Assignment

SolomonSilwal

2024-06-27

Task: Loading the data set from “accidents.csv”

```
# Load the readr package if not already loaded
library(readr)
library(ggplot2)
library(dplyr)
```

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

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

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

```
library(purrr)
library(tidyr)

# Load the data set from "accidents.csv"
df <- read_csv("accidents.csv")
```

```
## Rows: 2069 Columns: 14
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (5): Accident Date, 1st Road Class, Road Surface, Daylight/Dark, Local A...
## dbl (9): Number of Vehicles, Time (24hr), Lighting Conditions, Weather Condi...
##
## 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.
```

```
df
```

```
## # A tibble: 2,069 × 14
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <chr>                   <dbl> <chr>
##  1                    2 01-01-2017               2120 U
##  2                    2 04-01-2017               1500 U
##  3                    2 05-01-2017                732 A58
##  4                    2 05-01-2017                930 A646
##  5                    2 14-01-2017                909 U
##  6                    1 15-01-2017               1659 U
##  7                    1 16-01-2017               1059 A646
##  8                    3 19-01-2017               1849 B6138
##  9                    1 20-01-2017               1408 U
## 10                    1 22-01-2017               1325 A58
## # ℹ 2,059 more rows
## # ℹ 10 more variables: `Road Surface` <chr>, `Lighting Conditions` <dbl>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <dbl>, `Local Authority` <chr>,
## #   `Type of Vehicle` <dbl>, `Casualty Class` <dbl>, `Casualty Severity` <dbl>,
## #   `Sex of Casualty` <dbl>, `Age of Casualty` <dbl>
```

Task: Using dplyr and lubridate to handle the date and time columns. The Accident Date column is converted using dmy and the Time (24hr)

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
df_clean <- df
df_clean$`Accident Date` <- dmy(df_clean$`Accident Date`)
df_clean$`Time (24hr)` <- format(as.POSIXct(sprintf("%04d", df_clean$`Time (24hr)`),
                                            format = "%H%M", tz = "UTC"),
                                 format = "%H:%M", usetz = FALSE)
df_clean
```

```
## # A tibble: 2,069 × 14
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <chr>         <chr>
##  1                    2 2017-01-01      21:20         U
##  2                    2 2017-01-04      15:00         U
##  3                    2 2017-01-05      07:32         A58
##  4                    2 2017-01-05      09:30         A646
##  5                    2 2017-01-14      09:09         U
##  6                    1 2017-01-15      16:59         U
##  7                    1 2017-01-16      10:59         A646
##  8                    3 2017-01-19      18:49         B6138
##  9                    1 2017-01-20      14:08         U
## 10                    1 2017-01-22      13:25         A58
## # ℹ 2,059 more rows
## # ℹ 10 more variables: `Road Surface` <chr>, `Lighting Conditions` <dbl>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <dbl>, `Local Authority` <chr>,
## #   `Type of Vehicle` <dbl>, `Casualty Class` <dbl>, `Casualty Severity` <dbl>,
## #   `Sex of Casualty` <dbl>, `Age of Casualty` <dbl>
```

Task: calculating missing values and visualized them for the Daylight/Dark column.

```
missing_values <- colSums(is.na(df_clean))
print(missing_values)
```

```
##  Number of Vehicles       Accident Date         Time (24hr)      1st Road Class
##                   0                   0                   0                   0
##        Road Surface Lighting Conditions       Daylight/Dark  Weather Conditions
##                   0                   0                  18                   0
##     Local Authority     Type of Vehicle      Casualty Class   Casualty Severity
##                   0                   0                   0                   0
##     Sex of Casualty     Age of Casualty
##                   0                  19
```

Task : Ploting missing values of Daylight/Dark

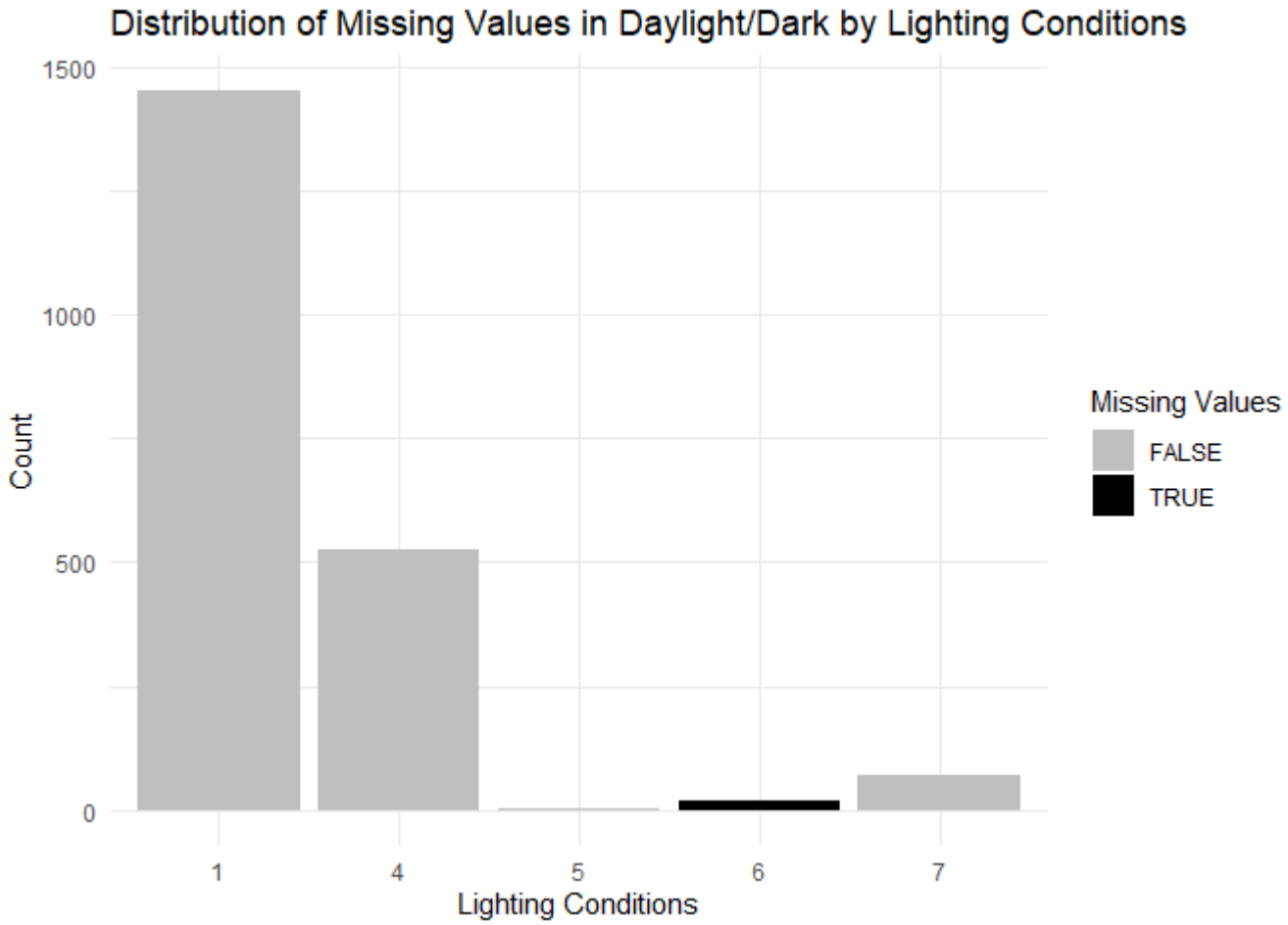
```
missing_values <- colSums(is.na(df_clean))
print("Columns with Missing Values:",)
```

```
## [1] "Columns with Missing Values:"
```

```
print(missing_values[missing_values > 0])
```

```
##   Daylight/Dark Age of Casualty
##              18              19
```

```
ggplot(df_clean, aes(x = factor(`Lighting Conditions`), fill = factor(is.na(`Daylight/Dark
`)))) +
  geom_bar(position = "stack") +
  labs(title = "Distribution of Missing Values in Daylight/Dark by Lighting Conditions",
       x = "Lighting Conditions",
       y = "Count") +
  scale_fill_manual(values = c("FALSE" = "gray", "TRUE" = "black"), name = "Missing Values")
+
  theme_minimal()
```




```
missing_day <- df_clean %>% filter(is.na(`Daylight/Dark`))
```

Task:Mapping and fixing anomalies in the “Road Surface” and “1st Road Class” columns.

```
first_road_class_mapping <- c("1" = "Motorway","2" = "A(M)","3" = "A","4" = "B","5" = "C","6"
= "Unclassified")

df_clean <- df_clean %>%
  mutate(`1st Road Class` = map_chr(`1st Road Class`, function(x) {
    ifelse(as.character(x) %in% names(first_road_class_mapping), first_road_class_mapping[as.
character(x)], as.character(x))
  }))

# Further consistency adjustments
df_clean <- df_clean %>%
  mutate(
    `1st Road Class` = case_when(
      `1st Road Class` %in% c("U") ~ "Unclassified",
      grepl("^A\\d+", `1st Road Class`) ~ "A",
      grepl("^A\\(M\\)$", `1st Road Class`) ~ "A(M)",
      grepl("^A\\d+\\(M\\)$", `1st Road Class`) ~ "A(M)",
      grepl("^B\\d+", `1st Road Class`) ~ "B",
      `1st Road Class` %in% c("Motorway", "A", "B", "C", "Unclassified", "A(M)") ~ `1st Road
Class`,
      `1st Road Class` %in% c("M62") ~ "Motorway",
      TRUE ~ `1st Road Class`
    )
  )

unique(df_clean$`1st Road Class`)
```

```
## [1] "Unclassified" "A"            "B"            "Motorway"     "A(M)"
```

Example: Missing at Random (MAR): Data is missing at random if the probability of missingness can be explained by observed variables in the dataset, such as age influencing the likelihood of reporting income.

Missing not at Random (MNAR): Data is missing not at random if the probability of missingness is related to unobserved data, such as severity of side effects influencing the likelihood of reporting symptoms in a clinical trial.

Task:

```
unique(df_clean$`Road Surface`)
```

```
##  [1] "Wet/Damp"      "Dry"           "Frost/Ice"     "Ice"
##  [5] "Snow"          "Wet"           "Wet \xa8 Damp" "2"
##  [9] "1"             "3"             "4"             "5"
```

Task:Transforming and standardizing values in the “Road Surface” column.

```
df_clean <- df_clean %>%
  mutate(
    `Road Surface` = case_when(
      `Road Surface` %in% c("Wet/Damp", "Wet") ~ "Wet / Damp",
      `Road Surface` %in% c("Frost/Ice", "Ice") ~ "Frost / Ice",
      `Road Surface` %in% c("1") ~ "Dry",
      `Road Surface` %in% c("2") ~ "Wet / Damp",
      `Road Surface` %in% c("3") ~ "Snow",
      `Road Surface` %in% c("4") ~ "Frost / Ice",
      `Road Surface` %in% c("5") ~ "Flood (surface water over 3cm deep)",
      `Road Surface` %in% c("Wet \xa8 Damp") ~ "Wet / Damp",
      TRUE ~ `Road Surface`
    )
  )

unique(df_clean$`Road Surface`)
```

```
## [1] "Wet / Damp"                          "Dry"
## [3] "Frost / Ice"                         "Snow"
## [5] "Flood (surface water over 3cm deep)"
```

```
unique(df_clean$`Lighting Conditions`)
```

```
## [1] 4 1 5 7 6
```

Task:Transforming values in the “Lighting Conditions” column to descriptive categories.

```
# Transform Lighting Conditions using case_when
df_clean <- df_clean %>%
  mutate(
    `Lighting Conditions` = case_when(
      `Lighting Conditions` == "1" ~ "Daylight: street lights present",
      `Lighting Conditions` == "2" ~ "Daylight: no street lighting",
      `Lighting Conditions` == "3" ~ "Daylight: street lighting unknown",
      `Lighting Conditions` == "4" ~ "Darkness: street lights present and lit",
      `Lighting Conditions` == "5" ~ "Darkness: street lights present but unlit",
      `Lighting Conditions` == "6" ~ "Darkness: no street lighting",
      `Lighting Conditions` == "7" ~ "Darkness: street lighting unknown",
      TRUE ~ as.character(`Lighting Conditions`)
    )
  )

# Display unique values after transformation
unique(df_clean$`Lighting Conditions`)
```

```
## [1] "Darkness: street lights present and lit"
## [2] "Daylight: street lights present"
## [3] "Darkness: street lights present but unlit"
## [4] "Darkness: street lighting unknown"
## [5] "Darkness: no street lighting"
```

```
df_clean
```

```
## # A tibble: 2,069 × 14
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <chr>         <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 10 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <dbl>, `Local Authority` <chr>,
## #   `Type of Vehicle` <dbl>, `Casualty Class` <dbl>, `Casualty Severity` <dbl>,
## #   `Sex of Casualty` <dbl>, `Age of Casualty` <dbl>
```

```
unique(df_clean$`Weather Conditions`)
```

```
## [1] 2 1 5 7 3 6 4 8 9
```

Task: Checking for inconsistencies and transforming values in the “Weather Conditions” column.

```
# Transform Weather Conditions using case_when
df_clean <- df_clean %>%
  mutate(
    `Weather Conditions` = case_when(
      `Weather Conditions` == "1" ~ "Fine without high winds",
      `Weather Conditions` == "2" ~ "Raining without high winds",
      `Weather Conditions` == "3" ~ "Snowing without high winds",
      `Weather Conditions` == "4" ~ "Fine with high winds",
      `Weather Conditions` == "5" ~ "Raining with high winds",
      `Weather Conditions` == "6" ~ "Snowing with high winds",
      `Weather Conditions` == "7" ~ "Fog or mist ? if hazard",
      `Weather Conditions` == "8" ~ "Other",
      `Weather Conditions` == "9" ~ "Unknown",
      TRUE ~ as.character(`Weather Conditions`)
    )
  )

# Display unique values after transformation
unique(df_clean$`Weather Conditions`)
```

```
## [1] "Raining without high winds" "Fine without high winds"
## [3] "Raining with high winds"    "Fog or mist ? if hazard"
## [5] "Snowing without high winds" "Snowing with high winds"
## [7] "Fine with high winds"       "Other"
## [9] "Unknown"
```

Task:

```
df_clean
```

```
## # A tibble: 2,069 × 14
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <chr>         <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 10 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Local Authority` <chr>,
## #   `Type of Vehicle` <dbl>, `Casualty Class` <dbl>, `Casualty Severity` <dbl>,
## #   `Sex of Casualty` <dbl>, `Age of Casualty` <dbl>
```

Task:Checking for inconsistency in Casualty Class column then mapping the related values

```
unique(df_clean$`Casualty Class`)
```

```
## [1] 2 3 1
```

Task: Transforming values in the “Casualty Class” column to descriptive categories.

```
# Transform Casualty Class using case_when
df_clean <- df_clean %>%
  mutate(
    `Casualty Class` = case_when(
      `Casualty Class` == "1" ~ "Driver or rider",
      `Casualty Class` == "2" ~ "Vehicle or pillion passenger",
      `Casualty Class` == "3" ~ "Pedestrian",
      TRUE ~ as.character(`Casualty Class`)
    )
  )

# Display unique values after transformation
unique(df_clean$`Casualty Class`)
```

```
## [1] "Vehicle or pillion passenger" "Pedestrian"
## [3] "Driver or rider"
```

```
df_clean
```

```
## # A tibble: 2,069 × 14
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <chr>         <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 10 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Local Authority` <chr>,
## #   `Type of Vehicle` <dbl>, `Casualty Class` <chr>, `Casualty Severity` <dbl>,
## #   `Sex of Casualty` <dbl>, `Age of Casualty` <dbl>
```

```
unique(df_clean$'Casualty Severity')
```

```
## [1] 2 3 1
```

Task: Transforming values in the “Casualty Severity” column to descriptive categories.

```
casualty_severity_mapping <- c("1" = "Fatal",
                               "2" = "Serious",
                               "3" = "Slight")

df_clean <- df_clean %>%
  mutate(`Casualty Severity` = map_chr(`Casualty Severity`, function(x) {
    ifelse(as.character(x) %in% names(casualty_severity_mapping), casualty_severity_mapping[a
s.character(x)], as.character(x))
  }))

# Display unique values after transformation
unique(df_clean$`Casualty Severity`)
```

```
## [1] "Serious" "Slight"  "Fatal"
```

```
df_clean
```

```
## # A tibble: 2,069 × 14
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <chr>         <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 10 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Local Authority` <chr>,
## #   `Type of Vehicle` <dbl>, `Casualty Class` <chr>, `Casualty Severity` <chr>,
## #   `Sex of Casualty` <dbl>, `Age of Casualty` <dbl>
```

```
unique(df_clean$`Type of Vehicle`)
```

```
##  [1]  9  4 19  8  3 21  2 10  5 11  1 97 90 20 23 22 17 18 16
```

Task: Transforming values in the “Type of Vehicle” column to descriptive categories.

```r
df_clean <- df_clean %>%
  mutate(
    `Type of Vehicle` = case_when(
      `Type of Vehicle` == "1" ~ "Pedal cycle",
      `Type of Vehicle` == "2" ~ "M/cycle 50cc and under",
      `Type of Vehicle` == "3" ~ "Motorcycle over 50cc and up to 125cc",
      `Type of Vehicle` == "4" ~ "Motorcycle over 125cc and up to 500cc",
      `Type of Vehicle` == "5" ~ "Motorcycle over 500cc",
      `Type of Vehicle` == "8" ~ "Taxi/Private hire car",
      `Type of Vehicle` == "9" ~ "Car",
      `Type of Vehicle` == "10" ~ "Minibus (8 – 16 passenger seats)",
      `Type of Vehicle` == "11" ~ "Bus or coach (17 or more passenger seats)",
      `Type of Vehicle` == "14" ~ "Other motor vehicle",
      `Type of Vehicle` == "15" ~ "Other non-motor vehicle",
      `Type of Vehicle` == "16" ~ "Ridden horse",
      `Type of Vehicle` == "17" ~ "Agricultural vehicle (includes diggers etc.)",
      `Type of Vehicle` == "18" ~ "Tram / Light rail",
      `Type of Vehicle` == "19" ~ "Goods vehicle 3.5 tonnes mgw and under",
      `Type of Vehicle` == "20" ~ "Goods vehicle over 3.5 tonnes and under 7.5 tonnes mgw",
      `Type of Vehicle` == "21" ~ "Goods vehicle 7.5 tonnes mgw and over",
      `Type of Vehicle` == "22" ~ "Mobility Scooter",
      `Type of Vehicle` == "90" ~ "Other Vehicle",
      `Type of Vehicle` == "97" ~ "Motorcycle - Unknown CC",
      TRUE ~ as.character(`Type of Vehicle`)
    )
  )

# Display unique values after transformation
unique(df_clean$`Type of Vehicle`)
```

```
##  [1] "Car"
##  [2] "Motorcycle over 125cc and up to 500cc"
##  [3] "Goods vehicle 3.5 tonnes mgw and under"
##  [4] "Taxi/Private hire car"
##  [5] "Motorcycle over 50cc and up to 125cc"
##  [6] "Goods vehicle 7.5 tonnes mgw and over"
##  [7] "M/cycle 50cc and under"
##  [8] "Minibus (8 – 16 passenger seats)"
##  [9] "Motorcycle over 500cc"
## [10] "Bus or coach (17 or more passenger seats)"
## [11] "Pedal cycle"
## [12] "Motorcycle - Unknown CC"
## [13] "Other Vehicle"
## [14] "Goods vehicle over 3.5 tonnes and under 7.5 tonnes mgw"
## [15] "23"
## [16] "Mobility Scooter"
## [17] "Agricultural vehicle (includes diggers etc.)"
## [18] "Tram / Light rail"
## [19] "Ridden horse"
```

```r
df_clean
```

```
## # A tibble: 2,069 × 14
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <chr>         <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 10 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Local Authority` <chr>,
## #   `Type of Vehicle` <chr>, `Casualty Class` <chr>, `Casualty Severity` <chr>,
## #   `Sex of Casualty` <dbl>, `Age of Casualty` <dbl>
```

```
unique(df_clean$`Sex of Casualty`)
```

```
## [1] 2 1
```

Task: Transforming values in the “Sex of Casualty” column to descriptive categories.

```
df_clean <- df_clean %>%
  mutate(
    `Sex of Casualty` = case_when(
      `Sex of Casualty` == "1" ~ "Male",
      `Sex of Casualty` == "2" ~ "Female",
      TRUE ~ as.character(`Sex of Casualty`)
    )
  )

# Display unique values after transformation
unique(df_clean$`Sex of Casualty`)
```

```
## [1] "Female" "Male"
```

```
df_clean
```

```
## # A tibble: 2,069 × 14
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <chr>         <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 10 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Local Authority` <chr>,
## #   `Type of Vehicle` <chr>, `Casualty Class` <chr>, `Casualty Severity` <chr>,
## #   `Sex of Casualty` <chr>, `Age of Casualty` <dbl>
```

Task:Identifying and removing column that is not needed.

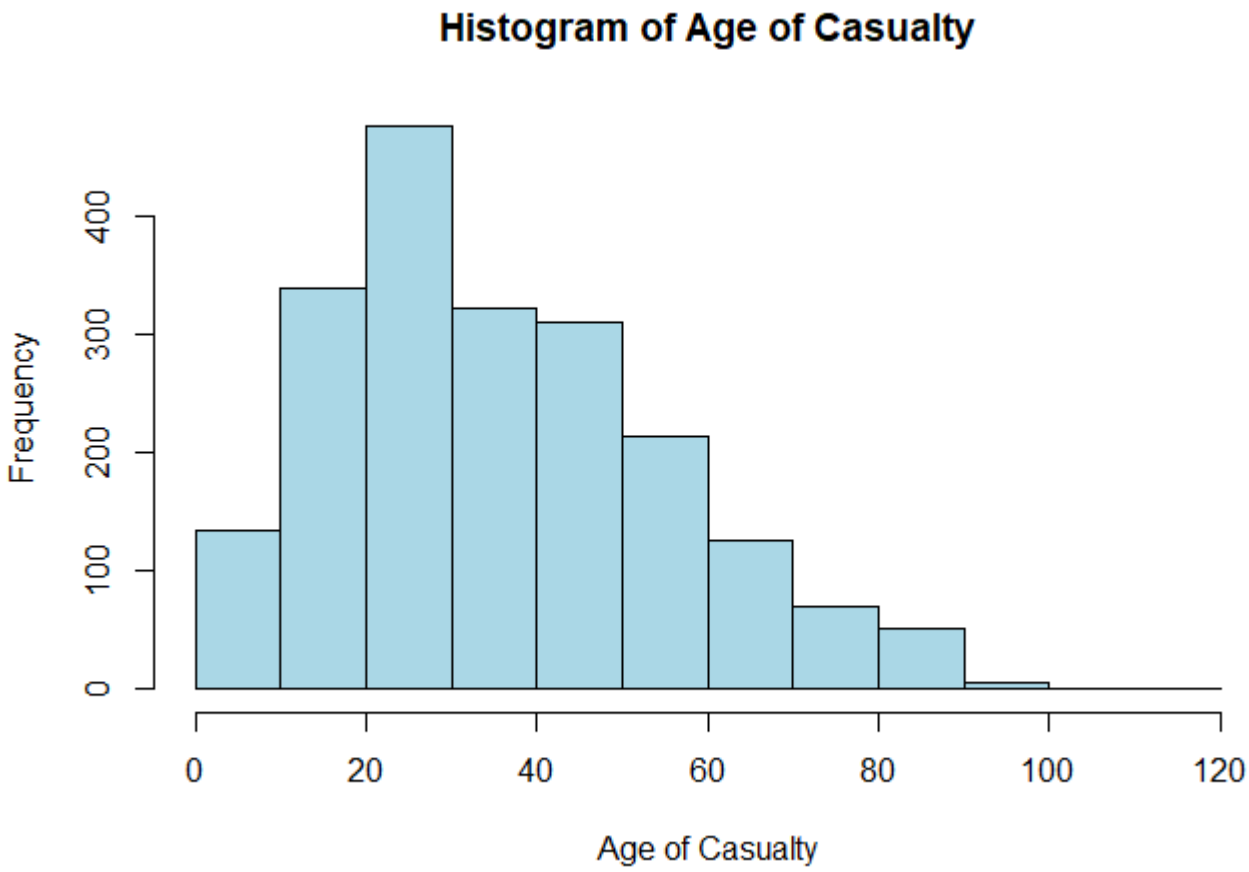
```
df_clean <- df_clean %>% select(-c(`Local Authority`))
```

```
rows_with_missing_daylight_dark <- df_clean[is.na(df_clean$`Daylight/Dark`), ]

print(rows_with_missing_daylight_dark)
```

```
## # A tibble: 18 × 13
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <chr>         <chr>
##  1                    1 2017-02-18      23:30         Unclassified
##  2                    1 2017-02-22      19:39         A
##  3                    1 2017-06-01      03:25         Unclassified
##  4                    2 2017-08-18      21:00         Unclassified
##  5                    2 2017-11-08      16:51         Unclassified
##  6                    1 2016-01-17      20:40         Unclassified
##  7                    2 2016-03-10      06:57         Unclassified
##  8                    1 2016-05-07      22:07         A
##  9                    1 2015-03-30      19:55         Unclassified
## 10                    1 2015-05-31      01:50         Unclassified
## 11                    1 2015-08-28      23:45         Unclassified
## 12                    1 2015-08-28      23:45         Unclassified
## 13                    1 2015-11-22      11:36         Unclassified
## 14                    1 2015-11-22      11:36         Unclassified
## 15                    1 2014-10-09      21:00         A
## 16                    1 2014-11-23      06:40         A
## 17                    1 2014-11-23      06:40         A
## 18                    1 2014-12-29      07:35         Unclassified
## # ℹ 9 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Type of Vehicle` <chr>,
## #   `Casualty Class` <chr>, `Casualty Severity` <chr>, `Sex of Casualty` <chr>,
## #   `Age of Casualty` <dbl>
```

Task: Ploting Histogram and then printing outliers

```
# Histogram of Age of Casualty
hist(df_clean$`Age of Casualty`, main = "Histogram of Age of Casualty",
     xlab = "Age of Casualty", col = "lightblue", border = "black", breaks = 10)
```

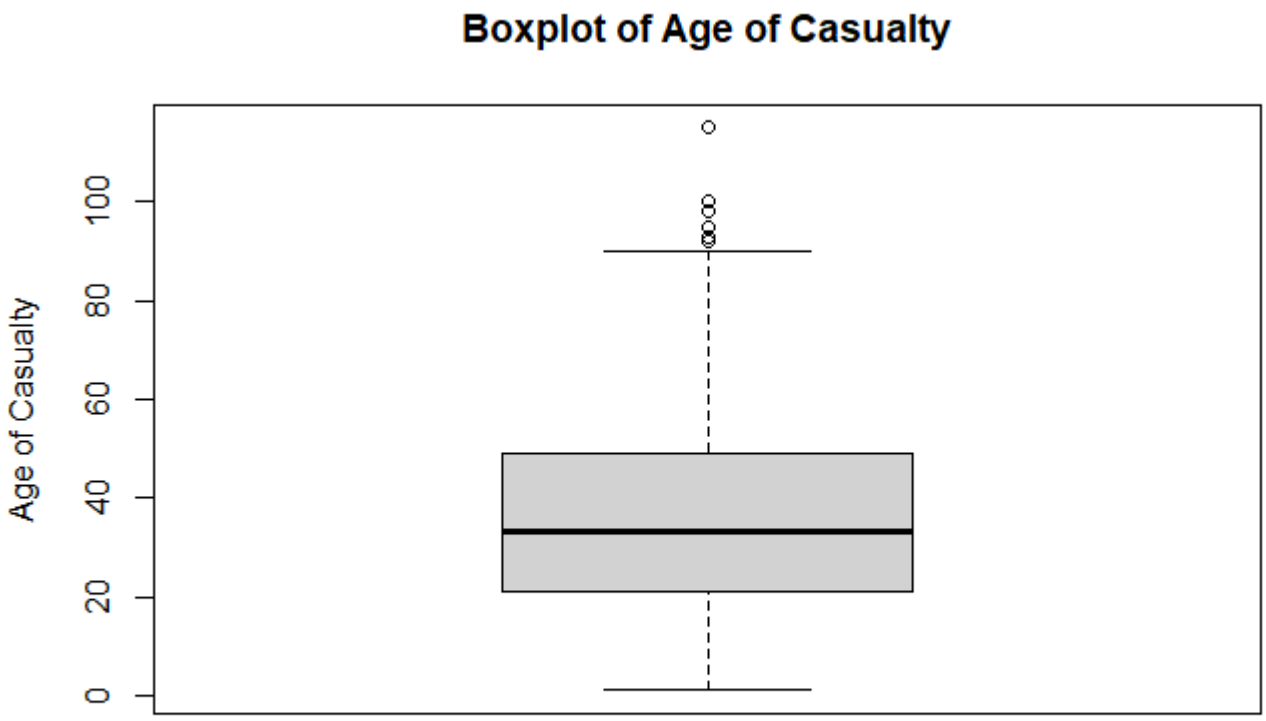



```
# Calculate IQR and bounds
iqr_age <- IQR(df_clean$`Age of Casualty`, na.rm = TRUE)
quantiles <- quantile(df_clean$`Age of Casualty`, probs = c(0.25, 0.75), na.rm = TRUE)
lower_bound <- max(0, quantiles[1] - 1.5 * iqr_age)
upper_bound <- quantiles[2] + 1.5 * iqr_age

# Identify outliers
outliers <- df_clean %>% filter(`Age of Casualty` < lower_bound | `Age of Casualty` > upper_b
ound)

# Display outliers
outliers$`Age of Casualty`
```

```
## [1] 115  93  93 100  92  95  98
```

Task:Performing outlier detection on the Age of Casualty column using methods like the IQR method, Z-score method, and visualizating

```
# Boxplot of Age of Casualty
boxplot(df_clean$`Age of Casualty`, main = "Boxplot of Age of Casualty",
        ylab = "Age of Casualty")
```

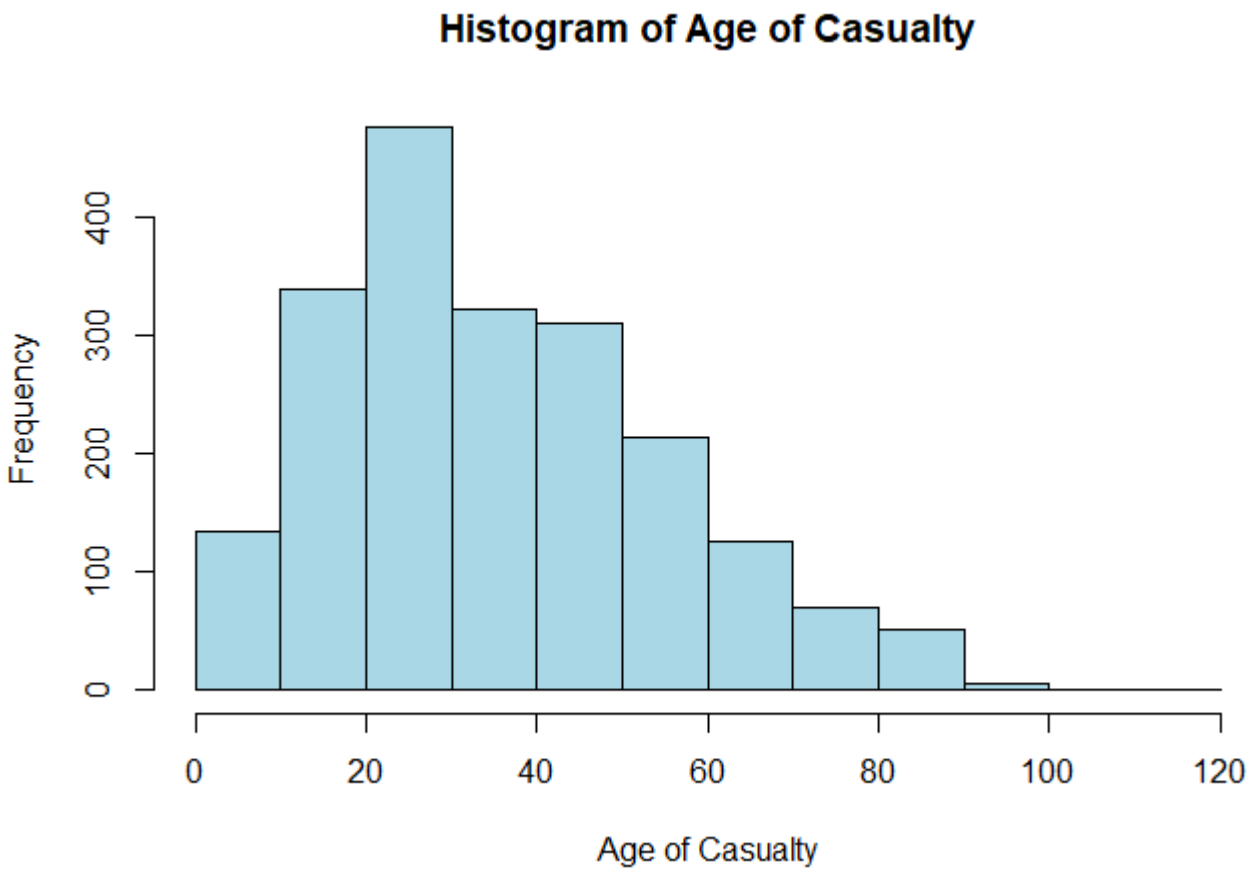



```
# Identify outliers
outlier_box <- boxplot(df_clean$`Age of Casualty`, plot = FALSE)$out

# Display outliers
outlier_box
```

```
## [1] 115  93  93 100  92  95  98
```

Task: 3sigma rule

```
# Histogram of Age of Casualty
hist(df_clean$`Age of Casualty`, main = "Histogram of Age of Casualty",
     xlab = "Age of Casualty", col = "lightblue", border = "black", breaks = 10)
```




```
# Calculate mean and standard deviation
mean_age <- mean(df_clean$`Age of Casualty`, na.rm = TRUE)
sd_age <- sd(df_clean$`Age of Casualty`, na.rm = TRUE)

# Calculate three sigma bounds
lower_bound <- mean_age - 3 * sd_age
upper_bound <- mean_age + 3 * sd_age

# Identify outliers
outliers <- df_clean %>% filter(`Age of Casualty` < lower_bound | `Age of Casualty` > upper_b
ound)

# Display outliers
outliers$`Age of Casualty`
```

```
## [1] 115 100  95  98
```

Task: Saving the cleaned dataset to a new CSV file named “clean_accident.csv”.

```
write_csv(df_clean, "clean_accident.csv")
```

#————————————————————————-PART-2————————————————————————

```
data <- read_csv("clean_accident.csv")
```

```
## Rows: 2069 Columns: 13
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr  (9): 1st Road Class, Road Surface, Lighting Conditions, Daylight/Dark, ...
## dbl  (2): Number of Vehicles, Age of Casualty
## date (1): Accident Date
## time (1): Time (24hr)
##
## 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.
```

```
data
```

```
## # A tibble: 2,069 × 13
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <time>        <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 9 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Type of Vehicle` <chr>,
## #   `Casualty Class` <chr>, `Casualty Severity` <chr>, `Sex of Casualty` <chr>,
## #   `Age of Casualty` <dbl>
```

Task: Filtering and summarizing the data to find the difference in the number of accidents between male and female drivers/riders for each weather condition.

```
# Load necessary libraries
library(dplyr)

# Filter and summarize data for male and female drivers/riders
gender_weather_accidents <- data %>%
  filter(`Casualty Class` == "Driver or rider") %>%
  group_by(`Weather Conditions`, `Sex of Casualty`) %>%
  summarise(accidents = n()) %>%
  spread(`Sex of Casualty`, accidents, fill = 0) %>%
  mutate(difference = Male - Female) %>%
  arrange(desc(difference))
```

```
## `summarise()` has grouped output by 'Weather Conditions'. You can override
## using the `.groups` argument.
```

```
# Display the result
gender_weather_accidents
```

```
## # A tibble: 9 × 4
## # Groups:   Weather Conditions [9]
##   `Weather Conditions`       Female  Male difference
##   <chr>                       <dbl> <dbl>      <dbl>
## 1 Fine without high winds       303   691        388
## 2 Raining without high winds     52    83         31
## 3 Raining with high winds        17    26          9
## 4 Fine with high winds            4     7          3
## 5 Fog or mist ? if hazard         3     6          3
## 6 Snowing without high winds      5     8          3
## 7 Snowing with high winds         0     2          2
## 8 Unknown                         7     5         -2
## 9 Other                           5     1         -4
```

Task:Extracting the year from the “Accident Date” and summarizing the number of casualties per year.

```
# Extract year from Accident Date
data <- data %>%
  mutate(Year = year(`Accident Date`))

# Summarize number of casualties per year
casualties_per_year <- data %>%
  group_by(Year) %>%
  summarise(total_casualties = n())

# Determine the year with the highest number of casualties
max_casualties_year <- casualties_per_year %>%
  filter(total_casualties == max(total_casualties))

# Display the result
casualties_per_year
```

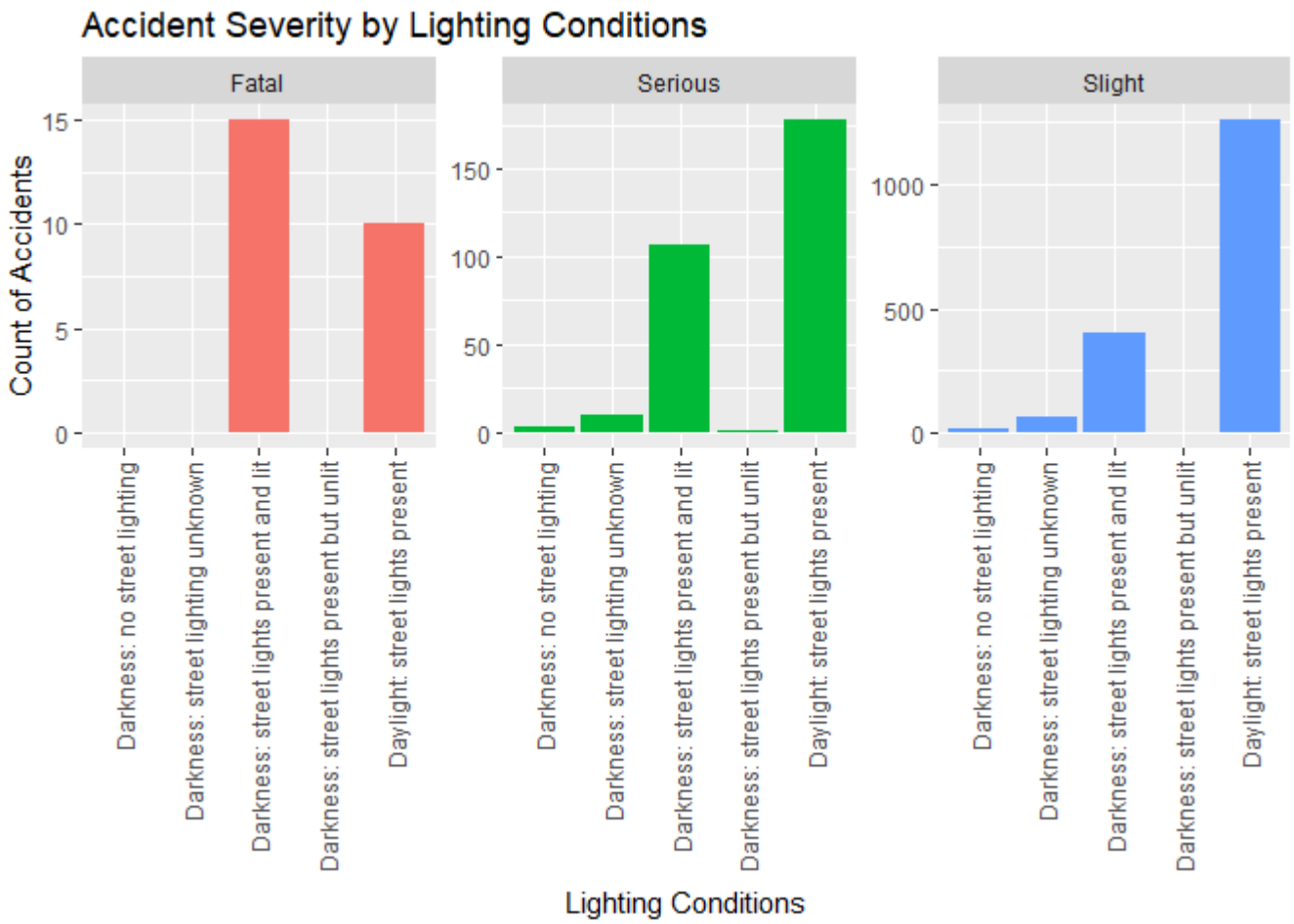
```
## # A tibble: 4 × 2
##    Year total_casualties
##   <dbl>            <int>
## 1  2014              623
## 2  2015              556
## 3  2016              554
## 4  2017              336
```

```
max_casualties_year
```

```
## # A tibble: 1 × 2
##    Year total_casualties
##   <dbl>            <int>
## 1  2014              623
```

Task:Visualizing how accident severity is distributed across different lighting conditions.

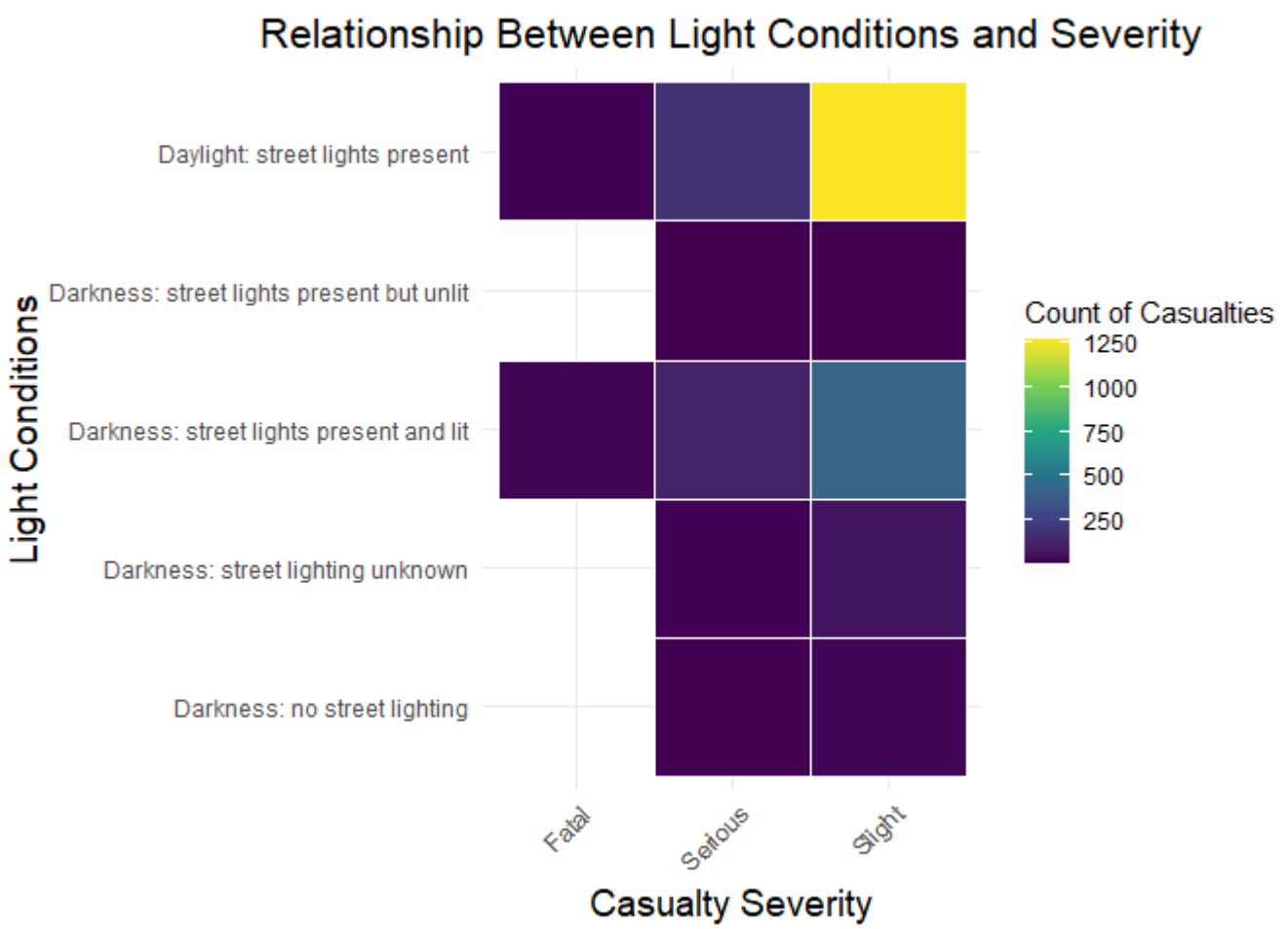
```
# Plot: Light conditions and severity
library(ggplot2)

ggplot(data, aes(x = `Lighting Conditions`, fill = `Casualty Severity`)) +
  geom_bar(position = "dodge") +
  labs(title = "Accident Severity by Lighting Conditions", x = "Lighting Conditions", y = "Co
unt of Accidents") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  facet_wrap(~`Casualty Severity`, scales = "free_y") +
  theme(legend.position = "none")
```



Task: Relation between Light Conditions and Severity using heatmap.

```
data %>%
  group_by(`Lighting Conditions`, `Casualty Severity`) %>%
  summarise(count = n()) %>%
  ggplot(aes(x = `Casualty Severity`, y = `Lighting Conditions`, fill = count)) +
  geom_tile(color = "white") +  # Add white borders around tiles for clarity
  labs(title = "Relationship Between Light Conditions and Severity",
       x = "Casualty Severity",
       y = "Light Conditions",
       fill = "Count of Casualties") +
  scale_fill_viridis_c() +  # Use viridis color palette for better differentiation
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(size = 16, hjust = 0.5),
        axis.title = element_text(size = 14))
```

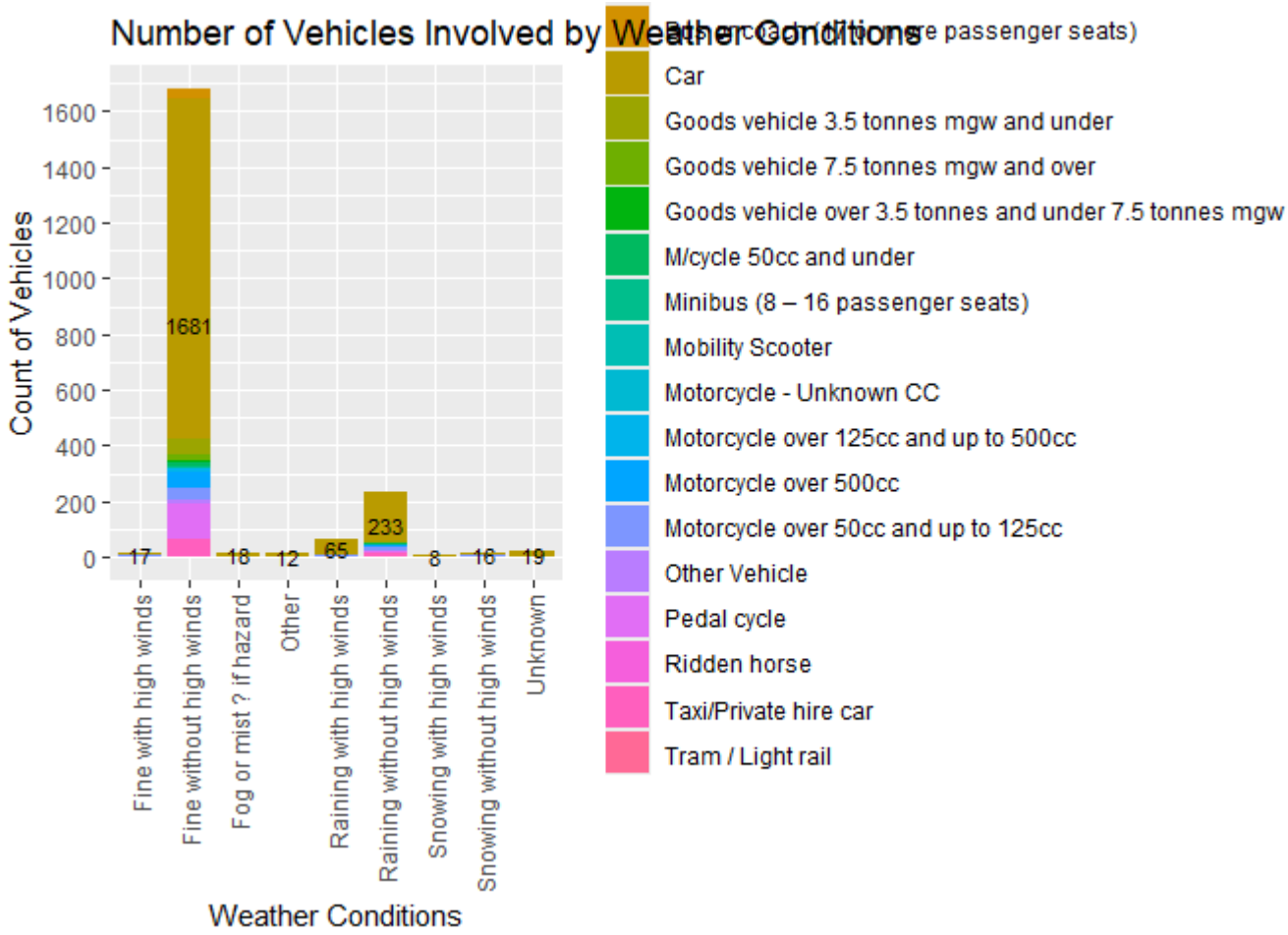
```
## `summarise()` has grouped output by 'Lighting Conditions'. You can override
## using the `.groups` argument.
```



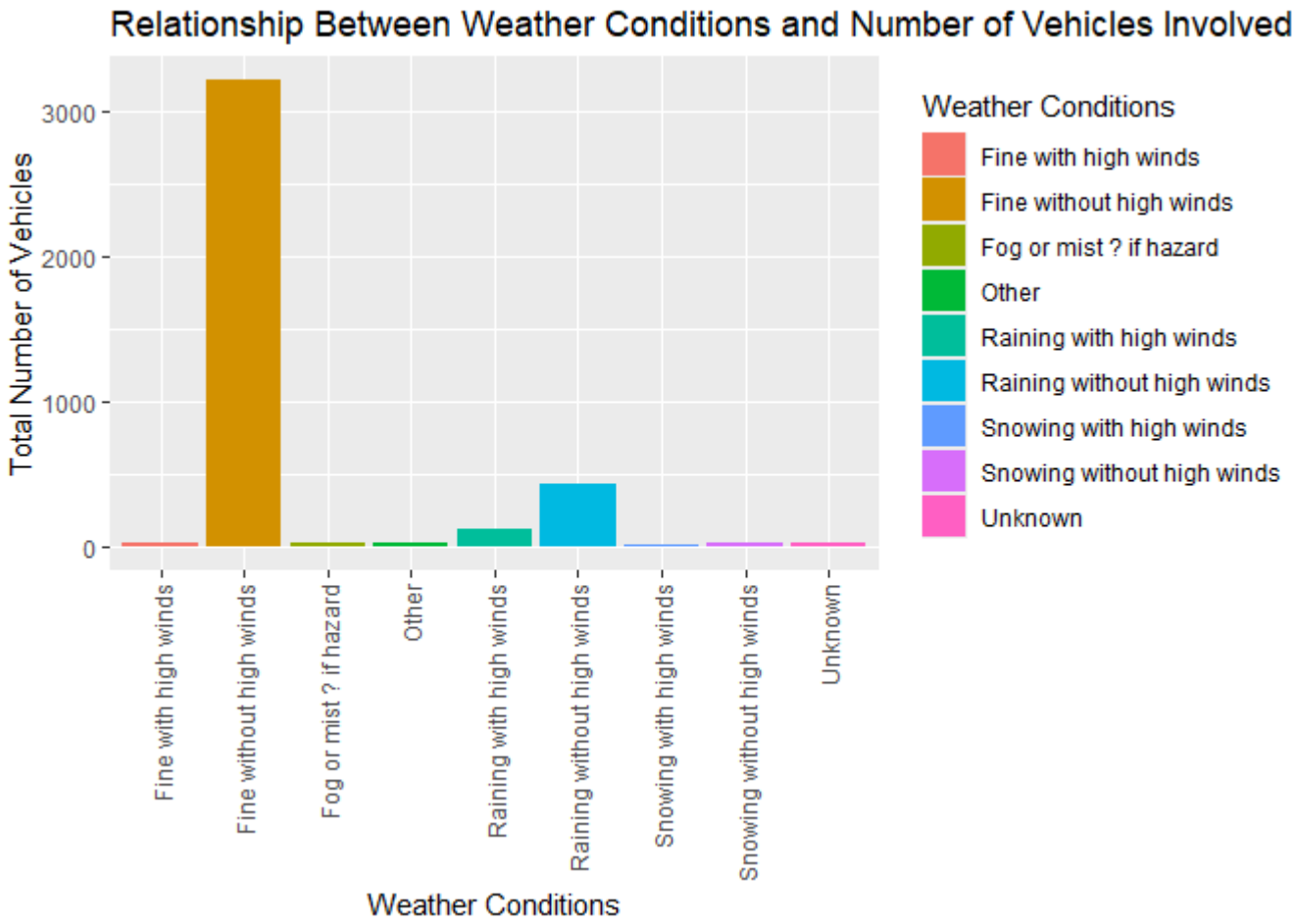
Task:Visualizing the distribution of vehicles involved categorized by different weather conditions.

```
# Plot: Weather condition and number of vehicles involved
ggplot(data, aes(x = `Weather Conditions`)) +
  geom_bar(aes(fill = `Type of Vehicle`), position = "stack") +
  labs(title = "Number of Vehicles Involved by Weather Conditions", x = "Weather Conditions",
y = "Count of Vehicles") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  geom_text(stat='count', aes(label=..count..), position=position_stack(vjust=0.5), size=3)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

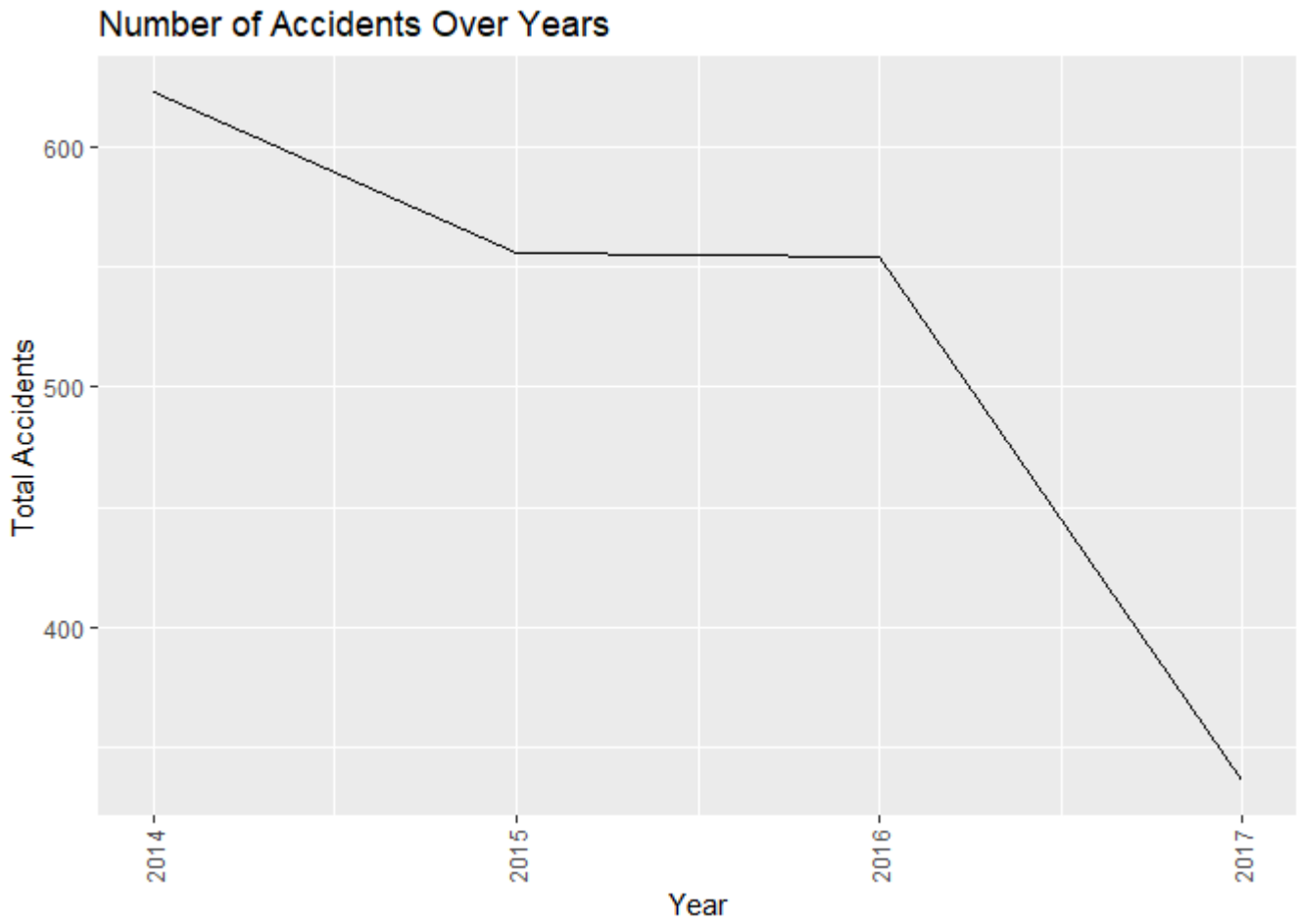



```
data %>%
  group_by(`Weather Conditions`) %>%
  summarise(total_vehicles = sum(`Number of Vehicles`)) %>%
  ggplot(aes(x = `Weather Conditions`, y = total_vehicles, fill = `Weather Conditions`)) +
  geom_bar(stat = "identity") +
  labs(title = "Relationship Between Weather Conditions and Number of Vehicles Involved",
       x = "Weather Conditions",
       y = "Total Number of Vehicles") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



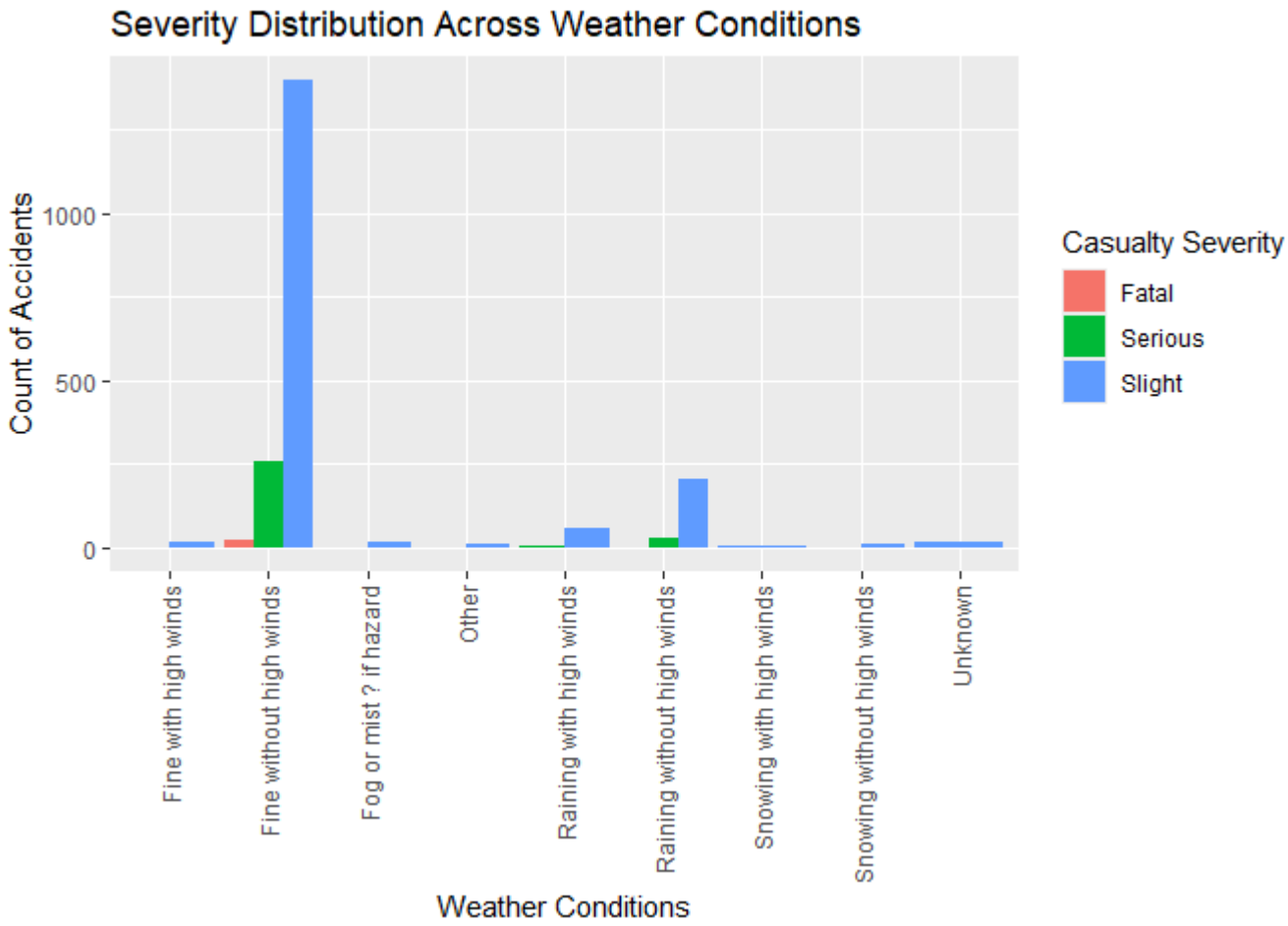
Task:Visualizing how the number of accidents changes over the years

```
# Line plot of accidents over years
ggplot(casualties_per_year, aes(x = Year, y = total_casualties)) +
  geom_line() +
  labs(title = "Number of Accidents Over Years", x = "Year", y = "Total Accidents") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



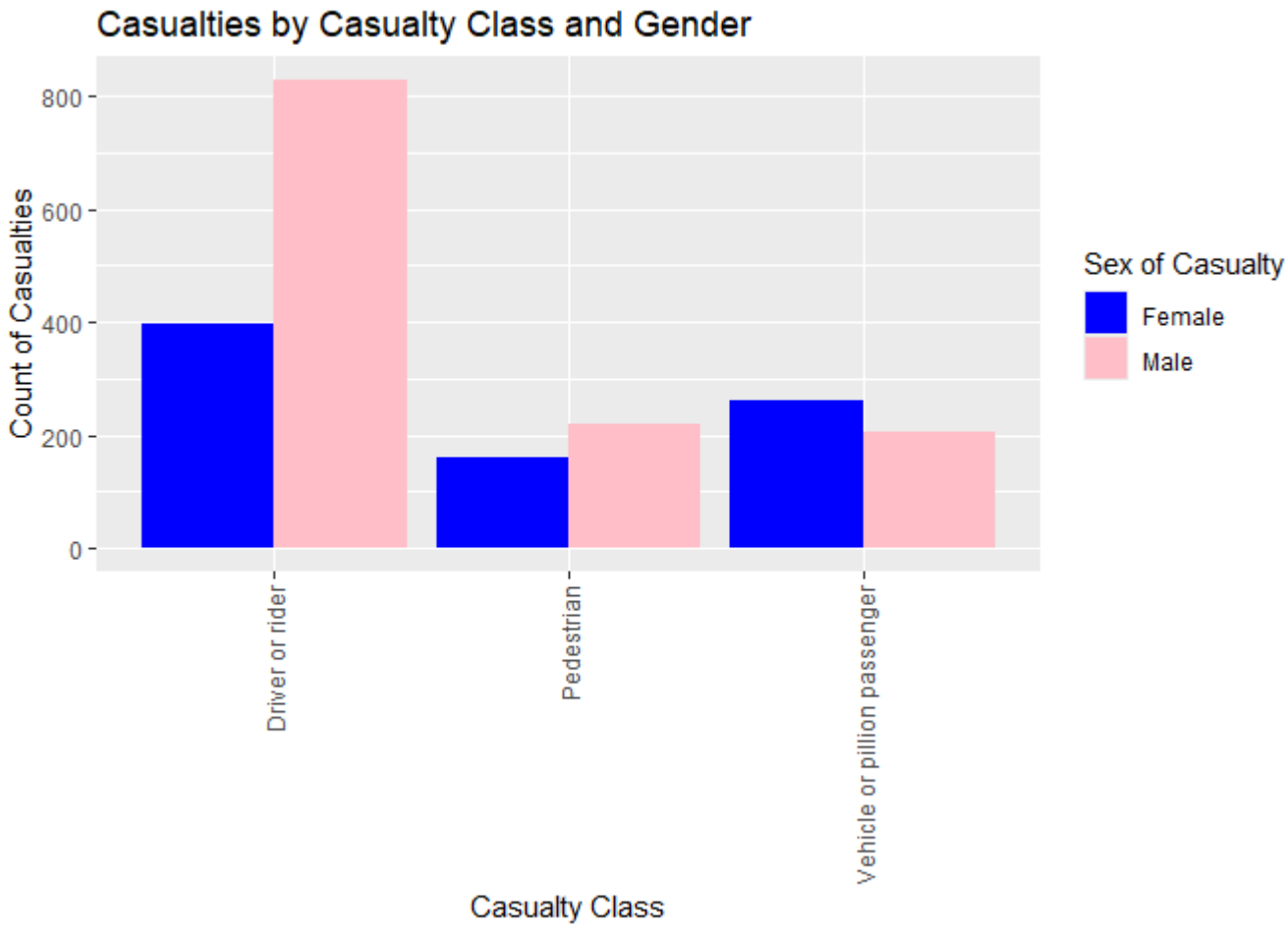
Task:Visualizing how severity is distributed across different weather conditions.

```
# Stacked bar chart of Weather Conditions and Severity
ggplot(data , aes(x = `Weather Conditions`, fill = `Casualty Severity`)) +
  geom_bar(position = "dodge") +
  labs(title = "Severity Distribution Across Weather Conditions", x = "Weather Conditions", y
= "Count of Accidents") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



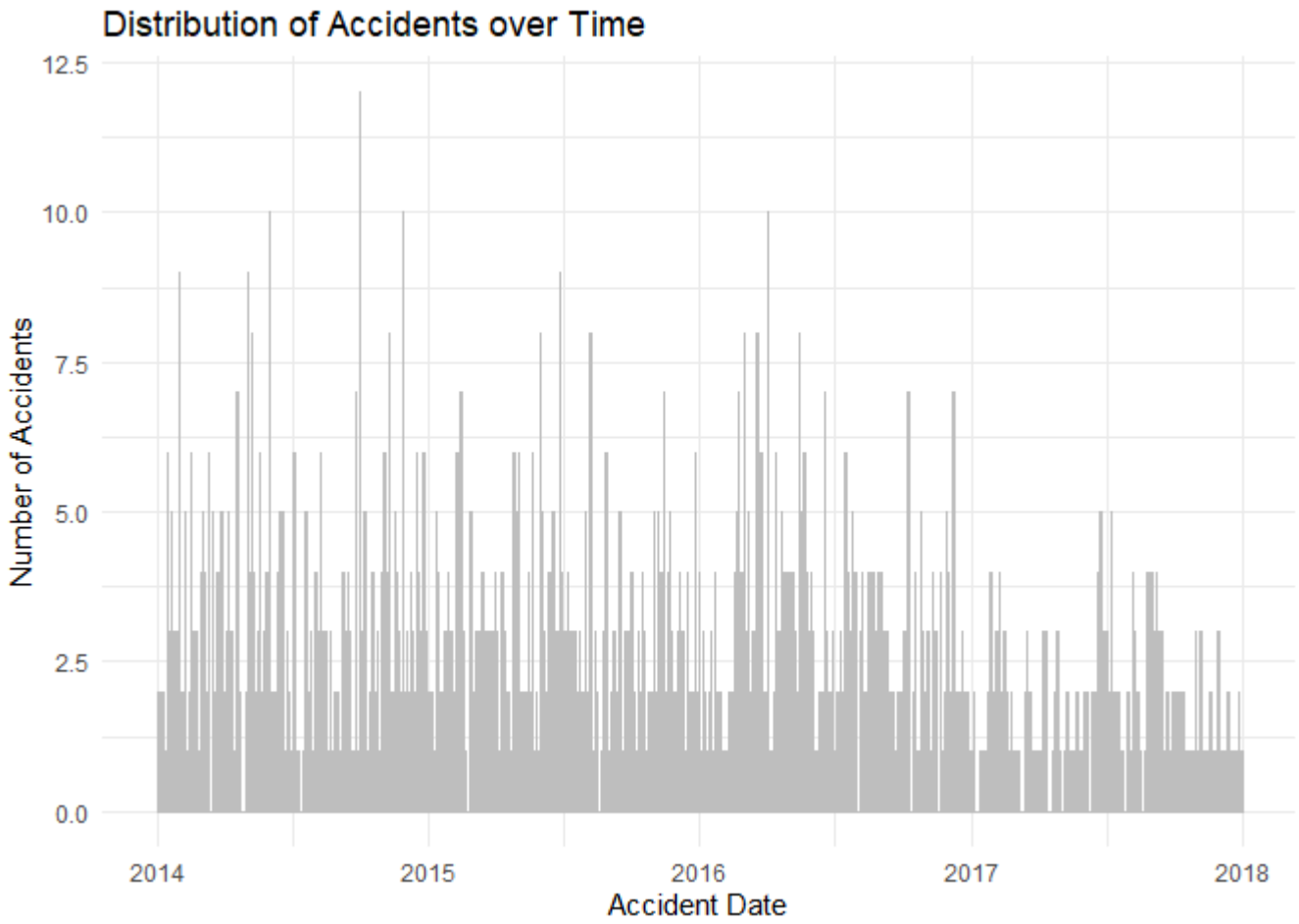
Task:Comparing the number of casualties between male and female across different casualty classes.

```
# Grouped bar chart of Casualty Class by Gender
ggplot(data, aes(x = `Casualty Class`, fill = `Sex of Casualty`)) +
  geom_bar(position = "dodge") +
  labs(title = "Casualties by Casualty Class and Gender", x = "Casualty Class", y = "Count of
Casualties") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  scale_fill_manual(values = c("blue", "pink"))
```



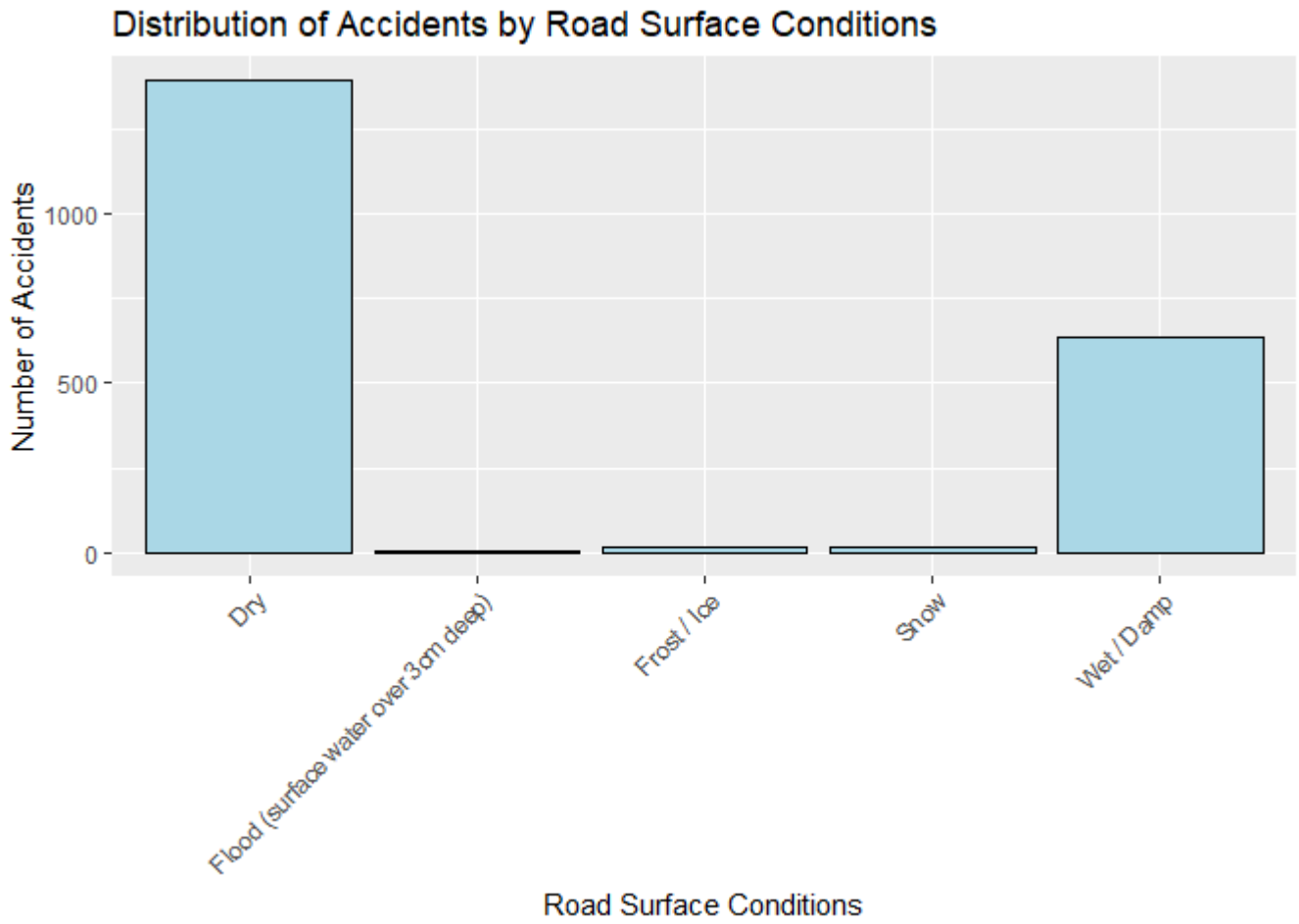
Task: Visualizing the distribution of accidents over time.

```
ggplot(data, aes(x = `Accident Date`)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "gray") +
  labs(title = "Distribution of Accidents over Time",
       x = "Accident Date",
       y = "Number of Accidents") +
  theme_minimal()
```



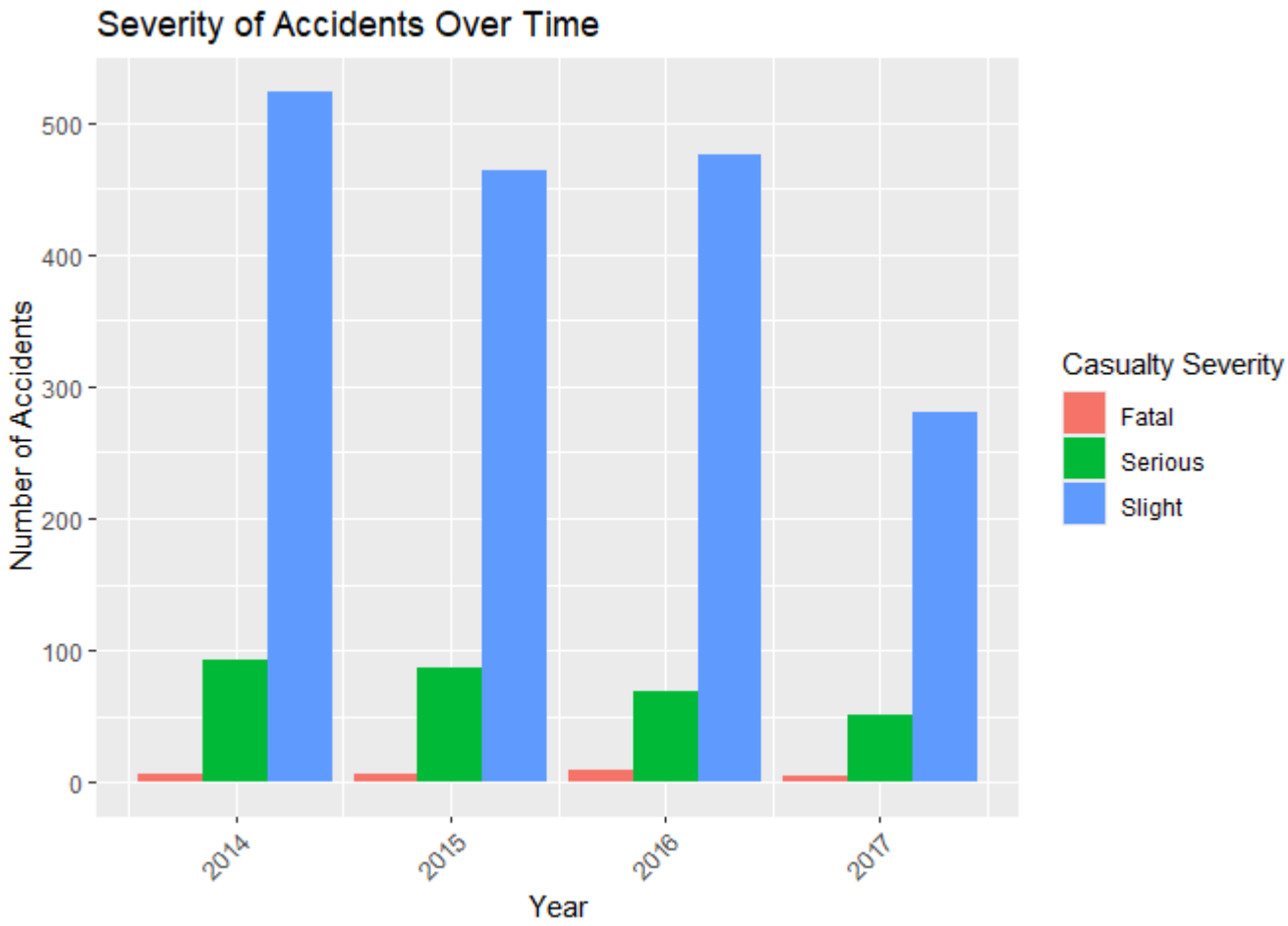
Task:Visualizing the distribution of accidents categorized by different road surface conditions.

```
# Load ggplot2 library
library(ggplot2)

# Bar plot of accidents by road surface conditions
ggplot(data, aes(x = `Road Surface`)) +
  geom_bar(fill = "lightblue", color = "black") +
  labs(title = "Distribution of Accidents by Road Surface Conditions",
       x = "Road Surface Conditions", y = "Number of Accidents") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



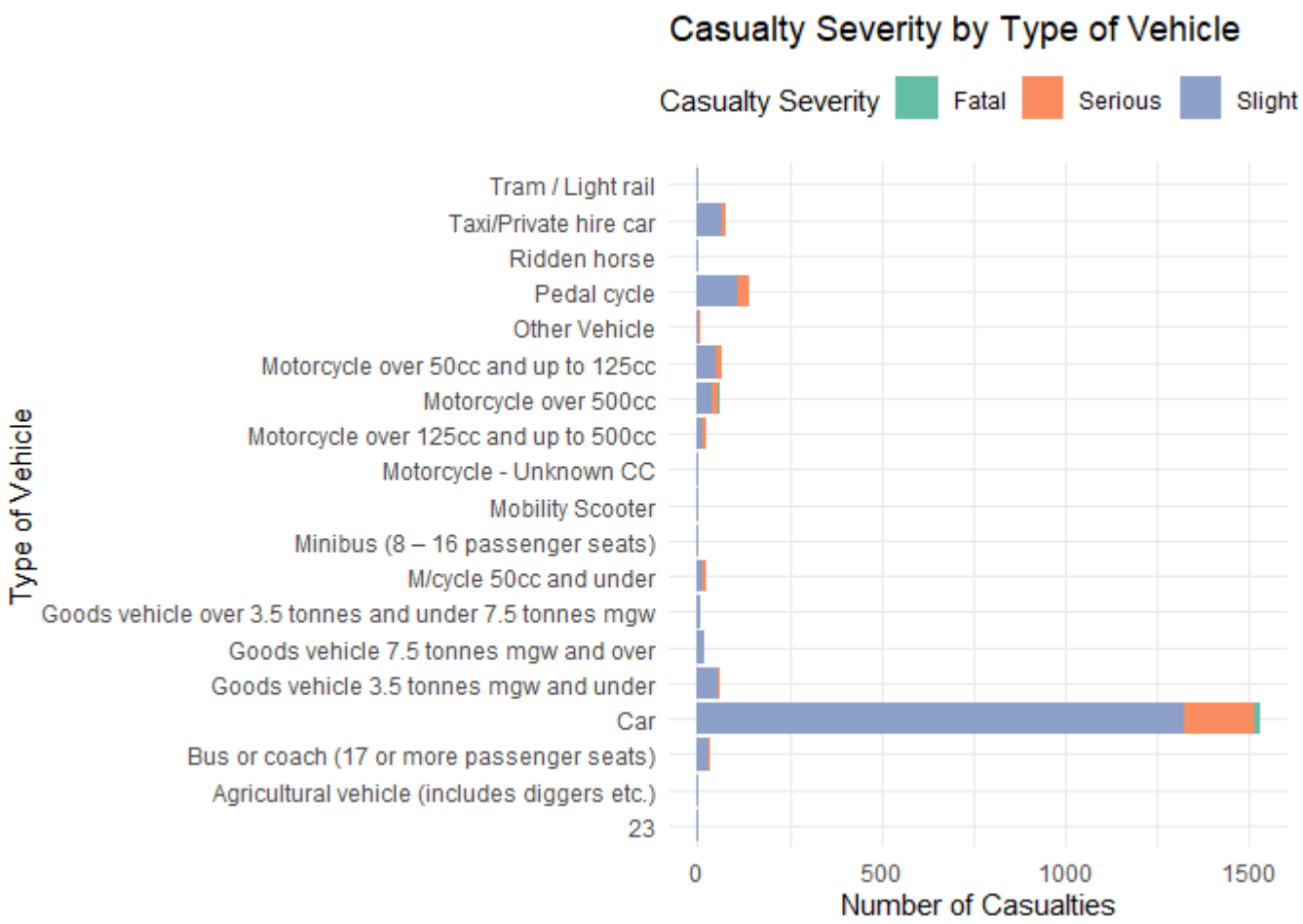
Task: Visualizing how the severity of accidents has changed over time.

```
# Extract year from Accident Date if not already extracted
data <- data %>%
  mutate(Year = year(`Accident Date`))

# Plot number of accidents by severity over time
ggplot(data, aes(x = Year, fill = `Casualty Severity`)) +
  geom_bar(position = "dodge") +
  labs(title = "Severity of Accidents Over Time",
       x = "Year", y = "Number of Accidents") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



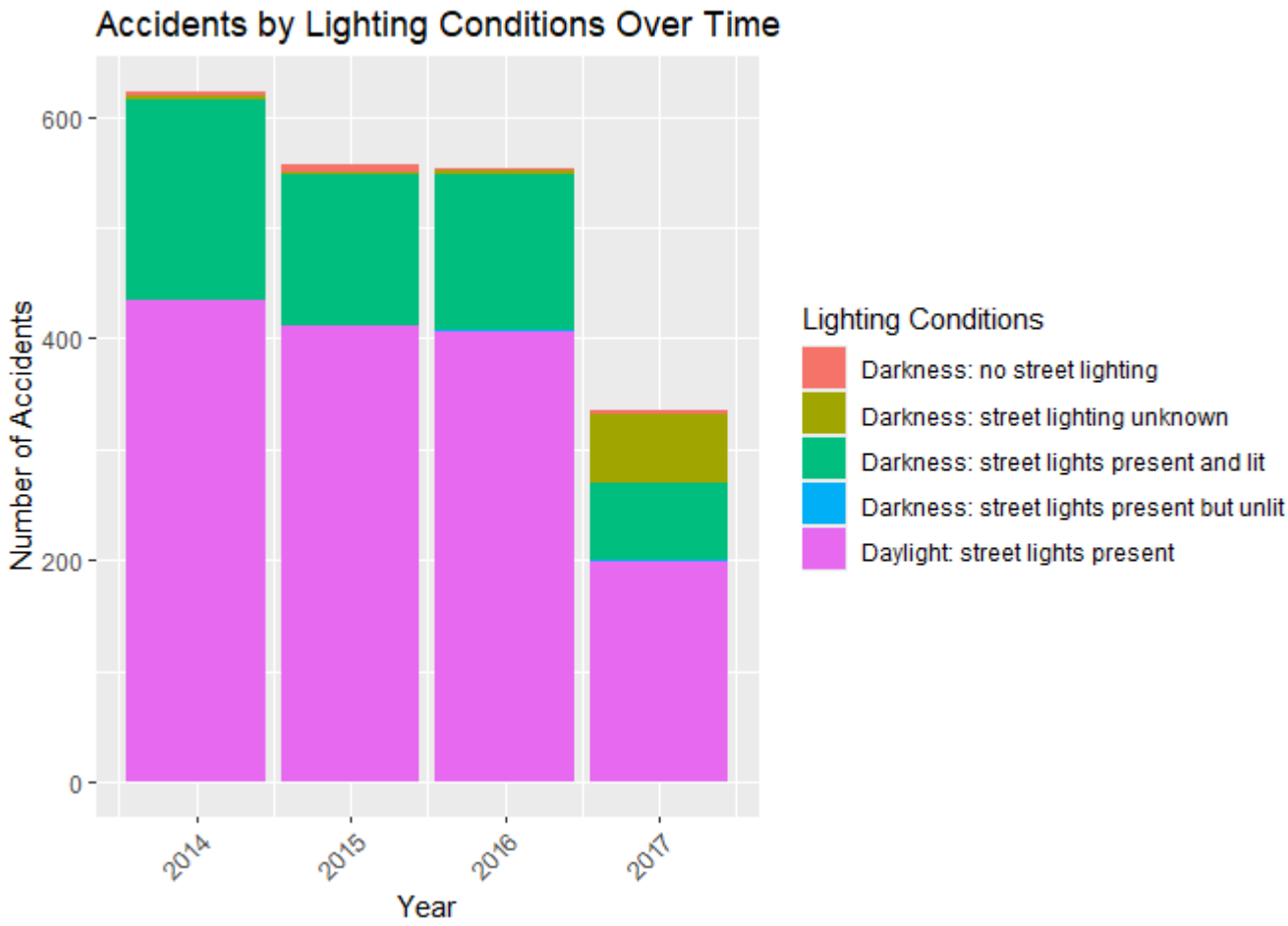
Task:Visualizing how casualty severity varies with different types of vehicles involved.

```
# Adjust plot size and clarity
ggplot(data, aes(x = `Type of Vehicle`, fill = `Casualty Severity`)) +
  geom_bar(position = "stack") +
  labs(title = "Casualty Severity by Type of Vehicle",
       x = "Type of Vehicle", y = "Number of Casualties") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        plot.title = element_text(size = 16, hjust = 0.5),
        axis.title = element_text(size = 14)) +
  theme_minimal() +
  theme(legend.position = "top") +
  coord_flip() +
  scale_fill_brewer(palette = "Set2")
```



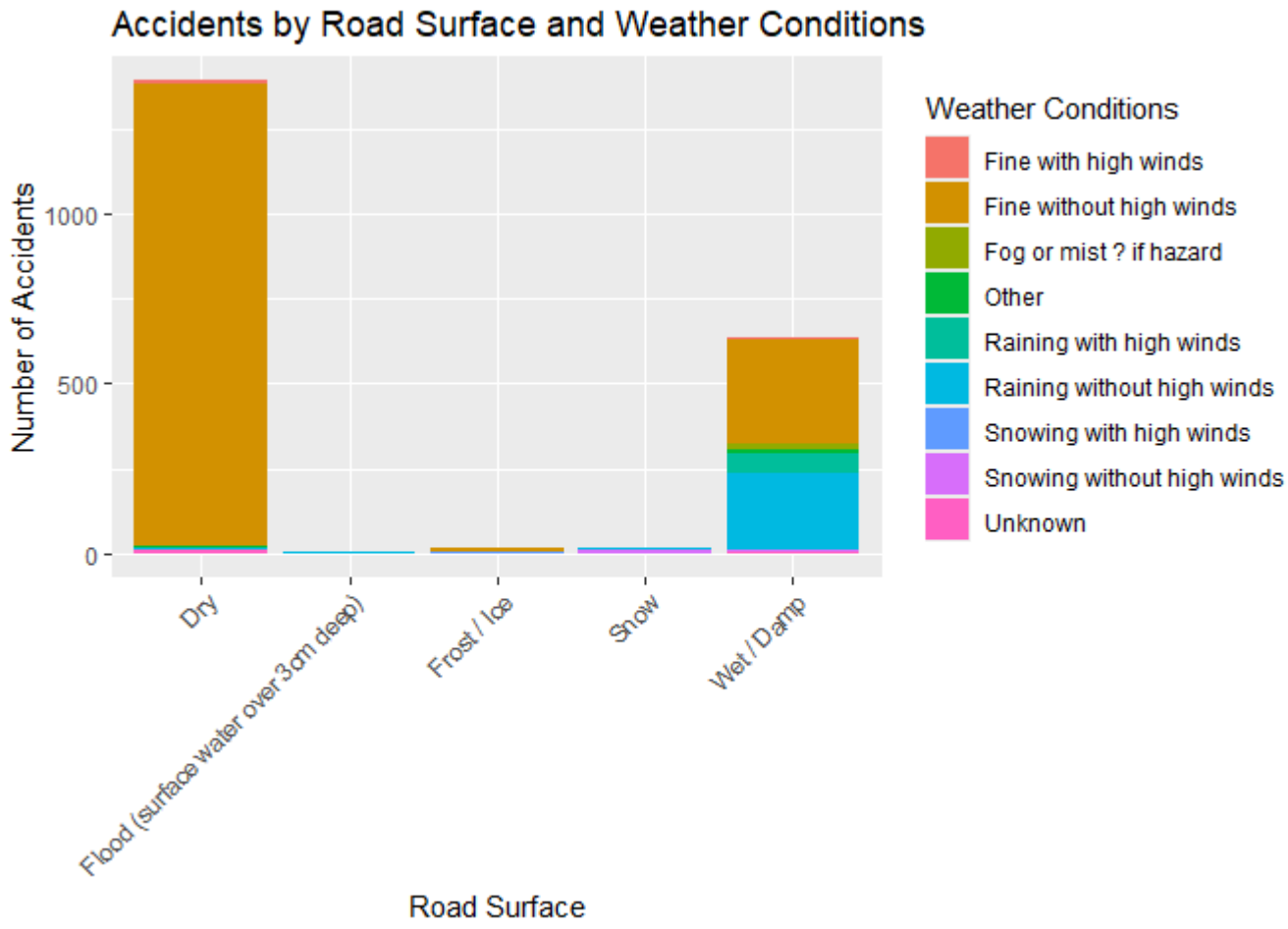
Task: Visualizing how accidents vary with different lighting conditions over time.

```
# Plot: Accidents by lighting conditions over time
ggplot(data, aes(x = Year, fill = `Lighting Conditions`)) +
  geom_bar(position = "stack") +
  labs(title = "Accidents by Lighting Conditions Over Time",
       x = "Year", y = "Number of Accidents") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Task: Visualizing how accidents are distributed across different combinations of road surface and weather conditions.

```
# Plot: Accidents by road surface and weather conditions
ggplot(data, aes(x = `Road Surface`, fill = `Weather Conditions`)) +
  geom_bar(position = "stack") +
  labs(title = "Accidents by Road Surface and Weather Conditions",
       x = "Road Surface", y = "Number of Accidents") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



#————————————————————————PART-3————————————————————————

Task: Loading necessary libraries

```
# List of packages to install
packages <- c("tidyr", "dplyr", "caret", "MASS", "ggplot2", "lubridate")

# Loop through each package
for (package in packages) {
  # Check if the package is not already installed
  if (!(package %in% installed.packages())) {
    # Install the package
    install.packages(package)
  }
}
```

Task:Load Cleaned Data

```
data <- read_csv("clean_accident.csv")
```

```
## Rows: 2069 Columns: 13
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr  (9): 1st Road Class, Road Surface, Lighting Conditions, Daylight/Dark, ...
## dbl  (2): Number of Vehicles, Age of Casualty
## date (1): Accident Date
## time (1): Time (24hr)
##
## 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.
```

```
data
```

```
## # A tibble: 2,069 × 13
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <time>        <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 9 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Type of Vehicle` <chr>,
## #   `Casualty Class` <chr>, `Casualty Severity` <chr>, `Sex of Casualty` <chr>,
## #   `Age of Casualty` <dbl>
```

Task:Ensuring No Missing Values in Selected Attributes

```
# Select relevant columns
#data <- data %>%
 # select(`Casualty Class`, `Casualty Severity`, `Type of Vehicle`, `Weather Conditions`, `Ag
e of Casualty`)

# Drop rows with missing values in predictor variables
#data <- data %>% drop_na(`Casualty Class`, `Casualty Severity`, `Type of Vehicle`, `Weather
Conditions`)

data
```

```
## # A tibble: 2,069 × 13
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <time>        <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 9 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <chr>, `Type of Vehicle` <chr>,
## #   `Casualty Class` <chr>, `Casualty Severity` <chr>, `Sex of Casualty` <chr>,
## #   `Age of Casualty` <dbl>
```

Task:Converting categorical variables to factors to prepare for model building.

```
data <- data %>%
  mutate(
    `Casualty Class` = as.factor(`Casualty Class`),
    `Casualty Severity` = as.factor(`Casualty Severity`),
    `Type of Vehicle` = as.factor(`Type of Vehicle`),
    `Weather Conditions` = as.factor(`Weather Conditions`)
  )
data
```

```
## # A tibble: 2,069 × 13
##    `Number of Vehicles` `Accident Date` `Time (24hr)` `1st Road Class`
##                   <dbl> <date>          <time>        <chr>
##  1                    2 2017-01-01      21:20         Unclassified
##  2                    2 2017-01-04      15:00         Unclassified
##  3                    2 2017-01-05      07:32         A
##  4                    2 2017-01-05      09:30         A
##  5                    2 2017-01-14      09:09         Unclassified
##  6                    1 2017-01-15      16:59         Unclassified
##  7                    1 2017-01-16      10:59         A
##  8                    3 2017-01-19      18:49         B
##  9                    1 2017-01-20      14:08         Unclassified
## 10                    1 2017-01-22      13:25         A
## # ℹ 2,059 more rows
## # ℹ 9 more variables: `Road Surface` <chr>, `Lighting Conditions` <chr>,
## #   `Daylight/Dark` <chr>, `Weather Conditions` <fct>, `Type of Vehicle` <fct>,
## #   `Casualty Class` <fct>, `Casualty Severity` <fct>, `Sex of Casualty` <chr>,
## #   `Age of Casualty` <dbl>
```

Task: Splitting the data into training and test sets for model training and evaluation.

```r
# Install and load caret package if not already installed
if (!requireNamespace("caret", quietly = TRUE)) {
  install.packages("caret")
}
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
# Assuming missing values in 'Age of Casualty' are handled
# Example: Imputing missing values with median
median_age <- median(data$`Age of Casualty`, na.rm = TRUE)
data$`Age of Casualty`[is.na(data$`Age of Casualty`)] <- median_age

# Now proceed with data partitioning
set.seed(123)
trainingIndex <- createDataPartition(data$`Age of Casualty`, p = 0.8, list = FALSE)
trainingData <- data[trainingIndex, ]
testingData <- data[-trainingIndex, ]
```

Task: Building a linear regression model to predict missing values in the “Age of Casualty” column.

```r
model <- lm(`Age of Casualty` ~ `Casualty Class` + `Casualty Severity` + `Type of Vehicle` +
`Weather Conditions`, data = trainingData)

# Print model summary
summary(model)
```

```
##
## Call:
## lm(formula = `Age of Casualty` ~ `Casualty Class` + `Casualty Severity` +
##     `Type of Vehicle` + `Weather Conditions`, data = trainingData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -41.63 -13.68  -3.68  11.19  67.92
##
## Coefficients:
##                                                                  Estimate
## (Intercept)                                                        27.357
## `Casualty Class`Pedestrian                                         -5.400
## `Casualty Class`Vehicle or pillion passenger                       -9.602
## `Casualty Severity`Serious                                          0.224
## `Casualty Severity`Slight                                          -3.067
## `Type of Vehicle`Bus or coach (17 or more passenger seats)         34.234
## `Type of Vehicle`Car                                               19.680
## `Type of Vehicle`Goods vehicle 3.5 tonnes mgw and under            18.669
## `Type of Vehicle`Goods vehicle 7.5 tonnes mgw and over             32.652
## `Type of Vehicle`Goods vehicle over 3.5 tonnes and under 7.5 tonnes mgw   36.516
## `Type of Vehicle`M/cycle 50cc and under                             5.875
## `Type of Vehicle`Minibus (8 – 16 passenger seats)                  11.200
## `Type of Vehicle`Mobility Scooter                                  30.000
## `Type of Vehicle`Motorcycle - Unknown CC                           60.400
## `Type of Vehicle`Motorcycle over 125cc and up to 500cc             15.960
## `Type of Vehicle`Motorcycle over 500cc                             22.150
## `Type of Vehicle`Motorcycle over 50cc and up to 125cc               7.089
## `Type of Vehicle`Other Vehicle                                     17.782
## `Type of Vehicle`Pedal cycle                                       14.807
## `Type of Vehicle`Ridden horse                                      11.000
## `Type of Vehicle`Taxi/Private hire car                             21.294
## `Type of Vehicle`Tram / Light rail                                  8.401
## `Weather Conditions`Fine without high winds                        -4.290
## `Weather Conditions`Fog or mist ? if hazard                         2.633
## `Weather Conditions`Other                                         -15.710
## `Weather Conditions`Raining with high winds                        -7.401
## `Weather Conditions`Raining without high winds                     -7.086
## `Weather Conditions`Snowing with high winds                        -5.768
## `Weather Conditions`Snowing without high winds                     -2.308
## `Weather Conditions`Unknown                                        -8.822
##                                                                 Std. Error
## (Intercept)                                                         20.130
## `Casualty Class`Pedestrian                                           1.294
## `Casualty Class`Vehicle or pillion passenger                         1.212
## `Casualty Severity`Serious                                           4.397
## `Casualty Severity`Slight                                            4.266
## `Type of Vehicle`Bus or coach (17 or more passenger seats)          19.172
## `Type of Vehicle`Car                                                18.826
## `Type of Vehicle`Goods vehicle 3.5 tonnes mgw and under             18.996
## `Type of Vehicle`Goods vehicle 7.5 tonnes mgw and over              19.337
## `Type of Vehicle`Goods vehicle over 3.5 tonnes and under 7.5 tonnes mgw     19.967
## `Type of Vehicle`M/cycle 50cc and under                             19.302
## `Type of Vehicle`Minibus (8 – 16 passenger seats)                   23.047
## `Type of Vehicle`Mobility Scooter                                   26.602
```

```
## `Type of Vehicle`Motorcycle - Unknown CC                                    26.633
## `Type of Vehicle`Motorcycle over 125cc and up to 500cc                      19.366
## `Type of Vehicle`Motorcycle over 500cc                                      19.015
## `Type of Vehicle`Motorcycle over 50cc and up to 125cc                       18.994
## `Type of Vehicle`Other Vehicle                                              19.969
## `Type of Vehicle`Pedal cycle                                                18.894
## `Type of Vehicle`Ridden horse                                               23.038
## `Type of Vehicle`Taxi/Private hire car                                      18.979
## `Type of Vehicle`Tram / Light rail                                          21.735
## `Weather Conditions`Fine without high winds                                  5.710
## `Weather Conditions`Fog or mist ? if hazard                                  7.888
## `Weather Conditions`Other                                                    7.872
## `Weather Conditions`Raining with high winds                                  6.271
## `Weather Conditions`Raining without high winds                               5.849
## `Weather Conditions`Snowing with high winds                                  9.125
## `Weather Conditions`Snowing without high winds                               7.601
## `Weather Conditions`Unknown                                                  7.478
##                                                                          t value
## (Intercept)                                                                1.359
## `Casualty Class`Pedestrian                                                -4.173
## `Casualty Class`Vehicle or pillion passenger                              -7.924
## `Casualty Severity`Serious                                                 0.051
## `Casualty Severity`Slight                                                 -0.719
## `Type of Vehicle`Bus or coach (17 or more passenger seats)                 1.786
## `Type of Vehicle`Car                                                       1.045
## `Type of Vehicle`Goods vehicle 3.5 tonnes mgw and under                    0.983
## `Type of Vehicle`Goods vehicle 7.5 tonnes mgw and over                     1.689
## `Type of Vehicle`Goods vehicle over 3.5 tonnes and under 7.5 tonnes mgw    1.829
## `Type of Vehicle`M/cycle 50cc and under                                    0.304
## `Type of Vehicle`Minibus (8 – 16 passenger seats)                          0.486
## `Type of Vehicle`Mobility Scooter                                          1.128
## `Type of Vehicle`Motorcycle - Unknown CC                                   2.268
## `Type of Vehicle`Motorcycle over 125cc and up to 500cc                     0.824
## `Type of Vehicle`Motorcycle over 500cc                                     1.165
## `Type of Vehicle`Motorcycle over 50cc and up to 125cc                      0.373
## `Type of Vehicle`Other Vehicle                                             0.890
## `Type of Vehicle`Pedal cycle                                               0.784
## `Type of Vehicle`Ridden horse                                              0.477
## `Type of Vehicle`Taxi/Private hire car                                     1.122
## `Type of Vehicle`Tram / Light rail                                         0.387
## `Weather Conditions`Fine without high winds                               -0.751
## `Weather Conditions`Fog or mist ? if hazard                                0.334
## `Weather Conditions`Other                                                 -1.996
## `Weather Conditions`Raining with high winds                               -1.180
## `Weather Conditions`Raining without high winds                            -1.212
## `Weather Conditions`Snowing with high winds                               -0.632
## `Weather Conditions`Snowing without high winds                            -0.304
## `Weather Conditions`Unknown                                               -1.180
##                                                                        Pr(>|t|)
## (Intercept)                                                              0.1743
## `Casualty Class`Pedestrian                                             3.16e-05
## `Casualty Class`Vehicle or pillion passenger                           4.22e-15
## `Casualty Severity`Serious                                               0.9594
## `Casualty Severity`Slight                                                0.4723
## `Type of Vehicle`Bus or coach (17 or more passenger seats)               0.0744
## `Type of Vehicle`Car                                                     0.2960
```

```
## `Type of Vehicle`Goods vehicle 3.5 tonnes mgw and under                    0.3259
## `Type of Vehicle`Goods vehicle 7.5 tonnes mgw and over                     0.0915
## `Type of Vehicle`Goods vehicle over 3.5 tonnes and under 7.5 tonnes mgw    0.0676
## `Type of Vehicle`M/cycle 50cc and under                                    0.7609
## `Type of Vehicle`Minibus (8 – 16 passenger seats)                          0.6271
## `Type of Vehicle`Mobility Scooter                                          0.2596
## `Type of Vehicle`Motorcycle - Unknown CC                                   0.0235
## `Type of Vehicle`Motorcycle over 125cc and up to 500cc                     0.4100
## `Type of Vehicle`Motorcycle over 500cc                                     0.2442
## `Type of Vehicle`Motorcycle over 50cc and up to 125cc                      0.7090
## `Type of Vehicle`Other Vehicle                                             0.3733
## `Type of Vehicle`Pedal cycle                                               0.4333
## `Type of Vehicle`Ridden horse                                              0.6331
## `Type of Vehicle`Taxi/Private hire car                                     0.2620
## `Type of Vehicle`Tram / Light rail                                         0.6992
## `Weather Conditions`Fine without high winds                                0.4527
## `Weather Conditions`Fog or mist ? if hazard                                0.7386
## `Weather Conditions`Other                                                  0.0461
## `Weather Conditions`Raining with high winds                                0.2381
## `Weather Conditions`Raining without high winds                             0.2259
## `Weather Conditions`Snowing with high winds                                0.5274
## `Weather Conditions`Snowing without high winds                             0.7614
## `Weather Conditions`Unknown                                                0.2383
##
## (Intercept)
## `Casualty Class`Pedestrian                                               ***
## `Casualty Class`Vehicle or pillion passenger                             ***
## `Casualty Severity`Serious
## `Casualty Severity`Slight
## `Type of Vehicle`Bus or coach (17 or more passenger seats)               .
## `Type of Vehicle`Car
## `Type of Vehicle`Goods vehicle 3.5 tonnes mgw and under
## `Type of Vehicle`Goods vehicle 7.5 tonnes mgw and over                   .
## `Type of Vehicle`Goods vehicle over 3.5 tonnes and under 7.5 tonnes mgw  .
## `Type of Vehicle`M/cycle 50cc and under
## `Type of Vehicle`Minibus (8 – 16 passenger seats)
## `Type of Vehicle`Mobility Scooter
## `Type of Vehicle`Motorcycle - Unknown CC                                 *
## `Type of Vehicle`Motorcycle over 125cc and up to 500cc
## `Type of Vehicle`Motorcycle over 500cc
## `Type of Vehicle`Motorcycle over 50cc and up to 125cc
## `Type of Vehicle`Other Vehicle
## `Type of Vehicle`Pedal cycle
## `Type of Vehicle`Ridden horse
## `Type of Vehicle`Taxi/Private hire car
## `Type of Vehicle`Tram / Light rail
## `Weather Conditions`Fine without high winds
## `Weather Conditions`Fog or mist ? if hazard
## `Weather Conditions`Other                                                *
## `Weather Conditions`Raining with high winds
## `Weather Conditions`Raining without high winds
## `Weather Conditions`Snowing with high winds
## `Weather Conditions`Snowing without high winds
## `Weather Conditions`Unknown
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 18.81 on 1626 degrees of freedom
## Multiple R-squared:  0.0825, Adjusted R-squared:  0.06613
## F-statistic: 5.041 on 29 and 1626 DF,  p-value: < 2.2e-16
```

Task: Using the trained model to predict and replace missing values in the “Age of Casualty” column.

```
# Get the rows with missing `Age of Casualty`
missing_age_data <- data %>% filter(is.na(`Age of Casualty`))

# Predict missing values
predicted_age <- predict(model, newdata = missing_age_data)

# Replace the missing values with predicted values
data <- data %>%
  mutate(`Age of Casualty` = ifelse(is.na(`Age of Casualty`), predicted_age, `Age of Casualty
`))
```

Task:

```
write_csv(data, "regression.csv")
```