WESTERN SYDNEY UNIVERSITY

SCHOOL of: Computer, Data and Mathematical Sciences

ASSIGNMENT COVER SHEET

STUDENT DETAILS

Name: Midhun Shyam **Student ID:** 22058122

SUBJECT AND TUTORIAL DETAILS

Subject Name: Advanced Statistical Methods **Subject code:** MATH 7002

Tutorial Group: Click or tap here to enter text. **Day:** Thursday **Time:** 9:00 - 13:00

Lecturer or Tutor name: Prof. Stuart Fitzpatrick

ASSIGNMENT DETAILS

Title: Applied Project: NSW Fire data analysis **Length:** 40 pages

Due Date: 28/10/2024 **Date submitted: 28/10/2024**

Home campus: Parramatta South

DECLARATION

By submitting your work using this link you are certifying that:

☒ You hold a copy of this submission if the original is lost or damaged.

☒ No part of this submission has been copied from any other student's work or from any other third party (including generative AI) except where due acknowledgment is made in the submission.

☒ No part of this submission has been submitted by you in another (previous or current) assessment, except where appropriately referenced, and with prior permission from the teacher/tutor/supervisor/ Subject Coordinator for this subject.

☒ No part of this submission has been written/produced for you by any other person or technology except where collaboration has been authorised by the teacher/tutor/ supervisor/Subject Coordinator either in the assessment resources section of the Learning Guide for this assessment task, in the instructions for this assessment task, or through vUWS.

☒ You are aware that this submission will be reproduced and submitted to detection software programs for the purpose of investigating possible breaches of the Student Misconduct Rule, for example, plagiarism, contract cheating, or unauthorised use of generative AI. Turnitin or other tools of investigation may retain a copy of the submission for the purposes of future investigation.

☒ You will not make this submission available to any other person unless required by the University.

***Instructions**: Please complete the requested details in the form, save it and convert to PDF before adding your signature below*

Student signature:

Note: An examiner or lecturer/tutor has the right to not mark this assignment if the above declaration has not been completed. Staff may contact you for permission to share a de-identified extract or copy of your submission with students or staff for teaching purposes, following guidelines for requesting and sharing exemplar assessment tasks.

Table of Contents

MATH7002

Midhun Shyam

2024-10-28

```
library(MASS)
require(hexbin)

## Loading required package: hexbin

## Warning: package 'hexbin' was built under R version 4.3.3

require(mixtools)

## Loading required package: mixtools

## mixtools package, version 2.0.0, Released 2022-12-04
## This package is based upon work supported by the National Science Foundati
on under Grant No. SES-0518772 and the Chan Zuckerberg Initiative: Essential
Open Source Software for Science (Grant No. 2020-255193).

require(mvtnorm)

## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 4.3.3

##
## Attaching package: 'mvtnorm'

## The following objects are masked from 'package:mixtools':
##
##     dmvnorm, rmvnorm

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.1

library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

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

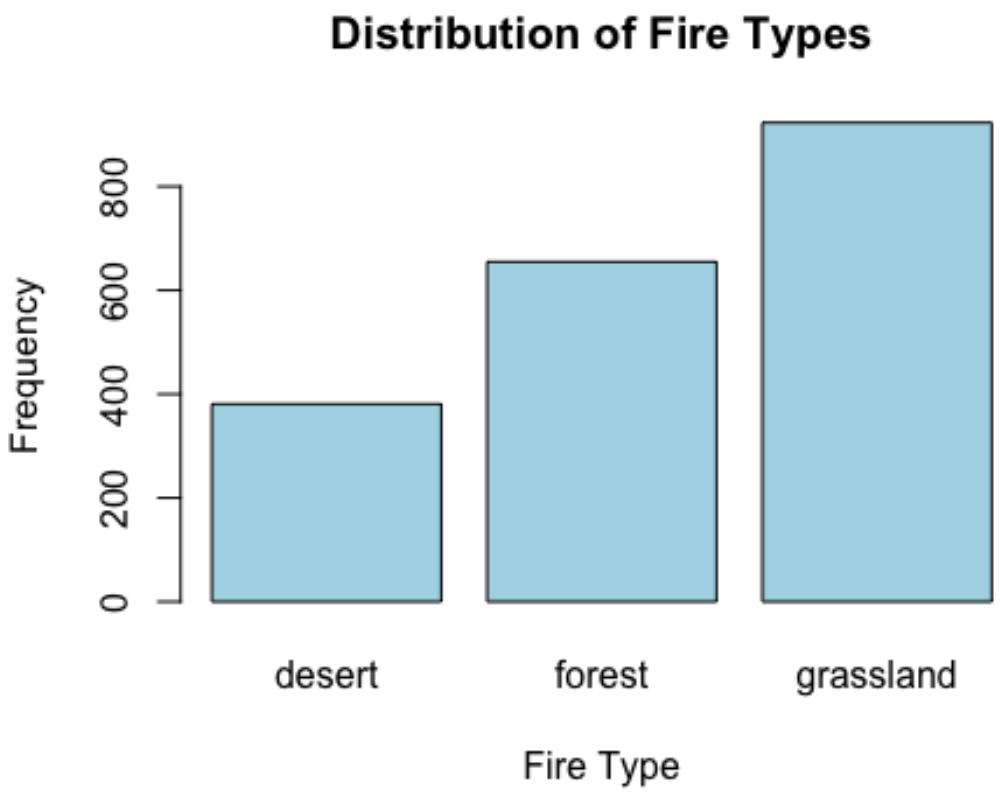
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(leaflet)

## Warning: package 'leaflet' was built under R version 4.3.1

library(leaflet.extras)

## Warning: package 'leaflet.extras' was built under R version 4.3.3

library(htmltools)

## Warning: package 'htmltools' was built under R version 4.3.1
```

1. Preliminary Analysis

```
data <- read.csv("fire2024.csv")
print(str(data))

## 'data.frame':    1959 obs. of  6 variables:
##  $ latitude : num  -34.6 -35.1 -34.5 -34.1 -34 ...
##  $ longitude: num  151 150 151 150 150 ...
##  $ type     : chr  "forest" "forest" "grassland" "grassland" ...
##  $ duration : int  36 26 10 16 49 17 11 12 16 35 ...
##  $ temp     : int  28 23 25 31 30 24 20 22 20 29 ...
##  $ humidity : int  85 84 52 77 56 35 30 49 35 56 ...
## NULL

summary(data)

##     latitude        longitude         type              duration
##  Min.   :-36.19   Min.   :144.4   Length:1959        Min.   : 3.00
##  1st Qu.:-34.22   1st Qu.:149.4   Class :character   1st Qu.:16.00
##  Median :-33.72   Median :150.3   Mode  :character   Median :23.00
##  Mean   :-33.71   Mean   :149.8                      Mean   :25.49
##  3rd Qu.:-33.19   3rd Qu.:150.8                      3rd Qu.:35.00
##  Max.   :-31.17   Max.   :151.9                      Max.   :60.00
##       temp          humidity
##  Min.   :13.00   Min.   :  1.00
##  1st Qu.:22.00   1st Qu.: 34.00
##  Median :27.00   Median : 53.00
##  Mean   :28.76   Mean   : 52.43
##  3rd Qu.:33.00   3rd Qu.: 74.00
##  Max.   :58.00   Max.   :100.00

print(head(data,2))

##    latitude longitude   type duration temp humidity
## 1 -34.59785  150.7966 forest       36   28       85
## 2 -35.10680  150.2604 forest       26   23       84
```

The summary of the dataset shows that the latitude ranges from -36.19 to -31.17, while the longitude varies from 144.4 to 151.9, indicating the geographical distribution of the data. The duration variable ranges from a minimum of 3 to a maximum of 60, with a mean of 25.49, suggesting varying time lengths. Temperature values range from 13 to 58, with a mean of 28.76, while humidity ranges from 1 to 100, with a mean of 52.43. This implies that the dataset contains diverse temperature and humidity observations over different durations and locations. The data type of type is character-based which needs to be converted to factor.

```
data$type <- as.factor(data$type)

# Create a table of fire types
fireDistribution <- table(data$type)

# Print the distribution with type names and frequencies
cat("The distribution of the fire across the type of fire is:\n")

## The distribution of the fire across the type of fire is:

for (i in 1:length(fireDistribution)) {
  cat(names(fireDistribution)[i], ":", fireDistribution[i], "\n")
}

## desert : 381
## forest : 655
## grassland : 923

# Create a bar plot for categorical variable type
barplot(fireDistribution, main="Distribution of Fire Types", xlab="Fire Type"
, ylab="Frequency", col="lightblue")
```



Chi-squared Test for Fire Type Distribution

The Chi-squared test is used to assess whether the **observed distribution of fire types** (desert, forest, grassland) differs significantly from an **expected distribution**. This test evaluates whether the differences between observed counts and expected counts are due to random chance or if they indicate a systematic variation in the distribution of fire types in the dataset.

Hypothesis

- **Null Hypothesis (H_0)**: The observed distribution of fire types matches the expected distribution; the difference in the distribution are due to random chance.
- **Alternative Hypothesis (H_1)**: The observed distribution of fire types significantly differs from the expected distribution, indicating a systematic variation.

```
# Observed frequency table
fireDistribution <- table(data$type)

# Expected probabilities (equal probabilities for each fire type)
expectedProb <- rep(1 / length(fireDistribution), length(fireDistribution))

# Calculate the expected frequencies
expectedFreq <- expectedProb * sum(fireDistribution)
```

```
cat("\nExpected frequencies (based on expected probabilities):\n")
```

```
##
## Expected frequencies (based on expected probabilities):
```

```
print(round(expectedFreq, 2))
```

```
## [1] 653 653 653
```

```
# Perform the Chi-squared test
chisq <- chisq.test(fireDistribution, p = expectedProb)

# Print the observed, expected frequencies, and the test result
cat("Observed frequencies:\n")
```

```
## Observed frequencies:
```

```
print(fireDistribution)
```

```
##
##    desert    forest grassland
##       381       655       923
```

```
cat("\nChi-squared test results:\n")
```

```
##
## Chi-squared test results:
```

```
print(chisq)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  fireDistribution
## X-squared = 224.94, df = 2, p-value < 2.2e-16
```

Chi-squared Test Results for Fire Type Distribution

The Chi-squared test was conducted to determine whether the observed distribution of fire types (desert, forest, grassland) deviates significantly from the expected distribution.

Results

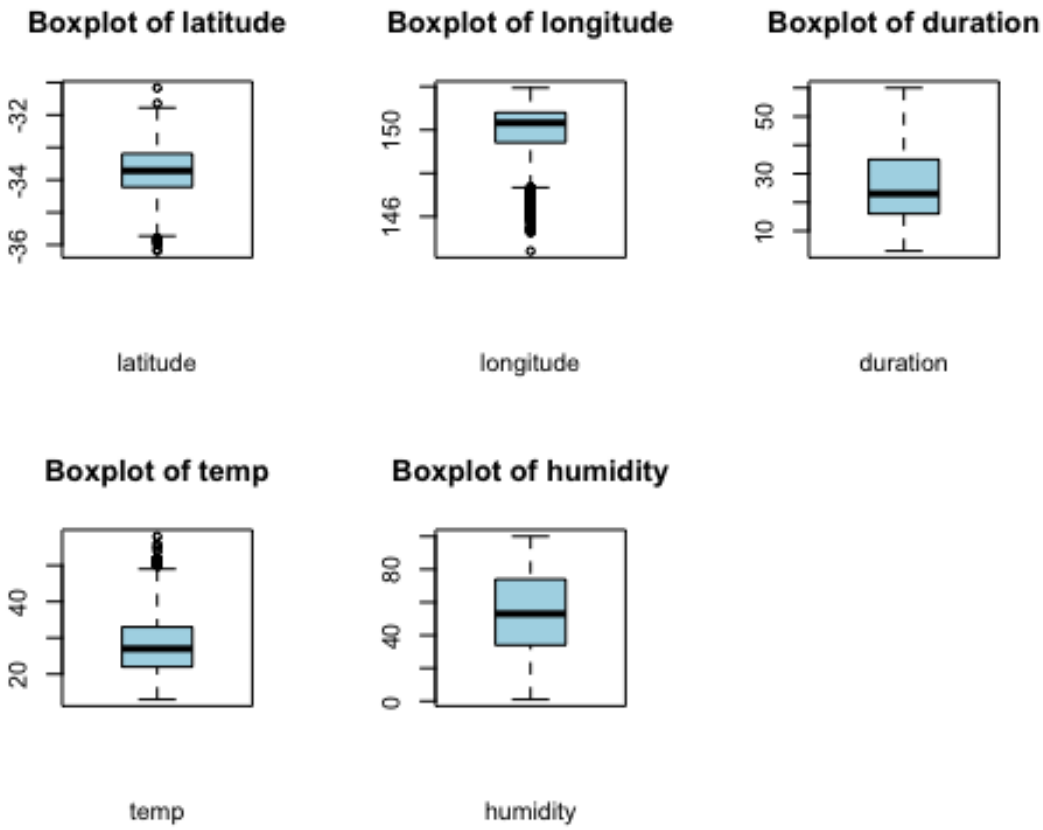
- **Chi-squared Statistic (X^2)**: 224.94
- **Degrees of Freedom (df)**: 2
- **p-value**: < 2.2e-16

Interpretation

The p-value is extremely low ($p < 0.05$), indicating that the observed distribution of fire types significantly differs from the expected distribution. Thus, we **reject the null hypothesis**,

suggesting that the distribution of fire types is not due to random chance and may be influenced by other factors.

```
par(mfrow = c(2,3))
# Loop through each column in the data frame
for (var in names(data)) {
  # Check if the column is numeric
  if (is.numeric(data[[var]])) {
    # Create a boxplot for the numeric column
    boxplot(data[[var]], main = paste("Boxplot of", var), xlab = var, col = "
lightblue")
  }
}
```



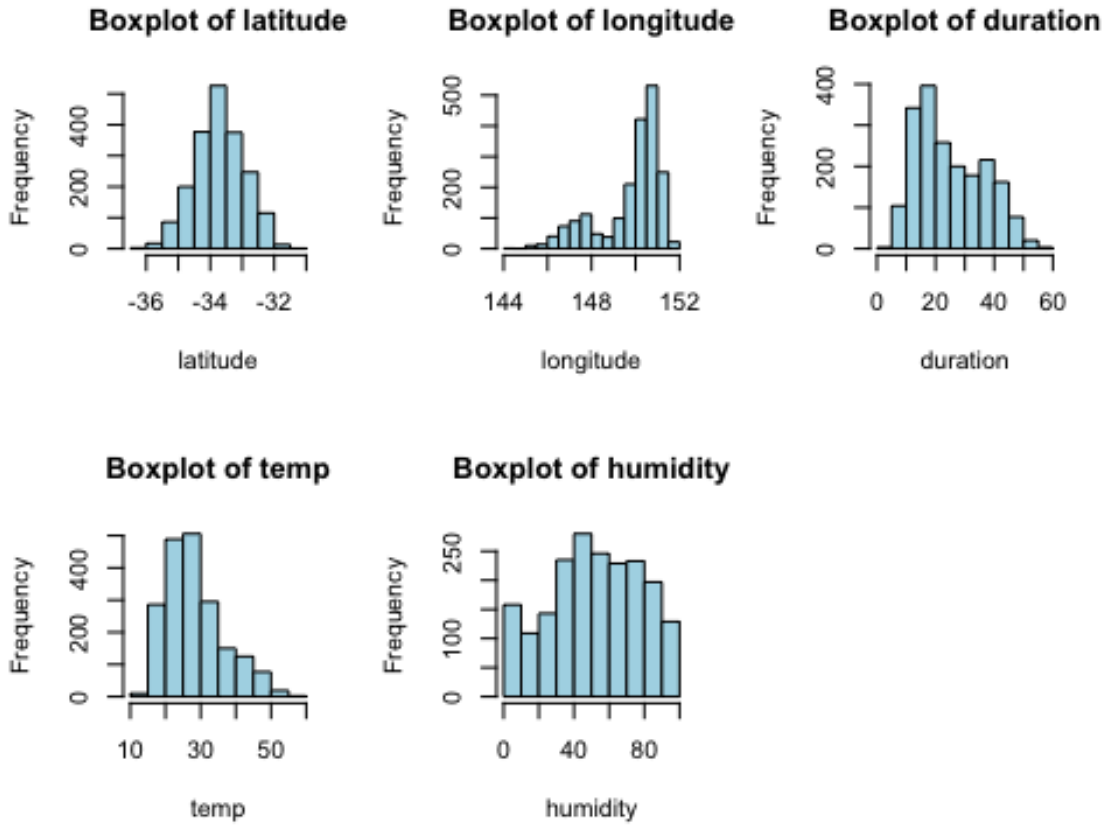
The five boxplots in the image display the distributions of the variables: latitude, longitude, duration, temperature (temp), and humidity.

1. **Latitude:**
   - The boxplot shows that the latitude data is centered around -34, with a relatively symmetric distribution.
   - There are a few outliers below -36 and above -32, as indicated by the dots.
   - The range between the first quartile (Q1) and third quartile (Q3) is relatively tight, suggesting limited variation in the middle 50% of values.

2. **Longitude:**
   - The longitude data has a slight right skew, with more spread below the median.
   - There are several outliers below 148, indicating some extreme low values that deviate significantly from the central distribution.
   - The interquartile range (IQR) is relatively narrow, indicating a concentration of values around the median.
3. **Duration:**
   - The duration variable appears skewed to the right, with the median closer to the lower quartile.
   - There are no extreme outliers, but the spread is wider above the median, suggesting higher variability in longer durations.
   - The boxplot shows a moderate variation, with most durations falling between 10 and 30.
4. **Temperature (Temp):**
   - The temperature boxplot reveals a clear right skew, with several outliers above 40, indicating the presence of extreme high values.
   - The median is closer to the lower quartile, suggesting a concentration of observations at lower temperature values.
   - The IQR suggests moderate variation in the temperature range, but with higher temperatures being less common.
5. **Humidity:**
   - The humidity data shows a fairly symmetric distribution, with the median positioned near the center of the box.
   - There are no apparent outliers, and the IQR indicates a balanced spread of values.
   - Most of the humidity values range between 20 and 80, indicating a wide spread of mid-range humidity levels.

The boxplots provide insights into the central tendency, variability, and presence of outliers in each variable. The distributions for longitude, duration, and temperature show skewness and outliers, while humidity is more evenly distributed.

```
par(mfrow = c(2,3))
# Loop through each column in the data frame
for (var in names(data)) {
  # Check if the column is numeric
  if (is.numeric(data[[var]])) {
    # Create a boxplot for the numeric column
    hist(data[[var]], main = paste("Boxplot of", var), xlab = var, col = "lig
htblue")
  }
}
```



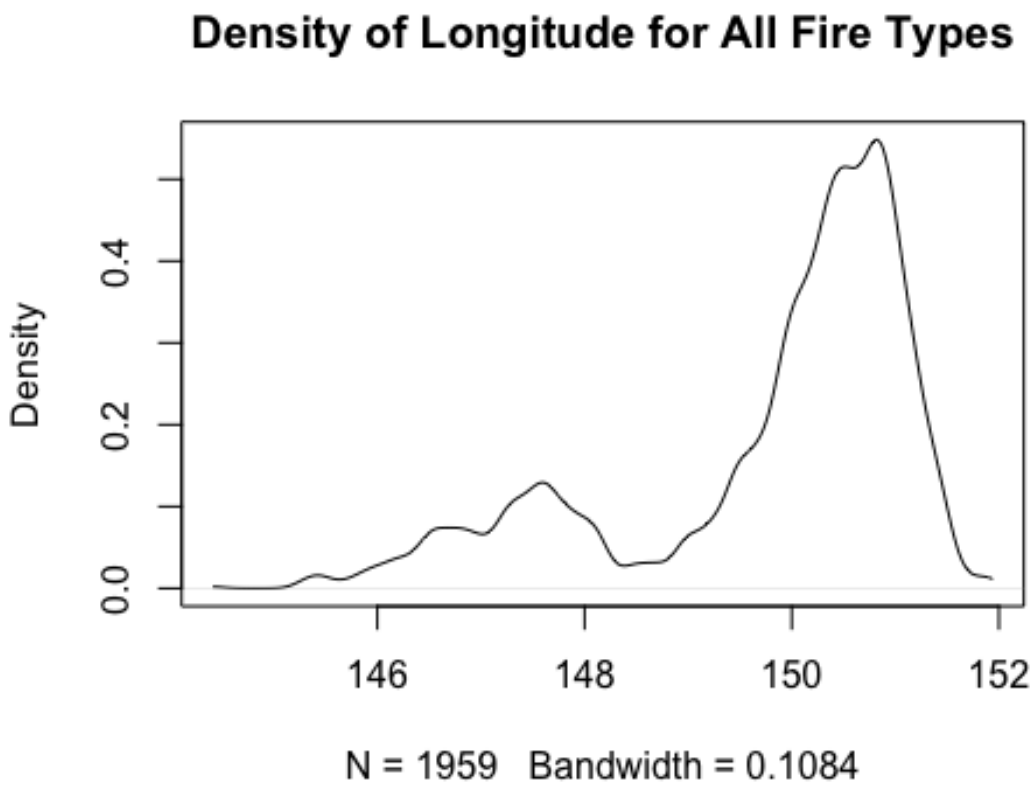
The six histograms in the image display the distributions of six variables: latitude, longitude, duration, temperature (temp), and humidity.

1. **Latitude:**
   - The distribution of latitude is approximately normal, centered around -34. This suggests that the data predominantly clusters around this latitude, with fewer observations as you move further away from the mean.
2. **Longitude:**
   - The longitude distribution is skewed to the left, indicating more data points at higher longitudes, with a concentration around 150-151. This suggests that the majority of the observations are located within this longitude range, with fewer observations at lower values.
3. **Duration:**
   - The duration variable shows a right-skewed distribution. There are more data points with shorter durations (around 10-20), while fewer data points have longer durations (above 50). This suggests that most of the observed durations are relatively short.
4. **Temperature (Temp):**
   - The temperature distribution is also right-skewed, with a majority of the values clustering around 20-30. The tail extends toward higher temperatures, indicating fewer observations with extreme temperature values.

5. **Humidity:**
   - The humidity distribution is fairly uniform, meaning there is no strong concentration of values in any specific range. The data is relatively evenly spread between 0 and 100, suggesting a wide variability in humidity levels.

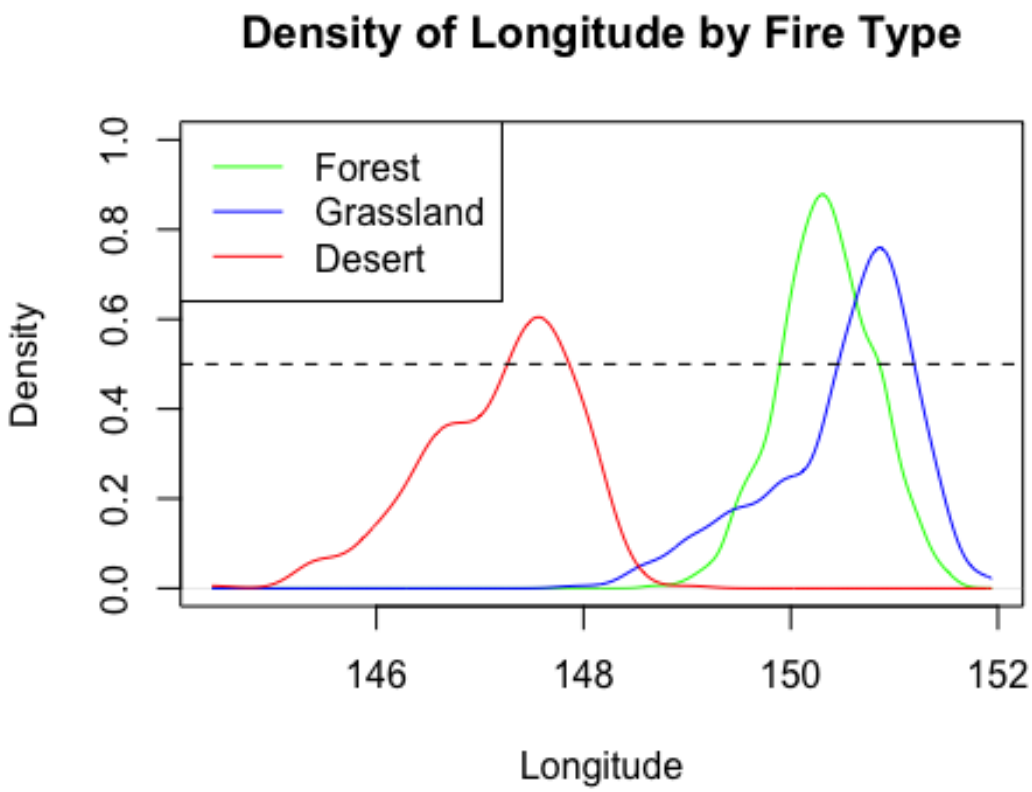
The observed distributions indicate a mix of normal, skewed, and uniform patterns across the different variables.

```
# Extract longitude and determine its range
x <- data$longitude
lowx <- min(x)
highx <- max(x)


# Compute density for all longitude data
f <- density(x, kernel = "gaussian", bw ="ucv", from = lowx, to = highx)
plot(f, main = "Density of Longitude for All Fire Types")
```

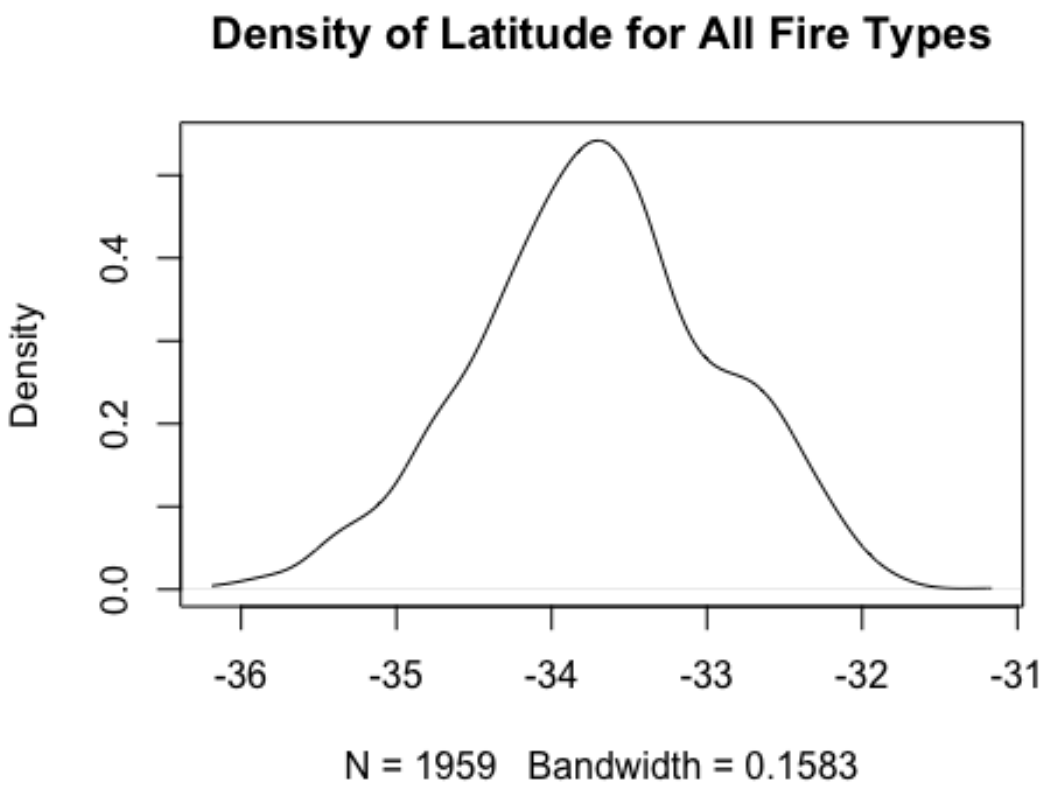



```
# Compute and plot densities for specific fire types
f.f = density(data$longitude[data$type=="forest"], from=lowx, to=highx)
f.g = density(data$longitude[data$type=="grassland"], from=lowx, to=highx)
f.d = density(data$longitude[data$type=="desert"], from=lowx, to=highx)
```

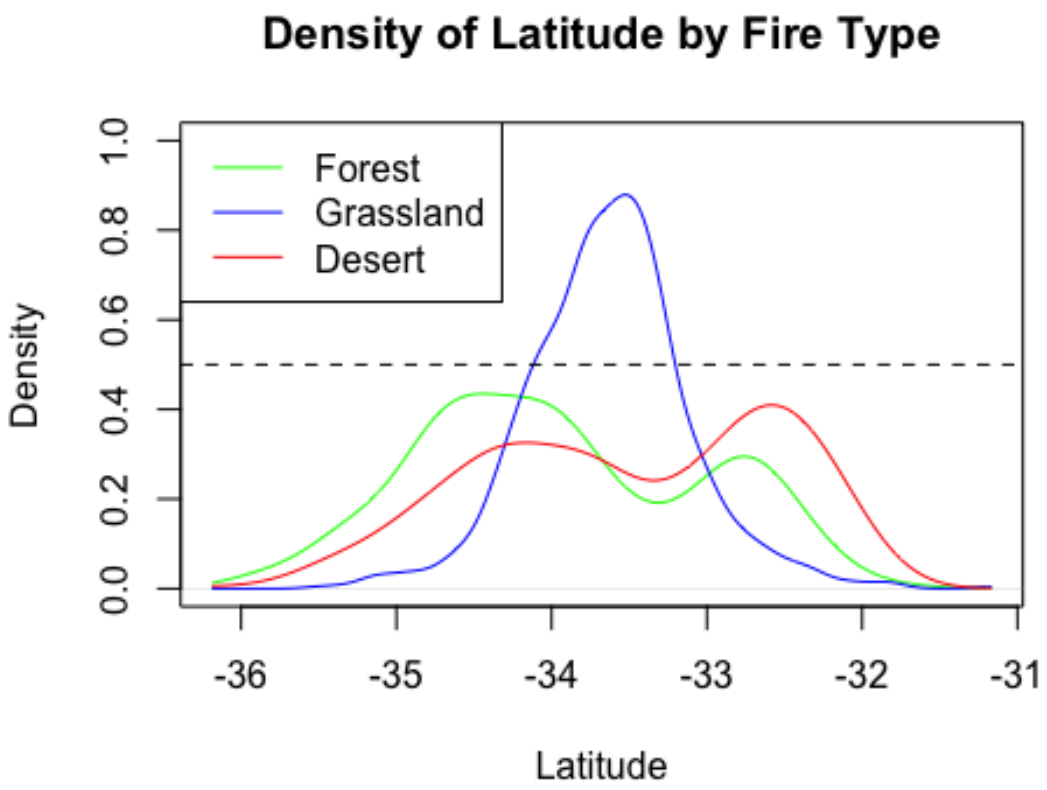
```r
# Plot the densities for specific types on the same graph
plot(f.f, col = "green", main = "Density of Longitude by Fire Type", xlab = "
Longitude", ylab = "Density",ylim = c(0,1))
lines(f.g, col = "blue")
lines(f.d, col = "red")
legend("topleft", legend = c("Forest", "Grassland", "Desert"), col = c("green
", "blue", "red"), lty = 1)
abline(h=0.5, lty=2)
```




```r
# Extract latitude and determine its range
y <- data$latitude
lowy <- min(y)
highy <- max(y)

# Compute density for all latitude data
f <- density(y, kernel = "gaussian", bw = "ucv", from = lowy, to = highy)
plot(f, main = "Density of Latitude for All Fire Types")
```

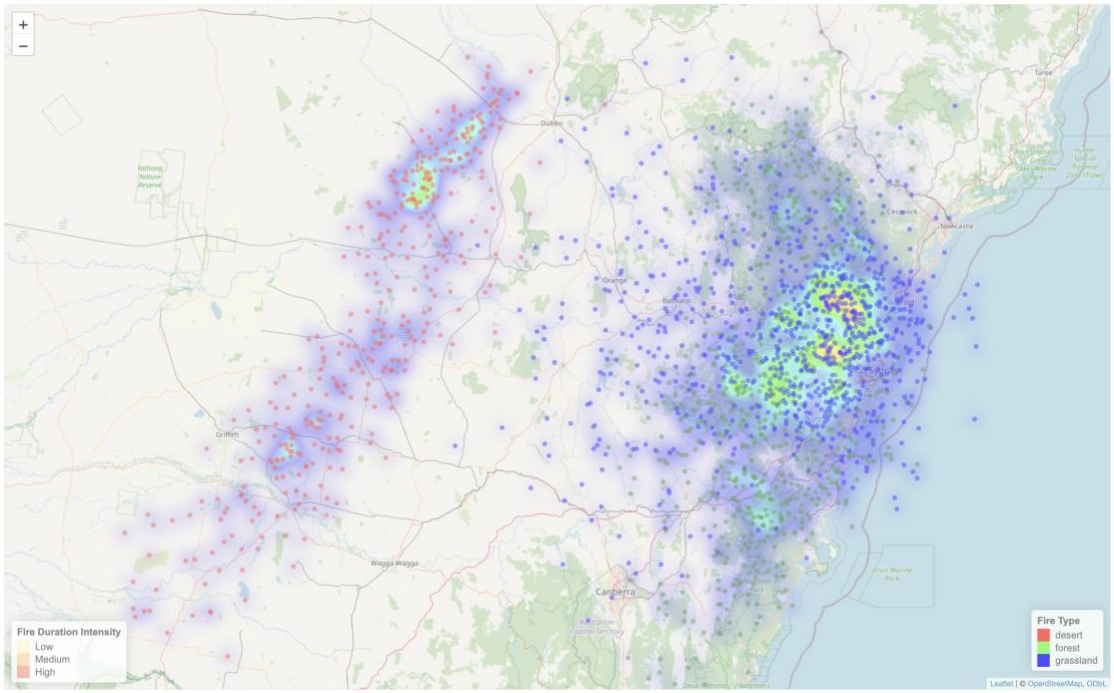



```r
# Compute and plot densities for specific fire types
f.f = density(data$latitude[data$type=="forest"], from=lowy, to=highy)
f.g = density(data$latitude[data$type=="grassland"], from=lowy, to=highy)
f.d = density(data$latitude[data$type=="desert"], from=lowy, to=highy)

# Plot the densities for specific types on the same graph
plot(f.f, col = "green", main = "Density of Latitude by Fire Type", xlab = "L
atitude", ylab = "Density",ylim = c(0, 1))
lines(f.g, col = "blue")
lines(f.d, col = "red")
legend("topleft", legend = c("Forest", "Grassland", "Desert"), col = c("green
", "blue", "red"), lty = 1)
abline(h=0.5, lty=2)
```




```r
# Create the map using leaflet
map <- leaflet(data) %>%
  addTiles() %>%  # Add default OpenStreetMap tiles
  addCircleMarkers(
    lng = ~longitude, lat = ~latitude,
    popup = paste("Type:", data$type,
                  "<br>Duration:", data$duration, "mins",
                  "<br>Temperature:", data$temp, "°C",
                  "<br>Humidity:", data$humidity, "%"),
    radius = 0.5, fillOpacity = 0.05, color = ~case_when(
      data$type == "desert" ~ "red",
      data$type == "forest" ~ "green",
      data$type == "grassland" ~ "blue",
      TRUE ~ "black" # Default color
    )
  ) %>%
  addHeatmap(lng = ~longitude, lat = ~latitude, intensity = ~sqrt(data$durati
on), blur = 20, max = 0.05, radius = 15) %>%
  setView(lng = mean(data$longitude), lat = mean(data$latitude), zoom = 7) %>
%
  addLegend("bottomright", pal = colorFactor(c("red", "green", "blue", "black
```

```
"), domain = c("desert", "forest", "grassland", "other")),
            values = ~data$type, title = "Fire Type", opacity = 1.0) %>%
  addLegend("bottomleft", title = "Fire Duration Intensity", labels = c("Low"
, "Medium", "High"),
            colors = c("#FFEDA0", "#FEB24C", "#F03B20"), opacity = 0.5
            )

# Print the map
# map # uncomment in the .rmd file to use a potential NSW map to view the loc
ations of fire
```



2. Task 1: Location of Fires

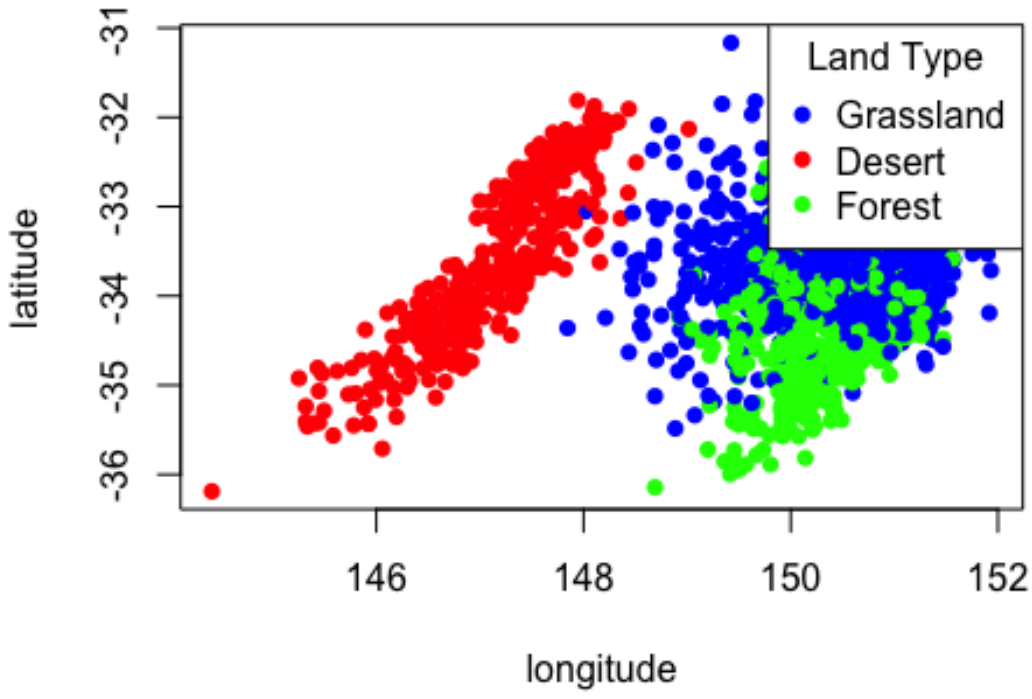
The Fire Rescue Services require density estimates for fire locations across NSW to help them position their stations. Plot and provide details of a model with computed coefficients that will allow them to compute the density estimate of a fire occurring, when given the location (latitude and longitude). Use the model to estimate the density at the WSU Parramatta South Campus.

H_0 : The geographical coordinates (latitude and longitude) have no significant effect on the density of fire occurrences in New South Wales (NSW).

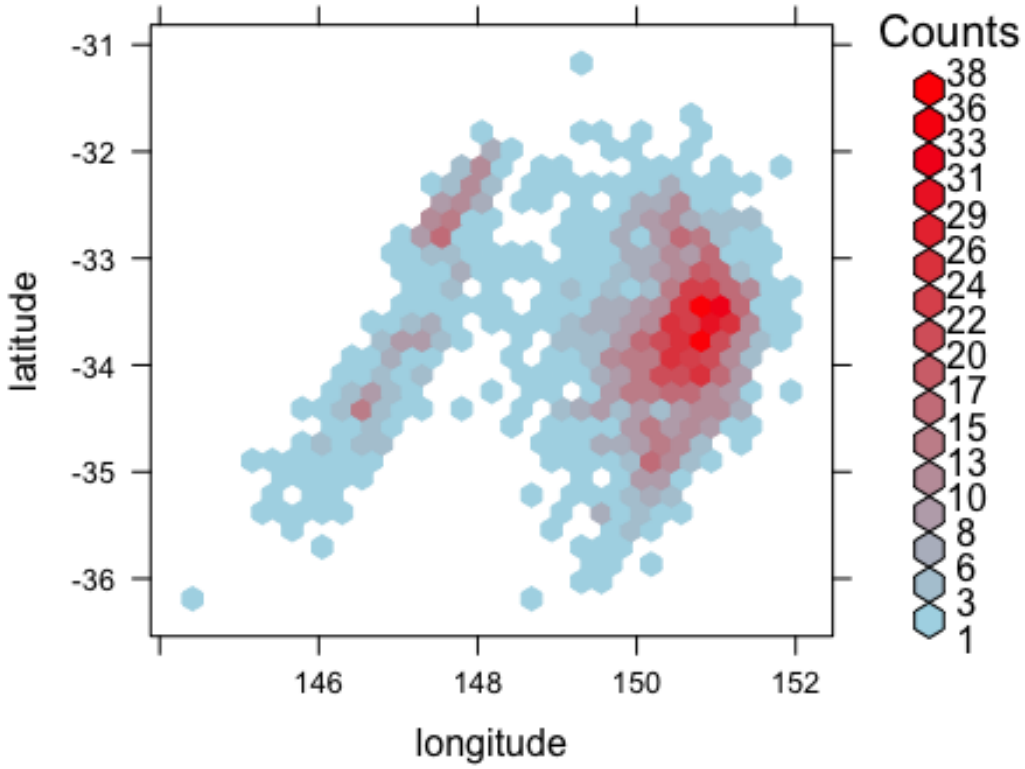
H_1 : The geographical coordinates (latitude and longitude) significantly affect the density of fire occurrences in New South Wales (NSW), influencing the strategic placement of fire rescue stations.

```
# Define a color mapping
colors <- ifelse(data$type == "grassland", "blue",
```

```
          ifelse(data$type == "desert", "red",
          ifelse(data$type == "forest", "green", NA)))

# Plot with specified colors
plot(latitude ~ longitude, data = data, pch = 16, col = colors)

# Add a legend to the plot
legend("topright",
       legend = c("Grassland", "Desert", "Forest"),
       col = c("blue", "red", "green"),
       pch = 16,
       title = "Land Type")
```




```
# Get the count of occurences
hexbinplot(latitude ~ longitude, data = data,
           colramp = function(n) colorRampPalette(c("lightblue", "red"))(n))
```



EM Algorithm for Mixtures of Multivariate Normals

boot.comp resulting in knit-crash issues.# uncomment in the r markdown to run the boot.comp

```
coords <- cbind(data$latitude, data$longitude)

# set.seed(13)
# sampleIndices <- sample(1:nrow(coords), 300, replace = FALSE)
# xSample <- coords[sampleIndices, ]
#
# obj <- boot.comp(xSample, max.comp = 4, mix.type = "mvnormalmix", maxit = 5
00, epsilon = 1e-1)
#
# print(obj$obs.log.lik)
# print(obj$p.values)
```

The boot.comp function in R is designed to estimate the number of components in a mixture model using statistical principles of bootstrapping and the Law of Large Numbers. By repeatedly resampling from the data, bootstrapping approximates the underlying distribution, capturing its characteristics without requiring the entire dataset. This resampling approach ensures that even a sufficiently large and random sample can approximate the overall population distribution, allowing boot.comp to reliably estimate the number of mixture components, as it represents the variance and structure of the data.

In boot.comp with the mvnormalmix (multivariate normal mixture) type, the Expectation-Maximization (EM) algorithm is employed to iteratively estimate mixture model parameters. The EM algorithm alternates between the Expectation (E-step), which calculates posterior probabilities for component membership based on current estimates, and the Maximization (M-step), which updates parameters (mean, covariance, and mixing proportions) to maximize likelihood. This iterative process continues until convergence, refining component estimates and enabling the identification of the optimal number of mixture components by accurately fitting the observed sample distribution.

The `boot.comp` function was used to determine the number of components in a multivariate normal mixture model for the given data sample (`xSampled`).

Parameters in `boot.comp`

- **`max.comp = 4`**: Sets the maximum number of components to test in the mixture model, up to 4, since we know there are 3 subgroups.
- **`mix.type = "mvnormalmix"`**: Specifies that the mixture model is multivariate normal.
- **`maxit = 500`**: Limits the maximum number of iterations for fitting the model to 500.
- **`epsilon = 1e-1`**: Sets the convergence threshold, meaning the algorithm stops if the improvement in log-likelihood is smaller than this value, the likelihood values are large for this particular data set.

P-values

The `p.values` vector contains the p-values corresponding to the hypothesis tests for different numbers of components. The values `[0.00, 0.02, 0.09]` represent models with 1, 2, and 3 components, respectively.

- **0.00 for 1 component**: This suggests that a single-component model is inadequate, rejecting the null hypothesis of 1 component.
- **0.02 for 2 components**: Indicates a significant improvement in fit when adding a second component, but there is still room for improvement.
- **0.09 for 3 components**: Suggests that the model with 3 components is not statistically different from a more complex model (e.g., with 4 components), implying that 3 components are sufficient to adequately describe the data.

Observed Log-Likelihood

The `obs.log.lik` vector shows the observed log-likelihood values for models with 1, 2, and 3 components, respectively. The values `[306.47742, 31.29849, 28.07490]` reflect the model's fit:

- **306.47742 for 1 component**: Represents the log-likelihood for a single-component model, indicating a poor fit.
- **31.29849 for 2 components**: Shows a significant increase in log-likelihood, suggesting a better fit with two components.
- **28.07490 for 3 components**: Further improves the fit, indicating that three components may be sufficient to capture the data structure.

The bootstrapping process identified that the optimal number of components, k, for the multivariate normal mixture model is 3. This indicates that the dataset is best represented by three distinct clusters, each modeled as a multivariate normal distribution.

```
set.seed(13)
mvnm <- mvnormalmixEM(coords, k = 3)

## number of iterations= 96

print(mvnm$lambda)

## [1] 0.07372982 0.79941762 0.12685256

print(mvnm$mu)

## [[1]]
## [1] -32.51256 147.70882
##
## [[2]]
## [1] -33.76988 150.44283
##
## [[3]]
## [1] -34.02369 146.95927

print(mvnm$sigma)

## [[1]]
##            [,1]       [,2]
## [1,] 0.08899355 0.07851446
## [2,] 0.07851446 0.09364042
##
## [[2]]
##            [,1]       [,2]
## [1,] 0.51767854 0.09998712
## [2,] 0.09998712 0.37543421
##
## [[3]]
##           [,1]      [,2]
## [1,] 0.5795997 0.5509203
## [2,] 0.5509203 0.6389261
```

Mixture Model and Density Estimation

Interpretation of Covariance Matrices in `mvnormalmixEM` Output

The output of `mvnm$sigma` represents the covariance matrices of the three components in the fitted multivariate normal mixture model. Each matrix describes the spread and orientation of the data distribution within each component in two-dimensional space (latitude and longitude). The covariance matrices provide insights into how variables interact and vary together within each component.

Covariance Matrices for Each Component

Component 1

- **Covariance Matrix**:

$$\begin{bmatrix} 0.08899353 & 0.07851445 \\ 0.07851445 & 0.09364042 \end{bmatrix}$$

This covariance matrix indicates a relatively small spread in both dimensions, suggesting that Component 1 is tightly clustered. The positive off-diagonal values (0.07851445) imply a positive correlation between the two variables (latitude and longitude), meaning that as one variable increases, the other tends to increase as well.

Component 2

- **Covariance Matrix**:

$$\begin{bmatrix} 0.5176786 & 0.0999871 \\ 0.0999871 & 0.3754343 \end{bmatrix}$$

This matrix shows a larger spread in both dimensions compared to Component 1, indicating that Component 2 has a more dispersed cluster. The positive off-diagonal value (0.0999871) suggests a weaker positive correlation between the variables, showing a mild tendency for both variables to increase together.

Component 3

- **Covariance Matrix**:

$$\begin{bmatrix} 0.5795997 & 0.5509202 \\ 0.5509202 & 0.6389258 \end{bmatrix}$$

Component 3 has the largest spread among the three components, as indicated by higher diagonal values. The off-diagonal values (0.5509202) suggest a strong positive correlation, implying that latitude and longitude are highly related within this component. This indicates an elongated cluster, where the variables exhibit strong co-variation.

The covariance matrices reveal that each component of the mixture model has a distinct spread and correlation structure. Component 1 is tightly clustered with a moderate positive correlation, Component 2 has a more moderate spread and weaker correlation, and Component 3 shows the largest spread with a strong positive correlation between the variables. These differences highlight the heterogeneity in the spatial distribution of the data across the three components.

Model for Density Estimation of Fire Occurrence

The density estimate of fire occurrence at a given location (x, y) can be calculated as a weighted sum of the densities of the three components:

$$f(x, y) = \lambda_1 \cdot f_1(x, y) + \lambda_2 \cdot f_2(x, y) + \lambda_3 \cdot f_3(x, y)$$

Here, $f_i(x, y)$ is the multivariate normal density for component i, defined as:

$$f_i(x, y) = \frac{1}{2\pi\sqrt{|\Sigma_i|}} \exp\left(-\frac{1}{2}(z - \mu_i)^T \Sigma_i^{-1}(z - \mu_i)\right)$$

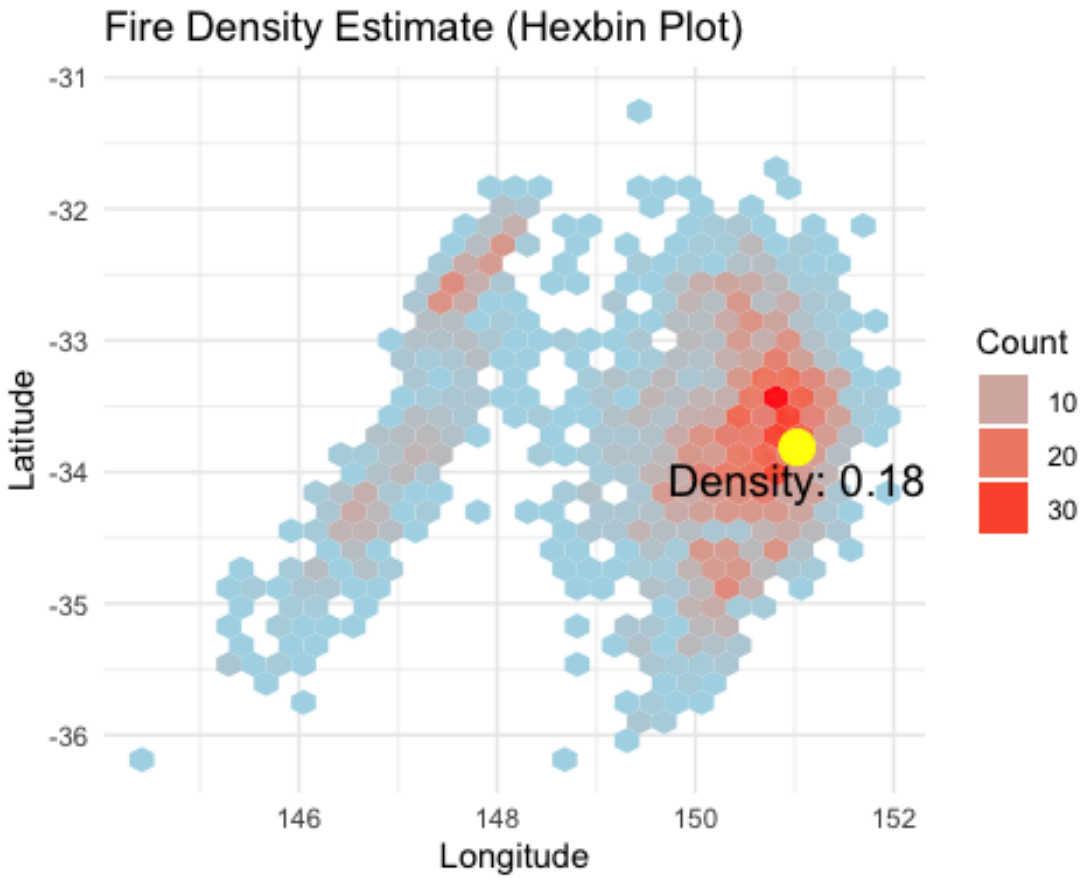
Where: - $z = \begin{bmatrix} x \\ y \end{bmatrix}$: The location vector for latitude and longitude. - μ_i: The mean vector for component i. - Σ_i: The covariance matrix for component i. - $|\Sigma_i|$: The determinant of the covariance matrix, which adjusts the scaling of the normal distribution.

This density function effectively captures the likelihood of fire occurrences across different locations by considering both the spatial distribution of historical occurrences and the spread of each cluster, as represented by the mixture components.

The fire density estimation function using the model

```
# Define a function to calculate density and plot for any given latitude and
longitude in NSW
plotFireDensity <- function(latitude, longitude) {
  # Function to calculate density at a given point
  mixtureDensity <- function(lat, lon, mvnm) {
    sum(sapply(1:length(mvnm$lambda), function(i) {
      mvnm$lambda[i] * dmvnorm(c(lat, lon), mean = mvnm$mu[[i]], sigma = mvnm
$sigma[[i]])
    }))
  }

  # Calculate the density at the provided coordinates
  densityAtLocation <- mixtureDensity(latitude, longitude, mvnm)

  # Generate a hexbin plot of the data
  p <- ggplot(data, aes(x = longitude, y = latitude)) +
    stat_binhex(aes(fill = ..count..), bins = 30) +
    scale_fill_gradient(low = "lightblue", high = "red", guide = "legend") +
    geom_point(aes(x = longitude, y = latitude), color = "grey", alpha = 0.00
5) +
    geom_point(aes(x = longitude, y = latitude), data = data.frame(longitude
= longitude, latitude = latitude), color = "yellow", size = 5) +
    geom_text(aes(x = longitude, y = latitude, label = sprintf("Density: %.2f
", densityAtLocation)),
              data = data.frame(longitude = longitude, latitude = latitude),
hjust = 0.5, vjust = 1.5, color = "black", size = 5) +
    labs(x = "Longitude", y = "Latitude", title = "Fire Density Estimate (Hex
bin Plot)", fill = "Count") +
    theme_minimal()

  # Print the plot
  print(p)
}

latWSU <- -33.81203745762795
longWSU <- 151.02499986441762150
plotFireDensity(latitude = latWSU, longitude = longWSU)

## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.
0.
```

```
## ℹ Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Western Sydney University Parramatta South Campus is in a region where over 30 fire incidents have been reported, and the estimated density at this location is 0.18, suggesting a medium likelihood of fire occurrence. This density value, derived from a multivariate normal mixture model, quantifies the concentration of fires at a specific point based on historical spatial patterns. While not an absolute probability, a density of 0.18 indicates a medium clustering of fires relative to other locations, helping identify areas of interest for risk assessment and resource allocation.

Conclusion

The mvnormalmixEM algorithm provides strong evidence to reject the null hypothesis- H_0 : The geographical coordinates (latitude and longitude) have no significant effect on the density of fire occurrences in New South Wales (NSW).

The mvnormalmixEM algorithm provides strong supporting evidence for the alternate hypothesis- H_1 : The geographical coordinates (latitude and longitude) significantly affect

the density of fire occurrences in New South Wales (NSW), influencing the strategic placement of fire rescue stations.

Two-Dimensional Kernel Density Estimation

```
# Define latitude and longitude vectors from dataset
longitude <- data$longitude # Longitude
latitude <- data$latitude   # Latitude

# Fit the KDE using kde2d
kdeFit <- kde2d(
  x = longitude,
  y = latitude,
  h = c(bandwidth.nrd(longitude), bandwidth.nrd(latitude)),
  n = 100
)

# WSU Parramatta coordinates
wsuParramattaLongitude <- 151.02499986441762150
wsuParramattaLatitude <- -33.81203745762795

# Interpolate the KDE estimate at the specified WSU Parramatta coordinates
# Interpolate along the x-axis (longitude)
kdeLon <- approx(kdeFit$x, kdeFit$z[, which.min(abs(kdeFit$y - wsuParramattaL
atitude))], xout = wsuParramattaLongitude)$y

# Interpolate along the y-axis (latitude)
kdeLat <- approx(kdeFit$y, kdeFit$z[which.min(abs(kdeFit$x - wsuParramattaLon
gitude)), ], xout = wsuParramattaLatitude)$y

# Compute the average of both interpolations as the KDE estimate at WSU Parra
matta
kdeWSU <- (kdeLon + kdeLat) / 2

# Display the KDE estimate
print(paste("KDE estimate at WSU Parramatta:", kdeWSU))

## [1] "KDE estimate at WSU Parramatta: 0.266352226792237"
```

The difference between the KDE and the multivariate normal mixture model lies in their statistical approaches to density estimation: KDE is a non-parametric method that uses a flexible, data-driven approach to capture local variations by placing a kernel (often Gaussian) on each data point, making it highly sensitive to nearby clusters and producing a localized density estimate based on bandwidth selection. In contrast, the mixture model is a parametric approach that assumes the data is generated from a finite number of Gaussian components, providing a smoother, global estimate by modeling broader patterns through weighted sums of Gaussian distributions. While the KDE is ideal for identifying finer local structures, the mixture model is more suited for capturing overall clusters in the data, each

characterized by its own mean and covariance matrix, making it less sensitive to small-scale variations but more interpretable in terms of distinct subpopulations.

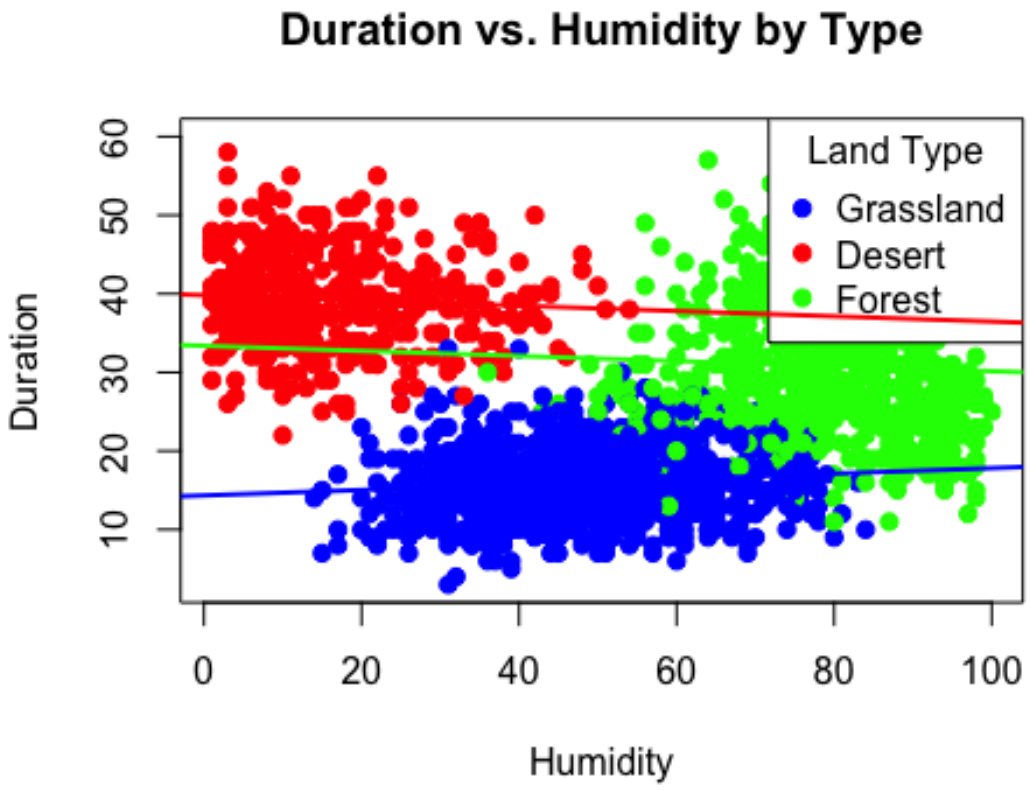
3. Task 2: Effect of Humidity on Duration

Information from the Fire Rescue Servies team shows that the duration of a fire is dependent on the humidity, but they are unsure if there are any hidden factors that have not been taken into account. Use a mixture of regressions to investigate if there are potential hidden factors.

H_0 : There are no underlying hidden factors while predicting the duration of fire based on humidity.

H_1 : There are underlying hidden factors while predicting the duration of fire based on humidity.

```
# Plot duration vs. humidity with custom colors
plot(data$humidity, data$duration,
     col = colors, pch = 19,
     xlab = "Humidity", ylab = "Duration",
     main = "Duration vs. Humidity by Type")

# Add legend
legend("topright",
       legend = c("Grassland", "Desert", "Forest"),
       col = c("blue", "red", "green"),
       pch = 19,
       title = "Land Type")


fit <- lm(duration~humidity*type, data = data) # fit a multiple linear model
to capture relationship between duration and humidity based on the type of fi
re.

cc = coef(fit) # extract the intercept & coefficients of the model.

# Fit the models on the plot
abline(cc[1], cc[2], col="red", lwd=2)
abline(cc[1]+cc[3], cc[2]+cc[5], col="green", lwd=2)
abline(cc[1]+cc[4], cc[2]+cc[6], col="blue", lwd=2)
```




```r
# Calculate variance of 'duration' and 'humidity' within each 'type'
typeSigma <- data %>%
  group_by(type) %>%
  summarise(
    durationVar = var(duration),  # Calculate variance for duration
    humidityVar = var(humidity),   # Calculate variance for humidity
    durationSigma = sd(duration),  # Calculate variance for duration
    humiditySigma = sd(humidity)   # Calculate variance for humidity
  )

# Print the variance for each group
print(typeSigma)
```

```
## # A tibble: 3 × 5
##   type      durationVar humidityVar durationSigma humiditySigma
##   <fct>           <dbl>       <dbl>         <dbl>         <dbl>
## 1 desert           38.1        138.          6.17          11.7
## 2 forest           75.7        135.          8.70          11.6
## 3 grassland        20.4        176.          4.52          13.3
```

```r
set.seed(4)
# Number of observations
n <- nrow(data)
```

```r
fitMix2 <- with(data, regmixEM(duration, humidity, k = 2, arbvar = TRUE)) # S
et arbvar to TRUE because the sigma values for the target variable differ sig
nificantly, with some being nearly double the others.

## number of iterations= 105

fitMix3 <- with(data, regmixEM(duration, humidity, k = 3, arbvar = TRUE))

## number of iterations= 722

fitMix4 <- with(data, regmixEM(duration, humidity, k = 4, arbvar = TRUE))

## Warning in anova.lm(X[[i]], ...): ANOVA F-tests on an essentially perfect
fit
## are unreliable

## Need new starting values due to singularity...
## number of iterations= 542

# Get summary to estimate the number of parameters
summary(fitMix2)

## summary of regmixEM object:
##          comp 1     comp 2
## lambda 0.589969  0.4100309
## sigma  4.695106  7.2544793
## beta1  9.712568 39.1450510
## beta2  0.139857 -0.0461692
## loglik at estimate:  -7187.401

summary(fitMix3)

## summary of regmixEM object:
##          comp 1    comp 2     comp 3
## lambda 0.312509  0.284683  0.4028080
## sigma  3.415847  3.442576  7.3389244
## beta1  8.374882 10.535903 38.9767191
## beta2  0.110896  0.186584 -0.0416006
## loglik at estimate:  -7168.426

summary(fitMix4)

## summary of regmixEM object:
##           comp 1   comp 2    comp 3     comp 4
## lambda 0.0330753 0.304005  0.259623  0.4032966
## sigma  1.9031523 3.188261  3.210490  7.3424184
## beta1  9.1364840 8.199735 10.915276 38.9512731
## beta2  0.0247540 0.124677  0.186943 -0.0412238
## loglik at estimate:  -7164.355
```

```r
# Extract log-likelihoods
logLik2 <- fitMix2$loglik
print(logLik2)
```

```
## [1] -7187.401
```

```r
logLik3 <- fitMix3$loglik
print(logLik3)
```

```
## [1] -7168.426
```

```r
logLik4 <- fitMix4$loglik
print(logLik4)
```

```
## [1] -7164.355
```

```r
print(fitMix2$lambda)
```

```
## [1] 0.5899691 0.4100309
```

```r
print(fitMix2$sigma)
```

```
## [1] 4.695106 7.254479
```

```r
print(fitMix2$beta)
```

```
##           comp.1      comp.2
## beta.0 9.7125676 39.14505105
## beta.1 0.1398574 -0.04616918
```

```r
print(fitMix3$lambda)
```

```
## [1] 0.3125091 0.2846829 0.4028080
```

```r
print(fitMix3$sigma)
```

```
## [1] 3.415847 3.442576 7.338924
```

```r
print(fitMix3$beta)
```

```
##           comp.1     comp.2      comp.3
## beta.0 8.3748820 10.5359035 38.97671912
## beta.1 0.1108962  0.1865836 -0.04160058
```

```r
print(fitMix4$lambda)
```

```
## [1] 0.03307526 0.30400525 0.25962293 0.40329656
```

```r
print(fitMix4$sigma)
```

```
## [1] 1.903152 3.188261 3.210490 7.342418
```

```r
print(fitMix4$beta)
```

```
##          comp.1    comp.2    comp.3      comp.4
## beta.0 9.136484 8.1997352 10.915276 38.95127313
## beta.1 0.024754 0.1246775  0.186943 -0.04122377

# Caculate AIC
aicM2 <- -2 * logLik2 + 2 * 7
print(aicM2)

## [1] 14388.8

aicM3 <- -2 * logLik3 + 2 * 11
print(aicM3)

## [1] 14358.85

aicM4 <- -2 * logLik4 + 2 * 15
print(aicM4)

## [1] 14358.71

# Caculate BIC
bicM2 <- -2 * logLik2 + log(n) * 7
print(bicM2)

## [1] 14427.86

bicM3 <- -2 * logLik3 + log(n) * 11
print(bicM3)

## [1] 14420.23

bicM4 <- -2 * logLik4 + log(n) * 15
print(bicM4)

## [1] 14442.41

# Print the values
cat("Log-Likelihood scores for Regression Mixture Models with k = 2, 3, 4 are
respectively\n",
    "LogLik M2 = ", logLik2,
    "LogLik M3 = ", logLik3,
    "LogLik M4 = ", logLik4)

## Log-Likelihood scores for Regression Mixture Models with k = 2, 3, 4 are r
espectively
##  LogLik M2 =  -7187.401 LogLik M3 =  -7168.426 LogLik M4 =  -7164.355

cat("AIC scores for Regression Mixture Models with k = 2,3,4 are respectively
\n",
    "AIC M2 = ", aicM2, "AIC M3 = ",aicM3, "AIC M4 = ", aicM4)

## AIC scores for Regression Mixture Models with k = 2,3,4 are respectively
##  AIC M2 =  14388.8 AIC M3 =  14358.85 AIC M4 =  14358.71
```

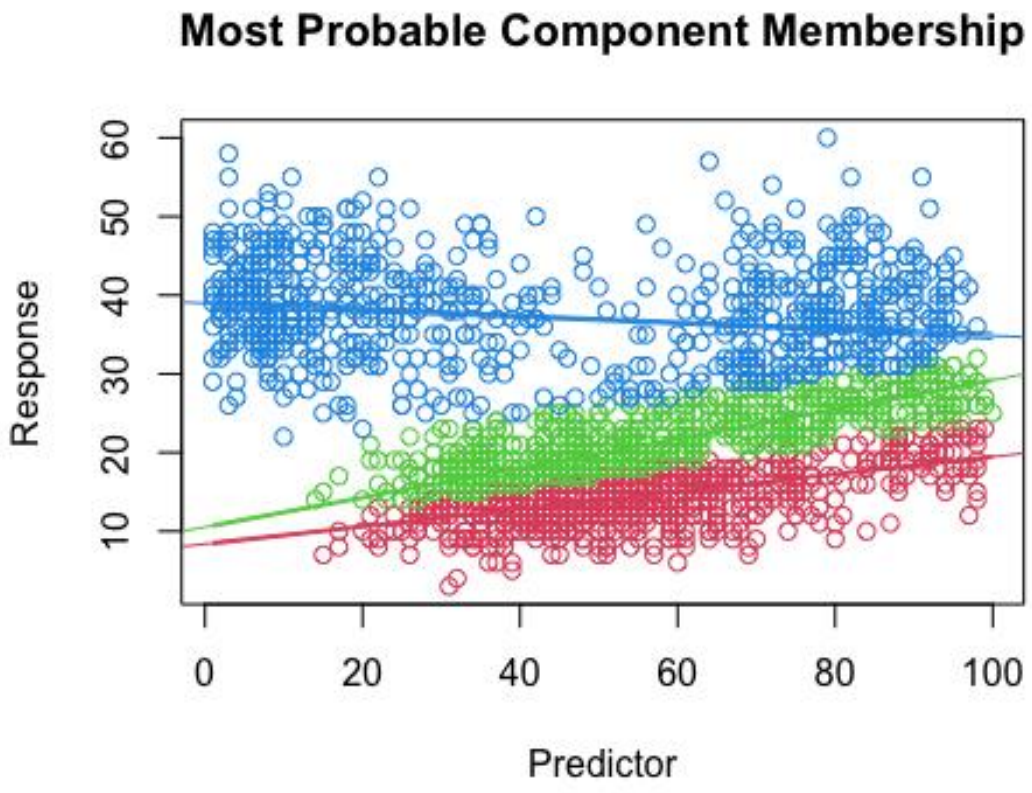
```
cat("BIC scores for Regression Mixture Models with k = 2, 3, 4 are respectively\n",
    "BIC M2 = ", bicM2,
    "BIC M3 = ", bicM3,
    "BIC M4 = ", bicM4)
```

```
## BIC scores for Regression Mixture Models with k = 2, 3, 4 are respectively
##  BIC M2 =  14427.86 BIC M3 =  14420.23 BIC M4 =  14442.41
```

To identify potential hidden factors, Expectation-Maximization (EM) Algorithm for Mixtures of Regressions was used.The `regmixEM` function, you applied mixture models with varying numbers of components (k=2, k=3, k=4) to explore hidden factors beyond just the primary variable (humidity). Log Likelihood values, AIC, and BIC were used to estimate the best value for k. The likelihood was minimal for k =4(-7164.35), but not a very significant difference compared to k=3 (-7168.426). AIC score for k=3(14358.85) can be considered since k=4(14358.71) is the same without considering the decimal point. Whereas, BIC is lowest for k=3(14420.23).

Overall analysis supports that optimal value for k is 3.

```
plot(fitMix3, whichplots = 2)
```



The analysis reveal complexities in the data that a single linear model might miss, and it supports the H_1 hypothesis that factors other than just humidity play a significant role in determining the duration of fires, such as the type of fire, since the three type of fires have 3 different distributions which influence the duration on top of humidity. The plot for k=3 depicts how the linear relationships are affected by the underlying hidden factor(type).

The mixture of regressions analysis reveals three clusters with distinct relationships between humidity and fire duration, suggesting hidden factors influencing fire behavior. The blue cluster (desert + forest) shows a slight negative relationship, indicating that increased humidity may reduce fire duration under certain conditions. In contrast, the green(grassland + forest) and red(grassland + forest) clusters display positive trends, where higher humidity correlates with longer fire duration, implying that in these cases, humidity alone does not dampen fires effectively. Humidity's effect on fire duration is not uniform; it varies based on underlying environmental conditions, such as vegetation and terrain, specific not only to each land type but more. This require further investigation such as environment conditions, wind speed, etc., and is not carried out in this project due to limitation of data.

Conclusion

The regmixEM provides strong evidence to reject H_0 : There are no underlying hidden factors while predicting the duration of fire based on humidity.

The regmixEM provides supporting evidence for H_1 : There are underlying hidden factors while predicting the duration of fire based on humidity.

4. Task 3: Allocating Equipment

The fire rescue services use fire retardants that are specific to the type of fire (either forest, grassland or desert fire retardant). The amount of retardant required is dependent on the duration of the fire. Over the past few years, the majority of fires faced have been from 35 to 45 minutes in duration and so the fire rescue services want to know what proportions of each type of retardant to purchase.

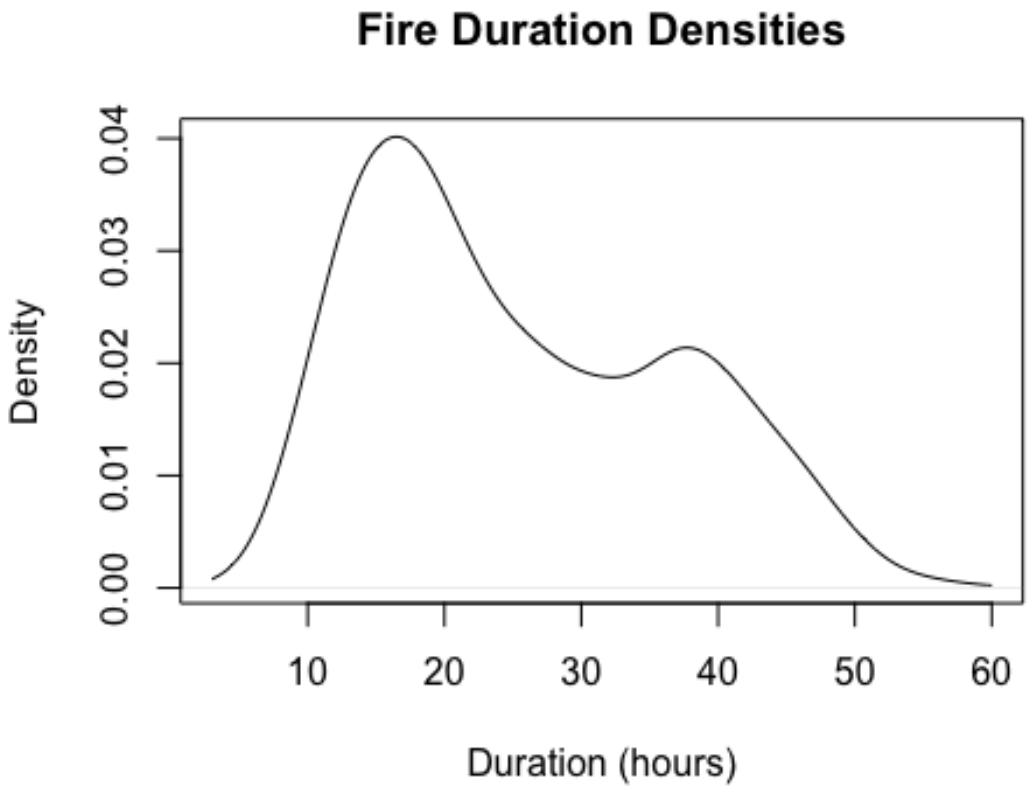
Compute the probability of each land type conditioned on duration of the fire, plot the result, and use the plot to provide estimates of the required proportions of retardant.

Null Hypothesis (H_0): The type of fire (forest, grassland, or desert) does not significantly influence the probability distribution of fire durations, implying that the required proportions of each fire retardant are independent of fire duration.

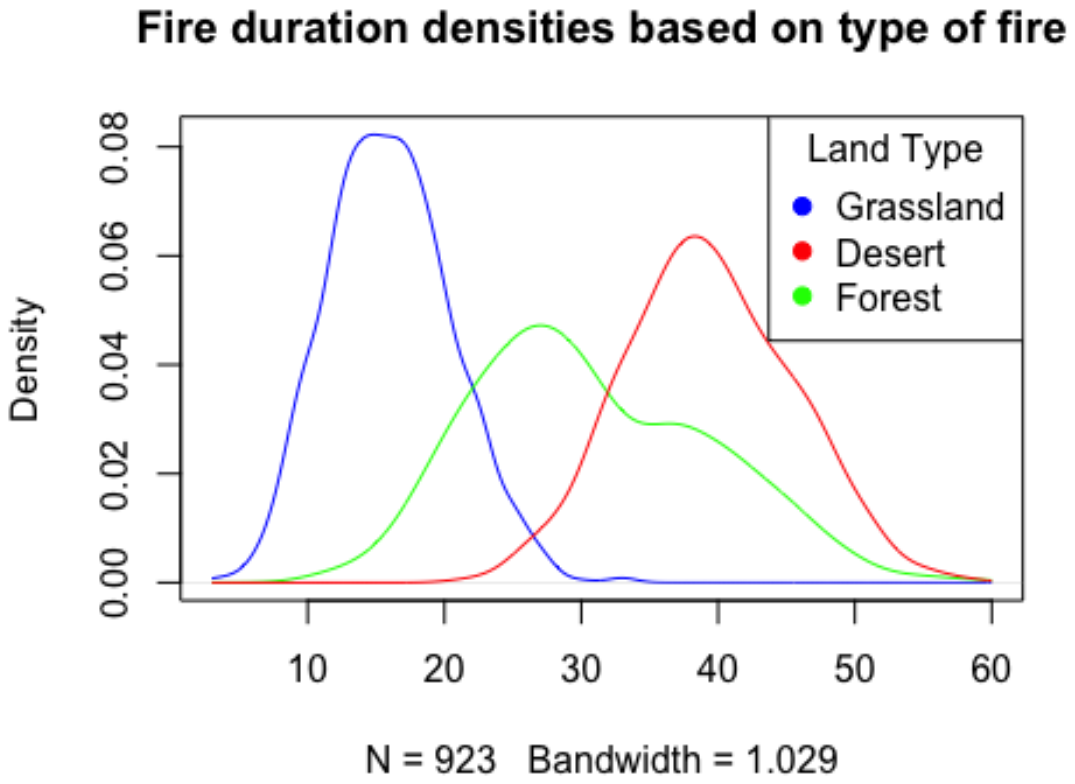
Alternative Hypothesis (H_1): The type of fire (forest, grassland, or desert) significantly influences the probability distribution of fire durations, indicating that the required proportions of each fire retardant are dependent on fire duration.

```
lo <- min(data$duration)  # Minimum fire duration
hi <- max(data$duration)  # Maximum fire duration

f <- density(data$duration, from = lo, to = hi)
```

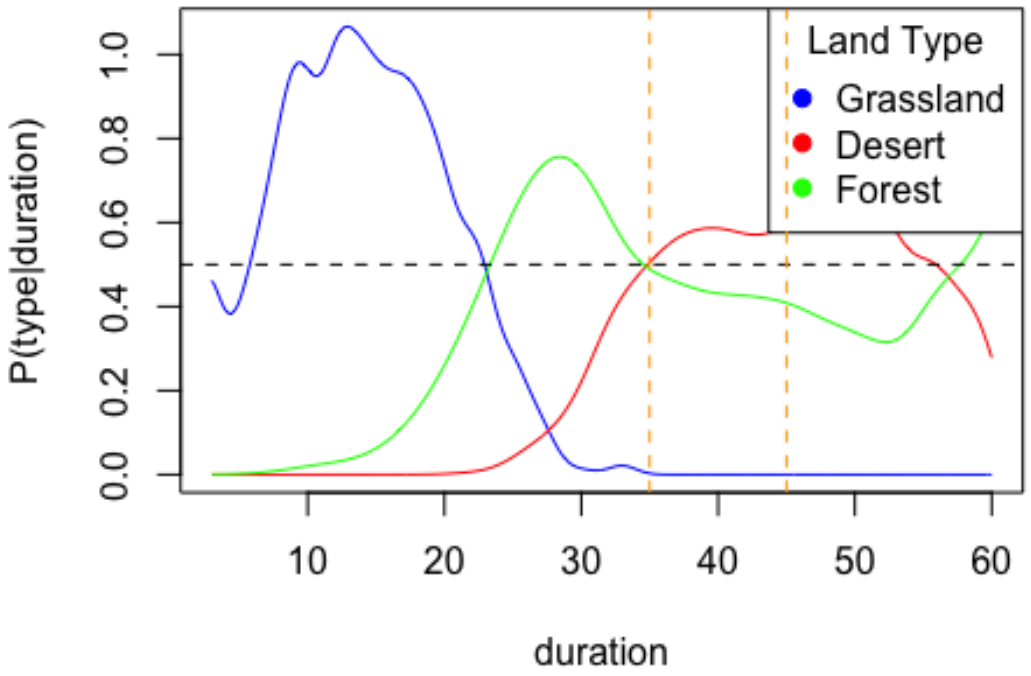
```
plot(f, main = "Fire Duration Densities", xlab = "Duration (hours)", ylab = "
Density")
```




```
p.g <- mean(data$type == "grassland")
p.f <- mean(data$type == "forest")
p.d <- mean(data$type == "desert")

f.g <- density(data$duration[data$type == "grassland"], from = lo, to = hi)
f.f <- density(data$duration[data$type == "forest"], from = lo, to = hi)
f.d <- density(data$duration[data$type == "desert"], from = lo, to = hi)



plot(f.g, col = "blue", main = "Fire duration densities based on type of fire
")
lines(f.f, col = "green")
lines(f.d, col = "red")

legend("topright",
       legend = c("Grassland", "Desert", "Forest"),
       col = c("blue", "red", "green"),
       pch = 19,
       title = "Land Type")
```




```
plot(f$x, p.g * f.g$y / f$y, type = "l", col = "blue", xlab = "duration", yla
b="P(type|duration)")
lines(f$x, p.d * f.d$y / f$y, type = "l", col = "red")
lines(f$x, p.f * f.f$y / f$y, col = "green")

legend("topright",
       legend = c("Grassland", "Desert", "Forest"),
       col = c("blue", "red", "green"),
       pch = 19,
       title = "Land Type")

abline(h = 0.5, lty = 2)
abline(v = c(35,45), lty = 2, col = "orange")
```



The overall density of fires is highest in grasslands, followed by deserts and then forest land types However, fire retardants are purchased based on the duration of the fire, as the amount of retardant used is proportional to the fire's duration; the longer the fire, the more retardant needed.

This plot presents the density estimates of fire durations for three land types: Grassland, Desert, and Forest. Fires in Grassland (blue) have the shortest duration, peaking sharply around 10-15 mins, indicating more consistent duration. In contrast, Desert fires (red) have the longest duration, with a broader peak around 30-40 mins, suggesting more variability and extended burning times. Forest fires (green) exhibit moderate duration, with the density peaking around 20-25 mins. Grassland fires are the shortest, desert fires the longest, and forest fires are in between, with varying spread across the land types.

```
# Set the specified duration range
duration_range <- c(35, 45)

# Calculate conditional probabilities for each type within the specified rang
e
prob.g <- integrate(approxfun(f.g$x, p.g * f.g$y / f$y), lower = duration_ran
ge[1], upper = duration_range[2])$value
prob.d <- integrate(approxfun(f.d$x, p.d * f.d$y / f$y), lower = duration_ran
ge[1], upper = duration_range[2])$value
prob.f <- integrate(approxfun(f.f$x, p.f * f.f$y / f$y), lower = duration_ran
```

```
ge[1], upper = duration_range[2])$value

# Calculate total probability to normalize
total_prob <- prob.g + prob.d + prob.f

# Proportion of each type of retardant required
prop.g <- prob.g / total_prob
prop.d <- prob.d / total_prob
prop.f <- prob.f / total_prob

# Display the results
cat("Proportion of Grassland retardant:", round(prop.g, 5), "\n")

## Proportion of Grassland retardant: 0.00013

cat("Proportion of Desert retardant:", round(prop.d, 5), "\n")

## Proportion of Desert retardant: 0.5652

cat("Proportion of Forest retardant:", round(prop.f, 5), "\n")

## Proportion of Forest retardant: 0.43466
```

Based on the provided information that the majority of fires have durations between 35 and 45 minutes, we can estimate the proportion of fire retardants to purchase. When the fire department places an order for 100 retardants, 56 out of 100 should be for desert fires, 43 for forest fires, and 1 for grassland fires, since placing an order for 0 grassland fire retardants wont make sense. This is a biased estimator ignoring the fact that grassland type fire are the most occurring in number. Making the ratio 5:4:1 (Desert:Forest:Grassland)

However, based on the plot, an estimated proportion would be 50% - 57 % desert retardants, 40% - 43% forest retardants, and 3% - 7% grassland retardants, as the probability of grassland fires decreases just before the 35 to 45-minute interval. It's also important to consider that grassland fires are the most frequent, followed by desert, and then fire.

Conclusion:

The statistical tests helped to estimate the the required proportions providing strong evidence to reject the null hypothesis (H_0): The type of fire (forest, grassland, or desert) does not significantly influence the probability distribution of fire durations, implying that the required proportions of each fire retardant are independent of fire duration.

The statistical tests helped to estimate the the required proportions providing strong supporting evidence for alternative hypothesis (H_1): The type of fire (forest, grassland, or desert) significantly influences the probability distribution of fire durations, indicating that the required proportions of each fire retardant are dependent on fire duration.

5. Task 4:Seasonal Effect

Finally, the head of fire safety has realised that the collected data does not have the dates of the fires recorded and that the time of year is likely to effect the duration of a fire. Fit a mixture model to the temperature variable to determine if seasonality can be identified.

Null Hypothesis (H_0): The temperature variable in the collected data does not exhibit distinct seasonal patterns that significantly affect the duration of a fire.

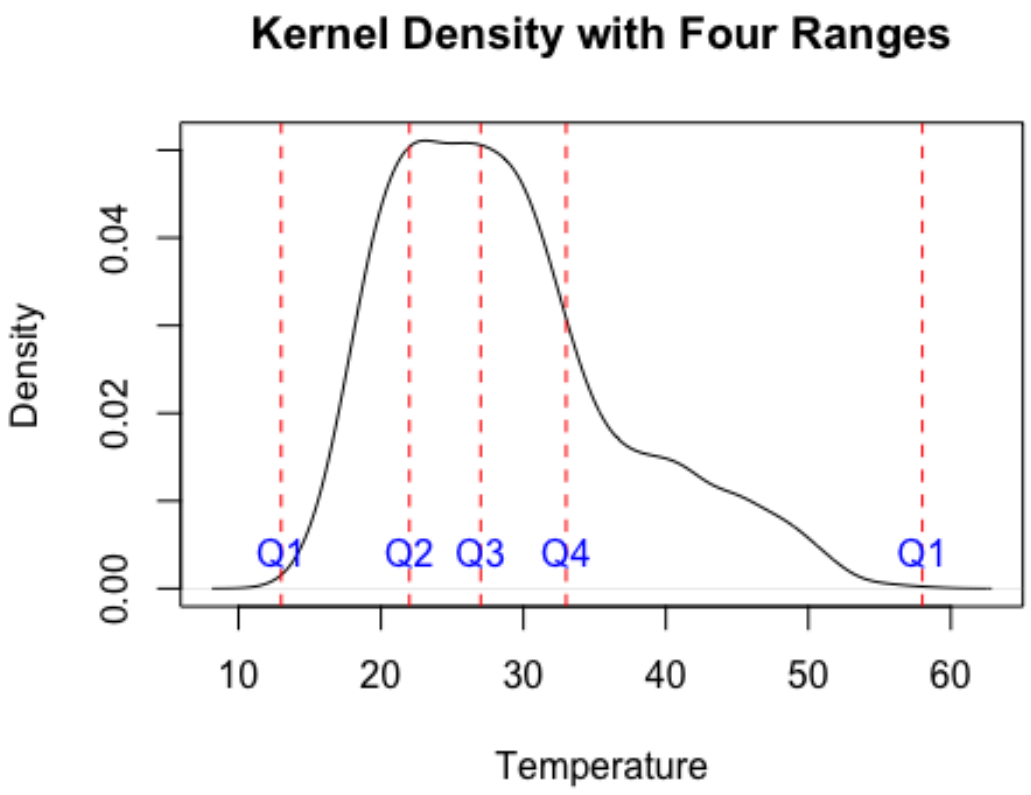
Alternative Hypothesis (H_1): The temperature variable in the collected data exhibits distinct seasonal patterns, indicating that the seasonality can be identified and is likely to affect the duration of a fire.

boot.comp has crashing issues while knitting. (uncomment the .rmd to run)

```
# temp.boot <- boot.comp(data$temp, max.comp = 4, mix.type = "normalmix", ver
bose =FALSE) # max.comp set to 4 because max number of seasons is 4
# print(temp.boot$p.values)
```

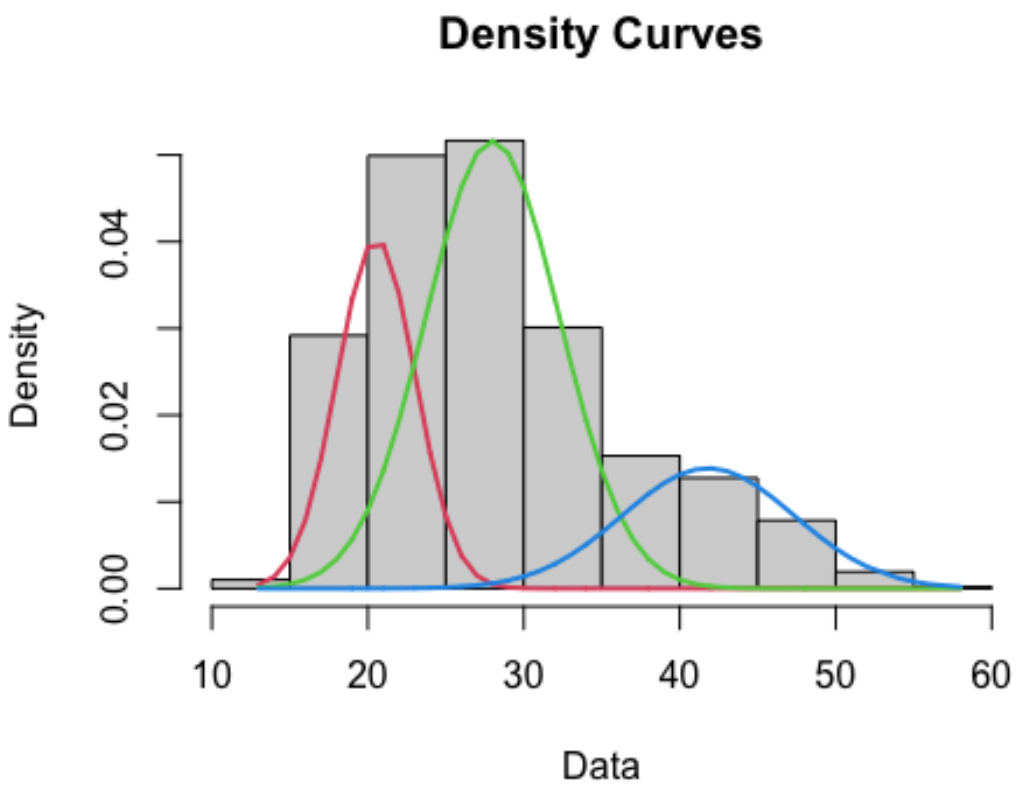
The boot.comp suggest that the optimal value for k is 3. This suggests that the four seasons cannot be fully identified from the temperature data because the temperature distribution is nearly the same for the first two mixing components, which are estimated to represent the temperature distributions of autumn and spring, making it difficult to distinguish the distribution of the fourth season.

```
par(mfrow=c(1,1))
# Calculate quartiles and assign ranges
quartiles <- quantile(data$temp, probs = seq(0, 1, 0.25))
tempRange <- cut(data$temp, breaks = quartiles, include.lowest = TRUE, labels
= paste0("Q", 1:4))

# Display quartile distribution
print(table(tempRange))

## tempRange
##  Q1  Q2  Q3  Q4
## 495 500 508 456

# Plot density with quartile boundaries
plot(density(data$temp), main = "Kernel Density with Four Ranges", xlab = "Te
mperature")
abline(v = quartiles, col = "red", lty = 2)
text(quartiles, 0, labels = paste0("Q", 1:4), pos = 3, col = "blue")
```

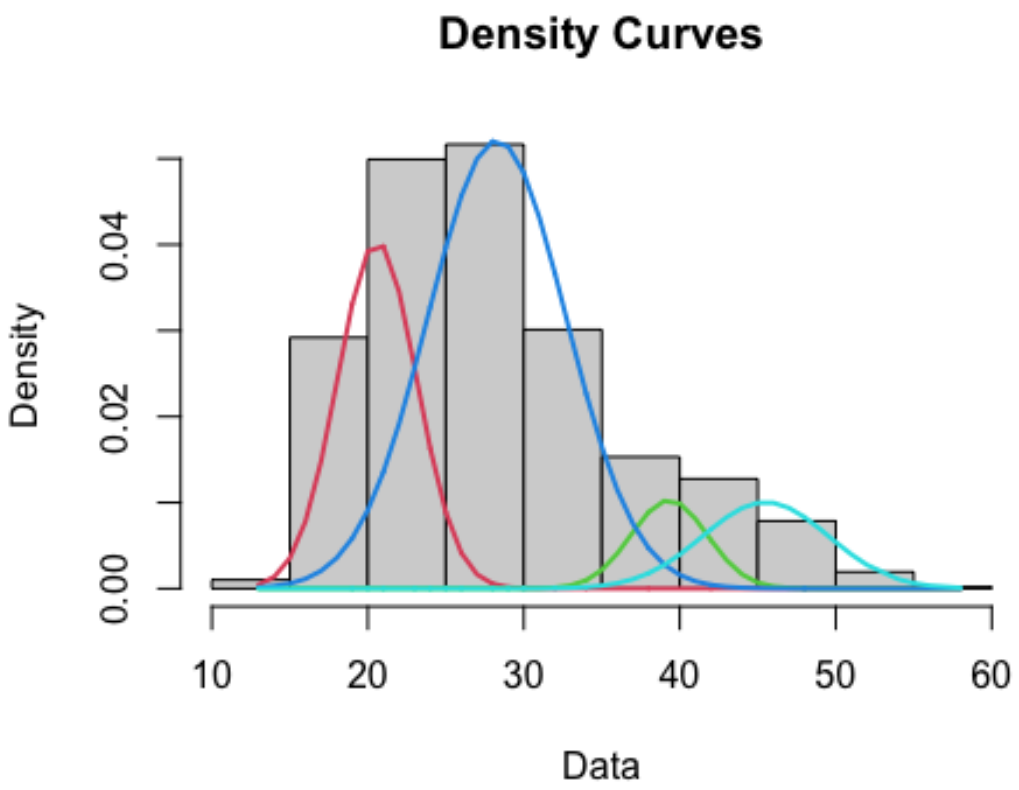


Manual fitting for k =3 and k =4 to investigate if four seasons can be identified.

```
m3.temp <- normalmixEM(data$temp, k=3)

## number of iterations= 535

summary(m3.temp)

## summary of normalmixEM object:
##           comp 1    comp 2    comp 3
## lambda  0.25392  0.554181  0.191899
## mu     20.53706 27.999106 41.819724
## sigma   2.51428  4.284208  5.515870
## loglik at estimate:  -6700.701

plot(m3.temp, whichplots=2)
```




```
m4.temp <- normalmixEM(data$temp, k=4, maxit = 1e5)

## number of iterations= 1679

summary(m4.temp)

## summary of normalmixEM object:
##           comp 1     comp 2    comp 3   comp 4
## lambda  0.256067  0.0630412  0.579202  0.10169
## mu     20.591000 39.3203429 28.276280 45.50455
## sigma   2.533985  2.4451557  4.434655  4.03722
## loglik at estimate:  -6698.756

plot(m4.temp, whichplots=2)
```

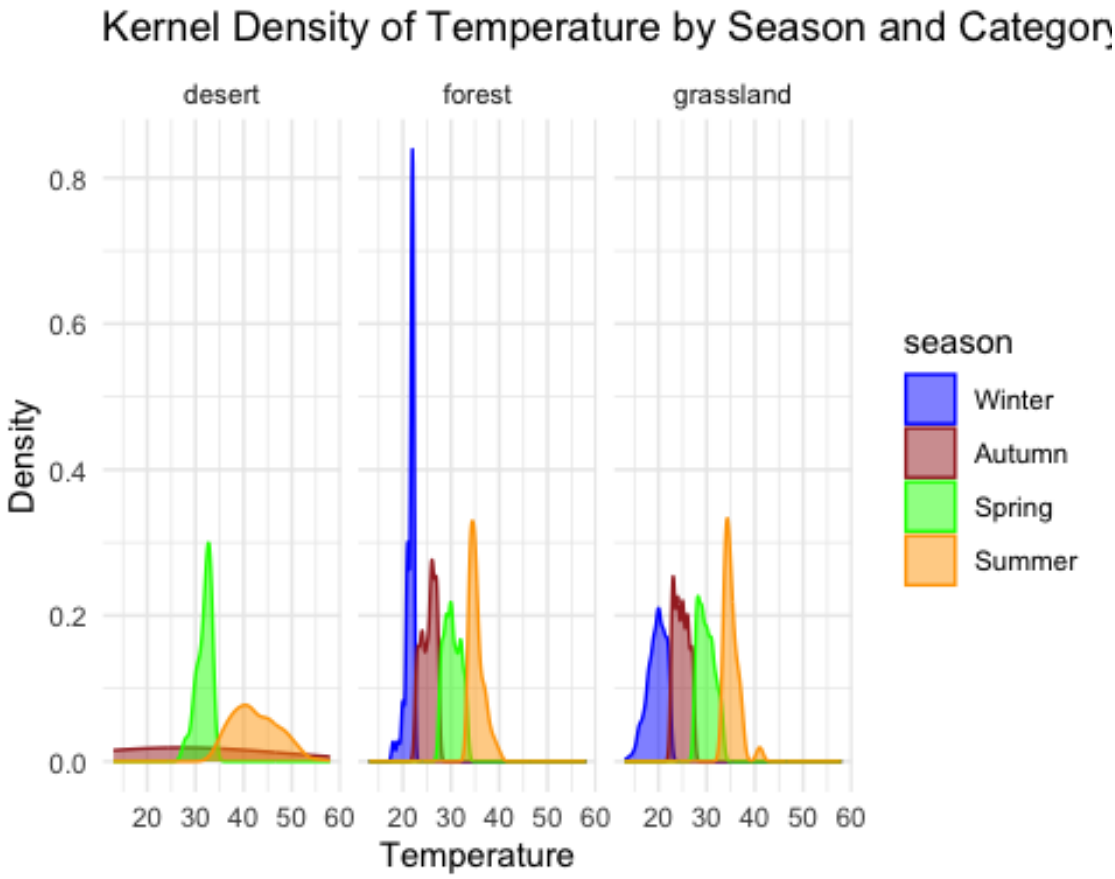



```
print(m3.temp$lambda)

## [1] 0.2539197 0.5541812 0.1918991

print(m3.temp$mu)

## [1] 20.53706 27.99911 41.81972

print(m3.temp$sigma)

## [1] 2.514279 4.284208 5.515870

print(m4.temp$lambda)

## [1] 0.25606695 0.06304121 0.57920173 0.10169011

print(m4.temp$mu)

## [1] 20.59100 39.32034 28.27628 45.50455

print(m4.temp$sigma)

## [1] 2.533985 2.445156 4.434655 4.037225

aic = c(-2*m3.temp$loglik + 2*(3*3-1),
        -2*m4.temp$loglik + 2*(3*4-1))
```

```
n = length(data$temp)

bic = c(-2*m3.temp$loglik + log(n)*(3*3-1),
        -2*m4.temp$loglik + log(n)*(3*4-1))

cat("AIC for m3.temp and m4.temp are", aic, "\n")

## AIC for m3.temp and m4.temp are 13417.4 13419.51

cat("BIC for m3.temp and m4.temp are", bic)

## BIC for m3.temp and m4.temp are 13462.04 13480.89

T1 = as.numeric(2*(m4.temp$loglik - m3.temp$loglik))
p.val = 1 - pchisq(T1,1)
print(p.val)

## [1] 0.04856153
```

All the methods indicate that there is no statistically significant improvement in the fit of the model with k = 4 compared to the simpler model with k =3 since p-value for likelihood ratio test is less than 0.05. The four seasons cannot be completely identified from temperature data due to the underlying distribution. The range of values for both Autumn and Spring could be in a similar range and follows a similar distribution which could be the cause of the limitation.

```
# Calculate quartiles and assign to seasons
quartiles <- quantile(data$temp, probs = c(0, 0.25, 0.5, 0.75, 1))
season <- cut(data$temp, breaks = quartiles, include.lowest = TRUE,
              labels = c("Winter", "Autumn", "Spring", "Summer"))


# Plot based on type and temperature
ggplot(data, aes(x = temp, fill = season, color = season)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("blue", "brown", "green", "orange")) +
  scale_color_manual(values = c("blue", "brown", "green", "orange")) +
  labs(title = "Kernel Density of Temperature by Season and Category",
       x = "Temperature",
       y = "Density") +
  facet_wrap(~ type) +  # Facet by the 'type' variable in 'data'
  theme_minimal()
```



However, using quartile-based estimation without statistical packages helped reveal insights into the distribution of fires based on the season. The plot confirms the occurrence of bushfires in NSW during the 2019–20 bushfire season, which was the largest on record in the state.

The kernel density plot shows temperature distributions across different seasons (Winter, Autumn, Spring, and Summer) and land cover categories (desert, forest, grassland). In the desert, temperatures peak sharply around 35°C in Spring and broaden to about 40°C in Summer, while Winter and Autumn have lower densities, indicating cooler, less variable temperatures. In forest areas, Winter has a pronounced peak around 20°C, suggesting consistent cooler temperatures, while Autumn and Spring show moderate peaks between 20°C and 35°C, with Summer peaking at 35°C, indicating warmer conditions. Grasslands follow a similar pattern, with Winter peaking around 20°C, Spring and Autumn between 25°C and 35°C, and Summer peaking near 35°C. Overall, the plot highlights clear seasonal temperature variations, with Summer showing the highest temperatures across all land types and Winter the lowest, while desert temperatures are more extreme compared to forests and grasslands.

Conclusion

Statistical testing using normalmixEM doesn't provide enough evidence to reject the null hypothesis (H_0): The temperature variable in the collected data does not exhibit distinct seasonal patterns that significantly affect the duration of a fire.

Therefore, we have enough evidence to support the alternative hypothesis (H_1): The temperature variable in the collected data exhibits distinct seasonal patterns, indicating that the seasonality can be identified and is likely to affect the duration of a fire.