ST565 Final Project Proposal & Early Insights

Brian Cervantes Alvarez
March 8, 2024

Introduction

The primary inquiry of this final project for the Time Series course is to explore whether there are discernible seasonal patterns in the severity of road traffic accidents and how these patterns vary across different district areas within the United Kingdom. Given the substantial public safety concerns and socio-economic repercussions associated with road traffic accidents in the UK, this research aims to harness time series analysis for delineating temporal trends and uncovering patterns in road accident data from diverse districts like Adur, York, Leeds, and Cardiff.

The investigation will delve into the extent to which seasonal factors, weather conditions, variations in daylight hours, and commuting behaviors influence the frequency and severity of road accidents within these regions. The ultimate objective is to unearth practical insights that could support the development of localized, time-sensitive road safety strategies. These strategies are envisioned to enhance community well-being by reducing road accident incidences through improved road maintenance timing, enhanced street lighting, and targeted public awareness initiatives.

Research Dimensions:

1. **Seasonal and Weather-Induced Variations:** This section will evaluate how seasonal shifts and distinct weather patterns specific to the UK affect road accident severities in varied districts, aiming to craft area-specific safety protocols that address these environmental factors.
2. **Impact of Daylight and Commuting Dynamics:** This part will assess the interplay between daylight changes, standard commuting times in the UK, and the occurrence of road accidents. Identifying high-risk timeframes will inform recommendations for augmenting road visibility and safety, particularly during peak travel times and seasons with limited daylight, adapting to the unique commuting landscapes and characteristics of different UK districts.

Part 1: Exploratory Data Analysis

Loading and Merging Datasets

```r
# Load necessary libraries for data manipulation and visualization
library(tidyverse)

# Reading datasets from various CSV files for accident analysis
urbanRuralArea <- read_csv("ubar_rural_area.csv")
vehicleType <- read_csv("vehicule_type.csv")
weatherConditions <- read_csv("wheather_condictions_accident.csv")
vehiclesInCollision <- read_csv("number-vehicules_accidentes.csv")
roadSurfaceCondition <- read_csv("road-surface-condictions_accident.csv")
roadType <- read_csv("road_type- corrigir G-sheets.csv")
longitude <- read_csv("longitutde_accident.csv")
latitude <- read_csv("latitutde_accident.csv")
darknessLevel <- read_csv("dark-light_accident.csv")
accidentDate <- read_csv("data_accident.csv")
districtArea <- read_csv("disctrit-area_accident.csv")
accidentSeverity <- read_csv("accident_fatal.csv")

# List of datasets
datasets <- list(
  urbanRuralArea, vehicleType, weatherConditions, vehiclesInCollision,
  roadSurfaceCondition, roadType, longitude, latitude, darknessLevel,
  accidentDate, districtArea, accidentSeverity
)

# Add index to each dataset
for (i in seq_along(datasets)) {
  datasets[[i]] <- mutate(datasets[[i]], index = row_number())
}

# Merge datasets
accidentDs <- Reduce(function(x, y) full_join(x, y, by = 'index'), datasets)

# Remove index column
accidentDs <- dplyr::select(accidentDs, -index)


# Optional: Save the fully joined dataset to a new CSV file
# write_csv(accidentDs, "fullAccidentRecord.csv")
```

Missing Value Investigation (To be explored more regarding MAR|MNAR on Data Quality Section Below)

```
accidentDs <- read_csv("fullAccidentRecord.csv")

# Check the first few rows to understand the structure of the combined dataset
head(accidentDs, 5)
```

```
# A tibble: 5 x 12
  Accident_Severity Urban_or_Rural_Area Vehicle_Type          Weather_Conditions
  <chr>             <chr>               <chr>                 <chr>
1 Fatal             Unallocated         Minibus (8 - 16 pass~ Snowing + high wi~
2 Fatal             Unallocated         Agricultural vehicle  Snowing + high wi~
3 Fatal             Unallocated         Agricultural vehicle  Snowing + high wi~
4 Fatal             Rural               Agricultural vehicle  Snowing + high wi~
5 Fatal             Rural               Minibus (8 - 16 pass~ Snowing + high wi~
# i 8 more variables: Number_of_Vehicles <dbl>, Road_Surface_Conditions <chr>,
#   Road_Type <chr>, Longitude <dbl>, Latitude <dbl>, Light_Conditions <chr>,
#   Accident_Date <date>, District_Area <chr>
```

```
# Calculate the total missing values for each column in the dataset
missingValues <- accidentDs %>%
  summarise_all(~sum(is.na(.)))

# Display the missing value count for each column
print(missingValues)
```

```
# A tibble: 1 x 12
  Accident_Severity Urban_or_Rural_Area Vehicle_Type Weather_Conditions
              <int>               <int>        <int>              <int>
1                 0                  15         3305             236934
# i 8 more variables: Number_of_Vehicles <int>, Road_Surface_Conditions <int>,
#   Road_Type <int>, Longitude <int>, Latitude <int>, Light_Conditions <int>,
#   Accident_Date <int>, District_Area <int>
```

```
# Filtering out incomplete rows except for 'Weather_Conditions' and 'Road_Type'
columns_to_check <- setdiff(names(accidentDs),
                            c("Weather_Conditions", "Road_Type"))
index_na <- complete.cases(accidentDs[columns_to_check])
accidentDs <- accidentDs[index_na, ]

# Recheck missing values after initial cleaning
missingValues <- accidentDs %>%
```

```
  summarise_all(~sum(is.na(.)))
print(missingValues)
```

```
# A tibble: 1 x 12
  Accident_Severity Urban_or_Rural_Area Vehicle_Type Weather_Conditions
              <int>               <int>        <int>              <int>
1                 0                   0            0             179903
# i 8 more variables: Number_of_Vehicles <int>, Road_Surface_Conditions <int>,
#   Road_Type <int>, Longitude <int>, Latitude <int>, Light_Conditions <int>,
#   Accident_Date <int>, District_Area <int>
```

```
# Validate the dataset post-cleaning for specific groups, excluding
# 'Weather_Conditions' and 'Road_Type'
accidentDs %>%
  group_by(Accident_Severity) %>%
  summarise(cases = n())
```

```
# A tibble: 3 x 2
  Accident_Severity  cases
  <chr>              <int>
1 Fatal               8661
2 Serious            31185
3 Slight            563792
```

```
accidentDs %>%
  group_by(Urban_or_Rural_Area) %>%
  summarise(cases = n())
```

```
# A tibble: 3 x 2
  Urban_or_Rural_Area  cases
  <chr>                <int>
1 Rural               238989
2 Unallocated              3
3 Urban               364646
```

```
accidentDs %>%
  group_by(Vehicle_Type) %>%
  summarise(cases = n())
```

```
# A tibble: 14 x 2
   Vehicle_Type                          cases
   <chr>                                 <int>
```

```
 1 Agricultural vehicle                  1947
 2 Car                                 497986
 3 Data missing or out of range             6
 4 Goods 7.5 tonnes mgw and over        17307
 5 Goods over 3.5t. and under 7.5t       6096
 6 Minibus (8 - 16 passenger seats)      1976
 7 Motorcycle 125cc and under           15269
 8 Motorcycle 50cc and under             7603
 9 Motorcycle over 500cc                25657
10 Other vehicle                         5637
11 Pedal cycle                            197
12 Ridden horse                             4
13 Taxi/Private hire car                13293
14 Van / Goods 3.5 tonnes mgw or under  10660
```

```
accidentDs %>%
  group_by(Road_Surface_Conditions) %>%
  summarise(cases = n())
```

```
# A tibble: 4 x 2
  Road_Surface_Conditions  cases
  <chr>                    <int>
1 Dry                     447815
2 Flood over 3cm. deep      1016
3 Snow                      5890
4 Wet or damp             148917
```

```
accidentDs %>%
  group_by(Light_Conditions) %>%
  summarise(cases = n())
```

```
# A tibble: 3 x 2
  Light_Conditions         cases
  <chr>                    <int>
1 Darkness - lights lit   116223
2 Darkness - lights unlit   2543
3 Daylight                484872
```

```
accidentDs %>%
  group_by(District_Area) %>%
  summarise(cases = n())
```

```
# A tibble: 385 x 2
   District_Area   cases
   <chr>           <int>
 1 Aberdeen City    1323
 2 Aberdeenshire    1930
 3 Adur              619
 4 Allerdale        1128
 5 Alnwick           232
 6 Amber Valley     1347
 7 Angus             796
 8 Argyll and Bute   836
 9 Arun             1376
10 Ashfield         1395
# i 375 more rows
```

```r
# Exclude 'Weather_Conditions' and 'Road_Type' from
# further analysis due to significant missing values
accidentDs <- accidentDs %>%
  dplyr::select(-c(Weather_Conditions, Road_Type))

# Recalculate and print missing values to ensure cleanliness of the data
missingValues <- accidentDs %>%
  summarise_all(~sum(is.na(.)))
print(missingValues)
```

```
# A tibble: 1 x 10
  Accident_Severity Urban_or_Rural_Area Vehicle_Type Number_of_Vehicles
              <int>               <int>        <int>              <int>
1                 0                   0            0                  0
# i 6 more variables: Road_Surface_Conditions <int>, Longitude <int>,
#   Latitude <int>, Light_Conditions <int>, Accident_Date <int>,
#   District_Area <int>
```

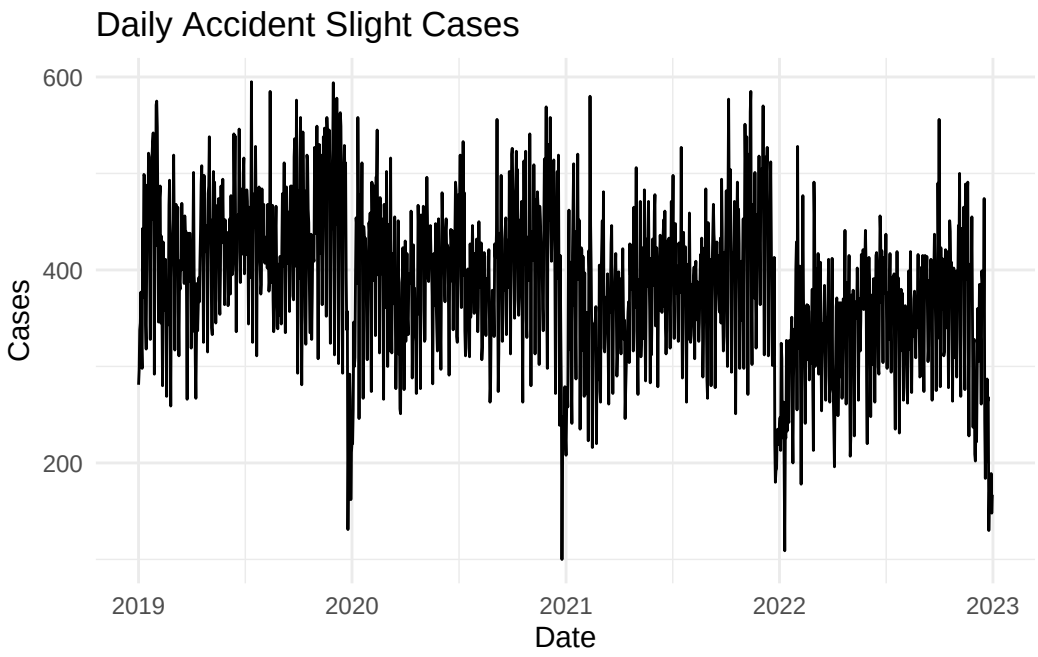
Data Wrangling

```
library(lubridate)

# Clean the dataset by filtering out irrelevant rows
# and converting certain columns to factors
accidentDs <- accidentDs %>%
  filter(Urban_or_Rural_Area != "Unallocated",
         Vehicle_Type != "Data missing or out of range") %>%
  mutate(Urban_or_Rural_Area = factor(Urban_or_Rural_Area),
         Vehicle_Type = factor(Vehicle_Type),
         Accident_Severity = factor(Accident_Severity),
         Road_Surface_Conditions = factor(Road_Surface_Conditions),
         Light_Conditions = factor(Light_Conditions),
         District_Area = factor(District_Area))


# Ensure Accident_Date is a Data class,
# Add Month and Week Dates
accidentDs <- accidentDs %>%
  mutate(Accident_Date = as.Date(Accident_Date),
         Month = floor_date(Accident_Date, "month"),
         Week = floor_date(Accident_Date, "week"))
```

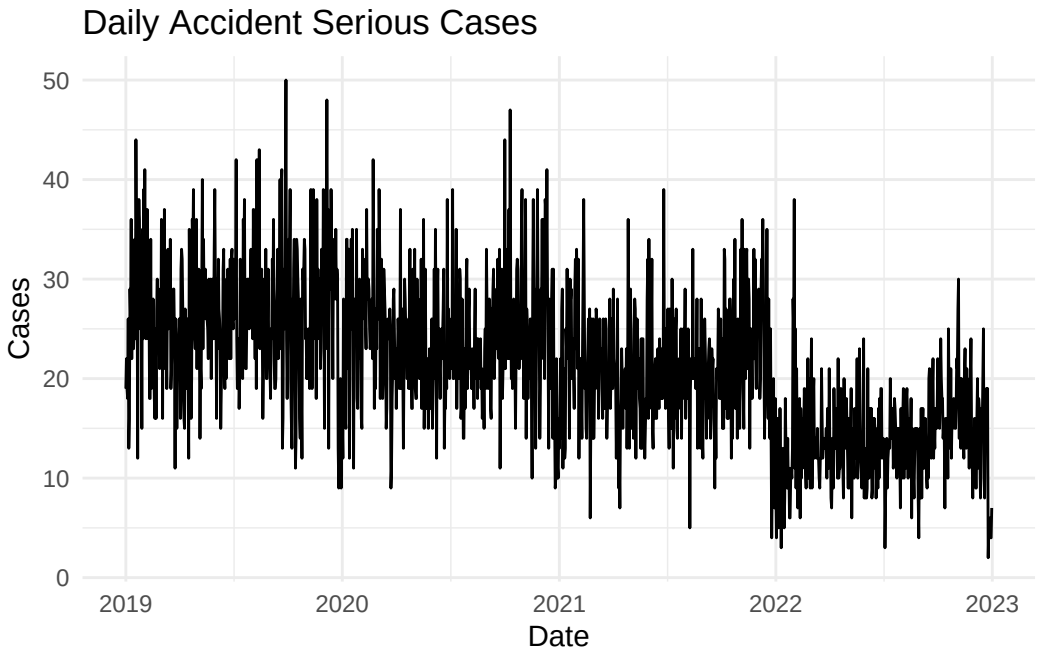
Visualizing Severity Trends

Looking at Daily Accident Severity Time Series Plots

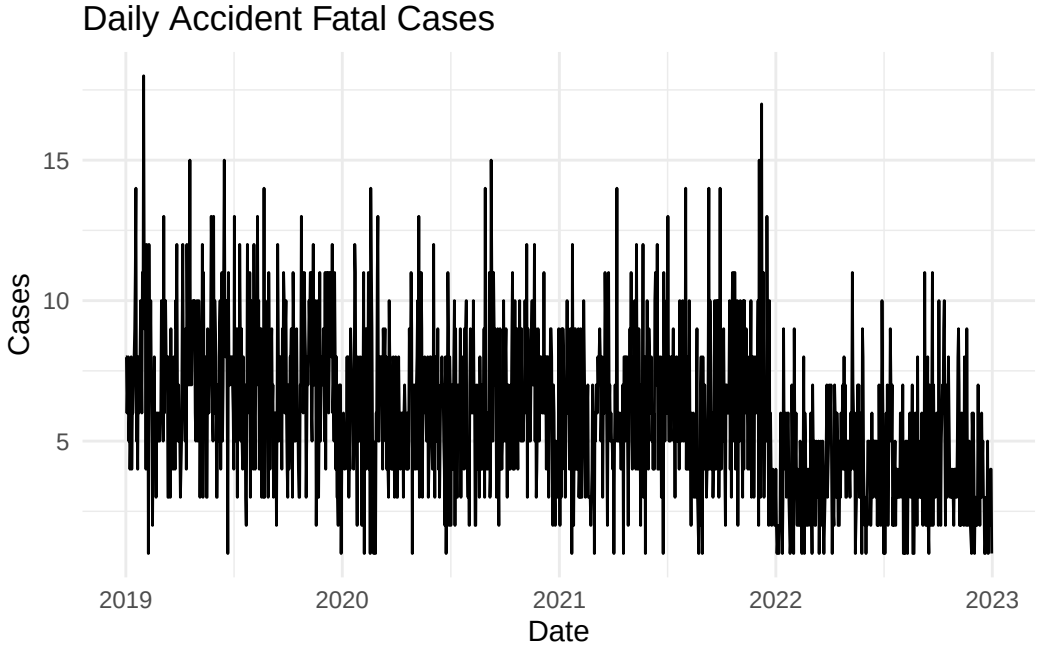
```r
# Analyze daily accident severity trends to identify
# any daily patterns or anomalies
accidentDs %>%
  filter(Accident_Severity == "Slight") %>%
  group_by(Accident_Severity, Accident_Date) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(x = Accident_Date, y = Count)) +
  geom_line() +
  labs(title = "Daily Accident Slight Cases",
       x = "Date",
       y = "Cases") +
  theme_minimal()
```




```r
accidentDs %>%
  filter(Accident_Severity == "Serious") %>%
  group_by(Accident_Severity, Accident_Date) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(x = Accident_Date, y = Count)) +
  geom_line() +
  labs(title = "Daily Accident Serious Cases",
       x = "Date",
       y = "Cases")  +
  theme_minimal()
```

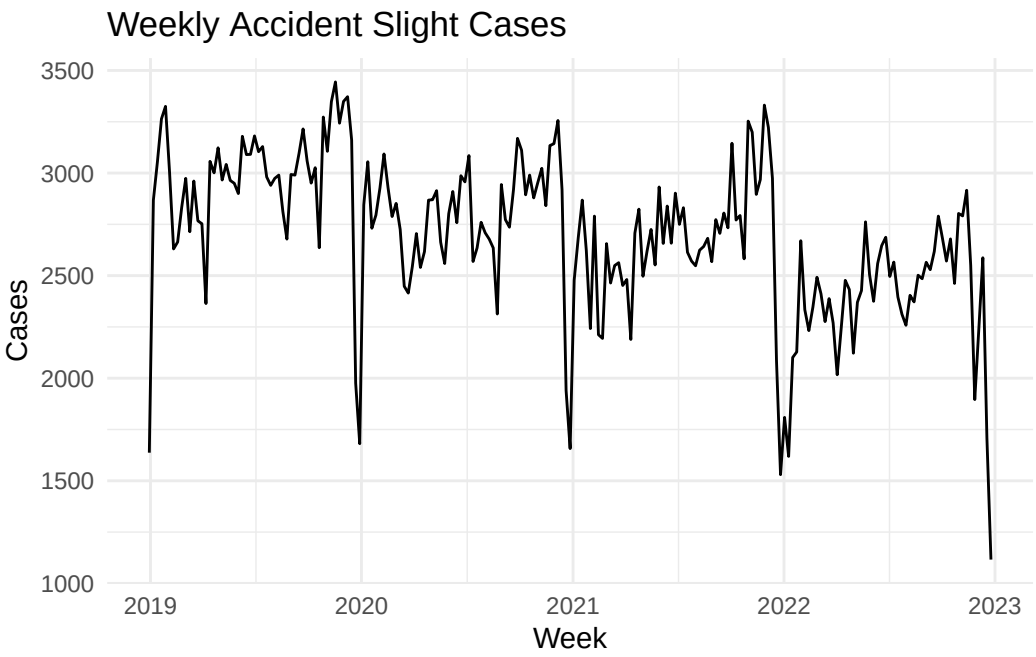



```
accidentDs %>%
  filter(Accident_Severity == "Fatal") %>%
  group_by(Accident_Severity, Accident_Date) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(x = Accident_Date, y = Count)) +
  geom_line() +
  labs(title = "Daily Accident Fatal Cases",
       x = "Date",
       y = "Cases")  +
  theme_minimal()
```

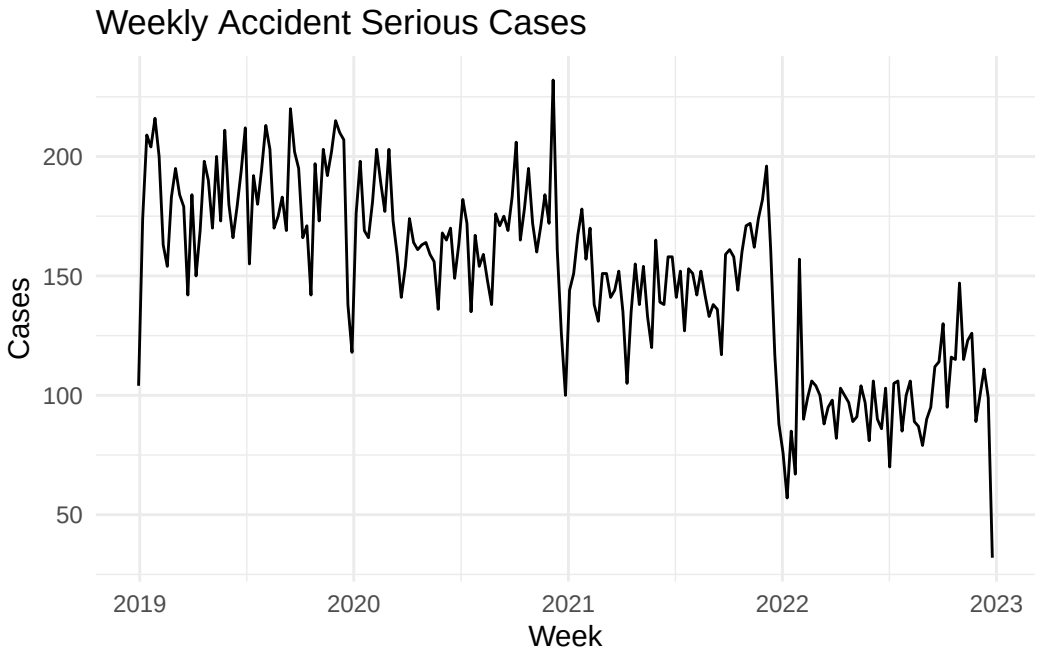


Looking at Weekly Accident Severity Time Series Plots

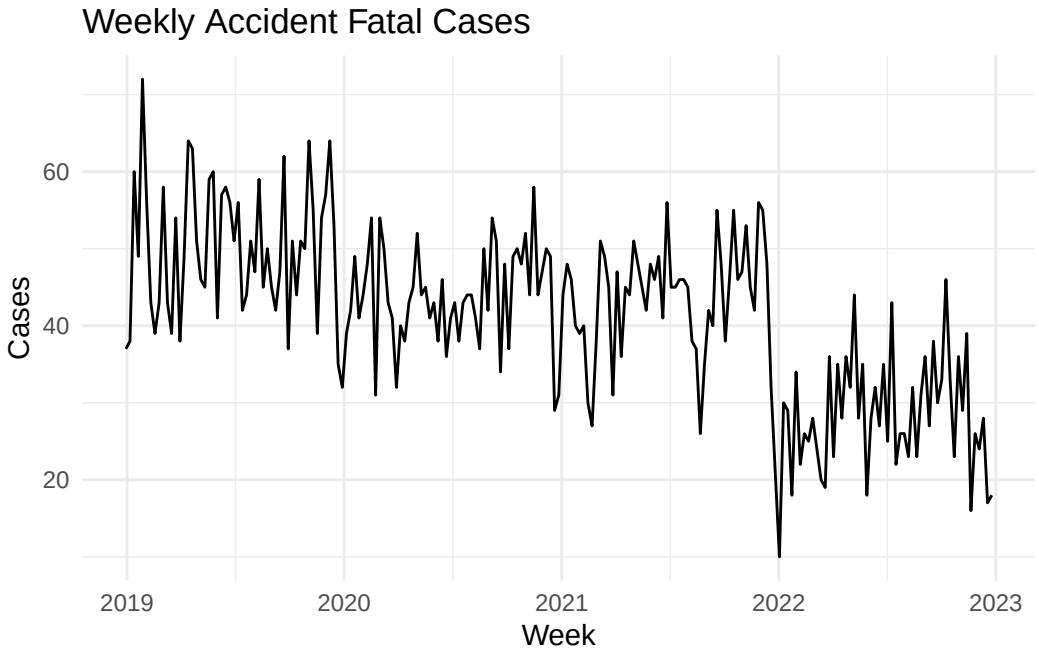
```r
# Plot weekly trends for each accident severity category
accidentDs %>%
  filter(Accident_Severity == "Slight") %>%
  group_by(Accident_Severity, Week) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(x = Week, y = Count)) +
  geom_line() +
  labs(title = "Weekly Accident Slight Cases",
       x = "Week",
       y = "Cases") +
  theme_minimal()
```




```r
accidentDs %>%
  filter(Accident_Severity == "Serious") %>%
  group_by(Accident_Severity, Week) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(x = Week, y = Count)) +
  geom_line() +
  labs(title = "Weekly Accident Serious Cases",
       x = "Week",
       y = "Cases")  +
  theme_minimal()
```

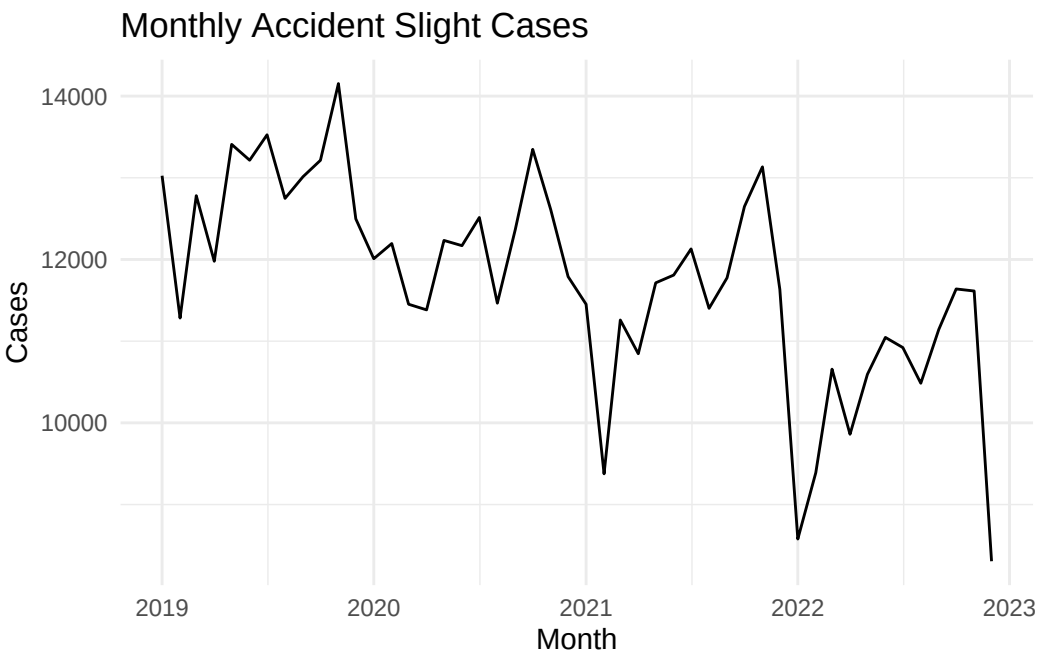



```
accidentDs %>%
  filter(Accident_Severity == "Fatal") %>%
  group_by(Accident_Severity, Week) %>%
  summarise(Count = n(), .groups = 'drop') %>%


  ggplot(aes(x = Week, y = Count)) +
  geom_line() +
  labs(title = "Weekly Accident Fatal Cases",
       x = "Week",
       y = "Cases")  +
  theme_minimal()
```



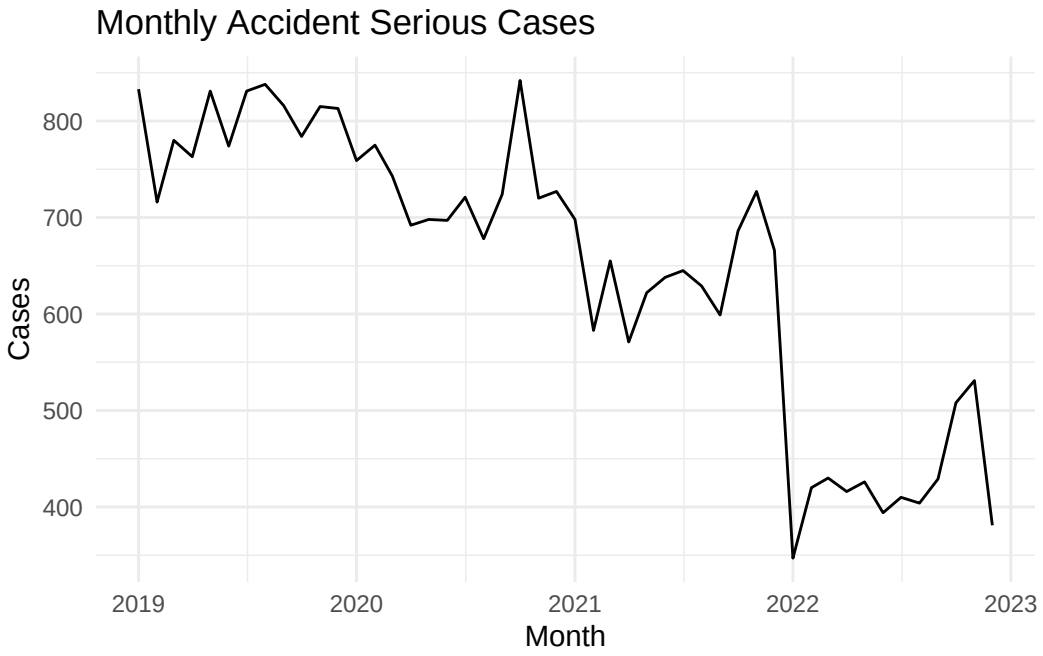
Looking at Monthly Accident Severity Time Series Plots

```r
# Load the lubridate package for date-time manipulation
library(lubridate)

# Enhance the dataset with month and week columns for time series analysis
accidentDs <- accidentDs %>%
  mutate(Accident_Date = as.Date(Accident_Date),
         Month = floor_date(Accident_Date, "month"),
         Week = floor_date(Accident_Date, "week"))

# Plot monthly trends for each accident severity category
accidentDs %>%
  filter(Accident_Severity == "Slight") %>%
  group_by(Accident_Severity, Month) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(x = Month, y = Count)) +
  geom_line() +
  labs(title = "Monthly Accident Slight Cases",
       x = "Month",
       y = "Cases") +
  theme_minimal()
```

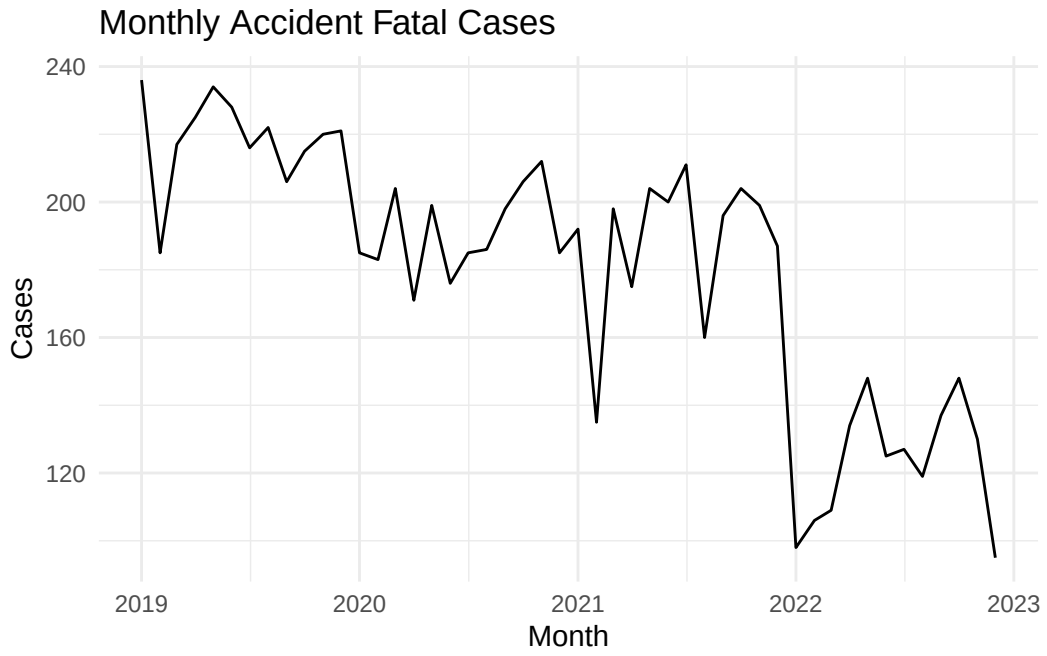



```r
accidentDs %>%
  filter(Accident_Severity == "Serious") %>%
  group_by(Accident_Severity, Month) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(x = Month, y = Count)) +
```

```
  geom_line() +
  labs(title = "Monthly Accident Serious Cases",
       x = "Month",
       y = "Cases")  +
  theme_minimal()
```




```
accidentDs %>%
  filter(Accident_Severity == "Fatal") %>%
  group_by(Accident_Severity, Month) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(x = Month, y = Count)) +
  geom_line() +
  labs(title = "Monthly Accident Fatal Cases",
       x = "Month",
       y = "Cases")  +
  theme_minimal()
```




```
districts <- unique(accidentDs$District_Area)
head(districts, 5)
```

```
[1] Adur Arun Bury Eden Fife
385 Levels: Aberdeen City Aberdeenshire Adur Allerdale Alnwick ... York
```

Part 2: Time Series Analysis

```
names(accidentDs)
```

```
 [1] "Accident_Severity"       "Urban_or_Rural_Area"
 [3] "Vehicle_Type"            "Number_of_Vehicles"
 [5] "Road_Surface_Conditions" "Longitude"
 [7] "Latitude"                "Light_Conditions"
 [9] "Accident_Date"           "District_Area"
[11] "Month"                   "Week"
```

Data Quality Assessment

Assessing Completeness and Plausibility

Stationarity and Seasonality Checks

Testing for Stationarity

Seasonality Analysis

Autocorrelation Analysis

Lag Analysis

Model Selection and Validation

Model Fit

Cross-Validation

Error Analysis

Residual Diagnostics

Robustness Checks and Sensitivity Analysis

Scenario Analysis

Reference

Gibin, W. O., & Sheen. (2023). ***Road Accident Casualties***. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/7292741