff layers and medium-size woody material.[1]
7.Burned area of the forest: area of forest burned
8. temperature
9: RH (relative humidity)
10: wind: wind speed
11: rain
12: DC (Drought Code): A numeric rating of the average moisture content of deep, compact organic layers. This code is a useful indicator of seasonal drought effects on forest fuels and the amount of smoldering in deep duff layers and large logs.
13. ISI (Initial Spread Index): expected rate of fire spread.This variable will be our response variable and we will try and establish a linear relationship between the myriad of weather and meterological factors of the forest experiencing fires and the future expected area burn. [2]

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## corrplot 0.84 loaded
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
##   X Y month day FFMC  DMC    DC  ISI temp RH wind rain area
## 1 7 5   mar fri 86.2 26.2  94.3  5.1  8.2 51  6.7  0.0    0
## 2 7 4   oct tue 90.6 35.4 669.1  6.7 18.0 33  0.9  0.0    0
## 3 7 4   oct sat 90.6 43.7 686.9  6.7 14.6 33  1.3  0.0    0
## 4 8 6   mar fri 91.7 33.3  77.5  9.0  8.3 97  4.0  0.2    0
## 5 8 6   mar sun 89.3 51.3 102.2  9.6 11.4 99  1.8  0.0    0
## 6 8 6   aug sun 92.3 85.3 488.0 14.7 22.2 29  5.4  0.0    0
```

2.4 Methodology

In order to draw a relationship between the various weather, meteorological variables in the dataset and the response variable (Initial Spread Index), it is vital to transform and preprocess our data to succesfully represent this data in a linear relationship.

Our methods to do the same involve the following:
1. Deleting redundant data that will fail to enhance the model's predictive power
2. Representing categorical data (months) as binary numerical data.
3. Transforming our response variable
4. Examine multicolinearity

1. Deleting Redundant Data

Several variables in the given dataset do not assist in creating the linear relationship between with the response variable (Initial Spread Index). Briefly:
a. X,Y: Coordinate data provided by the data can be considered redundant due to the coordintes being confined to the specific area of Montesinho Park in Portugal. This data can be removed because the goal of our model is to predict fires within countires that utilize the Canadian Fire Weather Index System.
b. Days: In isolation, the numerical value of a day does not provide any significant relevance to a datapoint

2. Categorical Data

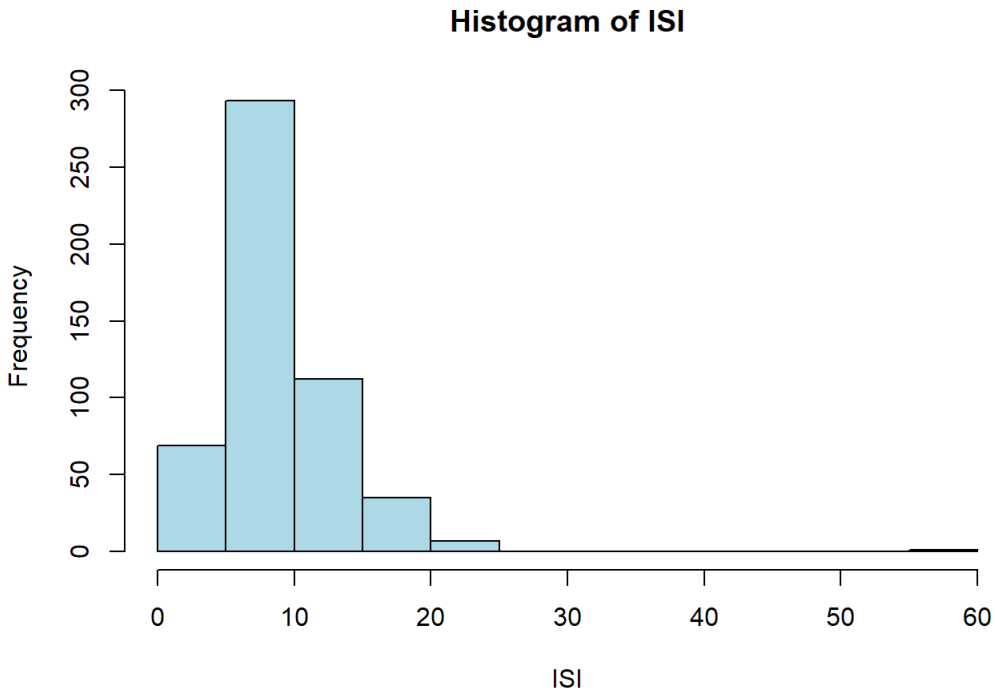
The variable of months can be grouped to create a categorical variable that signifies seasons. Thus, each season will be represented as nominal variables in our regressional model.

```
spring <- c('mar', 'apr', 'may')
summer <- c('jun', 'jul', 'aug')
autumn <- c('sep', 'oct','nov')
winter <- c('dec','jan', 'feb')

#clean data
# Representing months as the appropriate season it is a part of
season = function(x){
  if(x %in% spring){
    return('spring')
  } else if(x %in% summer){
    return('summer')
  } else if(x %in% autumn){
    return('autumn')
  } else{
    return('winter')
  }
}

fire$month = sapply(fire$month, season)

#renaming the month column to "seasons"
colnames(fire)[colnames(fire)=="month"] <- "season_"

#Converting the months column into 4 columns, one for each season
#fire <- dummy.data.frame(fire)
#invisible(get.dummy( fire, 'season_'))
```

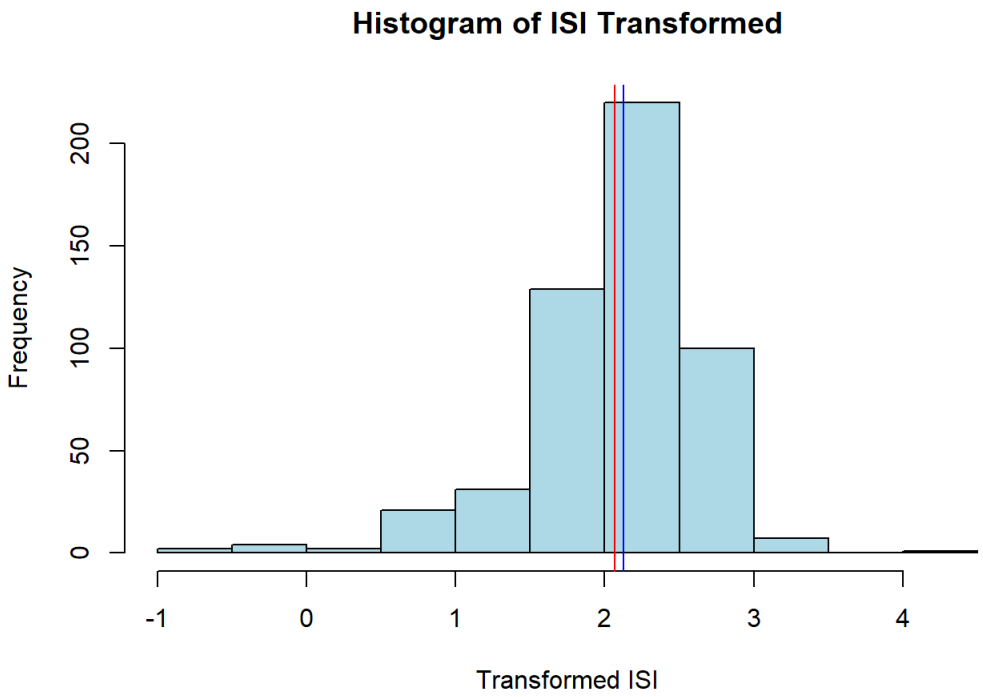
3.Transforming the response variable

The histogram below illustrates the distribution of the response variable as right-skewed. To prevent a normality violation in the regressional model, the data is transformed by taking the natural log of all ISI data.

```
#check normality of response variable
hist(fire$ISI, col = "lightblue", xlab = "ISI", main = "Histogram of ISI")
```



```
#transform data by taking log
trans = log(fire$ISI)

#change  the below case or error occurs
trans[380] = 1.2

hist(trans, xlab = "Transformed ISI", col = "lightblue", main = "Histogram of ISI Transformed")
abline(v = mean(trans), col = 'red')
abline(v = median(trans), col = 'blue')
```

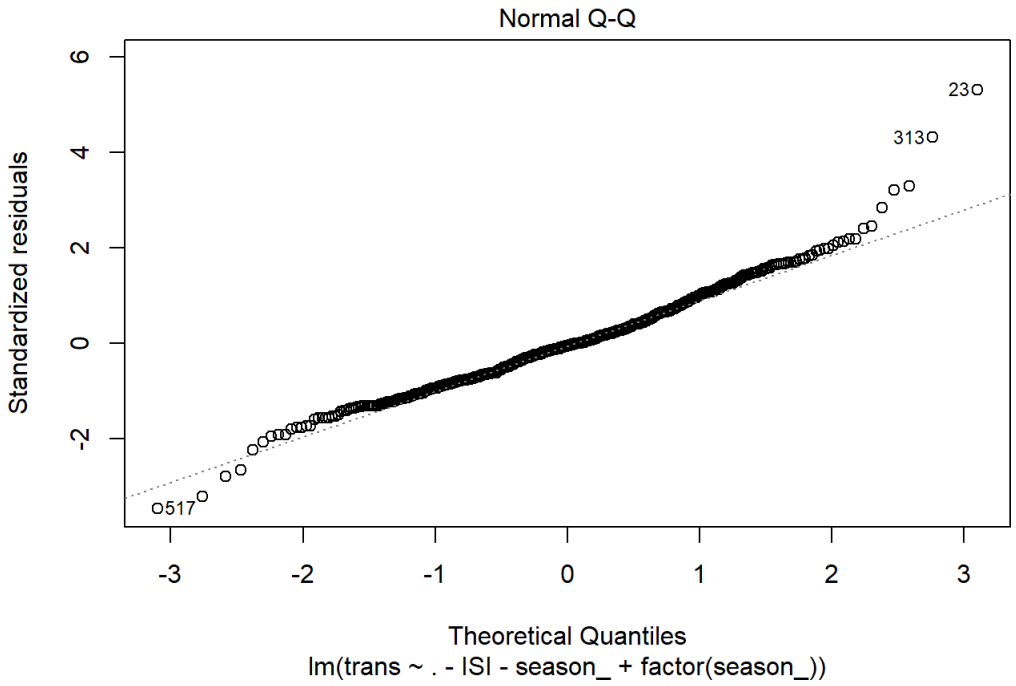


Consequentially, the distribution the response variable (ISI) is now normal, and we can proceed with selecting the appropriate features for our model. A QQ-normal plot is shown below to illustrate that the residual data is normally distributed.

```
#png("qqnorm.png", units = "px", width=960, height=960)
#use QQ plot check model
summary(lm(trans ~.-ISI - season_ + factor(season_) , data = fire))
```

```
##
## Call:
## lm(formula = trans ~ . - ISI - season_ + factor(season_), data = fire)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88416 -0.18826 -0.01478  0.15555  1.41482
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -7.5020503  0.3253664 -23.057  < 2e-16 ***
## FFMC                    0.1032423  0.0031915  32.349  < 2e-16 ***
## DMC                    -0.0002699  0.0003285  -0.821 0.411758
## DC                     -0.0001727  0.0001446  -1.194 0.232890
## temp                    0.0033244  0.0039828   0.835 0.404288
## RH                      0.0023911  0.0010969   2.180 0.029735 *
## wind                    0.0399209  0.0070624   5.653 2.65e-08 ***
## rain                   -0.0281188  0.0414902  -0.678 0.498258
## area                   -0.0000836  0.0001897  -0.441 0.659678
## factor(season_)spring -0.1859825  0.0955907  -1.946 0.052256 .
## factor(season_)summer  0.0844774  0.0385585   2.191 0.028916 *
## factor(season_)winter -0.3526551  0.0977276  -3.609 0.000339 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2712 on 505 degrees of freedom
## Multiple R-squared:  0.7781, Adjusted R-squared:  0.7732
## F-statistic: 160.9 on 11 and 505 DF,  p-value: < 2.2e-16
```

```
plot(lm(trans ~.-ISI - season_ + factor(season_), data = fire),2)
```




```
#dev.off()
```

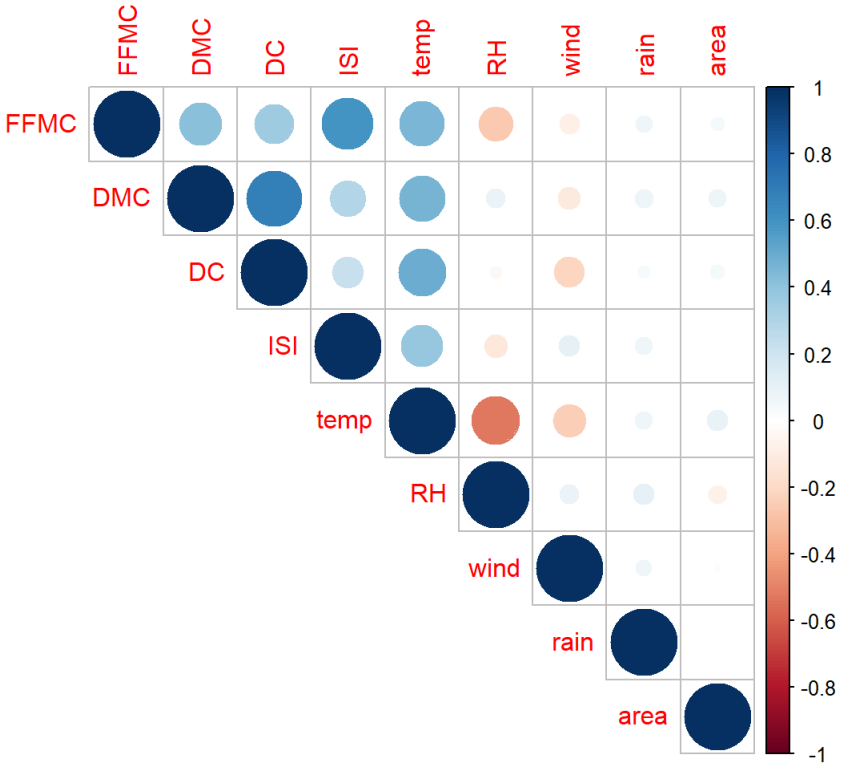
4.Examining Multicolinearity

Multicolinearity occurs when predictor variables are linearly correlated with the other. This implies that a change in any one of the predictor variables would entail a change in another highly correlated predictor variable.

Before we proceed to fit our model it is important to perform two vital checks. Firstly, there needs to be a reasonable correlation between the predictor variables and the response variable. An absence of any linear pattern does not warrant the use of a linear regression problem.

Colinearity Between Predictor Variables | Correlation Between Predictor and Response Variable

```
#removing seasons for correlation
fire_cor <- fire[,-match(c('season_'),names(fire))]

#Checking for multicolinearity. Here we will compare each of our predictor variables with all the o
thers predictor variables.
corr<-cor(fire_cor)
#plotting correlation head map
#png("corrplot.png", units = "px", width=960, height=960)
corrplot(corr, type = 'upper')
```




```
#dev.off()
```

2.5 Results & Discussion

After analyzing the correlation of the predictor variables in the dataset, the next steps comprise the following:
1. Feature Selection
2. Model fitting using Training Data
3. Analysis

1.Feature Selection

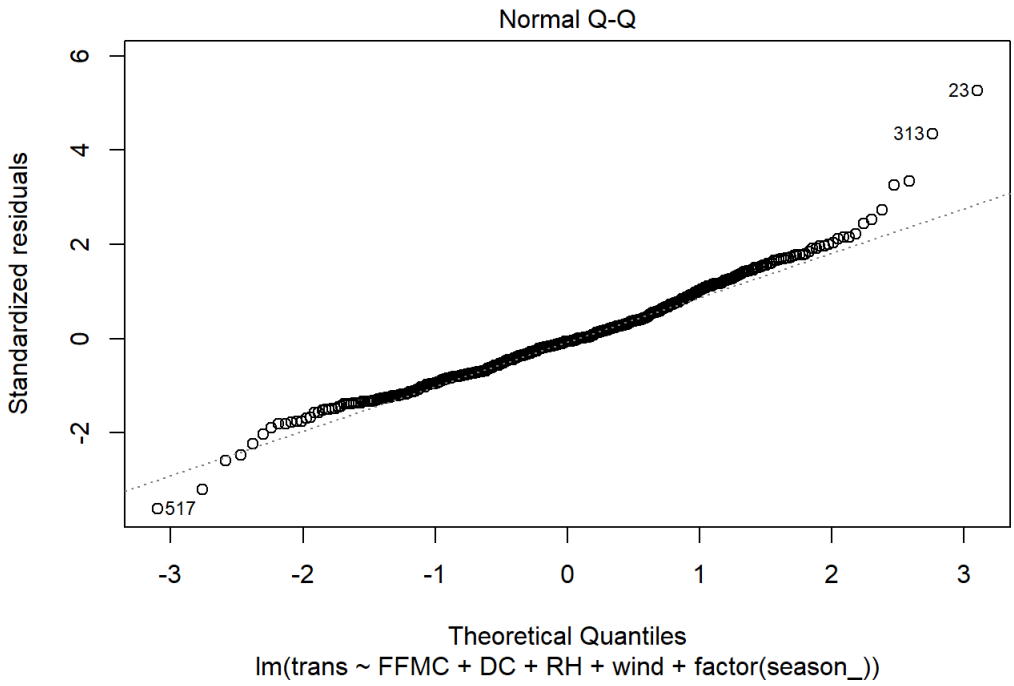
Using stepwise regression, we iterate through different possibilities of a linear model and choose one with features yileding the best (lowest) error metric.

The QQ-normal plot below illustrates the distribution of our data after utilizing stepwise regression for feature selection.

```
# forward stepwise regression
model = lm(trans ~.-ISI - season_ + factor(season_), data = fire)
new_model = step(model)
```

```
## Start:  AIC=-1337.49
## trans ~ (season_ + FFMC + DMC + DC + ISI + temp + RH + wind +
##     rain + area) - ISI - season_ + factor(season_)
##
##                   Df Sum of Sq     RSS      AIC
## - area             1     0.014  37.150 -1339.30
## - rain             1     0.034  37.170 -1339.02
## - DMC              1     0.050  37.186 -1338.80
## - temp             1     0.051  37.187 -1338.78
## - DC               1     0.105  37.241 -1338.04
## <none>                          37.136 -1337.49
## - RH               1     0.349  37.486 -1334.65
## - factor(season_)  3     2.223  39.359 -1313.44
## - wind             1     2.350  39.486 -1307.78
## - FFMC             1    76.954 114.091  -759.21
##
## Step:  AIC=-1339.3
## trans ~ FFMC + DMC + DC + temp + RH + wind + rain + factor(season_)
##
##                   Df Sum of Sq     RSS      AIC
## - rain             1     0.033  37.184 -1340.83
## - temp             1     0.048  37.199 -1340.62
## - DMC              1     0.055  37.205 -1340.53
## - DC               1     0.101  37.251 -1339.90
## <none>                          37.150 -1339.30
## - RH               1     0.353  37.504 -1336.41
## - factor(season_)  3     2.249  39.400 -1314.90
## - wind             1     2.338  39.488 -1309.74
## - FFMC             1    76.978 114.129  -761.04
##
## Step:  AIC=-1340.83
## trans ~ FFMC + DMC + DC + temp + RH + wind + factor(season_)
##
##                   Df Sum of Sq     RSS      AIC
## - temp             1     0.037  37.221 -1342.32
## - DMC              1     0.052  37.235 -1342.12
## - DC               1     0.108  37.292 -1341.33
## <none>                          37.184 -1340.83
## - RH               1     0.324  37.508 -1338.34
## - factor(season_)  3     2.315  39.498 -1315.61
## - wind             1     2.309  39.493 -1311.69
## - FFMC             1    77.107 114.291  -762.31
##
## Step:  AIC=-1342.32
## trans ~ FFMC + DMC + DC + RH + wind + factor(season_)
##
##                   Df Sum of Sq     RSS      AIC
## - DMC              1     0.038  37.258 -1343.79
## - DC               1     0.120  37.341 -1342.65
## <none>                          37.221 -1342.32
## - RH               1     0.355  37.576 -1339.41
## - wind             1     2.273  39.494 -1313.68
## - factor(season_)  3     3.918  41.139 -1296.57
## - FFMC             1    77.501 114.722  -762.36
##
## Step:  AIC=-1343.79
## trans ~ FFMC + DC + RH + wind + factor(season_)
##
##                   Df Sum of Sq     RSS      AIC
## <none>                          37.258 -1343.79
## - DC               1     0.319  37.577 -1341.39
## - RH               1     0.323  37.582 -1341.33
## - wind             1     2.298  39.556 -1314.86
## - factor(season_)  3     3.999  41.258 -1297.08
## - FFMC             1    80.155 117.414  -752.37
```

```
#new model QQ plot
#png("qqnorm2.png", units = "px", width=960, height=960)
plot(new_model, 2)
```




```
#dev.off()

new_model$terms
```

```
## trans ~ FFMC + DC + RH + wind + factor(season_)
## attr(,"variables")
## list(trans, FFMC, DC, RH, wind, factor(season_))
## attr(,"factors")
##                 FFMC DC RH wind factor(season_)
## trans              0  0  0    0               0
## FFMC               1  0  0    0               0
## DC                 0  1  0    0               0
## RH                 0  0  1    0               0
## wind               0  0  0    1               0
## factor(season_)    0  0  0    0               1
## attr(,"term.labels")
## [1] "FFMC"            "DC"              "RH"              "wind"
## [5] "factor(season_)"
## attr(,"order")
## [1] 1 1 1 1 1
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 1
## attr(,".Environment")
## <environment: R_GlobalEnv>
## attr(,"predvars")
## list(trans, FFMC, DC, RH, wind, factor(season_))
## attr(,"dataClasses")
##          trans           FFMC             DC             RH
##       "numeric"      "numeric"      "numeric"      "numeric"
##           wind factor(season_)
##       "numeric"       "factor"
```

2. Fitting the model

A portion of the data is separated to be utilized as a training set, while the remaining portion will be utilized as a test set.This will allow the accuracy of the model to be measured.

```
#attaching the trans column to the main fire dataset
fire_to_split <- cbind(fire, trans)

#Get 400 random numbers
ran <- sample(1:517, 400, replace=F)

#splitting the dataset into training and testing
train <- fire_to_split[ran,]

#test data: contains data not in training data
test = setdiff(fire_to_split, train)
```

Next, the training data is fitted to the model with the selected features. After removing features not selected from stepwise regression, the model includes `factor(season_)`, `FFMC`, `temp`, `RH`, and `wind`

```
#fitting the model
m = lm(train$trans ~ FFMC + temp + RH + wind - ISI - trans - season_ + factor(season_), data = trai
n)
summary(m)
```

```
##
## Call:
## lm(formula = train$trans ~ FFMC + temp + RH + wind - ISI - trans -
##     season_ + factor(season_), data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84704 -0.18749 -0.00731  0.15118  1.48802
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -7.141927   0.320495 -22.284  < 2e-16 ***
## FFMC                    0.099134   0.003185  31.122  < 2e-16 ***
## temp                   -0.002822   0.004473  -0.631  0.52846
## RH                      0.001577   0.001226   1.286  0.19920
## wind                    0.038331   0.008077   4.746 2.92e-06 ***
## factor(season_)spring -0.083923   0.053762  -1.561  0.11933
## factor(season_)summer  0.150126   0.033472   4.485 9.58e-06 ***
## factor(season_)winter -0.236693   0.076680  -3.087  0.00217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2716 on 392 degrees of freedom
## Multiple R-squared:  0.7912, Adjusted R-squared:  0.7874
## F-statistic: 212.1 on 7 and 392 DF,  p-value: < 2.2e-16
```

3. Analysis

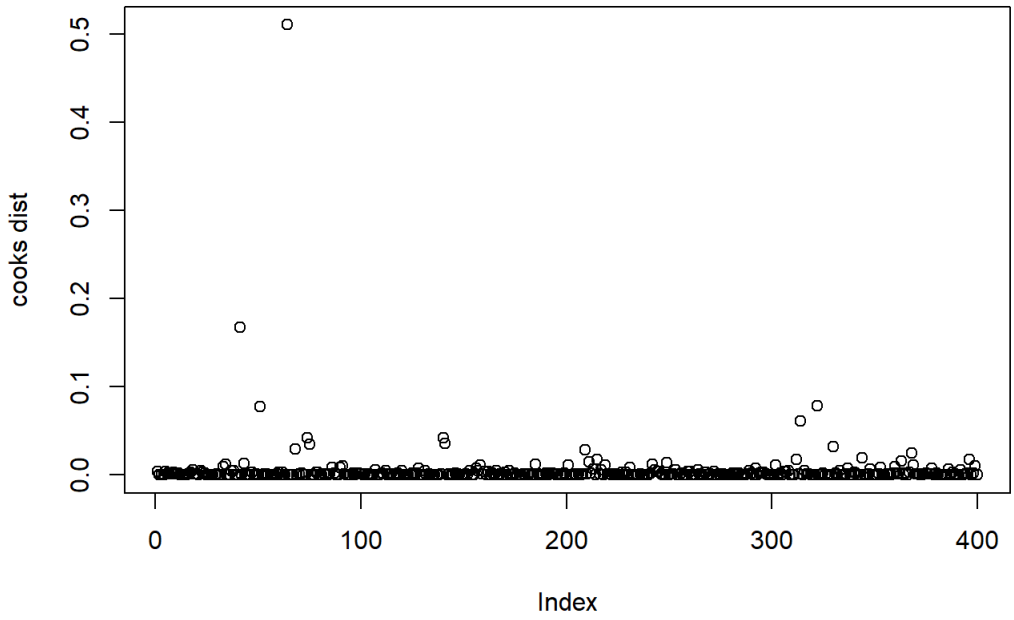
The following conclusions are the results of our model: - The overall P-value of the model is <2.2e-16. Being far below the indsutry standard of 5% significance level, it can be concluded that at least one of the features is useful in predicting the response variable (ISI). Therefore the null hypothesis \(H_{0}\) is rejected. - The model yields an r-squared value of 0.76, indicating that it would likely be moderately useful in predicting the initial rate of spread of a fire.

4. Cooks distance

After fitting the model, cook's distance is utilized to analyze if any influential data points affect the regression line. If any exist, this decreases the model's ability to generalize.

Based on the plot below, it can be confirmed that the training dataset one outlier point (>1). This outlier point is also influential in affecting the regression line of the model

```
#cooks distance
library(stats)
cook = cooks.distance(m)    # here m is the linear mode

#png("cook.png", units = "px", width=960, height=960)
plot(cook, ylab = 'cooks dist')
```




```
#dev.off()
```

5. Testing the model

The model can be tested, and resulting mean squared error support our conclusion that the model is moderately useful.

```
sum_square_errors <- sum((test$trans - predict(m,newdata = test))^2)
sum_square_errors
```

```
## [1] 9.046336
```

```
## Code archives

week = function(x){
  if(x == 'mon'){
    return(1)
  } else if(x == 'tue'){
    return(2)
  } else if(x == 'wed'){
    return(3)
  } else if(x == 'thu'){
    return(4)
  } else if(x == 'fri'){
    return(5)
  } else if(x == 'sat'){
    return(6)
  } else{
    return(7)
  }
}


#check correlation and distributions
#pairs.panels(fire)


#summary(lm(trans ~ month + FFMC + temp + RH + wind, data = fire))
#displaying first 5 rows
```

Footnotes:

1. http://cwfis.cfs.nrcan.gc.ca/background/summary/fwi ↩
2. http://cwfis.cfs.nrcan.gc.ca/background/summary/fwi ↩