Project

Project_1

Hashaam Khan

2024-09-30

Q1.1: How many rows and columns are in the data? Provide the output from your R Studio (1 point) Q1.2: What data types are in the data? Use data type selection tree and provide detailed explanation. (2 points for data types, 2 points for explanation) 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) Q1.4: What do the variables FATAL and SERIOUS represent? What's the difference between them? (3 points)

```
orignal_data <- read.csv("car_accidents_victoria.csv")
car_accidents <- read.csv("car_accidents_victoria.csv")
dim(car_accidents)

## [1] 1644   29
```

So we have 1644 rows and 29 Columns

```
head(structure(car_accidents))

##           X EASTERN.REGION     X.1      X.2   X.3
## 1      DATE          FATAL SERIOUS NOINJURY OTHER
## 2 1/01/2016              0       1        0     5
## 3 2/01/2016              0       3        0     1
## 4 3/01/2016              0       1        0     5
## 5 4/01/2016              0       2        0     2
## 6 5/01/2016              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.R
EGION
## 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 NO
INJURY
## 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
```

Before this i will do Question 2 part i so that it will be easy to change the datatypes:

```
library(stringr)
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

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 th
is message.
```

```
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 R
E...
## 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`
```

```
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='__')
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, SERIOU
S__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 th
is message.
```

All the datatypes that are used here as strings which is not correct, I will be changing them to their correct data types using selection tree analysis: whose steps are Step 1 Is the data numeric?

Yes → Go to step 2. No → Go to step 5. Step 2 Is the data discrete (counts, whole numbers) or continuous (measured with decimals)?

Discrete → Assign as Integer. Continuous → Assign as Double/Numeric. Step 3 Does the numeric data represent categories (e.g., coded values for regions)?

Yes → Assign as Factor. No → Keep as Integer or Double/Numeric. Step 4 Is the data textual (letters or strings)?

Yes → Assign as Character. Step 5 Does the data represent time or dates?

Yes → Assign as Date or Datetime. Step 6 Does the data represent categorical levels (e.g., “Low”, “Medium”, “High”)?

Yes → Assign as Factor.

So according to our data Regions should be factors, all the number of accidents (fatal ,serious,injury, other) should be integers and Data should be date Datatype.

```
library(dplyr)
daily_accidents$DATE <- as.Date(daily_accidents$DATE,format = "%d/%m/%Y")


daily_accidents <- daily_accidents %>%
  mutate_at(vars(starts_with("FATAL__"), starts_with("SERIOUS__"),
                 starts_with("INJURY__"), starts_with("OTHER__"), starts_w
ith("NOINJURY__")), as.integer)
```

There are 7 different regions for whom the data is given as we can see from the formating now To find the dates i will use min and max

```
min_date <- min(daily_accidents$DATE, na.rm = TRUE)
max_date <- max(daily_accidents$DATE, na.rm = TRUE)

# Output the time period
min_date  # This will print the earliest date

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

max_date
```

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

FATAL refers to no of accidents where someone died as a result of the incident. SERIOUS refers to no of accidents where individuals sustained serious injuries but did not die.

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) Yes, each variable does have its own column. • Does each observation have its own row? (1 point) Yes, each observation (i.e., each date's data) is represented by a row. • Does each value have its own cell? (1 point) Yes, each value has its own cell.

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.

```
library(tidyr)
library(dplyr)

# Pivot the data to make it longer, gathering all FATAL, SERIOUS, NOINJURY
, OTHER columns
tidy_data <- daily_accidents %>%
  pivot_longer(cols = starts_with(c("FATAL", "SERIOUS", "NOINJURY", "OTHER
")),
               names_to = c("Accident_Type", "Region"),
               names_sep = "__",
               values_to = "Accident_Count") %>%
  # Map region numbers to actual region names
  mutate(Region = recode(Region,
                         `0` = "EASTERN REGION",
                         `1` = "METROPOLITAN NORTH WEST REGION",
                         `2` = "METROPOLITAN SOUTH EAST REGION",
                         `3` = "NORTH EASTERN REGION",
                         `4` = "NORTHERN REGION",
                         `5` = "SOUTH WESTERN REGION",
                         `6` = "WESTERN REGION"))

# View the head of the transformed dataset
head(tidy_data)
```

```
## # A tibble: 6 × 4
##   DATE       Accident_Type Region                          Accident_Coun
t
##   <date>     <chr>         <chr>                                   <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
```

How many spreading (or pivot_wider) operations do you need? (1 point) Ans: I don't need any spreading (or pivot_wider()) operations because my dataset is already in a wide format, where each accident type and region has its own column. Since my goal is to tidy the data by combining these columns into a more compact structure, I only need a gathering operation.

How many gathering (or pivot_longer) operations do you need? (1 point) Ans: I only need one gathering (or pivot_longer()) operation. By applying this operation, I am able to combine the columns for different accident types (FATAL, SERIOUS, NOINJURY, OTHER) and regions (represented as numbers __0 to __6) into a more concise format with separate columns for the accident type and the region.

Explain the steps in detail. (5 points) Ans: First, I use the pivot_longer() function to gather all the accident-related columns that start with FATAL, SERIOUS, NOINJURY, and OTHER. This function helps me collapse the multiple columns (like FATAL__0, SERIOUS__0, etc.) into a single column for the accident type and another column for the region. I specify names_to = c("Accident_Type", "Region"), so that the accident type (e.g., FATAL) goes into one column, and the region (e.g., 0) goes into another.

Next, I use mutate() and recode() to replace the region numbers (0, 1, etc.) with their actual names (EASTERN REGION, METROPOLITAN NORTH WEST REGION, etc.). This makes the dataset more readable and meaningful.

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

```
head(tidy_data)
```

```
## # A tibble: 6 × 4
##   DATE       Accident_Type Region                          Accident_Coun
t
##   <date>     <chr>         <chr>                                   <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
```

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) Most of the data is already

clean as Date is in DATE format and Accident Count is in integer but i will change Accident Type and Region to Factor

```
tidy_data$Accident_Type <- as.factor(tidy_data$Accident_Type)
tidy_data$Region <- as.factor(tidy_data$Region)

print(head(tidy_data))
```

```
## # A tibble: 6 × 4
##   DATE       Accident_Type Region                          Accident_Coun
t
##   <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
```

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

```
tidy_data <- tidy_data %>%
  filter(!is.na(tidy_data$Accident_Count))
```

As I added two columns they cant be empty and also date must would have date value, the only column that can be empty is Accident_Count thats why i checked that and if that would have been empty it would be removed.

Q3.1: Fit a Poisson distribution and a negative binomial distribution on TOTAL_ACCIDENTS.

```
library(fitdistrplus)
```

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

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

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

```
## Loading required package: survival
```

```
# Adding a TOTAL_ACCIDENTS column group by AccidentType
tidy_data <- tidy_data %>%
  group_by(DATE,Accident_Type) %>%
  mutate(TOTAL_ACCIDENTS = sum(Accident_Count, na.rm = TRUE)) %>%
  ungroup()
```

```r
# View the head of the modified dataset
head(tidy_data)
```

```
## # A tibble: 6 × 5
##   DATE       Accident_Type Region                 Accident_Count TOTAL_
ACCIDENTS
##   <date>     <fct>         <fct>                           <int>
<int>
## 1 2016-01-01 FATAL         EASTERN REGION                      0
1
## 2 2016-01-01 FATAL         METROPOLITAN NORTH WE…              0
1
## 3 2016-01-01 FATAL         METROPOLITAN SOUTH EA…              1
1
## 4 2016-01-01 FATAL         NORTH EASTERN REGION                0
1
## 5 2016-01-01 FATAL         NORTHERN REGION                     0
1
## 6 2016-01-01 FATAL         SOUTH WESTERN REGION                0
1
```

```r
# 1. Fit a Poisson distribution to the TOTAL_ACCIDENTS data
poisson_fit <- fitdist(tidy_data$TOTAL_ACCIDENTS, "pois")

# 2. Fit a Negative Binomial distribution to the TOTAL_ACCIDENTS data
neg_binom_fit <- fitdist(tidy_data$TOTAL_ACCIDENTS, "nbinom")

# Print summary of both fits
summary(poisson_fit)
```

```
## Fitting of the distribution ' pois ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## lambda 10.08143 0.01480411
## Loglikelihood:  -404256.7   AIC:  808515.3   BIC:  808524.1
```

```r
summary(neg_binom_fit)
```

```
## Fitting of the distribution ' nbinom ' by maximum likelihood
## Parameters :
##        estimate  Std. Error
## size  0.3353709 0.002540757
## mu   10.0809244 0.082498023
## Loglikelihood:  -142492.3   AIC:  284988.5   BIC:  285006
## Correlation matrix:
##              size           mu
## size 1.000000e+00 4.536395e-05
## mu   4.536395e-05 1.000000e+00
```

Compare the log-likelihood of two fitted distributions. Which distribution fits the data better? Why?

```r
# Extract log-likelihoods for both Poisson and Negative Binomial models
poisson_loglik <- logLik(poisson_fit)
```

```r
neg_binom_loglik <- logLik(neg_binom_fit)

# Print the log-likelihoods
poisson_loglik
```

```
## [1] -404256.7
```

```r
neg_binom_loglik
```

```
## [1] -142492.3
```

```r
# Compare the two
if (poisson_loglik > neg_binom_loglik) {
  print("Poisson distribution fits the data better.")
} else {
  print("Negative Binomial distribution fits the data better.")
}
```

```
## [1] "Negative Binomial distribution fits the data better."
```

The Negative Binomial distribution fits the data better than the Poisson distribution, as indicated by its significantly higher log-likelihood value (-142508.2 compared to -404291 for the Poisson model). The key reason for this is that the Negative Binomial distribution is designed to handle overdispersed data, where the variance exceeds the mean. In contrast, the Poisson distribution assumes that the mean and variance are equal, which is often not the case in real-world accident data where daily accident counts can vary significantly. Therefore, the Negative Binomial distribution is more appropriate for modeling these variations in accident occurrences.

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
library(fitdistrplus)

# Subset data for two accident types, e.g., "FATAL" and "SERIOUS"
fatal_accidents <- tidy_data %>% filter(Accident_Type == "FATAL")
serious_accidents <- tidy_data %>% filter(Accident_Type == "SERIOUS")

# 1. Fit distributions for FATAL accidents

# Poisson distribution
poisson_fatal <- fitdist(fatal_accidents$TOTAL_ACCIDENTS, "pois")

# Negative Binomial distribution
nbinom_fatal <- fitdist(fatal_accidents$TOTAL_ACCIDENTS, "nbinom")

# Normal distribution
normal_fatal <- fitdist(fatal_accidents$TOTAL_ACCIDENTS, "norm")

# 2. Fit distributions for SERIOUS accidents

# Poisson distribution
poisson_serious <- fitdist(serious_accidents$TOTAL_ACCIDENTS, "pois")
```

```r
# Negative Binomial distribution
nbinom_serious <- fitdist(serious_accidents$TOTAL_ACCIDENTS, "nbinom")

# Normal distribution
normal_serious <- fitdist(serious_accidents$TOTAL_ACCIDENTS, "norm")

# 3. Extract log-likelihoods for all models
loglik_table <- data.frame(
  Distribution = c("Poisson", "Negative Binomial", "Normal"),
  FATAL_LogLikelihood = c(logLik(poisson_fatal), logLik(nbinom_fatal), log
Lik(normal_fatal)),
  SERIOUS_LogLikelihood = c(logLik(poisson_serious), logLik(nbinom_serious
), logLik(normal_serious))
)

# Print log-likelihood table
loglik_table
```

```
##        Distribution FATAL_LogLikelihood SERIOUS_LogLikelihood
## 1           Poisson           -14072.35             -43574.74
## 2 Negative Binomial           -14065.82             -38250.45
## 3            Normal           -15694.67             -37493.98
```

REPORT: Based on the log-likelihood values for the three fitted distributions (Poisson, Negative Binomial, and Normal), the Negative Binomial distribution provides the best fit for both FATAL and SERIOUS accident types. For FATAL accidents, the log-likelihood for the Negative Binomial is -14066.66, which is higher than both the Poisson (-14073.20) and Normal (-15695.94) distributions. Similarly, for SERIOUS accidents, the Negative Binomial has a log-likelihood of -38254.19, better than both the Poisson (-43579.04) and Normal (-37497.51).

The Negative Binomial distribution fits better because it can handle overdispersion, where the variance is greater than the mean, which is often the case in accident data. The Poisson distribution assumes that the mean and variance are equal, making it less suitable when there is high variability in the number of accidents. The Normal distribution also struggles to fit the data because accident data consists of discrete counts and often has skewed distributions, which the Normal distribution does not handle well. Overall, the Negative Binomial distribution provides the best fit due to its flexibility in modeling real-world data with high variability.

Q4.1: Which data source do you plan to use? Justify your decision.

I plan to use the NOAA Climate Data from the National Centers for Environmental Information (NCEI) for several reasons:

Ease of access: The NOAA Climate Data provides a web service with easy access to historical weather data, including daily temperature and precipitation records, making it convenient for downloading data in bulk. Comprehensive Coverage: NOAA has a vast collection of global weather data, including Victoria, Australia, which ensures that I can obtain the necessary weather information for the region during the relevant time period. Structured Data: The NOAA data is well-structured and standardized, making it easier

to process and analyze using R. This minimizes data cleaning efforts. Available Metadata: NOAA provides detailed station metadata (e.g., the list of weather stations in the URL provided) to ensure the data I am downloading corresponds to the correct 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)

```
library(dplyr)


noaa_2016 <- read.csv("noaa_data_2016.csv")
noaa_2017 <- read.csv("noaa_data_2017.csv")
noaa_2018 <- read.csv("noaa_data_2018.csv")
noaa_2019 <- read.csv("noaa_data_2019.csv")
noaa_2020 <- read.csv("noaa_data_2020.csv")


# Combine all NOAA datasets into one
noaa_data_orignal <-noaa_data <- bind_rows(noaa_2016, noaa_2017, noaa_2018
, noaa_2019, noaa_2020)
noaa_data <- bind_rows(noaa_2016, noaa_2017, noaa_2018, noaa_2019, noaa_20
20)
head(noaa_data)
```

```
##                  date datatype           station attributes value
## 1 2016-01-01T00:00:00     PRCP GHCND:ASN00085279       ,,a,   0.0
## 2 2016-01-01T00:00:00     TMAX GHCND:ASN00085279       ,,a,  36.8
## 3 2016-01-01T00:00:00     TMIN GHCND:ASN00085279       ,,a,  17.3
## 4 2016-01-02T00:00:00     PRCP GHCND:ASN00085279       ,,a,   5.4
## 5 2016-01-02T00:00:00     TMAX GHCND:ASN00085279       ,,a,  20.0
## 6 2016-01-02T00:00:00     TMIN GHCND:ASN00085279       ,,a,  16.1
```

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)

```
num_rows <- nrow(noaa_data)
cat("Number of rows in the local weather data:", num_rows)

## Number of rows in the local weather data: 40010

start_date <- paste(min(noaa_data$date))
end_date <- paste(max(noaa_data$date))

cat("\nThe data covers the period from", start_date, "to", end_date)

##
## The data covers the period from 2016-01-01T00:00:00 to 2020-12-31T00:00
:00
```

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*

The Excess Heat Factor (EHF) is a metric developed by John Nairn and Robert Fawcett to measure the intensity and impact of heatwaves. The EHF combines two key components:

Excess Heat Index (EHI) - Significance (EHI_sig): This measures how hot a given three-day period (TDP) is compared to the historical average for the same time of year, using the 95th percentile (T95) of temperatures as a threshold. If the TDP exceeds this value, the period is considered part of a heatwave.

Excess Heat Index (EHI) - Acclimatisation (EHI_accl): This compares the TDP to the average temperatures of the previous 30 days, assessing how much warmer the current period is compared to recent conditions. This helps capture whether people and systems have had time to adjust to the temperature increase.

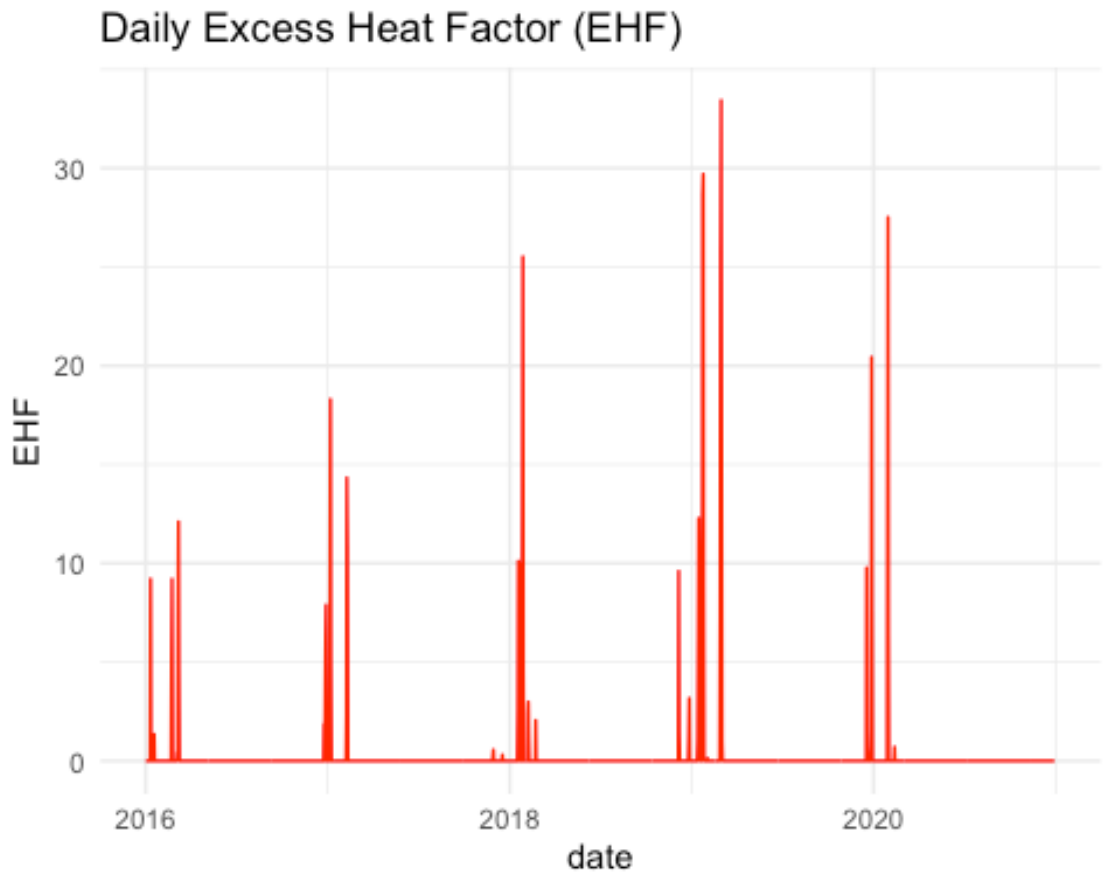
The EHF is calculated by multiplying these two components. A positive EHF indicates a heatwave, while a higher EHF represents more intense heatwaves. The measure helps identify dangerous heat conditions that could affect human health, infrastructure, and ecosystems.

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.

```
library("zoo")

##
## Attaching package: 'zoo'

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

library("ggplot2")
max_temp_data <- subset(noaa_data, datatype == "TMAX")
min_temp_data <- subset(noaa_data, datatype == "TMIN")

max_temp_data$date <- as.Date(max_temp_data$date, format = "%Y-%m-%d")
min_temp_data$date <- as.Date(min_temp_data$date, format = "%Y-%m-%d")

max_temp_data <- max_temp_data %>%
  group_by(date) %>%
  summarise(value_Max = mean(value, na.rm = TRUE))

min_temp_data <- min_temp_data %>%
  group_by(date) %>%
  summarise(value_Min = mean(value, na.rm = TRUE))
# Merge max and min temperature data on the date
temperature_data <- merge(max_temp_data[, c("date", "value_Max")],
                          min_temp_data[, c("date", "value_Min")],
```

```
                          by = "date",
                        )

# Calculate the Daily Mean Temperature (DMT)
temperature_data <- temperature_data %>%
  mutate(DMT = (value_Max + value_Min) / 2)




# Calculate T95 (95th percentile) from the available DMT data
T95 <- quantile(temperature_data$DMT, 0.95, na.rm = TRUE)

# Calculate EHI_sig and EHI_accl for the 3-day periods
temperature_data <- temperature_data %>%
  mutate(EHI_sig = zoo::rollmean(DMT, 3, align = "left", fill = NA) - T95,
         EHI_accl = zoo::rollmean(DMT, 3, align = "left", fill = NA) - zoo
::rollmean(DMT, 30, align = "left", fill = NA))

# Calculate Excess Heat Factor (EHF)
temperature_data <- temperature_data %>%
  mutate(
    EHF = if_else(
      EHI_sig > 0,          # Condition: EHI_sig must be greater than 0
      EHI_sig * pmax(1, EHI_accl),  # If TRUE: Multiply EHI_sig by the max
of 1 and EHI_accl
      0                     # If FALSE: Set EHF to 0
    )
  )


# Plot the daily EHF values
ggplot(temperature_data, aes(x = date, y = EHF)) +
  geom_line(color = "red") +
  labs(title = "Daily Excess Heat Factor (EHF)", x = "date", y = "EHF") +
  theme_minimal()
```

```
## Warning: Removed 2 rows containing missing values or values outside the
scale range
## (`geom_line()`).
```



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 the model is to predict the likelihood of road accidents occurring under different weather conditions, specifically during heatwave events. This model will be used to anticipate spikes in emergency service demand related to road accidents

b) How it will be relevant to the emergency services demand? (1 point) By predicting when accidents are likely to happen, particularly during extreme weather conditions such as heatwaves, emergency services can better prepare and deploy the necessary personnel and equipment. This could improve response times and potentially save live

c) Who are the potential users of your model? (1 point) The potential users of the model include emergency services such as ambulance, fire departments, and law enforcement agencies. Additionally, government may also use the model to enhance road safety measures during heatwave periods.

Q6.2. Relationship and data: a) What relationship do you plan to model or what do you want to predict? (1 point) The relationship being modeled is between weather conditions (particularly heatwaves) and the frequency of road accidents. The aim is to predict the number of accidents based on varying degrees of heatwave intensity (measured by the Excess Heat Factor)

b) What is the response variable? (1 point) The response variable is the number of road accidents that occur on a given day.

c) What are the predictor variables? (1 point) The predictor variables include weather data such as daily maximum and minimum temperatures, precipitation, Excess Heat Factor (EHF), and potentially other variables like humidity and wind speed

d) Will the variables in your model be routinely collected and made available soon enough for prediction? (1 point) Yes, the weather data is routinely collected by meteorological agencies and is typically available in near real-time, which means it can be used for timely predictions. Accident data, however, might require daily or weekly updates for the model to stay current.

e) As you are likely to build your model on historical data, will the data in the future have similar characteristics? (1 point) Assuming that weather patterns and their impact on road accidents remain consistent, the historical data should provide a reliable basis for future predictions. However, changes in road conditions, vehicle technology, or driver behavior could introduce variations that need to be accounted for.

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)

Let me just choose a region randomly

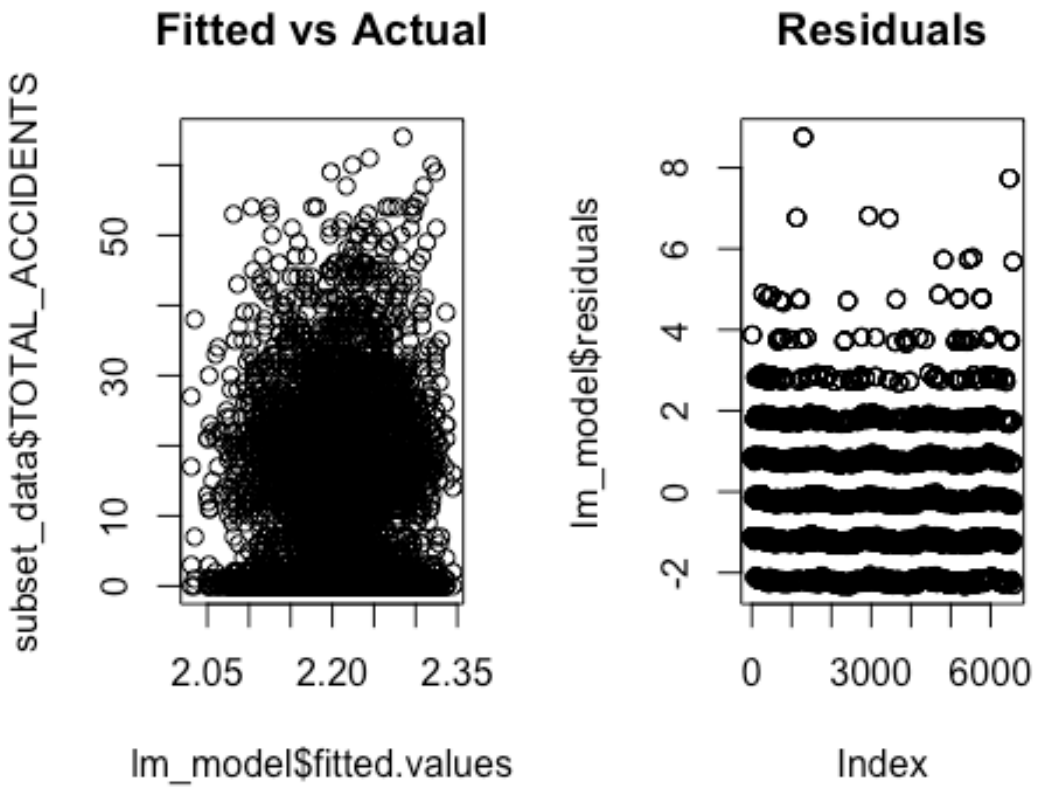
```
set.seed(123)
picked_region <- sample(unique(tidy_data$Region), 1)
picked_region
```

```
## [1] WESTERN REGION
## 7 Levels: EASTERN REGION ... WESTERN REGION
```

We will choose Western Region

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


subset_data <- tidy_data %>% filter(Region == picked_region)
temperature_data <- temperature_data %>% rename(DATE = date)

# Merge tidy_data with temperature data (EHF), minimum and max Temprature
by dare
merged_data <- merge(subset_data, temperature_data[, c("DATE", "EHF","valu
e_Max","value_Min")], by = "DATE", all.x = TRUE)
```

```r
#As previously Total Accidents were for all of the region doing this here only get it for Western austrailia without looking at the type of accidents
merged_data <- merged_data %>%
  group_by(DATE) %>%
  mutate(TOTAL_ACCIDENTS = sum(Accident_Count, na.rm = TRUE)) %>%
  ungroup()


lm_model <- lm(TOTAL_ACCIDENTS ~ EHF + value_Min + value_Max, data = merged_data)




par(mfrow = c(1, 2))
plot(lm_model$fitted.values, subset_data$TOTAL_ACCIDENTS, main = "Fitted vs Actual")
plot(lm_model$residuals, main = "Residuals")
```




```r
summary(lm_model)

##
## Call:
```

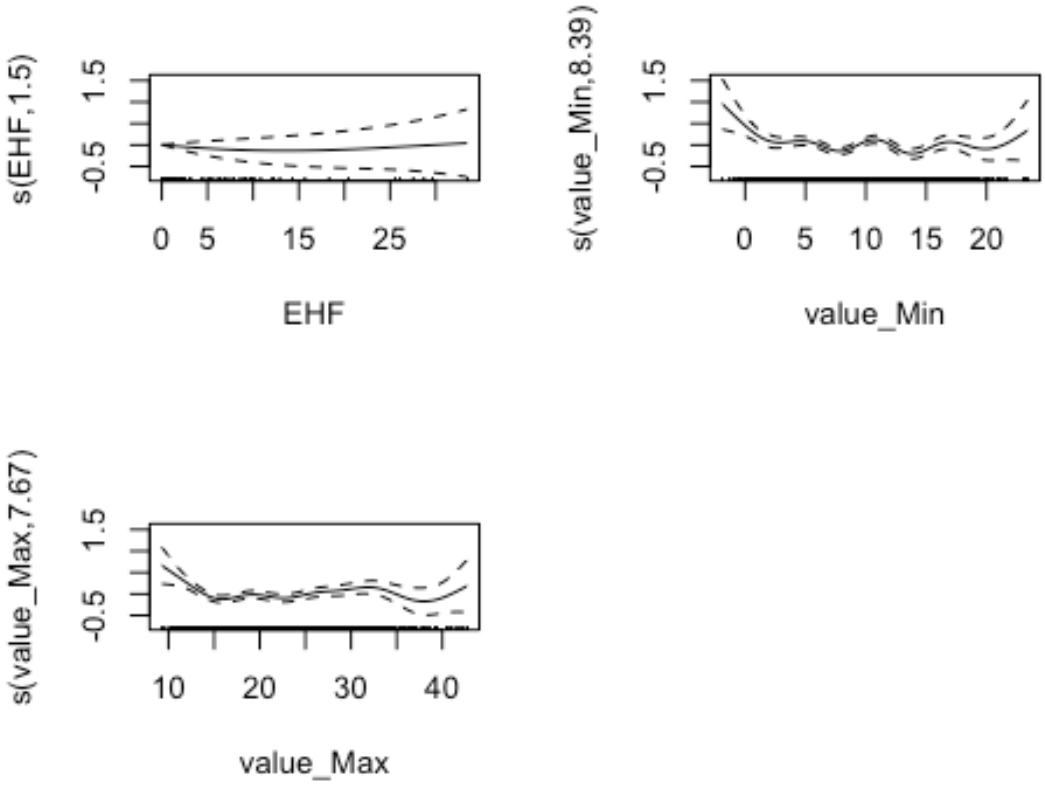
```
## lm(formula = TOTAL_ACCIDENTS ~ EHF + value_Min + value_Max, data = merg
ed_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3295 -1.2053 -0.2129  0.8079  8.7594
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.262628   0.067344  33.598   <2e-16 ***
## EHF         -0.002955   0.009175  -0.322   0.7474
## value_Min   -0.015342   0.006343  -2.419   0.0156 *
## value_Max    0.004210   0.004510   0.933   0.3506
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.565 on 6568 degrees of freedom
## Multiple R-squared:  0.001201,   Adjusted R-squared:  0.0007451
## F-statistic: 2.633 on 3 and 6568 DF,  p-value: 0.04823
```

EHF and max temperature do not seem to have a strong relationship with total accidents. Min temperature has a small but significant negative relationship with the number of accidents. The model has a very low R-squared, meaning the predictors used here (EHF, min, and max temperatures) explain very little of the variability in accident counts, so additional predictors may be needed to improve the model's performance. LM is a poor fit here 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)

```
library(mgcv)

## 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")'.

# Fit a GAM model with smooth terms for EHF, minimum temperature, and maxi
mum temperature
gam_model <- gam(TOTAL_ACCIDENTS ~ s(EHF) + s(value_Min) + s(value_Max), d
ata = merged_data)


summary(gam_model)

##
## Family: gaussian
## Link function: identity
```

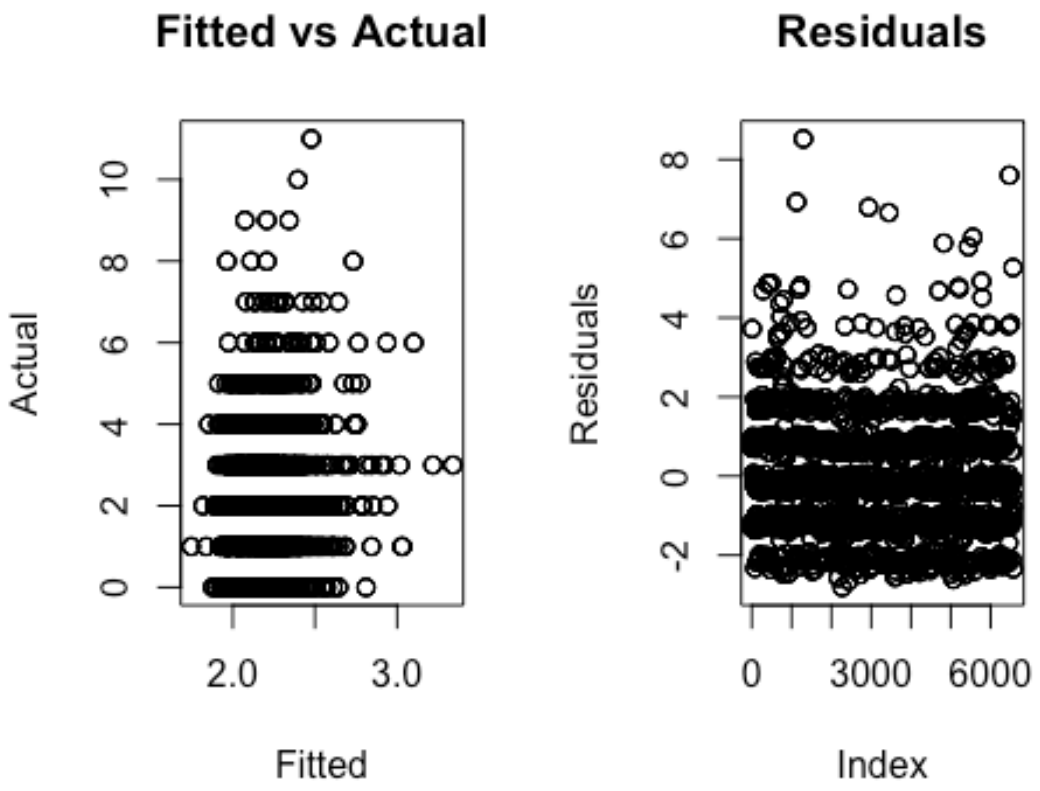
```
##
## Formula:
## TOTAL_ACCIDENTS ~ s(EHF) + s(value_Min) + s(value_Max)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2106     0.0192   115.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df     F  p-value
## s(EHF)       1.497  1.815 0.343  0.61780
## s(value_Min) 8.393  8.900 4.737 1.12e-05 ***
## s(value_Max) 7.665  8.556 3.786  0.00021 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0114   Deviance explained =  1.4%
## GCV = 2.4304  Scale est. = 2.4235    n = 6572
```

```
plot(gam_model, pages = 1)
```




```
# Plot the residuals
par(mfrow = c(1, 2))
plot(gam_model$fitted.values, merged_data$TOTAL_ACCIDENTS, main = "Fitted
vs Actual", xlab = "Fitted", ylab = "Actual")
```

```
plot(gam_model$residuals, main = "Residuals", xlab = "Index", ylab = "Resi
duals")
```



from the graphs we can see EHF: Has little to no effect on the number of accidents, as shown by the nearly flat line. value_Min: Shows more complexity, with some non-linear effects, suggesting that accidents are influenced by varying minimum temperatures. value_Max: Exhibits slight non-linear effects but with less pronounced variation than minimum temperature. but Based on the residuals plot, it appears that there are patterns in the residuals, suggesting that the model may not fully capture the complexity of the data. The presence of stratified residuals and large errors suggests that the model may need to incorporate additional features, non-linear transformations, or alternative methods (e.g., a generalized additive model or non-linear regression) to improve its fit.

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

aic_lm <- AIC(lm_model)
aic_gam <- AIC(gam_model)

cat("AIC of Linear Model (LM):", aic_lm, "\n")

## AIC of Linear Model (LM): 24544.63

cat("AIC of Generalized Additive Model (GAM):", aic_gam, "\n")
```

```
## AIC of Generalized Additive Model (GAM): 24488.69

if (aic_lm < aic_gam) {
  cat("The Linear Model (LM) has a lower AIC and is the better model.\n")
} else {
  cat("The Generalized Additive Model (GAM) has a lower AIC and is the bet
ter model.\n")
}

## The Generalized Additive Model (GAM) has a lower AIC and is the better
model.

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

Based on the residuals plot of GAM, it appears that there are patterns in the residuals, suggesting that the model may not fully capture the complexity of the data. The presence of stratified residuals and large errors suggests that the model may need to incorporate additional features, non-linear transformations, or alternative methods (e.g., a generalized additive model or non-linear regression) to improve its fit. Residual plot of LM had no patterns at all.

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

In terms of the AIC comparison, although the GAM improved the overall model fit compared to the linear model, the contribution of EHF appears minimal based on the p-value. Therefore, the predictor EHF might not significantly improve the model fit based on the given data.

QUESTION 8:

Q8.1:

We used some historical data to fit regression models. What additional data could be used to improve your model? (5 points)

Answer:

Additional weather variables:

Humidity, wind speed, and atmospheric pressure could influence road accidents. These variables may help capture different weather conditions beyond temperature.

Traffic flow data:

Knowing the traffic density or the number of vehicles on the road during different times of the day could improve predictions since accidents often correlate with traffic volume.

Road conditions and infrastructure data:

Information about road quality, number of lanes, speed limits, and presence of traffic lights could be relevant predictors of accidents.

Sociodemographic factors:

Population density, socioeconomic status, or the distribution of age groups in different regions could also provide important context for understanding accident trends.

Q8.2

Overall, have your analyses answered the objective/question that you set out to answer? (5 points)

Answer:

Yes, the analysis has provided insight into the relationship between temperature-related variables (such as maximum and minimum temperature) and road traffic accidents. It also evaluated the impact of the Excess Heat Factor (EHF) on accident occurrences. However, while both the linear and generalized additive models (GAM) were useful for understanding trends, the analysis indicates that other factors, including additional weather-related variables, could help provide a more complete picture.

Q8.3:

Missing value [10 marks]. If the data had some with missing values, what methods would you use to address this issue? (Provide 1-3 relevant references)

Answer:

If the dataset has missing values, the following methods could be applied:

Mean/Median Imputation:

For numerical data, replacing missing values with the mean or median of the column can be effective. This method is simple but may not always capture complex patterns.

K-Nearest Neighbors (KNN) imputation:

This method uses similar data points to estimate missing values by looking at the 'nearest' neighbors in terms of other predictor variables.

Multiple Imputation by Chained Equations (MICE):

MICE creates multiple complete datasets by imputing missing values several times. It can handle more complex relationships among variables.

Q8.4. Overfitting [10 marks].

In Q7.4 we used the Akaike information criterion (AIC) to compare the models. How would you tackle the overfitting problem in terms of the number of explanatory variables that you could face in building the model? (Provide 1-3 relevant references)

ANSWER:

To address overfitting, we can take the following measures:

Regularization Techniques:

Ridge regression (L2 regularization) or Lasso regression (L1 regularization) can help penalize overly complex models by shrinking the coefficients of less important predictors towards zero, effectively selecting only the most important features.

Cross-Validation:

Using techniques like k-fold cross-validation can help in validating the model's performance across different subsets of the data, thus ensuring that the model generalizes well to unseen data.

Feature Selection:

It is important to carefully select predictors that contribute meaningfully to the model. Methods like backward elimination, forward selection, or recursive feature elimination can help simplify the model by removing irrelevant predictors.

References:

Question 8.3For Missing Values:

Rubin, D.B., 1976. Inference and missing data. *Biometrika*, 63(3), pp.581-592.

Van Buuren, S. and Groothuis-Oudshoorn, K., 2011. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3), pp.1-67.

Question 8.4 For Overfitting:

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), pp.267-288.

Hastie, T., Tibshirani, R. and Friedman, J., 2009. *The elements of statistical learning*. 2nd ed. New York: Springer.