PROJECT

2024-09-25

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Q1: Car accident dataset Q1.1: How many rows and columns are in the data? Provide the output from your R Studio

```
# Load the dataset
car_accident_data <- read.csv("car_accidents_victoria.csv")
head(car_accident_data)
```

```
##            X EASTERN.REGION     X.1      X.2   X.3
## 1       DATE          FATAL SERIOUS NOINJURY OTHER
## 2 01/01/2016              0       1        0     5
## 3 02/01/2016              0       3        0     1
## 4 03/01/2016              0       1        0     5
## 5 04/01/2016              0       2        0     2
## 6 05/01/2016              0       1        0     6
##   METROPOLITAN.NORTH.WEST.REGION     X.4      X.5   X.6
## 1                          FATAL SERIOUS NOINJURY OTHER
## 2                              0       7        0     7
## 3                              0       2        0     4
## 4                              0       3        0     6
## 5                              0       2        0     5
## 6                              1       2        0     7
##   METROPOLITAN.SOUTH.EAST.REGION     X.7      X.8   X.9 NORTH.EASTERN.REGION
## 1                          FATAL SERIOUS NOINJURY OTHER                FATAL
## 2                              1       9        0     3                    0
## 3                              0       8        0     5                    1
## 4                              0       7        0     4                    0
## 5                              0       4        0     5                    0
## 6                              0       6        0     7                    0
##       X.10     X.11  X.12 NORTHERN.REGION    X.13     X.14  X.15
## 1 SERIOUS NOINJURY OTHER           FATAL SERIOUS NOINJURY OTHER
## 2       3        0     2               0       1        0     1
## 3       2        0     1               0       2        0     1
## 4       2        0     3               0       0        0     3
## 5       1        0     1               1       4        0     3
## 6       6        0     2               0       0        0     3
##   SOUTH.WESTERN.REGION    X.16     X.17  X.18 WESTERN.REGION    X.19     X.20
## 1                FATAL SERIOUS NOINJURY OTHER          FATAL SERIOUS NOINJURY
## 2                    0       2        0     0              0       1        0
## 3                    0       2        0     2              0       1        0
## 4                    0       2        0     1              0       2        0
## 5                    0       1        0     3              0       0        0
## 6                    0       3        0     1              0       2        0
##    X.21
## 1 OTHER
## 2     0
## 3     2
## 4     4
## 5     1
## 6     1
```

```r
# Get the number of rows and columns
dim(car_accident_data)
```

```
## [1] 1644   29
```

the data have 1644 column and 29 row based on the r code above, but the actual data have only 1642 column and 29 row, excluding the 2 header.

Q1.1: What data types are in the data? Use data type selection tree and provide detailed explanation.

```r
# Check the data types of each column
data <- read.csv("car_accidents_victoria.csv", skip = 2)
str(data)
```

```
## ’data.frame’:    1642 obs. of  29 variables:
##  $ X01.01.2016: chr  "02/01/2016" "03/01/2016" "04/01/2016" "05/01/2016" ...
##  $ X0         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ X1         : int  3 1 2 1 2 0 1 1 1 0 ...
##  $ X0.1       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X5         : int  1 5 2 6 1 2 1 5 2 4 ...
##  $ X0.2       : int  0 0 0 1 0 0 0 0 1 0 ...
##  $ X7         : int  2 3 2 2 5 3 5 7 0 1 ...
##  $ X0.3       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X7.1       : int  4 6 5 7 13 6 10 10 9 7 ...
##  $ X1.1       : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ X9         : int  8 7 4 6 7 10 4 5 3 3 ...
##  $ X0.4       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X3         : int  5 4 5 7 8 12 4 3 8 6 ...
##  $ X0.5       : int  1 0 0 0 0 0 0 1 0 0 ...
##  $ X3.1       : int  2 2 1 6 0 0 0 1 0 1 ...
##  $ X0.6       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X2         : int  1 3 1 2 0 0 2 3 2 3 ...
##  $ X0.7       : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ X1.2       : int  2 0 4 0 3 2 0 0 1 2 ...
##  $ X0.8       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1.3       : int  1 3 3 3 4 2 3 1 2 1 ...
##  $ X0.9       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X2.1       : int  2 2 1 3 1 1 1 0 5 2 ...
##  $ X0.10      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X0.11      : int  2 1 3 1 3 1 2 2 2 3 ...
##  $ X0.12      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1.4       : int  1 2 0 2 2 1 1 1 0 0 ...
##  $ X0.13      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X0.14      : int  2 4 1 1 1 2 2 2 3 2 ...
```

```r
head(data)
```

```
##   X01.01.2016 X0 X1 X0.1 X5 X0.2 X7 X0.3 X7.1 X1.1 X9 X0.4 X3 X0.5 X3.1 X0.6 X2
## 1  02/01/2016  0  3    0  1    0  2    0    4    0  8    0  5    1    2    0  1
## 2  03/01/2016  0  1    0  5    0  3    0    6    0  7    0  4    0    2    0  3
## 3  04/01/2016  0  2    0  2    0  2    0    5    0  4    0  5    0    1    0  1
## 4  05/01/2016  0  1    0  6    1  2    0    7    0  6    0  7    0    6    0  2
## 5  06/01/2016  0  2    0  1    0  5    0   13    0  7    0  8    0    0    0  0
## 6  07/01/2016  0  0    0  2    0  3    0    6    0 10    0 12    0    0    0  0
##   X0.7 X1.2 X0.8 X1.3 X0.9 X2.1 X0.10 X0.11 X0.12 X1.4 X0.13 X0.14
## 1    0    2    0    1    0    2     0     2     0    1     0     2
## 2    0    0    0    3    0    2     0     1     0    2     0     4
## 3    1    4    0    3    0    1     0     3     0    0     0     1
## 4    0    0    0    3    0    3     0     1     0    2     0     1
## 5    0    3    0    4    0    1     0     3     0    2     0     1
## 6    0    2    0    2    0    1     0     1     0    1     0     2
```

When i read the data, i skip the first 2 column because there are 2 level of row. The data type of first column (DATE COULUMN) is character and the other column (accident count) is integer.

Data Type Selection Tree: Categorical Data a.Ordinal: DATE

Numerical Data (stored as strings, needs conversion) a. Discrete: Accident counts (FATAL, SERIOUS, etc.)

Detailed Explanation:

Categorical Data: Ordinal: DATE: While stored as strings, represents ordered time points.

Numerical Data: Discrete: Accident counts for each type (FATAL, SERIOUS, NOINJURY, OTHER), represent whole numbers (integers). Etc: “0”, “1”, “2”, etc. in the data columns

Hence, For proper analysis: DATE should be converted to a date format

This mixed data structure requires appropriate data tidying for meaningful statistical analysis. This data structure indicates that the CSV file was read without properly handling the multi-row header. As a result, all data, including date, are being treated as character strings. As a result, this conversion is necessary to perform meaningful statistical analyses, hence, question 2 will be tidying the column.

Q1.3: How many regions are in the data? What time period does the data cover? Provide the output from your R Studio (2 point)

```r
# Identify unique regions
regions <- c("EASTERN.REGION", "METROPOLITAN.NORTH.WEST.REGION", "METROPOLITAN.SOUTH.EAST.REGION",
             "NORTH.EASTERN.REGION", "NORTHERN.REGION", "SOUTH.WESTERN.REGION", "WESTERN.REGION")

# Count the number of regions
num_regions <- length(regions)

# Print the number of regions
print(paste("Number of regions:", num_regions))
```

```
## [1] "Number of regions: 7"
```

```
# Extract the date range
dates <- car_accident_data$X[-1]  # Exclude the first row which contains column names
start_date <- min(as.Date(dates, format="%d/%m/%Y"))
end_date <- max(as.Date(dates, format="%d/%m/%Y"))

# Print the date range
print(paste("Date range: From", start_date, "to", end_date))
```

```
## [1] "Date range: From 2016-01-01 to 2020-06-30"
```

There are 7 regions in the data. The data covers the time period from January 1, 2016 to June 30, 2020.

Q1.4: What do the variables FATAL and SERIOUS represent? What's the difference between them? (3 points)

```
# First, let's look at the structure of our data
str(car_accident_data)
```

```
## ’data.frame':    1644 obs. of  29 variables:
##  $ X                             : chr  "DATE" "01/01/2016" "02/01/2016" "03/01/2016" ...
##  $ EASTERN.REGION                : chr  "FATAL" "0" "0" "0" ...
##  $ X.1                           : chr  "SERIOUS" "1" "3" "1" ...
##  $ X.2                           : chr  "NOINJURY" "0" "0" "0" ...
##  $ X.3                           : chr  "OTHER" "5" "1" "5" ...
##  $ METROPOLITAN.NORTH.WEST.REGION: chr  "FATAL" "0" "0" "0" ...
##  $ X.4                           : chr  "SERIOUS" "7" "2" "3" ...
##  $ X.5                           : chr  "NOINJURY" "0" "0" "0" ...
##  $ X.6                           : chr  "OTHER" "7" "4" "6" ...
##  $ METROPOLITAN.SOUTH.EAST.REGION: chr  "FATAL" "1" "0" "0" ...
##  $ X.7                           : chr  "SERIOUS" "9" "8" "7" ...
##  $ X.8                           : chr  "NOINJURY" "0" "0" "0" ...
##  $ X.9                           : chr  "OTHER" "3" "5" "4" ...
##  $ NORTH.EASTERN.REGION          : chr  "FATAL" "0" "1" "0" ...
##  $ X.10                          : chr  "SERIOUS" "3" "2" "2" ...
##  $ X.11                          : chr  "NOINJURY" "0" "0" "0" ...
##  $ X.12                          : chr  "OTHER" "2" "1" "3" ...
##  $ NORTHERN.REGION               : chr  "FATAL" "0" "0" "0" ...
##  $ X.13                          : chr  "SERIOUS" "1" "2" "0" ...
##  $ X.14                          : chr  "NOINJURY" "0" "0" "0" ...
##  $ X.15                          : chr  "OTHER" "1" "1" "3" ...
##  $ SOUTH.WESTERN.REGION          : chr  "FATAL" "0" "0" "0" ...
##  $ X.16                          : chr  "SERIOUS" "2" "2" "2" ...
##  $ X.17                          : chr  "NOINJURY" "0" "0" "0" ...
##  $ X.18                          : chr  "OTHER" "0" "2" "1" ...
##  $ WESTERN.REGION                : chr  "FATAL" "0" "0" "0" ...
##  $ X.19                          : chr  "SERIOUS" "1" "1" "2" ...
##  $ X.20                          : chr  "NOINJURY" "0" "0" "0" ...
##  $ X.21                          : chr  "OTHER" "0" "2" "4" ...
```

```r
# Now, let's examine the first few rows of the data
head(car_accident_data)
```

```
##            X EASTERN.REGION     X.1      X.2   X.3
## 1       DATE          FATAL SERIOUS NOINJURY OTHER
## 2 01/01/2016              0       1        0     5
## 3 02/01/2016              0       3        0     1
## 4 03/01/2016              0       1        0     5
## 5 04/01/2016              0       2        0     2
## 6 05/01/2016              0       1        0     6
##   METROPOLITAN.NORTH.WEST.REGION     X.4      X.5   X.6
## 1                          FATAL SERIOUS NOINJURY OTHER
## 2                              0       7        0     7
## 3                              0       2        0     4
## 4                              0       3        0     6
## 5                              0       2        0     5
## 6                              1       2        0     7
##   METROPOLITAN.SOUTH.EAST.REGION     X.7      X.8   X.9 NORTH.EASTERN.REGION
## 1                          FATAL SERIOUS NOINJURY OTHER                FATAL
## 2                              1       9        0     3                    0
## 3                              0       8        0     5                    1
## 4                              0       7        0     4                    0
## 5                              0       4        0     5                    0
## 6                              0       6        0     7                    0
##      X.10     X.11  X.12 NORTHERN.REGION    X.13     X.14  X.15
## 1 SERIOUS NOINJURY OTHER           FATAL SERIOUS NOINJURY OTHER
## 2       3        0     2               0       1        0     1
## 3       2        0     1               0       2        0     1
## 4       2        0     3               0       0        0     3
## 5       1        0     1               1       4        0     3
## 6       6        0     2               0       0        0     3
##   SOUTH.WESTERN.REGION    X.16     X.17  X.18 WESTERN.REGION    X.19     X.20
## 1                FATAL SERIOUS NOINJURY OTHER          FATAL SERIOUS NOINJURY
## 2                    0       2        0     0              0       1        0
## 3                    0       2        0     2              0       1        0
## 4                    0       2        0     1              0       2        0
## 5                    0       1        0     3              0       0        0
## 6                    0       3        0     1              0       2        0
##    X.21
## 1 OTHER
## 2     0
## 3     2
## 4     4
## 5     1
## 6     1
```

```r
# Function to sum numeric values in a column, skipping the header
sum_numeric <- function(x) sum(as.numeric(x[-1]), na.rm = TRUE)

# Identify FATAL and SERIOUS columns
fatal_cols <- c("EASTERN.REGION", "METROPOLITAN.NORTH.WEST.REGION", "METROPOLITAN.SOUTH.EAST.REGION",
                "NORTH.EASTERN.REGION", "NORTHERN.REGION", "SOUTH.WESTERN.REGION", "WESTERN.REGION")
serious_cols <- c("X.1", "X.4", "X.7", "X.10", "X.13", "X.16", "X.19")
```

```
# Calculate totals
total_fatal <- sum(sapply(car_accident_data[fatal_cols], sum_numeric))
total_serious <- sum(sapply(car_accident_data[serious_cols], sum_numeric))

# Output results
cat("Total FATAL accidents across all regions:", total_fatal, "\n")
```

```
## Total FATAL accidents across all regions: 1404
```

```
cat("Total SERIOUS accidents across all regions:", total_serious, "\n")
```

```
## Total SERIOUS accidents across all regions: 23332
```

```
# Calculate the ratio of SERIOUS to FATAL accidents
ratio <- total_serious / total_fatal
cat("Ratio of SERIOUS to FATAL accidents:", ratio, "\n")
```

```
## Ratio of SERIOUS to FATAL accidents: 16.61823
```

FATAL Represents accidents that resulted in at least one death. The values of FATAL indicate the number of fatal accidents on a given day in the REGION. The total number of fatal accidents across all regions is 1,404.

SERIOUS represents accidents that resulted in serious injuries, but no immediate fatalities.The values of SERIOUS indicate the number of serious accidents on a given day in the REGION. The total number of serious accidents across all regions is 23,332.

the main difference are FATAL involves loss of life, while SERIOUS involves severe injuries. SERIOUS accidents are much more common, occurring about 17 times more often than FATAL accidents (ratio of 16.62:1). While FATAL accidents have the most severe immediate outcome, SERIOUS accidents represent a larger-scale problem in terms of healthcare burden and long-term effects.

Q2: Tidy data (20 points) Q2.1 Cleaning up columns. You may notice that the road traffic accidents csv file has two rows of heading. This is quite common in data generated by BI reporting tools. Let's clean up the column names. Use the code below and print out a list of regions in the data set. (1 point):

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```r
library(stringr)

cav_data_link <- 'car_accidents_victoria.csv'
top_row <- read_csv(cav_data_link, col_names = FALSE, n_max = 1)
```

```
## Rows: 1 Columns: 29

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): X2, X6, X10, X14, X18, X22, X26
## lgl (22): X1, X3, X4, X5, X7, X8, X9, X11, X12, X13, X15, X16, X17, X19, X20...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
second_row <- read_csv(cav_data_link, n_max = 1)
```

```
## New names:
## Rows: 1 Columns: 29
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (29): ...1, EASTERN REGION, ...3, ...4, ...5, METROPOLITAN NORTH WEST RE...
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘‘ -> ‘...1‘
## * ‘‘ -> ‘...3‘
## * ‘‘ -> ‘...4‘
## * ‘‘ -> ‘...5‘
## * ‘‘ -> ‘...7‘
## * ‘‘ -> ‘...8‘
## * ‘‘ -> ‘...9‘
## * ‘‘ -> ‘...11‘
## * ‘‘ -> ‘...12‘
## * ‘‘ -> ‘...13‘
## * ‘‘ -> ‘...15‘
## * ‘‘ -> ‘...16‘
## * ‘‘ -> ‘...17‘
## * ‘‘ -> ‘...19‘
## * ‘‘ -> ‘...20‘
## * ‘‘ -> ‘...21‘
## * ‘‘ -> ‘...23‘
## * ‘‘ -> ‘...24‘
## * ‘‘ -> ‘...25‘
## * ‘‘ -> ‘...27‘
## * ‘‘ -> ‘...28‘
## * ‘‘ -> ‘...29‘
```

```r
column_names <- second_row %>%
unlist(., use.names=FALSE) %>%
make.unique(., sep = "__") # double underscore

column_names[2:5] <- str_c(column_names[2:5], '0', sep='__')
```

```r
daily_accidents <-read_csv(cav_data_link, skip = 2, col_names = column_names)
```

```
## Rows: 1643 Columns: 29
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): DATE
## dbl (28): FATAL__0, SERIOUS__0, NOINJURY__0, OTHER__0, FATAL__1, SERIOUS__1,...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
head(daily_accidents)
```

```
## # A tibble: 6 x 29
##   DATE  FATAL__0 SERIOUS__0 NOINJURY__0 OTHER__0 FATAL__1 SERIOUS__1 NOINJURY__1
##   <chr>    <dbl>      <dbl>       <dbl>    <dbl>    <dbl>      <dbl>       <dbl>
## 1 01/0~        0          1           0        5        0          7           0
## 2 02/0~        0          3           0        1        0          2           0
## 3 03/0~        0          1           0        5        0          3           0
## 4 04/0~        0          2           0        2        0          2           0
## 5 05/0~        0          1           0        6        1          2           0
## 6 06/0~        0          2           0        1        0          5           0
## # i 21 more variables: OTHER__1 <dbl>, FATAL__2 <dbl>, SERIOUS__2 <dbl>,
## #   NOINJURY__2 <dbl>, OTHER__2 <dbl>, FATAL__3 <dbl>, SERIOUS__3 <dbl>,
## #   NOINJURY__3 <dbl>, OTHER__3 <dbl>, FATAL__4 <dbl>, SERIOUS__4 <dbl>,
## #   NOINJURY__4 <dbl>, OTHER__4 <dbl>, FATAL__5 <dbl>, SERIOUS__5 <dbl>,
## #   NOINJURY__5 <dbl>, OTHER__5 <dbl>, FATAL__6 <dbl>, SERIOUS__6 <dbl>,
## #   NOINJURY__6 <dbl>, OTHER__6 <dbl>
```

Q2.2 Tidying data a) Now we have a data frame. Answer the following questions for this data frame. • Does each variable have its own column? (1 point) no, the accident types (FATAL, SERIOUS, NOINJURY, OTHER) are repeated for each region with suffixes.

- Does each observation have its own row? (1 point) yes, each observation which representing a day has its own row.

- Does each value have its own cell? (1 point) yes

b) Use spreading and/or gathering (or their pivot_wider and pivot_longer new equivalents) to transform the data frame into tidy data. The key is to put data from the same measurement source in a column and to put each observation in a row. Then, answer the following questions. I. How many spreading (or pivot_wider) operations do you need? (1 point) no spreading are needed. The dataset involves multiple similar variables spread across several columns hence these columns need to gather into fewer variables.

II. How many gathering (or pivot_longer) operations do you need? (1 point) 1 pivot_longer operation is enough to transform columns like FATAL___0, SERIOUS___0, etc., into key-value pairs, where the key is the accident type and region and the value is the number of accidents.

III. Explain the steps in detail. (5 points)

I start by identifying that the accident types (FATAL, SERIOUS, NOINJURY, and OTHER) repeat across columns with suffixes like **n, indicating different regions. Using pivot_longer, it select all columns starting with "FATAL", "SERIOUS", "NOINJURY", and "OTHER", except for the DATE column. These columns are gathered into three new columns: REGION, ACCIDENT_TYPE, and COUNT. The accident types and region numbers are separated by double underscores ()**. This transformation reshapes the data into a long format, where each row represents an accident type, region, and the corresponding count for a specific date. The final result includes columns like DATE, accident_type, region, and count, where accident_type represents FATAL, SERIOUS, etc., and region corresponds to the numbered regions.

```r
# Load necessary libraries
library(tidyr)
library(dplyr)

# Tidy the data using pivot_longer
tidy_accidents <- daily_accidents %>%
  pivot_longer(
    cols = starts_with("FATAL") | starts_with("SERIOUS") | starts_with("NOINJURY") | starts_with("OTHER
    names_to = c("accident_type", "region"),
    names_sep = "__",  # Separate accident types and region numbers by the double underscores
    values_to = "count"
  )

head(tidy_accidents)
```

```
## # A tibble: 6 x 4
##   DATE       accident_type region count
##   <chr>      <chr>         <chr>  <dbl>
## 1 01/01/2016 FATAL         0          0
## 2 01/01/2016 FATAL         1          0
## 3 01/01/2016 FATAL         2          1
## 4 01/01/2016 FATAL         3          0
## 5 01/01/2016 FATAL         4          0
## 6 01/01/2016 FATAL         5          0
```

IV. Provide/print the head of the dataset. (4 points).

```r
# Print the head of the tidy dataset
head(tidy_accidents)
```

```
## # A tibble: 6 x 4
##   DATE       accident_type region count
##   <chr>      <chr>         <chr>  <dbl>
## 1 01/01/2016 FATAL         0          0
## 2 01/01/2016 FATAL         1          0
## 3 01/01/2016 FATAL         2          1
## 4 01/01/2016 FATAL         3          0
## 5 01/01/2016 FATAL         4          0
## 6 01/01/2016 FATAL         5          0
```

c) Are the variables having the expected variable types in R? Clean up the data types and print the head of the dataset. (3 points) i change the data type for each variable . the data type should be correct now.

```
str(tidy_accidents)
```

```
## tibble [46,004 x 4] (S3: tbl_df/tbl/data.frame)
##  $ DATE         : chr [1:46004] "01/01/2016" "01/01/2016" "01/01/2016" "01/01/2016" ...
##  $ accident_type: chr [1:46004] "FATAL" "FATAL" "FATAL" "FATAL" ...
##  $ region       : chr [1:46004] "0" "1" "2" "3" ...
##  $ count        : num [1:46004] 0 0 1 0 0 0 0 1 7 9 ...
```

```
# Clean and rename region codes in the tidy_accidents dataset, ensuring region is a factor
tidy_accidents <- tidy_accidents %>%
  mutate(
    DATE = as.Date(DATE, format = "%d/%m/%Y"),   # Convert DATE to Date type
    accident_type = as.factor(accident_type),    # Convert accident_type to factor
    count = as.integer(count),                   # Ensure count is numeric
    region = as.factor(case_when(                # Rename region codes and ensure it's a factor
      region == 0 ~ "EASTERN.REGION",
      region == 1 ~ "METROPOLITAN.NORTH.WEST.REGION",
      region == 2 ~ "METROPOLITAN.SOUTH.EAST.REGION",
      region == 3 ~ "NORTH.EASTERN.REGION",
      region == 4 ~ "NORTHERN.REGION",
      region == 5 ~ "SOUTH.WESTERN.REGION",
      region == 6 ~ "WESTERN.REGION",
      TRUE ~ as.character(region)  # Handle any unmatched region values
    ))
  )

# Print the first few rows of the cleaned dataset
head(tidy_accidents)
```

```
## # A tibble: 6 x 4
##   DATE       accident_type region                          count
##   <date>     <fct>         <fct>                           <int>
## 1 2016-01-01 FATAL         EASTERN.REGION                      0
## 2 2016-01-01 FATAL         METROPOLITAN.NORTH.WEST.REGION      0
## 3 2016-01-01 FATAL         METROPOLITAN.SOUTH.EAST.REGION      1
## 4 2016-01-01 FATAL         NORTH.EASTERN.REGION                0
## 5 2016-01-01 FATAL         NORTHERN.REGION                     0
## 6 2016-01-01 FATAL         SOUTH.WESTERN.REGION                0
```

```
str(tidy_accidents)
```

```
## tibble [46,004 x 4] (S3: tbl_df/tbl/data.frame)
##  $ DATE         : Date[1:46004], format: "2016-01-01" "2016-01-01" ...
##  $ accident_type: Factor w/ 4 levels "FATAL","NOINJURY",..: 1 1 1 1 1 1 1 4 4 4 ...
##  $ region       : Factor w/ 7 levels "EASTERN.REGION",..: 1 2 3 4 5 6 7 1 2 3 ...
##  $ count        : int [1:46004] 0 0 1 0 0 0 0 1 7 9 ...
```

d) Are there any missing values? Fix the missing data. Justify your actions. (2 points)

```r
summary(tidy_accidents)
```

```
##       DATE              accident_type                                region
##  Min.   :2016-01-01   FATAL   :11501   EASTERN.REGION                   :6572
##  1st Qu.:2017-02-14   NOINJURY:11501   METROPOLITAN.NORTH.WEST.REGION:6572
##  Median :2018-04-01   OTHER   :11501   METROPOLITAN.SOUTH.EAST.REGION:6572
##  Mean   :2018-04-01   SERIOUS :11501   NORTH.EASTERN.REGION             :6572
##  3rd Qu.:2019-05-17                    NORTHERN.REGION                  :6572
##  Max.   :2020-06-30                    SOUTH.WESTERN.REGION             :6572
##                                        WESTERN.REGION                   :6572
##      count
##  Min.   : 0.00
##  1st Qu.: 0.00
##  Median : 0.00
##  Mean   : 1.44
##  3rd Qu.: 1.00
##  Max.   :33.00
##  NA's   :4
```

```r
# Replace missing values in the 'count' column with 0
tidy_accidents_cleaned <- tidy_accidents %>%
  mutate(count = ifelse(is.na(count), 0, count),
         count = as.integer(count))

# Print the summary of the cleaned dataset
summary(tidy_accidents_cleaned)
```

```
##       DATE              accident_type                                region
##  Min.   :2016-01-01   FATAL   :11501   EASTERN.REGION                   :6572
##  1st Qu.:2017-02-14   NOINJURY:11501   METROPOLITAN.NORTH.WEST.REGION:6572
##  Median :2018-04-01   OTHER   :11501   METROPOLITAN.SOUTH.EAST.REGION:6572
##  Mean   :2018-04-01   SERIOUS :11501   NORTH.EASTERN.REGION             :6572
##  3rd Qu.:2019-05-17                    NORTHERN.REGION                  :6572
##  Max.   :2020-06-30                    SOUTH.WESTERN.REGION             :6572
##                                        WESTERN.REGION                   :6572
##      count
##  Min.   : 0.00
##  1st Qu.: 0.00
##  Median : 0.00
##  Mean   : 1.44
##  3rd Qu.: 1.00
##  Max.   :33.00
##
```

```r
head(tidy_accidents_cleaned)
```

```
## # A tibble: 6 x 4
##   DATE       accident_type region                         count
##   <date>     <fct>         <fct>                          <int>
## 1 2016-01-01 FATAL         EASTERN.REGION                     0
## 2 2016-01-01 FATAL         METROPOLITAN.NORTH.WEST.REGION     0
## 3 2016-01-01 FATAL         METROPOLITAN.SOUTH.EAST.REGION     1
## 4 2016-01-01 FATAL         NORTH.EASTERN.REGION               0
## 5 2016-01-01 FATAL         NORTHERN.REGION                    0
## 6 2016-01-01 FATAL         SOUTH.WESTERN.REGION               0
```

based on the summary, The count column has 4 missing values that need to be address. Since the count column records the number of accidents, a missing value might imply that no accidents occurred, which can be represented as 0.

Q3: Fitting distributions (20 points) In this question, we will fit a couple of distributions to the "TOTAL_ACCIDENTS" data.

```
# Group by DATE and sum the 'count' to get TOTAL_ACCIDENTS for each date
total_accidents_summary <- tidy_accidents_cleaned %>%
  group_by(DATE) %>%
  summarize(TOTAL_ACCIDENTS = sum(count, na.rm = TRUE))

# Print the result to check the first few rows
head(total_accidents_summary)
```

```
## # A tibble: 6 x 2
##   DATE       TOTAL_ACCIDENTS
##   <date>               <int>
## 1 2016-01-01              43
## 2 2016-01-02              37
## 3 2016-01-03              43
## 4 2016-01-04              35
## 5 2016-01-05              48
## 6 2016-01-06              50
```

```
# Merge the total accidents back into the original dataset by DATE
tidy_accidents_cleaned <- tidy_accidents_cleaned %>%
  left_join(total_accidents_summary, by = "DATE")

head(tidy_accidents_cleaned)
```

```
## # A tibble: 6 x 5
##   DATE       accident_type region                          count TOTAL_ACCIDENTS
##   <date>     <fct>         <fct>                           <int>           <int>
## 1 2016-01-01 FATAL         EASTERN.REGION                      0              43
## 2 2016-01-01 FATAL         METROPOLITAN.NORTH.WEST.REGION      0              43
## 3 2016-01-01 FATAL         METROPOLITAN.SOUTH.EAST.REGION      1              43
## 4 2016-01-01 FATAL         NORTH.EASTERN.REGION                0              43
## 5 2016-01-01 FATAL         NORTHERN.REGION                     0              43
## 6 2016-01-01 FATAL         SOUTH.WESTERN.REGION                0              43
```

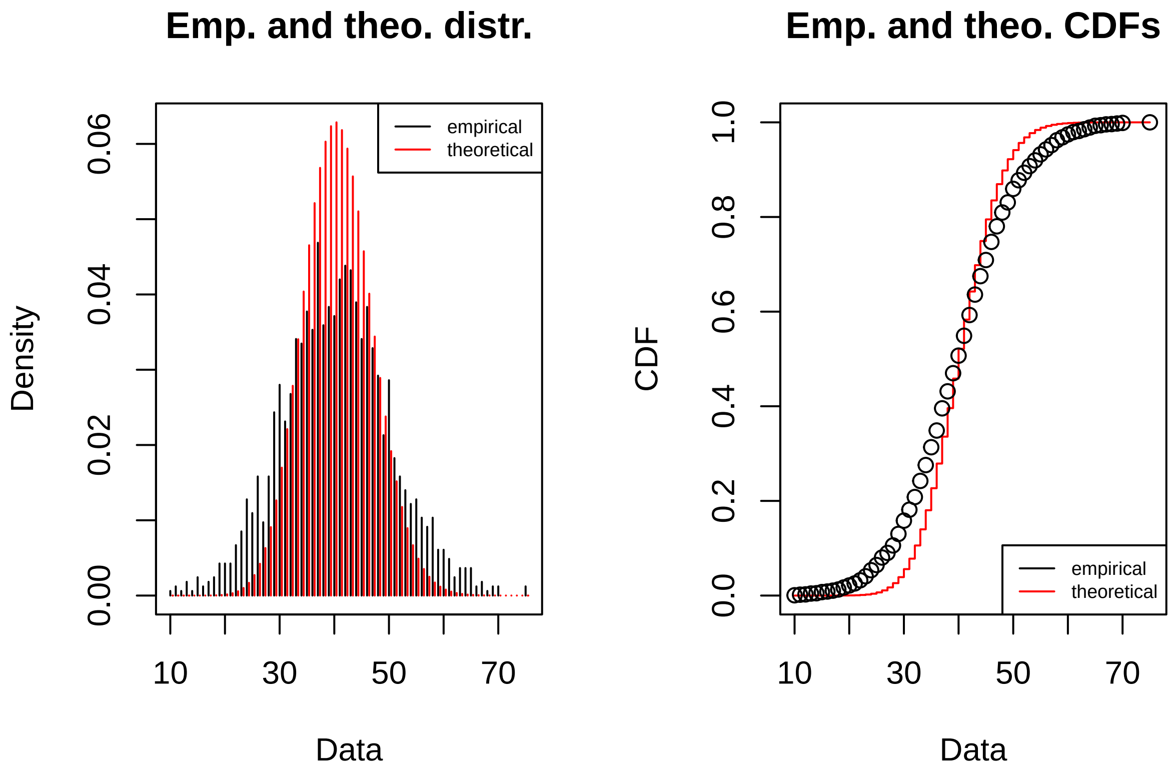
Q3.1: Fit a Poisson distribution and a negative binomial distribution on TOTAL_ACCIDENTS. You may use functions provided by the package fitdistrplus. (4 points)

```
library(fitdistrplus)
```

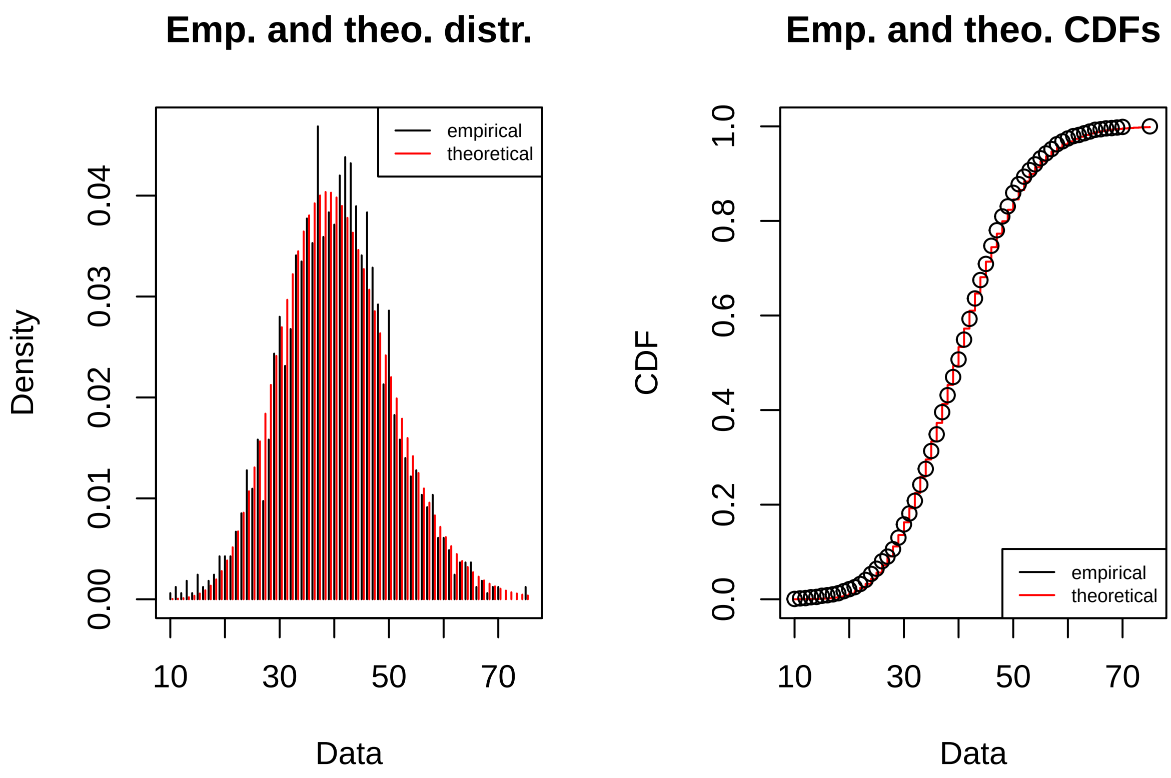
```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'

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

## Loading required package: survival
```

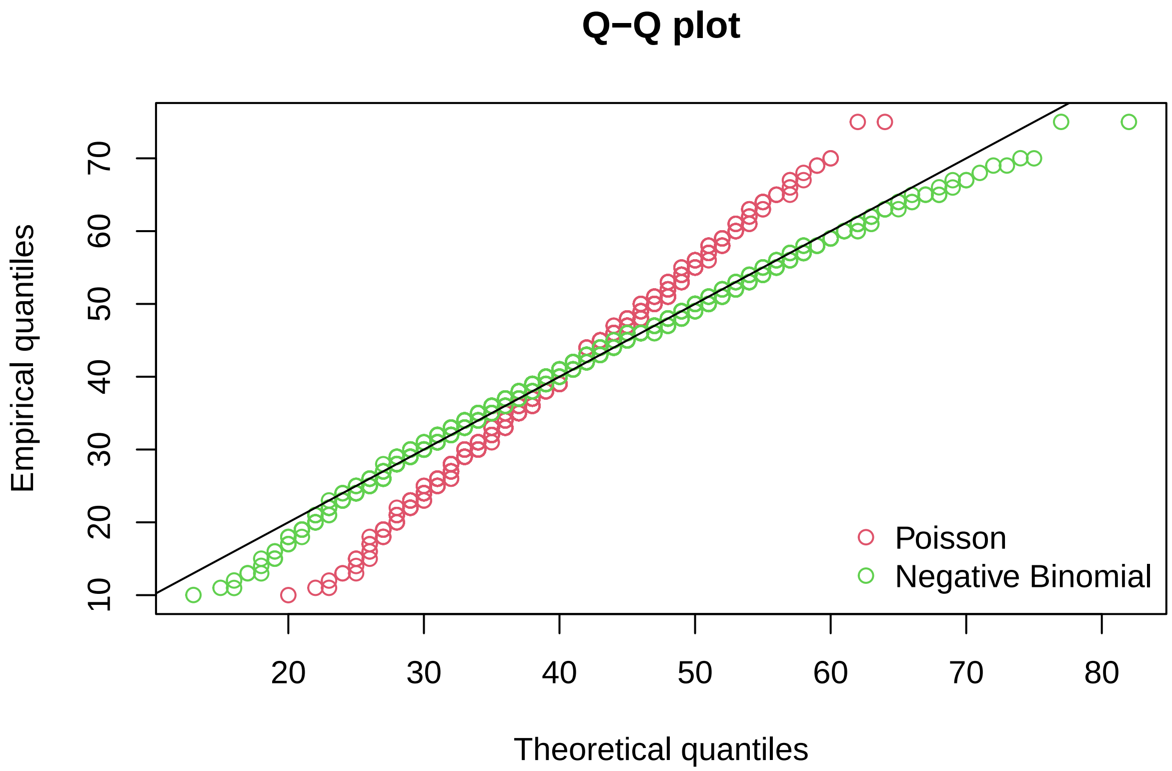
```
total_accidents_data <- total_accidents_summary$TOTAL_ACCIDENTS

# Fit a Poisson distribution to the TOTAL_ACCIDENTS data
poisson_fit <- fitdist(data = total_accidents_data, distr = "pois")

# Fit a Negative Binomial distribution to the TOTAL_ACCIDENTS data
nbinom_fit <- fitdist(total_accidents_data, "nbinom")

poisson_fit %>% plot
```



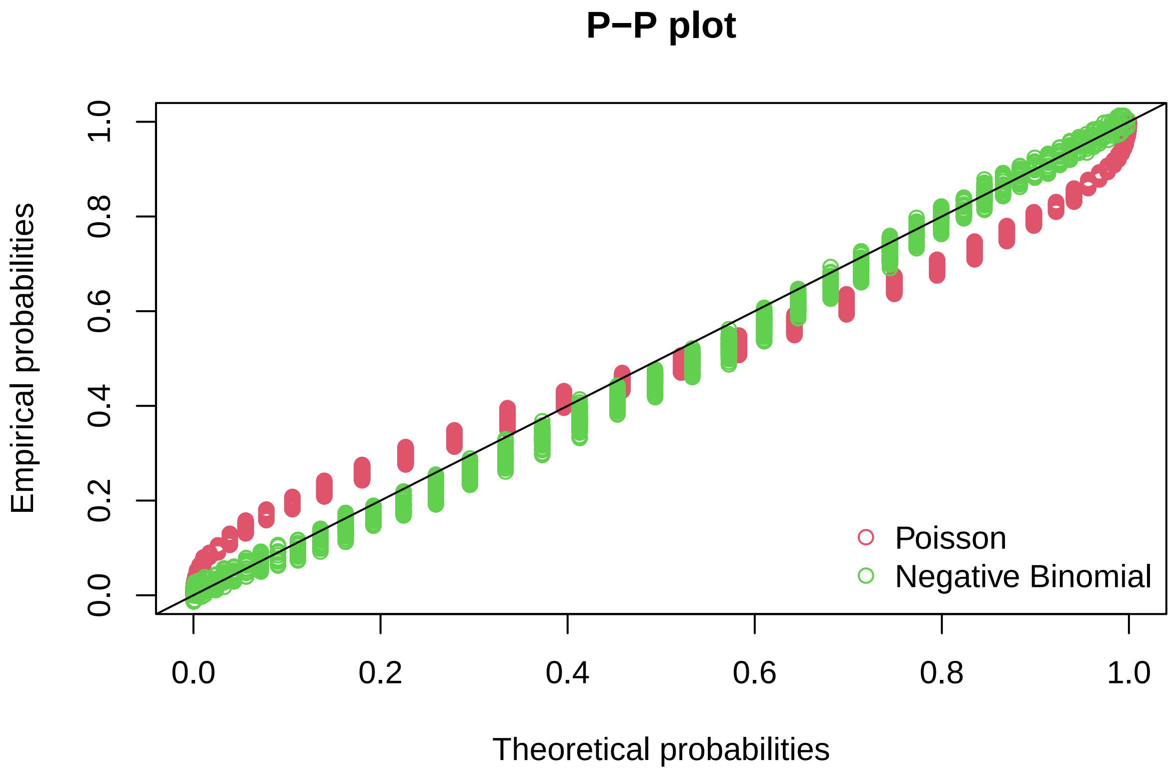
```
nbinom_fit %>% plot
```



```r
# Generate Q-Q plots to compare the empirical quantiles with the theoretical quantiles
qqcomp(list(poisson_fit, nbinom_fit), legendtext = c("Poisson", "Negative Binomial"))
```



```
# Generate P-P plots to compare the empirical and theoretical CDFs
ppcomp(list(poisson_fit, nbinom_fit), legendtext = c("Poisson", "Negative Binomial"))
```



```
# Summarize the results
summary(poisson_fit)
```

```
## Fitting of the distribution ' pois ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## lambda 40.32806  0.1566697
## Loglikelihood:  -6565.159   AIC:  13132.32   BIC:  13137.72
```

```
summary(nbinom_fit)
```

```
## Fitting of the distribution ' nbinom ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## size 26.89578  1.5909545
## mu   40.32640  0.2476714
## Loglikelihood:  -6110.248   AIC:  12224.5   BIC:  12235.3
## Correlation matrix:
##              size           mu
## size 1.0000000000 0.0002372421
## mu   0.0002372421 1.0000000000
```

```
mean(total_accidents_data)
```

```
## [1] 40.32806
```

```r
var(total_accidents_data)
```

```
## [1] 97.74676
```

Log-likelihood: The Negative Binomial distribution has a higher log-likelihood, indicating that it fits the data better.

AIC, BIC The Negative Binomial distribution has a lower AIC & BIC, A lower AIC & BIC indicates a better fit.

CDF: The Negative Binomial distribution appears to fit the empirical distribution better, as the theoretical curve (in red) is more closely aligned with the empirical histogram (black lines) than the Poisson distribution.

The theoretical density: The Negative Binomial distribution aligns more closely with the peaks and the spread of the empirical data compared to the Poisson distribution as Poisson distribution underestimates the spread and peak of the data.The is because Negative Binomial distribution accounts for the wider spread and variability in the data, which explains why it fits better than the Poisson.

PPplot: Negative Binomial distribution (green points) follows the diagonal line more closely than the Poisson (red points), indicating that it provides a better overall fit to the data, particularly at both lower and upper tails.

QQplot: Negative Binomial (green) points follow the line more closely than the Poisson (red) points, especially at higher and lower quantiles where the Poisson distribution diverges from the empirical data., indicates that the Negative Binomial fits the data better.

Poisson distribution assumes the mean and variance are equal (both lambda). In this case, that assumption is violated because the actual variance (97.74676) is much larger than the mean (40.32806). Negative Binomial distribution allows for overdispersion, which makes it potentially more suitable for this data. The size parameter (26.89578) accounts for this extra variability.The standard errors for the Negative Binomial estimates are larger than for the Poisson, reflecting the additional complexity of estimating two parameters instead of one. The larger standard errors in the Negative Binomial fit reflect the model's increased complexity and flexibility due to the true nature of overdispersed data.

In conclusion, while both distributions capture the central tendency of the data well, the presence of overdispersion suggests that the Negative Binomial distribution might be a more appropriate model for this data, as it can account for the extra variability that the Poisson distribution cannot.

Q3.2: Compare the log-likelihood of two fitted distributions. Which distribution fits the data better? Why? (6 points)

```r
# Extract log-likelihood values
log_likelihood_poisson <- poisson_fit$loglik
log_likelihood_nbinom <- nbinom_fit$loglik

# Print log-likelihood values for comparison
log_likelihood_poisson
```

```
## [1] -6565.159
```

```r
log_likelihood_nbinom
```

```
## [1] -6110.248
```

The Negative Binomial distribution has a higher log-likelihood (-6110.248 compared to -6565.159 for Poisson), indicating that it fits the data better.

Poisson assumes mean = variance, where events occur independently and at a constant rate. however, in real world scenario like accident event,this assumption is not valid. Accidents are influence by multiple factors such as weather, time of the day, road condition which lead to fluctuations in the frequency of events. This variability cause the variance to be much larger than the mean, which is known as overdispersion.

Negative Binomial distribution, on the other hand are more flexible as it allows overdispersion since variance can be larger than mean. As a result, Negative Binomial distribution makes it better suited for data like accident counts, where the number of accidents per day or per region can vary widely. The Negative Binomial distribution account for this variability, providing a more accurate fit when accident data exhibit significant fluctuations.

Q3.3 (Research Question): Try one more distribution. Try to fit all 3 distributions to two different accident types. Combine your results in the table below, analyse and explain the results with a short report (around 200 words).

```r
# Fit distributions for FATAL accidents
fatal_data <- tidy_accidents_cleaned %>% filter(accident_type == "FATAL") %>% pull(count)
mean(fatal_data)
```

```
## [1] 0.1220763
```

```r
var(fatal_data)
```

```
## [1] 0.12527
```

```r
# Fit Poisson, Negative Binomial, and Geometric distributions for FATAL accidents
poisson_fatal <- fitdist(fatal_data, "pois")
nbinom_fatal <- fitdist(fatal_data, "nbinom")
geom_fatal <- fitdist(fatal_data, "geom")

# Fit distributions for SERIOUS accidents
serious_data <- tidy_accidents_cleaned %>% filter(accident_type == "SERIOUS") %>% pull(count)
mean(serious_data)
```

```
## [1] 2.028693
```

```r
var(serious_data)
```

```
## [1] 6.286481
```

```r
# Fit Poisson, Negative Binomial, and Geometric distributions for SERIOUS accidents
poisson_serious <- fitdist(serious_data, "pois")
nbinom_serious <- fitdist(serious_data, "nbinom")
geom_serious <- fitdist(serious_data, "geom")

# Extract log-likelihood values for FATAL accidents
loglik_fatal <- data.frame(
  Distribution = c("Poisson", "Negative Binomial", "Geometric"),
  LogLikelihood = c(poisson_fatal$loglik, nbinom_fatal$loglik, geom_fatal$loglik)
)

# Extract log-likelihood values for SERIOUS accidents
loglik_serious <- data.frame(
  Distribution = c("Poisson", "Negative Binomial", "Geometric"),
  LogLikelihood = c(poisson_serious$loglik, nbinom_serious$loglik, geom_serious$loglik)
)

# Combine the results for FATAL and SERIOUS accidents
loglik_combined <- list(FATAL = loglik_fatal, SERIOUS = loglik_serious)
loglik_combined
```

```
## $FATAL
##        Distribution LogLikelihood
## 1           Poisson     -4428.277
## 2 Negative Binomial     -4426.347
## 3         Geometric     -4439.173
##
## $SERIOUS
##        Distribution LogLikelihood
## 1           Poisson     -26708.20
## 2 Negative Binomial     -22092.47
## 3         Geometric     -22094.67
```

In fitting the Poisson, Negative Binomial, and Geometric distributions to the FATAL and SERIOUS accident types, the Negative Binomial distribution fits the data best, as it has the highest log-likelihood values (closest to 0) across both accident types. This result is expected because accident counts often exhibit overdispersion, where the variance exceeds the mean, which can be explained by the negative binomial model, is shown above.

The Poisson distribution assume equal mean and variance, was less suitable for these data, especially in SERIOUS dataset where the variance is much higher than the mean compared to the FATAL dataset. The geometric distribution captured low frequency accident and perform slightly better slightly better than the Poisson in fitting SERIOUS accidents, was outperform by the negative binomial distribution that captured broader range of accidents count. This suggest that although geometric distribution can handle count data, it struggles with datasets that have high variability and larger accident counts.

In short, the negative binomial distribution is the best model for both accident type due to its ability to handle overdispersion and variability in the data, where Poisson distribution underestimate the spread and Geometric distribution is more appropriate for datasets with large numbers of low counts.

Q4: Source weather data (10 points) Above you have processed data for the road accidents of different types in a given region of Victoria. We still need to find local weather data from the same period. You are encouraged to find weather data online. Besides the NOAA data, you may also use data from the Bureau of Meteorology historical weather observations and statistics. (The NOAA Climate Data might be easier to process, also a full list of weather stations is provided here: https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt) Answer the following questions: Q4.1: Which data source do you plan to use? Justify your decision. (4 points)

The data source chosen is Melbourne Airport (ASN00086282) from NOAA.The data was download directly from the NOAA website (BEFORE NOAA WAS DOWN) and NOAA was selected due to its accessible and well-structured climate data, which makes it easier to process for analysis. Specifically, the Melbourne Airport weather station was chosen, There are several reasons for this choice. Firstly, airport weather stations typically maintain more complete and consistent data records due to their critical role in aviation, hence the dataset is less likely to have missing data points. Additionally, Melbourne Airport is strategically located to offer a balance between urban and suburban areas. It is close enough to the city to reflect metropolitan conditions, while also capturing weather patterns that affect the surrounding regions. This makes it highly relevant to multiple regions in the car accident dataset, including the Eastern Region, Metropolitan North-West Region, Northern Region, and Western Region.

Q4.2: From the data source identified, download daily temperature and precipitation data for the region during the relevant time period. (Hint: If you download data from NOAA https://www.ncdc.noaa.gov/cdo-web/, you need to request an NOAA web service token for accessing the data.) (2 points)

```
melbourne_airport_data <- read.csv("melbourne_airport_data.csv")
head(melbourne_airport_data)
```

```
##       STATION                  NAME       DATE PRCP TAVG TMAX TMIN
## 1 ASN00086282 MELBOURNE AIRPORT, AS 01/01/2016  0.0   69   84   68
## 2 ASN00086282 MELBOURNE AIRPORT, AS 02/01/2016  0.0   69   81   60
## 3 ASN00086282 MELBOURNE AIRPORT, AS 03/01/2016  0.0   68   83   61
## 4 ASN00086282 MELBOURNE AIRPORT, AS 04/01/2016  0.0   66   76   58
## 5 ASN00086282 MELBOURNE AIRPORT, AS 05/01/2016  0.2   69   78   57
## 6 ASN00086282 MELBOURNE AIRPORT, AS 06/01/2016  0.0   68   78   60
```

Q4.3: Answer the following questions (Provide the output from your R Studio): • How many rows are in your local weather data? (2 points) • What time period does the data cover? (2 points)

```
# Get the number of rows in the dataset
nrow(melbourne_airport_data)
```

```
## [1] 1643
```

```
# Convert the 'DATE' column to Date format
melbourne_airport_data$DATE <- as.Date(melbourne_airport_data$DATE, format="%d/%m/%Y")

# Get the start and end date (time period covered by the data)
start_date <- min(melbourne_airport_data$DATE, na.rm = TRUE)
end_date <- max(melbourne_airport_data$DATE, na.rm = TRUE)

start_date
```

```
## [1] "2016-01-01"
```

```
end_date
```

```
## [1] "2020-06-30"
```

```
head(melbourne_airport_data)
```

```
##       STATION                  NAME       DATE PRCP TAVG TMAX TMIN
## 1 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-01  0.0   69   84   68
## 2 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-02  0.0   69   81   60
## 3 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-03  0.0   68   83   61
## 4 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-04  0.0   66   76   58
## 5 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-05  0.2   69   78   57
## 6 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-06  0.0   68   78   60
```

```
summary(melbourne_airport_data)
```

```
##    STATION              NAME                 DATE                 PRCP
##  Length:1643        Length:1643        Min.   :2016-01-01   Min.   : 0.00
##  Class :character   Class :character   1st Qu.:2017-02-14   1st Qu.: 0.00
##  Mode  :character   Mode  :character   Median :2018-04-01   Median : 0.00
##                                        Mean   :2018-04-01   Mean   : 1.52
##                                        3rd Qu.:2019-05-16   3rd Qu.: 0.60
##                                        Max.   :2020-06-30   Max.   :69.00
##       TAVG             TMAX             TMIN
##  Min.   :40.00   Min.   : 47.00   Min.   :28.00
##  1st Qu.:51.00   1st Qu.: 60.00   1st Qu.:44.00
##  Median :57.00   Median : 67.00   Median :49.00
##  Mean   :58.95   Mean   : 69.64   Mean   :50.02
##  3rd Qu.:65.00   3rd Qu.: 77.00   3rd Qu.:56.00
##  Max.   :94.00   Max.   :115.00   Max.   :80.00
```

There are 1643 row and the time period cover from 1/1/2016 to 30/6/2020 which aligns with the time period covered by the car accident dataset.

Q5 Heatwaves, precipitation and road traffic accidents (10 points) The connection between weather and the road traffic accidents is widely reported. In this task, you will try to measure the heatwave and assess its impact on the road accident statistics. Accordingly, you will be using the car_accidents_victoria dataset together with the local weather data. Q5.1. John Nairn and Robert Fawcett from the Australian Bureau of Meteorology have proposed a measure for the heatwave, called the excess heat factor (EHF). Read the following article and summarise your understanding in terms of the definition of the EHF. https://dx.doi.org/10.3390%2Fijerph120100227 (4 points)

The Excess Heat Factor (EHF) is a measure of the intensity of a heatwave, only happens when temperature are unusually high compared. EHF is used to provide important information about heatwaves to the public. It helps the Bureau of Meteorology predict and issue warnings for severe and extreme heatwaves, allowing communities to assess their vulnerability and take necessary precautions.

The EHF combines two main components: 1. Significance index (EHIsig), compares a three-day average daily mean temperature to the 95th percentile temperature for that location, measures how extreme the current temperatures are compared to historical norms. 2. Acclimatisation index (EHIaccl), compares the same three-day average to the mean temperature of the previous 30 days, evaluates how much hotter the recent period has been compared to the previous 30 days, capturing the community's lack of adjustment to sudden heat increases.

The EHF is calculated by multiplying the EHIsig and the maximum of 1 or EHIaccl. This formula ensure that EHIaccl<= 1 if recent temperature is consistently high, as a result the impact of current temperature is not reduce, and EHIaccl > 1 if recent temperature is consistently low, as a result the effect of EHIaccl is greater than EHIsig. This method accounts for both absolute temperature extremes and recent temperature history, making it location-specific and adaptable to different climate zones. A heatwave is defined when the EHF is positive, making it a critical tool for public health and safety in Australia.

Q5.2: Use the NOAA data to calculate the daily EHF values for the area you chose during the relevant time period. Plot the daily EHF values. (6 points)

```
library(dplyr)
library(ggplot2)
library(lubridate)
```

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

```
library(zoo)  # for rolling mean calculation
```

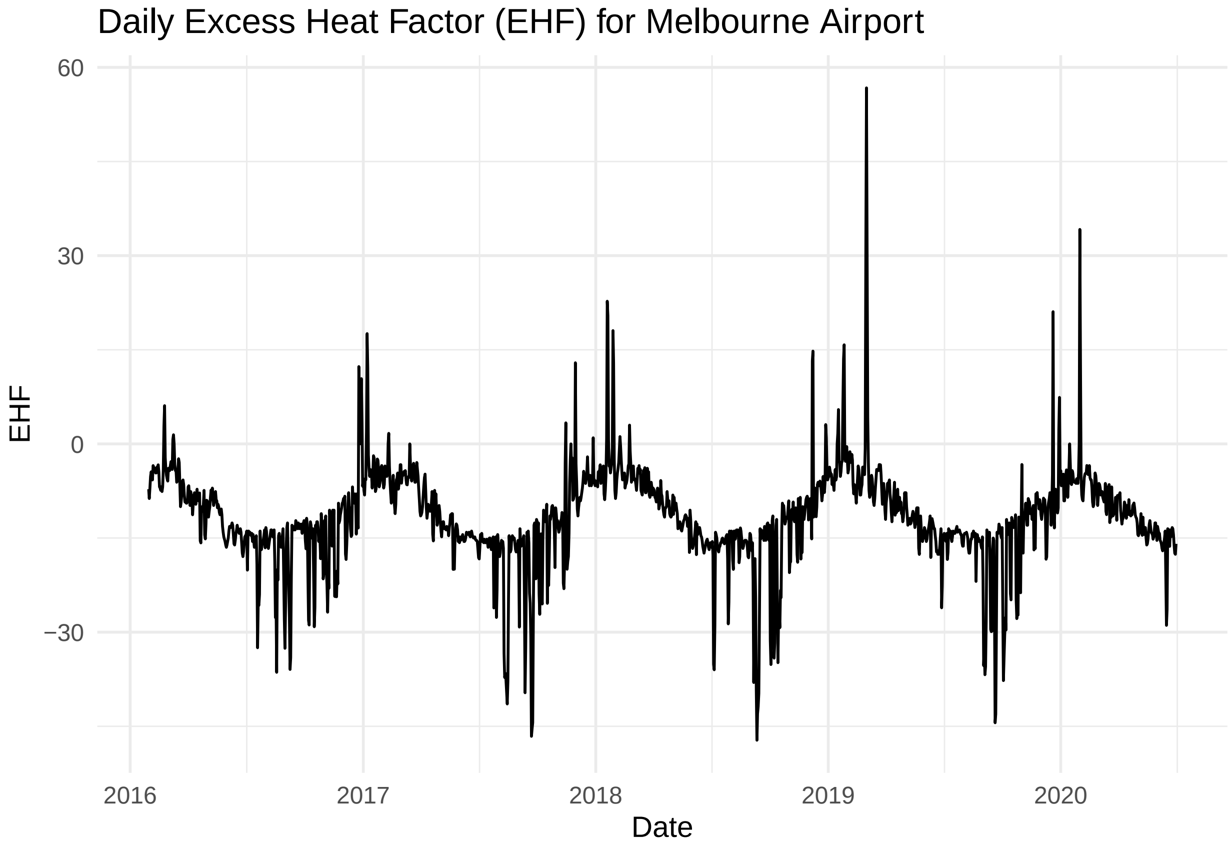
```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# Convert DATE to proper Date format and convert temperatures to Celsius
noaa_data <- melbourne_airport_data %>%
  mutate(
    TAVG = (as.numeric(TAVG) - 32) * 5/9,  # Convert Fahrenheit to Celsius
    TMAX = (as.numeric(TMAX) - 32) * 5/9,
    TMIN = (as.numeric(TMIN) - 32) * 5/9
  ) %>%
  arrange(DATE)

# Calculate 3-day and 30-day moving averages
noaa_data <- noaa_data %>%
  mutate(
    TAVG_3day = rollmean(TAVG, k = 3, fill = NA, align = "right"),
    TAVG_30day = rollmean(TAVG, k = 30, fill = NA, align = "right")
  )

# Calculate 95th percentile of long-term temperatures
T95 <- quantile(noaa_data$TAVG, 0.95, na.rm = TRUE)

# Calculate EHIsig, EHIaccl, and EHF
noaa_data <- noaa_data %>%
  mutate(
    EHIsig = TAVG_3day - T95,
    EHIaccl = TAVG_3day - TAVG_30day,
    EHF = EHIsig * pmax(1, EHIaccl)
  )

head(noaa_data)
```

```
##       STATION                  NAME       DATE PRCP     TAVG     TMAX     TMIN
## 1 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-01  0.0 20.55556 28.88889 20.00000
## 2 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-02  0.0 20.55556 27.22222 15.55556
## 3 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-03  0.0 20.00000 28.33333 16.11111
## 4 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-04  0.0 18.88889 24.44444 14.44444
## 5 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-05  0.2 20.55556 25.55556 13.88889
## 6 ASN00086282 MELBOURNE AIRPORT, AS 2016-01-06  0.0 20.00000 25.55556 15.55556
##   TAVG_3day TAVG_30day    EHIsig EHIaccl EHF
## 1        NA         NA        NA      NA  NA
## 2        NA         NA        NA      NA  NA
## 3  20.37037          NA -4.074074      NA  NA
## 4  19.81481          NA -4.629630      NA  NA
## 5  19.81481          NA -4.629630      NA  NA
## 6  19.81481          NA -4.629630      NA  NA
```

Due to NOAA server unavailability, I couldn't retrieve temperature data for December 2015, which is necessary for calculating the Excess Heat Factor (EHF) for early January 2016. As a result, EHF values for January 1-29, 2016 are unavailable (NA). I HAVE chosen to filter out these NA values, focusing our analysis on January 30, 2016 to June 30, 2020. This approach ensures data integrity, methodological consistency, and reliability of results by using only complete and accurate EHF calculations. While we lose some early January data, this method provides a more robust basis for analyzing the relationship between heat waves and road accidents in Victoria.

```
# Filter out NA values from EHF
noaa_data_filtered <- noaa_data %>% filter(!is.na(EHF))

# Plot daily EHF values
ggplot(noaa_data_filtered %>% filter(!is.na(EHF)), aes(x = DATE, y = EHF)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Daily Excess Heat Factor (EHF) for Melbourne Airport",
       x = "Date",
       y = "EHF") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")
```



```r
# Print summary statistics of EHF
summary(noaa_data_filtered$EHF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -47.222 -15.000 -11.852 -11.691  -7.222  56.735
```

The plot shows the Daily Excess Heat Factor (EHF) for Melbourne Airport from 30/1/2016 to 30/6/2020. Positive peaks indicate heatwave, with the most extreme peaks indicating serious heatwave, particularly in the summer. Negative peaks indicate cooler period with no heatwave activity. Based on the plot, 2019 has the most serious severe heatwaves, as reflected by the highest EHF peaks. This visualization highlights periods of extreme heat, may be further analyzed for potential correlations with road traffic accidents in the next few question. ???

Q6: Model planning (10 points) Careful planning is essential for a successful modelling effort. Please answer the following planning questions. Q6.1. Model planning: a) What is the main goal of your model, how it will be used? (1 point) The main goal of this model is to predict the number of road accidents during heatwave events. This model will be useful for forecasting potential spikes in accident rates during periods of extreme heat, enabling proactive planning and response by emergency services.

b) How it will be relevant to the emergency services demand? (1 point) The model will be relevant to the emergency services demand by providing advance warning of potential increases in road accidents during heatwaves, allowing for better allocation of resources and preparation such as increasing staff, ambulance and emergency personnel during high-risk periods.
c) Who are the potential users of your model? (1 point) Potential users are emergency service planners such as hospitals, police and firefighter, traffic authorities, and transportation department responsible for implement precautionary measures, weather reporter, government safety departments responsible for public safety and insurance company assess risk during extreme weather conditions.

Q6.2. Relationship and data: a) What relationship do you plan to model or what do you want to predict? (1 point) The model will predict the relationship between EHF and daily road accident rate across different region of Victoria. The goal is to predict the likelihood of number of accident happen during extreme heat event

b) What is the response variable? (1 point) Daily count of road accidents
c) What are the predictor variables? (1 point) The most important predictor variable of this model is EHF. Additional predictors could include temperature and precipitation to further explain the variation in accident rates.
d) Will the variables in your model be routinely collected and made available soon enough for prediction? (1 point) Yes, weather data such as temperature and precipitation are routinely collected through the weather stations and the EHF can be calculated from the data and made prediction. Additionally, daily road accident counts can be collected from traffic authorities, ensuring all data is available for timely predictions.
e) As you are likely to build your model on historical data, will the data in the future have similar characteristics? (1 point) Based on the daily EHF plot for Melbourne airport, the data in the future should have similar characteristics due to fixed seasonal patterns. However, long term climate change might affect temperature patterns which could affect the accuracy of prediction over an extended period. Thus, it is important to consider these potential changes when making long-term forecasts.

Q6.3. What statistical method(s) will be applied to generate the model? Why? (2 points) Generalized Linear Model (GLM) with a Negative Binomial distribution: This method is suitable for discrete, non-negative count data, such as daily accident counts. It accounts for overdispersion, where the variance is greater than the mean, which is common in accident data influenced by multiple factors such as weather & road conditions. GLM can handle both continuous and categorical predictors, making it flexible for modeling various variables. The output provides interpretable coefficients, which helps to understand the influence of each predictor on the likelihood of accidents.

Generalized Additive Models (GAM) with a Negative Binomial distribution : GAM may be more appropriate if there are non-linear relationships between predictors and the response variable, for example extremely high EHF could lead to more accidents due to heat stress, Similarly, extremely low or negative EHF indicate relatively normal or cooler conditions, could coincide with other factors like rain that may also increase accident rates. GAM allows for flexible, non-linear relationships through smooth functions and are able to capture complex patterns that GLM might miss, especially in weather-related variables like temperature or EHF.

Q7: Model the number of road traffic accidents (30 points) In this question you will build a model to predict the number of road traffic accidents. You will use the car_accidents_victoria dataset and the weather data. We can start with simple models and gradually make them more complex and improve them. For example, you can use the EHF as an additional predictor to augment the model. Let's denote by Y the road traffic accident variable. Randomly pick a region from the road traffic accidents data.

Q7.1 Which region do you pick? (1 point) the region i will pick is METROPOLITAN.NORTH.WEST.REGION because it is near to melbourne airport and the region is a urban area which expected to experience a variety of traffic and weather conditions, making it suitable for modeling road traffic accidents.

```r
# Filter accident data for METROPOLITAN.NORTH.WEST.REGION
metro_nw_accidents <- tidy_accidents_cleaned %>%
  filter(region == "METROPOLITAN.NORTH.WEST.REGION") %>%
  group_by(DATE) %>%
  summarize(daily_total_accidents = sum(count))

# Merge with weather data
combined_data <- metro_nw_accidents %>%
  left_join(noaa_data_filtered, by = "DATE") %>%
  mutate(
    day_of_week = factor(weekdays(DATE)),
    month = factor(month(DATE, label = TRUE))
  )

# Remove any rows with NA values
combined_data <- na.omit(combined_data)
```

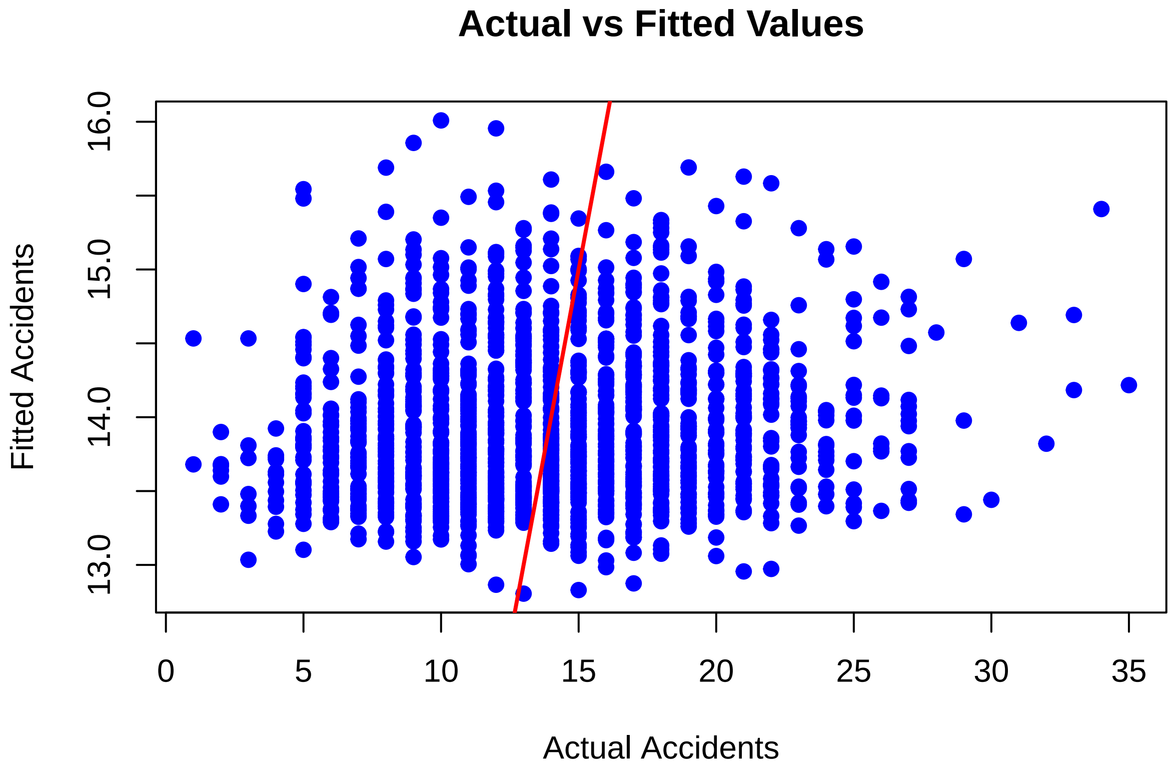
Q7.2 Fit a linear model for Y according to your model(s) above. Plot the fitted values and the residuals. Assess the model fit. Is a linear function sufficient for modelling the trend of Y? Support your conclusion with plots. (4 points)

```r
# Fit the linear model
model <- lm(daily_total_accidents ~ TMAX + PRCP + EHF , data = combined_data)

# Summary of the model
summary(model)
```

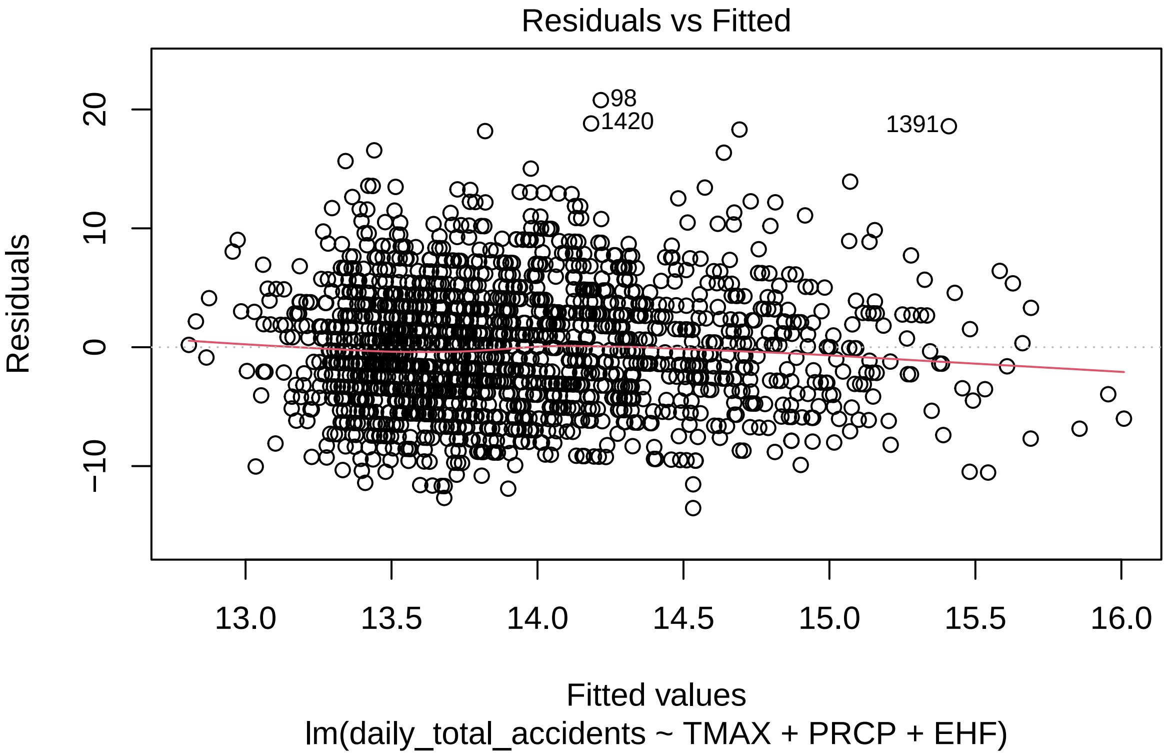
```
##
## Call:
```

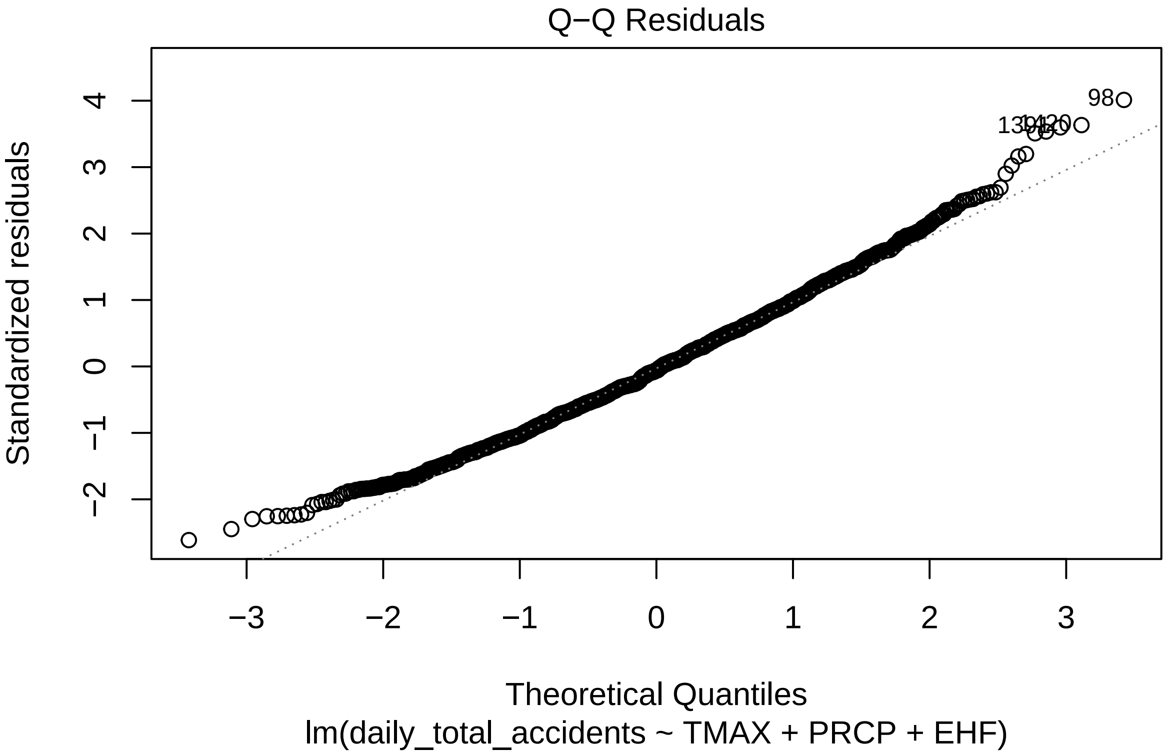
```
## lm(formula = daily_total_accidents ~ TMAX + PRCP + EHF, data = combined_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5332  -3.6240  -0.3133   3.3316  20.7829
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.69126    0.64668  18.079  < 2e-16 ***
## TMAX         0.08858    0.02278   3.888 0.000105 ***
## PRCP        -0.01137    0.02730  -0.416 0.677126
## EHF         -0.03654    0.01944  -1.880 0.060291 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.183 on 1610 degrees of freedom
## Multiple R-squared:  0.0101, Adjusted R-squared:  0.008258
## F-statistic: 5.477 on 3 and 1610 DF,  p-value: 0.0009597
```

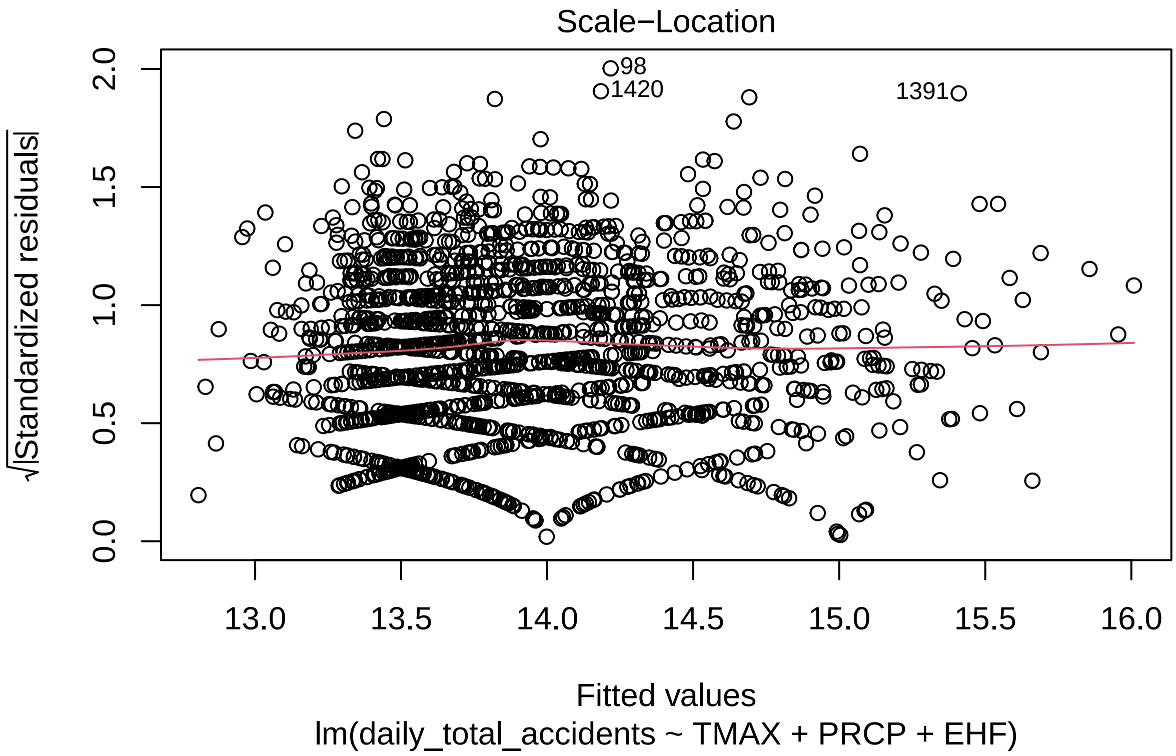
```r
# Plot: Actual vs Fitted values
plot(combined_data$daily_total_accidents, fitted(model),
     main = "Actual vs Fitted Values",
     xlab = "Actual Accidents",
     ylab = "Fitted Accidents",
     col = "blue", pch = 19)
abline(0, 1, col = "red", lwd = 2)  # Line y = x for reference
```

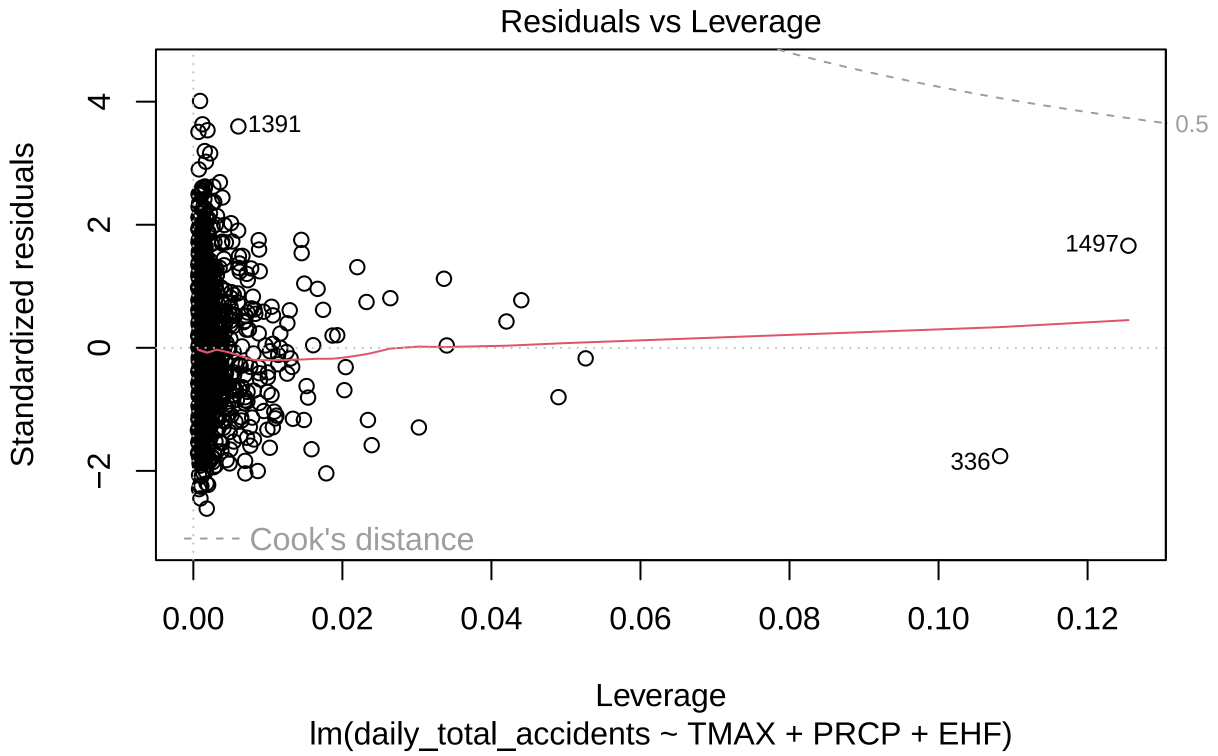


```
plot(model)
```


Residuals vs Fitted
Residuals
20
10
0
−10
13.0
13.5
14.0
14.5
15.0
15.5
16.0
Fitted values
lm(daily_total_accidents ~ TMAX + PRCP + EHF)
98
1420
1391


Q−Q Residuals
Standardized residuals
4
3
2
1
0
−1
−2
98
139
120
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(daily_total_accidents ~ TMAX + PRCP + EHF)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
13.0
13.5
14.0
14.5
15.0
15.5
16.0
98
1420
1391
Fitted values
lm(daily_total_accidents ~ TMAX + PRCP + EHF)



Residuals vs Fitted Plot: The residuals show a slight curved pattern, particularly at the lower and upper ends of the fitted values, suggesting that the relationship between the predictors (TMAX, PRCP, EHF) and the response variable (daily_total_accidents) may not be entirely linear.

Residual vs Leverage: although there are some outliers such as point (1497,336,1391), but all the point lies in the cook distance, indicate there are no extreme outlier.

Q-Q Plot: The residuals mostly follow the diagonal line, but there are deviations at both tails, indicates that while the residuals are approximately normally distributed, there are some extreme values that do not follow a normal distribution.

Scale Location: The pattern suggests some heteroscedasticity, meaning the linear model may not be fully appropriate, as the variance of the residuals is not constant.

Actual vs Fitted Values Plot: The points show a horizontal spread rather than closely following the red line, indicating that the model does not predict the actual number of accidents very accurately. This suggests that a linear model may not fully capture the underlying relationship between the predictors and the response variable.

The R^2 value of 0.0101 is very low, indicating that the model explains only 1% of the variance in daily total accidents. This suggests that the current model, which includes TMAX, PRCP, and EHF, is not very effective at predicting road accidents. The residual plots also show potential non-linearity and slight heteroscedasticity, suggesting that a more complex model, such as a Generalized Additive Model (GAM), may be needed to capture non-linear relationships, especially with variables like EHF and TMAX.

Q7.3 As we are not interested in the trend itself, relax the linearity assumption by fitting a generalised additive model (GAM). Assess the model fit. Do you see patterns in the residuals indicating insufficient model fit? (5 points)

```r
library(mgcv)   ## load the package
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

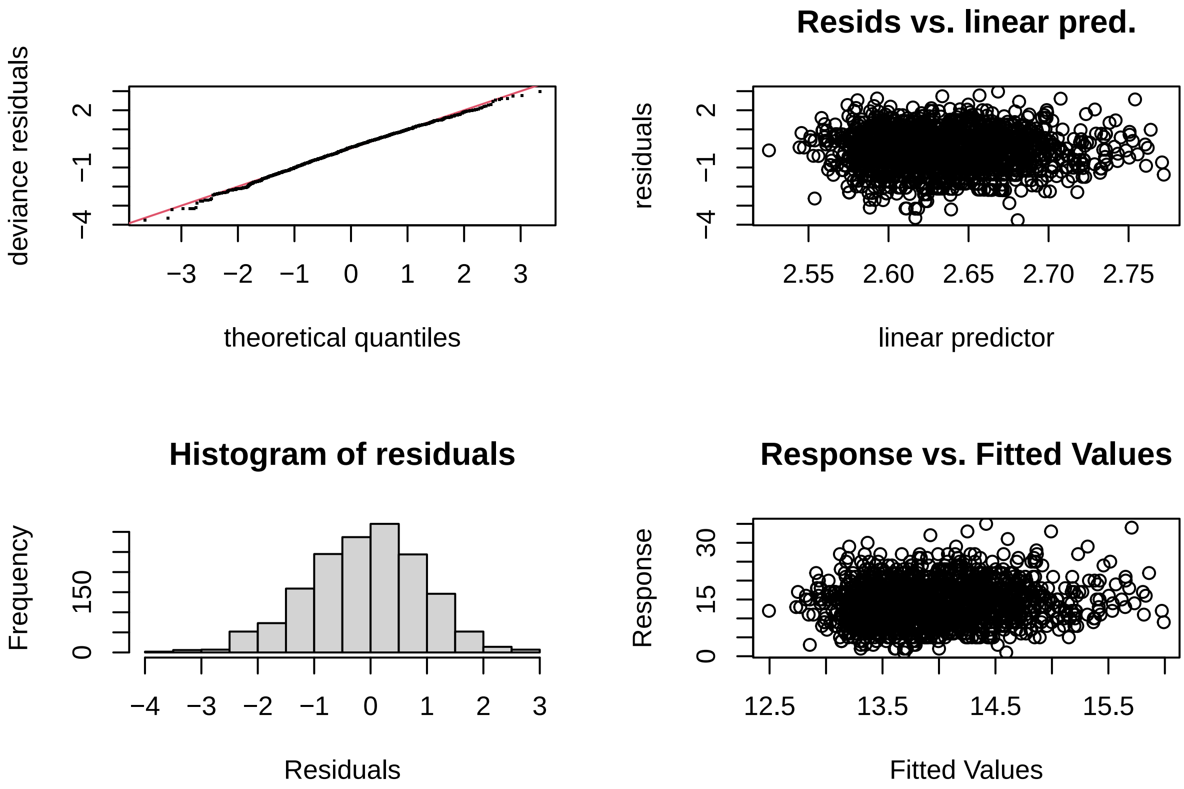
```r
library(gamair) ## load the data package

# Fit the Generalized Additive Model (GAM)
# Using smoothing splines for TMAX, PRCP, and EHF
gam_model <- gam(daily_total_accidents ~ s(TMAX) + s(PRCP) + s(EHF), family = nb(), data = combined_dat
gam_model
```

```
##
## Family: Negative Binomial(14.489)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX) + s(PRCP) + s(EHF)
##
## Estimated degrees of freedom:
## 1.84 1.00 2.69  total = 6.53
##
## REML score: 4941.033
```

```r
# Summary of the GAM model
summary(gam_model)
```

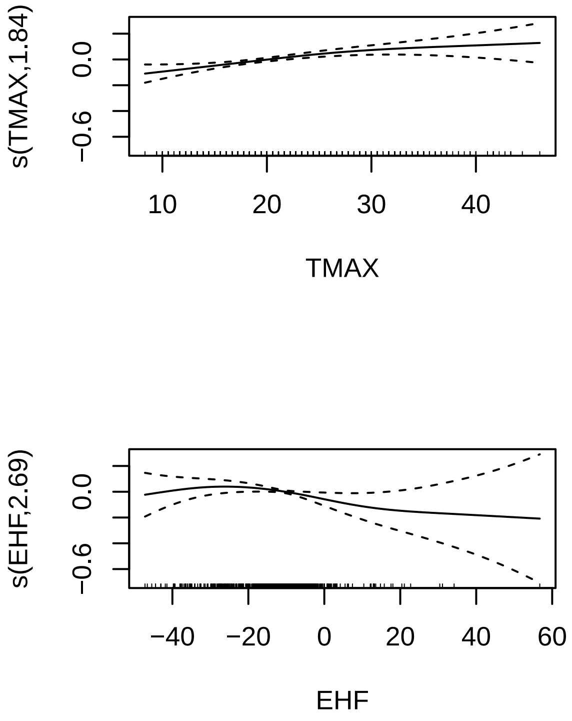
```
##
## Family: Negative Binomial(14.489)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX) + s(PRCP) + s(EHF)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.634057   0.009342     282   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df Chi.sq  p-value
## s(TMAX) 1.836  2.315 17.731 0.000236 ***
## s(PRCP) 1.003  1.005  0.093 0.763239
## s(EHF)  2.690  3.477  5.784 0.163175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0117   Deviance explained = 1.41%
## -REML =   4941  Scale est. = 1         n = 1614
```

```
# Diagnostic plots
par(mfrow = c(2,2))
gam.check(gam_model)
```



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-0.001177436,0.0005017431]
## (score 4941.033 & scale 1).
## Hessian positive definite, eigenvalue range [0.001175505,183.3327].
## Model rank =  28 / 28
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##           k'   edf k-index p-value
## s(TMAX) 9.00 1.84    0.84  <2e-16 ***
## s(PRCP) 9.00 1.00    0.80  <2e-16 ***
## s(EHF)  9.00 2.69    0.93   0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Additional residual plots
plot(gam_model, page = 1, all.terms = TRUE)
```



Estimated Degrees of Freedom (edf): TMAX: The edf is 1.84, indicating a non-linear relationship between maximum temperature and daily total accidents. PRCP: The edf is 1, indicating that the relationship between precipitation and daily total accidents is almost linear. EHF: The edf is 2.69, suggesting a non-linear relationship between the EHF and daily total accidents, though the relationship is more complex.

p-value: TMAX: p value of 0.000239 is highly significant, suggesting that temperature plays an important role in predicting daily total accidents. PRCP: p-value is 0.763, showing that precipitation is not significant in this model. This suggests that precipitation does not have a notable effect on daily total accidents. EHF: p-value is 0.164, indicates that EHF is not statistically significant at the typical 0.05 level. However, it shows some influence, given the complexity of the smooth term (edf = 2.69).

Rˆ2: The model explains only 1.17% of the variability in the data, which is very low. This suggests that other factors not included in the model may have a larger impact on daily total accidents.

Deviance: The model explains only 1.41% of the deviance, which is quite low, indicating that the model does not capture a large amount of variability in the accident counts.

TMAX plot:The curve is slightly upward sloping, suggests that as TMAX increases, the expected number of daily accidents slightly increases, though not in a strongly non-linear manner. The confidence intervals (dashed lines) are relatively narrow, indicating higher confidence in this trend.

PRCP plot:The plot suggests a very flat line, indicating that precipitation has minimal impact on daily accidents in this model. The confidence intervals (dashed lines) also show little change across the range of PRCP values, indicate the lack of significance for this predictor.

EHF plot:The curve shows a slight increase and then flattens out, with larger confidence intervals towards the extremes of the EHF values, suggesting some uncertainty in how EHF influences accidents at very low and very high values. This indicates that EHF might have a weak, non-linear relationship with accident counts.

Residual vs Fitted Values: The residuals appear randomly scattered around zero, which suggests no obvious patterns and indicates that the model has captured some aspects of the relationship. However, the high concentration around certain values indicates that there may still be a lack of fit.

Histogram of Residuals: The residuals are mostly centered around zero and seem to follow a normal distribution. This supports the assumption of residual normality, which is important for model validity.

Q-Q Plot: shows a fairly good alignment with the theoretical quantiles, indicating that the residuals are approximately normally distributed. There are some deviations at the tails, but nothing extreme.

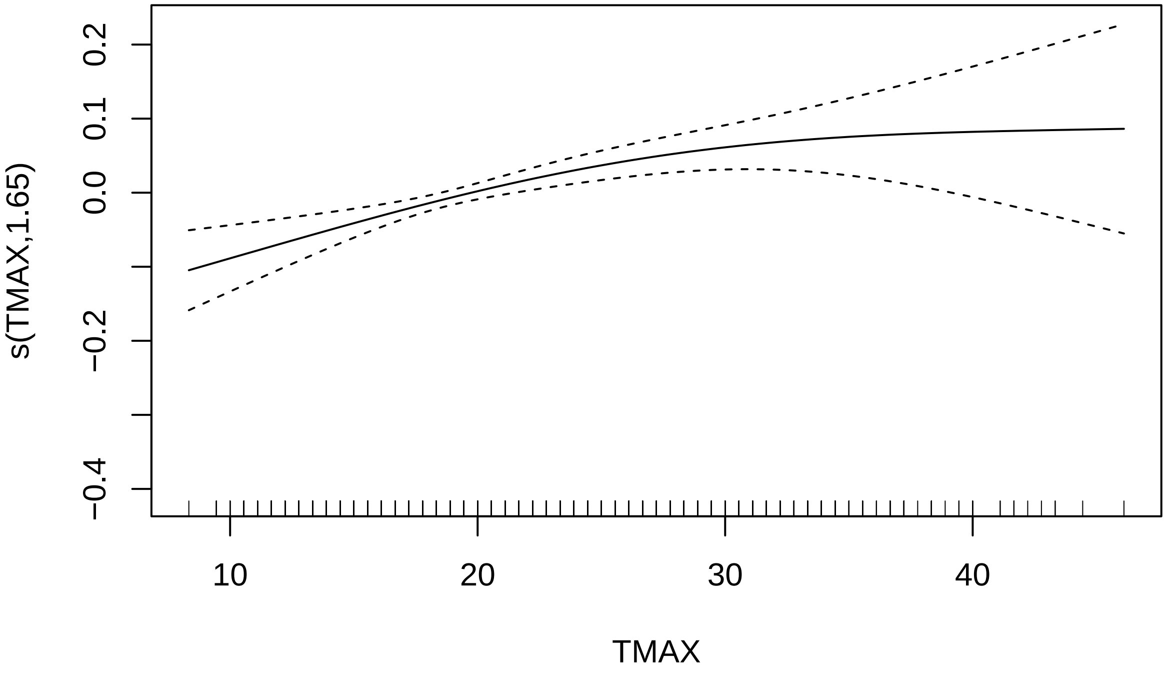
conclusion: TMAX is significant, non-linear relationship between temperature and daily total accidents, suggesting that temperature affects accident rates.Although EHF is not significant, there may be a non-linear relationship between EHF and accidents, but this model does not provide strong evidence for its influence. Precipitation does not seem to have a significant effect on daily total accidents in this model. Given the low adjusted R-squared and deviance explained, yes there is insufficient model fit, the model may need further improvement such as change in basis function or additional predictors to better capture the variability in daily total accidents.

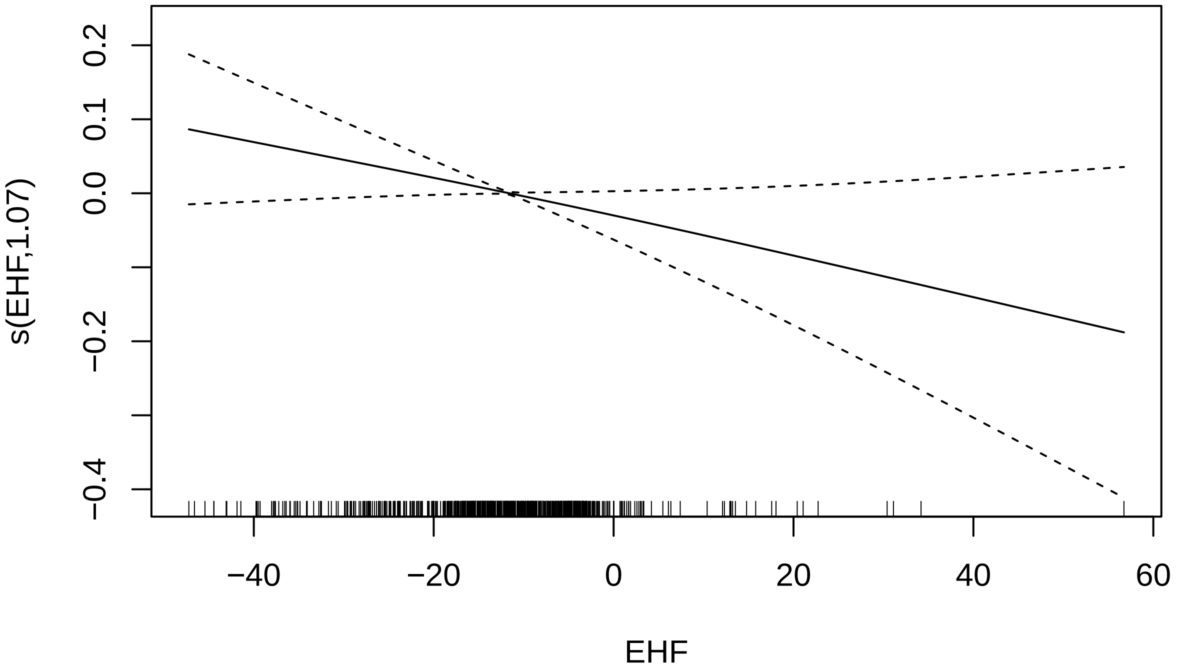
Since Precipitation do not have a significant effect on total accident, it is removed. Now, different basis dimensions (k=3,5,10) are analyse to determine the choice of k.

```
gam_model3 <- gam(daily_total_accidents ~ s(TMAX, k = 3) + s(EHF, k = 3),
                  family = nb(), data = combined_data)
summary(gam_model3)
```

```
##
## Family: Negative Binomial(14.438)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX, k = 3) + s(EHF, k = 3)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.63420    0.00935   281.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df Chi.sq  p-value
## s(TMAX) 1.645  1.873 20.442 0.000432 ***
## s(EHF)  1.071  1.136  3.159 0.081081 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.00993   Deviance explained =  1.1%
## -REML = 4937.7  Scale est. = 1         n = 1614
```

```r
plot(gam_model3)
```

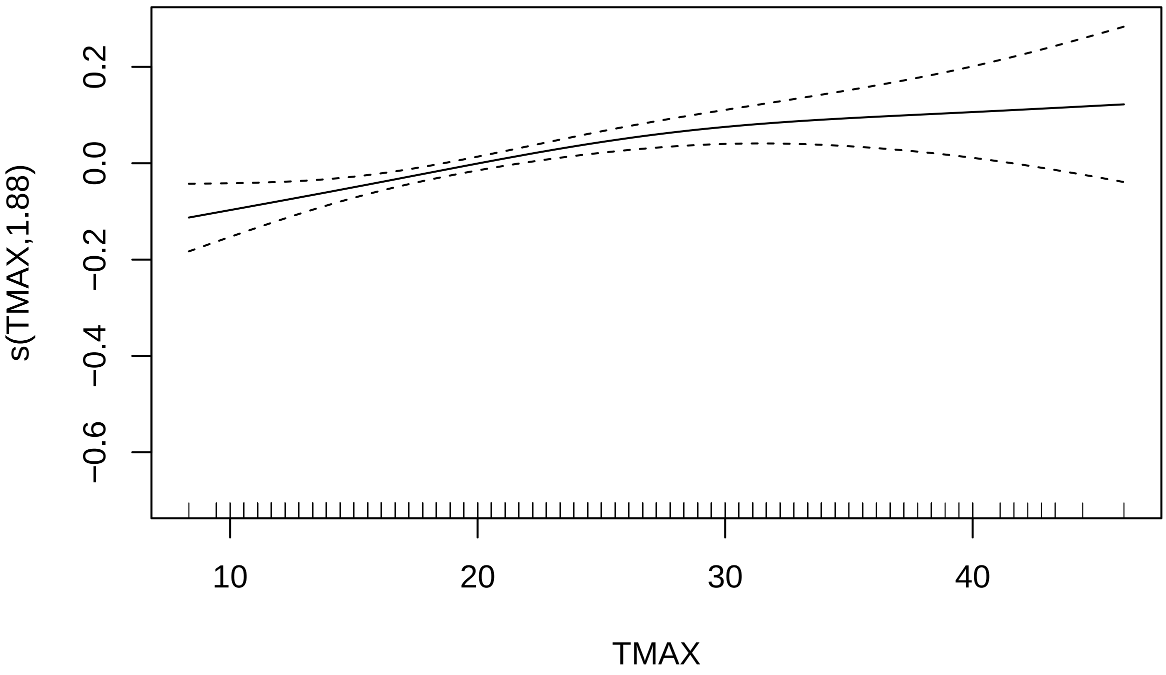


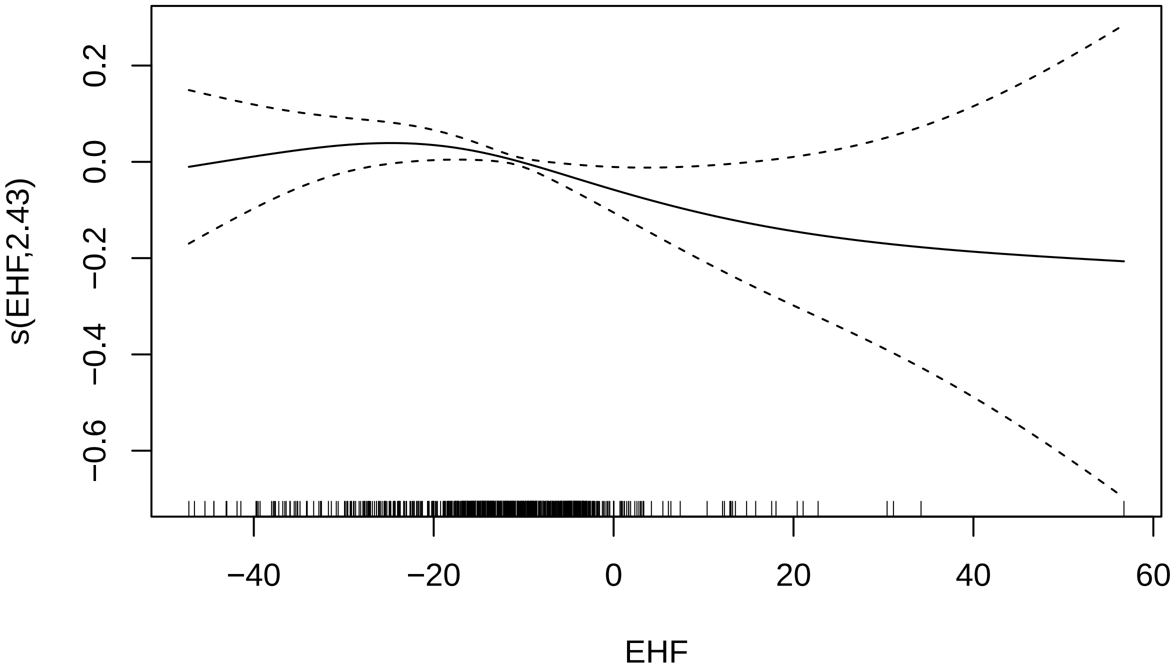


```r
gam_model5 <- gam(daily_total_accidents ~ s(TMAX, k = 5) + s(EHF, k = 5),
                  family = nb(), data = combined_data)
summary(gam_model5)
```

```
##
## Family: Negative Binomial(14.503)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX, k = 5) + s(EHF, k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.63407    0.00934     282   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##           edf Ref.df Chi.sq p-value
## s(TMAX) 1.878  2.322 19.988   8e-05 ***
## s(EHF)  2.426  2.995  6.358  0.0959 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.0122   Deviance explained = 1.38%
## -REML = 4937.3  Scale est. = 1         n = 1614
```

```r
plot(gam_model5)
```

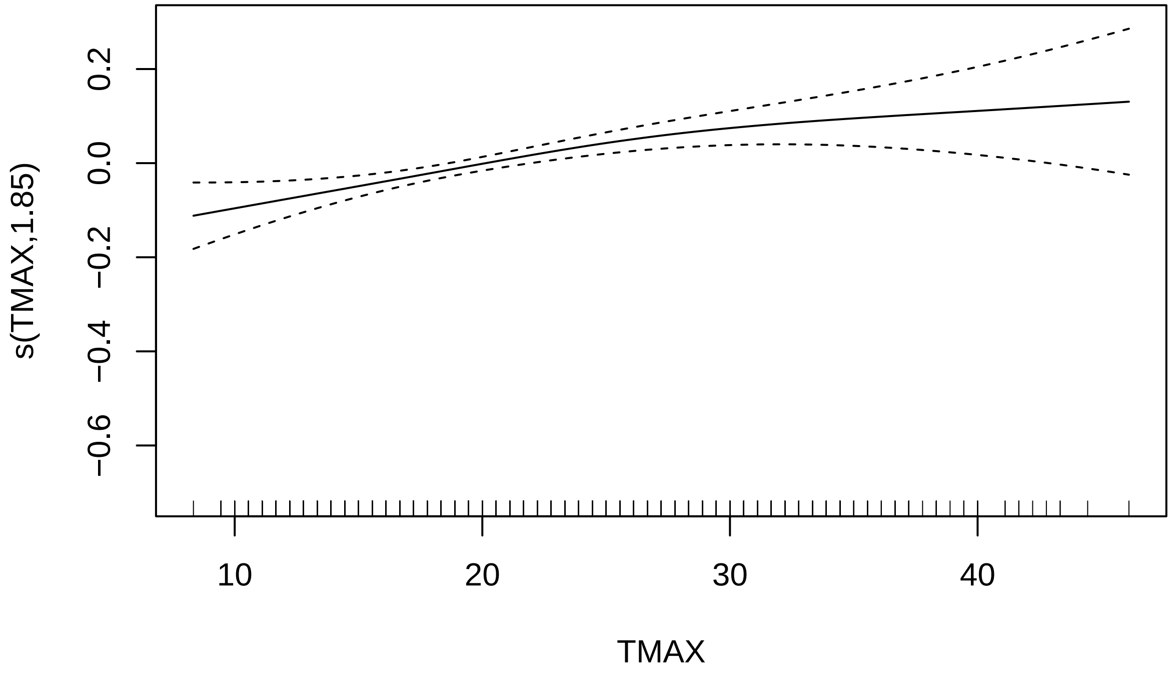


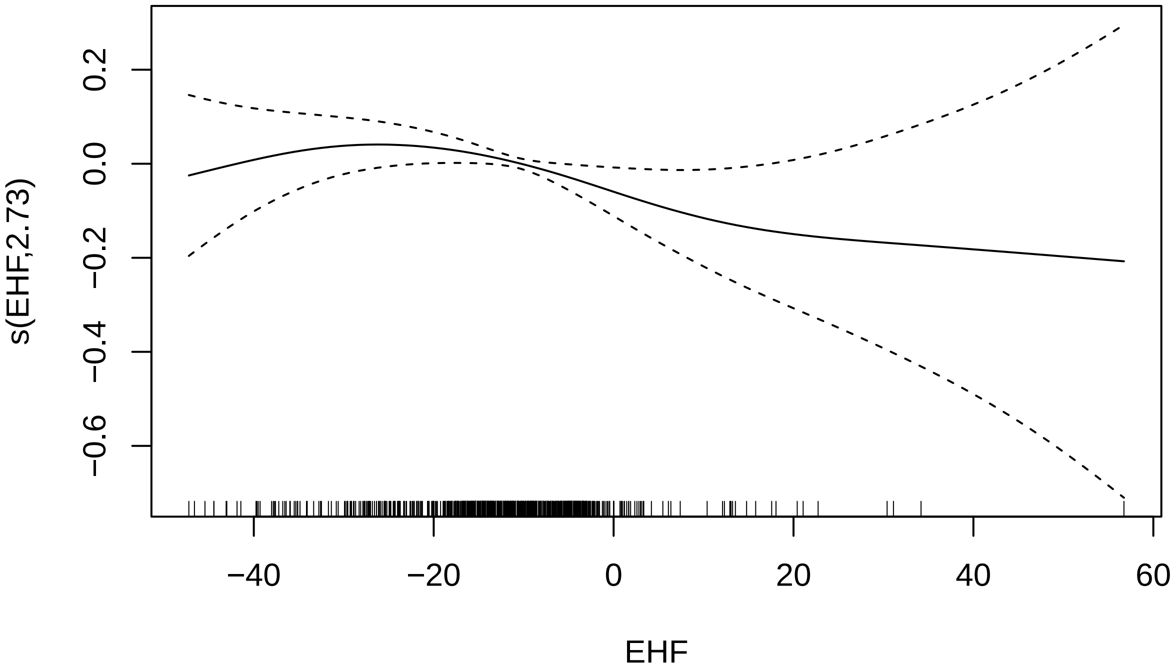


```r
gam_model10 <- gam(daily_total_accidents ~ s(TMAX, k = 10) + s(EHF, k = 10),
                   family = nb(), data = combined_data)
summary(gam_model10)
```

```
##
## Family: Negative Binomial(14.507)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX, k = 10) + s(EHF, k = 10)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.634057   0.009339     282   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df Chi.sq p-value
## s(TMAX) 1.846  2.328 18.902 0.00014 ***
## s(EHF)  2.735  3.533  6.113 0.14765
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0123   Deviance explained = 1.41%
## -REML = 4937.4  Scale est. = 1         n = 1614
```

```r
plot(gam_model10)
```





Q7.4 Compare the models using the Akaike information criterion (AIC). Report the best-fitted model through coefficient estimates and/or plots. (5 points)

```
AIC(gam_model3)
```

```
## [1] 9861.326
```

```
AIC(gam_model5)
```

```
## [1] 9861.173
```

```
AIC(gam_model10)
```

```
## [1] 9861.73
```

As k increases, the model become more complex (edf higher).

Model Fit: The fit improves from k=3 to k=5 (R^2 and deviance explained increase). However, there's a slight decrease in fit from k=5 to k=10.

AIC: The AIC is lowest for k=5 (9861.173), indicating this model provides the best balance between fit and complexity, although is not close to 0.

REML: REML is lowest for k=5, again suggesting this is the optimal model among the three.

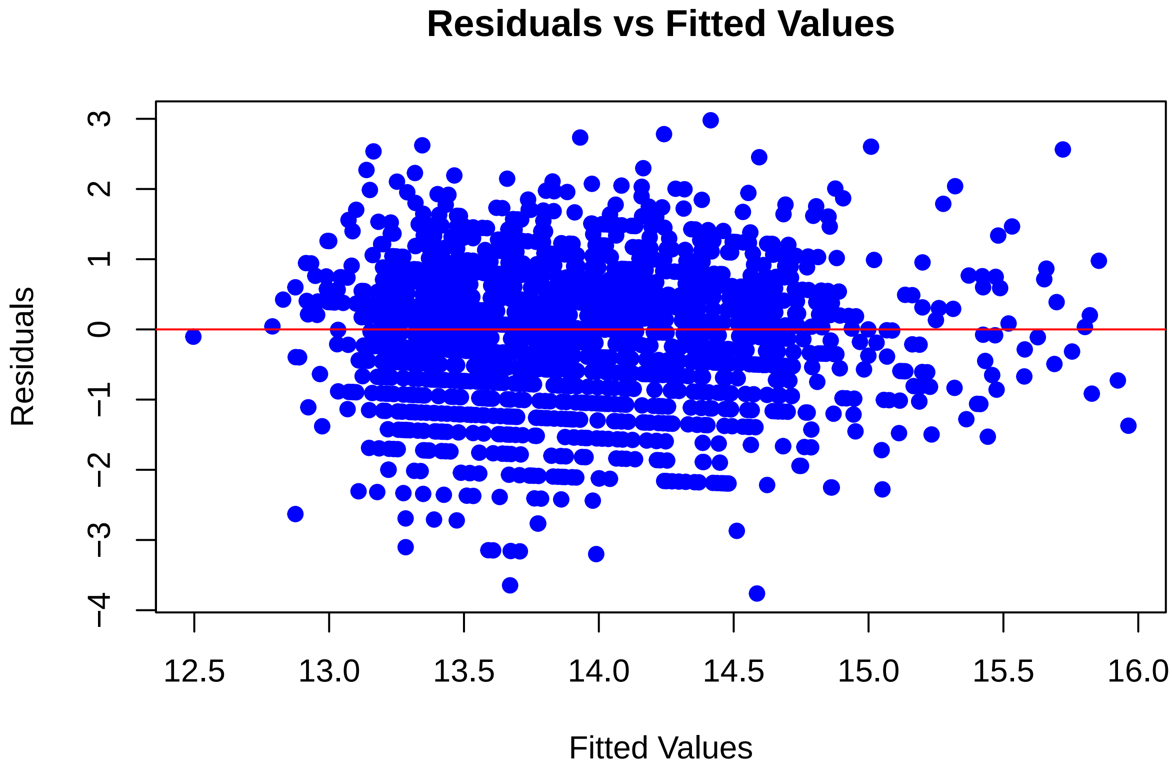
Overfitting Risk: While k=10 allows for more complexity, it does not improve the model, suggesting potential overfitting if higher k values were used.

Plot: k = 3: Likely underfits the data, missing some non-linear relationships. k = 5: Strikes the best balance, capturing non-linear relationships while avoiding overfitting. The AIC suggests this is the most appropriate model. k = 10: Introduces too much complexity, possibly overfitting the data, as seen by the wider confidence intervals.

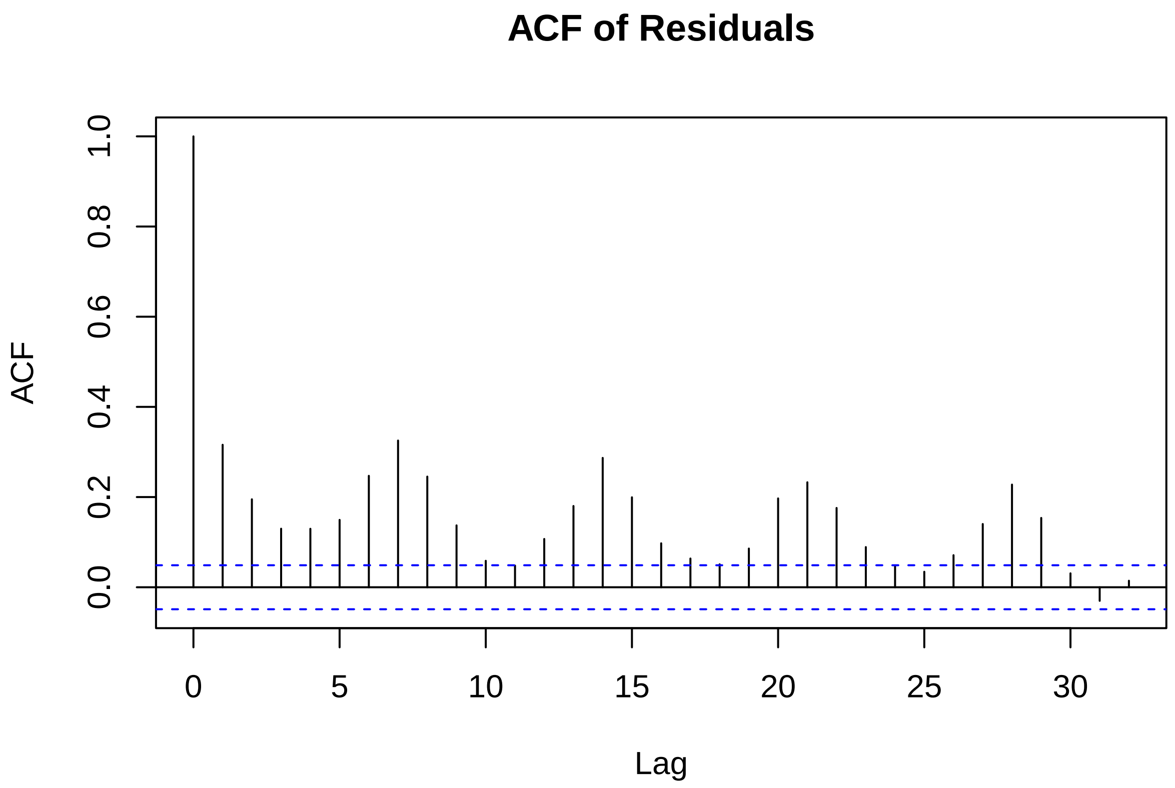
Conclusion: k=5 appears to be the optimal choice. It explains more variance, keeps the model complexity moderate, and yields the best model fit based on AIC and REML. The s(TMAX) (edf = 1.878, p-value = 8e-05) smoother shows a significant non-linear relationship, whereas s(EHF)(edf = 2.426, p-value = 0.0959) indicating some complexity in the relationship between EHF and accidents, but not strongly significant.

Q7.5 Analyse the residuals. Do you see any correlation patterns among the residuals? (4 points)

```r
# Plotting the smooth terms for the best model
# Residuals vs Fitted plot
plot(fitted(gam_model5), residuals(gam_model5),
     main = "Residuals vs Fitted Values",
     xlab = "Fitted Values",
     ylab = "Residuals", pch = 19, col = "blue")
abline(h = 0, col = "red")
```



```r
# ACF plot of residuals
acf(residuals(gam_model5), main = "ACF of Residuals")
```



residuals vs fitted values: plot shows that the residuals are randomly scattered around zero, indicating that the model has captured the underlying relationship without obvious patterns of misspecification.

ACF plot: does not indicate any significant autocorrelation, suggesting that the residuals are independent over time or sequence.

Overall, the residual diagnostics suggest that the model is well-specified, and there is no evidence of strong correlations or patterns in the residuals.

Q7.6 Does the predictor EHF improve the model fit? (1 point)

```
# Model with EHF as a predictor
gam_model_with_ehf <- gam(daily_total_accidents ~ s(TMAX, k = 5)  + s(EHF, k = 5),
                          family = nb(), data = combined_data)

# Model without EHF as a predictor
gam_model_without_ehf <- gam(daily_total_accidents ~ s(TMAX, k = 5),
                             family = nb(), data = combined_data)

# Compare AIC values
AIC_with_ehf <- AIC(gam_model_with_ehf)
AIC_without_ehf <- AIC(gam_model_without_ehf)

# Print AIC values
cat("AIC with EHF: ", AIC_with_ehf, "\n")
```

```
## AIC with EHF:  9861.173
```

```
cat("AIC without EHF: ", AIC_without_ehf, "\n")
```

```
## AIC without EHF:  9862.901
```

```
# Check significance of EHF in the model with EHF
summary(gam_model_with_ehf)
```

```
##
## Family: Negative Binomial(14.503)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX, k = 5) + s(EHF, k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.63407    0.00934     282   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df Chi.sq p-value
## s(TMAX) 1.878  2.322 19.988   8e-05 ***
## s(EHF)  2.426  2.995  6.358  0.0959 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0122   Deviance explained = 1.38%
## -REML = 4937.3  Scale est. = 1         n = 1614
```

Based on the model fit from Q7.6, EHF does improve the model fit slightly, but its predictive power is weak. The model with EHF had a lower AIC (9861.173), which suggests that it slightly improves the model, but the difference is not dramatic. The p-value of s(EHF) was 0.0959, indicating a weak level of statistical significance. The model with EHF explained 1.38% of the deviance, which is quite low. Thus, EHF alone does not seem to be a strong predictor of road traffic accidents, though it may contribute some value in combination with other variables.

Q7.7 Is EHF a good predictor for road traffic accidents? Can you think of extra weather features that may be more predictive of road traffic accident numbers? Try incorporating your feature into the model and see if it improves the model fit. Use AIC to prove your point. (10 points)

Since temperature are highly significant to the daily total accident, other factor such as Fluctuations in temperature during the day could have an effect on road safety. Larger temperature differences could contribute to adverse conditions, such as fog, ice, or other factors.

```
# Add Temperature Variability as a new predictor
combined_data$Temp_Variability <- combined_data$TMAX - combined_data$TMIN

# Fit a GAM model with EHF and Temp_Variability
gam_model_var <- gam(daily_total_accidents ~ s(TMAX, k=5) + s(EHF, k=5) + s(Temp_Variability, k=5) ,
                     family = nb(), data = combined_data)

# Summary of the new model
summary(gam_model_var)
```

```
##
## Family: Negative Binomial(14.928)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX, k = 5) + s(EHF, k = 5) + s(Temp_Variability,
##     k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.633116   0.009279   283.8   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                        edf Ref.df Chi.sq  p-value
## s(TMAX)              2.177  2.688  4.349 0.175881
## s(EHF)               1.009  1.017  0.651 0.426379
## s(Temp_Variability) 3.521  3.859 20.319 0.000229 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.0261   Deviance explained = 2.77%
## -REML = 4932.4  Scale est. = 1         n = 1614
```

```
# Compare AIC with the model that includes Temp_Variability
# AIC of the new model with Temp_Variability
AIC(gam_model_var)
```

```
## [1] 9841.971
```

```
# AIC of the model with EHF only
AIC(gam_model_with_ehf)
```

```
## [1] 9861.173
```

```
# Fit a GAM model with EHF and Temp_Variability
gam_model_latest<- gam(daily_total_accidents ~ s(TMAX, k=5) + s(Temp_Variability, k=5) ,
                     family = nb(), data = combined_data)

# Summary of the new model
summary(gam_model_latest)
```

```
##
## Family: Negative Binomial(14.938)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX, k = 5) + s(Temp_Variability,
##     k = 5)
##
## Parametric coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.633138   0.009277   283.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                       edf Ref.df Chi.sq  p-value
## s(TMAX)             2.255  2.771  4.807    0.153
## s(Temp_Variability) 3.517  3.856 22.939 7.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0263   Deviance explained = 2.74%
## -REML = 4929.1  Scale est. = 1         n = 1614
```

```
# Compare AIC with the model that includes Temp_Variability
# AIC of the new model with Temp_Variability
AIC(gam_model_var)
```

```
## [1] 9841.971
```

```
AIC(gam_model_latest)
```

```
## [1] 9840.524
```

model result: Deviance increased from 1.41% in the earlier models to 2.77% when Temp_Variability is added, showing an improvement in the model's explanatory power

R^2 increases to 2.63%, indicating that adding Temp_Variability provides more predictive power than EHF.

p-value for Temp_Variability is highly significant in both of the the model (with EHF or without EHF). This suggests that fluctuations in temperature during the day are a strong predictor of road traffic accidents.

TMAX is not as significant in the models, indicating that while temperature is important, variability in temperature plays a larger role in predicting accidents.

EHF, in contrast, remains non-significant, with a high p-value ($p = 0.426$), showing that it does not contribute much to the model when Temp_Variability is included.

AIC Comparison: AIC with TMAX, Temp_Variability and EHF: 9841.971 AIC with TMAX, Temp_Variability (without EHF): 9840.524 The AIC decreases when include Temp_Variability in the model, and it further decreases when remove EHF, suggesting that Temp_Variability is a useful predictor, while EHF does not contribute much to the model.

Deviance explained: Deviance explained is 2.77% with EHF and Temp_Variability, which slightly decreases to 2.74% when EHF is removed but 1.38% for the model with EHF and TMAX only. This minor decrease shows that EHF has a small impact on model performance, but Temp_Variability remains the most important predictor.

EHF is not a strong predictor of road traffic accidents. Its p-value is non-significant, and removing it from the model improves the AIC slightly. When i go through the data, there are only 50 positive EHF value from 1614 observation and 8 of them are 0. Hence there is limited information for the model to capture a meaningful relationship between EHF and road accidents. The small number of observations make it difficult for the model to accurately learn the influence of EHF on accidents. EHF is specifically designed to capture extreme heatwave events, which are relatively rare. Most of the time, temperatures may not reach the levels necessary for EHF to be positive. Hence, in regions or periods without frequent heatwaves, EHF may not contribute significantly to predicting daily accident rates.

EHF has some small contribution, but in general, it does not add much predictive value. Given the limited occurrences of positive EHF values, it is better to keep EHF as a secondary or supporting variable, as it might play a significant role during specific high-risk periods, making it valuable for certain use cases. Alternatively, continuing to focus on predictors such as Temperature Variability or predictors that are statistically significant and are able to explained the variability in the response data for improving model performance.

In this case, EXTRA FEATURES such as wind direction, wind speed and pressure is added into consideration. (DATA IS GET FROM Meteostat)

```
extra_data <- read.csv("extradata.csv")
head(extra_data)
```

```
##         date wdir wspd   pres
## 1 2016-01-30  207 13.7     NA
## 2 2016-01-31    1 23.3 1001.7
## 3 2016-02-01  194 14.1 1005.5
## 4 2016-02-02  335 15.6 1003.2
## 5 2016-02-03  203 23.8 1009.6
## 6 2016-02-04  180 25.3 1018.4
```

```
summary(extra_data)
```

```
##      date                wdir            wspd            pres
##  Length:1614        Min.   :  0.0   Min.   : 3.80   Min.   : 962.2
##  Class :character   1st Qu.:178.0   1st Qu.:13.80   1st Qu.:1011.5
##  Mode  :character   Median :225.0   Median :18.20   Median :1016.6
##                     Mean   :214.8   Mean   :19.45   Mean   :1016.4
##                     3rd Qu.:309.0   3rd Qu.:24.20   3rd Qu.:1021.6
##                     Max.   :360.0   Max.   :55.30   Max.   :1036.2
##                     NA's   :12      NA's   :1       NA's   :31
```

```
# Rename the 'date' column in extra_data to 'DATE' to match combined_data
colnames(extra_data)[colnames(extra_data) == "date"] <- "DATE"

# Ensure both DATE columns are in Date format
combined_data$DATE <- as.Date(combined_data$DATE, format = "%Y-%m-%d")
extra_data$DATE <- as.Date(extra_data$DATE, format = "%Y-%m-%d")

# Merge the datasets on the DATE column
combined_data <- merge(combined_data, extra_data, by = "DATE", all.x = TRUE)

# Check the merged dataset
head(combined_data)
```

```
##         DATE daily_total_accidents     STATION                  NAME PRCP
## 1 2016-01-30                    15 ASN00086282 MELBOURNE AIRPORT, AS  7.8
## 2 2016-01-31                     7 ASN00086282 MELBOURNE AIRPORT, AS  8.8
## 3 2016-02-01                    19 ASN00086282 MELBOURNE AIRPORT, AS  0.6
## 4 2016-02-02                    17 ASN00086282 MELBOURNE AIRPORT, AS  0.0
## 5 2016-02-03                    18 ASN00086282 MELBOURNE AIRPORT, AS  0.2
## 6 2016-02-04                    27 ASN00086282 MELBOURNE AIRPORT, AS  1.6
##       TAVG     TMAX     TMIN TAVG_3day TAVG_30day    EHIsig    EHIaccl
## 1 17.77778 22.77778 11.11111  17.22222   20.27778 -7.222222 -3.0555556
## 2 15.00000 20.55556 14.44444  15.74074   20.09259 -8.703704 -4.3518519
## 3 19.44444 25.00000 10.00000  17.40741   20.05556 -7.037037 -2.6481481
## 4 23.33333 32.22222 15.00000  19.25926   20.16667 -5.185185 -0.9074074
## 5 17.22222 21.11111 17.22222  20.00000   20.11111 -4.444444 -0.1111111
## 6 17.77778 21.11111 16.11111  19.44444   20.01852 -5.000000 -0.5740741
##         EHF day_of_week month Temp_Variability wdir wspd   pres
## 1 -7.222222    Saturday   Jan        11.666667  207 13.7     NA
## 2 -8.703704      Sunday   Jan         6.111111    1 23.3 1001.7
## 3 -7.037037      Monday   Feb        15.000000  194 14.1 1005.5
## 4 -5.185185     Tuesday   Feb        17.222222  335 15.6 1003.2
## 5 -4.444444   Wednesday   Feb         3.888889  203 23.8 1009.6
## 6 -5.000000    Thursday   Feb         5.000000  180 25.3 1018.4
```

```r
library(VIM)
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```r
# Perform KNN imputation on missing values in 'wdir', 'wspd', and 'pres'
combined_data <- kNN(combined_data, variable = c("wdir", "wspd", "pres"), k = 5)

# Check the dataset after imputation
summary(combined_data)
```

```
##       DATE             daily_total_accidents   STATION
##  Min.   :2016-01-30   Min.   : 1.00         Length:1614
##  1st Qu.:2017-03-08   1st Qu.:10.00         Class :character
##  Median :2018-04-15   Median :14.00         Mode  :character
##  Mean   :2018-04-15   Mean   :13.94
##  3rd Qu.:2019-05-23   3rd Qu.:17.00
##  Max.   :2020-06-30   Max.   :35.00
##
##      NAME                PRCP             TAVG             TMAX
##  Length:1614        Min.   : 0.00   Min.   : 4.444   Min.   : 8.333
##  Class :character   1st Qu.: 0.00   1st Qu.:10.556   1st Qu.:15.556
##  Mode  :character   Median : 0.00   Median :13.889   Median :19.444
##                     Mean   : 1.53   Mean   :14.875   Mean   :20.793
##                     3rd Qu.: 0.60   3rd Qu.:18.333   3rd Qu.:25.000
##                     Max.   :69.00   Max.   :34.444   Max.   :46.111
##
##       TMIN          TAVG_3day        TAVG_30day         EHIsig
##  Min.   :-2.222   Min.   : 6.111   Min.   : 8.704   Min.   :-18.333
##  1st Qu.: 6.667   1st Qu.:10.926   1st Qu.:10.986   1st Qu.:-13.519
##  Median : 9.444   Median :14.259   Median :15.074   Median :-10.185
##  Mean   : 9.912   Mean   :14.878   Mean   :14.969   Mean   : -9.566
##  3rd Qu.:12.778   3rd Qu.:18.333   3rd Qu.:19.218   3rd Qu.: -6.111
##  Max.   :26.667   Max.   :30.370   Max.   :22.481   Max.   :  5.926
##
##     EHIaccl              EHF             day_of_week      month
##  Min.   :-6.87037   Min.   :-47.222   Friday   :230   Mar    :155
##  1st Qu.:-1.75926   1st Qu.:-15.000   Monday   :231   May    :155
##  Median :-0.33333   Median :-11.852   Saturday :231   Apr    :150
##  Mean   :-0.09056   Mean   :-11.691   Sunday   :231   Jun    :150
##  3rd Qu.: 1.36574   3rd Qu.: -7.222   Thursday :230   Feb    :142
##  Max.   : 9.57407   Max.   : 56.735   Tuesday  :231   Jan    :126
##                                       Wednesday:230   (Other):736
##  Temp_Variability       wdir            wspd            pres
##  Min.   : 0.5556   Min.   :  0.0   Min.   : 3.80   Min.   : 962.2
##  1st Qu.: 7.2222   1st Qu.:179.0   1st Qu.:13.80   1st Qu.:1011.5
##  Median :10.0000   Median :225.0   Median :18.20   Median :1016.7
##  Mean   :10.8815   Mean   :214.9   Mean   :19.45   Mean   :1016.4
##  3rd Qu.:13.8889   3rd Qu.:309.0   3rd Qu.:24.18   3rd Qu.:1021.6
##  Max.   :32.7778   Max.   :360.0   Max.   :55.30   Max.   :1036.2
##
##   wdir_imp        wspd_imp        pres_imp
##  Mode :logical   Mode :logical   Mode :logical
##  FALSE:1602      FALSE:1613      FALSE:1583
##  TRUE :12        TRUE :1         TRUE :31
##
##
##
##
```

```r
# Fit the GAM model incorporating wind direction (wdir), wind speed (wspd), and pressure (pres)
gam_model_with_new_features <- gam(daily_total_accidents ~ s(TMAX, k=5) + s(Temp_Variability, k=5) +
                                    s(wdir, k=5) + s(wspd, k=5) + s(pres, k=5),
                                    family = nb(), data = combined_data)

# Display the summary of the model
summary(gam_model_with_new_features)
```

```
##
## Family: Negative Binomial(14.891)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX, k = 5) + s(Temp_Variability,
##     k = 5) + s(wdir, k = 5) + s(wspd, k = 5) + s(pres, k = 5)
##
```

```
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.633110   0.009284   283.6   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                      edf Ref.df Chi.sq  p-value
## s(TMAX)             2.256  2.771  4.830 0.152843
## s(Temp_Variability) 3.519  3.858 21.715 0.000124 ***
## s(wdir)             1.002  1.003  0.196 0.659280
## s(wspd)             1.002  1.003  0.001 0.988988
## s(pres)             1.338  1.608  0.085 0.908722
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.0249   Deviance explained =  2.8%
## -REML = 4940.1  Scale est. = 1         n = 1614
```

```r
AIC(gam_model_with_new_features)
```

```
## [1] 9846.754
```

Addint extra features wind direction, wind speed and pressure increase AIC from 9840.524 to 9846.754, hence the do not appear to be strong predictors individually (high p-values).

Now, additional factors for day of the week and month is added into consideration

```r
# Fit a GAM model with EHF and Temp_Variability
gam_model_l<- gam(daily_total_accidents ~ s(TMAX, k=5) + s(Temp_Variability, k=5) +
        factor(day_of_week) +
                          factor(month),
                     family = nb(), data = combined_data)

# Summary of the new model
summary(gam_model_l)
```

```
##
## Family: Negative Binomial(24.656)
## Link function: log
##
## Formula:
## daily_total_accidents ~ s(TMAX, k = 5) + s(Temp_Variability,
##     k = 5) + factor(day_of_week) + factor(month)
##
## Parametric coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    2.744011   0.021437 128.004  < 2e-16 ***
## factor(day_of_week)Monday    -0.164540   0.031145  -5.283 1.27e-07 ***
## factor(day_of_week)Saturday  -0.237394   0.031438  -7.551 4.31e-14 ***
## factor(day_of_week)Sunday    -0.423058   0.032610 -12.973  < 2e-16 ***
## factor(day_of_week)Thursday  -0.002033   0.030205  -0.067 0.946342
## factor(day_of_week)Tuesday   -0.027082   0.030357  -0.892 0.372323
## factor(day_of_week)Wednesday -0.013473   0.030362  -0.444 0.657223
## factor(month).L               0.095621   0.032732   2.921 0.003485 **
## factor(month).Q              -0.101035   0.047906  -2.109 0.034941 *
## factor(month).C               0.015370   0.031620   0.486 0.626896
## factor(month)^4              -0.121054   0.033659  -3.597 0.000322 ***
## factor(month)^5               0.089954   0.029389   3.061 0.002207 **
## factor(month)^6              -0.168648   0.030021  -5.618 1.94e-08 ***
## factor(month)^7               0.012337   0.028681   0.430 0.667087
## factor(month)^8               0.018955   0.028770   0.659 0.509999
## factor(month)^9               0.013249   0.029243   0.453 0.650500
## factor(month)^10              0.083489   0.029087   2.870 0.004101 **
## factor(month)^11             -0.046253   0.029121  -1.588 0.112219
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Approximate significance of smooth terms:
##                        edf Ref.df Chi.sq  p-value
## s(TMAX)              2.425  2.978  6.124 0.100045
## s(Temp_Variability) 3.563  3.880 21.077 0.000163 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## R-sq.(adj) =  0.201   Deviance explained = 21.1%
## -REML = 4800.7  Scale est. = 1         n = 1614
```

```r
AIC(gam_model_1)
```

```
## [1] 9527.408
```

Result: Temperature Variability remains highly significant (p = 0.000163), indicating that fluctuations in temperature during the day are a strong predictor of road accidents.

TMAX is also significant in some models, but its predictive power is less strong than Temperature Variability.

Day of the week and month are strong predictors of road traffic accidents. For instance, accident rates are generally higher on weekends, and the months reflect seasonality in driving conditions.

Wind direction, wind speed, and pressure do not appear to be strong predictors individually (high p-values), and after removing these predictors, the result improve.

The deviance explained by the model is now 21.1%, a major improvement compared to previous models where the deviance explained was around 2.74%. This shows that the added features, especially seasonal and day-of-week factors, are helping the model better explain road traffic accidents.

The AIC of 9527.408 is substantially lower than the earlier models, which had AIC values above 9840. The lower AIC suggests that the model with the temperature variability and time-based factors provides a significantly better fit to the data.

Q8: Reflection Q8.1. Additional Data to Improve the Model In the analysis of road accident data, there are several additional dataset that could improve the model. Firstly, Traffic volume data, which measures the number of vehicles on the road would provide crucial context for accident rates. Higher traffic volumes, especially during peak hours, are often associated with increased accident risks. Including traffic data can help the model better capture congestion-related accidents.

Another useful data source could be road surface conditions, as these conditions significantly affect vehicle control and accident likelihood. For example, wet or icy roads are known to increase the risk of accidents, and incorporating real-time road condition data would enhance the model's ability to predict accidents.

Finally, driver behaviour could offer insights such as driving without enough hour of sleep, playing phone while driving, speeding, reckless driving, or distracted driving, could offer valuable insights into accident causation. For example, speeding is one of the leading causes of accidents, and combining driver behaviour data with weather and traffic data would allow the model to predict high-risk situations more accurately.

By integrating these additional data sources, the model would have a more comprehensive understanding of the various factors influencing road traffic accidents, ultimately leading to improved predictive accuracy and more informed decision-making.

Q8.2. Have the Analyses Answered the Objective? The analysis partially answer the objective of determining the relationship between weather conditions and daily road traffic accidents in Melbourne. After the analysis, the high p-value of precipitation, wind and pressure proved that it is not a good predictor, EHF, maximum temperature (TMAX) were also not strong predictors of daily accident rates, as evidenced by low AIC values and R-squared.

However, the inclusion of temperature variability, day of the week, and month as predictors significantly improved the performance of model, indicating rapid weather changes and regular temporal patterns such as rush hour traffic or weekend travel may have a more substantial effect on accident rates than sustained weather conditions. Days like Sunday, Saturday, and Monday were statistically significant predictors, likely reflecting behavioural patterns.

I can expect that result, since Melbourne is known for its variable weather, often described as having 4 season in one day. While specific weather variables like wind and pressure did not significantly affect accident rates, temperature variability proved important, supporting the idea that quick shifts in weather could impact road safety. Still, the model explained only a portion of the variance in accident rates, indicating that other factors are at play.

In conclusion, while the analysis provided valuable insights into the relationship between weather variability and accident rates, it also highlighted the complex nature of factors influencing road accidents in Melbourne. Future studies should consider a broader range of weather-related and non-weather variables to more comprehensively model and predict road traffic accidents in this unique urban environment.

Q8.3 Ways to address missing value Several method can could be employed to address the issue of missing value in datasets. The most common method is mean substitution, where the mean value of the variable is use to substitute the missing value for that same variable. The advantage is that the method is very convenient and easy to implement and provides a quick way to handle missing data especially for continuous data such as temperature. For normally distributed data, mean is the reasonable estimate (Kang, 2013). However, mean result can lead to loss of variation of data when there are too many missing value in the data. Moreover, this method does not consider time-series characteristics or depend the relationship between variable, It can lead to a reduction in variance and may distort relationships between variables because it does not account for the uncertainty associated with missing values (Little & Rubin, 2002). Hence, it is important to understand the nature of data and hence consider the method to address the issue of missing value.

Another option is to use K-nearest neighbour (KNN). KNN imputation fills in missing values by finding the K most similar data points and using their values to impute the missing data. This method is advantageous because it can preserve the temporal patterns of time series data and relationships between variables in our weather data, and can handle non-linear data effectively (Troyanskaya et al., 2001). However, KNN can be computationally intensive, especially for high dimensional dataset. If the data is sparse or if the nearest neighbours are not representative, the imputed values may still be biased.

For time series specific method, methods like forward or backward filling or more advanced techniques such as Kalman filters are better suited to handle missing data. These methods maintain the temporal structure of weather patterns, which is essential for accurate prediction models. For example, forward filling might be appropriate to impute missing wind speed values during a storm by using the last known observation. More advanced methods, such as Kalman filters, offer a statistical approach to estimate missing values based on the trends in the weather data, accounting for fluctuations and uncertainty in weather conditions (Moritz & Bartz-Beielstein, 2017). These methods ensure that the missing values are filled while maintaining the temporal integrity of weather variables like temperature and precipitation, leading to more accurate models of road traffic accidents.

Q8.4. Tackling Overfitting Overfitting occurs when a model captures not only the underlying patterns in the data but also the noise, leading to poor generalization on new, unseen data. This is particularly problematic when building models with a large number of explanatory variables, as the model may become too complex and fit the specific noise of the training data, rather than the general trends.

One method to address overfitting is to use regularization techniques such as Lasso (L1 regularization) and Ridge (L2 regularization). Regularization techniques like Lasso and Ridge regression, explored by Zou and Hastie (2005), could help reduce model complexity by adding penalty terms to the loss function. These techniques add penalties to the model based on the magnitude of the coefficients, effectively shrinking or eliminating less important predictors, which reduces the model's complexity and helps prevent overfitting.

Another common strategy is to apply cross-validation would provide a robust method for assessing model performance on unseen data (Hastie et al., 2009). Cross-validation such as k-fold cross validation, splits the data into multiple subsets and trains the model on different combinations of the data. This approach ensures that the model performs well across various subsets of the data, providing a more reliable estimate of its ability to generalize. Additionally, feature selection methods, such as those discussed by Guyon and Elisseeff (2003), could be used to identify the most relevant predictors, thereby simplifying the model and reducing the risk of overfitting. By combining these approaches, we could develop a model that balances complexity with predictive power, ensuring its generalizability to new data while maintaining its explanatory capabilities for understanding the factors influencing road traffic accidents.

Reference list

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

Kang, H. (2013). The prevention and handling of the missing data. Korean Journal of Anesthesiology, 64(5), 402–406. https://doi.org/10.4097/kjae.2013.64.5.402

Little, R. J. A., & Rubin, D. B. (2002). Statistical Analysis with Missing Data. John Wiley & Sons, Inc. https://doi.org/10.1002/9781119013563

Moritz, S., & Bartz-Beielstein, T. (2017). imputeTS: Time Series Missing Value Imputation in R. The R Journal, 9(1), 207. https://doi.org/10.32614/rj-2017-009

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. Bioinformatics, 17(6), 520–525. https://doi.org/10.1093/bioinformatics/17.6.520

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x