Wildfire Network Analysis

Atharva Jagtap, Ishika Agarwal, Mariya Putwa, Prateek Balani

2024-11-11

Big network plot

```
# Load necessary libraries
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
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
library(geosphere)
library(ggplot2)
library(sf)
```

```
## Linking to GEOS 3.10.2, GDAL 3.4.1, PROJ 8.2.1; sf_use_s2() is TRUE
```

```r
library(RColorBrewer)
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(stringr)
library(dbscan)
```

```
##
## Attaching package: 'dbscan'

## The following object is masked from 'package:stats':
##
##     as.dendrogram
```

```r
library(sf)
library(rnaturalearth)
library(rnaturalearthdata)
```

```
##
## Attaching package: 'rnaturalearthdata'

## The following object is masked from 'package:rnaturalearth':
##
##     countries110
```

```r
library(ggspatial)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:viridis':
##
##     viridis_pal
```

```r
library(canadianmaps)
library(rgeoboundaries)
```

```
## Registered S3 method overwritten by 'hoardr':
##   method           from
##   print.cache_info httr
```

```r
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:igraph':
##
##     crossing
```

```r
library(tibble)
```

```
##
## Attaching package: 'tibble'

## The following object is masked from 'package:igraph':
##
##     as_data_frame
```

```r
library(ggraph)
library(tidygraph)
```

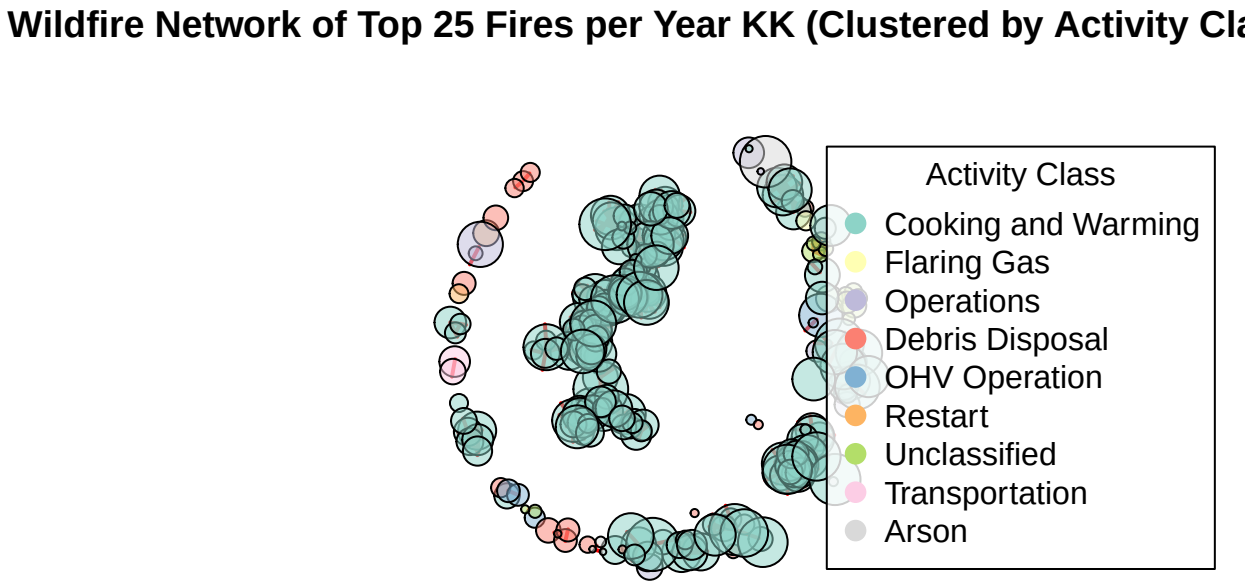
```
##
## Attaching package: 'tidygraph'

## The following object is masked from 'package:igraph':
##
##     groups

## The following object is masked from 'package:stats':
##
##     filter
```

Initial analysis upto Midpoint Report

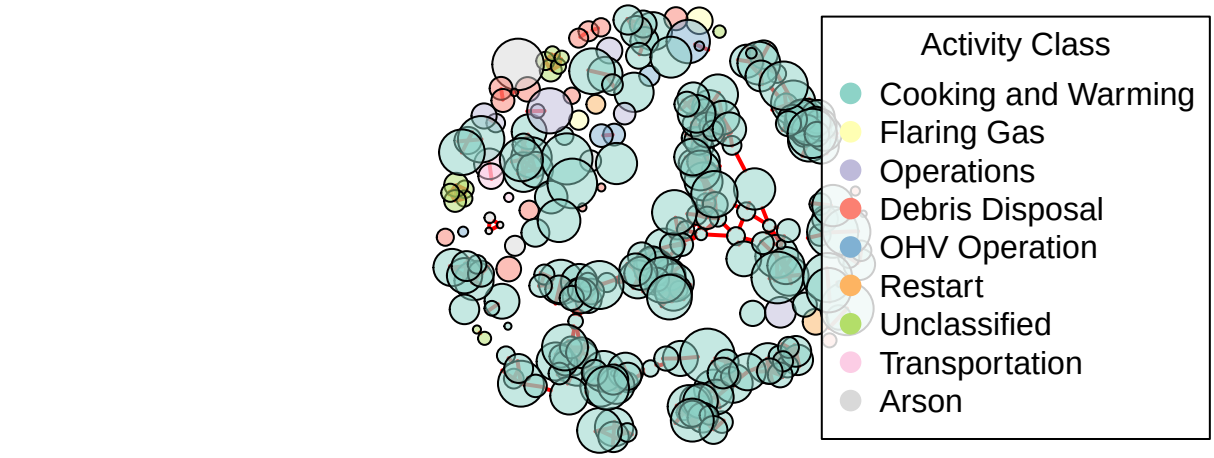
```r
wildfire_data <- read.csv("data_clean.csv")

# Filter top 100 fires per year based on `current_size`
top_fires <- wildfire_data %>%
  group_by(fire_year) %>%
  arrange(desc(current_size)) %>%
  slice_head(n = 25) %>%
  ungroup()

# Create a unique identifier by combining `fire_year` and `fire_number`
top_fires <- top_fires %>%
  mutate(unique_id = paste(fire_year, fire_number, sep = "_"))

# Prepare nodes (distinct fires with geographical coordinates)
nodes <- top_fires %>%
  select(unique_id, fire_location_latitude, fire_location_longitude, current_size, fire_year, activity_
  distinct()

# Define edges based on geographical proximity within each year
max_distance <- 50000  # Set max distance to 20 km for connecting fires
# Define edges based on geographical proximity within each `activity_class`
edges <- data.frame(from = character(), to = character(), stringsAsFactors = FALSE)

# Loop to create edges within each `activity_class` group based on proximity
```

```
for (class in unique(top_fires$activity_class)) {
  # Filter data for the current activity class and remove rows with missing coordinates
  class_data <- top_fires %>%
    filter(activity_class == class) %>%
    filter(!is.na(fire_location_latitude) & !is.na(fire_location_longitude))

  # Skip if no rows remain after filtering
  if (nrow(class_data) < 2) {
    next  # Skip this iteration if there aren't enough points to form edges
  }

  # Nested loop to calculate distances and create edges
  for (i in 1:(nrow(class_data) - 1)) {
    for (j in (i + 1):nrow(class_data)) {
      # Extract the coordinates as pairs
      coord_i <- c(class_data$fire_location_longitude[i], class_data$fire_location_latitude[i])
      coord_j <- c(class_data$fire_location_longitude[j], class_data$fire_location_latitude[j])

      # Check if coordinates are valid pairs of length 2
      if (length(coord_i) == 2 && length(coord_j) == 2) {
        # Calculate distance between points
        dist <- distHaversine(coord_i, coord_j)

        # Add edge if distance is within the maximum allowed distance
        if (!is.na(dist) && dist <= max_distance) {
          edges <- rbind(edges, data.frame(from = class_data$unique_id[i], to = class_data$unique_id[j]
        }
      }
    }
  }
}


# Filter edges to ensure IDs are in nodes
edges <- edges %>%
  filter(from %in% nodes$unique_id & to %in% nodes$unique_id)

# Create the graph from edges and nodes
g <- graph_from_data_frame(d = edges, vertices = nodes, directed = FALSE)

V(g)$activity_class <- nodes$activity_class

# Map `activity_class` to colors
activity_classes <- unique(nodes$activity_class)
color_palette <- brewer.pal(n = length(activity_classes), name = "Set3")  # Customize with any color pa
activity_colors <- setNames(color_palette, activity_classes)  # Map each activity_class to a color


# Apply colors to nodes based on `activity_class`
V(g)$color <- activity_colors[V(g)$activity_class]

# Node sizes based on `current_size`
V(g)$size <- log(nodes$current_size + 1) * 2
```

```r
# Adjust node colors for transparency
node_colors <- adjustcolor(V(g)$color, alpha.f = 0.5)  # 70% opacity

# Adjust edge color for transparency
edge_color <- adjustcolor("red", alpha.f = 1)

# Plot the network with Kamada-Kawai layout
layout_kk <- layout_with_kk(g)
plot(g,
     layout = layout_kk,
     vertex.size = V(g)$size,
     vertex.color = node_colors,  # Transparent node colors
     vertex.label = NA,
     edge.color = edge_color,      # Transparent edge color
     edge.width = 2,
     main = "Wildfire Network of Top 25 Fires per Year KK (Clustered by Activity Class)")

legend("right",
       legend = names(activity_colors),
       col = activity_colors,
       pch = 16,
       pt.cex = 1.5,   # Increase point size in legend for visibility
       title = "Activity Class",
       xpd = TRUE,
       inset = -0.05,
       bg = adjustcolor("white", alpha.f = 0.8))  # Transparent legend background
```




```
layout_fr <- layout_with_fr(g)
plot(g,
     layout = layout_fr,
     vertex.size = V(g)$size,
     vertex.color = node_colors,
     vertex.label = NA,
     edge.color = edge_color,
     edge.width = 2,   # Keep a higher width for better visibility
     main = "Wildfire Network of Top 25 Fires per Year FR (Clustered by Activity Class)")

legend("right",
       legend = names(activity_colors),
       col = activity_colors,
       pch = 16,
       pt.cex = 1.5,   # Increase point size in legend for visibility
       title = "Activity Class",
       xpd = TRUE,
       inset = -0.05,
       bg = adjustcolor("white", alpha.f = 0.8))  # Transparent legend background
```




```
cat("Number of edges in graph:", ecount(g), "\n")
```

```
## Number of edges in graph: 756
```

```
cat("Number of vertices in graph:", vcount(g), "\n")
```

```
## Number of vertices in graph: 275
```

```
# Data Analysis on the given data
Degree_centrality <- degree(g)
Closeness_centrality <- closeness(g)
Betweenness_centrality <- betweenness(g)
Eigenvector_centrality <- evcent(g)$vector
```

```
## Warning: `evcent()` was deprecated in igraph 2.0.0.
## i Please use `eigen_centrality()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
Clustering_coefficient <- transitivity(g, type = "local")

# Combine centrality measures into a data frame
```

```r
centrality_measures <- data.frame(
  node = V(g)$name,
  degree = Degree_centrality,
  closeness = Closeness_centrality,
  betweenness = Betweenness_centrality,
  eigenvector = Eigenvector_centrality,
  clustering = Clustering_coefficient
)

summary(centrality_measures)
```

```
##      node               degree         closeness        betweenness
##  Length:275         Min.   : 0.000   Min.   :0.00056   Min.   :   0.000
##  Class :character   1st Qu.: 2.000   1st Qu.:0.00082   1st Qu.:   0.000
##  Mode  :character   Median : 4.000   Median :0.00108   Median :   1.186
##                     Mean   : 5.498   Mean   :0.13781   Mean   : 266.655
##                     3rd Qu.: 9.000   3rd Qu.:0.16667   3rd Qu.: 129.000
##                     Max.   :18.000   Max.   :1.00000   Max.   :4389.000
##                                      NA's   :37
##   eigenvector        clustering
##  Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.6212
##  Median :0.00000   Median :0.7694
##  Mean   :0.05848   Mean   :0.7485
##  3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.0000
##                    NA's   :65
```

```r
# Top 5 nodes by degree centrality
top_degree <- centrality_measures[order(-centrality_measures$degree), ]
head(top_degree, 5)
```

```
##                    node degree  closeness betweenness eigenvector clustering
## 2019_MWF055 2019_MWF055     18 0.02631579   24.871861   1.0000000  0.7254902
## 2022_MWF022 2022_MWF022     18 0.02631579   24.871861   1.0000000  0.7254902
## 2019_MWF054 2019_MWF054     17 0.02564103   19.405952   0.9535935  0.7426471
## 2019_MWF063 2019_MWF063     17 0.02173913    6.370671   0.9729334  0.7720588
## 2022_MWF034 2022_MWF034     17 0.02173913    6.370671   0.9729334  0.7720588
```

```r
# Top 5 nodes by closeness centrality
top_closeness <- centrality_measures[order(-centrality_measures$closeness), ]
head(top_closeness, 5)
```

```
##                    node degree closeness betweenness  eigenvector clustering
## 2013_HWF037 2013_HWF037      1         1           0 5.820468e-18        NaN
## 2014_GWF044 2014_GWF044      1         1           0 1.762243e-18        NaN
## 2014_GWF043 2014_GWF043      1         1           0 9.998698e-18        NaN
## 2014_GWF018 2014_GWF018      1         1           0 6.867599e-18        NaN
## 2015_LWF122 2015_LWF122      1         1           0 5.355207e-18        NaN
```

```r
# Top 5 nodes by betweenness centrality
top_betweenness <- centrality_measures[order(-centrality_measures$betweenness), ]
head(top_betweenness, 5)
```

```
##                     node degree    closeness betweenness  eigenvector clustering
## 2014_MWF008 2014_MWF008      7 0.0011325028    4389.000 1.284586e-16  0.4285714
## 2021_MWF025 2021_MWF025      9 0.0011025358    3934.286 1.966527e-16  0.6111111
## 2017_SWF107 2017_SWF107     12 0.0010604454    3845.150 2.497806e-16  0.5303030
## 2013_HWF062 2013_HWF062     10 0.0011547344    3290.194 1.522472e-16  0.5555556
## 2013_MWF009 2013_MWF009      5 0.0009354537    3193.000 1.338110e-16  0.4000000
```

```r
# Top 5 nodes by eigenvector centrality
top_eigenvector <- centrality_measures[order(-centrality_measures$eigenvector), ]
head(top_eigenvector, 5)
```

```
##                     node degree  closeness betweenness eigenvector clustering
## 2019_MWF055 2019_MWF055     18 0.02631579   24.871861   1.0000000  0.7254902
## 2022_MWF022 2022_MWF022     18 0.02631579   24.871861   1.0000000  0.7254902
## 2019_MWF063 2019_MWF063     17 0.02173913    6.370671   0.9729334  0.7720588
## 2022_MWF034 2022_MWF034     17 0.02173913    6.370671   0.9729334  0.7720588
## 2019_MWF054 2019_MWF054     17 0.02564103   19.405952   0.9535935  0.7426471
```

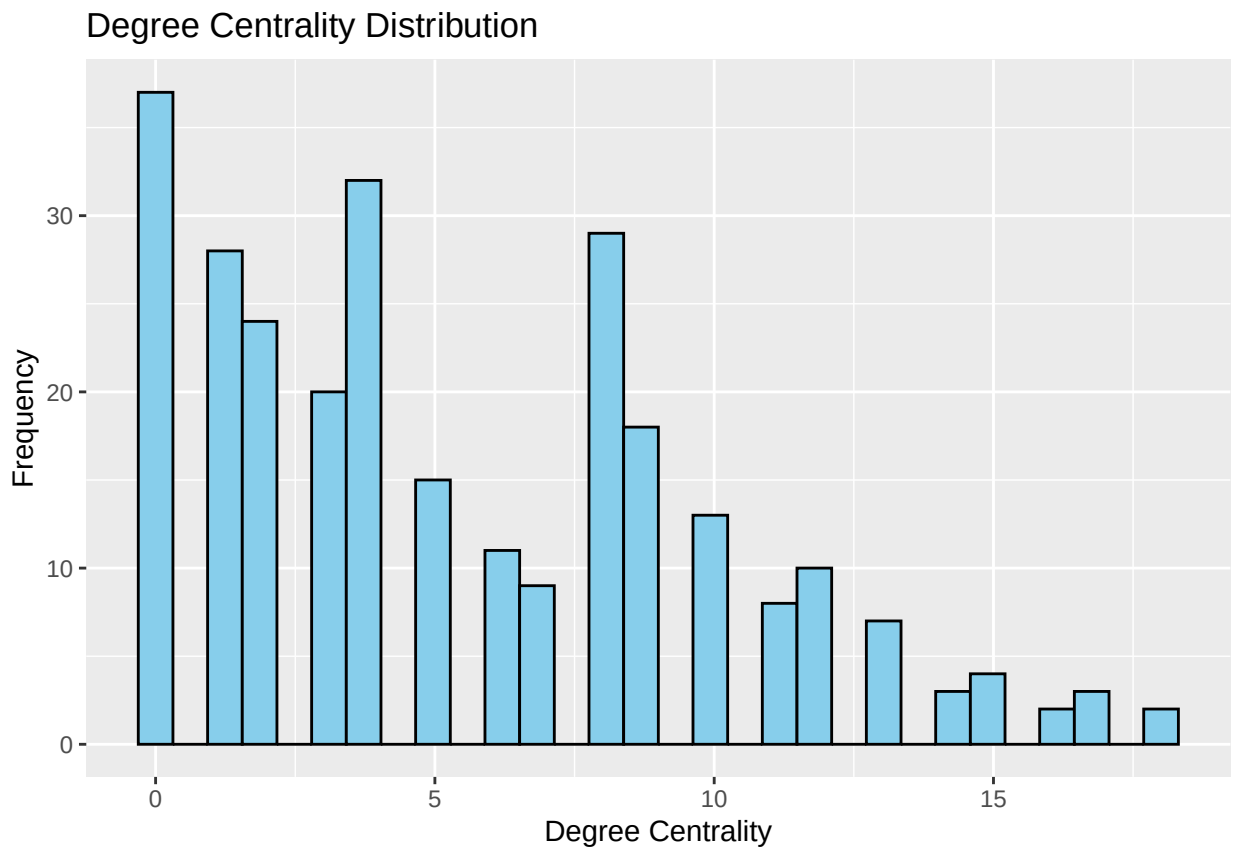
```r
# Top 5 nodes by clustering coefficient
top_clustering <- centrality_measures[order(-centrality_measures$clustering), ]
head(top_clustering, 5)
```

```
##                     node degree closeness betweenness  eigenvector clustering
## 2013_HWF023 2013_HWF023      4 0.2500000           0 1.193019e-16          1
## 2013_HWF050 2013_HWF050      4 0.2500000           0 1.219775e-16          1
## 2014_RWF034 2014_RWF034      3 0.1428571           0 6.893731e-17          1
## 2014_LWF002 2014_LWF002      4 0.2500000           0 1.521112e-16          1
## 2014_LWF001 2014_LWF001      4 0.2500000           0 1.200423e-16          1
```

```r
# Calculate network density
network_density <- edge_density(g)
cat("\nNetwork Density:", network_density)
```

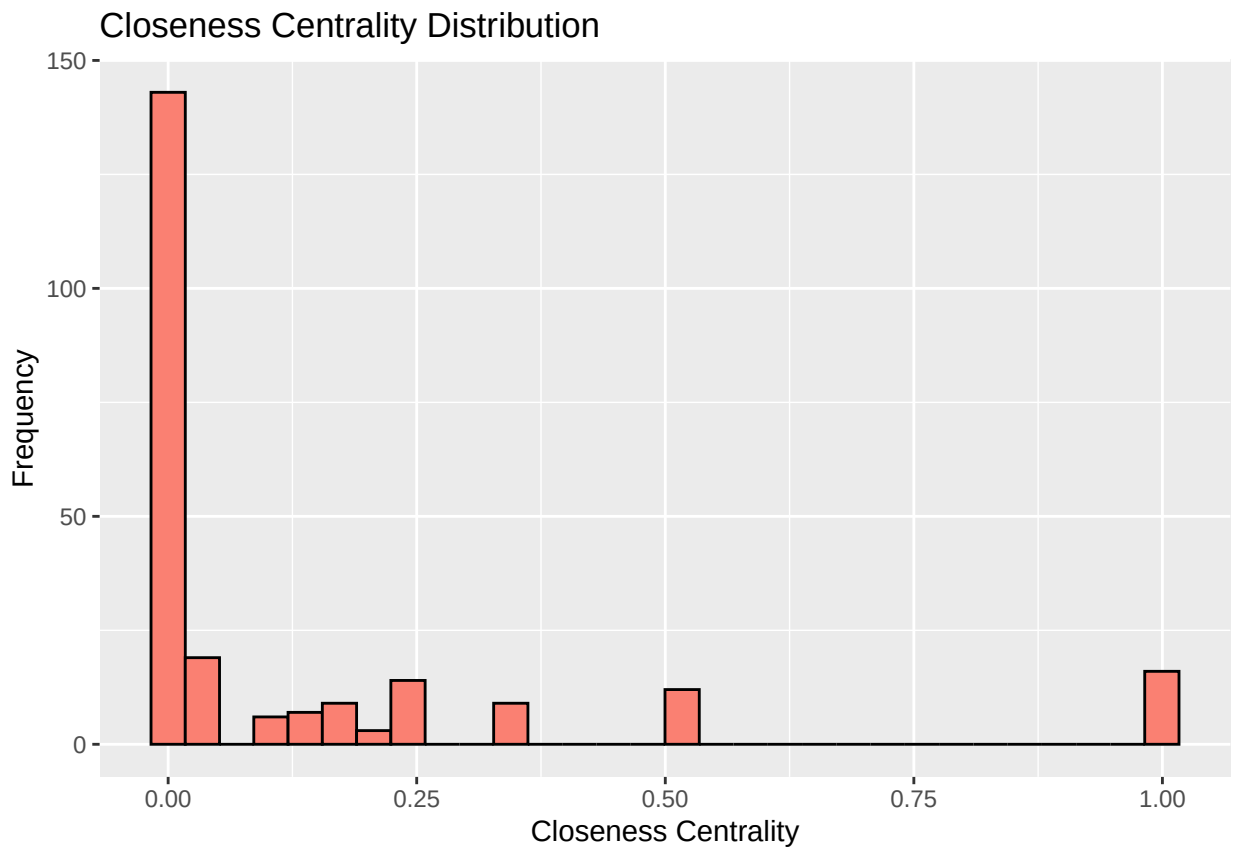
```
##
## Network Density: 0.02006636
```

```r
# 1. Visualize Degree Centrality Distribution
degree_centrality_df <- data.frame(node = names(Degree_centrality), degree = Degree_centrality)
ggplot(degree_centrality_df, aes(x = degree)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  labs(title = "Degree Centrality Distribution", x = "Degree Centrality", y = "Frequency")
```

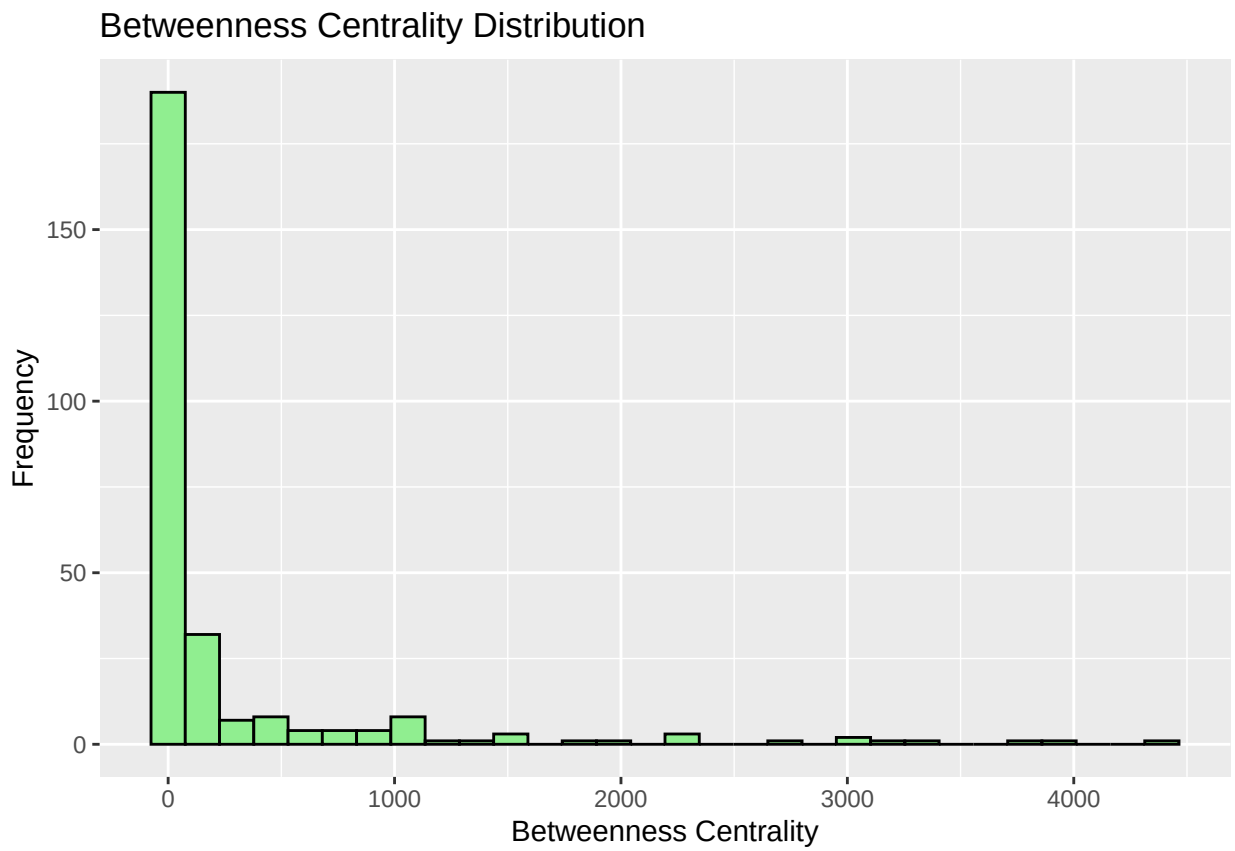



```
# 2. Visualize Closeness Centrality Distribution
closeness_centrality_df <- data.frame(node = names(Closeness_centrality), closeness = Closeness_centrali
ggplot(closeness_centrality_df, aes(x = closeness)) +
  geom_histogram(bins = 30, fill = "salmon", color = "black") +
  labs(title = "Closeness Centrality Distribution", x = "Closeness Centrality", y = "Frequency")
```

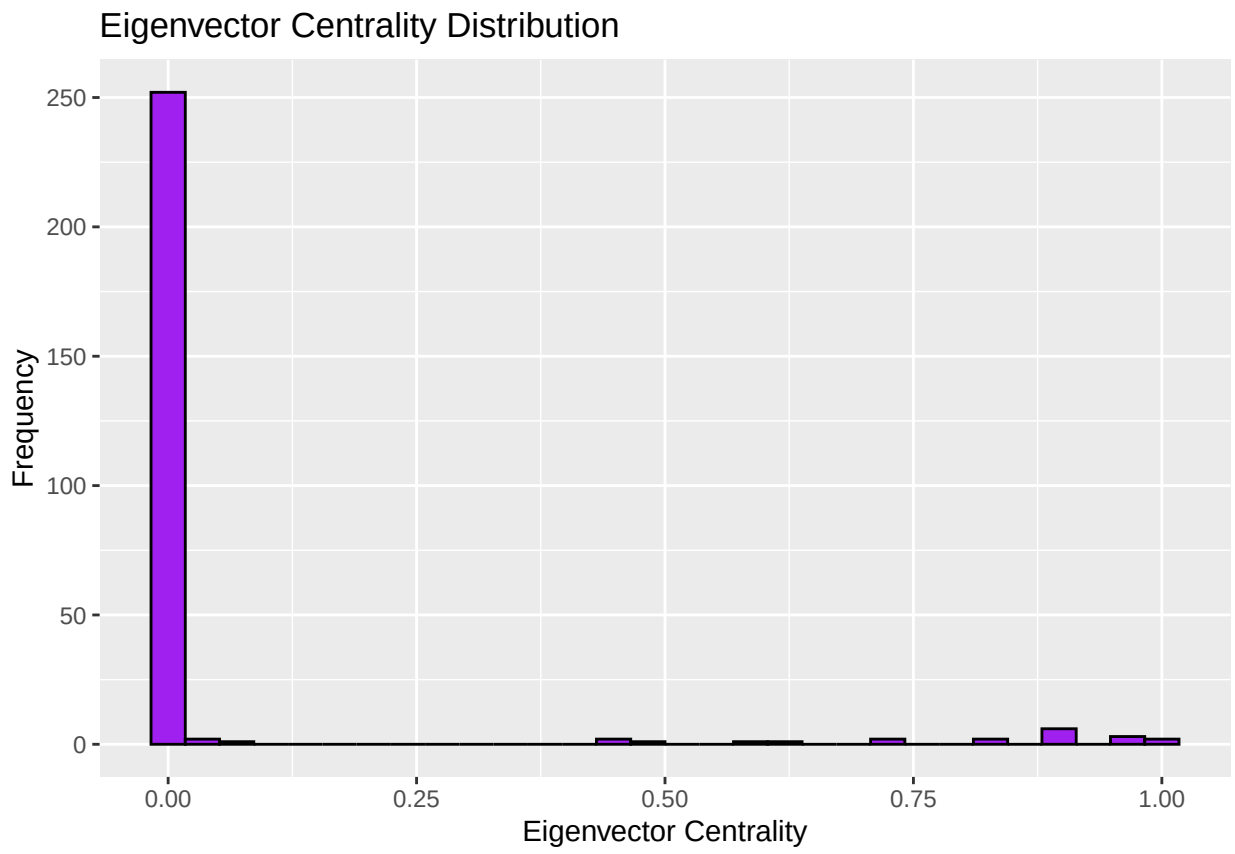
```
## Warning: Removed 37 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```r
# 3. Visualize Betweenness Centrality Distribution
betweenness_centrality_df <- data.frame(node = names(Betweenness_centrality), betweenness = Betweenness
ggplot(betweenness_centrality_df, aes(x = betweenness)) +
  geom_histogram(bins = 30, fill = "lightgreen", color = "black") +
  labs(title = "Betweenness Centrality Distribution", x = "Betweenness Centrality", y = "Frequency")
```




```r
# 4. Visualize Eigenvector Centrality Distribution
eigenvector_centrality_df <- data.frame(node = names(Eigenvector_centrality), eigenvector = Eigenvector
ggplot(eigenvector_centrality_df, aes(x = eigenvector)) +
  geom_histogram(bins = 30, fill = "purple", color = "black") +
  labs(title = "Eigenvector Centrality Distribution", x = "Eigenvector Centrality", y = "Frequency")
```




```
# 5. Community Detection and Visualization
# Detect communities using a community detection algorithm (Louvain)
communities <- cluster_louvain(g)
V(g)$community <- membership(communities)

# Count the total number of communities
total_communities <- length(unique(V(g)$community))
cat("Total number of communities:", total_communities)
```

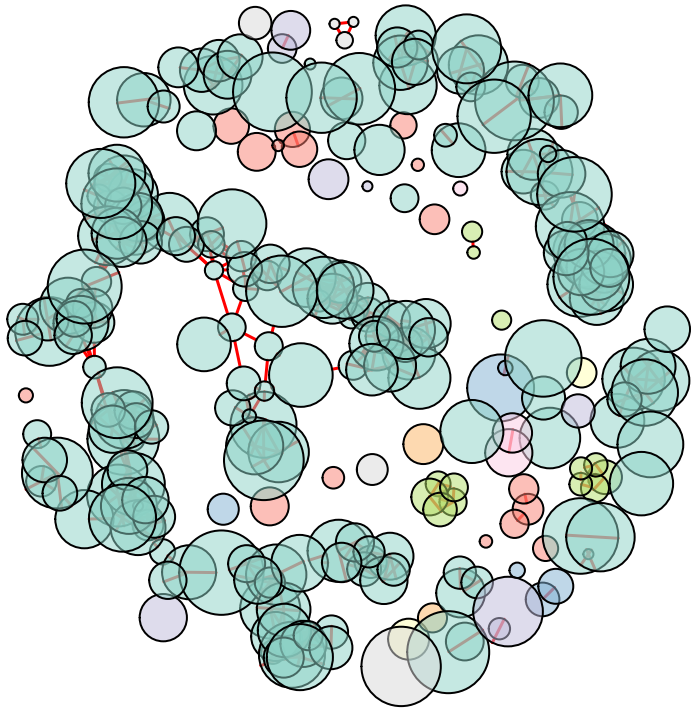
```
## Total number of communities: 69
```

```
# Identify the largest community and its size
community_sizes <- sizes(communities)
largest_community <- which.max(community_sizes)
largest_community_size <- community_sizes[largest_community]
cat("\nSize of the largest community:", largest_community_size)
```

```
##
## Size of the largest community: 25
```

```
# Plot the network with nodes
set.seed(42)  # For reproducible layout
par(mar = c(1, 1, 1, 1))
plot(g,
```

```
    vertex.color = adjustcolor(V(g)$color, alpha.f = 0.5),
    vertex.size = V(g)$size,
    vertex.label = NA,
    edge.width = 1.5,
    edge.color = adjustcolor("red"),
    main = "Wildfire Network with Communities Highlighted")
```



Final Report Analysis, answering our research questions:

```
# Question 1: What are the most common causes of wildfires in different provinces, and how do these cau

#filtering data as to prevent any missing values and in turn creating issues

wildfire_filtered <- wildfire_data %>%
  mutate(across(where(is.character), ~ trimws(.))) %>% # Remove whitespace
  filter(!is.na(fire_location_latitude), !is.na(fire_location_longitude), !is.na(activity_class)) %>%
  mutate(activity_class = as.factor(activity_class))


# Converting wildire coordinates to spatial data and using cors=4326 as it is the most common coordinat
wildfire_sf <- st_as_sf(
  wildfire_filtered,
  coords = c("fire_location_longitude", "fire_location_latitude"),
```

```
  crs = 4326
)

#matrix that stores a point (i.e wildire ) and its coordinates
coords <- st_coordinates(wildfire_sf)


wild_fire_db <- dbscan(coords, eps = 0.2, minPts = 5) #using a small eps value to prevent any overlappi

# Add cluster information to the spatial data
wildfire_sf <- wildfire_sf %>%
  mutate(cluster = as.factor(wild_fire_db$cluster))


# Finding the common causes of wildfires and their average size
common_causes <- wildfire_sf %>%
  group_by(activity_class) %>%
  summarise(
    count = n(),
    avg_fire_size = mean(current_size, na.rm = TRUE)
  ) %>%
  arrange(desc(count))

print("Most Common Causes of Wildfires:")
```

```
## [1] "Most Common Causes of Wildfires:"
```

```
print(common_causes)
```

```
## Simple feature collection with 12 features and 3 fields
## Geometry type: MULTIPOINT
## Dimension:     XY
## Bounding box:  xmin: -120 ymin: 48.9982 xmax: -110.0068 ymax: 59.99951
## Geodetic CRS:  WGS 84
## # A tibble: 12 x 4
##    activity_class      count avg_fire_size                          geometry
##    <fct>               <int>         <dbl>                  <MULTIPOINT [°]>
##  1 Cooking and Warming  8320       479.    ((-114.7449 49.61562), (-114.7439 49~
##  2 Debris Disposal      1716         2.56  ((-114.2199 49.38937), (-114.3645 49~
##  3 Operations           1129        47.3   ((-113.5789 48.9982), (-114.2839 49.~
##  4 Unclassified         1109         3.15  ((-114.4052 49.30995), (-114.2779 49~
##  5 Arson                 712       317.    ((-114.5285 49.62955), (-114.4497 49~
##  6 Transportation        338         6.60  ((-114.4121 49.49233), (-114.4386 49~
##  7 Refuse Disposal       206         0.353 ((-113.9342 49.79343), (-114.4525 50~
##  8 OHV Operation         160       223.    ((-114.548 49.43175), (-113.8897 49.~
##  9 Structure Fire        126         0.403 ((-114.5233 49.63865), (-114.4594 50~
## 10 Flaring Gas           106        56.1   ((-114.7812 52.81767), (-114.9439 52~
## 11 Restart                41        18.7   ((-115.4428 52.37455), (-116.3097 52~
## 12 Prescribed Fire         8         4.84  ((-117.151 56.7101), (-116.1521 55.4~
```

```
# Effect of causes on spread and clustering
spread_and_clustering <- wildfire_sf %>%
```

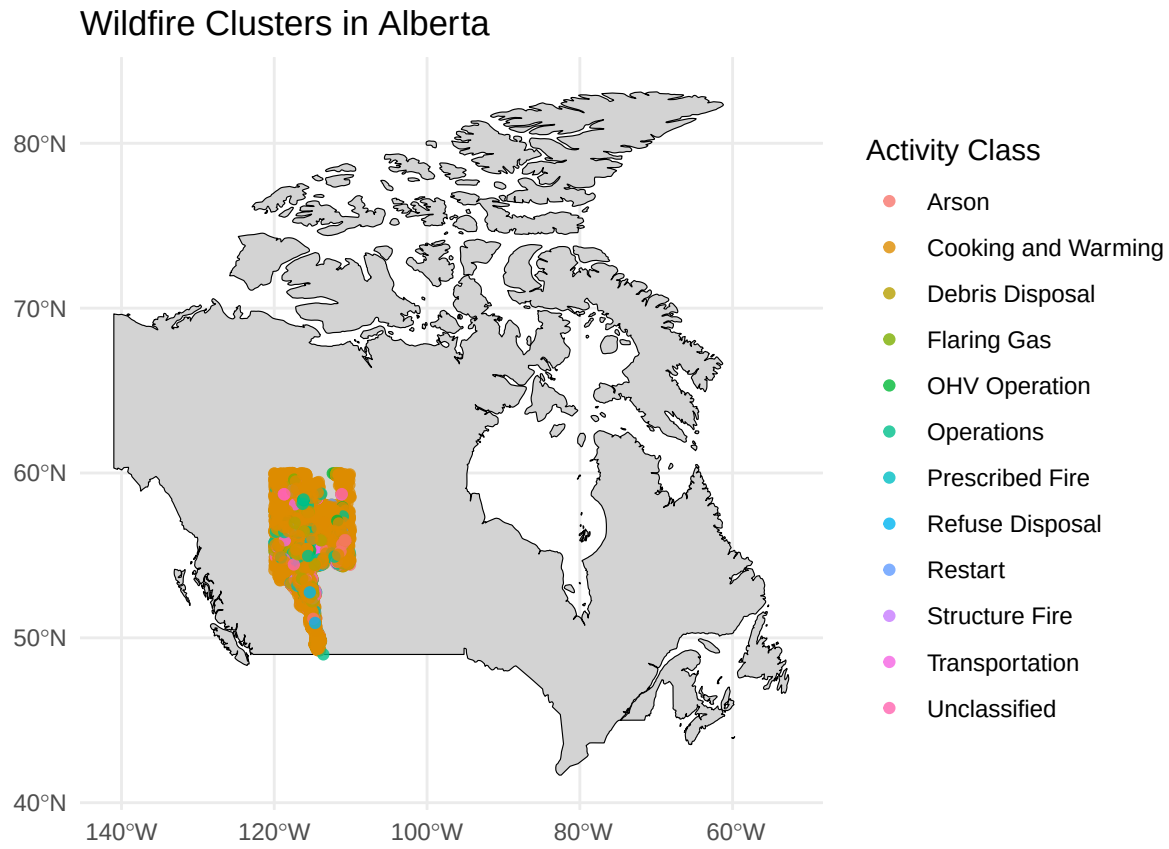
```
  group_by(activity_class) %>%
  summarise(
    avg_spread_rate = mean(fire_spread_rate, na.rm = TRUE),
    avg_cluster_size = mean(current_size, na.rm = TRUE),
    cluster_count = n_distinct(cluster)
  )%>%
  arrange(desc(cluster_count))

print("Effect of Causes on Spread and Clustering:")
```

```
## [1] "Effect of Causes on Spread and Clustering:"
```

```
print(spread_and_clustering)
```

```
## Simple feature collection with 12 features and 4 fields
## Geometry type: MULTIPOINT
## Dimension:     XY
## Bounding box:  xmin: -120 ymin: 48.9982 xmax: -110.0068 ymax: 59.99951
## Geodetic CRS:  WGS 84
## # A tibble: 12 x 5
##    activity_class      avg_spread_rate avg_cluster_size cluster_count
##    <fct>                         <dbl>            <dbl>         <int>
##  1 Cooking and Warming           1.06           479.                8
##  2 Debris Disposal               0.478            2.56              4
##  3 OHV Operation                 1.70           223.                3
##  4 Operations                    0.661           47.3               3
##  5 Unclassified                  0.839            3.15              3
##  6 Arson                         1.08           317.                2
##  7 Refuse Disposal               0.387            0.353             2
##  8 Structure Fire                0.268            0.403             2
##  9 Transportation                0.447            6.60              2
## 10 Flaring Gas                   1.21            56.1               1
## 11 Prescribed Fire               1.2              4.84              1
## 12 Restart                       0.607           18.7               1
## # i 1 more variable: geometry <MULTIPOINT [°]>
```

```
# Plotting the wildfire clusters on a map of Canada
canada_map <- ne_countries(scale = "medium", returnclass = "sf") %>%
  filter(admin == "Canada")

# Plot wildfire clusters on the map
ggplot() +
  geom_sf(data = canada_map, fill = "lightgray", color = "black") + # Map of Canada
  geom_sf(data = wildfire_sf, aes(color = activity_class), alpha = 0.8) + # Wildfire points
  labs(
    title = "Wildfire Clusters in Alberta",
    color = "Activity Class"
  ) +
  theme_minimal()
```

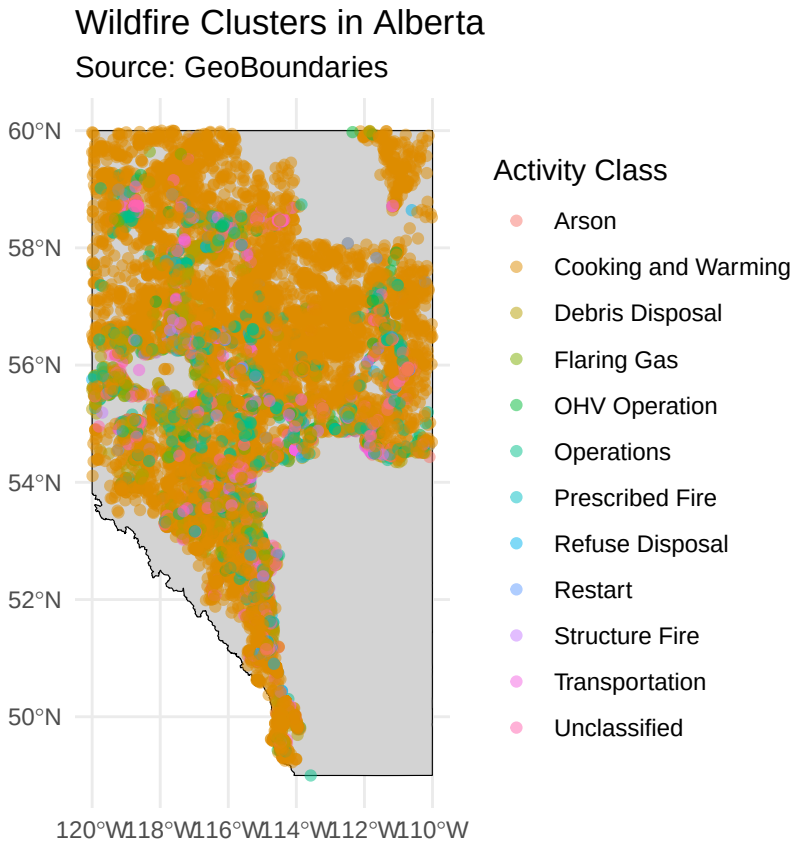



```
# Plotting the wildfire clusters on a map of Alberta

alberta_boundary <- st_read("../Network-Science-Team-4/geoBoundaries-CAN-ADM1-all/geoBoundaries-CAN-ADM
  filter(shapeName == "Alberta")
```

```
## Reading layer ‘geoBoundaries-CAN-ADM1_simplified’ from data source
##   ‘/home/mariya/Documents/Year 4/COSC 421/Network-Science-Team-4/geoBoundaries-CAN-ADM1-all/geoBounda
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 13 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -141.0181 ymin: 41.68142 xmax: -52.61937 ymax: 83.13699
## Geodetic CRS:  WGS 84
```

```
wildfire_sf <- st_transform(wildfire_sf, st_crs(alberta_boundary))


# Plot wildfire clusters on the map
ggplot() +
  geom_sf(data = alberta_boundary, fill = "lightgray", color = "black") +
  geom_sf(data = wildfire_sf, aes(color = activity_class), alpha = 0.5) + # Wildfire points
  labs(
    title = "Wildfire Clusters in Alberta",
    color = "Activity Class",
    subtitle = "Source: GeoBoundaries"
```

```
  ) +
  theme_minimal()
```

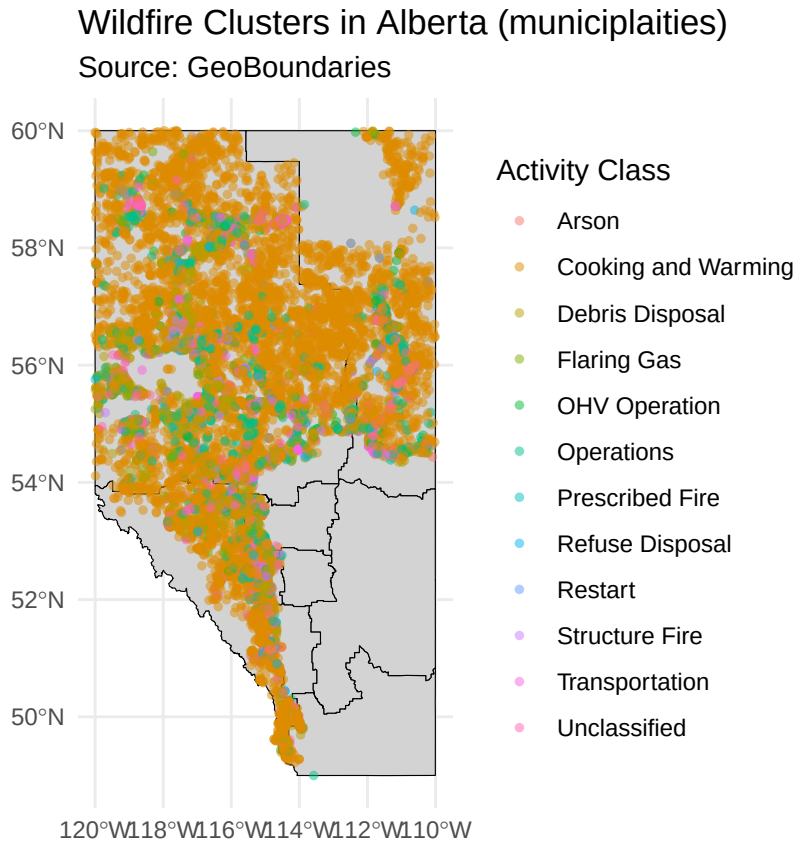



```
# Plotting the wildfire clusters on a map of Alberta (municiplaities)

alberta_municipalities <- st_read("../Network-Science-Team-4/geoBoundaries-CAN-ADM2-all/geoBoundaries-CA
 filter(shapeName %in% c("Calgary", "Edmonton", "Red Deer", "Lethbridge--Medicine Hat",
                         "Camrose--Drumheller", "Athabasca--Grande Prairie--Pe*",
                         "Wood Buffalo--Cold Lake", "Banff--Jasper--Rocky Mountain*"))
```

```
## Reading layer ‘geoBoundaries-CAN-ADM2’ from data source
##   ‘/home/mariya/Documents/Year 4/COSC 421/Network-Science-Team-4/geoBoundaries-CAN-ADM2-all/geoBounda
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 76 features and 5 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -141.0181 ymin: 41.68144 xmax: -52.61941 ymax: 83.1355
## Geodetic CRS:  WGS 84
```

```
wildfire_sf <- st_transform(wildfire_sf, st_crs(alberta_municipalities))

# Plot wildfire clusters on the map
ggplot() +
  geom_sf(data = alberta_municipalities, fill = "lightgray", color = "black", size = 3) +
```

```
  geom_sf(data = wildfire_sf, aes(color = activity_class), alpha = 0.5, size = 1) + # Wildfire points
  labs(
    title = "Wildfire Clusters in Alberta (municiplaities)",
    color = "Activity Class",
    subtitle = "Source: GeoBoundaries"
  ) +
  theme_minimal()
```




```
# Question 4: Which regions experience the most fires, and what are the primary causes by region?
# and some extra stuff

wildfires_with_municipalities <- st_join(wildfire_sf, alberta_municipalities, join = st_within)

wildfires_with_municipalities_df <- wildfires_with_municipalities %>%
  st_drop_geometry() %>%
  distinct(shapeName, activity_class)

fires_by_mun <- wildfires_with_municipalities_df %>%
  group_by(shapeName) %>%
  summarise(total_fires = n(), .groups = "drop") %>%
  arrange(desc(total_fires))

fire_cause_by_mun <- wildfires_with_municipalities_df %>%
  group_by(shapeName, activity_class) %>%
  summarise(count = n(), .groups = "drop") %>%
```

```r
  arrange(desc(count))


# making a adjacency matrix to see the relationship between the wildfires and the municipalities
wildfire_adjacency <- wildfires_with_municipalities_df %>%
  select(shapeName, activity_class) %>%
  distinct() %>%
  inner_join(., ., by = "activity_class") %>%
  filter(shapeName.x != shapeName.y) %>%  # Remove self-joins
  group_by(shapeName.x, shapeName.y) %>%
  summarise(shared_cause_count = n(), .groups = "drop")
```

```
## Warning in inner_join(., ., by = "activity_class"): Detected an unexpected many-to-many relationship
## i Row 1 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 2 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

```r
adj_matrix <- wildfire_adjacency %>%
  pivot_wider(names_from = shapeName.y, values_from = shared_cause_count, values_fill = 0) %>%
  column_to_rownames(var = "shapeName.x") %>%
  as.matrix()

# Ensure the matrix is symmetric
adj_matrix <- pmax(adj_matrix, t(adj_matrix))  # Symmetrize

#create graph from adj matrix

wildfire_graph <- graph_from_adjacency_matrix(adj_matrix, mode = "undirected", weighted = TRUE)
```

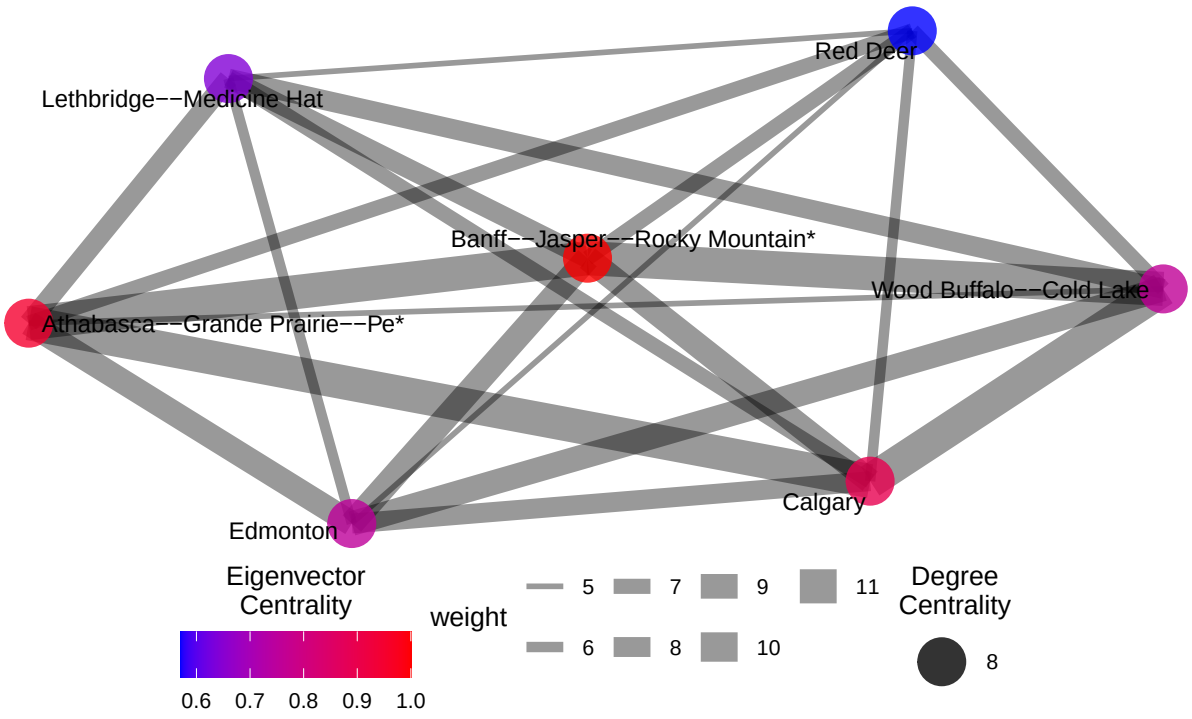
```
## Warning: The ‘adjmatrix‘ argument of ‘graph_from_adjacency_matrix()‘ must be symmetric
## with mode = "undirected" as of igraph 1.6.0.
## i Use mode = "max" to achieve the original behavior.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
degree_centrality_mun <- degree(wildfire_graph)
betweenness_centrality_mun <- betweenness(wildfire_graph)
closeness_centrality_mun <- closeness(wildfire_graph)
eigenvector_centrality_mun <- eigen_centrality(wildfire_graph)$vector

# Combine centrality measures into a data frame
centrality_measures_mun <- data.frame(
  degree = degree_centrality_mun,
  betweenness = betweenness_centrality_mun,
  closeness = closeness_centrality_mun,
  eigenvector = eigenvector_centrality_mun
)

centrality_measures_mun <- centrality_measures_mun %>% arrange(desc(degree))
print(centrality_measures_mun)
```

```
##                                degree betweenness  closeness eigenvector
## Banff--Jasper--Rocky Mountain*      8           0 0.01818182   1.0000000
## Calgary                             8           0 0.01923077   0.8929191
## Edmonton                            8           0 0.02173913   0.7533177
## Lethbridge--Medicine Hat            8           0 0.02380952   0.6444575
## Red Deer                            8           0 0.02702703   0.5716011
## Wood Buffalo--Cold Lake             8           0 0.01960784   0.7580741
## Athabasca--Grande Prairie--Pe*      8           0 0.01960784   0.9485992
```

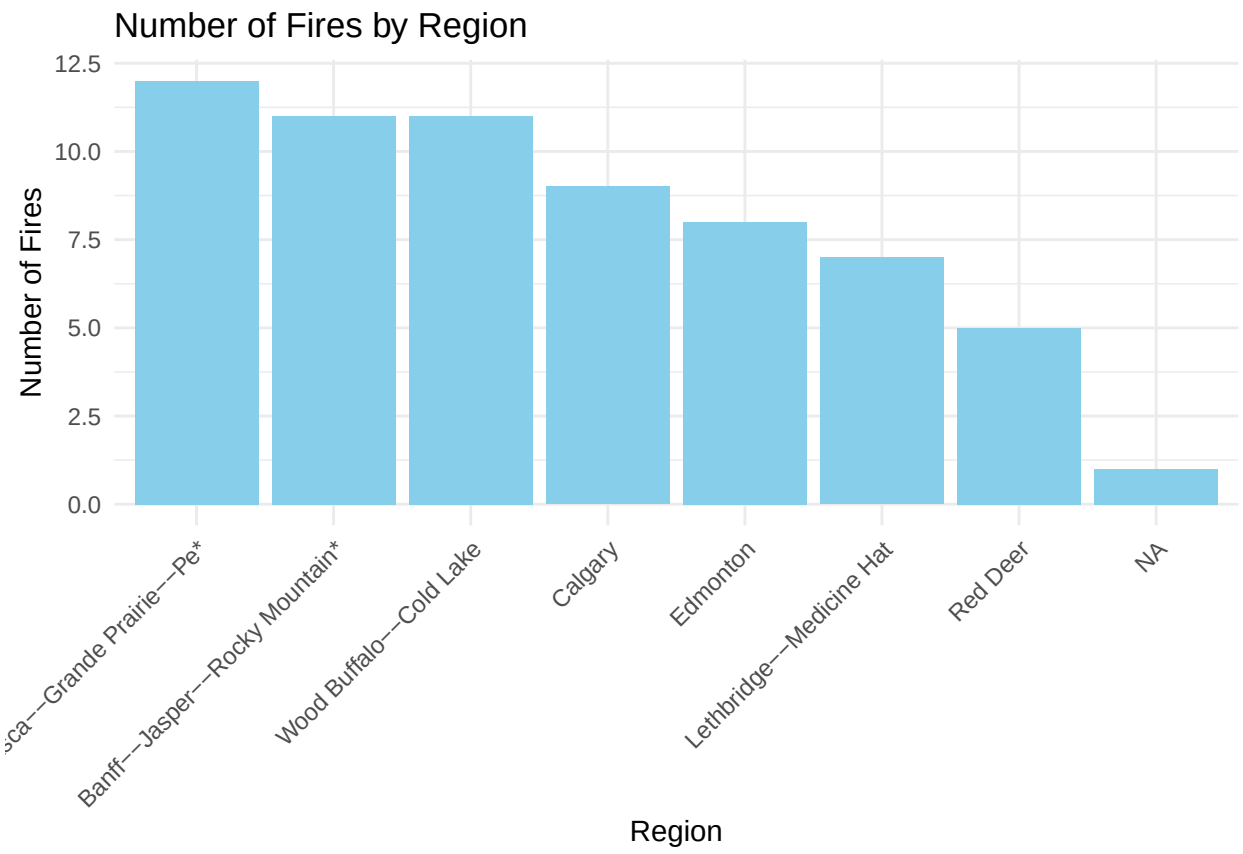
```
ggraph(wildfire_graph, layout = "fr") +
  geom_edge_link(aes(width = weight), alpha = 0.4, color = "black") +
  geom_node_point(aes(size = degree_centrality_mun, color = eigenvector_centrality_mun), alpha = 0.8) +
  geom_node_text(aes(label = name), repel = TRUE, size = 3, color = "black") +
  theme_void() +
  scale_color_gradient(low = "blue", high = "red", name = "Eigenvector\nCentrality") + # Gradient for e
  scale_size(range = c(3, 10), name = "Degree\nCentrality") + # Scaled node size
  guides(
    size = guide_legend(title.position = "top", title.hjust = 0.5),
    color = guide_colorbar(title.position = "top", title.hjust = 0.5)
  ) +
  labs(
    title = "Wildfire Network in Alberta",
    subtitle = "Node size: Degree Centrality, Color: Eigenvector Centrality",
    caption = "Edges represent shared wildfire causes between municipalities"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5, size = 12, margin = margin(b = 10)),
    legend.position = "bottom",
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 8),
    plot.caption = element_text(size = 8, margin = margin(t = 10))
  )
```

```
## Warning: The ‘trans‘ argument of ‘continuous_scale()‘ is deprecated as of ggplot2 3.5.0.
## i Please use the ‘transform‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

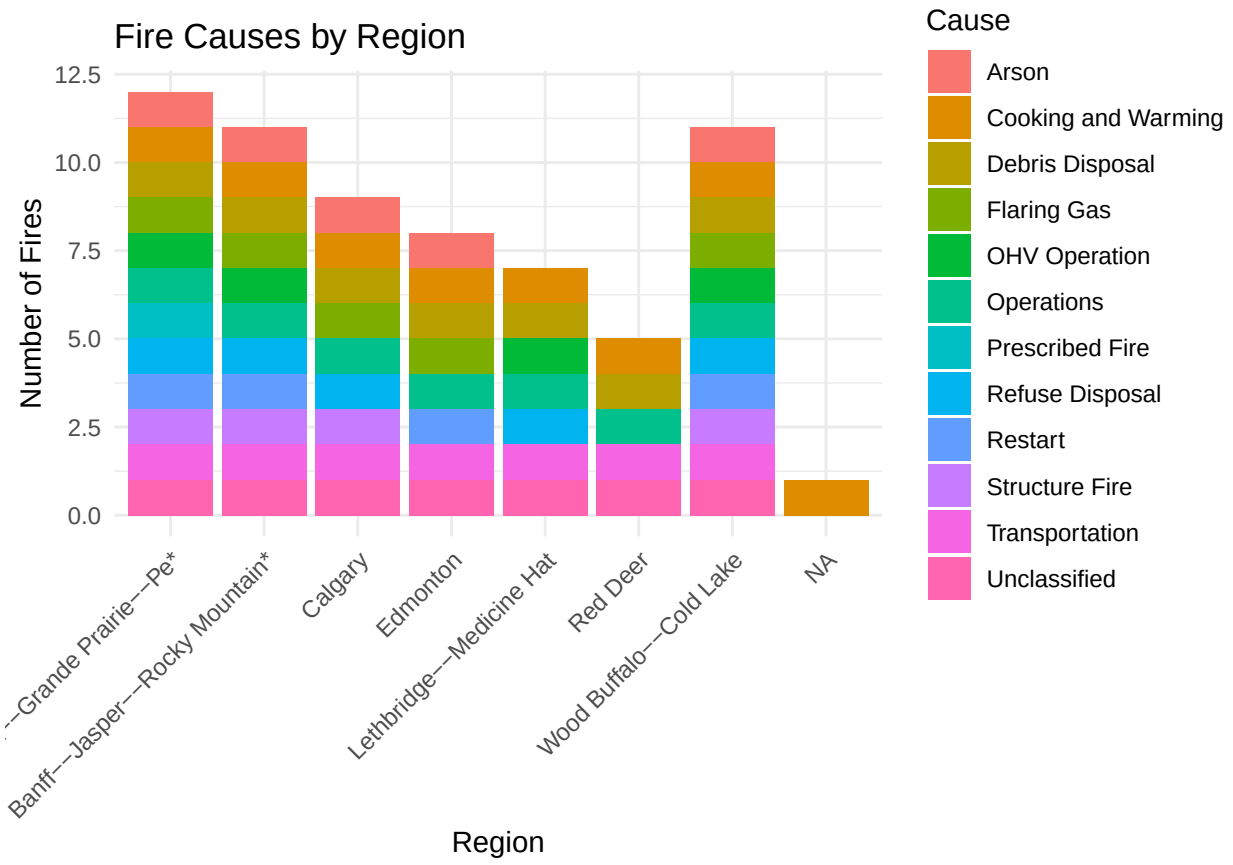


Edges represent shared wildfire causes between municipalities

```
# Bar plot: Number of fires by region
ggplot(fires_by_mun, aes(x = reorder(shapeName, -total_fires), y = total_fires)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  theme_minimal() +
  labs(
    title = "Number of Fires by Region",
    x = "Region",
    y = "Number of Fires"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



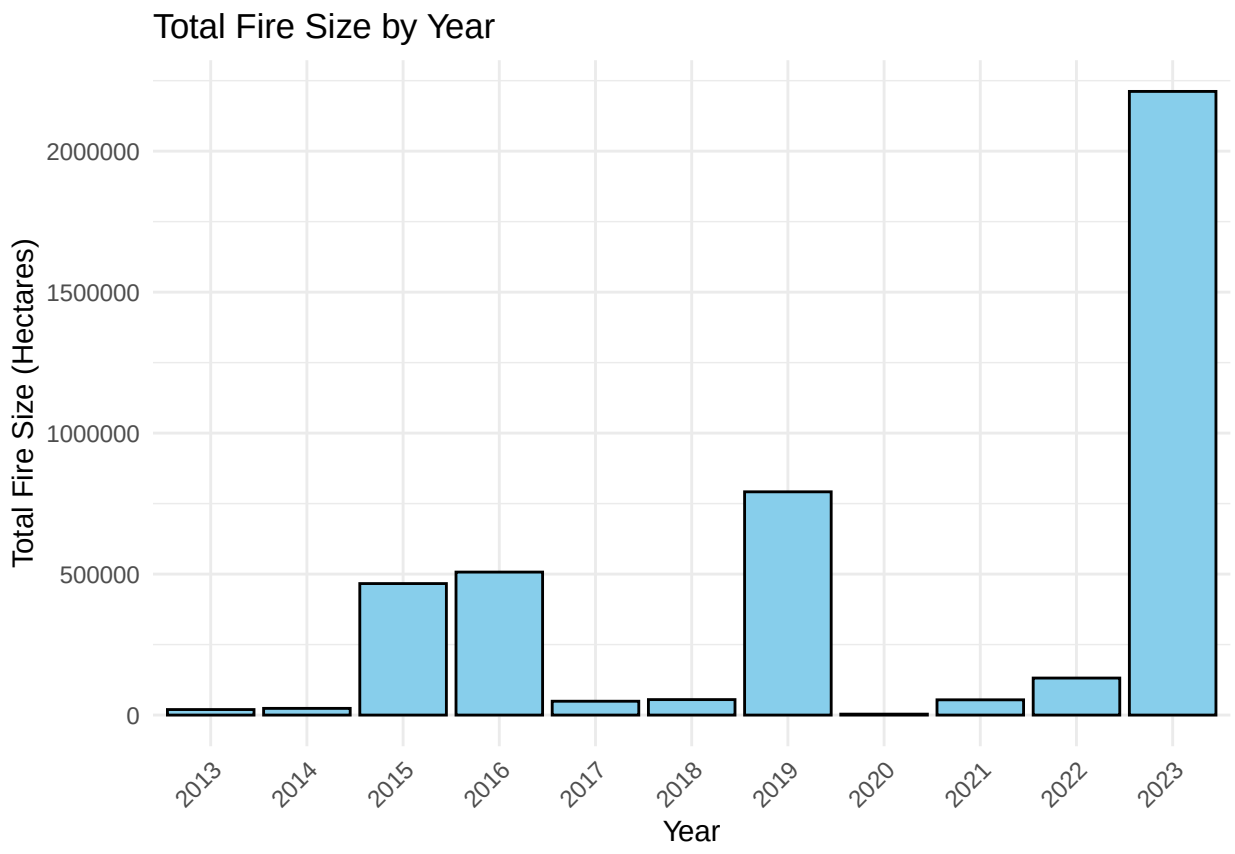

```
# Stacked bar plot: Fire causes by region
ggplot(fire_cause_by_mun, aes(x = reorder(shapeName, -count), y = count, fill = activity_class)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(
    title = "Fire Causes by Region",
    x = "Region",
    y = "Number of Fires",
    fill = "Cause"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



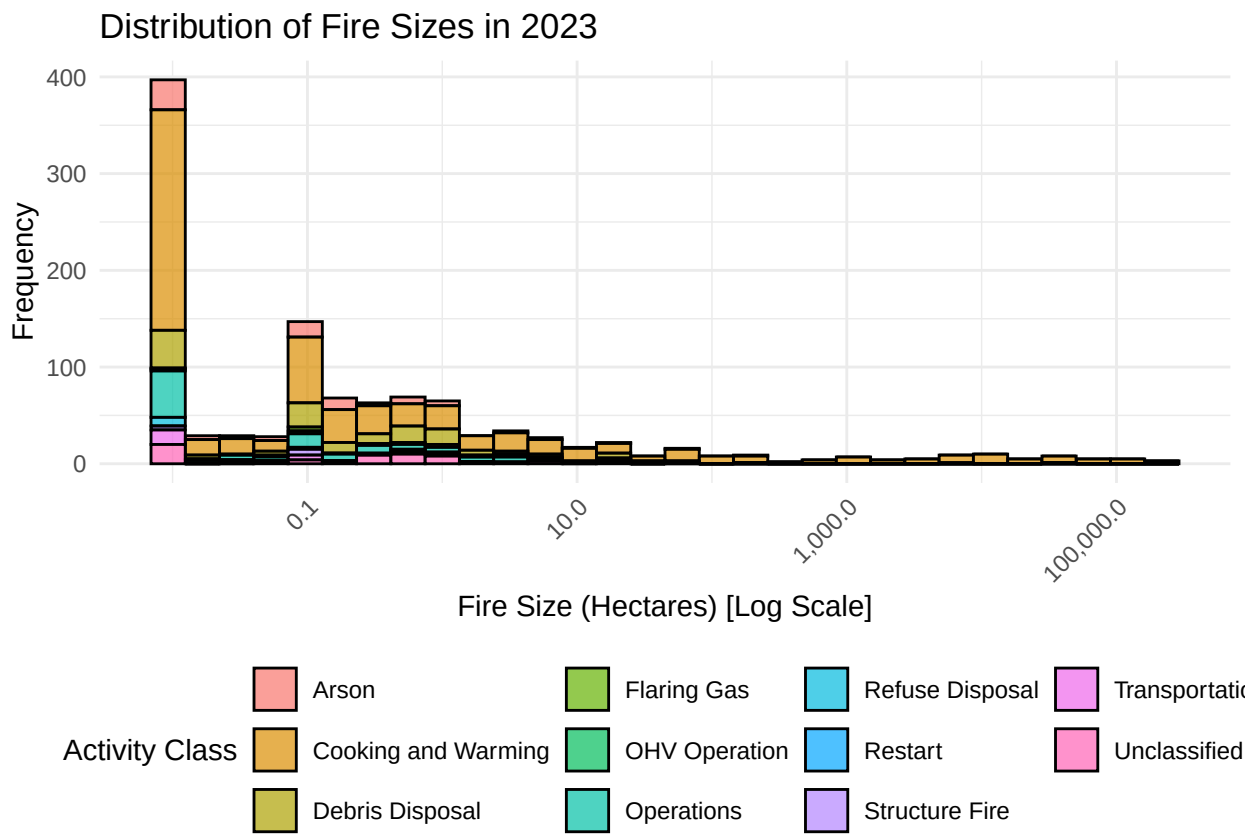

```
# Question 2: Which year had the largest fires, and what were their characteristics?

# Count the number of records for each fire year
fire_year_counts <- wildfire_data %>%
  group_by(fire_year) %>%
  summarise(record_count = n()) %>%
  arrange(desc(fire_year))

fire_year_counts
```

```
## # A tibble: 11 x 2
##    fire_year record_count
##        <int>        <int>
##  1      2023         1132
##  2      2022         1276
##  3      2021         1342
##  4      2020          723
##  5      2019         1005
##  6      2018         1279
##  7      2017         1244
##  8      2016         1376
##  9      2015         1898
## 10      2014         1470
## 11      2013         1226
```

```r
# Calculate the total fire size for each year
yearly_fire_sizes <- wildfire_data %>%
  group_by(fire_year) %>%
  summarise(total_fire_size = sum(current_size, na.rm = TRUE),
            max_fire_size = max(current_size, na.rm = TRUE)) %>%
  arrange(desc(total_fire_size))

# Identify the year with the largest fires (total size)
largest_fire_year <- yearly_fire_sizes$fire_year[1]

# Filter data for the year with the largest fires
largest_fire_year_data <- wildfire_data %>%
  filter(fire_year == largest_fire_year)

# Visualization: Total fire size per year
ggplot(yearly_fire_sizes, aes(x = factor(fire_year), y = total_fire_size)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(
    title = "Total Fire Size by Year",
    x = "Year",
    y = "Total Fire Size (Hectares)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

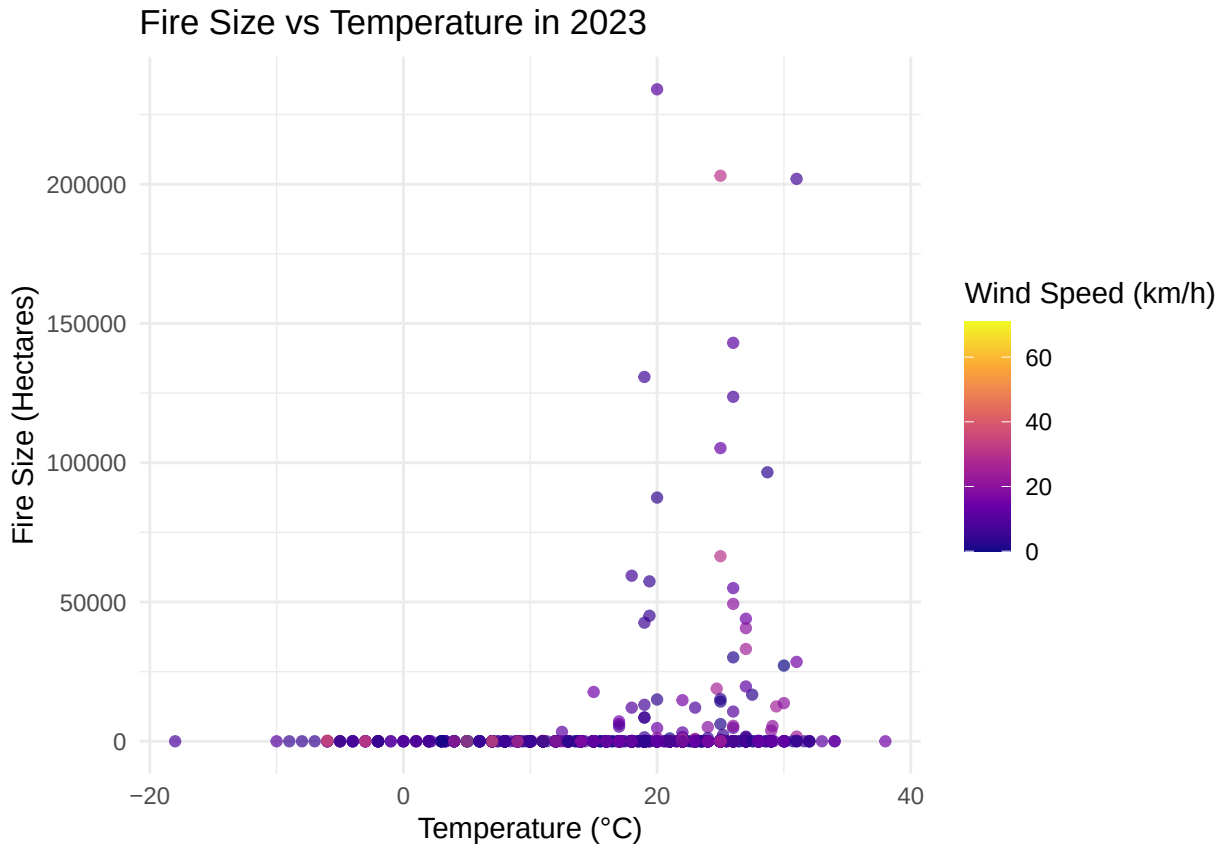


```r
# Histogram with log scale on the x-axis
ggplot(largest_fire_year_data, aes(x = current_size, fill = activity_class)) +
  geom_histogram(bins = 30, color = "black", alpha = 0.7) +
  scale_x_log10(labels = scales::comma) +  # Apply log scale and format x-axis labels
  labs(
    title = paste("Distribution of Fire Sizes in", largest_fire_year),
    x = "Fire Size (Hectares) [Log Scale]",
    y = "Frequency",
    fill = "Activity Class"
  ) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```



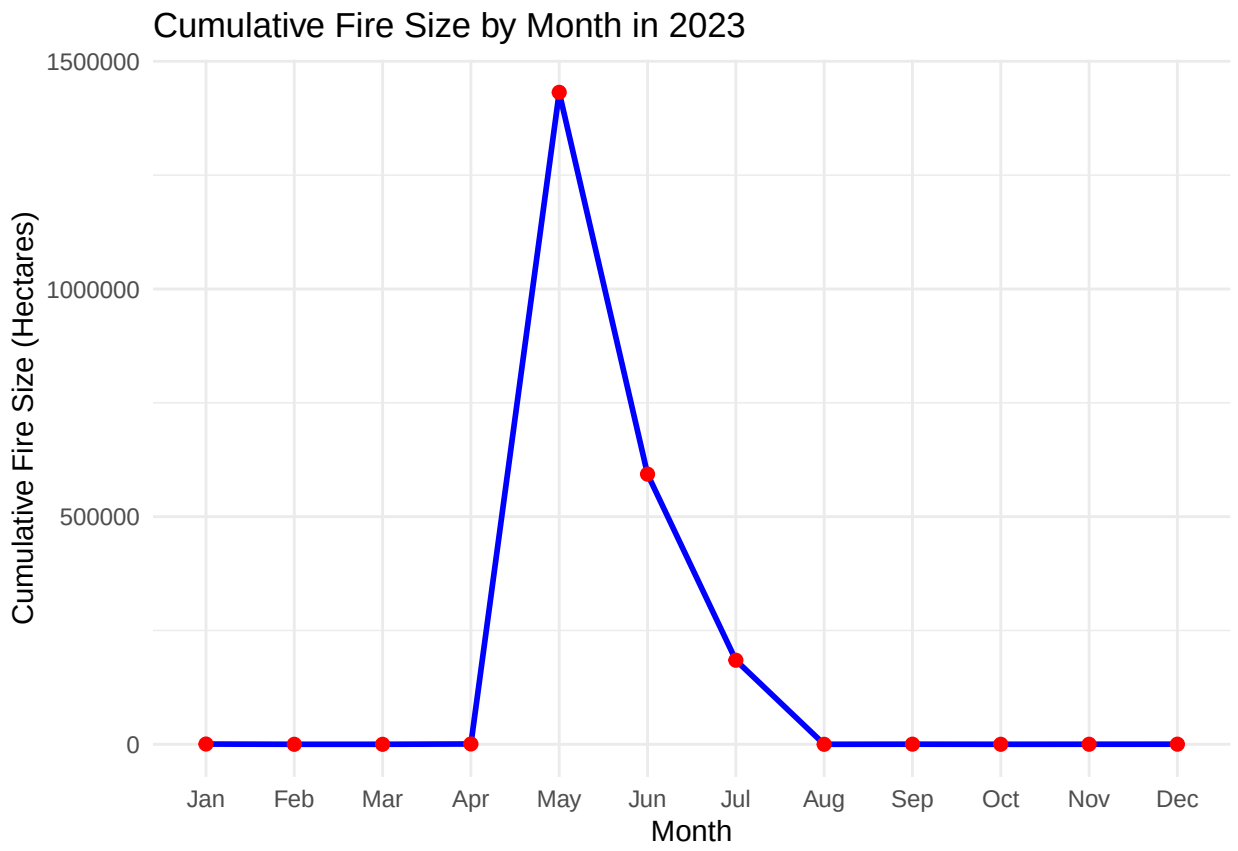

```r
# Scatter plot: Relationship between fire size and environmental conditions
ggplot(largest_fire_year_data, aes(x = temperature, y = current_size, color = wind_speed)) +
  geom_point(alpha = 0.7) +
  scale_color_viridis_c(option = "plasma") +
  labs(
    title = paste("Fire Size vs Temperature in", largest_fire_year),
    x = "Temperature (°C)",
    y = "Fire Size (Hectares)",
    color = "Wind Speed (km/h)"
```

```
  ) +
  theme_minimal()
```



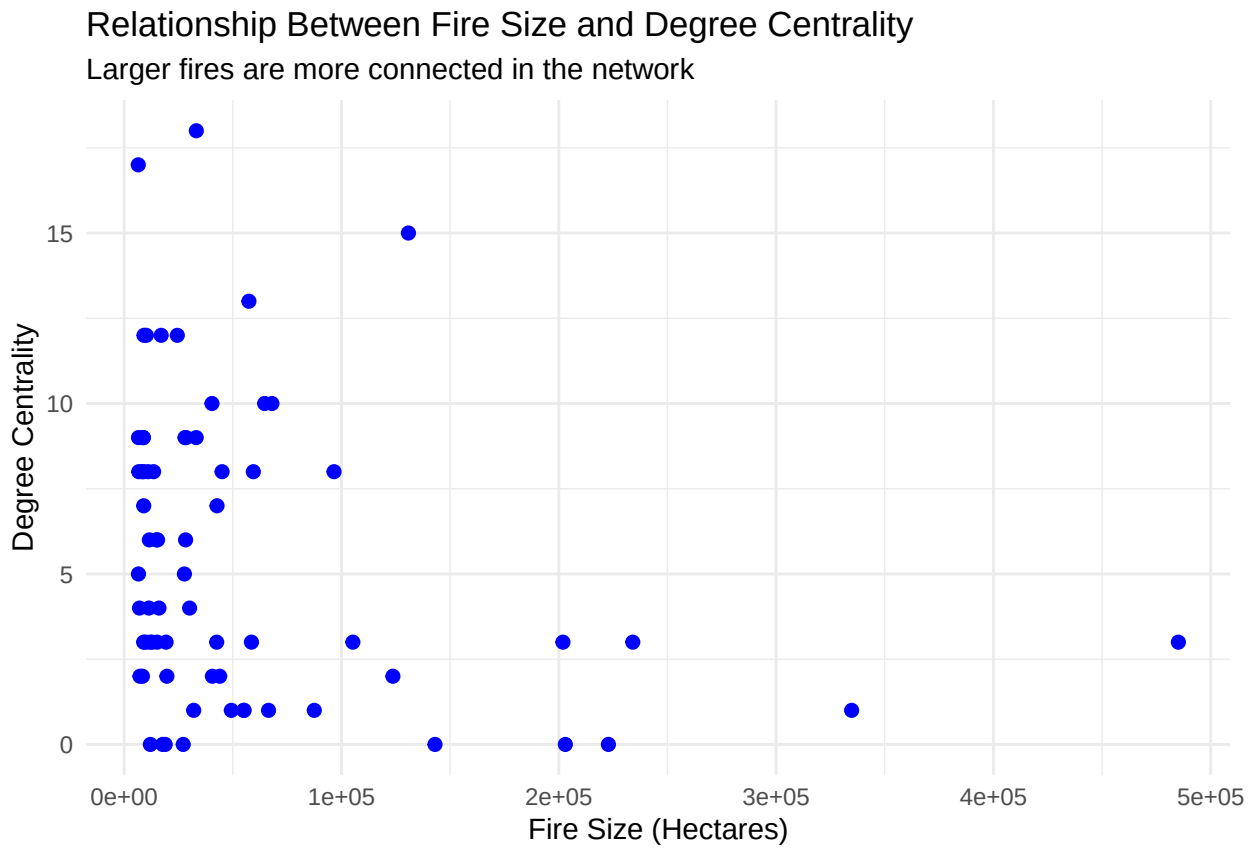

```
# Add month information to the dataset
largest_fire_year_data <- largest_fire_year_data %>%
  mutate(month = lubridate::month(fire_start_date, label = TRUE))

# Cumulative fire size by month
ggplot(largest_fire_year_data, aes(x = month, y = current_size, group = 1)) +
  geom_line(stat = "summary", fun = "sum", color = "blue", size = 1) +
  geom_point(stat = "summary", fun = "sum", color = "red", size = 2) +
  labs(
    title = paste("Cumulative Fire Size by Month in", largest_fire_year),
    x = "Month",
    y = "Cumulative Fire Size (Hectares)"
  ) +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



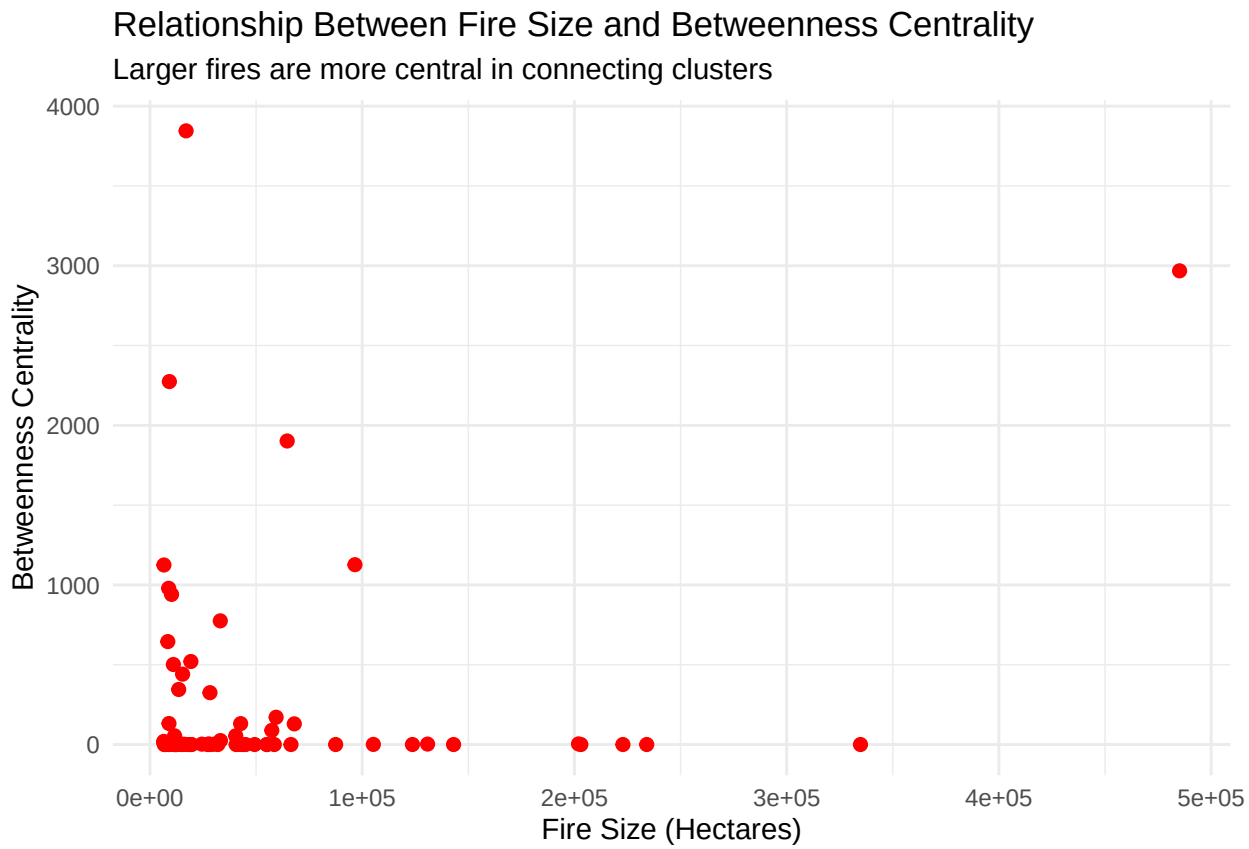

```r
# Question 3: Are larger wildfires more likely to have multiple causes, and do they play a more signifi

# Identify Large Fires
quantile_threshold <- 0.75
large_fires <- wildfire_data %>%
  filter(current_size > quantile(current_size, quantile_threshold, na.rm = TRUE))

# Add cause diversity
cause_diversity <- large_fires %>%
  group_by(fire_number) %>%
  summarise(cause_count = n_distinct(activity_class))

# Calculate Centrality Metrics

wildfire_network <- graph_from_data_frame(edges, vertices = nodes, directed = FALSE)

# Add centrality measures
V(wildfire_network)$degree <- degree(wildfire_network, mode = "all")
V(wildfire_network)$betweenness <- betweenness(wildfire_network, directed = FALSE)

# Merge centrality metrics with node attributes
node_metrics <- data.frame(
  name = V(wildfire_network)$name,
  degree = V(wildfire_network)$degree,
  betweenness = V(wildfire_network)$betweenness,
  size = V(wildfire_network)$current_size
```

```
)

# Filter for large fires
large_fire_metrics <- node_metrics %>%
  filter(size > quantile(size, quantile_threshold, na.rm = TRUE))

# Refined Visualizations
# (a) Scatterplot: Fire Size vs Degree Centrality
ggplot(large_fire_metrics, aes(x = size, y = degree)) +
  geom_point(color = "blue", size = 2) +
  labs(
    title = "Relationship Between Fire Size and Degree Centrality",
    subtitle = "Larger fires are more connected in the network",
    x = "Fire Size (Hectares)",
    y = "Degree Centrality"
  ) +
  theme_minimal()
```

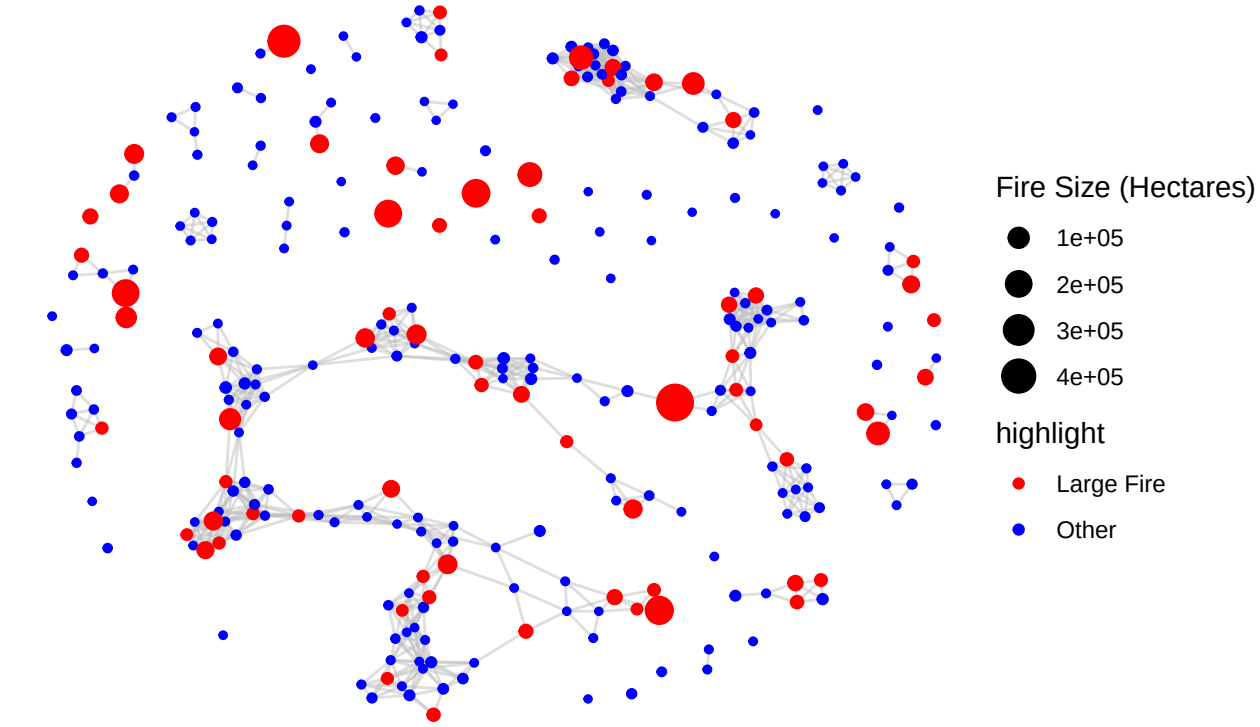



```
# (b) Scatterplot: Fire Size vs Betweenness Centrality
ggplot(large_fire_metrics, aes(x = size, y = betweenness)) +
  geom_point(color = "red", size = 2) +
  labs(
    title = "Relationship Between Fire Size and Betweenness Centrality",
    subtitle = "Larger fires are more central in connecting clusters",
    x = "Fire Size (Hectares)",
```

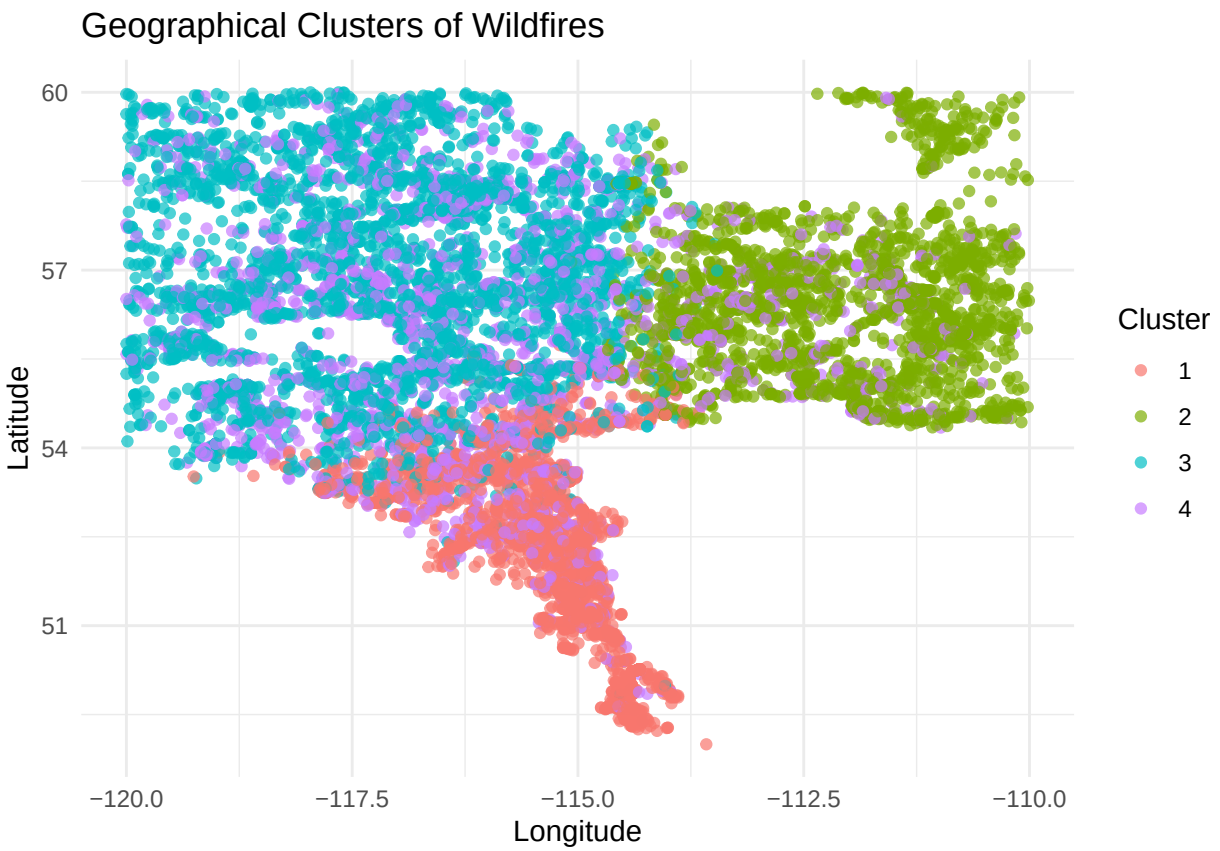
```
    y = "Betweenness Centrality"
  ) +
  theme_minimal()
```



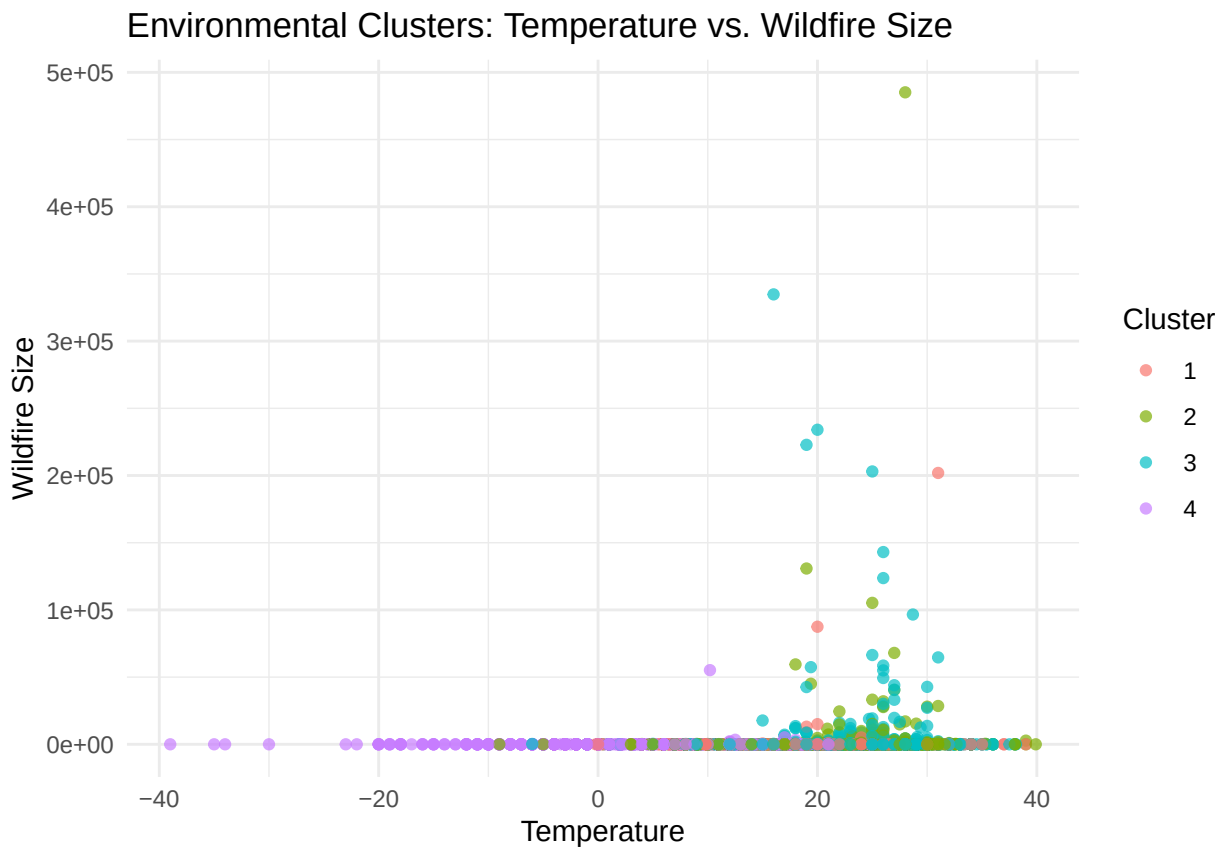

```
# (c) Network Visualization: Highlighting Large Fires
V(wildfire_network)$highlight <- ifelse(
  V(wildfire_network)$current_size > quantile(V(wildfire_network)$current_size, quantile_threshold, na.
  "Large Fire", "Other"
)

wildfire_tbl_graph <- as_tbl_graph(wildfire_network)

ggraph(wildfire_tbl_graph, layout = "fr") +
  geom_edge_link(color = "grey", alpha = 0.5) +
  geom_node_point(aes(color = highlight, size = current_size)) +
  scale_color_manual(values = c("Large Fire" = "red", "Other" = "blue")) +
  labs(
    title = "Wildfire Network with Highlighted Large Fires",
    subtitle = "Red nodes represent fires in the top 25% by size",
    size = "Fire Size (Hectares)"
  ) +
  theme_void()
```




```r
# Question 5: How do environmental and geographical factors influence wildfire clustering and dominant

# latitude, longitude, temperature, humidity, speed are chosen factors for DBScan clustering
data_for_clustering <- wildfire_data %>%
  select(fire_location_latitude, fire_location_longitude, temperature, relative_humidity, wind_speed) %
  na.omit()  # to omit rows with missing values

dbscan_result <- dbscan::dbscan(data_for_clustering, eps = 0.5, minPts = 10)

# Add cluster labels to the data
wildfire_data$cluster <- dbscan_result$cluster

# Scale data for K-Means
scaled_data <- scale(data_for_clustering)

# Run K-Means with a predefined number of clusters
kmeans_result <- kmeans(scaled_data, centers = 4, nstart = 25)

# Add cluster labels to the data
wildfire_data$cluster <- kmeans_result$cluster

# Scatterplot of clusters (latitude vs. longitude)
ggplot(wildfire_data, aes(x = fire_location_longitude, y = fire_location_latitude, color = as.factor(cl
  geom_point(alpha = 0.7) +
  labs(title = "Geographical Clusters of Wildfires", x = "Longitude", y = "Latitude", color = "Cluster"
  theme_minimal()
```




```
# Scatterplot of clusters (temperature vs. wildfire size)
ggplot(wildfire_data, aes(x = temperature, y = current_size, color = as.factor(cluster))) +
  geom_point(alpha = 0.7) +
  labs(title = "Environmental Clusters: Temperature vs. Wildfire Size", x = "Temperature", y = "Wildfir
  theme_minimal()
```




```
# Summarize environmental factors for each cluster
cluster_summary <- wildfire_data %>%
  group_by(cluster) %>%
  summarise(
    avg_temperature = mean(temperature, na.rm = TRUE),
    avg_humidity = mean(relative_humidity, na.rm = TRUE),
    avg_wind_speed = mean(wind_speed, na.rm = TRUE),
    avg_size = mean(current_size, na.rm = TRUE),
    n_wildfires = n()
  )

print(cluster_summary)
```

```
## # A tibble: 4 x 6
##   cluster avg_temperature avg_humidity avg_wind_speed avg_size n_wildfires
##     <int>           <dbl>        <dbl>          <dbl>    <dbl>       <int>
## 1       1            18.8         37.7           6.87     84.5        4081
## 2       2            19.6         40.3           9.32    455.         3049
## 3       3            21.5         37.6          11.2     618.         4059
## 4       4            11.5         69.0           5.68     26.5        2782
```

```
#
```