

# Project 1

## Exploring the dataset

We were given the dataset `2016collision.csv`. If we inspect the file, we can see that we have 14023 observations of 13 variables. Some of those variables are categorical, but others are not, so we need to be careful when we import the dataset so that R doesn't convert a numerical variable into a factor.

```
df <- read.csv('2016collisionsfinal.csv', header = T, stringsAsFactors = F)
```

And the name of the 13 variables are:

```
colnames(df)
```

```
## [1] "Record"                 "Location"  
## [3] "X"                     "Y"  
## [5] "Date"                  "Time"  
## [7] "Environment"           "Road_Surface"  
## [9] "Traffic_Control"       "Collision_Location"  
## [11] "Light"                 "Collision_Classification"  
## [13] "Impact_type"
```

## The variables

Here is a short interpretation of each variable:

- **Record**: a unique number, ranging from 1 to 14023, given to each observation.
- **Location**: the location, in the form of an address.
- **X,Y**: the longitude and latitude of the collision based on the Modified Transverse Mercator (MTM) coordinate system.
- **Date**: the date of the collision (in the format '%m/%d%y')
- **Time**: the time of the collision (in the format '%H:%M')
- **Environment**: a categorical variable that specifies the type of weather at the moment of the accident. It has levels:

```
df$Environment <- as.factor(df$Environment)  
levels(df$Environment)  
  
## [1] "00 - Unknown"          "01 - Clear"  
## [3] "02 - Rain"            "03 - Snow"  
## [5] "04 - Freezing Rain"   "05 - Drifting Snow"  
## [7] "06 - Strong wind"     "07 - Fog, mist, smoke, dust"  
## [9] "99 - Other"
```

Most of the levels are self-explanatory.

- **Road\_Surface**: a categorical variable that specifies the condition of the road at the moment of the collision. It has levels:

```
df$Road_Surface <- as.factor(df$Road_Surface)  
levels(df$Road_Surface)  
  
## [1] "00 - Unknown"          "01 - Dry"  
## [3] "02 - Wet"              "03 - Loose snow"  
## [5] "04 - Slush"            "05 - Packed snow"  
## [7] "06 - Ice"               "07 - Mud"  
## [9] "08 - Loose sand or gravel" "99 - Other"
```

- **Traffic\_Control**: a categorical variable that specifies the type of traffic control in place at the time of the accident. When we take a look at the levels of this variable, we can see that one is labeled "" (i.e. a null string). Further inspection maybe required to determine if this is due to a missing value or if this is an actual level.

```
df$Traffic_Control <- as.factor(df$Traffic_Control)
levels(df$Traffic_Control)

## [1] ""           "01 - Traffic signal" "02 - Stop sign"
## [4] "03 - Yield sign"   "04 - Ped. crossover" "10 - No control"
## [7] "11 - Roundabout"
```

- **Collision\_Location**: a categorical variable that specifies where the collision happened, but in terms of road types. We again got the label "", however we also have two labels for other road types: "98 - Other" and "99 - Other". This is probably a data entry error, because all the other variables use the label "99 - Other".

```
df$Collision_Location <- as.factor(df$Collision_Location)
levels(df$Collision_Location)

## [1] ""           "01 - Non intersection"
## [3] "02 - Intersection related" "03 - At intersection"
## [5] "04 - At/near private drive" "05 - At railway crossing"
## [7] "06 - Underpass or tunnel"   "07 - Overpass or bridge"
## [9] "98 - Other"               "99 - Other"
```

If we look at which observation are not label properly,

```
which(df$Collision_Location == "98 - Other")

## [1] 6193 6510

which(df$Collision_Location == "")
```

```
## [1] 7562 9781 11883 12679 12771 13261
```

we can see that only two observation are label "98 - Other". This is probably an error, we can remap them to "99 - Other".

```
df$Collision_Location[which(df$Collision_Location == "98 - Other")] <- "99 - Other"
```

However, for the value that have the label "", there's not much we can do other than exclude them from the dataset if they pose any problem.

- **Light**: a categorical variable that specifies the amount of light (natural or artificial) at the moment of the collision. We again got the "" label.

```
df$Light <- as.factor(df$Light)
levels(df$Light)

## [1] ""           "00 - Unknown"
## [3] "01 - Daylight"   "02 - Daylight, artificial"
## [5] "03 - Dawn"       "04 - Dawn, artificial"
## [7] "05 - Dusk"       "06 - Dusk, artificial"
## [9] "07 - Dark"        "08 - Dark, artificial"
## [11] "99 - Other"
```

Since there are only two observations in the "" label, we may assume that these are missing values or data entry errors.

```
df$Collision_Classification <- as.factor(df$Collision_Classification)
```

- **Collision\_Classification**: a categorical variable with 3 levels:
  - 01 - *Fatal injury*: Implies that one or more person has deceased due to the collision.
  - 02 - *Non-fatal injury*: Implies that one or more person got injured as a result of the collision, but no one died.
  - 03 - *P.D. only*: Implies that only some property damage was caused by the collision.
- **Impact\_type**: a categorical variable that specifies the type of impact for the collision. It has levels:

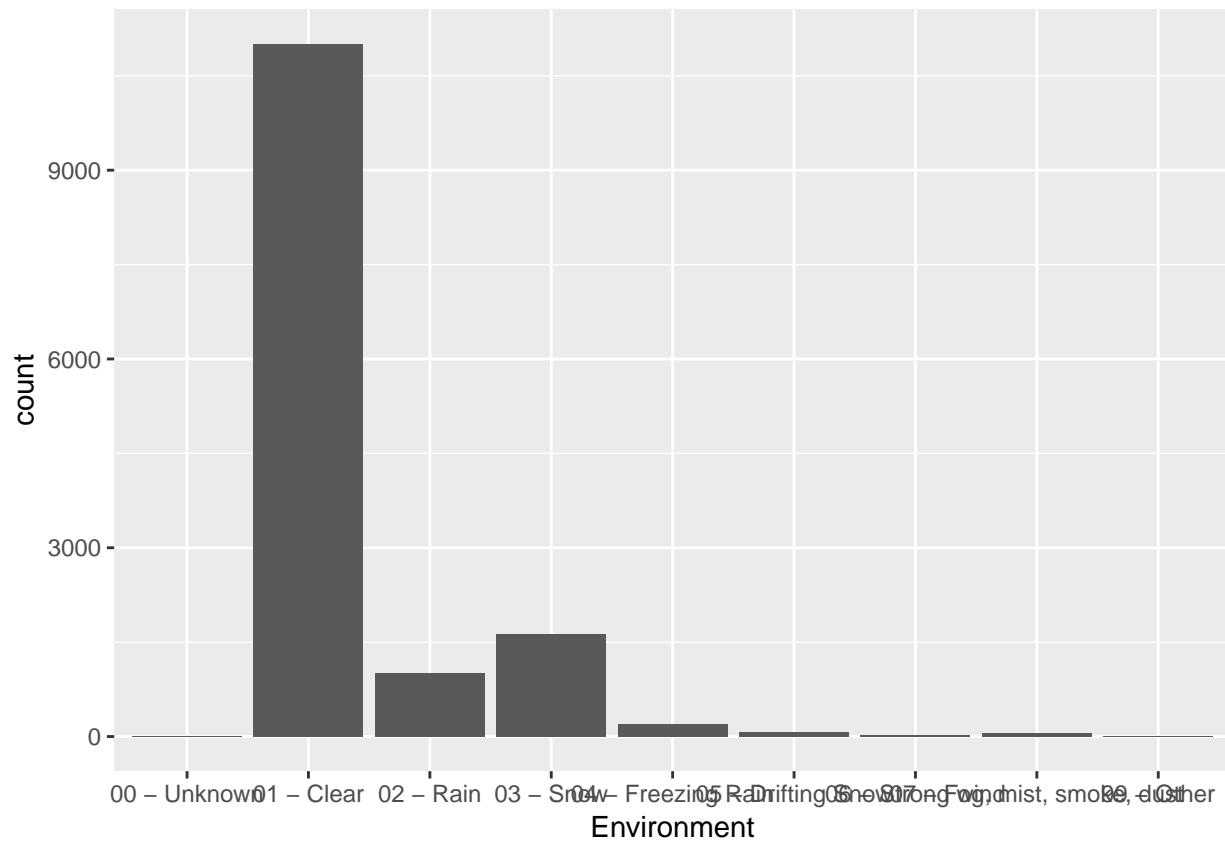
```
df$Impact_type <- as.factor(df$Impact_type)
levels(df$Impact_type)
```

```
## [1] "01 - Approaching"          "02 - Angle"
## [3] "03 - Rear end"            "04 - Sideswipe"
## [5] "05 - Turning movement"    "06 - SMV unattended vehicle"
## [7] "07 - SMV other"           "99 - Other"
```

### Simple plot of the categorical variables

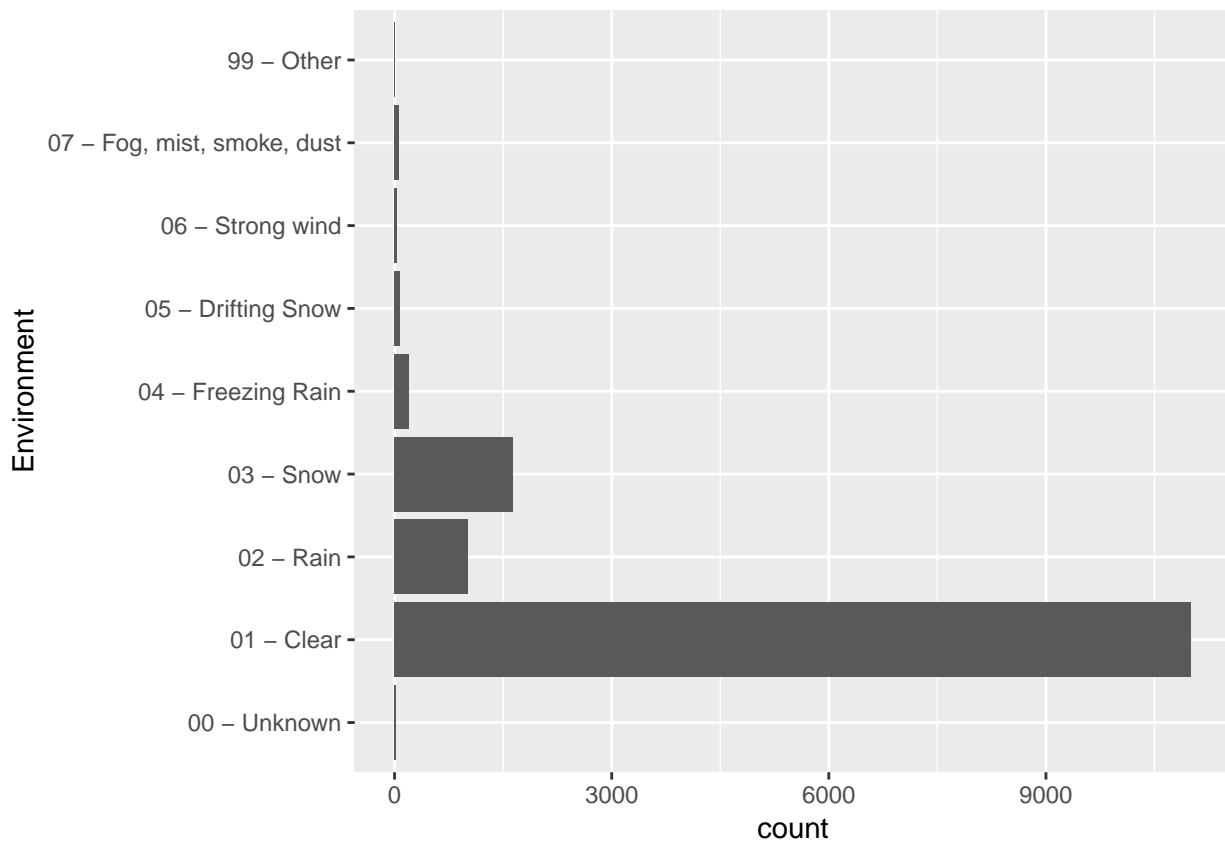
#### Environment

```
p <- ggplot(data = df)
p + geom_bar() + aes(Environment)
```



The labels are hard to read, rotating the plot might be a good idea.

```
p + geom_bar() + aes(Environment) + coord_flip() + xlab("Environment")
```

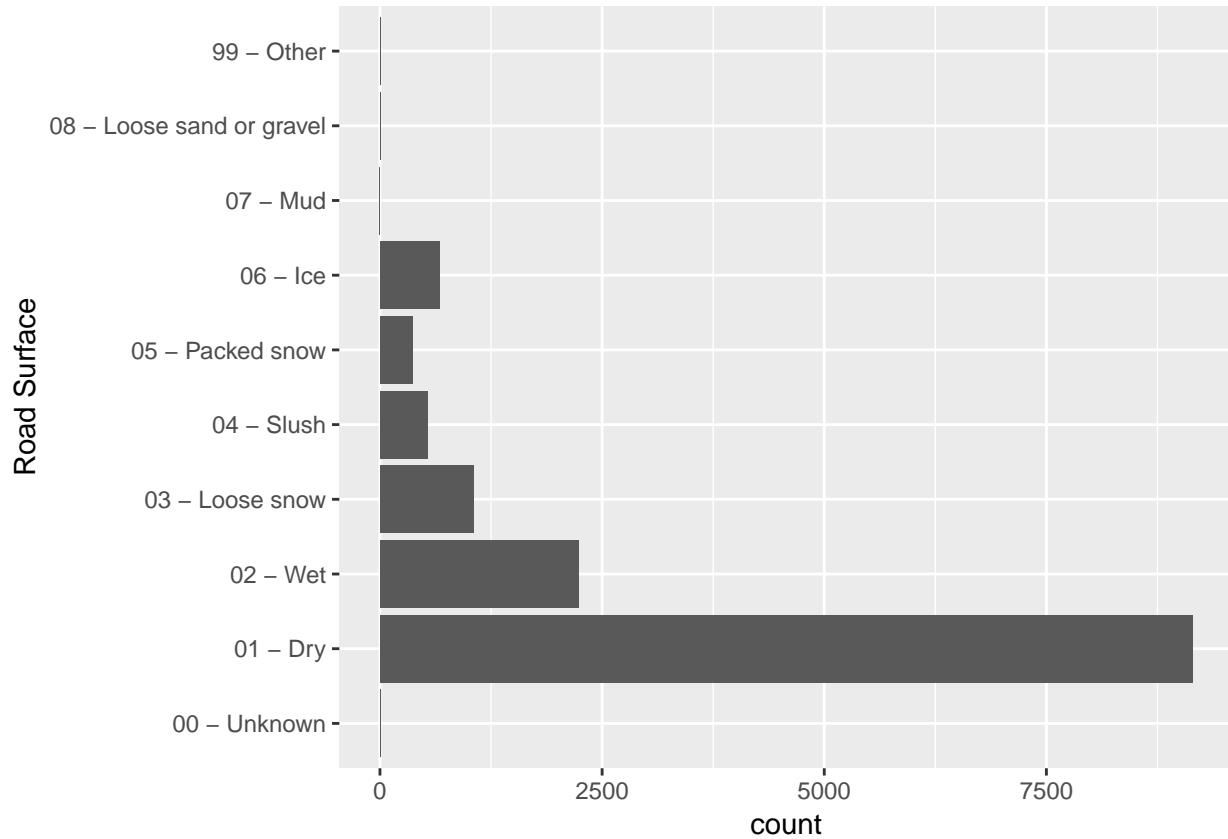


We can see that the most collisions occur in a clear environment.

### Road Surface

We can do the same with the road surface.

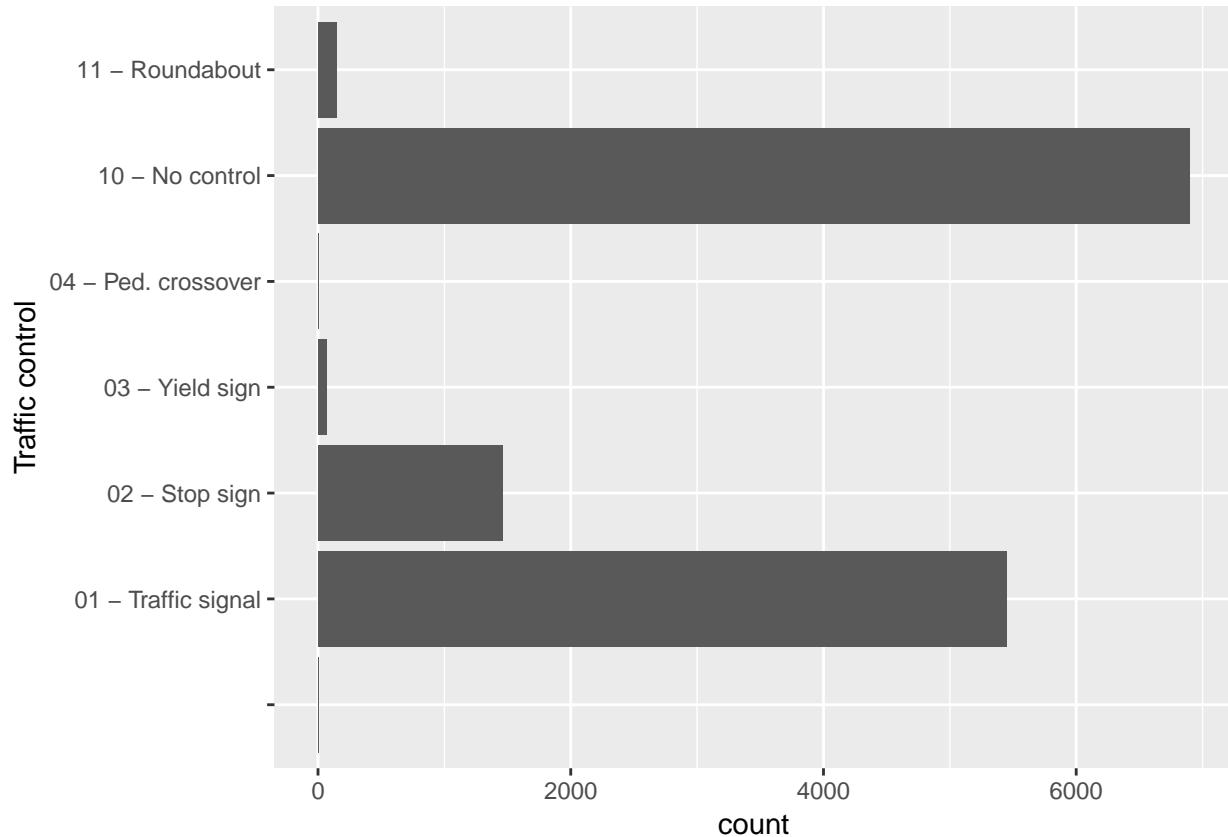
```
p + aes(x=df$Road_Surface) + geom_bar() + coord_flip() + xlab("Road Surface")
```



Accordingly to the result we got with the Environment variable, most collisions happen when the road surface is dry.

### Trafic Control

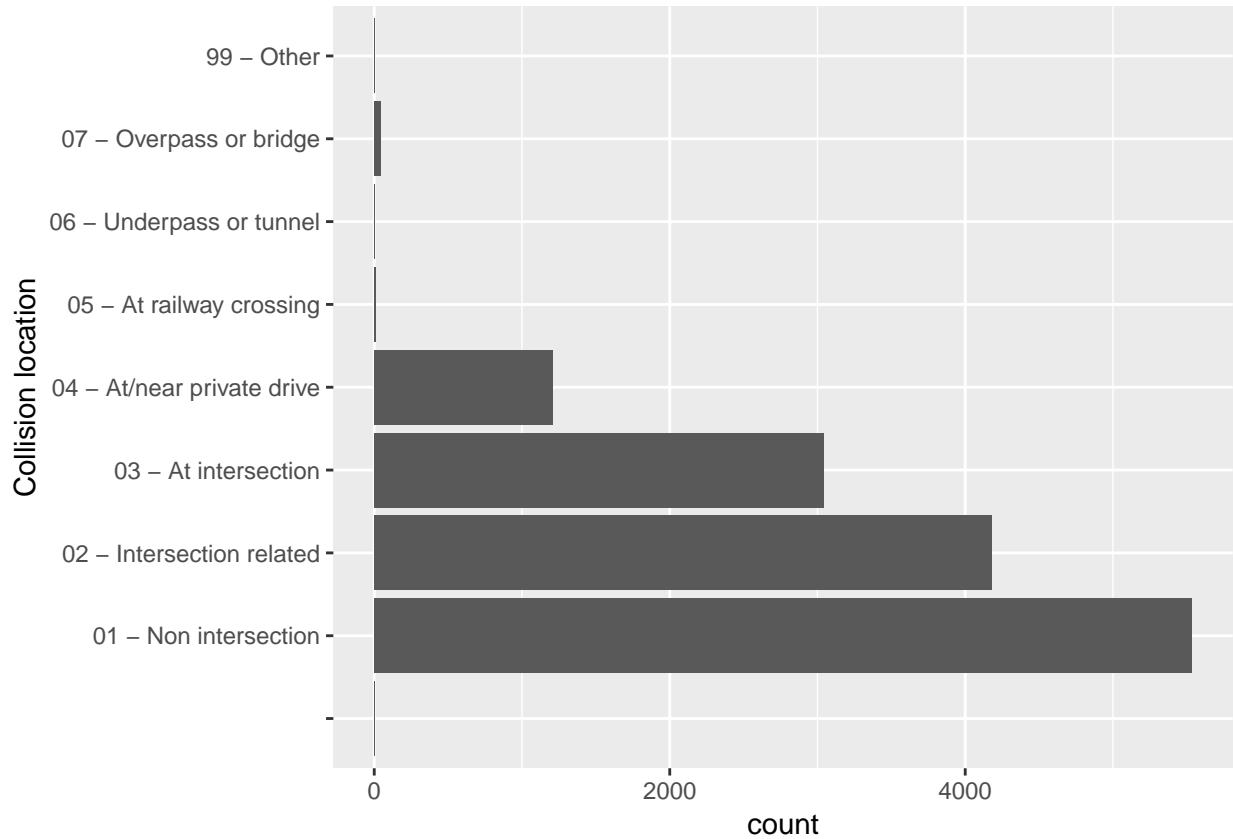
```
p + aes(x = df$Traffic_Control) + geom_bar() + coord_flip() + xlab("Traffic control")
```



In this plot, we can see that most collisions happen when there was no traffic control, but a lot of accidents also happen at a traffic light or at a stop sign.

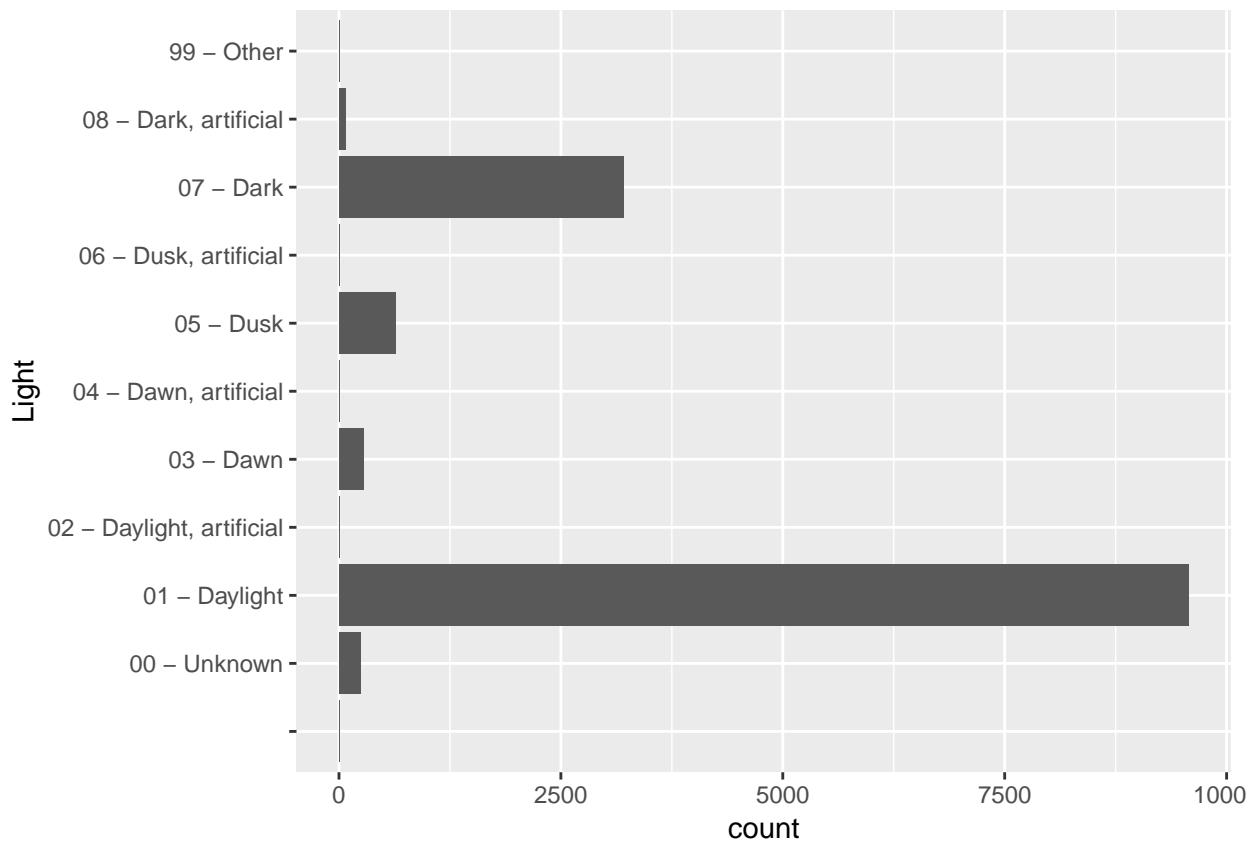
### Collision Location

```
p + aes(x = df$Collision_Location) + geom_bar() + coord_flip() + xlab('Collision location')
```



## Light

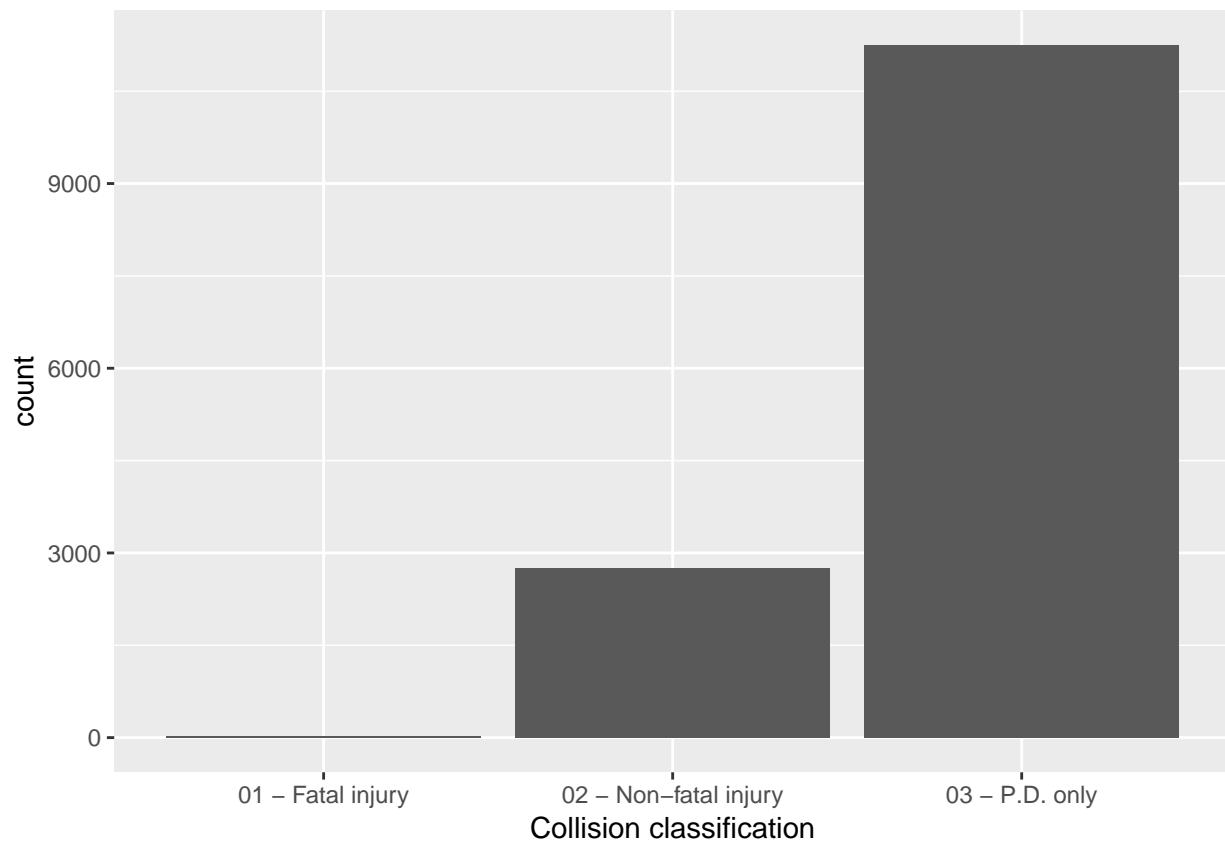
```
p <- ggplot(df, aes(x = df$Light)) + geom_bar() + coord_flip() + xlab('Light')
show(p)
```



Here, we can clearly see that most collision happen in daylight.

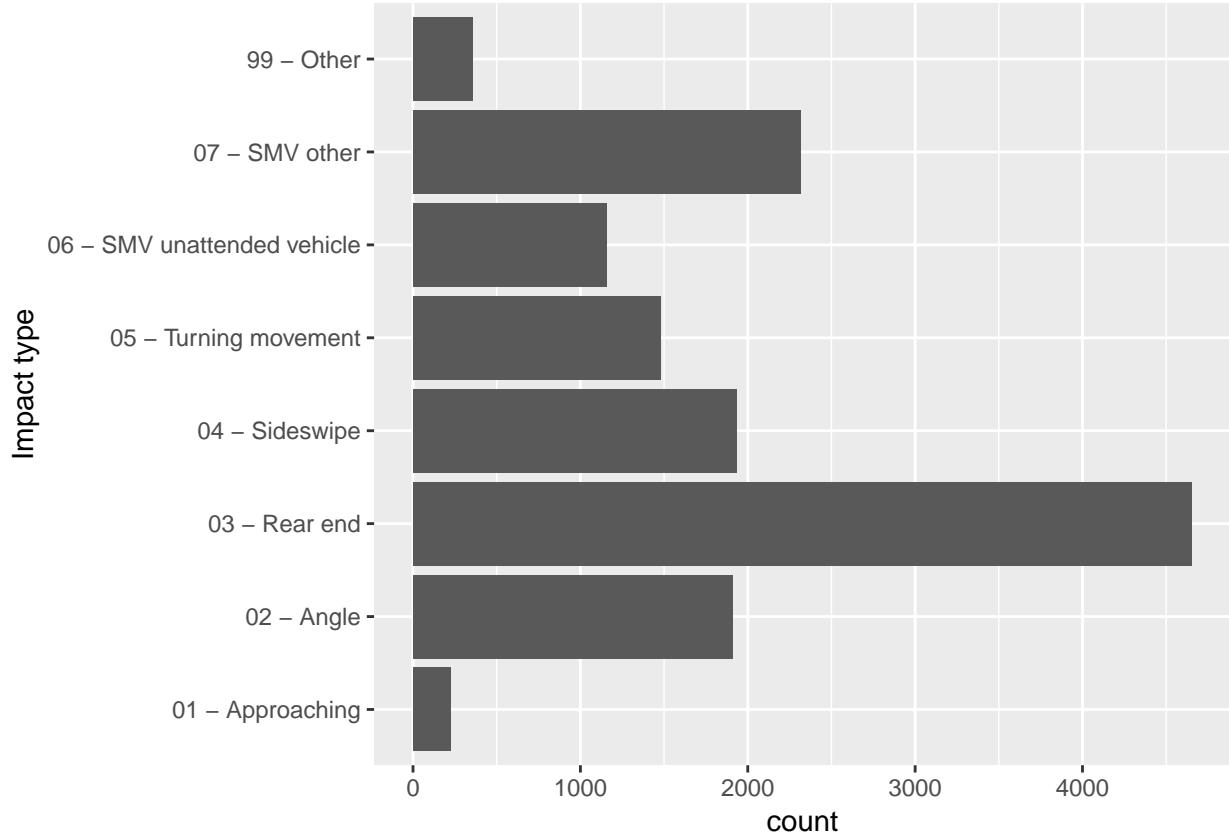
### Collision classification

```
p <- ggplot(df, aes(x = df$Collision_Classification)) + geom_bar() + xlab("Collision classification")  
show(p)
```



### Impact Type

```
p <- ggplot(df, aes(df$Impact_type)) + geom_bar() + coord_flip() + xlab("Impact type")
show(p)
```



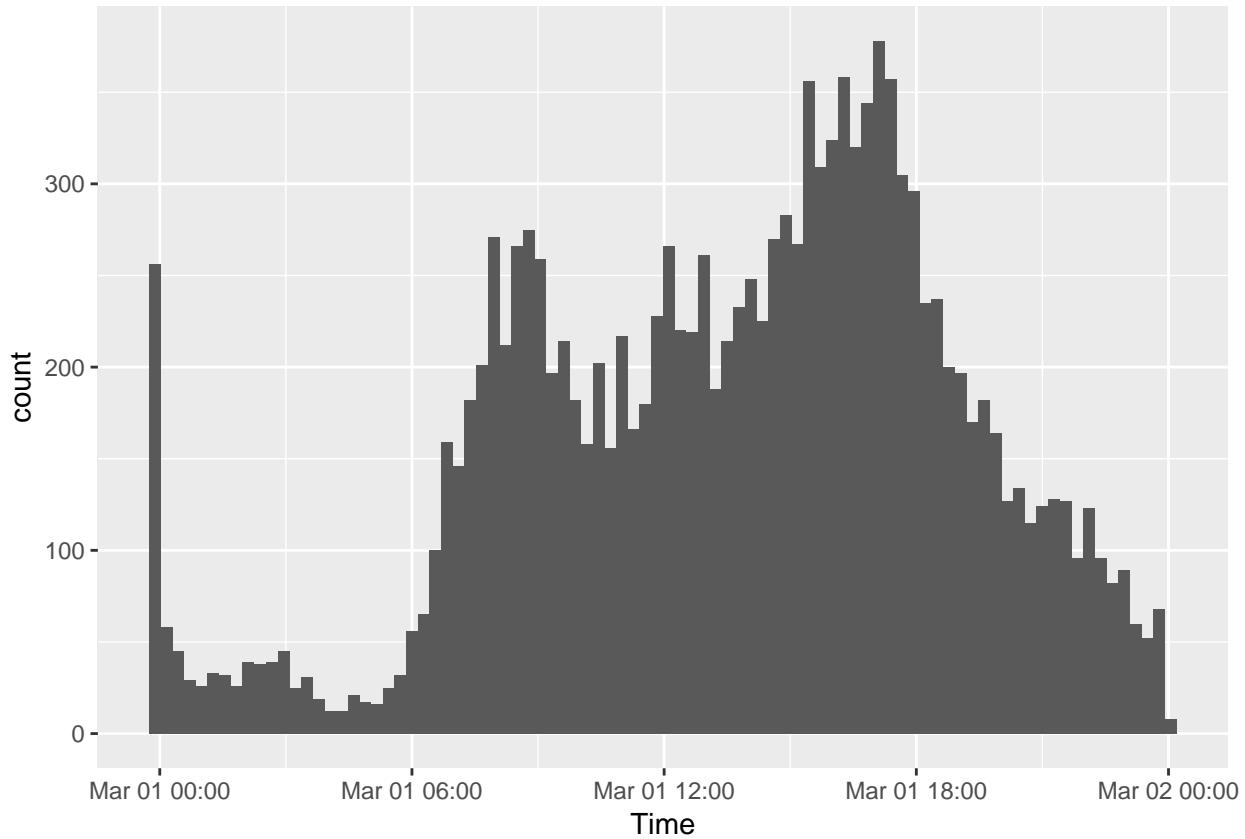
### Collision w.r.t the time of the day

Since we have access to the time of the collision, one could ask at what time do most collisions happen?

To do this, we need to convert the Time vector of string into something more useful, such as a vector of datetime object.

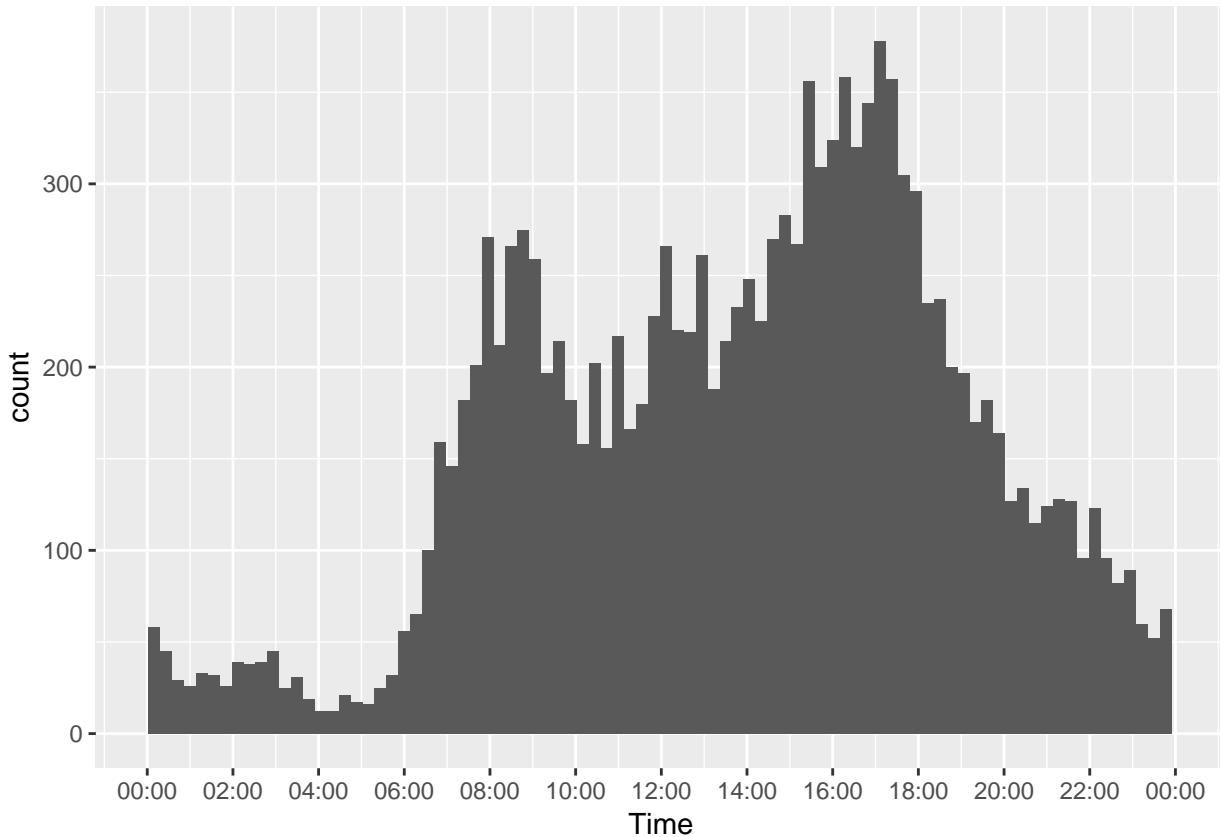
```
df$Time <- as.POSIXct(df$Time, format = '%H:%M')
plot_lim <- as.POSIXct(c('00:00', '23:59'), format = '%H:%M')

p <- ggplot(data = df, aes(x = Time)) + geom_histogram(binwidth = 1000)
show(p)
```



The first thing to say about this plot is that it is very unlikely that a number this large of collisions happened exactly at 00:00. This could indicate many things, but most likely that some entries are missing or ill formatted.

```
p + scale_x_datetime(limits =plot_lim, breaks=date_breaks("2 hours"), labels=date_format("%H:%M", tz="Asia/Tokyo"))
## Warning: Removed 2 rows containing missing values (geom_bar).
```



While trying to fix the range and format of the x axis, for some reason, `geom_bar` automatically removes the missing value. From this visualization, it seems that there is more collisions around 8am and around 4pm. This is probably due to rush hour.

### Colision along the year

\*\* In not sure why, but I can't run this code. I have made pretty much the same graph, but without doing database manipulations\*\*

First, the hour and the month of each collision could be of interest, therefore we extract this information from the `Time` and the `Date` variables :

```
library(ggpubr)

## Loading required package: magrittr
##
## Attaching package: 'magrittr'
## The following object is masked from 'package:ggmap':
##     inset
## The following object is masked from 'package:purrr':
##     set_names
## The following object is masked from 'package:tidyverse':
##     extract
```

```

library(lubridate)
library(hexbin)
library(plyr)

## -----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## -----
## 
## Attaching package: 'plyr'
## The following object is masked from 'package:ggpubr':
## 
##     mutate
## The following object is masked from 'package:lubridate':
## 
##     here
## The following objects are masked from 'package:dplyr':
## 
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarise
## The following object is masked from 'package:purrr':
## 
##     compact
library(reshape2)

## 
## Attaching package: 'reshape2'
## The following object is masked from 'package:tidyverse':
## 
##     smiths
#extracting the month out of the date variable and making a new variable for month

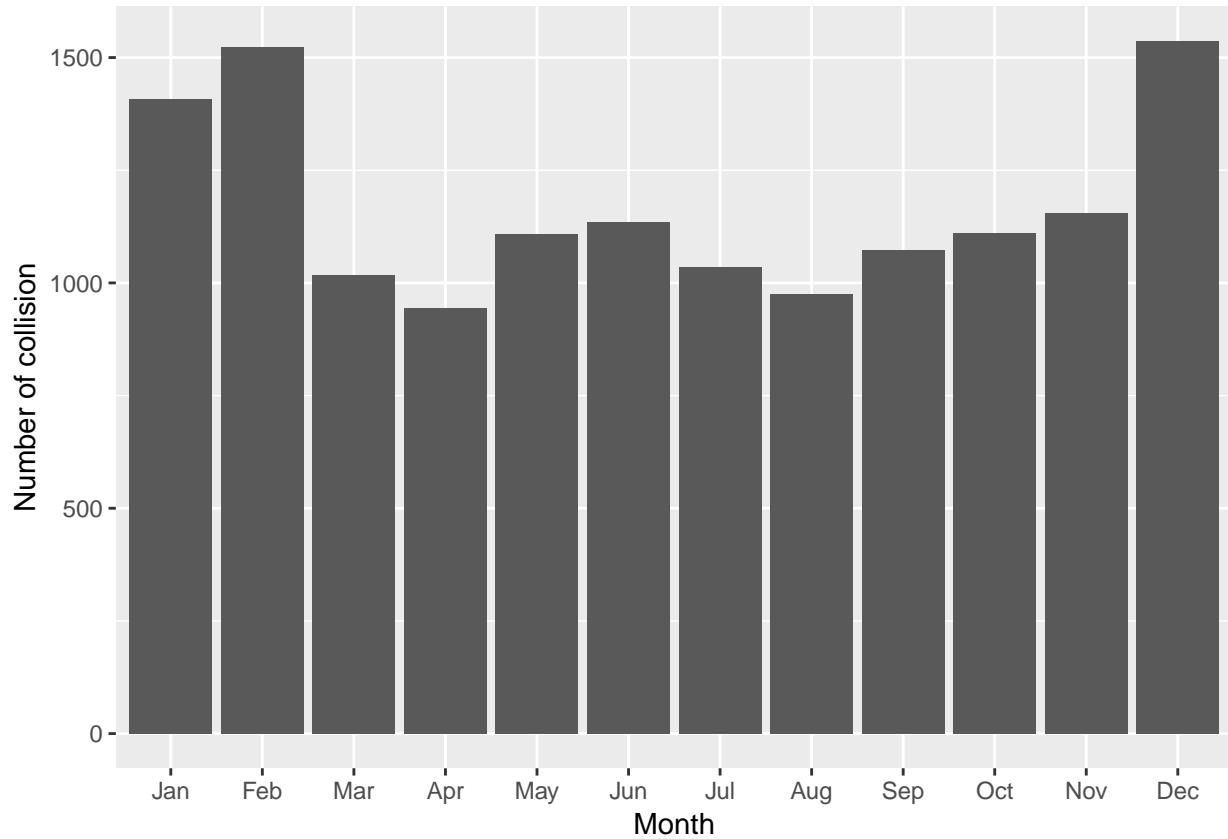
df$Month <- as.factor(month(as.POSIXlt(df$date, format="%m/%d/%Y")))
levels(df$Month) <- month.abb[1:12]

#extracting the hour out of the time variable and making a new variable for hour

df$Hour <- as.factor(format(as.POSIXct(df$time, format="%H:%M"), "%H"))

p <- ggplot(df, aes(x = df$Month)) + geom_bar() + xlab("Month") + ylab("Number of collision")
show(p)

```



As we can see, it seems that there is a higher total number of accident in December, January and February. This might be due to winter and the snow. More on this will be explored in the bivariate part of this document.

## Map Visualizations

Since this dataset represents collisions in Ottawa that happened in 2016, we can plot the collisions on a map, using the coordinates (variables X and Y) from the dataset; this way, we can observe spatial details of the collisions data.

Since the coordinates in the dataset were recorded according to the Modified Transverse Mercator coordinate system, we need to convert the coordinates to the latitude-longitude system since the Ottawa map is scaled according to the lat-long system.

First, we have to determine if there are any missing coordinates, because the converting function doesn't work if there are missing values :

```
df$X <- as.numeric(df$X)

## Warning: NAs introduced by coercion
df$Y <- as.numeric(df$Y)

## Warning: NAs introduced by coercion
#finding records with missing values (when in numeric form)

for (i in 1:14023) {
  if (is.na(df$X[i])) {
    print(i)
  }
}
```

```

}

## [1] 5150
## [1] 5666
## [1] 5942
## [1] 5943
## [1] 9238
## [1] 12111
## [1] 12290
## [1] 13828

```

With a little bit more investigating, we see that the records with missing coordinates are actually entries in a different format.(ex: 356,762.29 instead of 356762.29) Therefore, we will manually input the correctly formatted values for these entries:

```
#manually change the format of entries with different formats
```

```

df[5150,3] <- "356762.29"
df[5150,4] <- "5015593.96"

df[5666,3] <- "365191.84"
df[5666,4] <- "5011253.07"

df[5942,3] <- "384542.73"
df[5942,4] <- "5034620.66"

df[5943,3] <- "384542.73"
df[5943,4] <- "5034620.66"

df[9238,3] <- "384661.88"
df[9238,4] <- "5034144.30"

df[12111,3] <- "356864.55"
df[12111,4] <- "5015648.07"

df[12290,3] <- "382197.79"
df[12290,4] <- "5033332.03"

df[13828,3] <- "356535.81"
df[13828,4] <- "5015509.05"

```

```
#change X and Y character to num again
```

```

df$X <- as.numeric(df$X)
df$Y <- as.numeric(df$Y)

```

Now, since we fixed the missing values, we can convert the coordinates to lat-long :

```
#make variable of coords
coords <- cbind(Easting = df[,3], Northing = df[,4])

#make dataset without coords
data_less_points <- df[,-(3:4)]

#make data frame of spatial coords
utms <- SpatialPointsDataFrame(coords, data = data_less_points, proj4string = CRS("+init=epsg:2018 +datat"))

```

```

#for conversion
latlong = "+init=epsg:4269"

#transforming mtms to lat-long
gps_coords <- spTransform(utm, CRS(latlong))

#making new data.frame with lat-long coords and all other variables
data_with_lat.long <- data.frame(gps_coords)
data_with_lat.long <- data_with_lat.long[,1:15]
names(data_with_lat.long) <-c("Record", "Location", "Date", "Time", "Environment", "Road_Surface", "Traffic_Control", "Collision_Location", "Light", "Collision_Classification", "Impact_Type", "Month", "Hour", "Longitude", "Latitude")

```

Finally, the coordinates are converted to the lat-long system :

```
head(data_with_lat.long)
```

	Record	Location	Date		
## 1	1	105 S OF COMMISSIONER ST @ ALBERT ST	8/4/16		
## 2	2	210 W OF MERIVALE RD @ WEST HUNT CLUB RD	3/30/16		
## 3	3	210 W OF MERIVALE RD @ WEST HUNT CLUB RD	9/2/16		
## 4	4	210 W OF MERIVALE RD @ WEST HUNT CLUB RD	3/12/16		
## 5	5	210 W OF MERIVALE RD @ WEST HUNT CLUB RD	8/23/16		
## 6	6	210 W OF MERIVALE RD @ WEST HUNT CLUB RD	7/6/16		
	Time	Environment	Road_Surface	Traffic_Control	
## 1	2019-03-01 17:03:00	01 - Clear	01 - Dry	01 - Traffic signal	
## 2	2019-03-01 15:44:00	01 - Clear	01 - Dry	01 - Traffic signal	
## 3	2019-03-01 16:02:00	01 - Clear	01 - Dry	01 - Traffic signal	
## 4	2019-03-01 17:30:00	01 - Clear	01 - Dry	01 - Traffic signal	
## 5	2019-03-01 15:52:00	01 - Clear	01 - Dry	01 - Traffic signal	
## 6	2019-03-01 16:19:00	01 - Clear	01 - Dry	01 - Traffic signal	
	Collision_Location	Light	Collision_Classification		
## 1	02 - Intersection related	01 - Daylight	03 - P.D. only		
## 2	03 - At intersection	01 - Daylight	02 - Non-fatal injury		
## 3	03 - At intersection	01 - Daylight	03 - P.D. only		
## 4	03 - At intersection	01 - Daylight	03 - P.D. only		
## 5	02 - Intersection related	01 - Daylight	03 - P.D. only		
## 6	02 - Intersection related	01 - Daylight	03 - P.D. only		
	Impact_Type	Month	Hour	Longitude	Latitude
## 1	04 - Sideswipe	Aug	17	-75.70955	45.41493
## 2	05 - Turning movement	Mar	15	-75.72644	45.33362
## 3	05 - Turning movement	Sep	16	-75.72643	45.33361
## 4	05 - Turning movement	Mar	17	-75.72645	45.33362
## 5	03 - Rear end	Aug	15	-75.72645	45.33362
## 6	03 - Rear end	Jul	16	-75.72643	45.33361

Next, we can get map tiles of Ottawa from Stamen Maps. We can change the size of the maps to focus on different areas of the city:

(Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.)

```
#getting tiles from Stamen and creating different maps
```

```
ottawa.map <- get_stamenmap(bbox = c(left = -76.5, bottom = 45.0, right = -75.0, top = 45.8), zoom = 11)

ottawa.map2 <- get_stamenmap(bbox = c(left = -77.1384, bottom = 45.14344, right = -76.6394, top = 45.55))
```

```

ottawa.map3 <- get_stamenmap(bbox = c(left = -76.2384, bottom = 45.14344, right = -75.2, top = 45.5506))

ottawa.map4 <- get_stamenmap(bbox = c(left = -76.3384, bottom = 45.0, right = -75.0, top = 45.5506), zoom = 10)

ottawa.map.zoom <- get_stamenmap(bbox = c(left = -75.8, bottom = 45.3, right = -75.6, top = 45.45), zoom = 11)

ottawa.map.zoomer <- get_stamenmap(bbox = c(left = -75.75, bottom = 45.38, right = -75.65, top = 45.44), zoom = 12)

```

## Maps

Finally, we are ready to plot the collisions on the map.

This is a broad view of the city, including the suburbs :

```

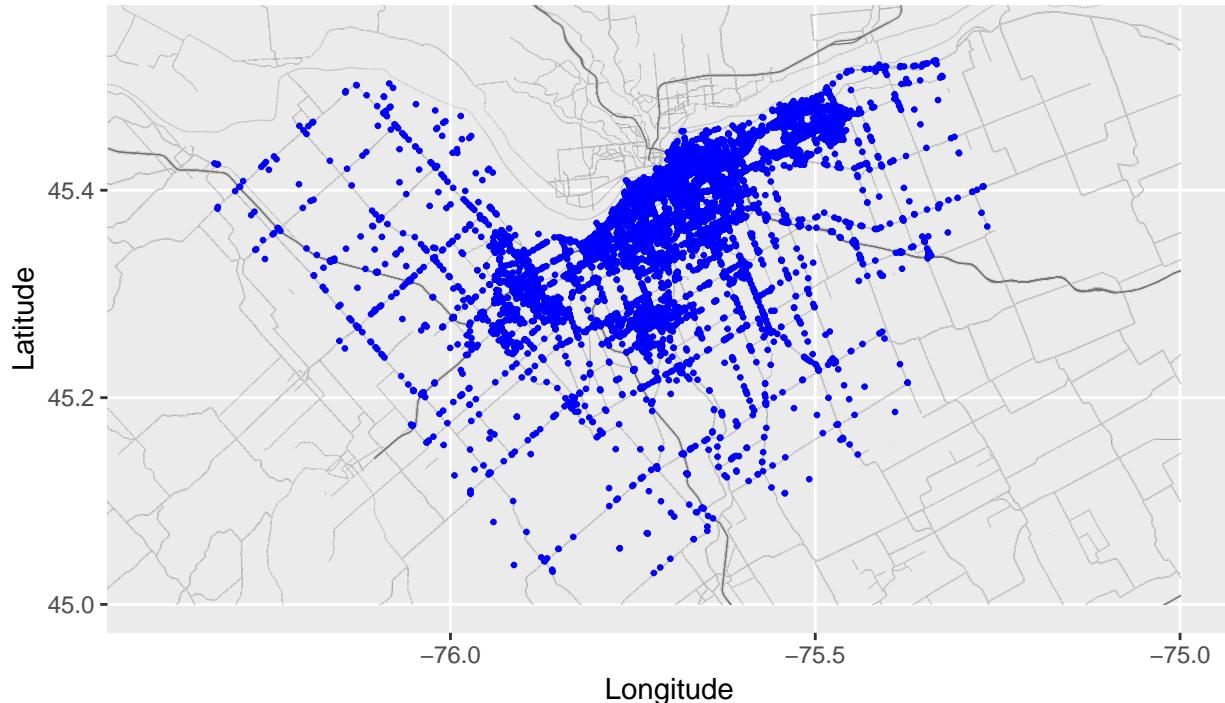
#broad view of Ottawa
ggmap(ottawa.map)+ geom_point(data = data_with_lat.long, aes(x=Longitude, y=Latitude), size=0.5, col='blue')
  scale_x_continuous(limits=c(-76.4,-75.0)) + labs(x="Longitude", y="Latitude") + ggtitle("Vehicle Collisions in Ottawa, in 2016")

## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.

## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.

```

**Vehicle Collisions in Ottawa, in 2016**



We can see from the above plot that there are more collisions in the more populated areas of the city (downtown, Orléans, Barrhaven, Kanata...) which is not really surprising. We may assume that, since there is a higher volume of vehicles on the road in more urban areas, there will be more collisions compared to more rural areas. It is also interesting to see the shape of the city, since the city limits are clearly distinguishable by the distribution of the collisions.

We can change the map by using a different scale, and get a closer look at the city:

```

# getting a better zoom

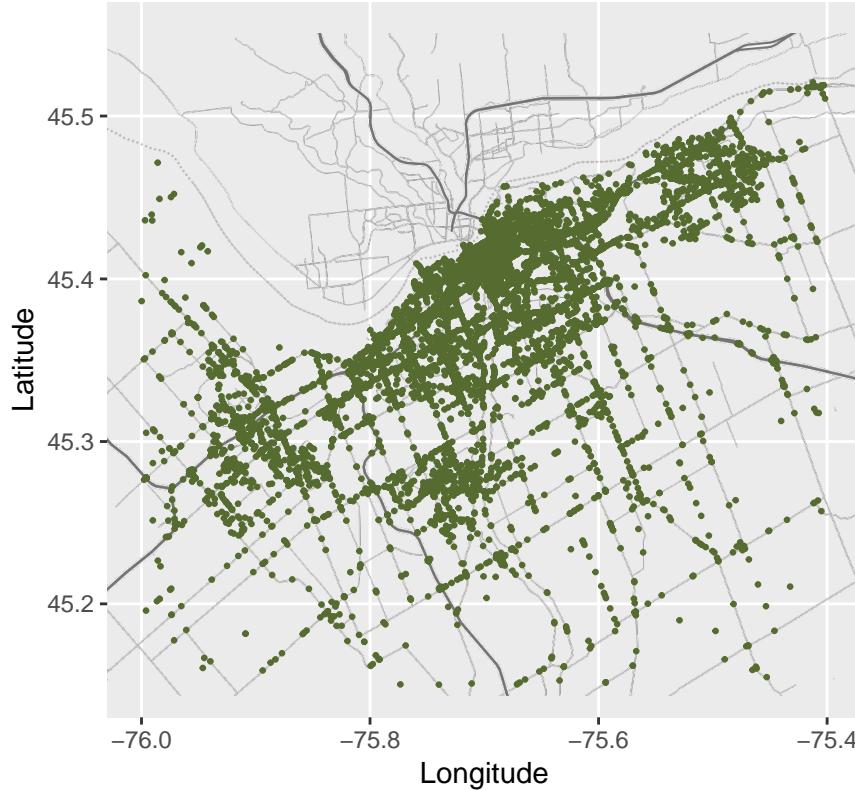
ggmap(ottawa.map3)+ geom_point(data = data_with_lat.long, aes(x=Longitude, y=Latitude), size=0.5, col="darkgreen")

## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.

## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.

```

## Vehicle Collisions in Ottawa, in 2016



Again, we can clearly see that there are more collisions on the main roads, with the concentration of collisions increasing as the roads get closer to downtown Ottawa or to a suburb.

We can also observe the distribution of the collisions in Downtown Ottawa by using a map with a different scale :

```

# view of Downtown Ottawa

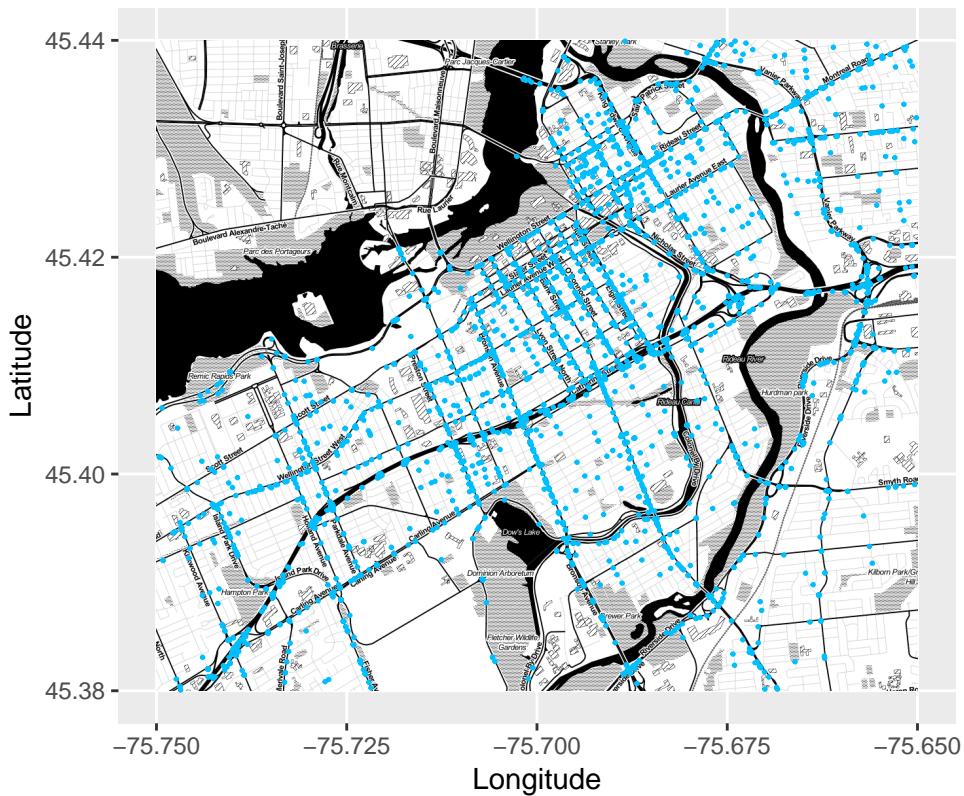
ggmap(ottawa.map.zoomer)+ geom_point(data = data_with_lat.long, aes(x=Longitude, y=Latitude), size=0.3, col="darkgreen")

## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.

## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.

```

## 2016 Collisions in Downtown Ottawa



In this case, we can see that the roads leading into the downtown core, like Bank Street or Bronson Avenue, have a high frequency of collisions, presumably because these are typical roads for commuters and workers coming into town. Also, with this map, it is interesting to see that some intersections, like Elgin Street and Laurier Avenue West, or Slater Street and Lyon Street North, have a higher density of collisions points than other intersections, than say Laurier Avenue West and Bank Street. Possibly, some intersections may be dealing with a higher volume of vehicles than others, or maybe the configuration of the intersections may lead to it being more collision prone.

Thus, these maps can help us identify possibly problematic areas with a many collisions points that would require further investigation.

### ###Bivariate Visualizations

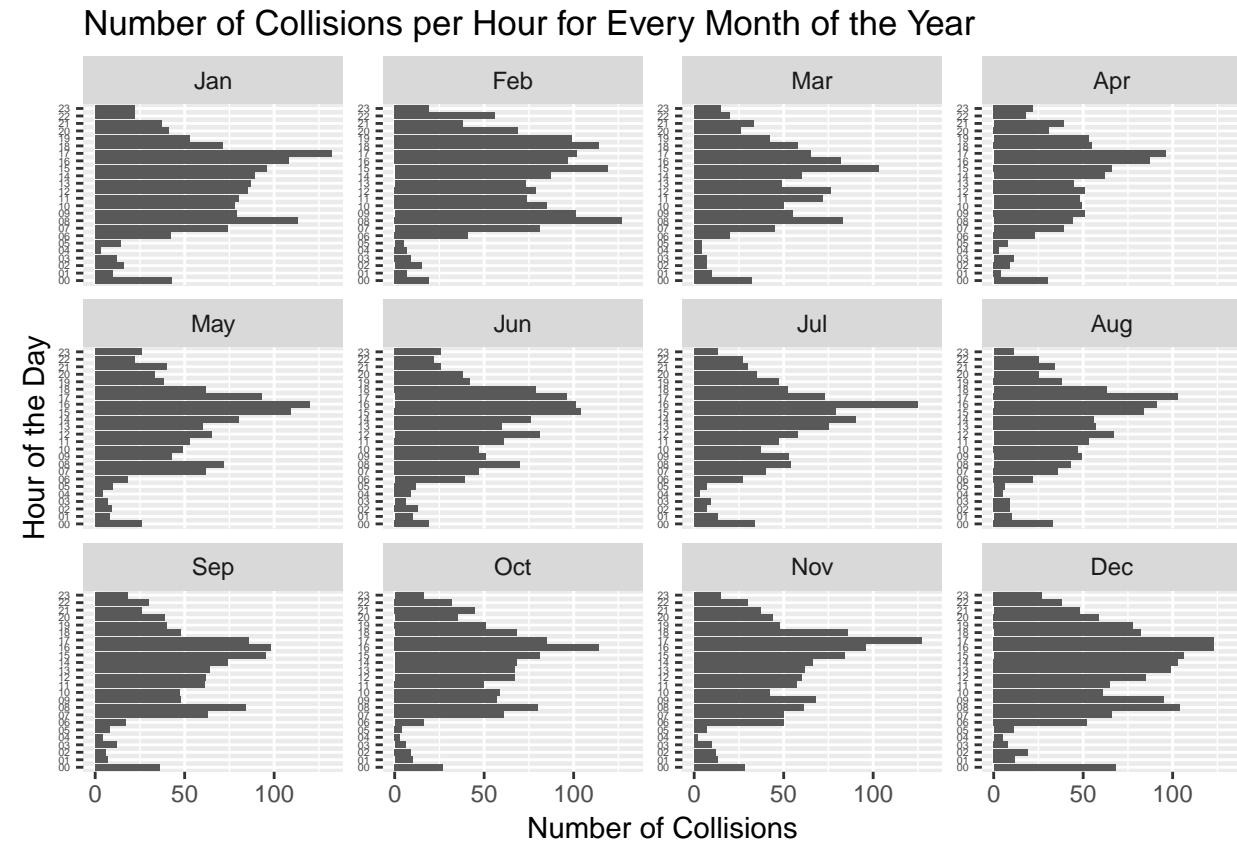
Furthermore, other variables from the dataset plotted pairwise can give us a high-level understanding and overview of possible trends or patterns in the data. The next step in our data exploration will be to look at some bivariate plots.

### ###Time Variables

In the collisions dataset, we got variables for the time of the collision and the date of the collision. It may be interesting to see how these time variables interact with each other, and with other variables.

Now, since we have the necessary information, we can see if there is any trends. The graphs below let us compare the total number of collisions by hour for every month of the year.

```
ggplot(df) + geom_bar(aes(Hour)) + facet_wrap(~Month, scales="free_y") + coord_flip() + theme(axis.text
```



When we look at the above plots, there is a similar trend throughout the first months of the year (January, February, and March) and the last months of the year (September, October, November and December) : the distribution of the collisions is bimodal, meaning that there seems to be a higher number of collisions at two times in the day. This is expected and in line with the common knowledge of rush-hour traffic in the morning, and in the afternoon in a city like Ottawa.

On the other hand, for the typical “summer” months (April, May, June, July and August), the distribution of the collisions by hour seems to be more unimodal, with a peak during afternoon rush hour. There is still a peak of collisions during the morning rush-hours, but they are not as prominent in the summer months. This may be due to the fact that Ottawa residents typically take vacations during the summer months and maybe are on the roads later in the morning, or work hours may be less rigid during the summer, and workers go into work at later hours, leading to a more uniform number of vehicles on the road in the morning hours, leading to a more uniform number of collisions.

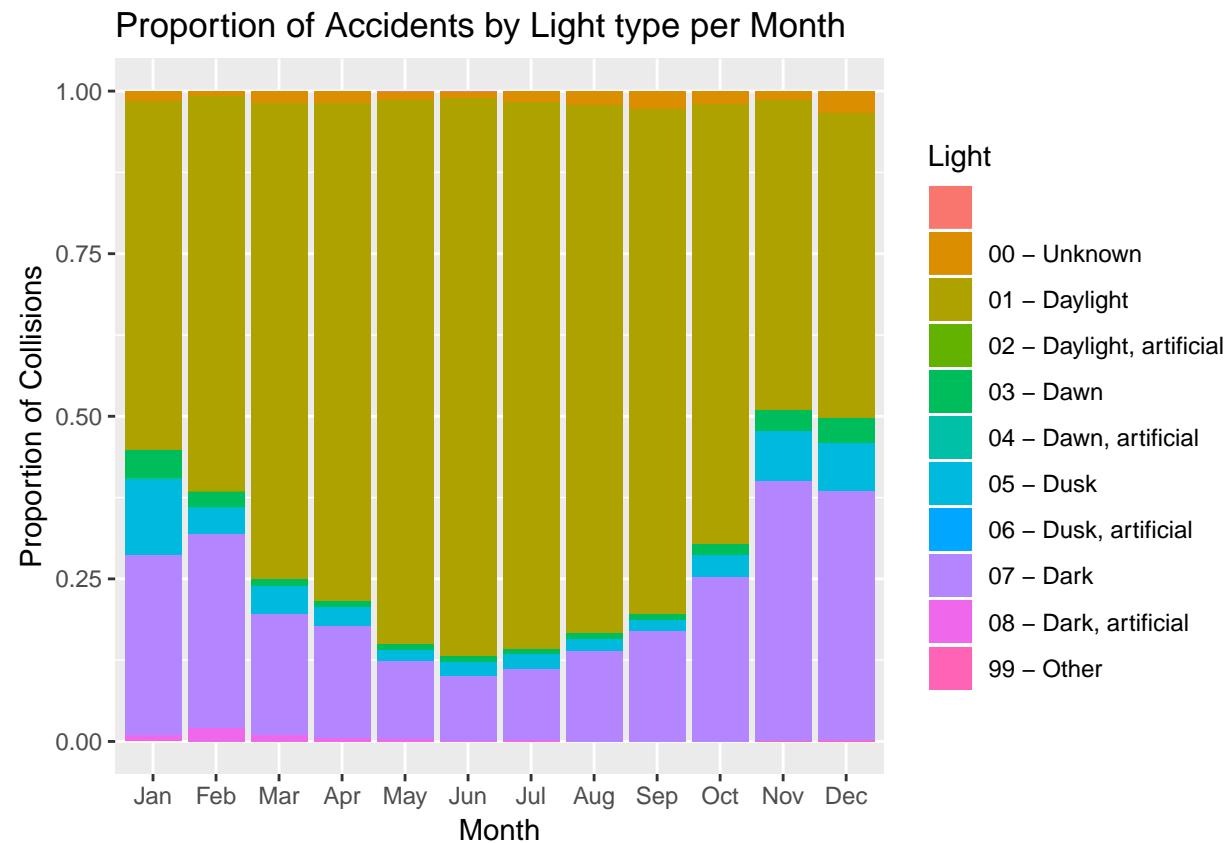
Also, if we specifically look at the peak collision hours in the afternoon/evening throughout the year, we see that generally, the peak collision hour for Jan-Apr is 16:00-17:00, while it is 15:00-16:00 for May-Sept, and again 16:17 for Oct-Dec. Similarly, if we look at the peak collision hours in the morning throughout the year, we generally see that during the winter months, the peak morning hour collision time is earlier in the morning (7:00-8:00), while in the summer months, the morning peak hour collision time is later in the morning (8:00-9:00). Again, these observations are not really surprising, if we assume that people may leave later for work and leave earlier from work during the summer months, leading to more vehicles on the road during these times.

#### #####2) Month and Light Type

In the following plot, we will see the proportion of collisions that occurred during different lighting types, by month.

```
#Light and Month
```

```
ggplot(df) + geom_bar(aes(Month, fill=Light), position="fill") + theme(axis.text.x=element_text(angle=0))
```



Not surprisingly, we see what we would expect to see, knowing how sunlight behaves throughout the year in Canada: a higher proportion of accidents happen during the daylight during the summer months, and in darkness during the winter months, since there are more daylight hours during the summer and darkness hours during the winter months.

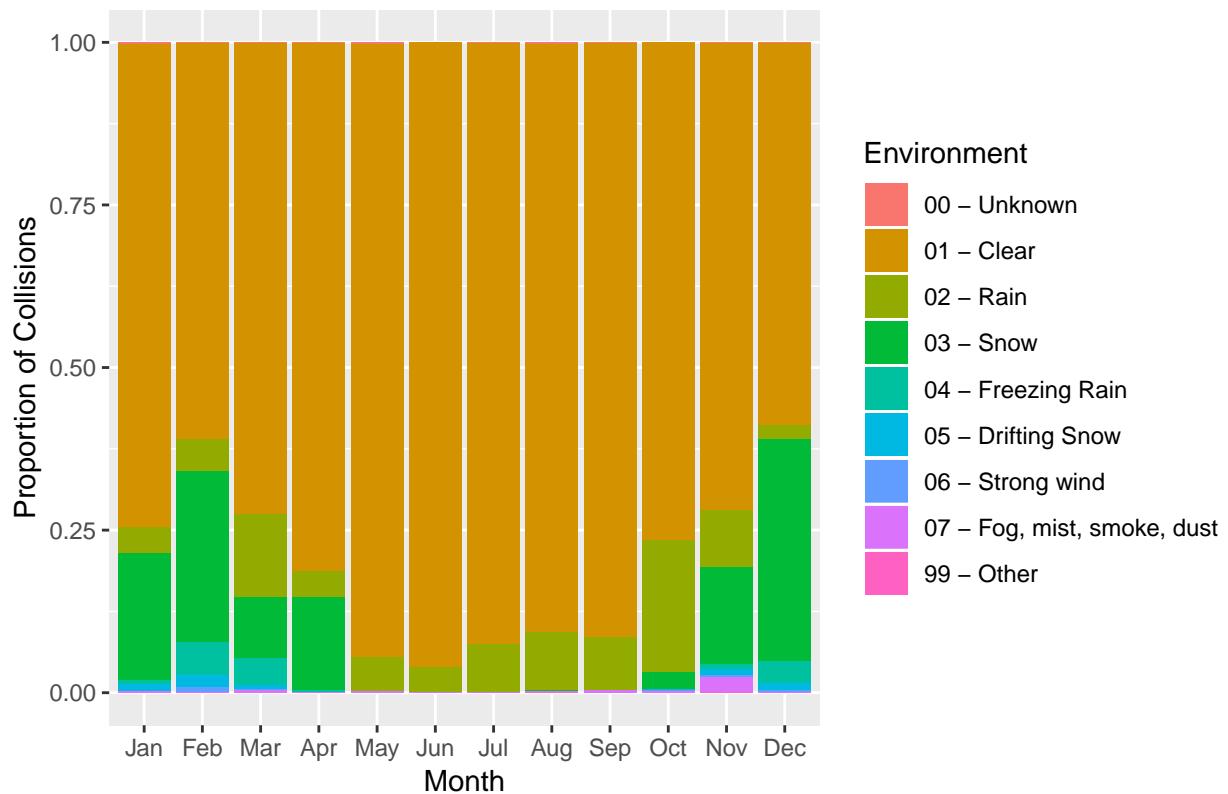
```
#####3) Month and Environment Type/Month and Road_Surface
```

Next, we can plot the distribution of collisions by *Environment* and *Month*, as well as *Road Surface* and *Month*. Again, we expect to see results aligned with common sense.

```
#Month and Environment
```

```
ggplot(df) + geom_bar(aes(Month, fill=Environment), position="fill") + ggtitle("Proportion of Accidents by Month and Environment")
```

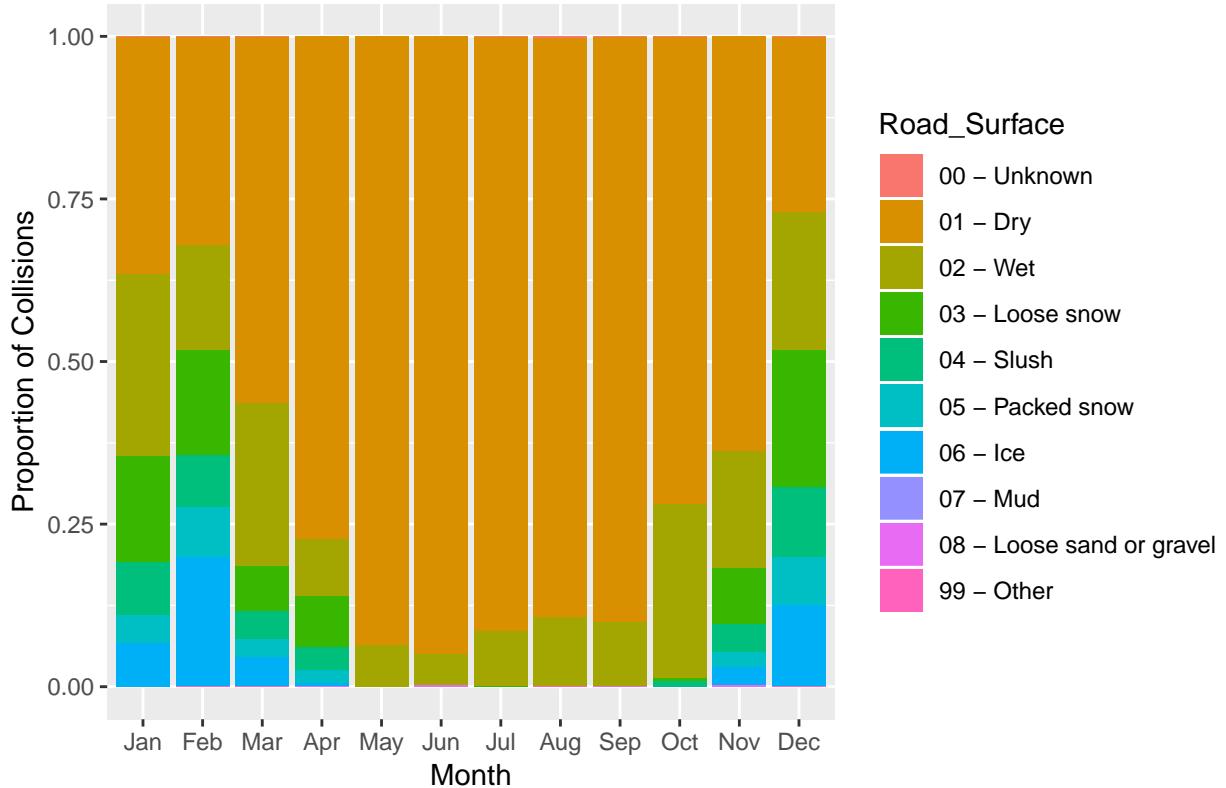
## Proportion of Accidents by Environment Type per Month



#Month and Environment

```
ggplot(df) + geom_bar(aes(Month, fill=Road_Surface), position="fill") + ggtitle("Proportion of Accidents by Month and Environment")
```

## Proportion of Accidents by Road Condition Type per Month



In fact, we see result that we expected. For instance, in Dec-Jan-Feb, between 20% and 40% of collisions happened when it was snowing. However, a bit unusually, there were more accidents in April when it was snowing than in March. Since there usually is less snow in April than in March, this is a interesting. Perhaps winter tires were no longer on the cars, which resulted in more collisions during snowy conditions? Also, fog/mist/smoke/dust seemed to be more present during collisions in November.

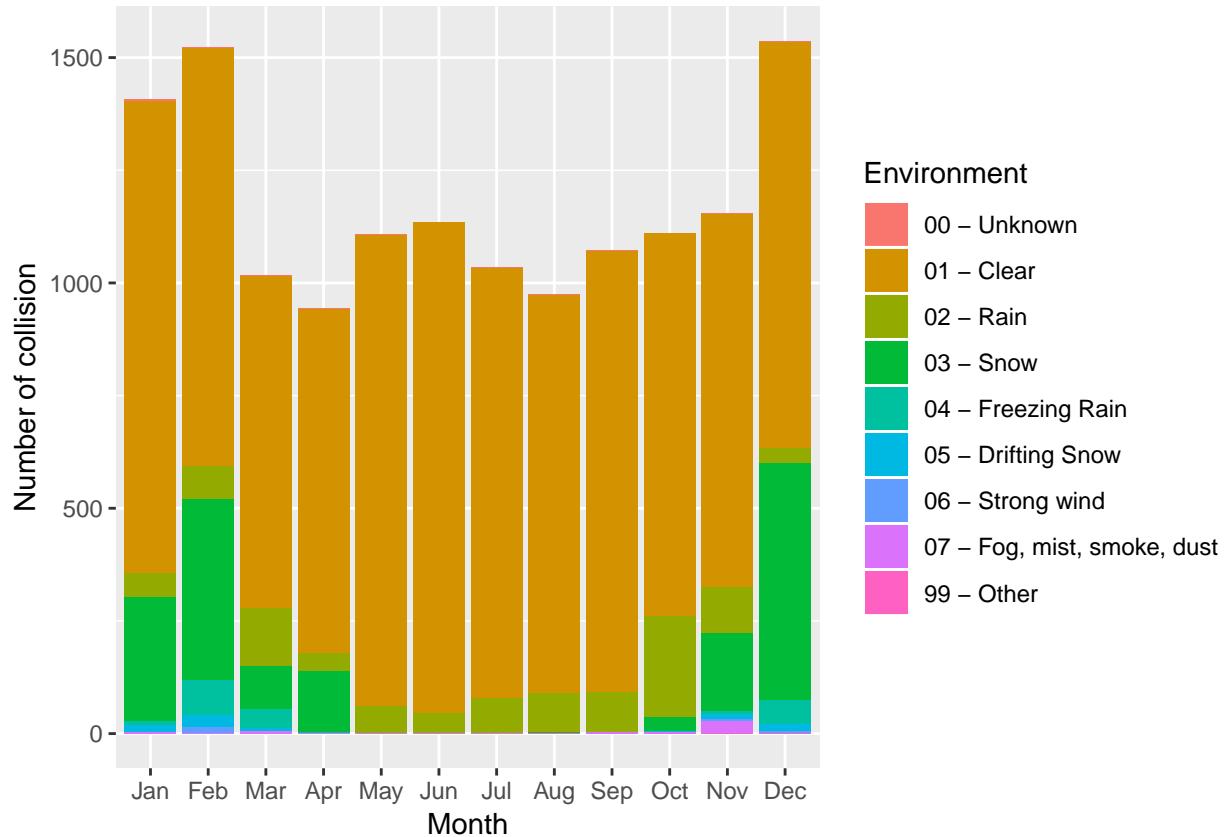
The results for the road surface are equally predictable. In the summer months, 80-90% of collisions happened when the road was dry, while 15-20% of collisions in December and February happened when the road was icy.

These observations are all in line with basic knowledge of Canadian driving conditions. Later on, we will plot the collisions on the map and identify them by road\_surface and environment conditions, which can maybe give us a sense or help us target specific roads that may be problematic in terms of road conditions.

If take a look at the pretty much the same graph, but with the y axis being the total number of collision per month, we can see that the increase of total collision in the Jan, Feb and Dec seems to be partly due to snow. However, we would need to cross-reference this to a dataset of the weather in Ottawa in 2016 in order to determine if the proportion of collision that happened in the snow is greater than general.

On the other hand, we can redo this graph with the number of collisions on the y-axis, instead of the proportion.

```
ggplot(df,aes(x = Month, fill=Environment)) + geom_bar() + xlab("Month") + ylab("Number of collision")
```



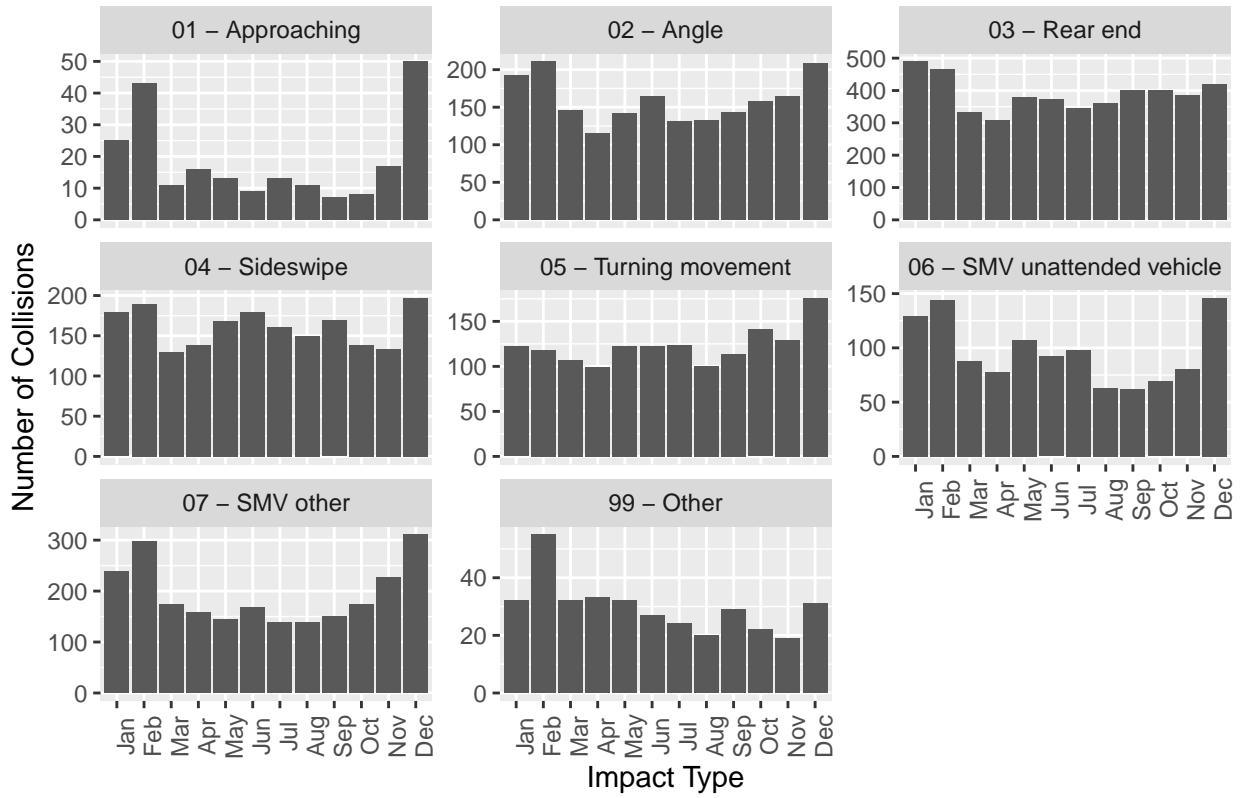
When we look at this version of the plot, it looks like the number of collisions when the environment is clear is more or less consistent throughout the year and it is clear that the higher number of collisions in the winter months is due to the environment at that time of year: snow, freezing rain and drifting snow. Therefore, it could be reasonable to assume that snow and winter weather in general may lead more accidents.

#### #####4) Month and Impact Type

One may wonder if some collisions with specific impact types are more frequent during certain months of the year.

```
ggplot(df) + geom_bar(aes(Month)) +
  facet_wrap(~Impact_type, scales="free_y") + theme(axis.text.x=element_text(angle=90)) + gtitle("Collisio
```

## Collisions by Impact Type and Month



From the above plots, we can see that “Sideswipe”, “Turning movement”, “Rear end”, and “Other” types of impact happen fairly consistently throughout the year (except for “Other” types of impact in February). However, “Approaching”, “SMV unattended” and “SVM” types of impact happened with a little bit more variability across the months. In fact, in these types of collision were more frequent during winter months, which could suggest that weather and road conditions lead to these impact types.

### ####Other Variables

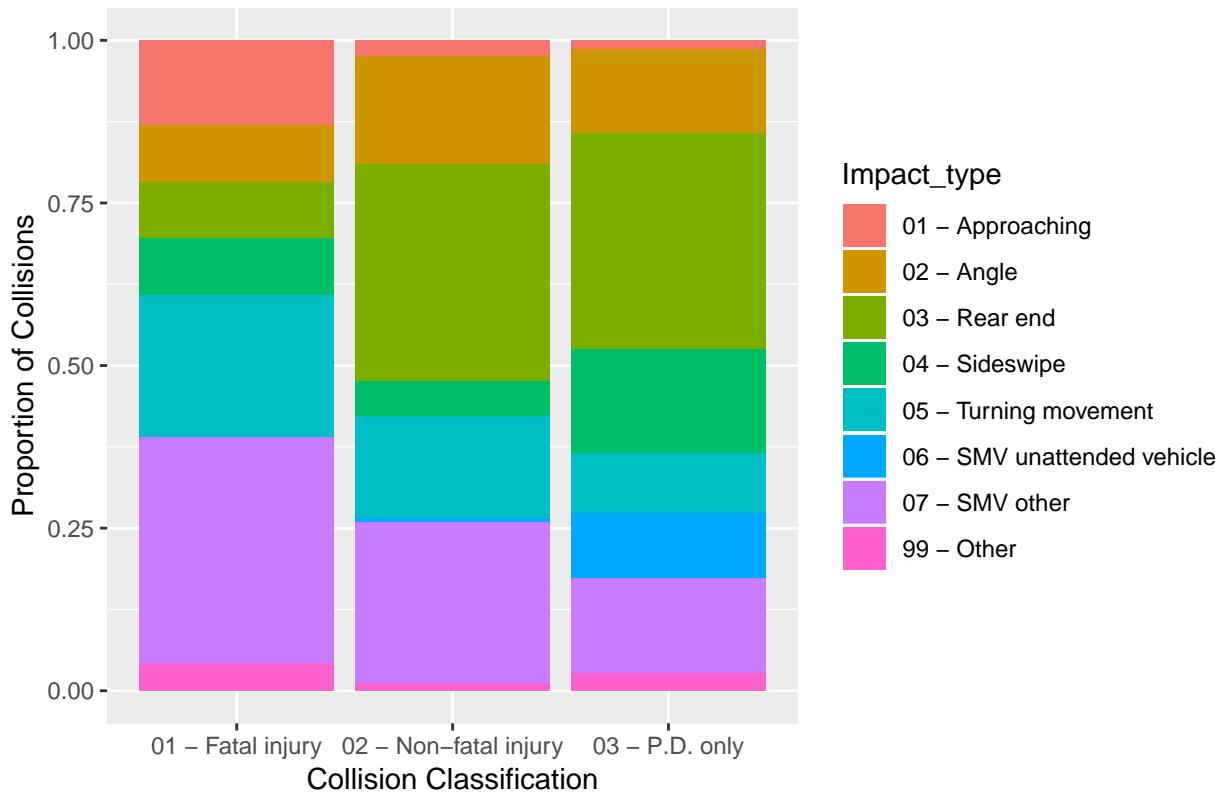
We will also try to observe the relationship between other non time related variables.

#### #####1) Impact Type and Collision Classification

This graph separates the collisions by classification type (fatal injury, non-fatal injury and property damage only) and indicates the proportion of accidents by impact type:

```
ggplot(df) + geom_bar(aes(Collision_Classification, fill=Impact_type), position="fill") + ggtitle("Propo")
```

## Proportion of Collision by Classification and Impact Type

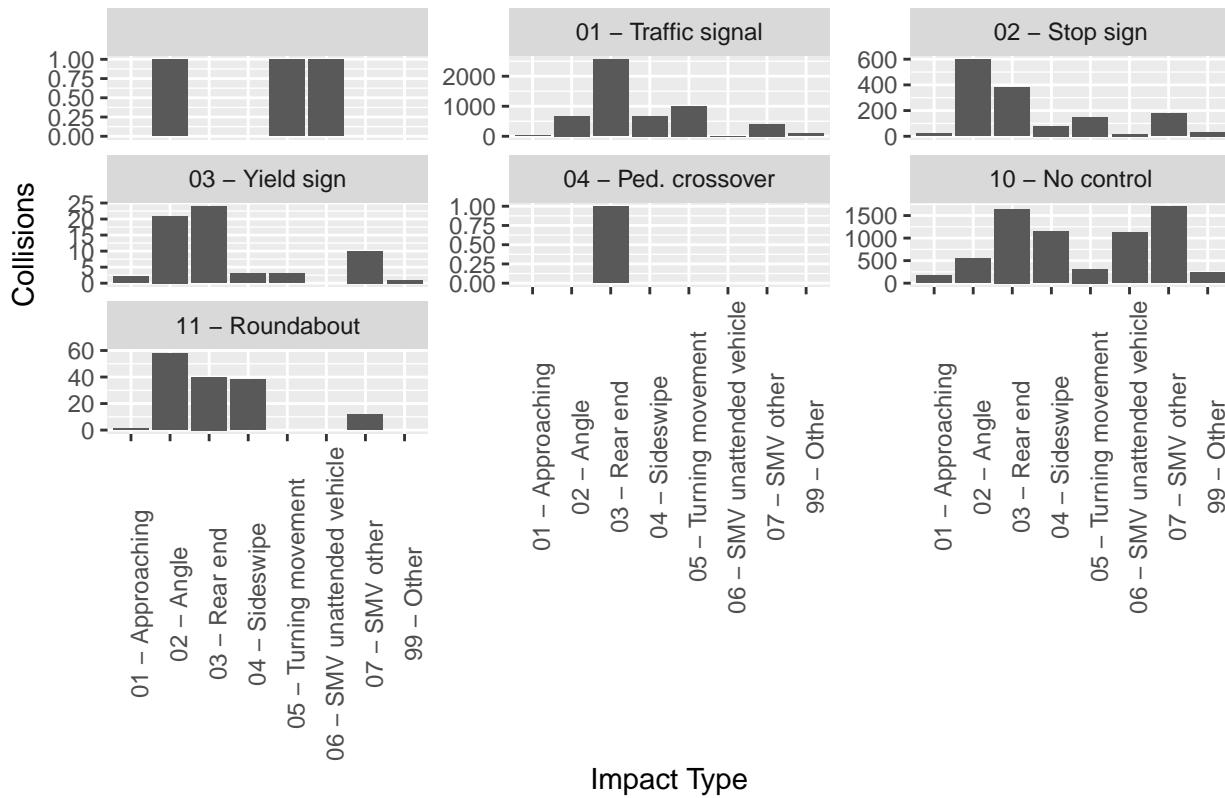


From the graph above, we see that collisions that resulted in fatal injuries usually involved a SMV (slow moving vehicle), a turning movement or an approaching vehicle. On the other hand, non-fatal and p.d only collisions were often rear-end collisions.

#####2) Impact Type and Traffic\_Control

```
ggplot(df) + geom_bar(aes(Impact_type)) +
  facet_wrap(~Traffic_Control, scales="free_y") + theme(axis.text.x=element_text(angle=90)) + ggtitle("Impact Type and Traffic Control")
```

## Collisions by Impact Type per Traffic Control

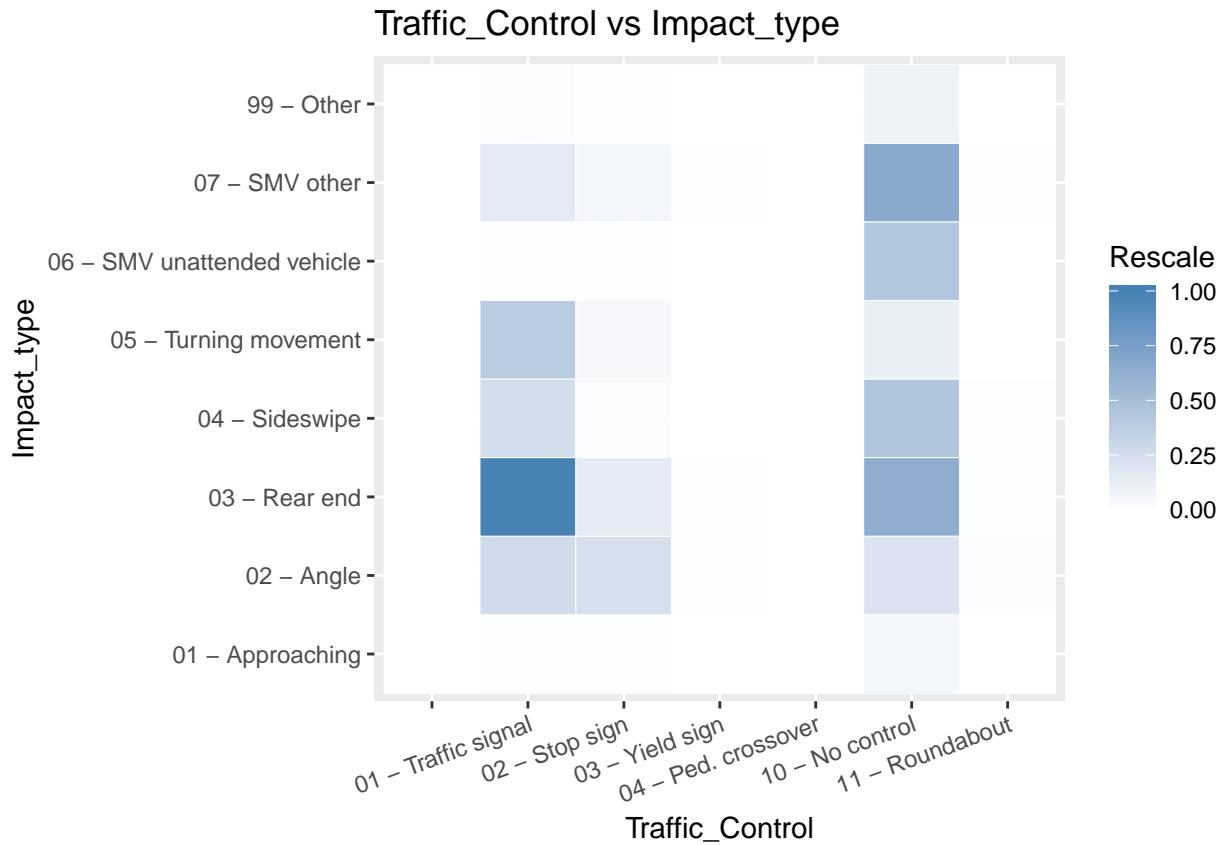


From the above graphs, it is interesting to see that rear end collisions is the most frequent impact type at roundabouts, while angle collisions are the most common form of collisions at stop signs. We may want to reflect on why this is.

Also, we have to be careful in the interpretation of the pedestrian crossover graph. Since there was only one collision involving a pedestrian crossover, the fact that 100% of pedestrian crossover collisions were rear end collisions does not mean anything, since it is only based on one observation.

To visualize the relationship between the two variables, we can also make a heatmap:

```
dat <- df %>% select(Traffic_Control, Impact_type) %>% table() %>% melt()
colnames(dat) <- c("Traffic_Control", "Impact_type", "Value")
dat$Rescale <- rescale(dat$value)
ggplot(dat, aes(Traffic_Control, Impact_type)) + geom_tile(aes(fill = Rescale), color = "white") + scale
```



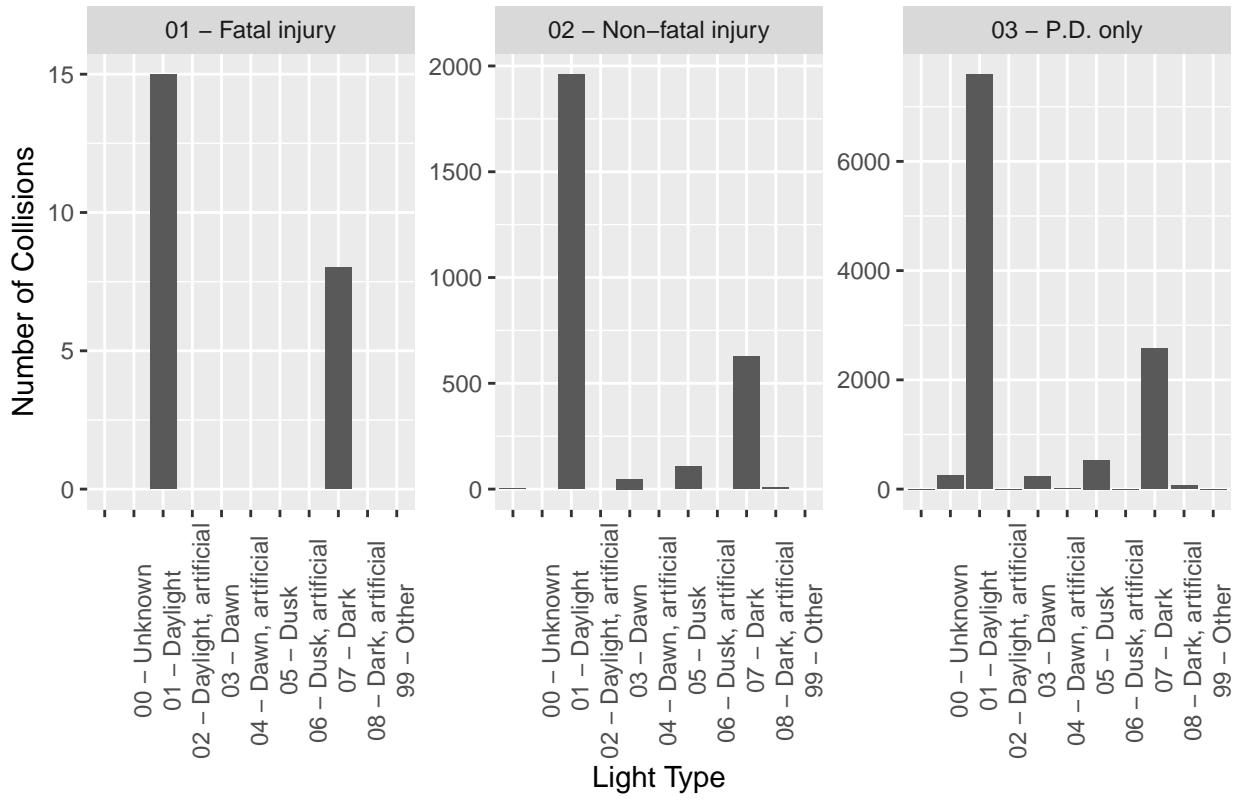
As mentioned previously, rear-end collisions were the most frequent types of collisions at traffic signals, but with this heatmap, we can also clearly see that when there is no type of traffic control, the impact types are a lot more varied. This is not surprising because the fact that there is not traffic structure leads to a wider range of road shenanigans a less organized road system, which could perhaps lead to more accidents.

#### #####3) Light and Collision Classification

In the next graph, we will try to determine links between the type of light at the time of the collision, and the collision classification:

```
ggplot(df) + geom_bar(aes(Light)) +
  facet_wrap(~Collision_Classification, scales="free_y") +
  theme(axis.text.x=element_text(angle=90)) +
  scale_fill_brewer(palette = "Set1", name = "Light") + ggtitle("Number of Accidents by Light Type per Collision Classification")
```

## Number of Accidents by Light Type per Collision Classification

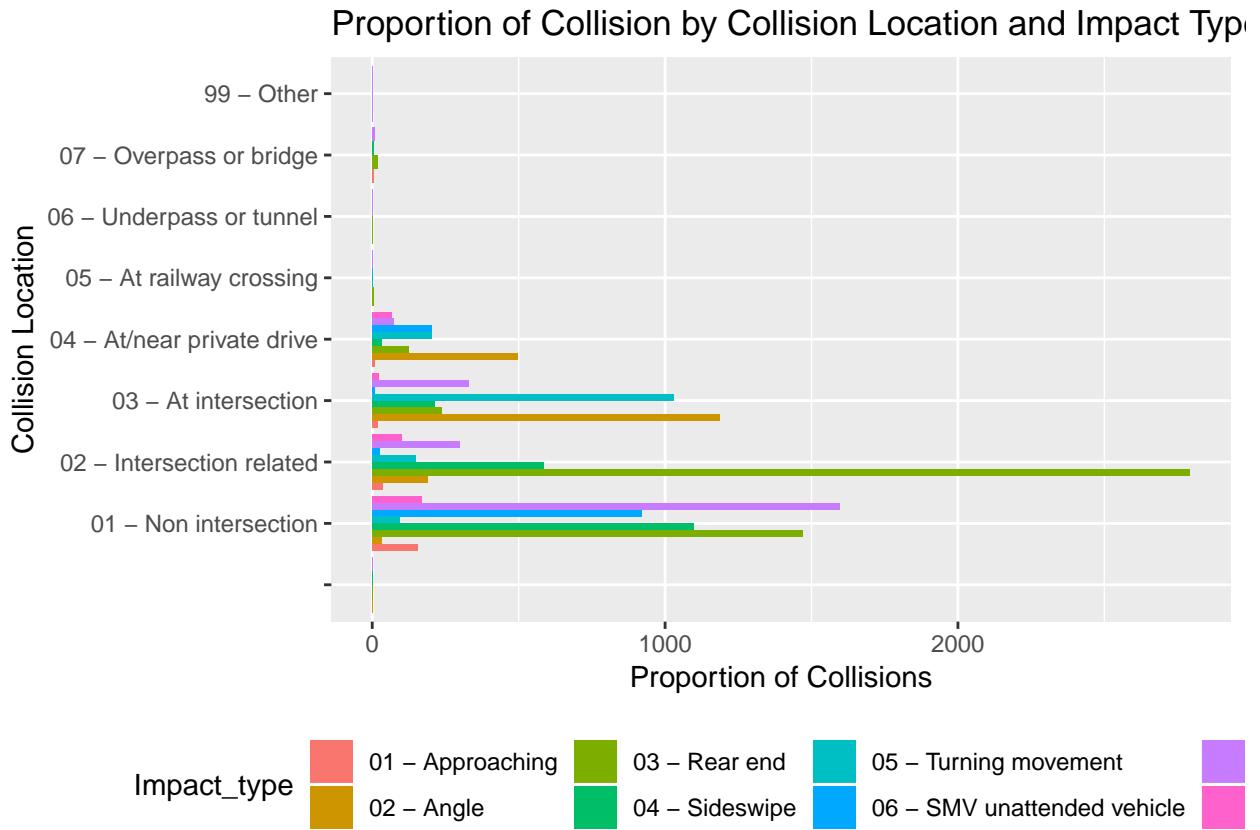


Across the different collision classifications, the trend regarding the light is quite evident : most accidents in all classification types occur during the daylight, which makes sense since there are presumably more cars on the road at this time. Most of the collisions that don't happen during daylight happen when it is dark outside.

### #####4) Impact Type and Collision Location

We might be interested to see the distribution of the impact type at different collision locations.

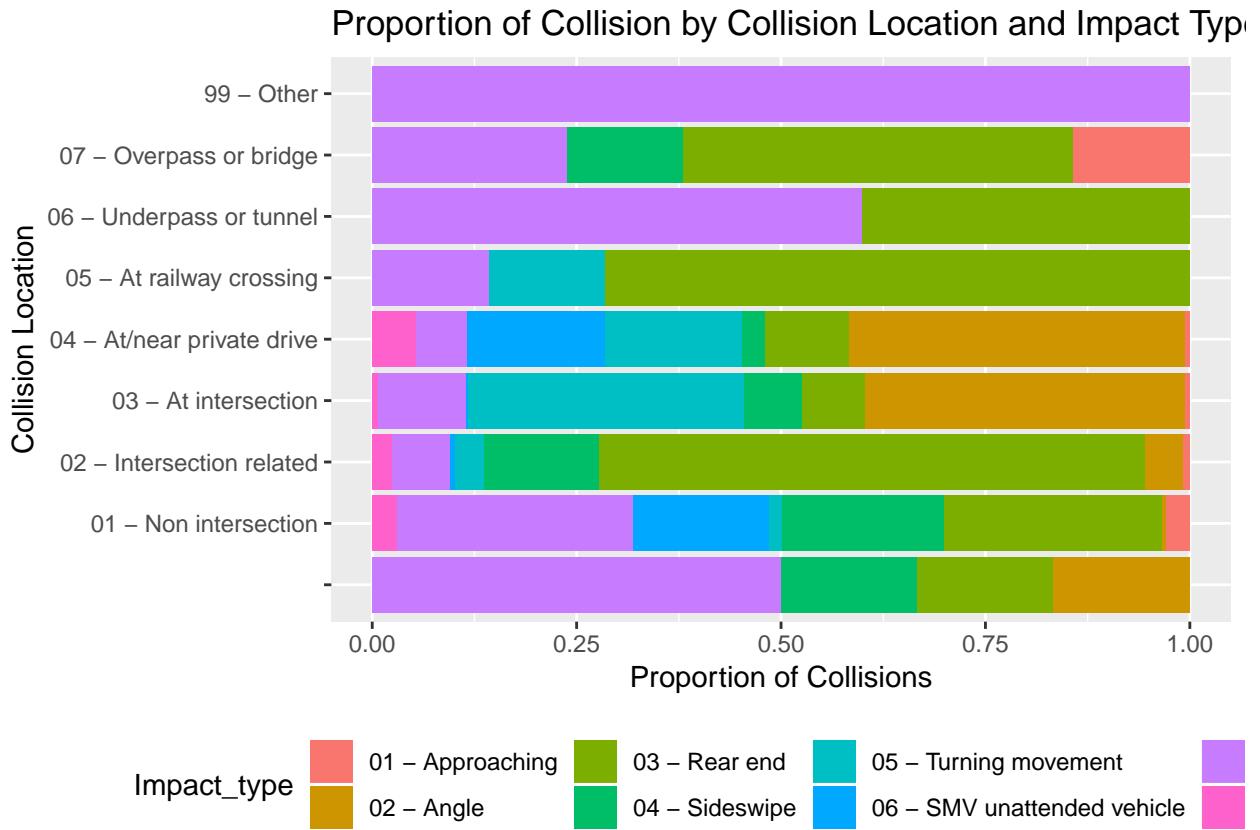
```
ggplot(df) + geom_bar(aes(Collision_Location, fill=Impact_type), position="dodge") + ggttitle("Proportion")
```



Concerning intersection related collisions, we can see that rear-end collisions are the most frequent impact types, with the sideswipe impact being a distant second. However, for collisions that happened at an intersection, angle and turning movement were the most common types of impacts. As for non intersection collisions, SMV and rear-end collisions are the most frequent impact types, with SMV unattended vehicle and Sideswipe also being recurring collision impact types. On the other hand, angle was the most common impact type for collisions that happened at or near a private drive.

Since the collisions happen much more often at the above four collision locations, it is hard to interpret the graph for the less common locations. We could try to visualize the information in a different way:

```
ggplot(df) + geom_bar(aes(Collision_Location, fill=Impact_type), position="fill") + ggtitle("Proportion of Collision by Collision Location and Impact Type")
```



Other than the four collisions already mentioned, only the “Overpass or Bridge” location is probably worthwhile to interpret, since the other locations have less than 10 instances each, which is not sufficient to draw any conclusions. From the above plot, we see that rear end collisions represent about half of collisions on overpasses or bridges, while roughly 25% are SVM collisions.

## Word Map

Since we have the name of each street/intersection of where each collision happened, we can try to do some word cloud map to see which road comes up more often.

```
library("tm")
## Loading required package: NLP
##
## Attaching package: 'NLP'
##
## The following object is masked from 'package:ggplot2':
## 
##     annotate
library("wordcloud")
## Loading required package: RColorBrewer
library("RColorBrewer")

docs <- Corpus(VectorSource(df$Location))
docs <- tm_map(docs, tolower)

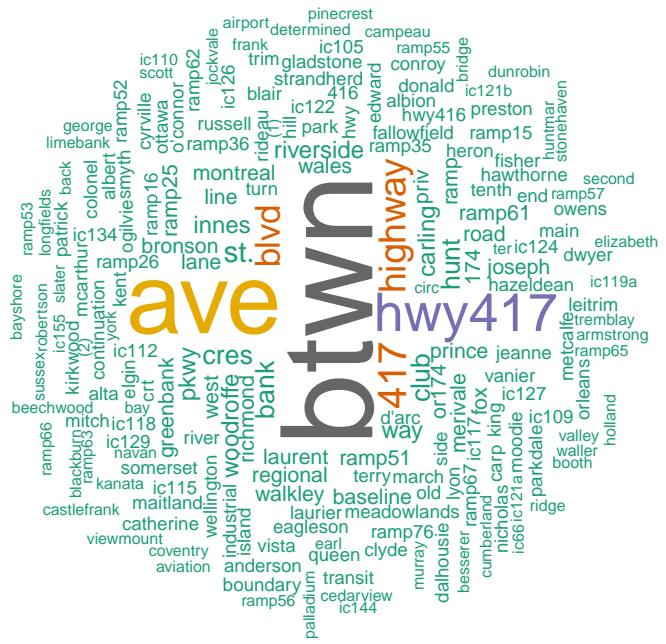
## Warning in tm_map.SimpleCorpus(docs, tolower): transformation drops
```

```

## documents

dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
set.seed(1)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
           max.words=200, random.order=FALSE, rot.per=0.35,
           colors=brewer.pal(8, "Dark2"))

```



From this, we can see that the most used word is

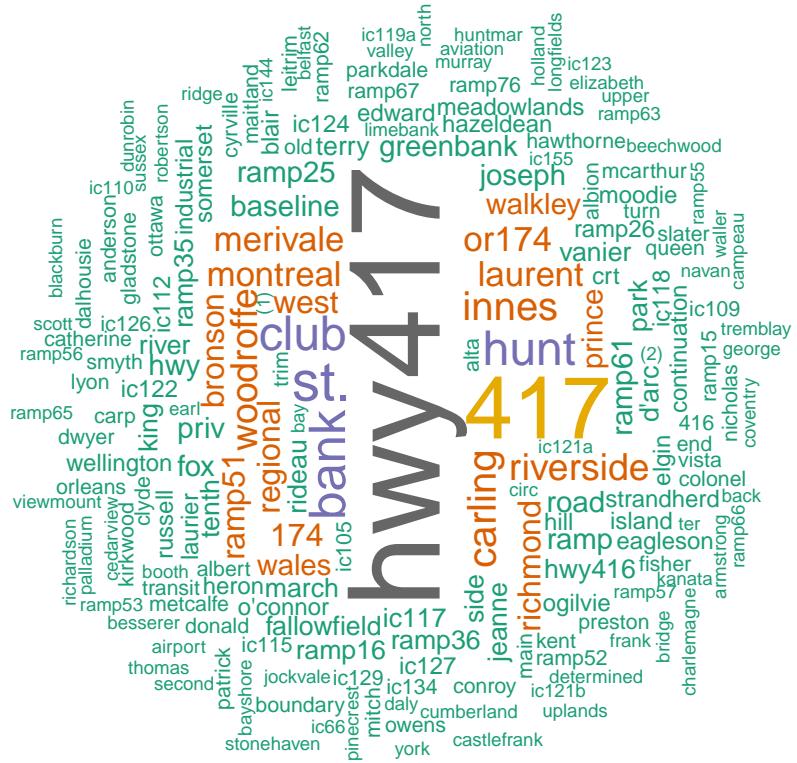
btwn, which doesn't give much more insight other than how the collision are reported. This between is probably the most frequent word because many accidents were probably reported as having happened "between" two specific roads. Although, we can also see that words like ave, blvd, and highway are frequent, telling us that more collision were reported on those type of road.

We could try to remove some popular road type name, in order to get a word cloud containing mostly road names.

```
docs_road <- tm_map(docs, removeWords, c("ave", "btwn", "blvd", "pkwy", "highway", "st.", "st.", "cres", "way"))

## Warning in tm_map.SimpleCorpus(docs, removeWords, c("ave", "btwn",
## "blvd", : transformation drops documents

dtm <- TermDocumentMatrix(docs_road)
m <- as.matrix(dtm)
v <- sort(rowSums(m), decreasing=TRUE)
d <- data.frame(word = names(v), freq=v)
set.seed(1)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```



When we take out the unnecessary words, we see that the road names that come up more are the ones that we expect to see, such as Highway 417, Carling Ave, Bronson Ave, Innes, and Riverside. They are all very frequently and highly travelled roads, thus it is not surprising that they have a higher frequency of accidents than less popular or travelled roads. Therefore, this road map coincides with what we observed on the maps, and what we would deduce from our general knowledge of Ottawa roads.

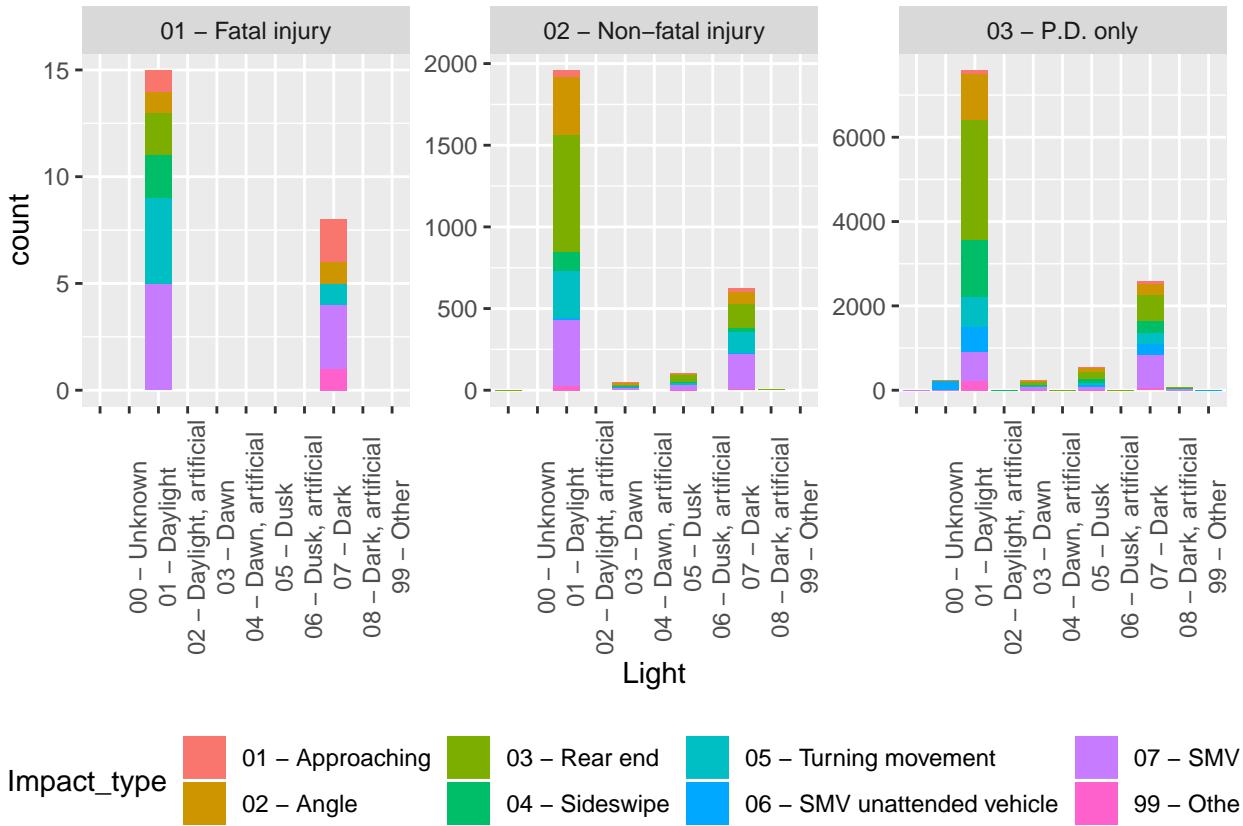
## Multivariate Visualizations

Finally, it may be interesting to observe the interactions of more than 2 variables at a time. In this next section, multivariate visualizations will hopefully give us interesting information about the data.

## Collision Classification and Light Type by Impact Type

To start off, we can check if there seems to be trends regarding impact type and light, depending on the severity of the collision.

```
#Light and impact_type by collision_classification
ggplot(df) +geom_bar(aes(Light,
  fill = Impact_type)) +
  theme(legend.position="bottom",
        axis.text.x=element_text(angle=90)) +
  facet_wrap(~Collision_Classification, scales="free_y")
```

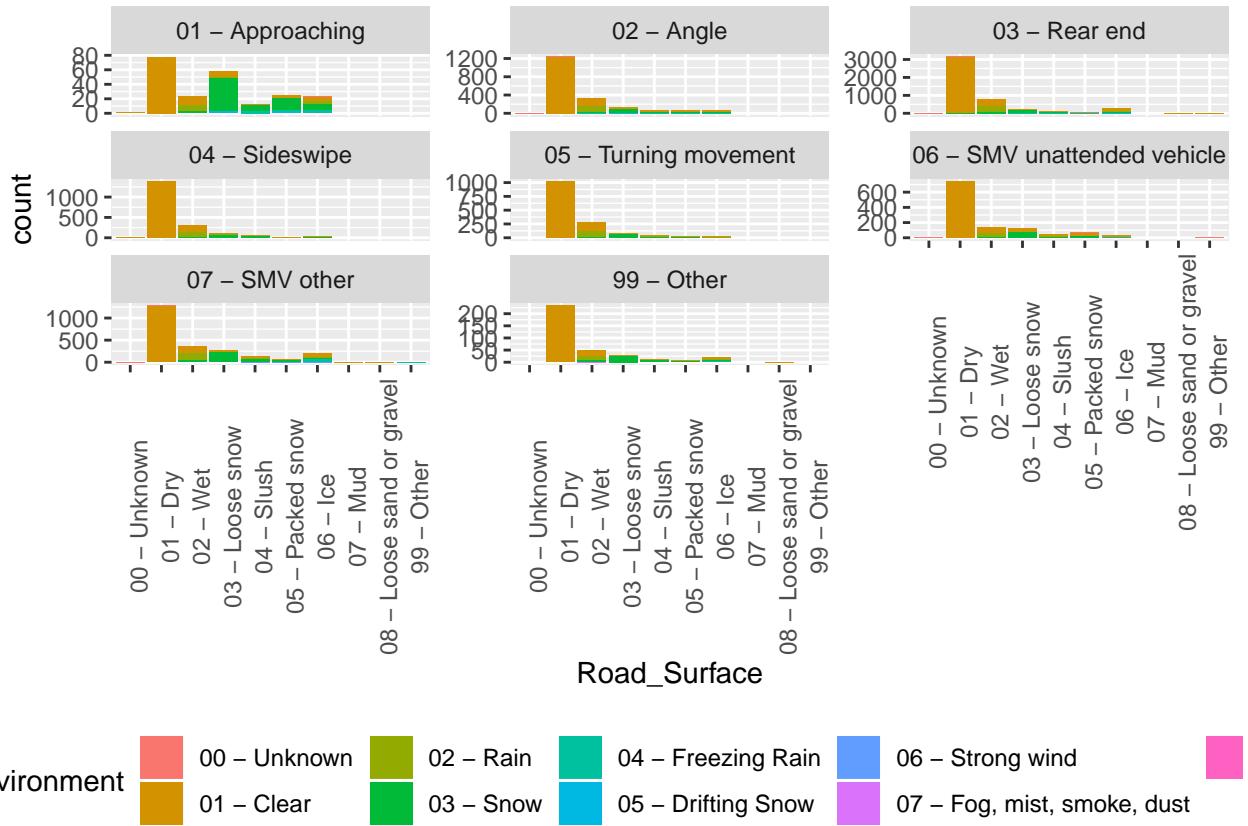


As we have previously seen, almost all collisions happened during the daylight hours, or when it was dark outside. However, while keeping in mind that the number of fatal collisions is small, it is interesting to observe that when it is dark outside, approaching and other types of collisions were more frequent, but turning movement, rear end and sideswipe collisions were more frequent in the daylight. Furthermore, for non-fatal injury and property damage only collisions, the distribution of impact types for collisions in the dark was fairly similar, while the proportion of collisions in the daylight in the form of turning movement was higher for collisions involving non-fatal injuries, while the proportion of sideswipe collisions was higher for the p.d. only collisions in the daylight.

### Impact Type by Environment and Road Surface

We may wonder if a specific type of impact is prone to happening in different weather conditions, therefore plotting the collisions by impact type for different environment and road conditions may be a good idea.

```
#Impact_type by environement and road surface
ggplot(df) +geom_bar(aes(Road_Surface,
  fill = Environment)) +
  theme(legend.position="bottom",
  axis.text.x=element_text(angle=90)) +
  facet_wrap(~Impact_type, scales="free_y")
```



Again, as mentioned before, dry road conditions have the highest frequency of accidents, no matter the impact type, but this is to be expected. Wet road conditions (due to rain) and loose snow (due to snow) are usually the runner ups in terms of road surface considered to have a higher frequency of accidents, and this is true across almost all impact types. However, there were more collisions of the type approaching occurred when the road conditions were snowy or icy than when the roads were dry. This is maybe worth investigating in more details, since we do not see such a trend for other impact types.

### Impact Type by Environment and Road Surface for Approaching Collisions

From the previous observations, it seems that there is a high number of impact of type “01 - Approaching” that occurred when it was snowing or when it was icy. We can get a better picture of this with the following graph :

```
library(gridExtra)

##
## Attaching package: 'gridExtra'

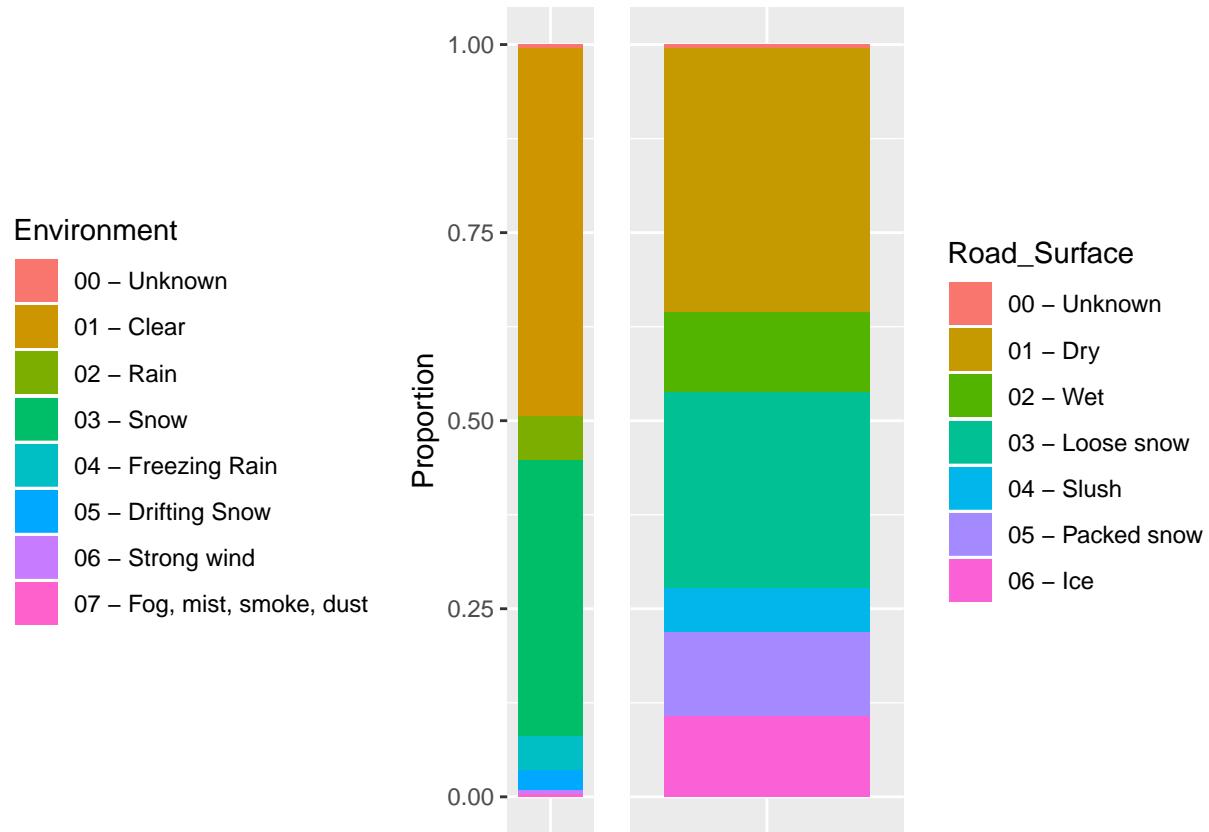
## The following object is masked from 'package:dplyr':
##   combine

p <- ggplot(df %>% filter(Impact_type == "01 - Approaching"))
p1 <- p + geom_bar(aes(Impact_type, fill=Environment), position="fill") + ylab("Proportion") +
  theme(legend.position = "left",
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
p2 <- p + geom_bar(aes(Impact_type, fill = Road_Surface), position = "fill") +
```

```

theme(legend.position = "right",
      axis.title.x=element_blank(),
      axis.text.x=element_blank(),
      axis.ticks.x=element_blank(),
      axis.text.y = element_blank(),
      axis.title.y = element_blank(),
      axis.ticks.y = element_blank())
grid.arrange(p1,p2, ncol=2)

```



It is clear that more than half of the collisions of type approaching happened when the road was wet, snowy or icy. This correlates with the fact that during half of all the approaching type of collisions, it was snowing or raining. Therefore, we may want to investigate these approaching types of collisions in winter specifically, since all these observations seem to indicate that maybe these types of collisions are more susceptible to happening in the winter weather.

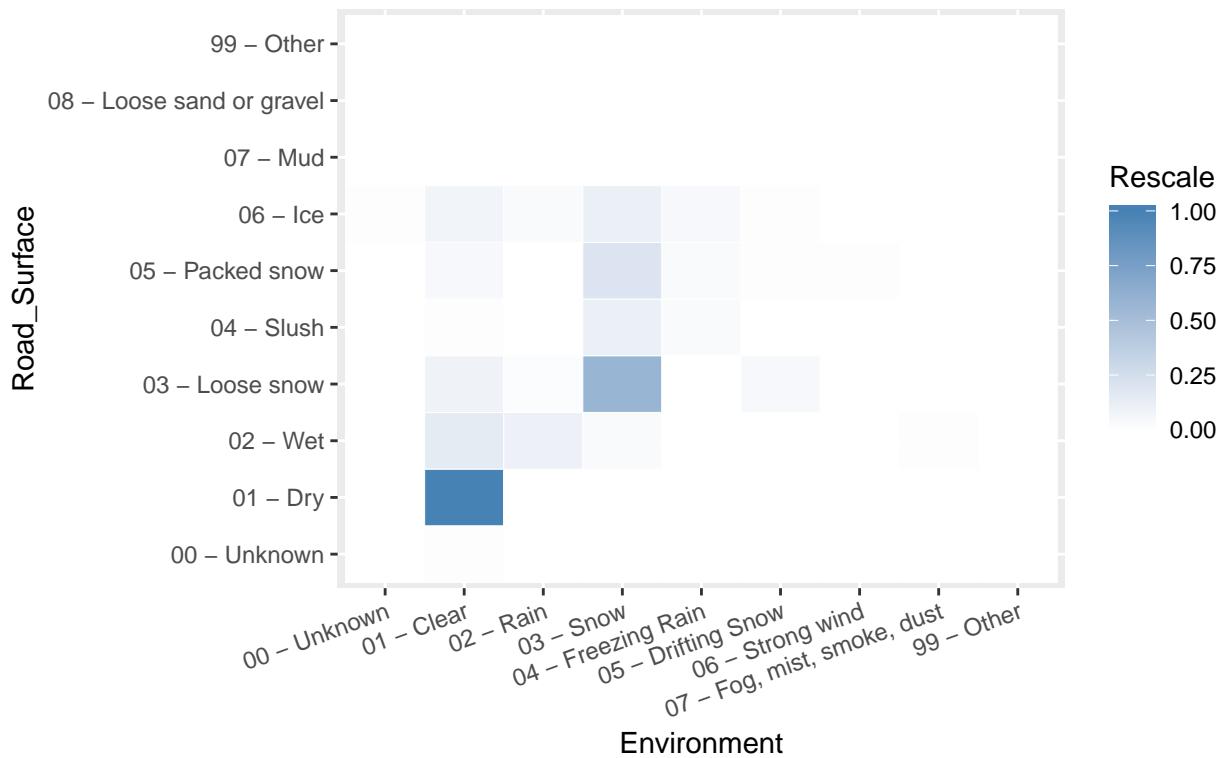
Another way to observe this relationship between approaching collisions and winter weather/conditions is with a heat map:

```

appr_only <- df %>% filter(Impact_type == "01 - Approaching")
dat <- table(appr_only$Environment, appr_only$Road_Surface)
dat <- melt(dat)
colnames(dat) <- c("Environment", "Road_Surface", "Value")
dat$Rescale <- rescale(dat$Value)
ggplot(dat, aes(Environment, Road_Surface)) + geom_tile(aes(fill = Rescale), color = "white") + scale_fi

```

Heatmap of Environment vs Road Surface  
considering only approaching impact.



Again here, we can see that snow seems to be an important factor in approaching types of collisions.

#### #### Map Visualizations

We can again plot the collisions points on a map of Ottawa, but identify the collisions more specifically with regards to their characteristics, like collision classification or time of the day or light.

#### ##### 1) Map with regards to Impact Type

```
ggmap(ottawa.map3)+ geom_point(data = data_with_lat.long, aes(x=Longitude, y=Latitude, color=Impact_Type))

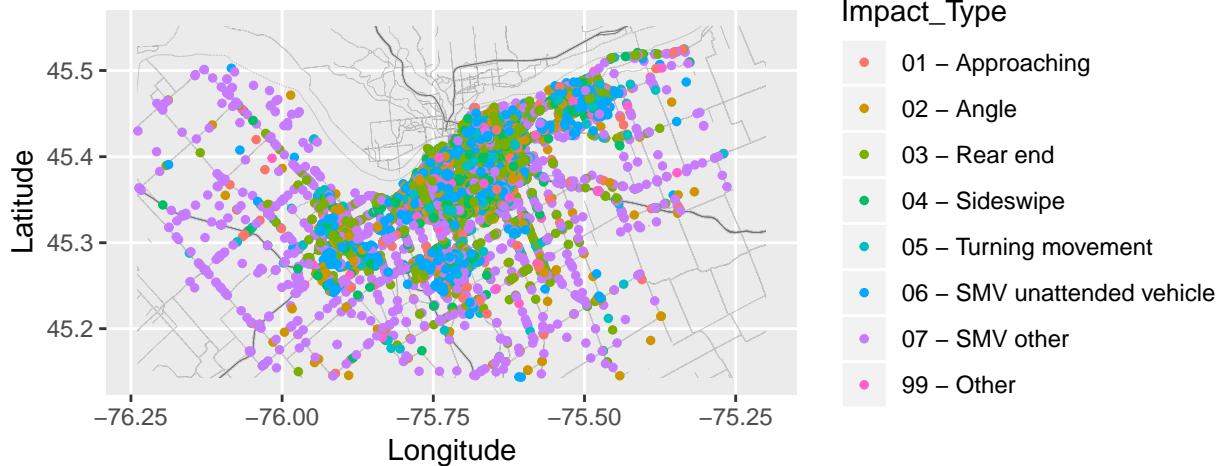
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.

## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.

## Warning: Removed 1 rows containing missing values (geom_rect).

## Warning: Removed 125 rows containing missing values (geom_point).
```

## 2016 Collisions in Ottawa by Impact Type



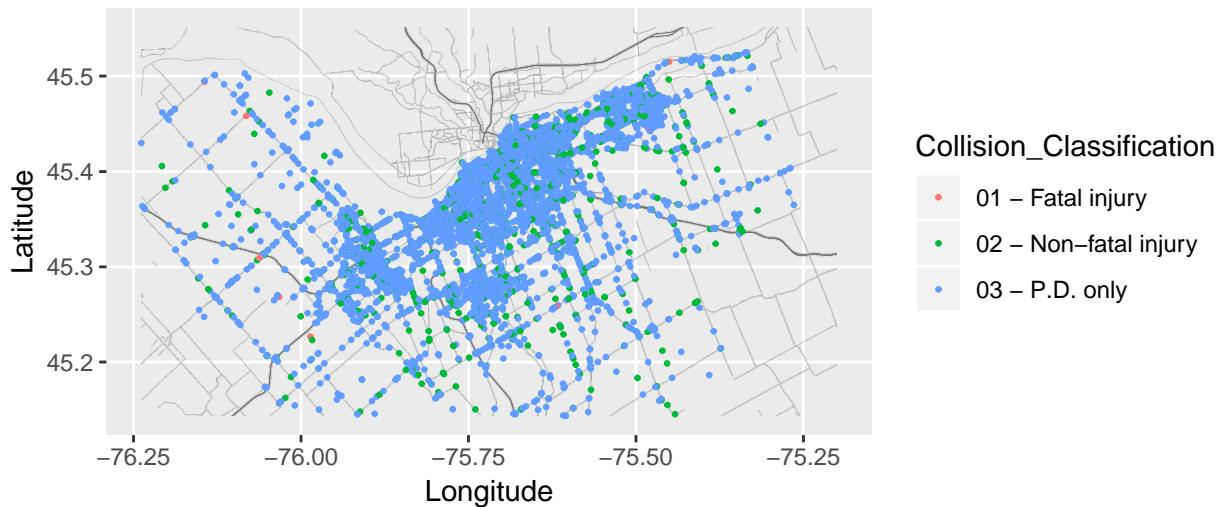
This plot is interesting because we can see that a big proportion of the accidents in “rural” areas of Ottawa have SMV other collisions. This is interesting but not entirely surprising, since there are probably more slow moving vehicle outside big city centres, such as farm vehicles. Still, the amount of purple in rural areas could be concerning, and we may want to address that in further analysis.

#####2) Map with regards to Collision Classification

Next, locating fatal collisions, non-fatal injury collisions and p.d only collisions on the map may give us more information as to where certain types of collisions are more common.

```
ggmap(ottawa.map3)+ geom_point(data = data_with_lat.long, aes(x=Longitude, y=Latitude, color=Collision_Classification))
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
## Warning: Removed 1 rows containing missing values (geom_rect).
## Warning: Removed 125 rows containing missing values (geom_point).
```

## 2016 Collisions in Ottawa by Collision Classification



Obviously, most collisions are P.D only. However, we see a few fatal injury collisions that stand out in the

more rural areas. It is hard to see if there are any fatal collisions closer to downtown Ottawa, therefore we will plot fatal injuries only in the next few plots.

#### #####3) Fatal Collisions Only

Fatal collisions might be of interest, because they are the most severe and serious collisions. Here are a few plots looking at fatal collisions on the map, with various other variables of interest.

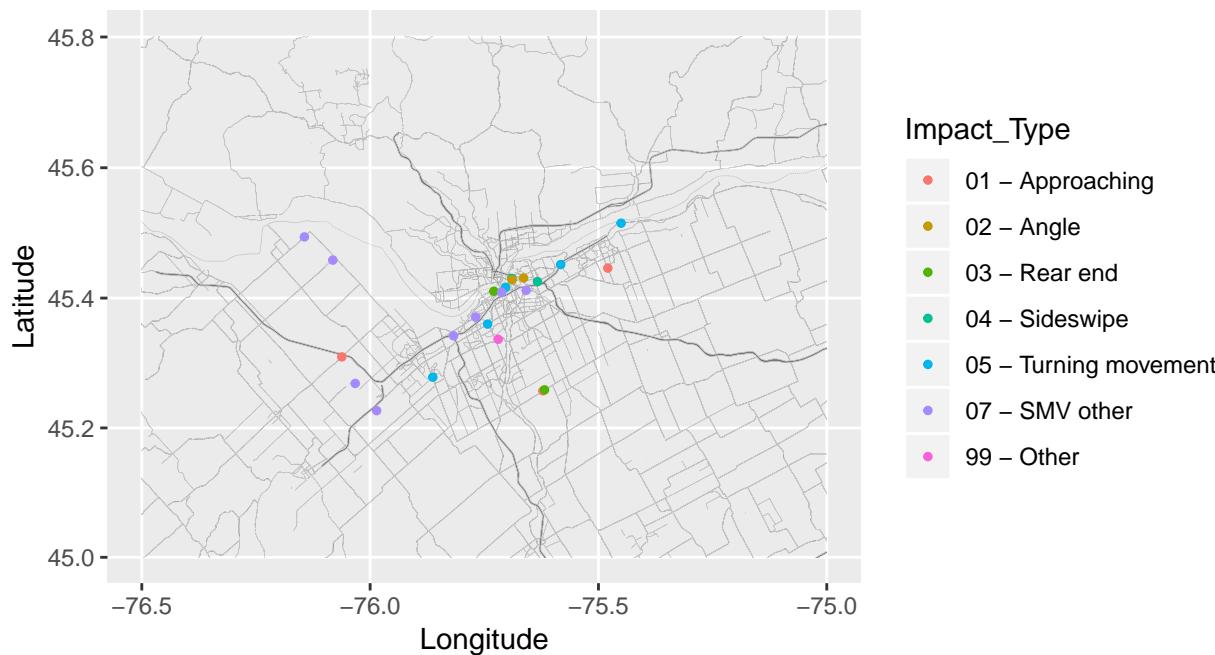
```
#Impact_Type
```

```
ggmap(ottawa.map)+ geom_point(data = data_with_lat.long %>% filter(Collision_Classification == "01 - Fa
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```

### 2016 Fatal Collisions in Ottawa by Impact Type



Plotting only the fatal collisions, we see that they seem to be spread out randomly and without a clear trend across the city. Also, there were many impact types, but there seems to be a higher frequency of “SMV other” impact types, as well as “Turning movement”.

Let’s now look at the road surface of fatal collisions:

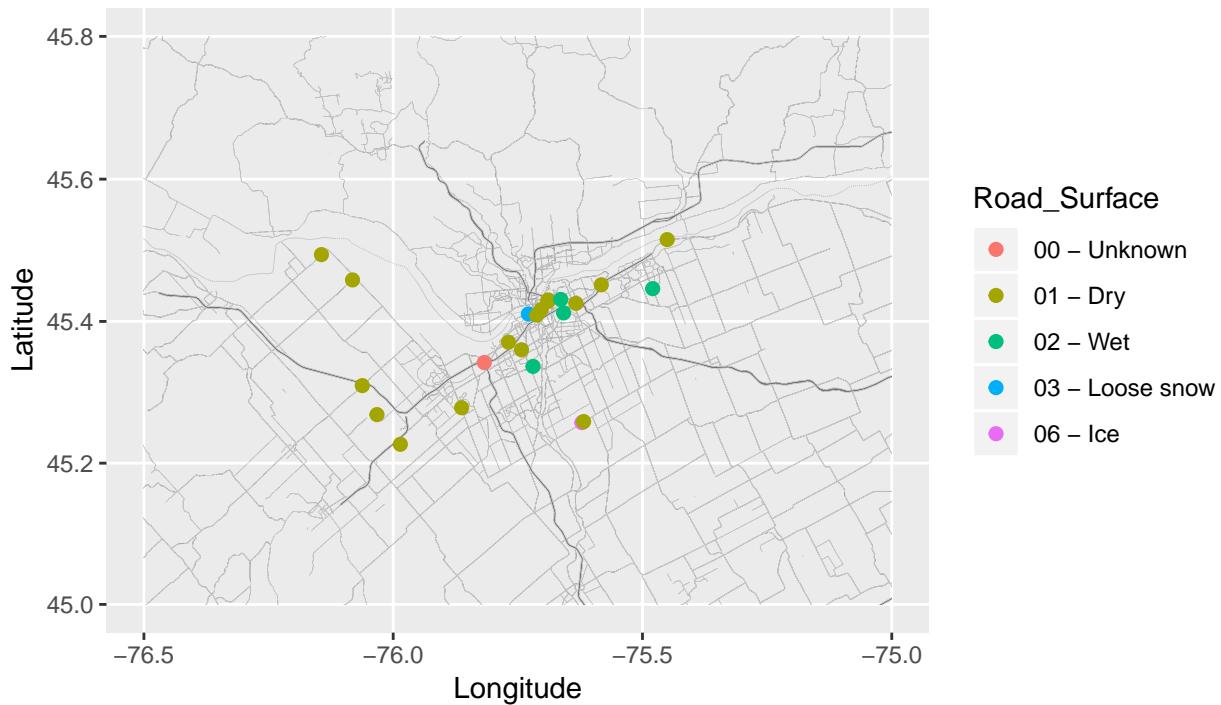
```
#Road_Surface
```

```
ggmap(ottawa.map)+ geom_point(data = data_with_lat.long %>% filter(Collision_Classification == "01 - Fa
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```

## 2016 Fatal Collisions in Ottawa by Road Surface



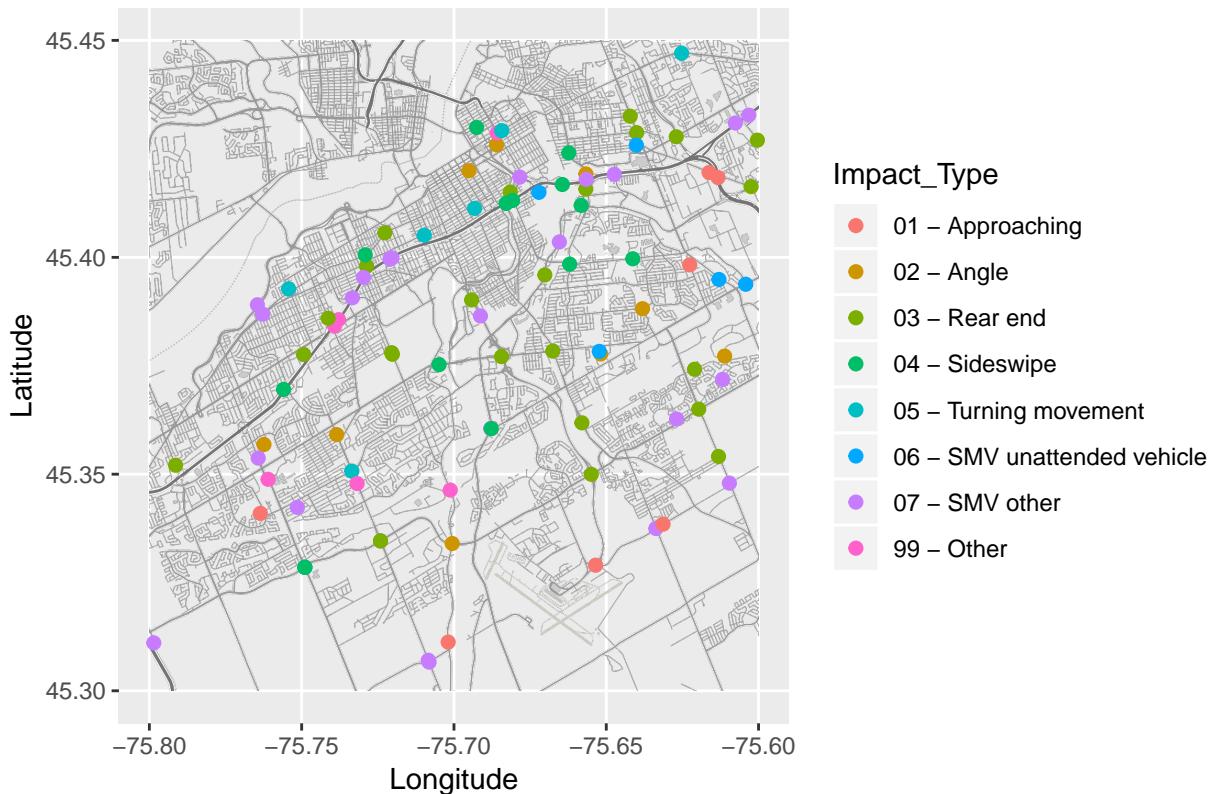
The above plot demonstrates that more than half of the fatal collisions happened when the road surface was dry, and a lot of these happened outside city centres. Most of the accidents that occurred when the road was wet took place in the city or in suburban areas.

### #####4) Collisions in Freezing Rain and Impact Type

Since Freezing Rain is not unheard of during the winter in Ottawa, it can be interesting to determine if any impact types are more frequent in certain areas of the city when it is freezing rain.

```
ggmap(ottawa.map.zoom)+ geom_point(data = data_with_lat.long %>% filter(Environment == "04 - Freezing Rain"))
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
## Warning: Removed 100 rows containing missing values (geom_point).
```

## 2016 Collisions in Freezing Rain by Impact Type in Ottawa



In terms of environment, and more specifically when it is freezing rain, the collisions seem to be distributed roughly uniformly on the map with regards to impact type, with seemingly no interesting trend.

### Riverplot

Finally, the following riverplots allow us to visualize the data in a different way.

```
library(riverplot)
library(RColorBrewer)
```

```
#### Non-Fatal Injury Collisions and Impact Type
```

We have not yet focused on the non-fatal injury types of collisions, therefore the next visualization yields information on these kinds of collisions and their impact types.

```
fatal_only <- df[which(df$Collision_Classification == "02 - Non-fatal injury"),]
fatal_only$Collision_Classification <- droplevels(fatal_only$Collision_Classification)
v2 <- fatal_only$Collision_Classification
v1 <- as.factor(fatal_only$Impact_type)
v3 <- as.factor(fatal_only$Road_Surface)

#from_label <- str_extract(levels(v1), "([a-zA-Z]+)( ?)([a-zA-Z]+)")
#to_label <- str_extract(levels(v2), "([a-zA-Z]+)( ?)([a-zA-Z]+)")

#levels(v1) <- paste(from_label)

edges <- as.data.frame(table(v1, v2))
colnames(edges) <- c("N1", "N2", "Value")
nodes <- data.frame(
```

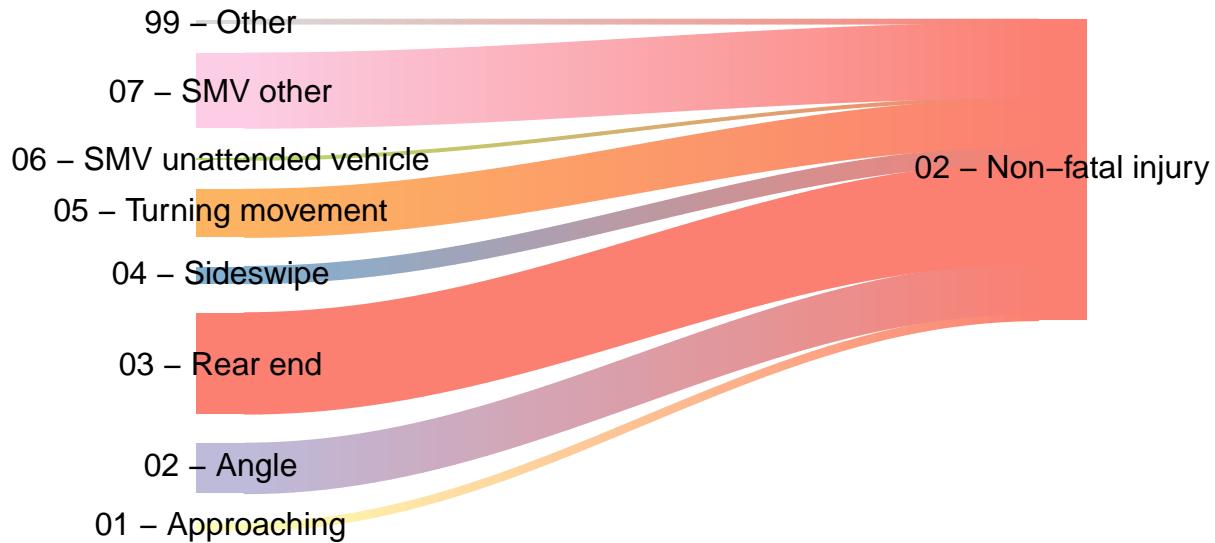
```

ID = c(levels(v1), levels(v2)),
x = c(rep(1,length(levels(v1))),rep(2,length(levels(v2))))
)

palette = paste0(brewer.pal(9, "Set3"))
styles <- lapply(
  c(1:8,3),
  function(n) {
    list(col = palette[n+1])
  })

names(styles) <- nodes$ID
r <- makeRiver(nodes, edges, node_styles = styles)
plot(r, srt=0, lty=0, node_margin = 0.1, plot_area = 1, fix.pdf = T)

```



```

## -----
## Default style:
## -----
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##

```

```

## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
## -----
## [1] "checking edges"
## [1] "adding mid points"
## Updated styles:
## $`01 - Approaching`
## $col
## [1] "#FFFFB3"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`02 - Angle`
## $col
## [1] "#BEBADA"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol

```

```

## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`03 - Rear end
## $col
## [1] "#FB8072"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`04 - Sideswipe
## $col
## [1] "#80B1D3"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1

```

```

##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`05 - Turning movement`
## $col
## [1] "#FDB462"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`06 - SMV unattended vehicle`
## $col
## [1] "#B3DE69"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##

```

```

## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`07 - SMV other`
## $col
## [1] "#FCCDE5"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`99 - Other`
## $col
## [1] "#D9D9D9"
##
## $srt
## [1] 0
##
## $nodestyle

```

```

## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`02 - Non-fatal injury`
## $col
## [1] "#FB8072"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`01 - Approaching->02 - Non-fatal injury`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"

```

```

## 
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`02 - Angle->02 - Non-fatal injury`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`03 - Rear end->02 - Non-fatal injury`
## $srt
## [1] 0
##

```

```

## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`04 - Sideswipe->02 - Non-fatal injury`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`05 - Turning movement->02 - Non-fatal injury`
## $srt

```

```

## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`06 - SMV unattended vehicle->02 - Non-fatal injury`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##

```

```

## $`07 - SMV other->02 - Non-fatal injury`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`99 - Other->02 - Non-fatal injury`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")

```

```

## [1] "list"           "riverplotStyle"
##
## -----
## [1] "calculating sizes"
## $lefts
##          01 - Approaching          02 - Angle
##                      0                  0
##          03 - Rear end            04 - Sideswipe
##                      0                  0
##          05 - Turning movement 06 - SMV unattended vehicle
##                      0                  0
##          07 - SMV other            99 - Other
##                      0                  0
##          02 - Non-fatal injury
##                      2748
##
## $rights
##          01 - Approaching          02 - Angle
##                      66                 453
##          03 - Rear end            04 - Sideswipe
##                      918                149
##          05 - Turning movement 06 - SMV unattended vehicle
##                      431                 18
##          07 - SMV other            99 - Other
##                      678                35
##          02 - Non-fatal injury
##                      0
##
## $sizey
##          01 - Approaching          02 - Angle
##                      66                 453
##          03 - Rear end            04 - Sideswipe
##                      918                149
##          05 - Turning movement 06 - SMV unattended vehicle
##                      431                 18
##          07 - SMV other            99 - Other
##                      678                35
##          02 - Non-fatal injury
##                      2748
##
## [1] "calculating positions"
## [1] 274.8
##                                     ID x
## 01 - Approaching          01 - Approaching 1
## 02 - Angle                02 - Angle 1
## 03 - Rear end              03 - Rear end 1
## 04 - Sideswipe             04 - Sideswipe 1
## 05 - Turning movement     05 - Turning movement 1
## 06 - SMV unattended vehicle 06 - SMV unattended vehicle 1
## 07 - SMV other              07 - SMV other 1
## 99 - Other                 99 - Other 1
## 02 - Non-fatal injury      02 - Non-fatal injury 2
## [1] "done"
## [1] "drawing edges"

```

```
## [1] "drawing nodes"
```

This river plot clearly indicates that SMV other, rear end, turning movement and angle collision types are the most frequent impact types for collisions with non fatal injuries.

```
#### Turning Movement with Environment and Road_Surface
```

The next river plot focuses on the turning movement collisions, since in previous plots, we have seen that they sometimes lead to fatal and non-fatal injuries. Therefore, we may want to see if there is some type of environment or road surface that is more frequently observed with this type of impact.

```
appr_only <- df %>% filter(Impact_type == "05 - Turning movement")
v2 <- as.factor(appr_only$Environment)
v1 <- as.factor(appr_only$Road_Surface)

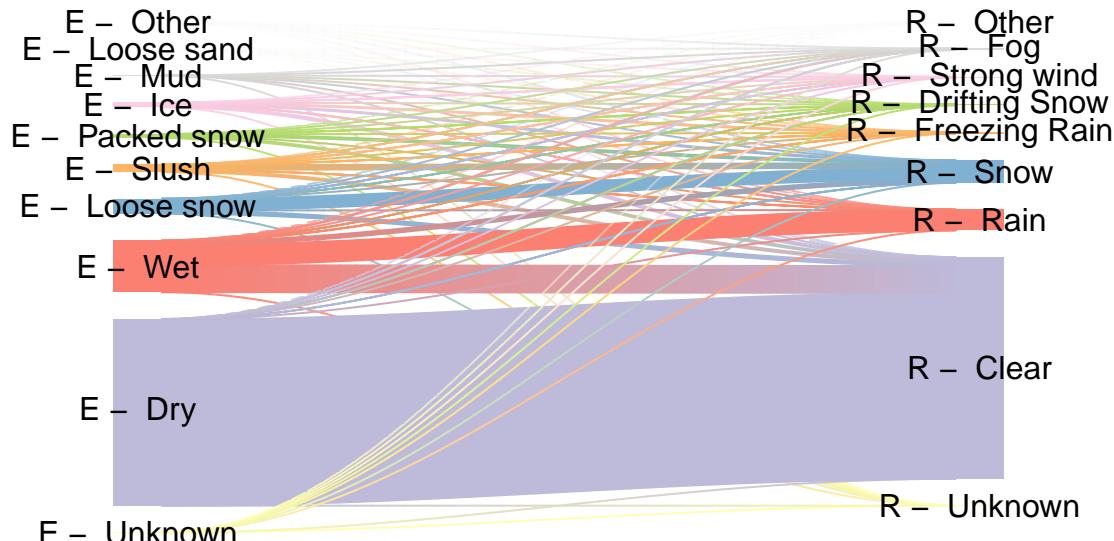
from_label <- str_extract(levels(v1), "[a-zA-Z]+( ?)([a-zA-Z]+)")
to_label <- str_extract(levels(v2), "[a-zA-Z]+( ?)([a-zA-Z]+)")

levels(v1) <- paste("E - ", from_label)
levels(v2) <- paste("R - ", to_label)

edges <- as.data.frame(table(v1, v2))
colnames(edges) <- c("N1", "N2", "Value")
nodes <- data.frame(
  ID = c(levels(v1), levels(v2)),
  x = c(rep(1, length(levels(v1))), rep(2, length(levels(v2))))
)

palette = paste0(brewer.pal(9, "Set3"))
styles <- lapply(
  c(1:10, 1:9),
  function(n) {
    list(col = palette[n+1])
  })

names(styles) = nodes$ID
r <- makeRiver(nodes, edges, node_styles = styles)
plot(r, srt=0, lty=0, node_margin = 0.1, plot_area = 1, fix.pdf = T)
```



```

## -----
## Default style:
## -----
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
## -----
## [1] "checking edges"
## [1] "adding mid points"
## Updated styles:
## $`E - Unknown` 
## $col
## [1] "#FFFFB3"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"

```

```

##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry`
## $col
## [1] "#BEBADA"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet`
## $col
## [1] "#FB8072"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##

```

```

## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose snow`
## $col
## [1] "#80B1D3"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Slush`
## $col
## [1] "#FDB462"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex

```

```

## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow`
## $col
## [1] "#B3DE69"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice`
## $col
## [1] "#FCCDE5"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"

```

```

##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Mud`
## $col
## [1] "#D9D9D9"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose sand`
## $col
## [1] NA
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##

```

```

## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other`$col
## [1] NA
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Unknown`$col
## [1] "#FFFFB3"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty

```

```

## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Clear`$col
## [1] "#BEBADA"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Rain`$col
## [1] "#FB8072"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"

```

```

##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(),"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Snow`
## $col
## [1] "#80B1D3"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(),"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Freezing Rain`
## $col
## [1] "#FDB462"
##
## $srt
## [1] 0
##

```

```

## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Drifting Snow` 
## $col
## [1] "#B3DE69"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Strong wind` 
## $col
## [1] "#FCCDE5"
##
## $srt

```

```

## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Fog`$col
## [1] "#D9D9D9"
##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`R - Other`$col
## [1] NA

```

```

##
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Unknown->R - Unknown` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry->R - Unknown` 

```

```

## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet->R - Unknown` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"

```

```

##  

## $`E -  Loose snow->R -  Unknown`  

## $srt  

## [1] 0  

##  

## $nodestyle  

## [1] "regular"  

##  

## $col  

## [1] "grey"  

##  

## $lty  

## [1] 1  

##  

## $textcol  

## [1] "black"  

##  

## $textcex  

## [1] 1  

##  

## $edgecol  

## [1] "gradient"  

##  

## $edgestyle  

## [1] "sin"  

##  

## attr(,"class")  

## [1] "list"           "riverplotStyle"  

##  

## $`E -  Slush->R -  Unknown`  

## $srt  

## [1] 0  

##  

## $nodestyle  

## [1] "regular"  

##  

## $col  

## [1] "grey"  

##  

## $lty  

## [1] 1  

##  

## $textcol  

## [1] "black"  

##  

## $textcex  

## [1] 1  

##  

## $edgecol  

## [1] "gradient"  

##  

## $edgestyle  

## [1] "sin"  

##
```

```

## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow->R - Unknown` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice->R - Unknown` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle

```

```

## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Mud->R - Unknown` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose sand->R - Unknown` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"

```

```

##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other->R - Unknown` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Unknown->R - Clear` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##

```

```

## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex

```

```

## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose snow->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Slush->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"

```

```

##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##

```

```

## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Mud->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose sand->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty

```

```

## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other->R - Clear`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Unknown->R - Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"

```

```

##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(),"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry->R - Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(),"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet->R - Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##

```

```

## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose snow->R - Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Slush->R - Rain`
## $srt
## [1] 0
##
## $nodestyle

```

```

## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow->R - Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice->R - Rain`
## $srt
## [1] 0

```

```

##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Mud->R - Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose sand->R - Rain`
```

```

## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other->R - Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"

```

```

##  

## $`E - Unknown->R - Snow`  

## $srt  

## [1] 0  

##  

## $nodestyle  

## [1] "regular"  

##  

## $col  

## [1] "grey"  

##  

## $lty  

## [1] 1  

##  

## $textcol  

## [1] "black"  

##  

## $textcex  

## [1] 1  

##  

## $edgecol  

## [1] "gradient"  

##  

## $edgestyle  

## [1] "sin"  

##  

## attr(,"class")  

## [1] "list"           "riverplotStyle"  

##  

## $`E - Dry->R - Snow`  

## $srt  

## [1] 0  

##  

## $nodestyle  

## [1] "regular"  

##  

## $col  

## [1] "grey"  

##  

## $lty  

## [1] 1  

##  

## $textcol  

## [1] "black"  

##  

## $textcex  

## [1] 1  

##  

## $edgecol  

## [1] "gradient"  

##  

## $edgestyle  

## [1] "sin"  

##
```

```

## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet->R - Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose snow->R - Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle

```

```

## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Slush->R - Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow->R - Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"

```

```

##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice->R - Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Mud->R - Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##

```

```

## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose sand->R - Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other->R - Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex

```

```

## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Unknown->R - Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry->R - Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"

```

```

##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E -  Wet->R -  Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E -  Loose snow->R -  Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##

```

```

## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Slush->R - Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow->R - Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty

```

```

## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice->R - Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Mud->R - Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"

```

```

##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(),"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose sand->R - Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(),"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other->R - Freezing Rain`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##

```

```

## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Unknown->R - Drifting Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry->R - Drifting Snow`
## $srt
## [1] 0
##
## $nodestyle

```

```

## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet->R - Drifting Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose snow->R - Drifting Snow`
## $srt
## [1] 0

```

```

##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Slush->R - Drifting Snow` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow->R - Drifting Snow` 

```

```

## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice->R - Drifting Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"

```

```

##  

## $`E - Mud->R - Drifting Snow`  

## $srt  

## [1] 0  

##  

## $nodestyle  

## [1] "regular"  

##  

## $col  

## [1] "grey"  

##  

## $lty  

## [1] 1  

##  

## $textcol  

## [1] "black"  

##  

## $textcex  

## [1] 1  

##  

## $edgecol  

## [1] "gradient"  

##  

## $edgestyle  

## [1] "sin"  

##  

## attr(,"class")  

## [1] "list"           "riverplotStyle"  

##  

## $`E - Loose sand->R - Drifting Snow`  

## $srt  

## [1] 0  

##  

## $nodestyle  

## [1] "regular"  

##  

## $col  

## [1] "grey"  

##  

## $lty  

## [1] 1  

##  

## $textcol  

## [1] "black"  

##  

## $textcex  

## [1] 1  

##  

## $edgecol  

## [1] "gradient"  

##  

## $edgestyle  

## [1] "sin"  

##
```

```

## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other->R - Drifting Snow`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Unknown->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle

```

```

## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"

```

```

##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose snow->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Slush->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##

```

```

## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex

```

```

## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Mud->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose sand->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"

```

```

##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other->R - Strong wind`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Unknown->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##

```

```

## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty

```

```

## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose snow->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Slush->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"

```

```

##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(),"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Packed snow->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(),"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Ice->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##

```

```

## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Mud->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose sand->R - Fog`
## $srt
## [1] 0
##
## $nodestyle

```

```

## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Other->R - Fog`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Unknown->R - Other`
## $srt
## [1] 0

```

```

##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Dry->R - Other` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Wet->R - Other` 

```

```

## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E - Loose snow->R - Other`
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"

```

```

##  

## $`E - Slush->R - Other`  

## $srt  

## [1] 0  

##  

## $nodestyle  

## [1] "regular"  

##  

## $col  

## [1] "grey"  

##  

## $lty  

## [1] 1  

##  

## $textcol  

## [1] "black"  

##  

## $textcex  

## [1] 1  

##  

## $edgecol  

## [1] "gradient"  

##  

## $edgestyle  

## [1] "sin"  

##  

## attr(,"class")  

## [1] "list"           "riverplotStyle"  

##  

## $`E - Packed snow->R - Other`  

## $srt  

## [1] 0  

##  

## $nodestyle  

## [1] "regular"  

##  

## $col  

## [1] "grey"  

##  

## $lty  

## [1] 1  

##  

## $textcol  

## [1] "black"  

##  

## $textcex  

## [1] 1  

##  

## $edgecol  

## [1] "gradient"  

##  

## $edgestyle  

## [1] "sin"  

##
```

```

## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E -  Ice->R -  Other` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E -  Mud->R -  Other` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle

```

```

## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E -  Loose sand->R -  Other` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"
##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## $`E -  Other->R -  Other` 
## $srt
## [1] 0
##
## $nodestyle
## [1] "regular"
##
## $col
## [1] "grey"
##
## $lty
## [1] 1
##
## $textcol
## [1] "black"
##
## $textcex
## [1] 1
##
## $edgecol
## [1] "gradient"

```

```

##
## $edgestyle
## [1] "sin"
##
## attr(,"class")
## [1] "list"           "riverplotStyle"
##
## -----
## [1] "calculating sizes"
## $lefts
##      E - Unknown          E - Dry          E - Wet
##      0                      0                  0
##      E - Loose snow        E - Slush        E - Packed snow
##      0                      0                  0
##      E - Ice               E - Mud          E - Loose sand
##      0                      0                  0
##      E - Other             R - Unknown      R - Clear
##      0                      0                  1217
##      R - Rain              R - Snow         R - Freezing Rain
##      113                   123                  8
## R - Drifting Snow       R - Strong wind    R - Fog
##                         4                     4                  7
##      R - Other
##      0
##
## $rights
##      E - Unknown          E - Dry          E - Wet
##      0                      1025                 284
##      E - Loose snow        E - Slush        E - Packed snow
##      81                     40                  20
##      E - Ice               E - Mud          E - Loose sand
##      26                     0                  0
##      E - Other             R - Unknown      R - Clear
##      0                      0                  0
##      R - Rain              R - Snow         R - Freezing Rain
##      0                     0                  0
## R - Drifting Snow       R - Strong wind    R - Fog
##                         0                     0
##      R - Other
##      0
##
## $sizey
##      E - Unknown          E - Dry          E - Wet
##      0                      1025                 284
##      E - Loose snow        E - Slush        E - Packed snow
##      81                     40                  20
##      E - Ice               E - Mud          E - Loose sand
##      26                     0                  0
##      E - Other             R - Unknown      R - Clear
##      0                      0                  1217
##      R - Rain              R - Snow         R - Freezing Rain
##      113                   123                  8
## R - Drifting Snow       R - Strong wind    R - Fog
##                         4                     4                  7

```

```

##          R -  Other
##                  0
##
## [1] "calculating positions"
## [1] 147.6
##           ID x
## E - Unknown      E - Unknown 1
## E - Dry          E - Dry 1
## E - Wet          E - Wet 1
## E - Loose snow   E - Loose snow 1
## E - Slush         E - Slush 1
## E - Packed snow  E - Packed snow 1
## E - Ice           E - Ice 1
## E - Mud           E - Mud 1
## E - Loose sand   E - Loose sand 1
## E - Other         E - Other 1
## R - Unknown       R - Unknown 2
## R - Clear         R - Clear 2
## R - Rain          R - Rain 2
## R - Snow          R - Snow 2
## R - Freezing Rain R - Freezing Rain 2
## R - Drifting Snow R - Drifting Snow 2
## R - Strong wind   R - Strong wind 2
## R - Fog            R - Fog 2
## R - Other          R - Other 2
## [1] "done"
## [1] "drawing edges"
## [1] "drawing nodes"

```

The above plot indicates that a large portion of turning movement collisions happen when it is dry and clear. Although, it is interesting to see that a wet road and rain are more frequent conditions for this type of impact. This may be due to the fact that drivers are generally more cautious in snowy conditions, and maybe underestimate the effect of a wet road and rain.

### ###Conclusion

Finally, after exploring the dataset, we are better equipped to determine if this dataset is trustworthy. In general, this dataset seems to be trustworthy. There doesn't seem to be too many missing values, and the few missing values seem to be missing at random. In general, these missing values do not seem to have a big impact on the dataset, therefore further analysis is probably possible, if imputation methods are used to fill in the missing values, or if they are disregarded completely. Also, there are some data entry errors, as some values were not entered in the same format as the rest (ex. a few X and Y coordinates, a few time values, etc.). However, datasets are rarely perfect, and this one is no exception, but the few problems with it do not warrant it unusable.

Depending on the needs and the questions that we want to answer, this data can be useful. For instance, since we have the geographic locations of all the collisions, we could compare different areas of Ottawa to see if certain roads/regions/neighbourhoods seem to be particularly more dangerous in terms of collisions. We also determined that roads with no traffic control have a variety of different types of impact. Further investigation into this issue may prove useful in the way that we could maybe determine areas that would benefit from having some type of traffic control (ex. adding some stop signs, lights, etc. on some specific roads that don't already have these traffic controls). This dataset is also useful in gathering information related to fatal collisions, and maybe the analysis of this data could lead to new discoveries or implementations that will reduce the number of fatal collisions in the years to come.

Thus, this dataset is a decent one, from which we can gather some information that could possibly save lives

or at least limit the number of accidents that could be avoided with some small changes to road maintenance or traffic control.