# MU Inc

According to the Canadian Wildland Fire Information System (CWFIS), British Columbia (BC) has recorded 1.4M wildfires over the past 10 years.

Thanks to the data collected by multiple satellite sources, such as AVHRR, MODIS, and VIIRS, we now know how serious this problem is. Even though satellites are highly accurate technology, some information was missing. To solve this problem, we deleted the N/A values and converted the necessary data types.

If you want to know the causes of BC's wildfires, we did some Exploratory Data Analysis which included analyzing key statistics such as averages, frequencies, and correlation coefficients, to uncover patterns and relationships within the data. The causes are:

- British Columbia’s vegetation is mostly composed of Coniferous Forest Fuel Types (C1 to C7) and bog, which means it is persistent to fire and difficult to extinguish.
- From the analysis it’s known that trees in the province are mostly leafless, affecting available fuel.
- In 2017, 2018, and 2021 precipitations reached the lowest point in the past 10 years, coincidentally those were the years with more wildfires.

If you’re really in love with this province as we are, you’ll get a sense of how important is the ecosystem for BC-ians. For 10yrs MU, Inc. has been studying wildfire behavior and come up with an amazing idea, which would be the anti-wildfire spray.

From our work in the field, we know that there is a negative but moderate correlation between Drought Code and the Age of the trees, specifically –0.4. So, in that sense, we have created a product to rejuvenate older trees and make them moister.

The product will moisturize the decomposed organic layers, important for smoldering fires, that would result in the that would result in less spread of wildfires and potentially mitigating wildfire ignitions. Based on the data, it has been shown that moisture levels in British Columbia are lower compared to other provinces. Addressing this issue could help mitigate potential problems in BC for the next decade.

Never let a wildfire happen again.  Visit our website to schedule your free product demo at your closest park.

**MU, Inc.**

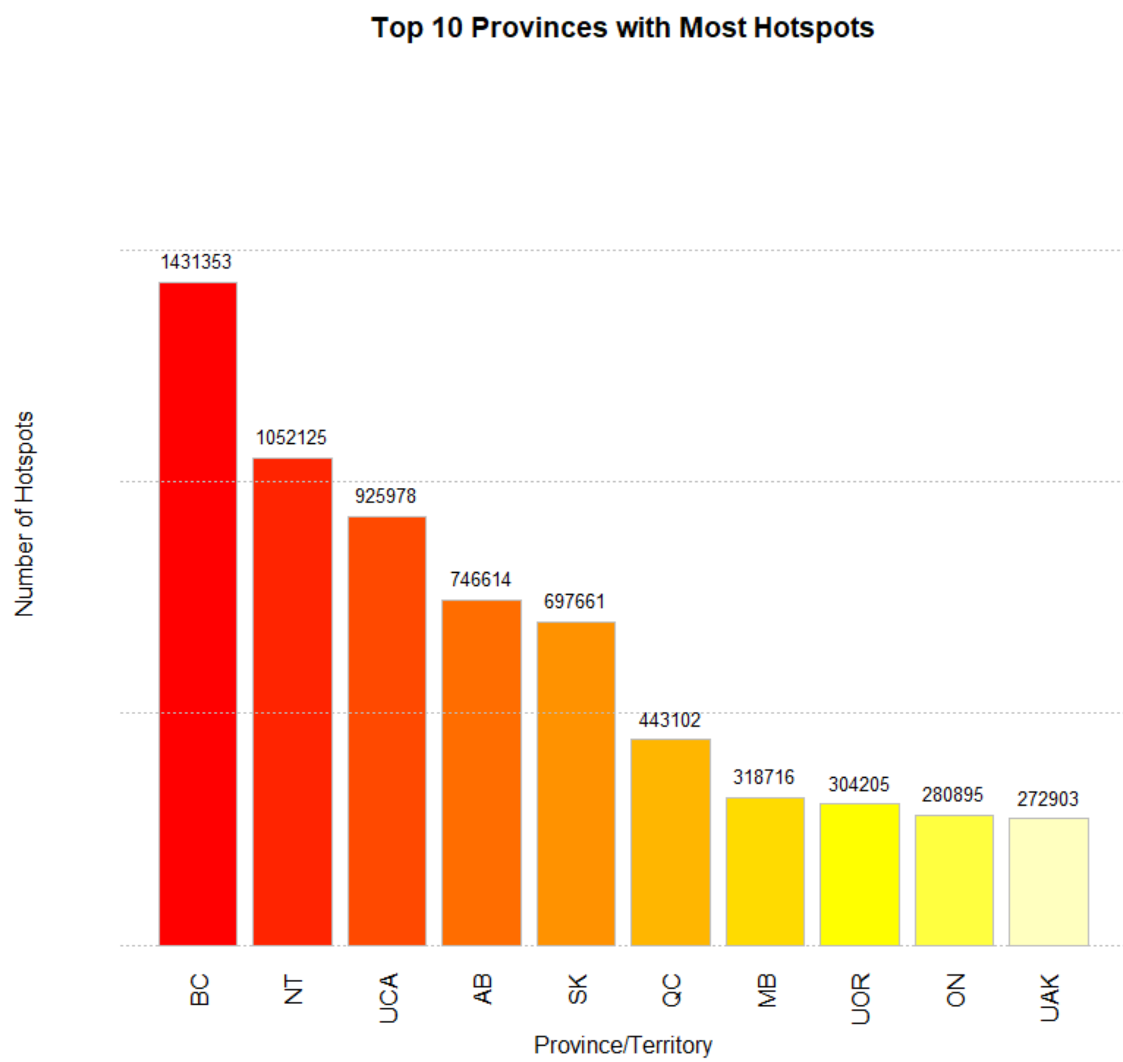
LET WILDFIRES NEVER HAPPEN AGAIN

**Appendix**

```
install.packages(c("dplyr", "tidyr", "ggplot2", "lubridate"))

library(dplyr)
library(tidyr)
library(ggplot2)
library(sf)
library(lubridate)

# Path of the folder containing the CSV files
path <- "D4800_Proj2_Data/"

# Get a list of all CSV files
files <- list.files(path, pattern = "*.csv", full.names = TRUE)

# Function to read each CSV file and normalize column names
read_and_normalize <- function(file) {
  data <- read.csv(file)  # Read the CSV file
  colnames(data) <- tolower(colnames(data))  # Convert column
names to lowercase
  return(data)  # Return the normalized data frame
}

# Load all data sets and combine them into one
combined_data <- bind_rows(lapply(files, read_and_normalize))

#----------------- DATA CLEANING ----------------------

# Remove records with NA values for latitude and longitude
combined_data <- combined_data %>%
  filter(!is.na(lat), !is.na(lon))
```

```
# Handle varying formats dynamically
combined_data$rep_date <- parse_date_time(
  combined_data$rep_date,
  orders = c("Y-m-d H:M:S", "Y-m-d H:M:S.OS", "Y-m-d H:M", "Y-m-
d")
)

# Clean agency non-response
combined_data$agency[combined_data$agency == "-"] <- NA

# Replace "" and "unknown" values in the fuel column with NA
combined_data$fuel[combined_data$fuel == ""] <- NA
combined_data$fuel[combined_data$fuel == "unknown"] <- NA



#-------------------- Exploratory Data Analysis ------------------
-

# Count the number of hotspots per province/territory
province_counts <-
table(combined_data$agency[!is.na(combined_data$agency)])
# Sort the province counts in descending order
sorted_province_counts <- sort(province_counts, decreasing = TRUE)
# Select the top 10 provinces
top_10_provinces <- sorted_province_counts[1:10]

# Calculate a dynamic ylim to give extra space above the bars
max_value <- max(top_10_provinces)
ylim_value <- max_value * 1.3
```

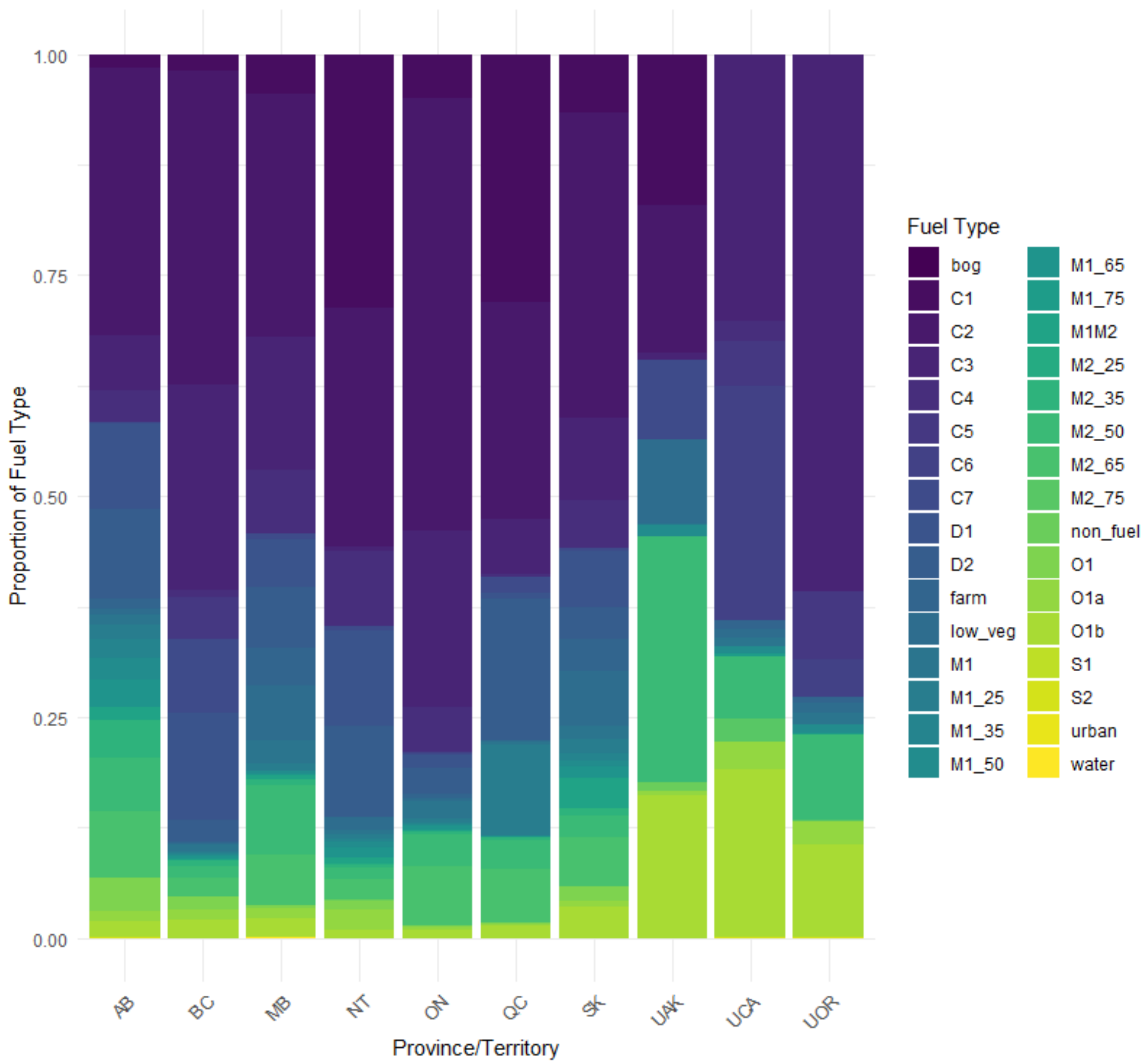
```
# Create the bar plot for the top 10 provinces
bar_positions <- barplot(top_10_provinces,
                         main = "Top 10 Provinces with Most
Hotspots",  # Add a descriptive title
                         xlab = "Province/Territory",
# Label the x-axis
                         ylab = "Number of Hotspots",
# Label the y-axis
                         col = heat.colors(10),
# Use a gradient color palette
                         las = 2,
# Rotate x-axis labels for better readability
                         border = "gray",
# Subtle border color
                         ylim = c(0, ylim_value),
# Dynamically set Y-axis limit
                         yaxt = "n")
# Suppress Y-axis values


# Add a grid for better readability
grid(nx = NA, ny = NULL, col = "gray", lty = "dotted")


# Add exact counts on top of the bars with proper alignment
text(x = bar_positions,
     y = top_10_provinces,
     labels = top_10_provinces,
     pos = 3, offset = 0.5, cex = 0.8, col = "black")  # Offset
added for better visibility
```



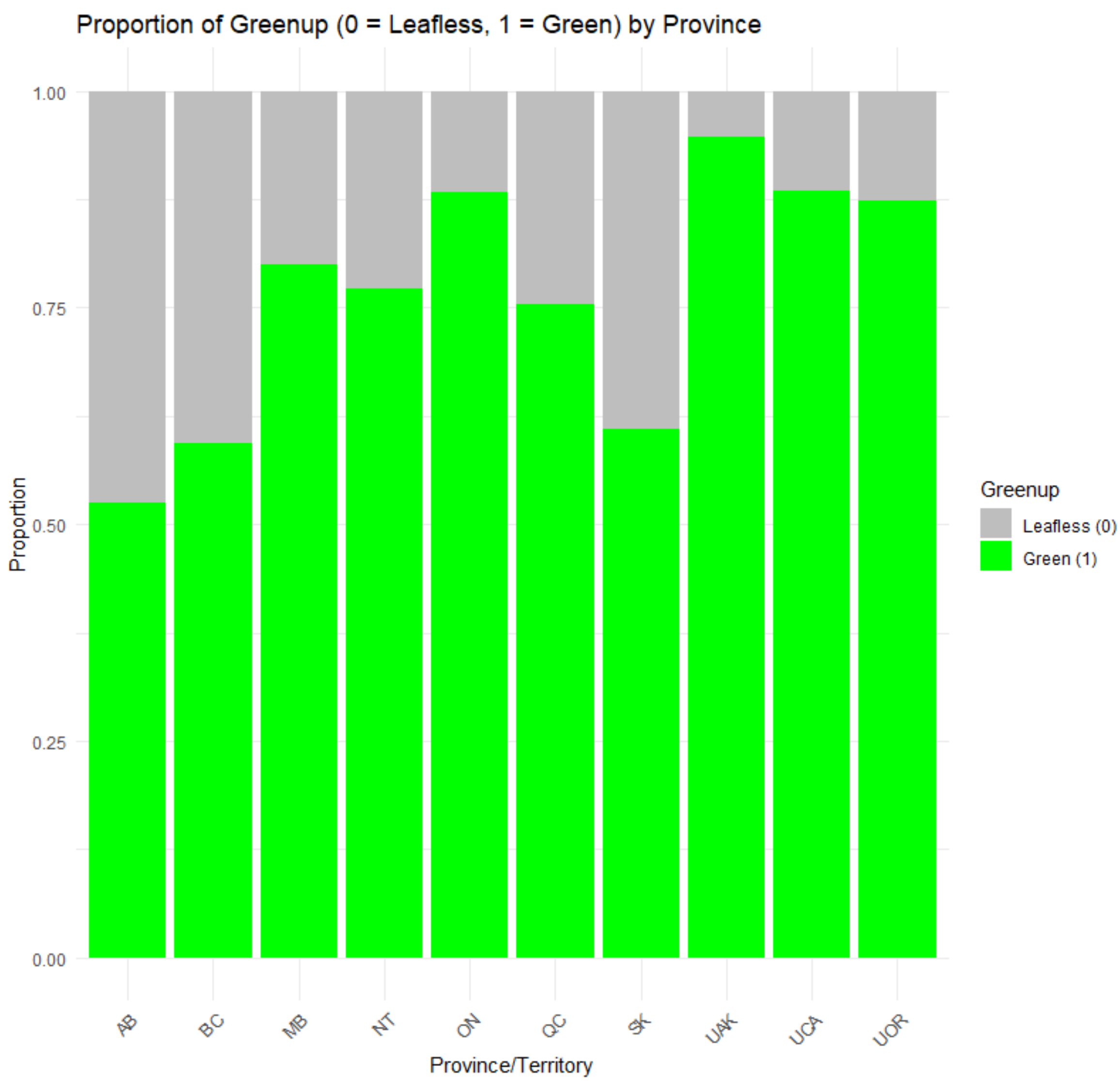

```
# Filter data for the top 10 provinces
top_province_names <- names(top_10_provinces)

# Get the top 10 provinces/territories with the most hotspots
top_10_agencies <- names(top_10_provinces)

filtered_data <- combined_data %>%
  filter(agency %in% top_10_agencies)
```

```r
# Convert the 'fuel' column to a factor and drop unused levels
filtered_data$fuel <- factor(filtered_data$fuel)
filtered_data$fuel <- droplevels(filtered_data$fuel)

# Create a contingency table between 'agency' and 'fuel' for the top 10
contingency_table <- table(filtered_data$agency, filtered_data$fuel)

# Perform the chi-square test
chi_square_test <- chisq.test(contingency_table)

# Print the test results
print("Chi-Square Test Results:")
print(chi_square_test)

# Normalize fuel type counts by calculating the proportion for each agency
fuel_type_proportions <- combined_data %>%
  filter(agency %in% top_province_names, !is.na(fuel)) %>%
  group_by(agency, fuel) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(agency) %>%
  mutate(proportion = count / sum(count)) %>%
  ungroup()

# Stacked bar chart for normalized fuel type proportions
ggplot(fuel_type_proportions, aes(x = agency, y = proportion, fill = fuel)) +
  geom_bar(stat = "identity") +
  scale_fill_viridis_d() +
  labs(
```

```
    title = "Normalized Fuel Type Distribution in Top 10
Provinces",
    x = "Province/Territory",
    y = "Proportion of Fuel Type",
    fill = "Fuel Type"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



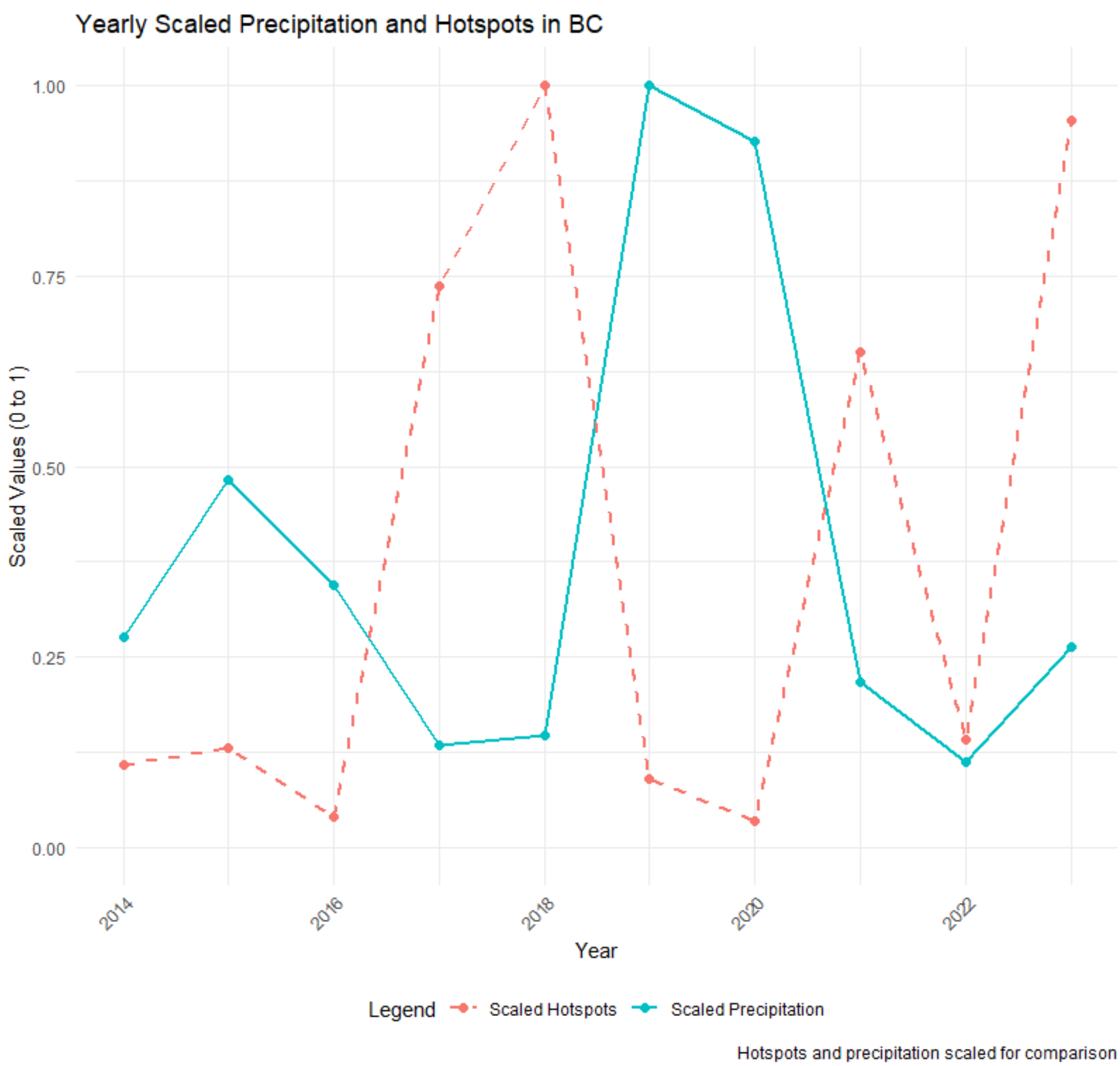
```
# Calculate the proportion of 0 and 1 greenup for each province
greenup_proportions <- combined_data %>%
  filter(agency %in% names(top_10_provinces)) %>%
  filter(!is.na(greenup)) %>%
  group_by(agency, greenup) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(agency) %>%
  mutate(proportion = count / sum(count))


ggplot(greenup_proportions, aes(x = agency, y = proportion, fill =
as.factor(greenup))) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("0" = "gray", "1" = "green"),
labels = c("Leafless (0)", "Green (1)")) +
  labs(
    title = "Proportion of Greenup (0 = Leafless, 1 = Green) by
Province",
    x = "Province/Territory",
    y = "Proportion",
    fill = "Greenup"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Filter the data for British Columbia (BC)
bc_data <- combined_data %>%
  filter(agency == "BC") %>%
  mutate(
    year_month = format(rep_date, "%Y-%m")  # Extract year and
month in "YYYY-MM" format
  )

# Extract the year from the 'rep_date' column
bc_data <- bc_data %>%
```

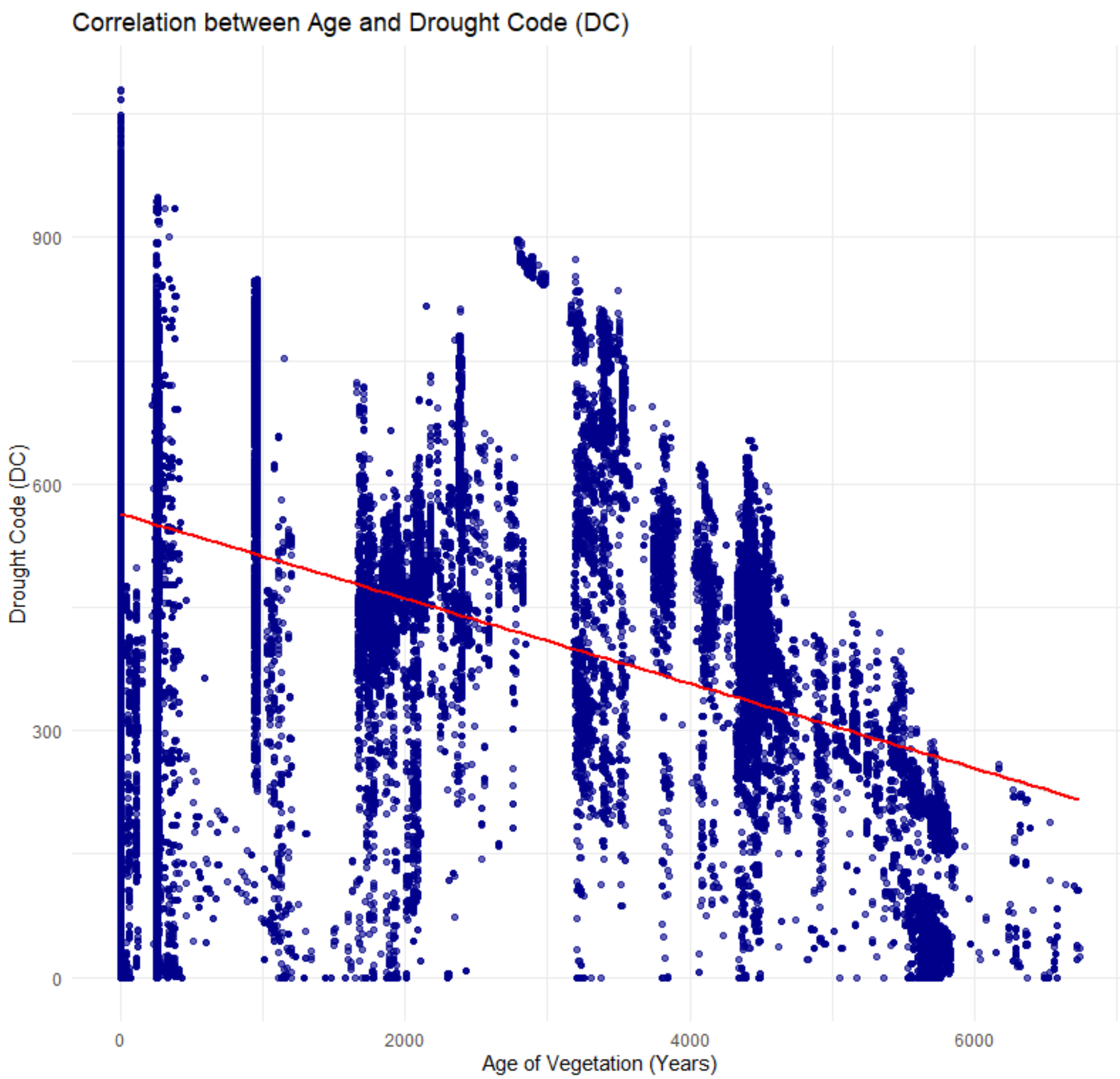
```r
  mutate(year = format(rep_date, "%Y"))

# Calculate the average PCP and the number of hotspots per year
bc_avg_yearly <- bc_data %>%
  group_by(year) %>%
  summarize(
    average_pcp = mean(pcp, na.rm = TRUE),   # Average
precipitation
    average_hotspots = n()                   # Total hotspots
(count of rows)
  )

bc_avg_yearly <- bc_avg_yearly %>%
  mutate(
    scaled_hotspots = average_hotspots / max(average_hotspots),
    scaled_pcp = average_pcp / max(average_pcp)  # Scale
precipitation to [0, 1]
  )

# Ensure 'year' is numeric for proper plotting
bc_avg_yearly <- bc_avg_yearly %>%
  mutate(year = as.numeric(year))  # Convert 'year' to numeric

# Create the plot
ggplot(bc_avg_yearly, aes(x = year)) +
  # Line plot for scaled precipitation
  geom_line(aes(y = scaled_pcp, color = "Scaled Precipitation"),
size = 1, linetype = "solid") +
  geom_point(aes(y = scaled_pcp, color = "Scaled Precipitation"),
size = 2) +
  # Line plot for scaled hotspots
```

```
  geom_line(aes(y = scaled_hotspots, color = "Scaled Hotspots"), 
size = 1, linetype = "dashed") +

  geom_point(aes(y = scaled_hotspots, color = "Scaled Hotspots"), 
size = 2) +

  # Labels and title

  labs(

    title = "Yearly Scaled Precipitation and Hotspots in BC",

    x = "Year",

    y = "Values (Scaled to 0-1)",

    color = "Legend",

    caption = "Hotspots and precipitation scaled for comparison"

  ) +

  scale_y_continuous(

    limits = c(0, 1),  # Ensure both metrics are displayed in the 
same range

    name = "Scaled Values (0 to 1)"

  ) +

  theme_minimal() +

  theme(

    axis.text.x = element_text(angle = 45, hjust = 1),

    axis.title.y = element_text(color = "black"),

    legend.position = "bottom"

  )
```




```
# Filter data for valid DC and age

dc_age_data <- combined_data %>%

  filter(!is.na(age), !is.na(dc), agency == "BC")  # Remove rows
with missing DC or age


# Scatter plot: DC vs Age

# Scatter plot: DC vs Age with regression line

ggplot(dc_age_data, aes(x = age, y = dc)) +

  geom_point(alpha = 0.6, color = "darkblue") +  # Scatter points
```

```
  geom_smooth(method = "lm", color = "red", se = TRUE) +  # Regression line with confidence interval
  labs(
    title = "Correlation between Age and Drought Code (DC)",
    x = "Age of Vegetation (Years)",
    y = "Drought Code (DC)"
  ) +
  theme_minimal()
```

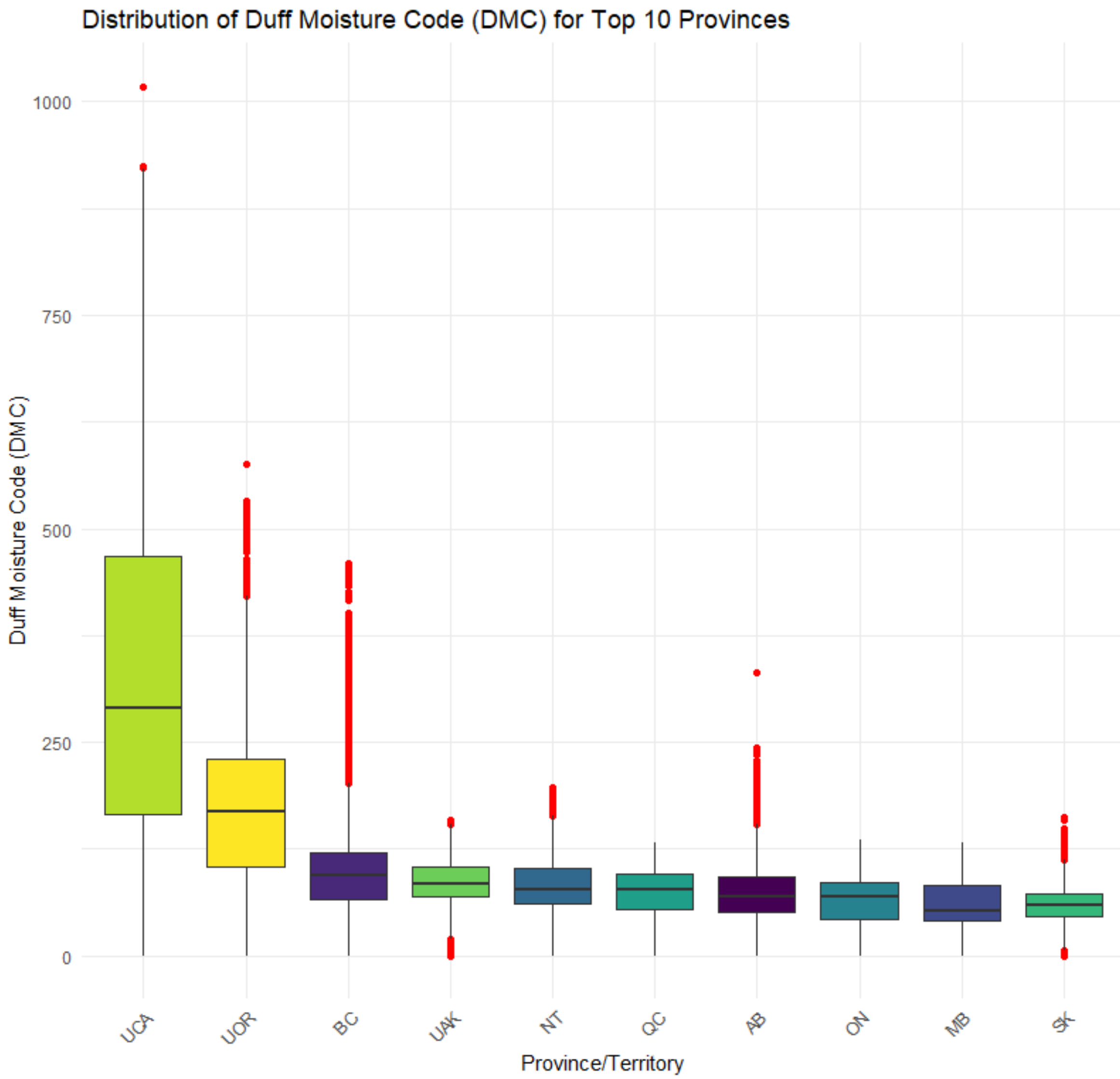



```
# Print correlation coefficient
```

```r
correlation_age_dc <- cor(dc_age_data$age, dc_age_data$dc, use = "complete.obs")

print(paste("Correlation coefficient between Age and Drought Code (DC):", round(correlation_age_dc, 2)))
```

```
 [1] "Correlation coefficient between Age and Drought Code: -0.4"
```

```r
correlation_age_tfc <- cor(tfc_age_data$age, tfc_age_data$tfc, use = "complete.obs")

print(paste("Correlation coefficient between Age and TFC:", round(correlation_age_tfc, 2)))


# Filter the data for the top 10 provinces and non-NA DMC values

dmc_top_provinces <- combined_data %>%

  filter(agency %in% top_province_names, !is.na(dmc))  # Filter for top provinces and non-NA DMC


# Create a boxplot for DMC by province

ggplot(dmc_top_provinces, aes(x = reorder(agency, -dmc), y = dmc, fill = agency)) +

  geom_boxplot(show.legend = FALSE, outlier.color = "red", outlier.size = 1.5) +  # Boxplot with outlier styling

  scale_fill_viridis_d() +  # Use a visually appealing color palette

  labs(

    title = "Distribution of Duff Moisture Code (DMC) for Top 10 Provinces",

    x = "Province/Territory",

    y = "Duff Moisture Code (DMC)"

  ) +

  theme_minimal() +

  theme(

    axis.text.x = element_text(angle = 45, hjust = 1)  # Rotate x-axis labels for readability

  )
```


Distribution of Duff Moisture Code (DMC) for Top 10 Provinces
Duff Moisture Code (DMC)
1000
750
500
250
0
UCA
UOR
BC
UAK
NT
QC
AB
ON
MB
SK
Province/Territory