# Final Project: Road Accidents in U.K.

Sanjay Jaras

02/12/2020

## Contents



## Section 1 – Week 9

- Introduction

  Accidents data can be used for numerous applications such as real-time accident prediction, studying accident hotspot locations, casualty analysis and extracting cause and effect rules to predict accidents, and studying the impact of vehicle age, road conditions, speed limits, environmental stimuli and road conditions on accident occurrence. This dataset contains the data of road accidents happened in U.K. within the time frame of 2010-2014. The data is very extensive, so it can give many insights on accidents. It contains location information, vehicle information, weather information, driver information, time of accidents etc.

- Research questions

  1. What are the factors those are more correlated to severity of accidents?
  2. Is vehicle power something to do with accidents?
  3. Is a particular day of time, when accidents happens more?
  4. Is number of accidents vary by road types?
  5. Is accidents increase in winter season ?

- Approach

  If required the data will be normalized and cleaned. If null values are present in data, I need to take care of them by either removing those reocrds or using mean value from that column. After cleaning the data, I will analyze data to try answering research questions. While analyzing I will try to use graphs to support/better understand the data.

- 

  **How your approach addresses (fully or partially) the problem.**

- Data

  Source Link:

  https://www.kaggle.com/stefanoleone992/adm-project-road-accidents-in-uk

  Columns in the dataset:

  1. Accident_Index: Accident index
  2. Latitude: Accident latitude
  3. Longitude: Accident longitude
  4. Region: Accident region
  5. Urban_or_Rural_Area: Accident area (rural or urban)
  6. X1st_Road_Class: Accident road class
  7. Driver_IMD_Decile: Road IMD Decile
  8. Speed_limit: Road speed
  9. Road_Type: Road type
  10. Road_Surface_Conditions: Road surface condition
  11. Weather: Weather
  12. High_Wind: High wind
  13. Lights: Road lights
  14. Datetime: Accident datetime
  15. Year: Accident year
  16. Season: Accident season
  17. Month_of_Year: Accident month
  18. Day_of_Month: Accident day of month
  19. Day_of_Week: Accident day of week
  20. Hour_of_Day: Accident hour of day
  21. Number_of_Vehicles: Accident number of vehicles
  22. Age_of_Driver: Driver age
  23. Age_of_Vehicle: Vehicle age
  24. Junction_Detail: Accident junction detail
  25. Junction_Location: Accident junction location
  26. X1st_Point_of_Impact: Vehicle first point of impact
  27. Driver_Journey_Purpose: Driver journey purpose
  28. Engine_CC: Vehicle engine power (in CC)
  29. Propulsion_Code: Vehicle propulsion code
  30. Vehicle_Make: Vehicle make

31. Vehicle_Category: Vehicle category
32. Vehicle_Manoeuvre: Vehicle manoeuvre
33. Accident_Severity: Accident severity

    This data is from 2010 through 2014. The dataset is very extensive with location information, vehicle information, weather information, driver information, time of accidents.

- Required Packages

  I will be using below packages for my analysis: ggplot2, car, dplyr, tidyr, broom, corrplot, fastDummies, caret

- Plots and Table Needs

  I will be using scatter plots, time-series plot and histograms to analyze and visualize the data patterns.

- Questions for future steps.

  The dataset is having lot of information, currently I am not sure if I can create a plot with map of all accidents. It can help us to find out if there is one particular region where accidents happened most.

# Section 2 – Week 10

- 

  **How to import my data ?**

  Load dataset into data frame

  load adm-project-road-accidents-in-uk.csv data into data frame

```
df <- read.csv("adm-project-road-accidents-in-uk.csv")
str(df)
```

```
## 'data.frame':    251832 obs. of  33 variables:
##  $ Accident_Index         : Factor w/ 210056 levels "201001BS70015",..: 1 5 6 8 10 11 12 15 15 16 ..
##  $ Latitude               : num  51.5 51.5 51.5 51.5 51.5 ...
##  $ Longitude              : num  -0.178 -0.169 -0.179 -0.196 -0.208 ...
##  $ Region                 : Factor w/ 11 levels "East England",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Urban_or_Rural_Area    : Factor w/ 2 levels "Rural","Urban": 2 2 2 2 2 2 2 2 2 2 ...
##  $ X1st_Road_Class        : Factor w/ 6 levels "A","A(M)","B",..: 1 3 4 1 3 1 6 1 1 3 ...
##  $ Driver_IMD_Decile      : int  2 8 7 7 5 3 5 2 4 3 ...
##  $ Speed_limit            : int  30 30 30 30 30 30 30 30 30 30 ...
##  $ Road_Type              : Factor w/ 5 levels "Dual carriageway",..: 1 4 4 4 4 4 4 4 4 4 ...
##  $ Road_Surface_Conditions: Factor w/ 5 levels "Dry","Flood over 3cm. deep",..: 5 1 1 5 5 1 1 1 1 5
##  $ Weather                : Factor w/ 6 levels "Fine","Fog or mist",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ High_Wind              : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Lights                 : Factor w/ 4 levels "Darkness - lighting unknown",..: 2 4 4 2 2 4 4 4 4 4
##  $ Datetime               : Factor w/ 182109 levels "2010-01-01 00:01:00",..: 553 1890 1448 2208 318
##  $ Year                   : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ Season                 : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Month_of_Year          : int  1 1 1 1 2 2 3 3 3 3 ...
##  $ Day_of_Month           : int  7 24 19 27 5 8 3 4 4 12 ...
##  $ Day_of_Week            : int  4 7 2 3 5 1 3 4 4 5 ...
##  $ Hour_of_Day            : num  0.899 0.521 0.729 0.76 0.257 0.475 0.267 0.566 0.566 0.67 ...
##  $ Number_of_Vehicles     : int  2 2 2 1 2 2 2 2 2 1 ...
##  $ Age_of_Driver          : int  4 4 7 3 5 5 5 3 3 4 ...
```

```
##  $ Age_of_Vehicle         : int  8 3 8 2 12 2 11 5 1 4 ...
##  $ Junction_Detail        : Factor w/ 8 levels "Crossroads","More than 4 arms (not roundabout)",..: 
##  $ Junction_Location      : Factor w/ 9 levels "Approaching junction or waiting/parked at junction ap
##  $ X1st_Point_of_Impact   : Factor w/ 5 levels "Back","Did not impact",..: 3 3 3 5 3 5 3 5 4 4 ...
##  $ Driver_Journey_Purpose : Factor w/ 5 levels "Commuting to/from work",..: 3 3 3 3 3 3 2 2 3 2 ...
##  $ Engine_CC              : int  1896 599 1781 649 600 2987 998 2179 108 2198 ...
##  $ Propulsion_Code        : Factor w/ 2 levels "Heavy oil","Petrol": 1 2 2 2 2 1 2 1 2 1 ...
##  $ Vehicle_Make           : Factor w/ 25 levels "Audi","BMW","Citroen",..: 23 6 1 14 20 11 13 3 6 5
##  $ Vehicle_Category       : Factor w/ 6 levels "Bus/minibus",..: 5 3 2 3 3 2 2 6 3 6 ...
##  $ Vehicle_Manoeuvre      : Factor w/ 11 levels "Changing lane",..: 2 2 2 3 2 11 2 9 4 4 ...
##  $ Accident_Severity      : Factor w/ 2 levels "Fatal_Serious",..: 2 2 2 2 2 2 2 2 2 2 ...
```

- 

  **How and why to clean data?**

  Data cleansing: In this process we go through all the data and either remove or update the information that is incorrect, duplicate or incomplete. Data cleansing is important because it will lead wrong conclusions, decisions and wrong analysis. Many a times data cannot be used as it is and needs preparation in a way so that it can be used. Data cleansing also involves filtering of irrelavant data We have two options to correct or add the missing incomplete data in numerical data, either remove the row or put mean value of that column.

Check for NA values available in data

```
any(is.na(df))
```

```
## [1] FALSE
```

In this dataset, no records with NA's are available, otherwise we would have to replace them by mean value or remove those rows as mentioned above.

Data is normalized: e.g. hours of day data is converted to 0 to 1 range ny using Min-Max normalization. Driver age is transformed to from range of 1-10 by using Unit vector normalization.Regression and neural networks are insensitive to standardization. Advantages of standardization are as follows.

It improves the numerical stability of model

It may speed up the training process

It gives equal considerations for each feature.

All non numeric features are converted to factors.

```
summary(df$Hour_of_Day)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0010  0.4340  0.6180  0.5895  0.7420  0.9990
```

```
summary(df$Age_of_Driver)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   3.000   4.000   3.903   5.000   8.000
```

Each factor column can be further split into mulitple columns with each factor type to make dataset tidy. To transform this we can use dummy_cols function from fastDummies package.

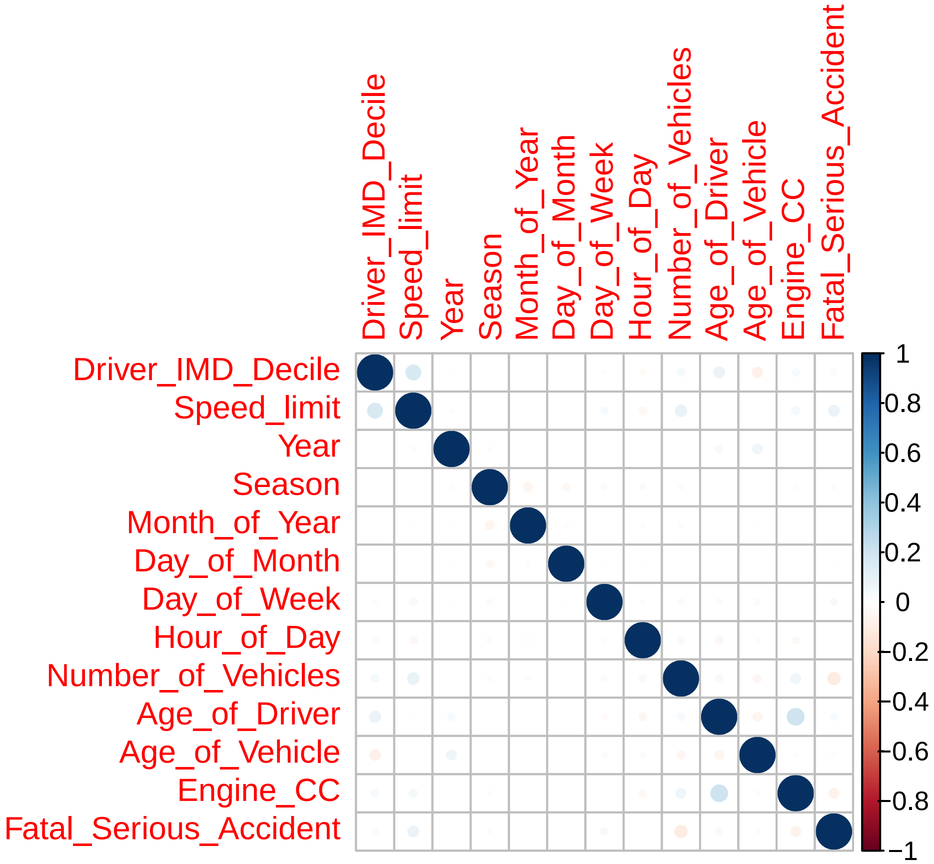
Split factor feature to multiple column for analysis

Create dummy variables with binary values for features with Factors and characters types

```
df_with_dummy <- dummy_cols(df, select_columns = c("Accident_Severity", "Region", "Urban_or_Rural_Area"
```

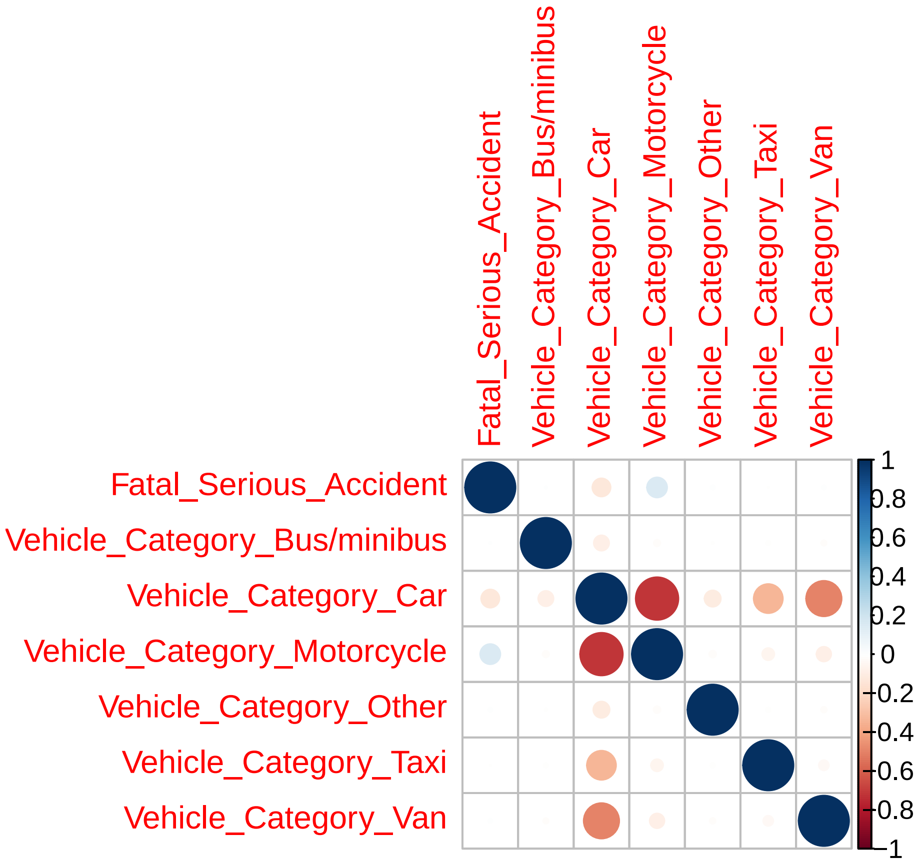
Correlation plot for numeric features

```
num_df <- select_if(df, is.numeric)
num_df <- num_df[, !(names(num_df) %in% c("Latitude", "Longitude"))]
num_df$Fatal_Serious_Accident <- df_with_dummy$Accident_Severity_Fatal_Serious
num_df.cor <- cor(num_df)
corrplot(num_df.cor)
```




Correlation plot for features with factors.

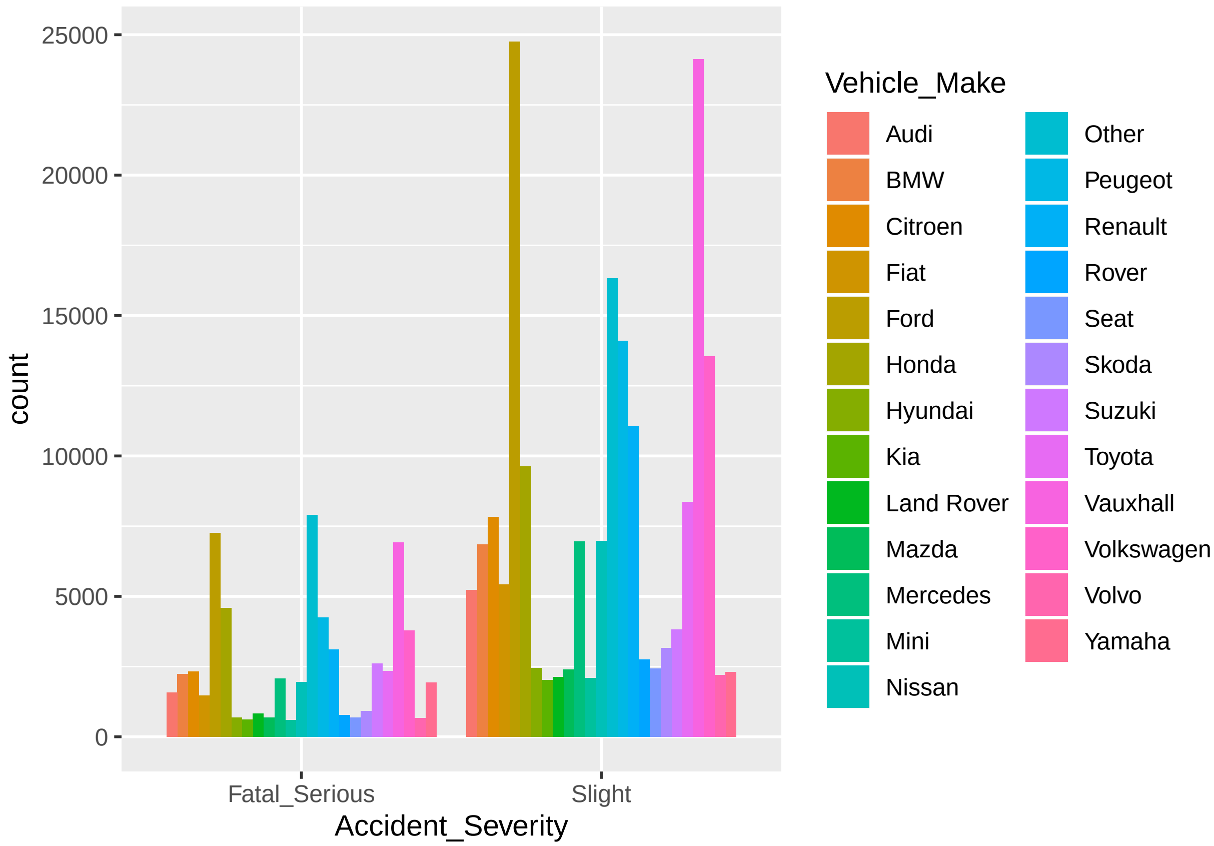
```
cols_from_num_df <- names(num_df)
cols_to_drop <- c(cols_from_num_df, "Datetime", "Accident_Severity", "Region", "Accident_Index", "Latitu
                  "High_Wind", "High_Wind_No", "Lights", "Junction_Detail", "Junction_Location", "X1st_I
df_combined <- cbind(num_df, df_with_dummy[!names(df_with_dummy) %in% cols_to_drop])
cols_for_corr <- c("Fatal_Serious_Accident", "Vehicle_Category_Bus/minibus", "Vehicle_Category_Car", "Ve
corrplot(cor(df_combined[cols_for_corr]), method = "circle")
```




Cleaned Data for prediction model. Remove columns those should not be part of prediction model

```
df_prediction <- as_tibble(df[, !(names(df) %in% c("Accident_Index", "Latitude", 'Longitude', 'Datetime
ggplot(df_prediction, aes(Accident_Severity, fill = Vehicle_Make)) + geom_histogram(stat = "count", pos
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




- 

**What does the final data set look like?**

Dataset for Exploratory analysis

```
str(df_combined)
```

```
## 'data.frame':    251832 obs. of  125 variables:
##  $ Driver_IMD_Decile                                  : int  2 8 7 7 5 3 5
##  $ Speed_limit                                        : int  30 30 30 30 30
##  $ Year                                               : int  2010 2010 2010
##  $ Season                                             : int  4 4 4 4 4 4 4
##  $ Month_of_Year                                      : int  1 1 1 1 2 2 3
##  $ Day_of_Month                                       : int  7 24 19 27 5 8
##  $ Day_of_Week                                        : int  4 7 2 3 5 1 3
##  $ Hour_of_Day                                        : num  0.899 0.521 0
##  $ Number_of_Vehicles                                 : int  2 2 2 1 2 2 2
##  $ Age_of_Driver                                      : int  4 4 7 3 5 5 5
##  $ Age_of_Vehicle                                     : int  8 3 8 2 12 2 1
##  $ Engine_CC                                          : int  1896 599 1781
##  $ Fatal_Serious_Accident                             : int  0 0 0 0 0 0 0
##  $ Accident_Severity_Fatal_Serious                    : int  0 0 0 0 0 0 0
##  $ Accident_Severity_Slight                           : int  1 1 1 1 1 1 1
##  $ Region_East England                                : int  0 0 0 0 0 0 0
##  $ Region_East Midlands                               : int  0 0 0 0 0 0 0
##  $ Region_London                                      : int  1 1 1 1 1 1 1
##  $ Region_North East England                          : int  0 0 0 0 0 0 0
```

```
##  $ Region_North West England                                            : int  0 0 0 0 0 0 0
##  $ Region_Scotland                                                      : int  0 0 0 0 0 0 0
##  $ Region_South East England                                            : int  0 0 0 0 0 0 0
##  $ Region_South West England                                            : int  0 0 0 0 0 0 0
##  $ Region_Wales                                                         : int  0 0 0 0 0 0 0
##  $ Region_Wast Midlands                                                 : int  0 0 0 0 0 0 0
##  $ Region_Yorkshire and the Humber                                      : int  0 0 0 0 0 0 0
##  $ Urban_or_Rural_Area_Rural                                            : int  0 0 0 0 0 0 0
##  $ Urban_or_Rural_Area_Urban                                            : int  1 1 1 1 1 1 1
##  $ X1st_Road_Class_A                                                    : int  1 0 0 1 0 1 0
##  $ X1st_Road_Class_A(M)                                                 : int  0 0 0 0 0 0 0
##  $ X1st_Road_Class_B                                                    : int  0 1 0 0 1 0 0
##  $ X1st_Road_Class_C                                                    : int  0 0 1 0 0 0 0
##  $ X1st_Road_Class_Motorway                                             : int  0 0 0 0 0 0 0
##  $ X1st_Road_Class_Unclassified                                         : int  0 0 0 0 0 0 1
##  $ Road_Type_Dual carriageway                                           : int  1 0 0 0 0 0 0
##  $ Road_Type_One way street                                             : int  0 0 0 0 0 0 0
##  $ Road_Type_Roundabout                                                 : int  0 0 0 0 0 0 0
##  $ Road_Type_Single carriageway                                         : int  0 1 1 1 1 1 1
##  $ Road_Type_Slip road                                                  : int  0 0 0 0 0 0 0
##  $ Road_Surface_Conditions_Dry                                          : int  0 1 1 0 0 1 1
##  $ Road_Surface_Conditions_Flood over 3cm. deep                         : int  0 0 0 0 0 0 0
##  $ Road_Surface_Conditions_Frost or ice                                 : int  0 0 0 0 0 0 0
##  $ Road_Surface_Conditions_Snow                                         : int  0 0 0 0 0 0 0
##  $ Road_Surface_Conditions_Wet or damp                                  : int  1 0 0 1 1 0 0
##  $ Weather_Fine                                                         : int  1 1 1 0 1 1 1
##  $ Weather_Fog or mist                                                  : int  0 0 0 0 0 0 0
##  $ Weather_Other                                                        : int  0 0 0 1 0 0 0
##  $ Weather_Raining                                                      : int  0 0 0 0 0 0 0
##  $ Weather_Snowing                                                      : int  0 0 0 0 0 0 0
##  $ Weather_Unknown                                                      : int  0 0 0 0 0 0 0
##  $ High_Wind_Yes                                                        : int  0 0 0 0 0 0 0
##  $ Lights_Darkness - lighting unknown                                   : int  0 0 0 0 0 0 0
##  $ Lights_Darkness - lights                                             : int  1 0 0 1 1 0 0
##  $ Lights_Darkness - no lights                                          : int  0 0 0 0 0 0 0
##  $ Lights_Daylight                                                      : int  0 1 1 0 0 1 1
##  $ Junction_Detail_Crossroads                                           : int  1 0 0 0 1 0 1
##  $ Junction_Detail_More than 4 arms (not roundabout)                    : int  0 0 0 0 0 1 0
##  $ Junction_Detail_Not at junction or within 20 metres                  : int  0 0 1 1 0 0 0
##  $ Junction_Detail_Other junction                                       : int  0 0 0 0 0 0 0
##  $ Junction_Detail_Private drive or entrance                            : int  0 0 0 0 0 0 0
##  $ Junction_Detail_Roundabout                                           : int  0 0 0 0 0 0 0
##  $ Junction_Detail_Slip road                                            : int  0 0 0 0 0 0 0
##  $ Junction_Detail_T or staggered junction                              : int  0 1 0 0 0 0 0
##  $ Junction_Location_Approaching junction or waiting/parked at junction approach: int  0 0 0 0 0 0 0
##  $ Junction_Location_Cleared junction or waiting/parked at junction exit       : int  0 0 0 0 0 0 0
##  $ Junction_Location_Entering from slip road                            : int  0 0 0 0 0 0 0
##  $ Junction_Location_Entering main road                                 : int  0 0 0 0 0 0 0
##  $ Junction_Location_Entering roundabout                                : int  0 0 0 0 0 0 0
##  $ Junction_Location_Leaving main road                                  : int  0 0 0 0 0 0 0
##  $ Junction_Location_Leaving roundabout                                 : int  0 0 0 0 0 0 0
##  $ Junction_Location_Mid Junction - on roundabout or on main road       : int  1 1 0 0 1 1 1
##  $ Junction_Location_Not at or within 20 metres of junction             : int  0 0 1 1 0 0 0
##  $ X1st_Point_of_Impact_Back                                            : int  0 0 0 0 0 0 0
```

```
##  $ X1st_Point_of_Impact_Did not impact                                         : int  0 0 0 0 0 0 0
##  $ X1st_Point_of_Impact_Front                                                  : int  1 1 1 0 1 0 1
##  $ X1st_Point_of_Impact_Nearside                                               : int  0 0 0 0 0 0 0
##  $ X1st_Point_of_Impact_Offside                                                : int  0 0 0 1 0 1 0
##  $ Driver_Journey_Purpose_Commuting to/from work                               : int  0 0 0 0 0 0 0
##  $ Driver_Journey_Purpose_Journey as part of work                              : int  0 0 0 0 0 0 1
##  $ Driver_Journey_Purpose_Other/Not known                                      : int  1 1 1 1 1 1 0
##  $ Driver_Journey_Purpose_Pupil riding to/from school                          : int  0 0 0 0 0 0 0
##  $ Driver_Journey_Purpose_Taking pupil to/from school                          : int  0 0 0 0 0 0 0
##  $ Propulsion_Code_Heavy oil                                                   : int  1 0 0 0 0 1 0
##  $ Vehicle_Make_Audi                                                           : int  0 0 1 0 0 0 0
##  $ Vehicle_Make_BMW                                                            : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Citroen                                                        : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Fiat                                                           : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Ford                                                           : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Honda                                                          : int  0 1 0 0 0 0 0
##  $ Vehicle_Make_Hyundai                                                        : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Kia                                                            : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Land Rover                                                     : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Mazda                                                          : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Mercedes                                                       : int  0 0 0 0 0 1 0
##  $ Vehicle_Make_Mini                                                           : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Nissan                                                         : int  0 0 0 0 0 0 1
##  $ Vehicle_Make_Other                                                          : int  0 0 0 1 0 0 0
##  $ Vehicle_Make_Peugeot                                                        : int  0 0 0 0 0 0 0
##  $ Vehicle_Make_Renault                                                        : int  0 0 0 0 0 0 0
##   [list output truncated]
```

Dataset for Prediction model

```r
str(df_prediction)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    251832 obs. of  29 variables:
##  $ Region                 : Factor w/ 11 levels "East England",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Urban_or_Rural_Area    : Factor w/ 2 levels "Rural","Urban": 2 2 2 2 2 2 2 2 2 2 ...
##  $ X1st_Road_Class        : Factor w/ 6 levels "A","A(M)","B",..: 1 3 4 1 3 1 6 1 1 3 ...
##  $ Driver_IMD_Decile      : int  2 8 7 7 5 3 5 2 4 3 ...
##  $ Speed_limit            : int  30 30 30 30 30 30 30 30 30 30 ...
##  $ Road_Type              : Factor w/ 5 levels "Dual carriageway",..: 1 4 4 4 4 4 4 4 4 4 ...
##  $ Road_Surface_Conditions: Factor w/ 5 levels "Dry","Flood over 3cm. deep",..: 5 1 1 5 5 1 1 1 1 5
##  $ Weather                : Factor w/ 6 levels "Fine","Fog or mist",..: 1 1 1 3 1 1 1 1 1 1 ...
##  $ High_Wind              : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Lights                 : Factor w/ 4 levels "Darkness - lighting unknown",..: 2 4 4 2 2 4 4 4 4 4
##  $ Year                   : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ Season                 : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Month_of_Year          : int  1 1 1 1 2 2 3 3 3 3 ...
##  $ Day_of_Month           : int  7 24 19 27 5 8 3 4 4 12 ...
##  $ Day_of_Week            : int  4 7 2 3 5 1 3 4 4 5 ...
##  $ Hour_of_Day            : num  0.899 0.521 0.729 0.76 0.257 0.475 0.267 0.566 0.566 0.67 ...
##  $ Number_of_Vehicles     : int  2 2 2 1 2 2 2 2 2 1 ...
##  $ Age_of_Driver          : int  4 4 7 3 5 5 5 3 3 4 ...
##  $ Age_of_Vehicle         : int  8 3 8 2 12 2 11 5 1 4 ...
##  $ Junction_Detail        : Factor w/ 8 levels "Crossroads","More than 4 arms (not roundabout)",..: 
##  $ Junction_Location      : Factor w/ 9 levels "Approaching junction or waiting/parked at junction ap
##  $ X1st_Point_of_Impact   : Factor w/ 5 levels "Back","Did not impact",..: 3 3 3 5 3 5 3 5 4 4 ...
```

```
##  $ Driver_Journey_Purpose : Factor w/ 5 levels "Commuting to/from work",..: 3 3 3 3 3 3 2 2 3 2 ...
##  $ Engine_CC              : int  1896 599 1781 649 600 2987 998 2179 108 2198 ...
##  $ Propulsion_Code        : Factor w/ 2 levels "Heavy oil","Petrol": 1 2 2 2 2 1 2 1 2 1 ...
##  $ Vehicle_Make           : Factor w/ 25 levels "Audi","BMW","Citroen",..: 23 6 1 14 20 11 13 3 6 5
##  $ Vehicle_Category       : Factor w/ 6 levels "Bus/minibus",..: 5 3 2 3 3 2 2 6 3 6 ...
##  $ Vehicle_Manoeuvre      : Factor w/ 11 levels "Changing lane",..: 2 2 2 3 2 11 2 9 4 4 ...
##  $ Accident_Severity      : Factor w/ 2 levels "Fatal_Serious",..: 2 2 2 2 2 2 2 2 2 2 ...
```

- **Questions for future steps.**

1. How to create plot with geo-spatial coordinates?
2. I am planning to use Nearest Neighbour algorithm to create prediction model, find out if is it different model that can give better accuracy?

# Section 3 – Week 11

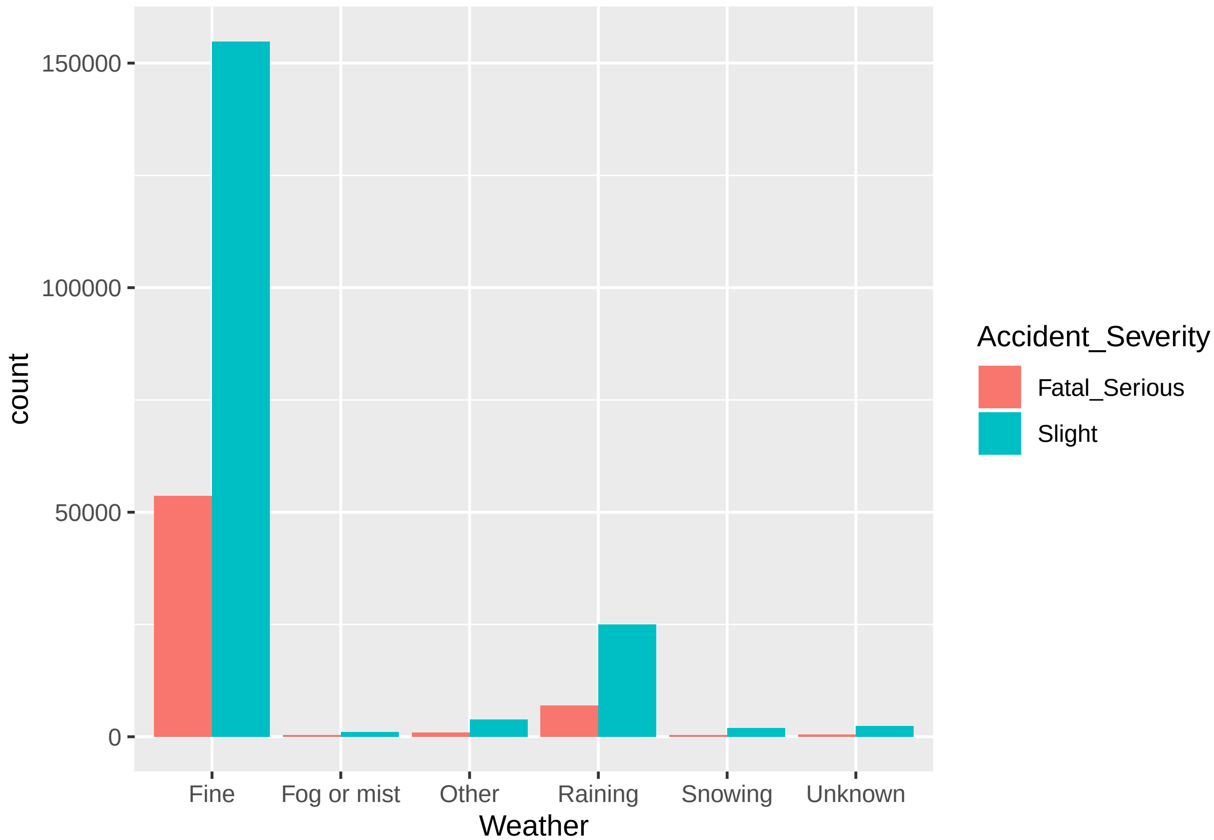
- **What information is not self-evident?**

1. The Season feature is numeric column however the mapping is not evident.

- **What are different ways you could look at this data?**

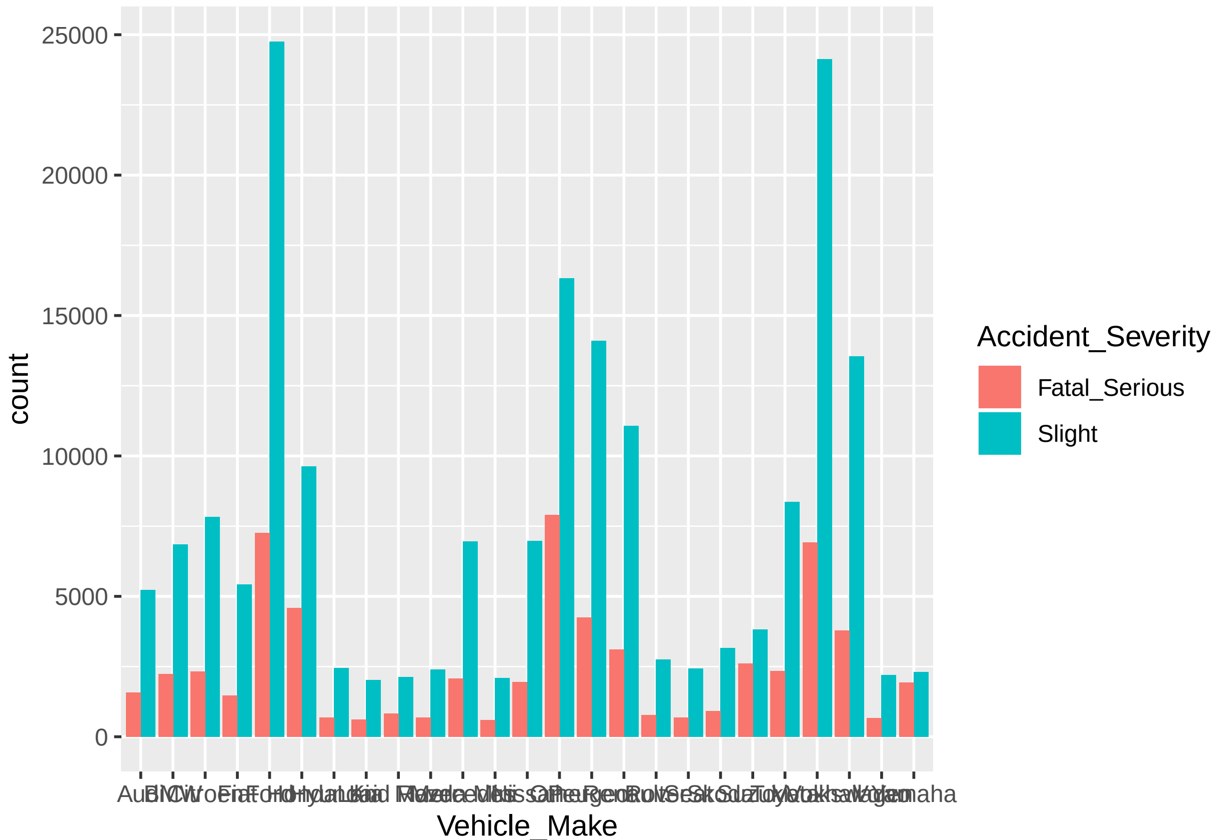
```r
ggplot(df_prediction, aes(Weather, fill = Accident_Severity)) + geom_histogram(stat = "count", position
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
ggplot(df_prediction, aes(Vehicle_Make, fill = Accident_Severity)) + geom_histogram(stat = "count", pos
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




* ### How do you plan to slice and dice the data? * ### How could you summarize your data to answer key questions? * ### What types of plots and tables will help you to illustrate the findings to your questions? * ### Do you plan on incorporating any machine learning techniques to answer your research questions? Explain. * ### Questions for future steps. # Section 4 – Week 12 * ### A story / narrative that emerged from your data. Follow this structure. * ### Introduction. * ### The problem statement you addressed. * ### How you addressed this problem statement * ### Analysis. * ### Implications. * ### Limitations. * ### Concluding Remarks

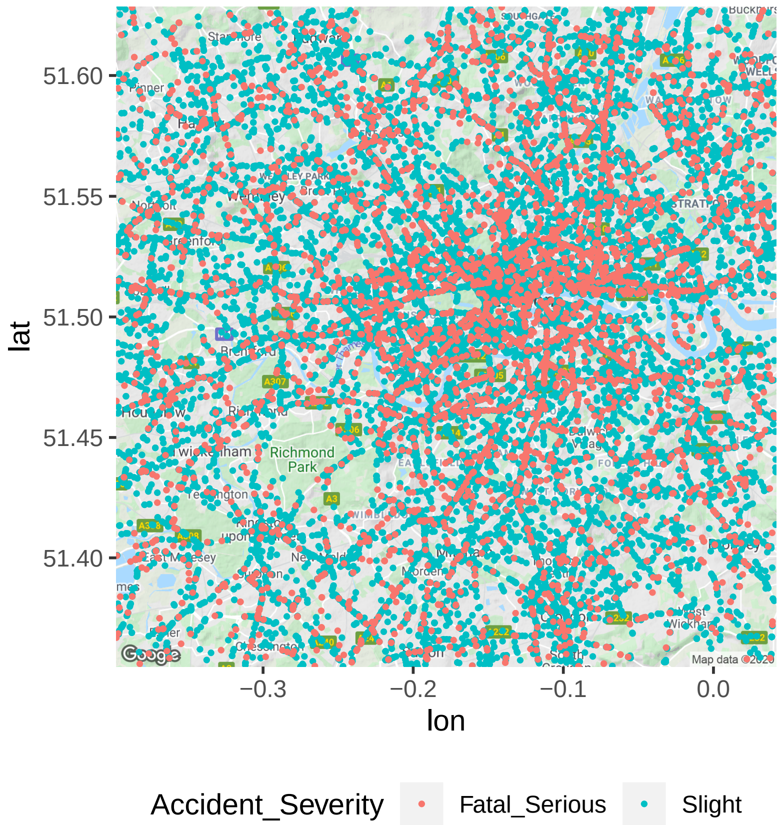
**Show Accidents by Location**

```
p <- ggmap(get_googlemap(center = c(lon = -0.178376, lat = 51.49204),
                         zoom = 11, scale = 2,
                         maptype = 'terrain',
                         color = 'color'))
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=51.49204,-0.178376&zoom=11&size=640x6
```

```
p +
  geom_point(aes(x = Longitude, y = Latitude, colour = Accident_Severity), data = df, size = 0.5) +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 225118 rows containing missing values (geom_point).
```


51.60
51.55
51.50
51.45
51.40
lat
−0.3
−0.2
−0.1
0.0
lon
Accident_Severity
Fatal_Serious
Slight