Forest Fire Analysis

2023 December

Project Objective: To analyse fire occurrence data to uncover patterns and relationships between various factors such as month, area, rain, fire severity, and so on.

Language: R

Techniques used:

Data Preprocessing:

- Arrange values (e.g., month and date) in the correct order for intuitive analysis.
- Pivot the data into a longer format to make it easier to plot (for scatter plots).

Data Visualisation using ggplot:

- Create a histogram to understand the pattern of fire occurrences by month.
- Use a scatter plot to find relationships between the variable 'month' and other variables (area, rain, etc.) and fire severity.
- Identify outliers through summary statistics from the scatter plot.
- Remove outliers to better visualize relationships between variables.

photo by Pixabay, https://www.stockvault.net/photo/207624/forest-fire#

About the dataset:

`X` : X-axis spatial coordinate within the Montesinho park map: 1 to 9

`Y` : Y-axis spatial coordinate within the Montesinho park map: 2 to 9 `month`: Month of the year: ‘jan’ to ‘dec’

`day` : Day of the week: ‘mon’ to ‘sun’

`FFMC` : Fine Fuel Moisture Code index from the FWI system: 18.7 to 96.20

`DMC` : Duff Moisture Code index from the FWI system: 1.1 to 291.3

`DC` : Drought Code index from the FWI system: 7.9 to 860.6

`ISI` : Initial Spread Index from the FWI system: 0.0 to 56.10

`temp` : Temperature in Celsius degrees: 2.2 to 33.30

`RH` : Relative humidity in percentage: 15.0 to 100

`wind` : Wind speed in km/h: 0.40 to 9.40

`rain` : Outside rain in mm/m2 : 0.0 to 6.4

`area` : The burned area of the forest (in ha): 0.00 to 1090.84

Note:

- A single row corresponds to the location of a fire and some characteristics of the fire itself.
- Higher water presence is typically associated with less fire spread, therefore we can expect the water-related variables (`DMC` and `rain`) to be associated with `area`.

Import required libraries/packages:

```r
library(ggplot2)
library(tidyr)
library(readr)
library(dplyr)
```

```
##
## 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
```

Load data:

```
setwd("C:\\Users\\S\\Desktop\\R_test")
df <- read_csv("forestfires.csv", show_col_types = FALSE)
df
```

```
## # A tibble: 517 × 13
##        X     Y month day    FFMC   DMC    DC   ISI  temp    RH  wind  rain  area
##    <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1     7     5 mar   fri    86.2  26.2  94.3   5.1   8.2    51   6.7   0       0
##  2     7     4 oct   tue    90.6  35.4 669.    6.7  18      33   0.9   0       0
##  3     7     4 oct   sat    90.6  43.7 687.    6.7  14.6    33   1.3   0       0
##  4     8     6 mar   fri    91.7  33.3  77.5   9     8.3    97   4     0.2     0
##  5     8     6 mar   sun    89.3  51.3 102.    9.6  11.4    99   1.8   0       0
##  6     8     6 aug   sun    92.3  85.3 488    14.7  22.2    29   5.4   0       0
##  7     8     6 aug   mon    92.3  88.9 496.    8.5  24.1    27   3.1   0       0
##  8     8     6 aug   mon    91.5 145.  608.   10.7   8      86   2.2   0       0
##  9     8     6 sep   tue    91   130.  693.    7    13.1    63   5.4   0       0
## 10     7     5 sep   sat    92.5  88   699.    7.1  22.8    40   4     0       0
## # ℹ 507 more rows
```

Pre-Processing Data: Organise month and date in the correct order:

We can see that values in `month` and `date` are not in the right order. We will arrange them in the correct order to facilitate intuitive representation and analysis.

```
# Check the order of values


df %>% pull(month) %>% unique
```

```
##  [1] "mar" "oct" "aug" "sep" "apr" "jun" "jul" "feb" "jan" "dec" "may" "nov"
```

```
df %>% pull(day) %>% unique
```

```
## [1] "fri" "tue" "sat" "sun" "mon" "wed" "thu"
```

```
# Arrange values in the correct order:


df <- df %>%
  mutate(month_reordered = factor(month, levels = c("jan", "feb", "mar", "apr", "may",
"jun", "jul", "aug", "sep", "oct", "nov", "dec")), day_reordered = factor(day, levels =
c("mon", "tue", "wed", "thu", "fri", "sat", "sun"))
  )
# Check if the values of 'Month' and 'day' are ordered properly:
```

```
df %>% pull(month_reordered) %>% unique
```

```
##  [1] mar oct aug sep apr jun jul feb jan dec may nov
## Levels: jan feb mar apr may jun jul aug sep oct nov dec
```

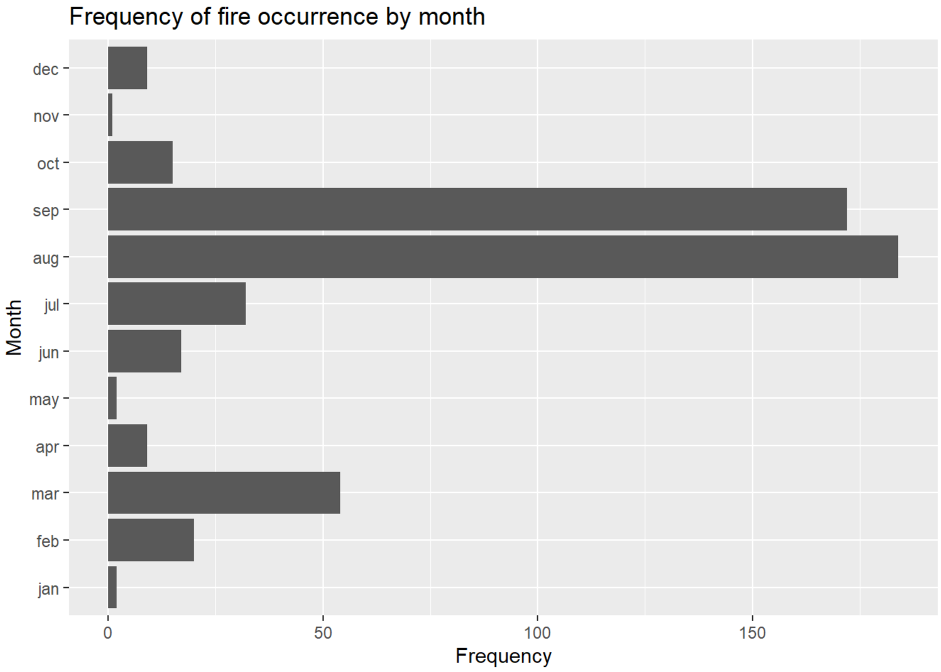
```
df %>% pull(day_reordered) %>% unique
```

```
## [1] fri tue sat sun mon wed thu
## Levels: mon tue wed thu fri sat sun
```

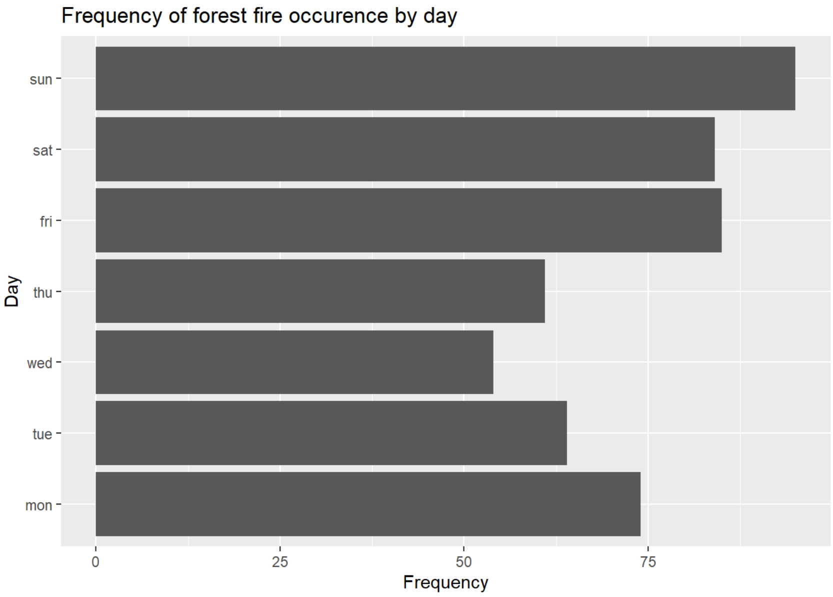
When do most forest fires occur?

Let’s understand the pattern of forest fires first. Find more about the frequency of fire occurrence by month and day, respectively.

```
# Fire occurrence by month

df_occurence_month <- df %>%
  group_by(month_reordered) %>%
  summarize(count = n())

df_occurence_month %>%
  ggplot(aes(x=count, y=month_reordered))+
  geom_bar(stat = "identity")+
  labs(
    title="Frequency of fire occurrence by month",
    x= "Frequency",
    y= "Month"
  )
```



```r
# Fire occurrence by day

df_occurrence_day  <- df %>%
  group_by(day_reordered) %>%
  summarize(count = n())

df_occurrence_day %>%
  ggplot(aes(x=count, y=day_reordered))+
  geom_bar(stat = "identity")+
  labs(
    title="Frequency of forest fire occurence by day",
    x= "Frequency",
    y= "Day"
  )
```



Observations:

- August and September see more forest fires than other months.
- Weekend have more fires (Friday, Saturday, and Sunday).

```r
# Further analysis: Total number of fires for each combination of 'month_reordered' and 'day_reordered'

df_month_day <- df %>%
  group_by(month_reordered, day_reordered) %>%
  summarize(total = n())
```

```
## `summarise()` has grouped output by 'month_reordered'. You can override using
## the `.groups` argument.
```

```r
df_month_day
```

```
## # A tibble: 64 × 3
## # Groups:   month_reordered [12]
##    month_reordered day_reordered total
##    <fct>           <fct>         <int>
##  1 jan             sat               1
##  2 jan             sun               1
##  3 feb             mon               3
##  4 feb             tue               2
##  5 feb             wed               1
##  6 feb             thu               1
##  7 feb             fri               5
##  8 feb             sat               4
##  9 feb             sun               4
## 10 mar             mon              12
## # ℹ 54 more rows
```

How each of the other 8 variables (**`FFMC`** ~ **`rain`**) relates to month?:

For this analysis, we chose `month` as our main variable as it can vary a lot between seasons.

To find relationship between month and the 8 other variables, we will first need to pivot the data into a longer dimension to make it easier to plot.

```
# Pivoting the data

df_pivoted <- df%>%
  pivot_longer(cols= c(FFMC, DMC, DC, ISI, temp, RH, wind, rain),
               names_to = "column",
               values_to = "value"
               )
df_pivoted
```

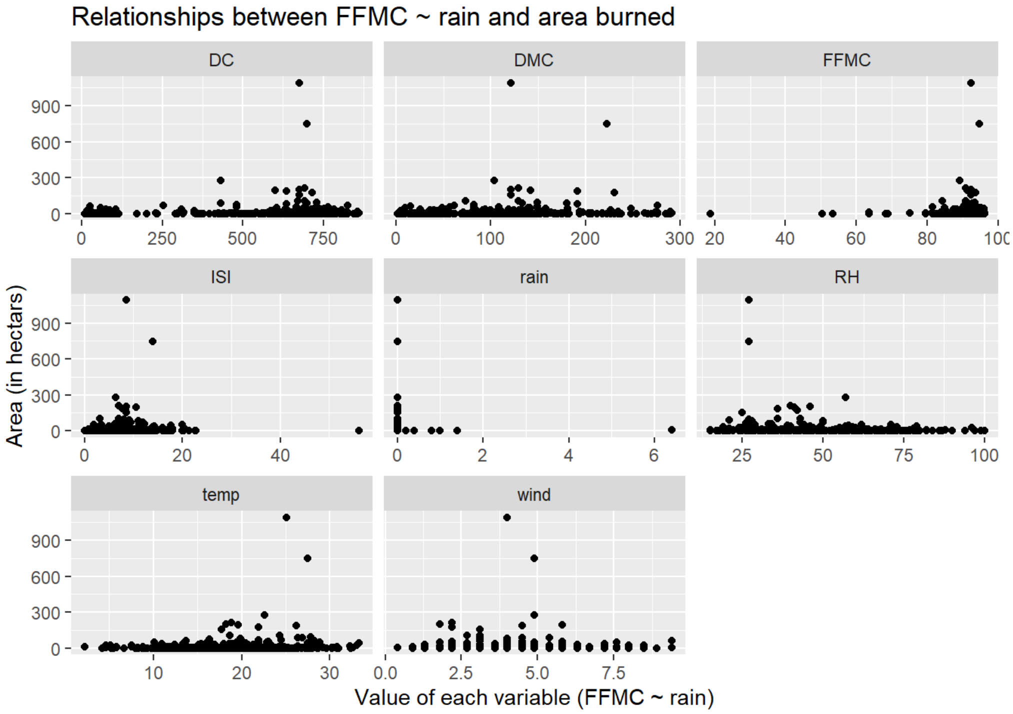
```
## # A tibble: 4,136 × 9
##        X     Y month day    area month_reordered day_reordered column value
##    <dbl> <dbl> <chr> <chr> <dbl> <fct>           <fct>         <chr>  <dbl>
##  1     7     5 mar   fri       0 mar             fri           FFMC    86.2
##  2     7     5 mar   fri       0 mar             fri           DMC     26.2
##  3     7     5 mar   fri       0 mar             fri           DC      94.3
##  4     7     5 mar   fri       0 mar             fri           ISI      5.1
##  5     7     5 mar   fri       0 mar             fri           temp     8.2
##  6     7     5 mar   fri       0 mar             fri           RH      51  
##  7     7     5 mar   fri       0 mar             fri           wind     6.7
##  8     7     5 mar   fri       0 mar             fri           rain     0  
##  9     7     4 oct   tue       0 oct             tue           FFMC    90.6
## 10     7     4 oct   tue       0 oct             tue           DMC     35.4
## # ℹ 4,126 more rows
```

Examining Forest Fire Severity:

The `area` contains data on the number of hectares of forest that burned during the forest fire. We will use this variable as an indicator of the severity of the fire.

We will use a scatter plot to learn about relationships between the area burnt and the 8 variables.

```
# Using scatter plot to find relationships between the area burnt and the 8 variables

df_pivoted%>%
    ggplot(aes(x = value, y = area))+
    geom_point()+
    facet_wrap(vars(column), scale = "free_x")+
    labs(
        title = "Relationships between FFMC ~ rain and area burned",
        x = "Value of each variable (FFMC ~ rain)",
        y = "Area (in hectars)"
    )
```



Observations:

- The outliers in the plots represent fires that caused inordinate amounts of damage compared to the other fires.

Outliers:

From the scatter plot above, we noticed some outliers of values of the 8 different variables (`FFMC` ~ `rain`). We will investigate further by employing summary statistics and histograms through analysis.

```
# Summary stat

summary_stat_area <- df_pivoted %>%
  summarize(
    count = n(),
    sum_val = sum(area),
    min_val = min(area),
    max_val = max(area),
    med_val = median(area),
    avg = mean(area),
    upper_quartile_75 = quantile(area, probs = 0.75),
    upper_quartile_90 = quantile(area, probs = 0.9)
  )
```

```
# Convert summary statistics to a data frame

summary_table <- as.data.frame(t(summary_stat_area))
summary_table
```

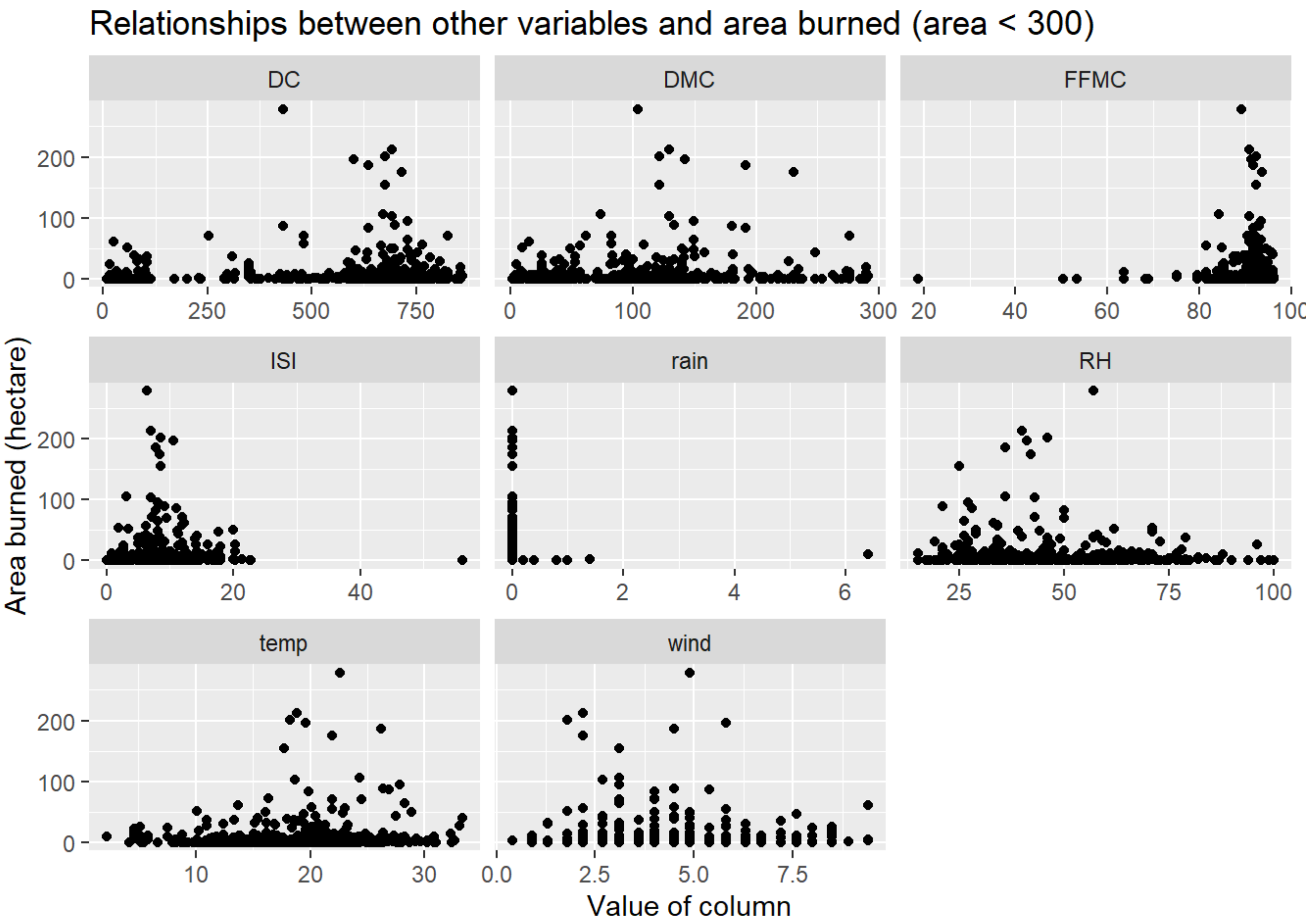
```
##                        V1
## count            4136.00000
## sum_val         53136.40000
## min_val             0.00000
## max_val          1090.84000
## med_val             0.52000
## avg               12.84729
## upper_quartile_75    6.57000
## upper_quartile_90   26.00000
```

Observations:

- From the summary statistics, we can notice that there is a huge gap between `avg` and `max`.
- `upper_quartile_75` of 6.57 is less affected by the outlier.
- We increased the upper quartile to 90%. Likewise, `upper_quartile_90` is still less affected by the outlier.

To better **visualise relationships between variables, we filtered `area' values except for rows with** very high values of area:

```
### answer from solutions - which I still have no idea of

df_pivoted %>%
  filter(area < 300) %>%
  ggplot(aes(x = value, y = area)) +
  geom_point() +
  facet_wrap(vars(column), scales = "free_x") +
  labs(
    title = "Relationships between other variables and area burned (area < 300)",
    x = "Value of column",
    y = "Area burned (hectare)")
```



Relationships between other variables and area burned (area < 300)