Wildfire Analysis in U.S.

Stat Methods and App I - CU Boulder

Xingyu Chen

December 05, 2021

Contents

List of Figures

1 Introduction

1.1 Motivation

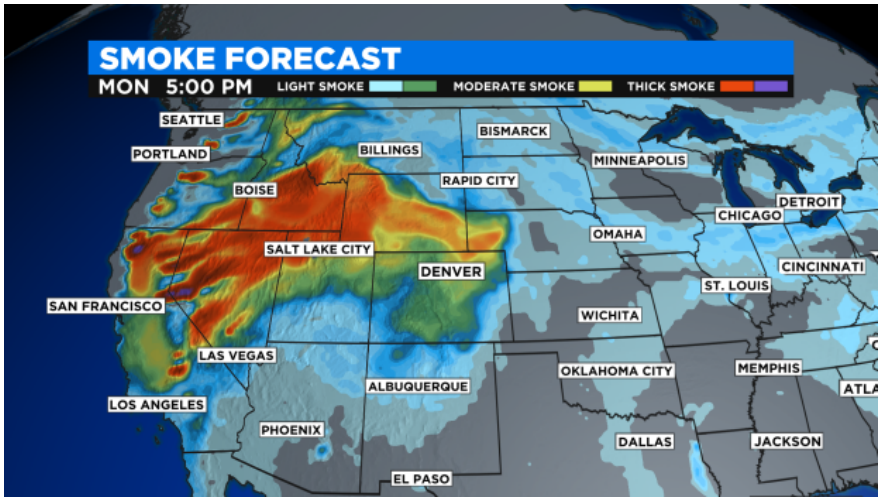


Figure 1: Colorado's Air Quality is Pretty Bad Today and Will Get Worse

Wildfires, like many other natural disasters, demand everyones' attention. From being a direct threat to the constant reminder of a smoke caused haze. The smoke from the California wildfires in 2021 has massively impacted the air quality of Colorado State.

The noticeable air pollution reached such levels that the state government recommended active children and adults reduce prolonged or heavy outdoor activities. This impact helped inspire this report.

The general motivation behind this project is to understand the history of wildfires in the U.S., see how they've changed over time and understand when and where they are most severe. The following questions are what this report will specifically try to answer.

1. Have the number of wildfires increased over time? Have the fires that occur become more severe?

2. During which time of the year is there the most wildfire activity?

3. Which states have the most wildfire activity? Of the top state, which counties had the most wildfires activity?

This is a continuous work after the team project done in Data Science as a Field course. The continuous work focus on the machine learning method that predict the Wildfire in the U.S.

2 Design of Data and Methodology

2.1 Data Resource and Explanation of Variables

The dataset used for this report was found via the US Department of Agriculture. It provides information on 2,166,753 wildfires in the U.S. from 1992-2018, with a variety of information including spacial, cause, size, discovery/containment dates and different classifications.

Thanks to **U.S. DEPARTMENT OF AGRICULTURE** for providing the dataset.

```
##Read the dataset
# create db connection
conn <- dbConnect(SQLite(), 'FPA_FOD_20210617.sqlite')
# pull the fires table into RAM
fires <- tbl(conn, "Fires") %>% collect()
# disconnect from db
dbDisconnect(conn)
# select the column I need for this project
fires <- fires[,c('FIRE_NAME', 'FIRE_YEAR', 'DISCOVERY_DATE',
                  'NWCG_CAUSE_CLASSIFICATION',
                  'NWCG_GENERAL_CAUSE', 'FIRE_SIZE',
                  'FIRE_SIZE_CLASS', 'STATE', 'FIPS_CODE')]
```

```
## Description for attributes
# get column names and rename
fire_df_colname <- matrix(colnames(fires), ncol = 1)
colnames(fire_df_colname)[1] <- "Related-Variable"
# cbind the description for variable
fire_df_colname <-
  cbind(fire_df_colname,
  Description=
  c('Name of the incident from the fire report',
  'Date of Year on that fire',
  'Date on which the fire was discovered or confirmed to exist',
  'Code for the (statistical) cause of the fire',
  'Description of the (statistical) cause of the fire.',
  'Estimate of acres within the final perimeter of the fire.',
  'Code for fire size based on the number of acres within the final fire perimeter expen
  'Two-letter alphabetic code for the state in which the fire burned (or originated), ba
  'Numbers which uniquely identify geographic areas.'))
# kable related variable
kbl(as.data.frame(fire_df_colname), booktabs = T, longtable = T,
    caption = "The Variables of Interest in the Dataset") %>%
  kable_styling(full_width = T) %>%
  column_spec(1, color = "red") %>%
  column_spec(2, width = "25em")
```

Table 1: The Variables of Interest in the Dataset

Related-Variable	Description
FIRE_NAME	Name of the incident from the fire report
FIRE_YEAR	Date of Year on that fire
DISCOVERY_DATE	Date on which the fire was discovered or confirmed to exist
NWCG_CAUSE_CLASSIFICATION	Code for the (statistical) cause of the fire
NWCG_GENERAL_CAUSE	Description of the (statistical) cause of the fire.
FIRE_SIZE	Estimate of acres within the final perimeter of the fire.
FIRE_SIZE_CLASS	Code for fire size based on the number of acres within the final fire perimeter expenditures (A=greater than 0 but less than or equal to 0.25 acres, B=0.26-9.9 acres, C=10.0-99.9 acres, D=100-299 acres, E=300 to 999 acres, F=1000 to 4999 acres, and G=5000+ acres).
STATE	Two-letter alphabetic code for the state in which the fire burned (or originated), based on the nominal designation in the fire report.
FIPS_CODE	Numbers which uniquely identify geographic areas.

The data for this project was obtained via the websites mentioned above and compiled into one sqlite file to be read into R. There are some missing some values and certain information. The na.omit function was used to remove empty rows from the dataset and the usmap library was used to filter the state name column of the dataset.

The description for each variable inside the dataset can be found in the **Kaggle** dataset website. This website provides the yearly wildfire data for the United States. Although it is an out-of-date dataset the description for the variable still useful for our dataset. This website provides reshaped data to some extent which is originally from the national Fire Program Analysis (**FPA**).

2.2 Preparing the Data

The following was done to prepare the dataset for analysis:

1. Drill in on States/Counties impacted most by wildfires using the "include" parameter in the plot_usmap() function.
2. Remove rows missing information on fire size and fire cause.
3. Due to different categories for the dataset, only a subset of the columns were used in order to not duplicate the information.
4. Format date information to a more usable form.
5. Create a table that provides more information about each variable in the dataset.

2.3 R Library Foundations of the Project

This project may be imported into the RStudio environment and compiled by researchers wishing to reproduce this work for newest plot with future data sets, and having new findings or discussions from that.

> The Core of Statistics were done using R 4.1.0 (R Core Team, 2021-05-18), the ggplot2 (v3.3.5; RStudio Team, 2021-06-25), and the knitr (v1.34; Yihui, 2021-09-08) packages.

ggplot2 Package: this package has been used for creating graphics such as box plot, line plot, bar plot, and density plot from the reshaped datasets.

knitr Package: this report is constructed to have reproducibility that it can regenerate the plot based on the latest dataset contains yearly report in the future, using literate programming techniques for dynamic report generation in R.

> The Initial Scenarios package is usmap 0.5.2 (Paolo Di Lorenzo, 2021-01-21).

usmap Package: I use plot_usmap(based on ggplot object) to plot the US map. The map data frames include Alaska and Hawaii placed to the bottom left.

> The Most Frequently Used package is dplyr (v1.0.7; RStudio Team, 2021-06-18).

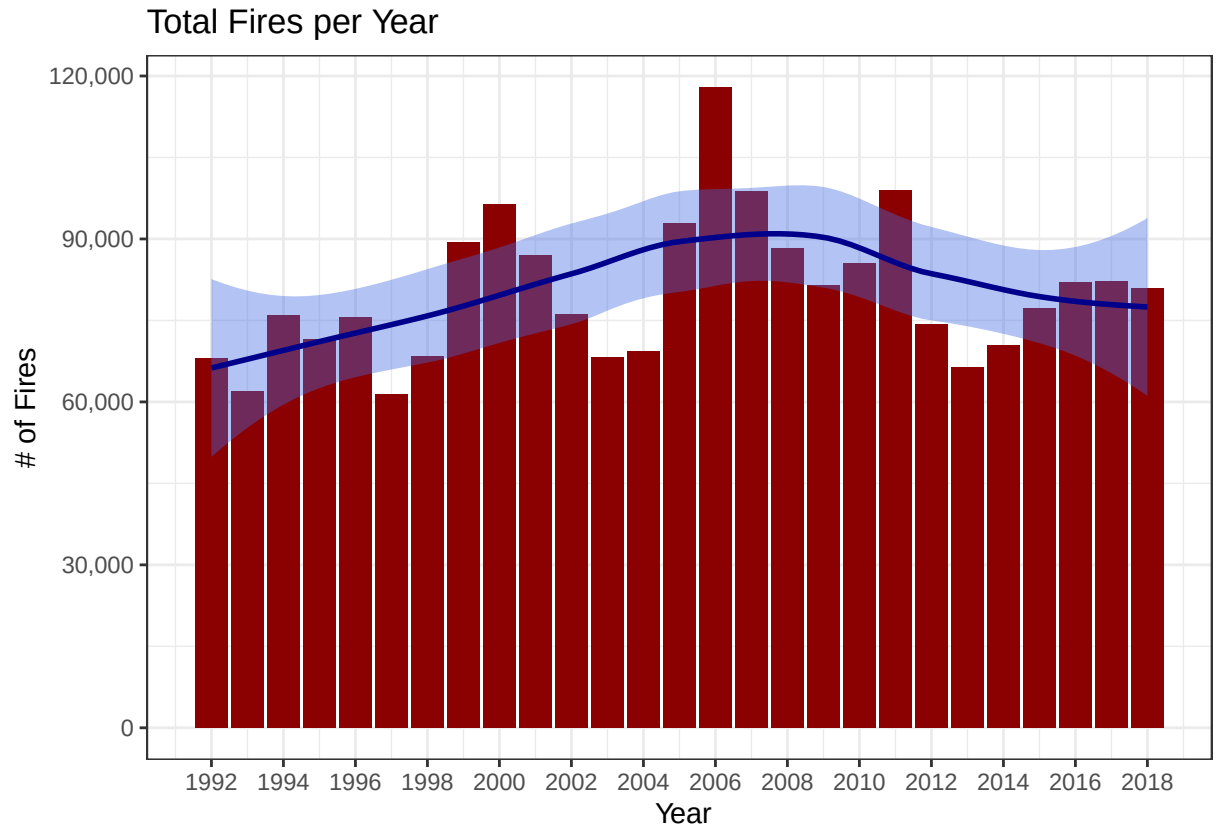
dplyr Package: many functions were used to reshape the dataset and working with data frames.

Note: There were other packages used for limited purposes, they are not listed here.

3 Exploration

3.1 Number of Fires Over Time

```
fires_date <- fires %>%
  select(FIRE_YEAR, FIRE_SIZE) %>%
  group_by(FIRE_YEAR) %>%
  summarise(Total_Fires = n(), Burn_Size = sum(FIRE_SIZE))
fires_date %>% ggplot(aes(x= FIRE_YEAR)) +
  geom_col(aes(y = Total_Fires), fill = "darkred") +
  stat_smooth(aes(method = "lm", y = Total_Fires),
              color = "darkblue", fill = "royalblue") +
  scale_x_continuous(name = " Year",
                     breaks = round(seq(min(fires_date$FIRE_YEAR),
                                        max(fires_date$FIRE_YEAR), by = 2),1)) +
  scale_y_continuous(name = "# of Fires", labels = scales::comma) +
  ggtitle("Total Fires per Year") + theme_bw()
```



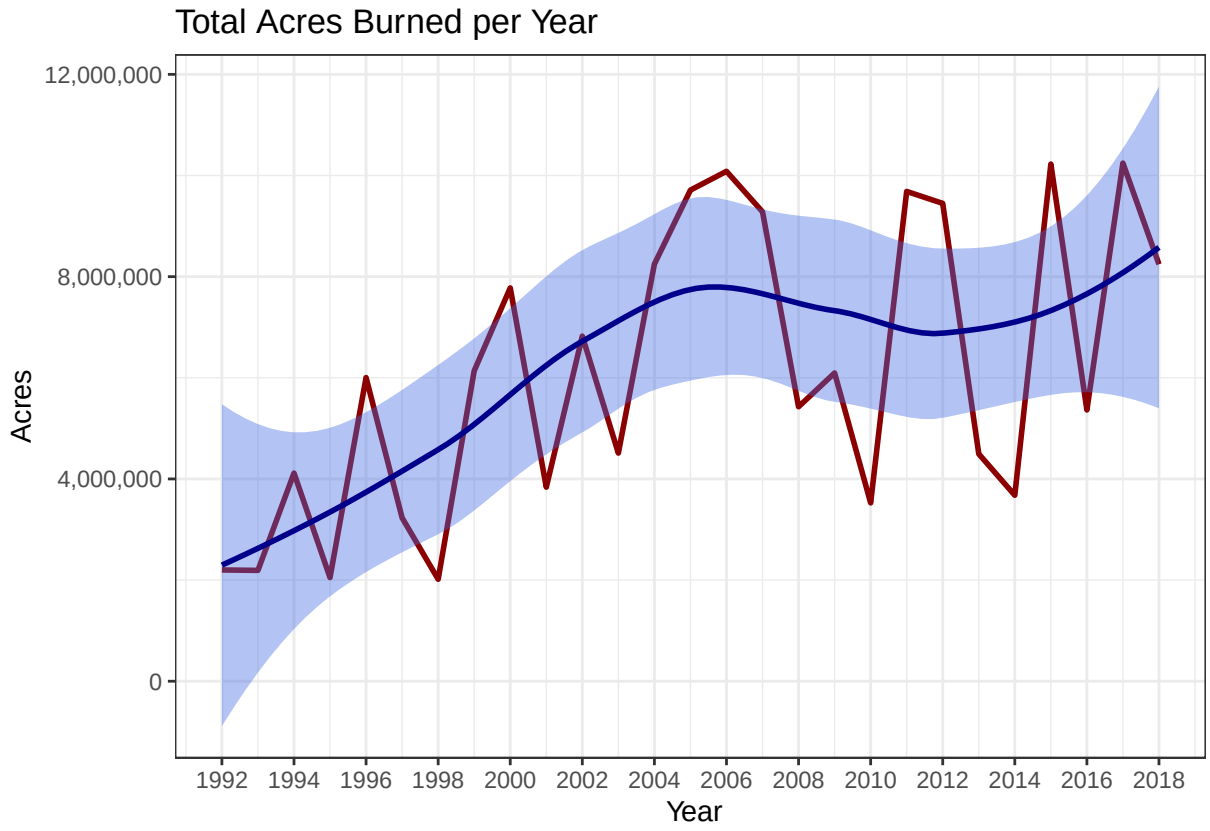
Observing the first plot, *Total Fires Per Year*, it shows the number of fires has not increased. In fact the data shows there was a peak around 2006 after which the number of fires decreased. A linear regression model was applied further confirming that there is not a relationship between year and fire count.

3.2 Fire Severity Over Time and Model

```
simple.fit = lm(FIRE_YEAR~Burn_Size, data=fires_date)
summary(simple.fit)
```

```
##
## Call:
## lm(formula = FIRE_YEAR ~ Burn_Size, data = fires_date)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.854 -5.455 -0.600  4.028 12.796
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.995e+03  3.106e+00 642.383  < 2e-16 ***
## Burn_Size   1.568e-06  4.634e-07   3.383  0.00236 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.704 on 25 degrees of freedom
## Multiple R-squared:  0.3141, Adjusted R-squared:  0.2866
## F-statistic: 11.45 on 1 and 25 DF,  p-value: 0.002363
```

```
fires_date %>% ggplot(aes(x= FIRE_YEAR)) +
  geom_line(aes(y = Burn_Size),size = 1, color = "darkred") +
  stat_smooth(aes(method = "lm", y = Burn_Size),
              fill = "royalblue", color = "darkblue") +
  scale_x_continuous(name = " Year",
                     breaks = round(seq(min(fires_date$FIRE_YEAR),
                                        max(fires_date$FIRE_YEAR), by = 2),1)) +
  scale_y_continuous(name = "Acres", labels = scales::comma) +
  ggtitle("Total Acres Burned per Year") + theme_bw()
```



In the second plot, *Total Acres Burned per Year*, the number of acres burned per year or in other terms, the severity of the fires. The linear regression model for the relationship shows a positive correlation between year and total acres burned.

3.3 Time Period with the Most Wildfire Activity

```
fires_1 <- as.data.frame(fires)
fires_1$DISCOVERY_DATE<-as.Date(fires_1$DISCOVERY_DATE, format = "%m/%d/%Y")
fires_1 <- fires_1 %>%
  mutate(day = format(DISCOVERY_DATE, "%d"),
         month = format(DISCOVERY_DATE, "%m")) %>%
  group_by(month, day) %>%
  summarise(total = sum(FIRE_SIZE)/27) %>%
  mutate(date = make_date(month = month, day = day))
ggplot() + geom_line(aes(x = date, y = total), fires_1, color = 'darkred') +
  scale_x_date(date_breaks= "1 month", date_labels = "%b") +
  xlab("Day of Year") + ylab("Average Number of Acres Burned") +
  theme(plot.background = element_rect(fill = "#BFD5E3"))
```

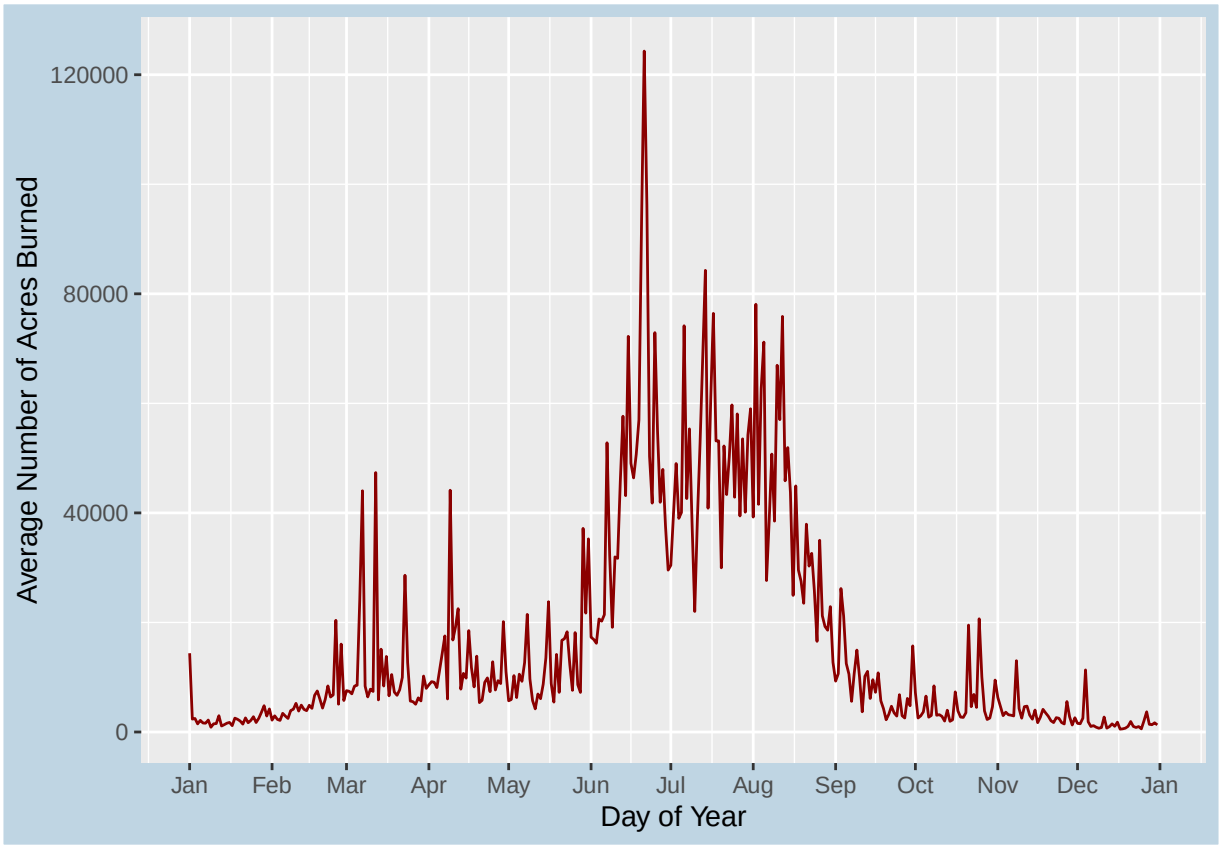


Figure 2: Average Number of Acres Burned by Day of Year

The graph was plotted between the average number of acres burned and day of year from 1992 to 2018. The graph shows there is a peak during June to September every year. Possible reasons for this are hot temperature and low rainfall during that part of the year. The graph is similar when limiting the data to 2013 to 2018, indicating that this correlation has not changed over time.

3.4 States with the Most Wildfire Activity

```
fires_6 <- as.data.frame(fires)
fires_6 <- fires_6 %>%
  group_by(STATE) %>%
  summarize(total = sum(FIRE_SIZE)/27) %>%
  na.omit()
fires_6 <- as.data.frame(fires_6)
colnames(fires_6)[1] = "state"
plot_usmap(data = fires_6, values = "total",
           color = "darkred", exclude = c("AK"), labels = TRUE) +
  scale_fill_continuous(low = "white", high = "darkred",
                        name = "Acres Burned per Year", label = scales::comma) +
  theme(legend.position = "right",
        legend.title = element_text(size=14),
        legend.text = element_text(size=16),
        plot.caption = element_text(size=20))
```

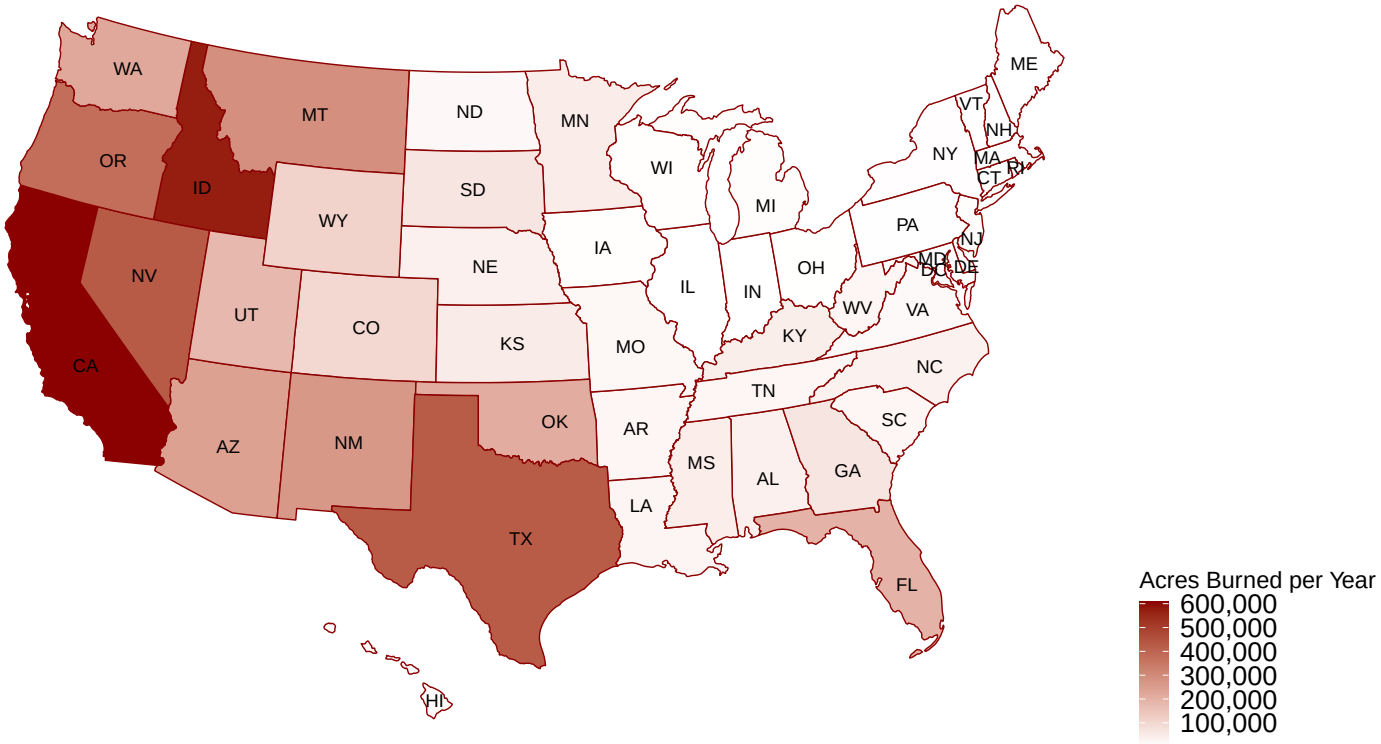


Figure 3: US Wildfres, 1992-2018. The spectrum from white to darkred indicates worse severity of wildfires in that State

The heat map of total acres burned of the U.S. show the western states like California, Idaho, and Texas have more severe wildfires than the eastern states. When Alaska is included in the calculations it has the most acres burned than any other state. However, due to the low population of this state, wildfires are allowed to burn with minimal intervention, so for this

analysis, Alaska is excluded. This is to highlight states where wildfires are a true threat to the population.

3.5 CA Counties with the Most Wildfire Activity

```
fires_7 <- as.data.frame(fires)
fires_7 <- fires_7 %>%
  filter(STATE == 'CA') %>%
  group_by(FIPS_CODE) %>%
  summarize(total = sum(FIRE_SIZE)/27) %>% na.omit()
fires_7 <- as.data.frame(fires_7)
colnames(fires_7)[1] = "fips"
plot_usmap(data = fires_7, values = "total", "counties",
           include = c("CA"), labels = FALSE, size = 0.4) +
  scale_fill_continuous(low = "white", high = "darkred",
                        name = "Acres Burned per Year", label = scales::comma) +
  theme(legend.position = "right",
        legend.title = element_text(size=16),
        legend.text = element_text(size=18),
        plot.caption = element_text(size=22))
```

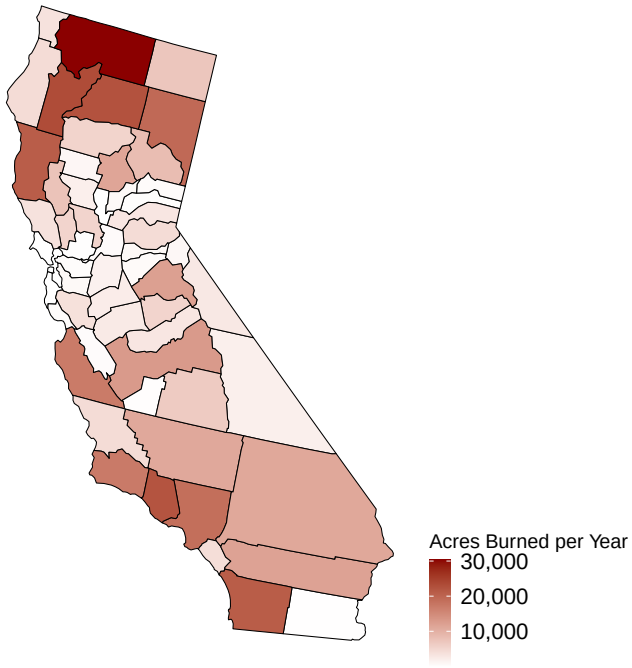


Figure 4: US Wildfires in CA, 1992-2018. The spectrum from white to darkred indicates worse severity of wildfires in that county

Highlighting California, the North and South counties are most at risk for wildfires. Longer periods of drought, high temperatures and high winds have caused these more severe wildfires.

4 Conclusion and Sources of Bias

Through this analysis, several conclusions can be made. First, that while the number of wildfires each year has not increased, the number of acres burned has steadily increased, indicating the severity of the wildfires has gotten worse over time. Second, the peak wildfire season is from June to September, and this season has not changed over the time frame of this dataset. Lastly, the region of the U.S. with the most severe wildfires is the West, specifically California, Idaho, and Texas when Alaska is excluded from the analysis.

However, there are possible sources of bias which could be influencing the findings of this report. It's unclear if the method of counting wildfires has changed over the years. It's possible what used to count as a wildfire does not, or in other terms, fires that occurred in the 90's might not count as a wildfire now, and would not be included in the data. Over the years how the estimate of acres burned by wildfires may have changed as technologies (such as satellite imaging) has improved. There is also the issue of recording smaller fires, or fires that do not persist for long periods of time, those may be missed and not recorded. Finally counties and states across the U.S. may be inconsistent in their reporting, causing bias by geographic location.

4.1 Wildfire by Size Class

```
fires_2 <- as.data.frame(fires)
size_classes <- c('A' = '0-0.25', 'B' = '0.26-9.9','C' = '10.0-99.9', 'D' = '100-299',
                  'E' = '300-999', 'F' = '1000-4999','G' = '5000+')
fires_2 <- fires_2 %>%
  group_by(FIRE_SIZE_CLASS) %>%
  summarize(total = n()/27) %>%
  mutate(FIRE_SIZE_CLASS = size_classes[FIRE_SIZE_CLASS])
ggplot(data = fires_2, aes(x=FIRE_SIZE_CLASS, y = total, fill =FIRE_SIZE_CLASS)) +
  geom_bar(stat = "identity") + scale_fill_brewer(palette = "Reds") +
  xlab("Number of Acres Burned") + ylab("Number of wildfires per Year") +
  geom_text(label = paste0(round(fires_2$total/sum(fires_2$total)*100, 1), "%")) +
  theme(plot.background = element_rect(fill = "#BFD5E3"))
```

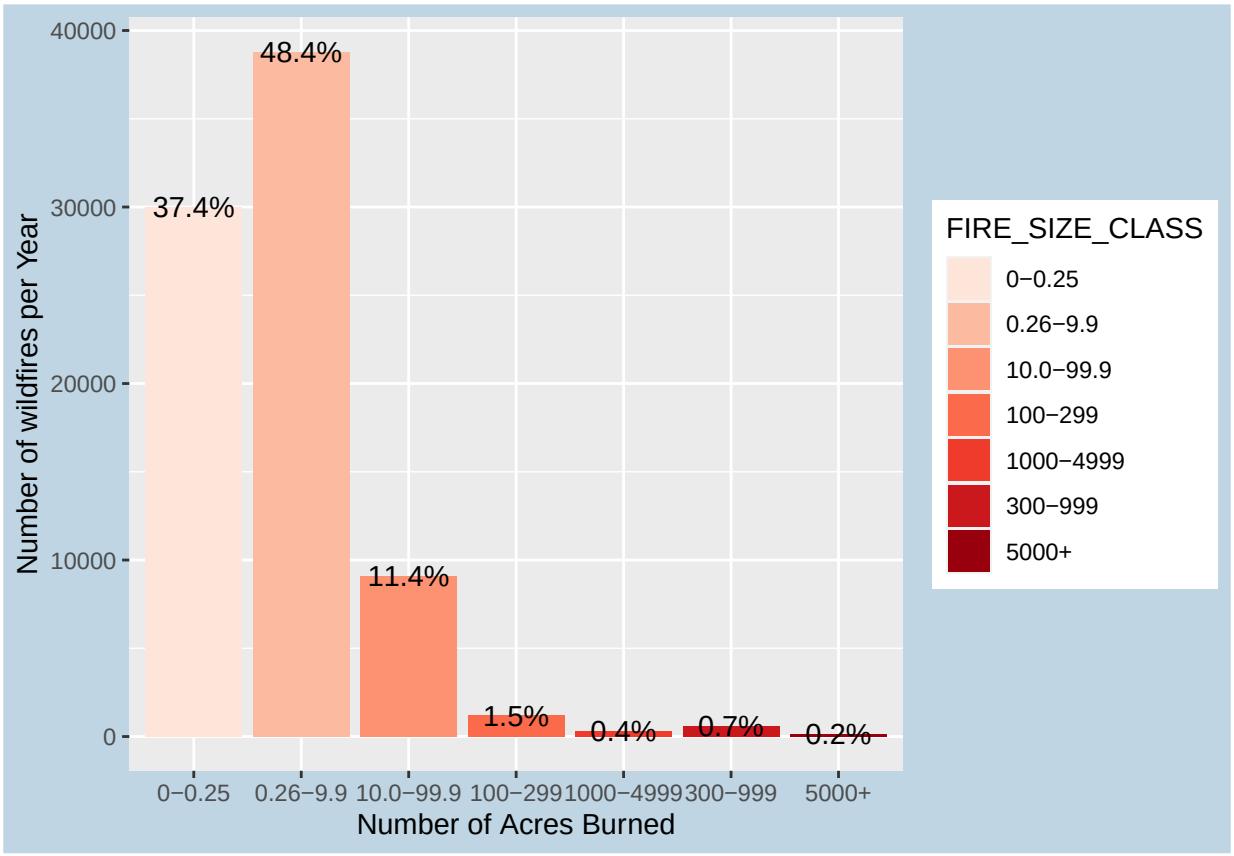


Figure 5: Number of Wildfires by Size Class

This plot shows the distribution of wildfires by their class size, a predetermined classification based on total acres burned.

4.2 Wildfires by General Cause

```
fires_5 <- as.data.frame(fires)
fires_5 <- fires_5 %>%
  group_by(NWCG_GENERAL_CAUSE) %>%
  summarize(mean_size = mean(FIRE_SIZE, na.rm = TRUE)) %>%
  na.omit() %>%
  arrange(desc(mean_size))
ggplot(data = fires_5) +
  geom_bar(aes(x = reorder(NWCG_GENERAL_CAUSE, mean_size), y = mean_size), stat = "ident
  coord_flip() +
  xlab("WILDFIRE CAUSE") + ylab("Number of Acres Burned per Fire") +
  theme(plot.background = element_rect(fill = "#BFD5E3"))
```

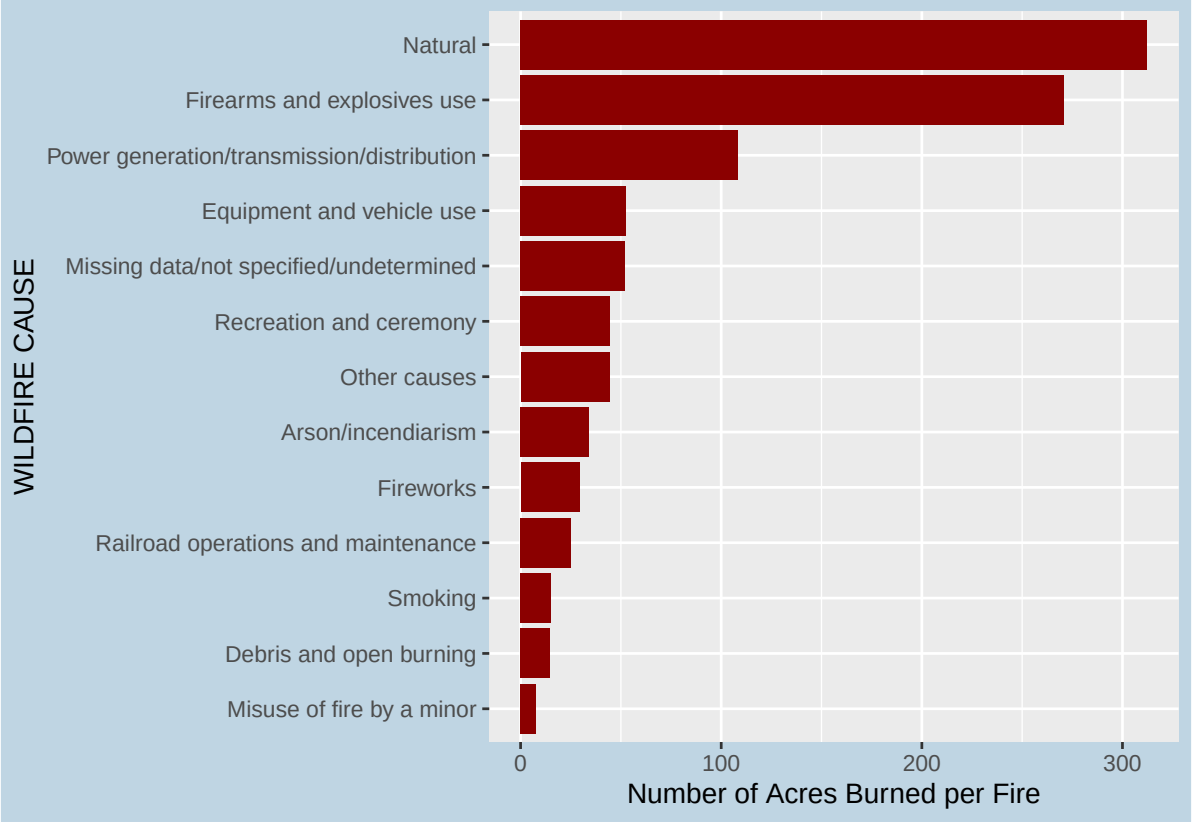


Figure 6: Average Wildfire Size by Cause

This plot shows the relationship between cause and average wildfire size.

5 Predict the Wildfire in U.S.

I have already explore enough details of the wildfire data set, now it is time to build a data model to predict the cause of a wildfire based on the size, location and date. In other words, given the cause of the each wildfire in the data set as well as numerical attributes such as fire_size, fire_date, fire_location. Could we use Machine learning method to predict the cause of a fire. I'm currently new to ML method so I will not build any complex model to increase the acceptable level of precision.

```
library(RSQLite)
library(tidyverse)
library(dbplyr)
library(lubridate)
library(data.table)
library(rpart.plot)
library(scales)
library(caret)
library(usmap)
library(kableExtra)
library(png)
library(doSNOW)
library(parallel)
# read data set
conn <- dbConnect(SQLite(), 'FPA_FOD_20210617.sqlite')
fires <- tbl(conn, "Fires") %>% collect()
dbDisconnect(conn)
# select the column I need for this project
fires <- fires[,c('FIRE_YEAR', 'DISCOVERY_DOY',
                  'NWCG_GENERAL_CAUSE', 'FIRE_SIZE',
                  'LATITUDE', 'LONGITUDE')]
# Randomly sample observations from the data
set.seed(123)
index <- sample(c(TRUE, FALSE), nrow(fires), replace = TRUE, prob = c(0.6, 0.4))
fires <- fires[index, ]
# features to use
features <- c('FIRE_SIZE')
fires$NWCG_GENERAL_CAUSE <- as.factor(fires$NWCG_GENERAL_CAUSE)
# index for train/test split
set.seed(123)
train_index <- sample(c(TRUE, FALSE), nrow(fires), replace = TRUE, prob = c(0.8, 0.2))
test_index <- !train_index
# Create x/y, train/test data
x_train <- as.data.frame(fires[train_index, features])
y_train <- fires$NWCG_GENERAL_CAUSE[train_index]
x_test <- as.data.frame(fires[test_index, features])
```

```
y_test <- fires$NWCG_GENERAL_CAUSE[test_index]
```

I read the data from the sql database, then build a random subset of the data (60%, which is 100,0000, one million) to make sure the training time less than couple hours (it will definitely takes more hours base on CPU, Mine is 3.6 GHz). Now I'm ready to test in different data model.

5.1 Decision Tree Model with Single Feature

I'm start with a simple decision tree with R Library 'caret', which provides a common API for ML. The single feature I choose is 'FIRE_SIZE'.

```
print(Sys.time())
```

```
## [1] "2021-12-05 22:32:01 MST"
```

```
# create the training control object.
tr_control <- trainControl(method = 'cv', number = 3)
# Train the decision tree model
set.seed(123)
dtree <- train(x = x_train,
               y = y_train,
               method = 'rpart',
               trControl = tr_control)
# make predictions using test set
preds <- predict(dtree, newdata = x_test)
# calculate accuracy on test set
test_set_acc <- round(sum(y_test == preds)/length(preds), 4)
print(paste(c("Accuracy:" , test_set_acc)))
```

```
## [1] "Accuracy:" "0.2904"
```

```
print(Sys.time())
```

```
## [1] "2021-12-05 22:32:28 MST"
```

The accuracy of the simple decision tree model is around 29% accuracy on the test set. However, I can definitely do better because it only spent less than one minute to finish the training. I can use larger sample size or more features to improve accuracy.

5.2 Decision Tree Model with More Features

I will use more features in the same decision tree model. Some of the variables in the data set are categorical features with multiple factor levels and take more than 24 hours to train the model. Instead, I will only use numerical values for training.

Because I've added more parameters so I will train this model allowing for more values of the complexity parameter. We can do this by increasing the tuneLength parameter, which will allow the possibility for deeper trees.

```
print(Sys.time())
```

```
## [1] "2021-12-05 20:27:04 MST"
```

```
features <- c('FIRE_YEAR','FIRE_SIZE', 'DISCOVERY_DOY', 'LATITUDE', 'LONGITUDE')
# index for train/test split
x_train <- as.data.frame(fires[train_index, features])
y_train <- fires$NWCG_GENERAL_CAUSE[train_index]
x_test <- as.data.frame(fires[test_index, features])
y_test <- fires$NWCG_GENERAL_CAUSE[test_index]
# Train the decision tree model
set.seed(123)
dtree <- train(x = x_train,
               y = y_train,
               method = 'rpart',
               tuneLength = 8,
               trControl = tr_control)
# make predictions using test set
preds <- predict(dtree, newdata = x_test)
# calculate accuracy on test set
test_set_acc <- sum(y_test == preds)/length(preds)
print(paste(c("Accuracy:" , round(test_set_acc, 4))))
```

```
## [1] "Accuracy:" "0.4121"
```

```
print(dtree$resample)
```

```
##    Accuracy     Kappa Resample
## 1 0.4103227 0.2508893    Fold1
## 2 0.4129800 0.2559618    Fold2
## 3 0.4114781 0.2522630    Fold3
```

```
print(Sys.time())
```

```
## [1] "2021-12-05 20:29:45 MST"
```

This model shows an even more improved accuracy on the test set of 41%. Let's try other models to compare.

5.3 XGBoost Model from Gradient Boosting Algorithms

The xgboost algorithm is different from the methods we've used so far in that it can handle missing values.

```
numberofcores = detectCores()  # review what number of cores does for your environment
cl <- makeCluster(numberofcores, type = "SOCK")
# Register cluster so that caret will know to train in parallel.
registerDoSNOW(cl)
print(Sys.time())
```

```
## [1] "2021-12-05 20:30:18 MST"
```

```
tr_control <- trainControl(
    method = 'cv',
    number = 5,
    verboseIter = TRUE,
    allowParallel = TRUE)
tune_grid <- expand.grid(
    nrounds = c(100),
    max_depth = c(8),
    eta = c(0.01),
    gamma = c(1),
    colsample_bytree = c(0.5),
    subsample = c(1),
    min_child_weight = c(0))
# Train the decision tree model
set.seed(123)
xgbmodel <- train(
    x = x_train,
    y = y_train,
    method = 'xgbTree',
    trControl = tr_control,
    tuneGrid = tune_grid)
```

```
## Aggregating results
## Fitting final model on full training set
```

```
# make predictions using test set
preds <- predict(xgbmodel, newdata = x_test)
# calculate accuracy on test set
test_set_acc <- sum(y_test == preds)/length(preds)
print(paste(c("Accuracy:" , round(test_set_acc, 4))))
```

```
## [1] "Accuracy:" "0.4752"
```

```
print(xgbmodel$resample)
```

```
##    Accuracy     Kappa Resample
## 1 0.4743806 0.3296025    Fold1
## 2 0.4737798 0.3277070    Fold2
## 3 0.4765378 0.3316245    Fold3
## 4 0.4728621 0.3271220    Fold4
## 5 0.4739194 0.3282534    Fold5
```

```
print(Sys.time())
```

```
## [1] "2021-12-05 21:09:01 MST"
```

```
stopCluster(cl)
```

The xgboost model gives me around 47% accuracy, which is better than decision tree with less training time.

5.4 Random Forest Model from Ensembling Decision Tree

At this point, the previous decision tree model may have the possibility of over-fitting. We don't want our model to simply 'memorize' the training data. We want the model to generalize well to unseen data. So we'll move from using a single decision tree to using a whole bunch of simpler trees and combining them. We'll use random forest, which performs the ensembling while also randomly selecting a subset of the features on which to seek optimal splits at each node.

Here we run a random forest model. To keep the runtime reasonable we'll set it to use 100 trees. We could certainly increase this however if we wished.

```
numberofcores = detectCores()  # review what number of cores does for your environment
cl <- makeCluster(numberofcores, type = "SOCK")
print(Sys.time())
```

```
## [1] "2021-12-05 22:32:45 MST"
```

```
# Train the decision tree model
set.seed(123)
rfmodel <- train(x = x_train,
                 y = y_train,
                 method = 'rf',
                 tuneLength = 8,
                 ntree = 100,
                 trControl = tr_control)
```

```
## note: only 4 unique complexity parameters in default grid. Truncating the grid to 4 .
##
## + Fold1: mtry=2
## - Fold1: mtry=2
## + Fold1: mtry=3
## - Fold1: mtry=3
## + Fold1: mtry=4
## - Fold1: mtry=4
## + Fold1: mtry=5
## - Fold1: mtry=5
## + Fold2: mtry=2
## - Fold2: mtry=2
## + Fold2: mtry=3
## - Fold2: mtry=3
## + Fold2: mtry=4
## - Fold2: mtry=4
## + Fold2: mtry=5
## - Fold2: mtry=5
## + Fold3: mtry=2
## - Fold3: mtry=2
```

```
## + Fold3: mtry=3
## - Fold3: mtry=3
## + Fold3: mtry=4
## - Fold3: mtry=4
## + Fold3: mtry=5
## - Fold3: mtry=5
## + Fold4: mtry=2
## - Fold4: mtry=2
## + Fold4: mtry=3
## - Fold4: mtry=3
## + Fold4: mtry=4
## - Fold4: mtry=4
## + Fold4: mtry=5
## - Fold4: mtry=5
## + Fold5: mtry=2
## - Fold5: mtry=2
## + Fold5: mtry=3
## - Fold5: mtry=3
## + Fold5: mtry=4
## - Fold5: mtry=4
## + Fold5: mtry=5
## - Fold5: mtry=5
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 3 on full training set
```

```
# make predictions using test set
preds <- predict(rfmodel, newdata = x_test)
# calculate accuracy on test set
test_set_acc <- sum(y_test == preds)/length(preds)
print(paste(c("Accuracy:" , round(test_set_acc, 4))))
```

```
## [1] "Accuracy:" "0.5868"
```

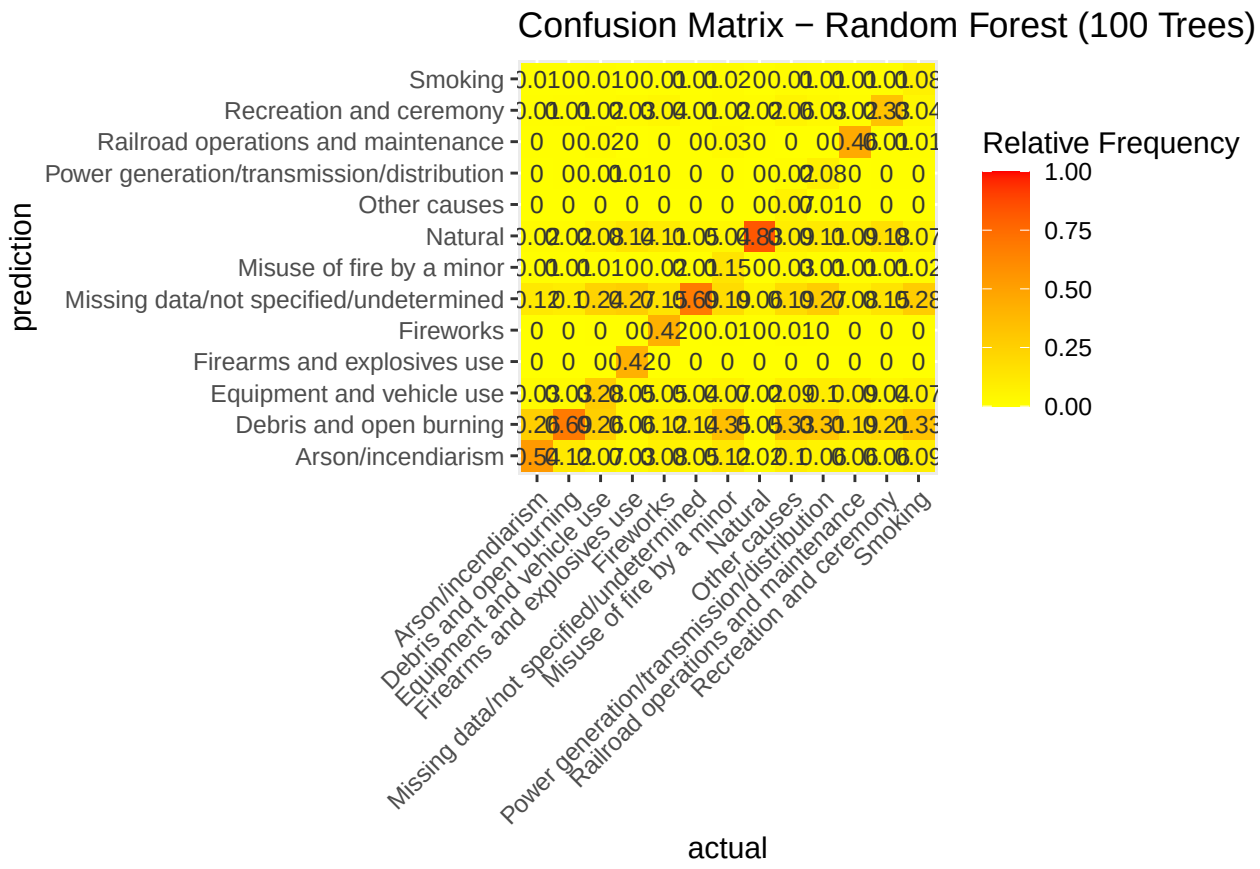
```
print(rfmodel$resample)
```

```
##    Accuracy     Kappa Resample
## 1 0.5802837 0.4830611    Fold1
## 2 0.5817451 0.4848442    Fold2
## 3 0.5796736 0.4821891    Fold3
## 4 0.5799674 0.4827965    Fold4
## 5 0.5804663 0.4831609    Fold5
```

```
print(Sys.time())
```

```
## [1] "2021-12-06 00:06:27 MST"
```

```
confusionMatrix(y_test, preds)$table %>%
    prop.table(margin = 1) %>%
    as.data.frame.matrix() %>%
    rownames_to_column(var = 'actual') %>%
    gather(key = 'prediction', value = 'freq',-actual) %>%
    ggplot(aes(x = actual, y = prediction, fill = freq)) +
    geom_tile() +
    geom_text(aes(label = round(freq, 2)), size = 3, color = 'gray20') +
    scale_fill_gradient(low = 'yellow', high = 'red', limits = c(0,1), name = 'Relative
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    ggtitle('Confusion Matrix - Random Forest (100 Trees)')
```

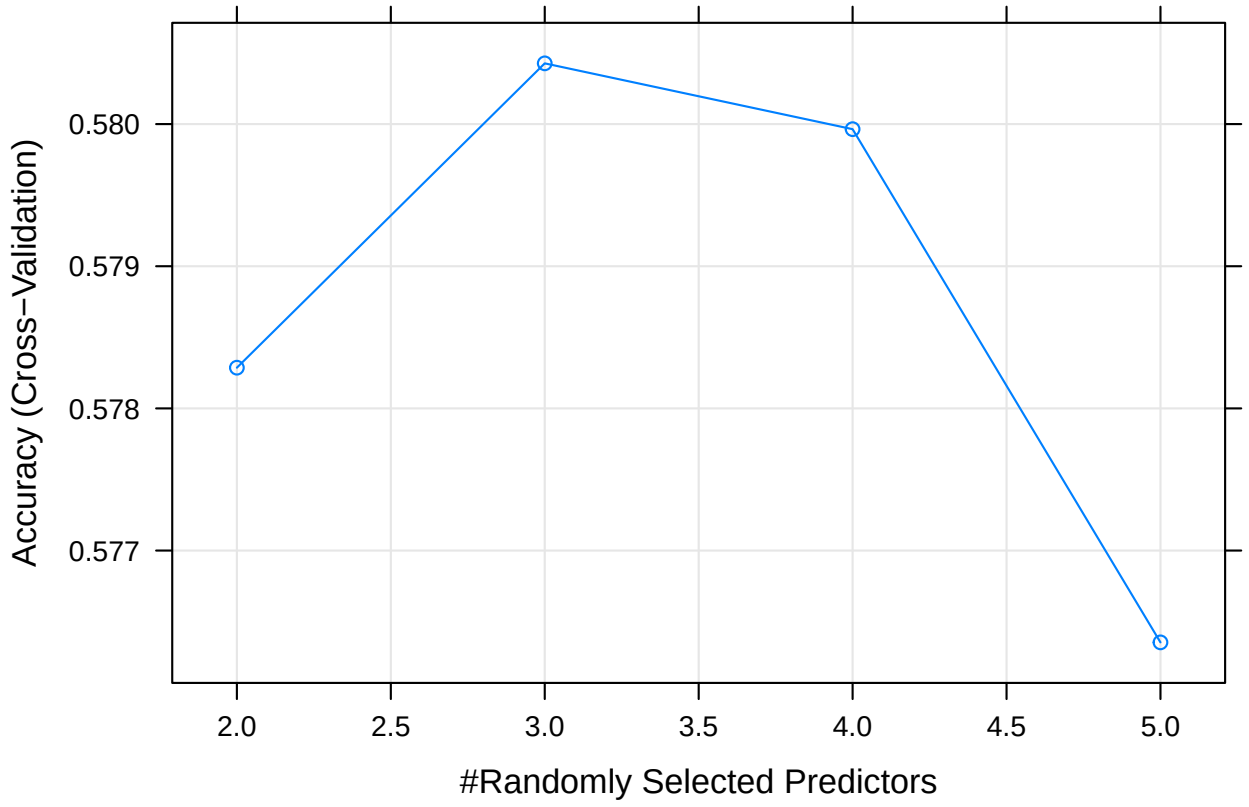



```
# show confusion matrix
confusionMatrix(y_test, preds)$table %>%
    as.data.frame.matrix() %>%
    kable("pipe") %>%
    kable_styling(bootstrap_options = c('striped'), font_size = 4)
```

	Arson/incendiarism	Debris and open burning	Equipment and vehicle use	Firearms and explosives use	Fireworks	Missing data/not specified/undetermined	Misuse of fire by a minor	Natural	Other causes	Power generation/transmission/distribution	Railroad operations and maintenance	Recreation and ceremony	Smoking
Arson/incendiarism	1981	5660	1011	3	66	4386	408	842	15	48	126	364	216
Debris and open burning	7256	42068	1816	4	109	6143	582	1487	53	170	268	903	304
Equipment and vehicle use	1454	5597	5917	9	76	5069	238	1701	35	145	438	408	190
Firearms and explosives use	7	16	14	118	1	74	1	39	0	2	0	7	0
Fireworks	167	244	98	3	890	314	39	236	4	6	6	90	16
Missing data/not specified/undetermined	3111	9145	2934	9	132	44924	431	3022	47	190	319	866	389
Misuse of fire by a minor	955	2667	543	0	55	1464	1150	333	19	22	198	173	131
Natural	684	1719	691	4	68	2298	81	3124	42	51	116	648	77
Other causes	111	381	100	1	13	219	38	106	77	18	2	67	16
Power generation/transmission/distribution	201	1041	329	2	15	912	28	381	20	285	5	117	28
Railroad operations and maintenance	279	839	383	0	15	336	44	385	2	4	2035	78	35

	Arson	Debris and open burning	Equipment and vehicle use	Firearms and explosives use	Fireworks	Missing data/not specified/undetermined	Misuse of fire by a minor	Natural	Other causes	Power generation/transmission/distribution	Railroad operation and maintenance	Recreation and ceremony	Smoking
Recreation and ceremony	609	2366	464	3	37	1657	109	1951	18	42	59	3584	120
Smoking	688	2456	541	0	31	2066	181	493	17	35	85	332	572

```
plot(rfmodel)
```




```
stopCluster(cl)
```

The random forest model gives me 58% accuracy, which is better than all previous models. However, there are definitely a lot of things I can do to improve the accuracy:

1. Add more features such as “STATE”, “Buring_Period”.
2. Use more data. The current sample size is 60% of original data set.

3. Try other model such as Neural Networks, KNN, Logistic Regression, SVM, and etc.
4. Try different parameters and rounds to see if improve the accuracy.

6 References

[1] Colorado's Air Quality is Pretty Bad Today And Will Get Worse
https://www.cpr.org/2021/08/05/colorado-air-quality-bad-today-will-get-worse/

[2] California WildFires (2013-2020) - Kaggle Data website,
https://www.kaggle.com/ananthu017/california-wildfire-incidents-20132020

[3] Spatial wildfire occurrence data for the United States, 1992-2018 - U.S. Department of Agriculture,
https://www.fs.usda.gov/rds/archive/Catalog/RDS-2013-0009.5

[4] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria,
http://www.R-project.org/, 2021

[5] Yihui Xie knitr: A general-purpose package for dynamic report generation in R,
http://yihui.name/knitr/, 2021

[6] Different Ways of Plotting U.S. Map in R,
https://jtr13.github.io/cc19/different-ways-of-plotting-u-s-map-in-r.html#using-usmap-package, 2021

[7] Census Regions and Divisions of the United States - U.S. Census Bureau,
https://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf, 2021

[8] Easy way to mix multiple graphs on the same page - ggplot2 package,
http://www.sthda.com/english/articles/24-ggpubr-publication-ready-plots/81-ggplot2-easy-way-to-mix-multiple-graphs-on-the-same-page/

[9] 1.88 Million US Wildfires - Kaggle Data website,
https://www.kaggle.com/rtatman/188-million-us-wildfires

[10] Figures, Tables, Captions - R Markdown for Scientists,
https://rmd4sci.njtierney.com/figures-tables-captions-.html, 2021

[11] Yang Liu ggplot US state heatmap - usmap package,
https://liuyanguu.github.io/post/2020/06/12/ggplot-us-state-and-china-province-heatmap/, 2021