Assignment4_Amogh_21236437

Amogh M. Agnihotri Student ID - 21236437 (MSc-AI)

01/04/2022

Visualizing Traffic at the Junction Through the Day

In this assignment we will visualize various distribution and volumes of traffic going through the junction near university. We will use various visualization techniques to show the asked visual along with preprocessing the data ccordingly. We have been provided with the data in long format for a dat from 7 AM in the morning till 7 PM in the evening with number of veicles coming in and going out of the junction through various exits. We will see how they are distributed with their types and time of the day.

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(RColorBrewer)
library(ggforce)
library(hms)
library(colorspace)
library(scales)
library(treemapify)
library(ggalluvial)
library(treemap)
library(sf)
library(plotly)
library(reshape)
```

Initial Data Pre-processing

Initially we will take the given data which is in long format to in a variable and save it. We will create a datafram with that data in order to use it for further questions. We will also convert the count column to integer type from charachter type to perform further operation.

For further questions we will preprocess the data as required for each question.

```
#Taking input of given long file in a variable
file <- "E:/College Wiki/Data Visualisation/Assignment 4/Junction Turning Counts 2016 Outside
DSI_LONG_FORMAT.csv"

#Reading the long file and storing it in a dataframe
data <- st_read(file, quiet = TRUE)

#Converting the count column to numeric type from charachter
data_m <- transform(data, count = as.numeric(count))
```

Question 1 - visualisation of the distributions of vehicles - with Multiple Strip plots

Explaination

Here the ask is to visualize the distribution of traffic at junction every 15 minutes. We don’t need to show where the vehicles are comming from but just the amount of vehicles as per their types passing throught the vehicle at 15 minutes interval throughot the day. We will be plotting the Multiple strip plots to visualize this as this will show distribution of all the categories in a better way.

Design Decisions

The distribution can be shown with various ways such as line graphs, density plots, violin plots, box plots, strip plots and etc. The density plot will get too corwded to see all the 10 classes we need to show and similar is the case with the line plots. THe violin plots and box plots can show the visualization of distribution, but in my opinion they are specifically used to show outliers and distribution among the quarte of the quantity mentioned on x axis. Violin plots along with strip plots are also used to show outliers. Here using violin plots with strip plot can be a solution but it will be too crowded and we don’t need to show anything regarding the outliers or the distribution of quantity in quarters. The Multiple strip plot approach fits to this sense perfectly where you can easily see the distribution of vehicles along 15 minutes interval for all the classes pretty easily. We dont want to show the exact number of vehicles, but the overall idea of the distribution of the count through the day which is effectively conveyed by plotted graph. We plot time on X axis with continuous scale giving 1 hour interval so that the x axis is not too crowded and also the 15 minutes intervals can be judged fairly easily. On Y axis, we plot the names of vehicle types with proper name and not the abbreviations as asked. With the jitter among the geom points we can easily see and judge the distribution of vehicles. The jitter helps in showing the coinciding points which help us see the poper proportion of the count.

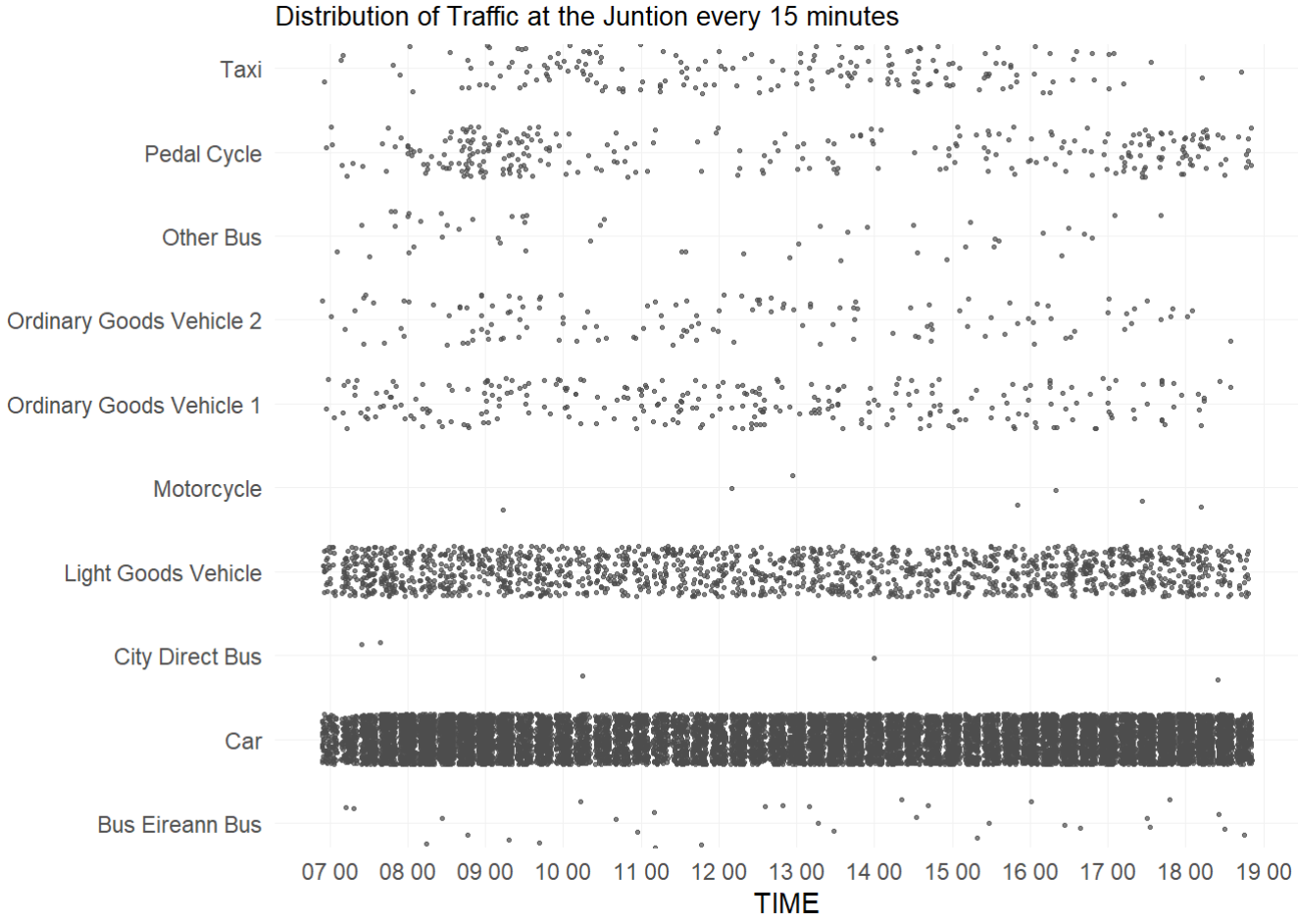
Data Pre-processing

In data preprocessing for this task, we will intially group the dataframe with TIME and vehicle columns and sum the count. We will convert the TIME column to POSIXct format so that we can use it in scale in the graph at X-axis. Further we will change the names of the vehicles from their abbreviations to the proper names. We will untable data creating the entries equivalent to number of count so that in our strip plot we can show the correct volume of distribution. For example, if Car has 10 count at 7 AM, it creates 10 entries of Car so that can be plotted.

```
# Data Pre-processing for question 1

#Creating a dataframe from original data with grouping with Time and Vehicle as we need to sh
ow all the traffic at
#this junction
q1_data <- data_m %>%
  group_by(TIME, vehicle) %>%
  dplyr::summarize(group_count = sum(count)) %>%
  as.data.frame()

#Converting TIME column to POSIXct format for further processing
q1_data$TIME <- as.POSIXct(q1_data$TIME)

#Changing the abbreviations of vehicle types to meaningful names
q1_data <- q1_data %>%
  mutate(
    vehicle = case_when(
      vehicle == "PCL" ~ "Pedal Cycle",
      vehicle == "MCL" ~ "Motorcycle",
      vehicle == "CAR" ~ "Car",
      vehicle == "TAXI" ~ "Taxi",
      vehicle == "LGV" ~ "Light Goods Vehicle",
      vehicle == "OGV1" ~ "Ordinary Goods Vehicle 1",
      vehicle == "OGV2" ~ "Ordinary Goods Vehicle 2",
      vehicle == "CDB" ~ "City Direct Bus",
      vehicle == "BEB" ~ "Bus Eireann Bus",
      vehicle == "OB" ~ "Other Bus",
      )
  )

#Un-tabling the data points in order to create instances for every count of any vehicle
data_points <- untable(q1_data, q1_data$group_count)
```

Plotting the Multiple Strip Plot

```
#Plotting the Multiple strip plots
ggplot(data_points) +
  geom_point(aes(x = TIME , y= vehicle ), color= "grey30",
             size = 0.5, alpha = 0.7,
             position = position_jitter(height = 0.3),  show.legend=F) +
  scale_y_discrete(expand = c(0,0))+
  #Defining descrete datetime on x axis
  scale_x_datetime(date_breaks = "1 hour", labels = date_format("%H %M")) +

  ggtitle("Distribution of Traffic at the Juntion every 15 minutes") +
  theme_minimal() +
  theme (
    panel.grid.major.y =element_line(colour = "gray95", size =0.25),
    panel.grid.major.x =element_line(colour = "gray95", size =0.25),
    panel.grid.minor.y =element_line(colour = "gray95", size =0.15),
    axis.line = element_blank(),
    axis.title.y = element_blank(),
    plot.title = element_text(size=10),
    legend.background = element_blank(),
    legend.key        = element_blank(),
    panel.background  = element_blank(),
    panel.border      = element_blank(),
    strip.background  = element_blank(),
    plot.background   = element_blank(),
    panel.grid= element_blank())
```



Question 2 - Visualising the proportion of traffic comming from D - with Parallel Sets plots

Explaination

In this task, we have to show the proportion of the traffi coming from exit D and going into exit A, B and C at particular time. We will be clubbing the time intervals in a category such as Early Morning, Late Morning, etc. We don't need to care abount the type of the vehicle, we just need to show how many cars are coming in from 'D' and going in to 'A', 'B' or 'C' and at what time category.

Design Decison

The task asks us to visualize a flow of quantity from one to different forms. The Parallel set plot does this job with perfect visuals. The other plots such as stacked bar plots or facetted bar plots can also be used in this case but they will be too crowded and not easily interpretable to eyes for judging the proportion; besides we dont want to convey the exact count but the proportion of the traffic. With Parallel sets plot we define the first axes as 'D' which is the origin of coming traffic then we plot the middle axes which a=say where it turns, i.e. DA means going to exit 'A' and at last we plot the axes with time category to show what proportion of the traffic turning to a particular direction is present wat which time. We plot the names of axes on the X-axis and count of the vehicle on the Y axis, which viewer can use to guage the approximate count and get an idea about the movement of traffic. The plot makes the seamless transition from one axes to other and makes it easy for viewrer to judge the movement of the vehicle over the other plots.

Data Pre-processing

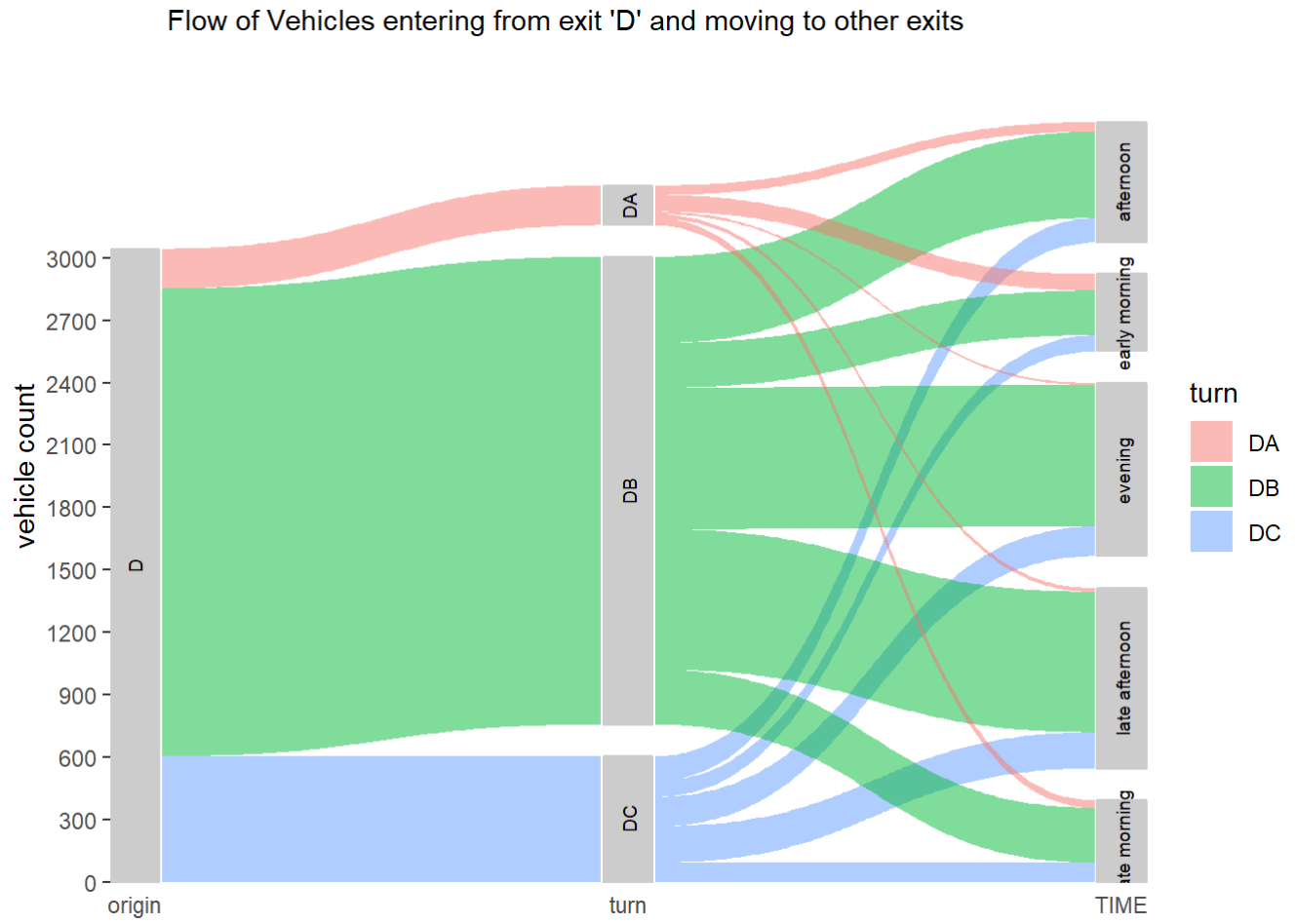
Initially, we will filter the data into a new data frame which only contains the vehicles entering from 'D'. We will create a new column as origin, and put all the values as 'D'. As we need this to add in our leftmost axis. We will then group the dataframe by columns TIME and turn as we dont need vehicles descreation here and sum the count. We will then change the values in TIME column to proper categories as given and factor them by the appropriate order, that is the time of the day. We will then create a new dataframe for parallel sets using gather_set_data function which will give us the relationships between the axes we need to plot. Finally, we will have out axes in 'x' column with newly formed DF, we will factor it according to the flow we are trying to visualize.

```
#Filtering data with vehivles turning from 'D' to other exits
q2_data <- filter(data_m, turn == "DA" | turn == "DB" | turn == "DC")
#Grouping data based on Time and turn as we need to show distrbution of traffic coming from
#'D' and going to other directions
q2_data <- q2_data %>%
group_by(TIME, turn) %>%
  dplyr::summarize(group_count = sum(count)) %>%
  as.data.frame()

#Restructuring TIME column to given categories, then factoring them and grouping them
q2_data <- q2_data %>%
  mutate(
    TIME = case_when(
      TIME == "2016-11-23 07:00:00" ~ "early morning",
      TIME == "2016-11-23 07:15:00" ~ "early morning",
      TIME == "2016-11-23 07:30:00" ~ "early morning",
      TIME == "2016-11-23 07:45:00" ~ "early morning",
      TIME == "2016-11-23 08:00:00" ~ "early morning",
      TIME == "2016-11-23 08:15:00" ~ "early morning",
      TIME == "2016-11-23 08:30:00" ~ "early morning",
      TIME == "2016-11-23 08:45:00" ~ "early morning",
      TIME == "2016-11-23 09:00:00" ~ "early morning",
      TIME == "2016-11-23 09:15:00" ~ "early morning",
      TIME == "2016-11-23 09:30:00" ~ "early morning",
      TIME == "2016-11-23 09:45:00" ~ "late morning",
      TIME == "2016-11-23 10:00:00" ~ "late morning",
      TIME == "2016-11-23 10:15:00" ~ "late morning",
      TIME == "2016-11-23 10:30:00" ~ "late morning",
      TIME == "2016-11-23 10:45:00" ~ "late morning",
      TIME == "2016-11-23 11:00:00" ~ "late morning",
      TIME == "2016-11-23 11:15:00" ~ "late morning",
      TIME == "2016-11-23 11:30:00" ~ "late morning",
      TIME == "2016-11-23 11:45:00" ~ "late morning",
      TIME == "2016-11-23 12:00:00" ~ "late morning",
      TIME == "2016-11-23 12:15:00" ~ "afternoon",
      TIME == "2016-11-23 12:30:00" ~ "afternoon",
      TIME == "2016-11-23 12:45:00" ~ "afternoon",
      TIME == "2016-11-23 13:00:00" ~ "afternoon",
      TIME == "2016-11-23 13:15:00" ~ "afternoon",
      TIME == "2016-11-23 13:30:00" ~ "afternoon",
      TIME == "2016-11-23 13:45:00" ~ "afternoon",
      TIME == "2016-11-23 14:00:00" ~ "afternoon",
      TIME == "2016-11-23 14:15:00" ~ "afternoon",
      TIME == "2016-11-23 14:30:00" ~ "afternoon",
      TIME == "2016-11-23 14:45:00" ~ "late afternoon",
      TIME == "2016-11-23 15:00:00" ~ "late afternoon",
      TIME == "2016-11-23 15:15:00" ~ "late afternoon",
      TIME == "2016-11-23 15:30:00" ~ "late afternoon",
      TIME == "2016-11-23 15:45:00" ~ "late afternoon",
      TIME == "2016-11-23 16:00:00" ~ "late afternoon",
      TIME == "2016-11-23 16:15:00" ~ "late afternoon",
      TIME == "2016-11-23 16:30:00" ~ "late afternoon",
      TIME == "2016-11-23 16:45:00" ~ "late afternoon",
      TIME == "2016-11-23 17:00:00" ~ "late afternoon",
      TIME == "2016-11-23 17:15:00" ~ "evening",
```

```
      TIME == "2016-11-23 17:30:00" ~ "evening",
      TIME == "2016-11-23 17:45:00" ~ "evening",
      TIME == "2016-11-23 18:00:00" ~ "evening",
      TIME == "2016-11-23 18:15:00" ~ "evening",
      TIME == "2016-11-23 18:30:00" ~ "evening",
      TIME == "2016-11-23 18:45:00" ~ "evening",
      TIME == "2016-11-23 19:00:00" ~ "evening"

    )
  ) %>%
  mutate(TIME = factor(TIME, levels = c("early morning", "late morning", "afternoon", "late a
fternoon", "evening"))) %>%
  group_by(TIME, turn) %>%
  dplyr::summarize(group_count = sum(group_count)) %>%
  as.data.frame()

#Adding a coulumn as Origin of turn and adding all values as 'D'
q2_data$origin <- "D"

#Creating a dataframe in the format for Parallel set
q2_data_ps <- gather_set_data(q2_data, c(4,2,1))

#Factoring the X column
q2_data_ps$x <- factor(q2_data_ps$x, levels = c("origin","turn", "TIME"))
```

Plotting the Parallel sets plot

```
ggplot(q2_data_ps, aes(x, id = id, split = y, value = group_count)) +
  #Defining parallel sets
  geom_parallel_sets(aes(fill= turn), alpha = 0.5, axis.width = 0.11) +
  #Defining Parallel set axes
  geom_parallel_sets_axes(axis.width = 0.1, fill = "grey80", color = "grey80") +
  #Defining parallel set labels
  geom_parallel_sets_labels(
    color = 'black',
    size = 7/.pt,
    angle = 90
  ) +
  #Defining discrete X scale
  scale_x_discrete(
    name = NULL,
    expand = c(0.0, 0.0)
  ) +
  #defining continuous y scale
  scale_y_continuous(
    #breaks = NULL,
    expand = c(0, 0),
    limits = c(0, 4000),
    breaks = seq(0,3.0e3 , by =3.0e2),
    labels = seq(0,3000, by =300),
    name = "vehicle count"
  )+

  theme(
    axis.line = element_blank(),
    axis.ticks.x      = element_blank(),
    legend.background = element_blank(),
    legend.key        = element_blank(),
    panel.background  = element_blank(),
    panel.border      = element_blank(),
    strip.background  = element_blank(),
    plot.background   = element_blank(),
    panel.grid= element_blank(),
    plot.title = element_text(size = 11, hjust = 0.24)

  )+
  #Plotting
  ggtitle("Flow of Vehicles entering from exit 'D' and moving to other exits")
```



Question 3 - Visualising the volume of vehicles per vehicle type at each timestamp- with Heat Map

Explaination

In this task, we are asked to show a visualization which depicts the volume of vehicle at the junction for all the given time stamps, for all the vehicle types. We need to show which vehicles are present more at what time. The visualization chosen for this task is heatmap as it shows clear changes in volumes with help of continuous color scale.

Design Decision

The heatmap is effective way of showing the change in volume over the time period with the help of continuous color scale. WE can easily see the changes and refer to legend to know exactly what amount of vehicles were passing at the given time for the whole day. It is really easy to see that Cars and LGV are the types of vehicles having large volume throughout the day and others are pretty low in comparison. The advantage of heat map is that we can choose a custom color scale as per our count or values of volume in the data frame and it communicates the message easily and effectively as compared to other histogram methods such as Density plots or bar graphs will be too crowded to show such visuals. Another consideration for the task was again a box plot but it doesn't really convey the message of showing variation in volume.

Data Pre-Processing

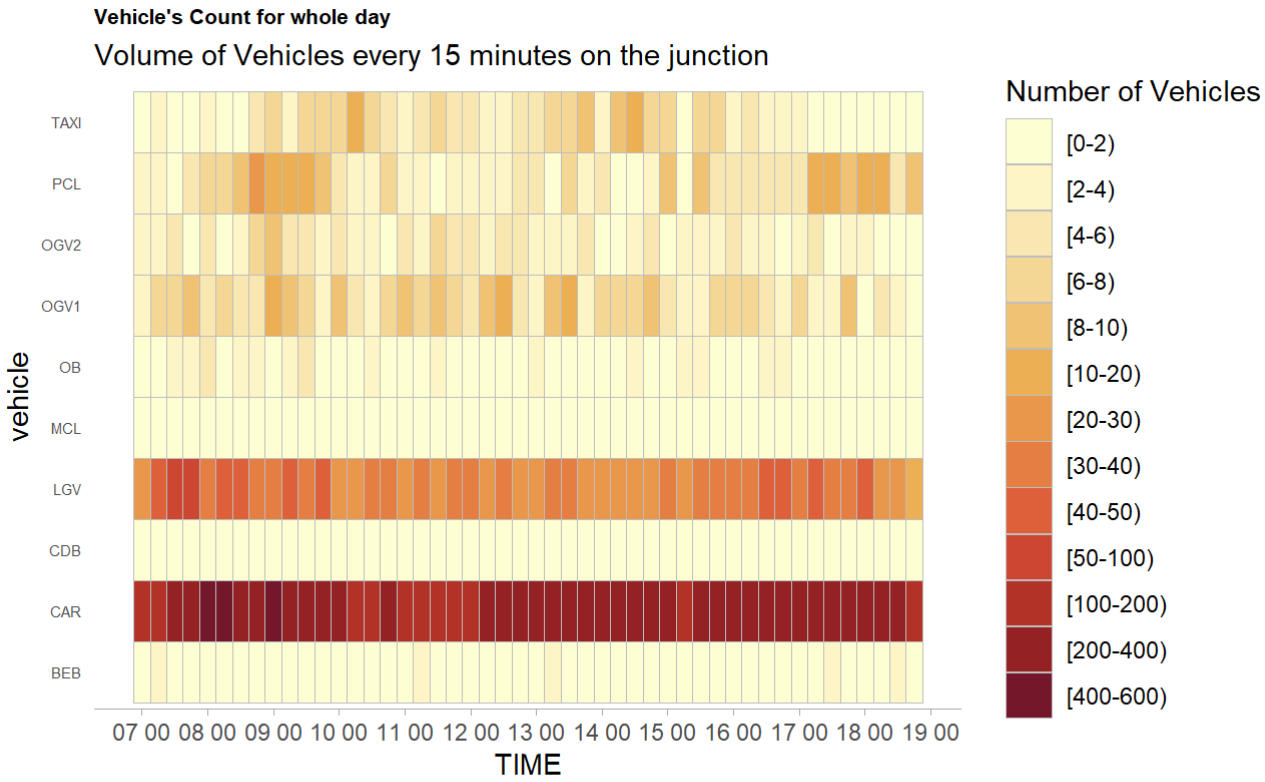
For this we will initially group the data by TIME and vehicle column as we have nothing to do with turns, we will sum the count. We will get the maximum and minimum of the count and store it in order to create a scale. The we will create a desecrate scale of colors based on the observation in the count column. We will also create a

legend with labels of our discrete scale.

```
#Groupping the data with TIME and Vehicle in order show volume of vehicles at junction at eac
h timestamp given
q3_data <- data_m %>%
  group_by(TIME, vehicle) %>%
  dplyr::summarize(group_count = sum(count)) %>%
  as.data.frame()


q3_data$TIME <- as.POSIXct(q3_data$TIME)

#Storing minimum and Maximum for scaling the heatmap colors
scale_minimum <- min(q3_data$group_count)
scale_maximum<- max(q3_data$group_count)

#Adding breaks as appropriate
breakss<-c(0,2,4,6,8,10,20,30,40,50,100,200,400,600)


#Adding intervals to DF for Splitting the count
q3_data$descrete_scale <- cut(q3_data$group_count,
                                        breaks = breakss, right = FALSE
                                        )


# Selecting YlOrRd color pallet and number of colors equal to intervals
nlevels <- nlevels(q3_data$descrete_scale )
pal <- hcl.colors(nlevels, "YlOrRd", rev = TRUE)
pal_desat<-desaturate(pal,amount = 0.2)


labs <- breakss/1
labs_plot <- paste0("[", labs[1:nlevels],"-", labs[1:nlevels+1], ")")
```

PLotting the Heatmap

```
ggplot(q3_data, aes(x=TIME, y=vehicle, fill = descrete_scale)) +
  geom_tile(
          color = "gray") +

  labs(subtitle = "Volume of Vehicles every 15 minutes on the junction")+
  # Custom palette
  scale_fill_manual(values = pal_desat,
                    drop = FALSE,
                    na.value = "grey80",
                    label = labs_plot,
                    # Legend
                    guide = guide_legend(direction = "vertical",
                                         ncol = 1,
                                         label.position = "right",
                                         title = "Number of Vehicles"
                    )) +
  scale_x_datetime(date_breaks = "1 hours", labels = date_format("%H %M")) +

                     ggtitle("Vehicle's Count for whole day") +

                     coord_cartesian(clip = 'off') +

                     theme_minimal() +
  theme(axis.text.y = element_text(size=5.6),
        axis.ticks.x = element_line(size=0.3, colour = "darkgrey"),
        axis.line.x = element_line(size=0.3, colour = "darkgrey"),
        axis.ticks.y = element_blank(),
        axis.line.y = element_blank(),
        panel.background = element_blank(),
        panel.grid = element_blank(),
        plot.margin = unit(c(0.5, 0.5, 2, 0.5), "cm"),
        plot.title = element_text(size=8, face="bold"))
```



Question 4 - Visualising the proportion of Categories and SubCategories of vehicles over the 12 hour period - with Tree Map

Explaination

In this task, we want visualize the proportion of Categories and subcategories of vehicles throughout the day. We will be grouping vehicles into categories as mentioned in the assignment and plot it in a way that the visualization demonstrates the category to subcategory visualization in order to show the proportion of vehicles throughout the day. As the time is for all day, the notion of plotting time disappears. WE dont want to make the map crowded with unnecessary things. Here, the choice of visualization is Tree Map as it can effectively show the category and subcategory relation along with the appropriate coloring scale for the volume of the count.

Design Decision

The best visualization to demonstrate the Category and subcategory visualization is tree map. It can effectively show the relation between two and the color scaling can be used to show the effective volume of the category or subcategory. As we have clumped up the vehicles into a categories, we will plot tiles of categories inside which there are tiles of subcategories. I dont think any other plot can achieve the similar notion with the exception of strip plot, but here as we needed to show the proportion over whole day we can use the continuous color scales in tree map, which is differentiable by categories and volume can be judged by either the fading of color in sub categories or the exact number given in the tile.

Data Pre-Processing

While pre processing, we created a new column called category and for every vehicle we put them in appropriate category to which they belong. We will put them in a group, and factor them in with category they are grouped in. We will create a new DF which creates areas and color combination.

```
q4_data <- data_m
#Creating a new column as 'category'
q4_data$category
```

```
## NULL
```

```
#Putting vehicles in defined categories and grouping them after factoring the categories
q4_data <- q4_data %>%
  mutate(
    category = case_when(
      vehicle == "CAR"| vehicle == "TAXI"  ~ "CARS",
      vehicle == "PCL" | vehicle == "MCL" ~ "TWO-WHEEL VEHICLES",
      vehicle == "LGV" | vehicle == "OGV1" | vehicle == "OGV2" ~ "GOODS VEHICLES",
      vehicle == "CDB" | vehicle == "BEB" | vehicle == "OB" ~ "BUSES AND PUBLIC TRANSPORT"

    )
  )%>%
  mutate(category = factor(category, levels = c("CARS", "TWO-WHEEL VEHICLES", "afternoon", "G
OODS VEHICLES", "BUSES AND PUBLIC TRANSPORT"))) %>%
  group_by(vehicle, category) %>%
  dplyr::summarize(count = sum(count)) %>%
  as.data.frame()

#Saving number of categories
n <- length(unique(q4_data$category))

#Creating a new data-frame with mapping the color and max count for defining tile area
q4_data_df2 <- q4_data %>%
  mutate(index = as.numeric(factor(category))- 1) %>%
  group_by(index) %>%
  mutate(
    max_count = max(count),
    colour = gradient_n_pal(
      sequential_hcl(
        6,
        h = 360 * index[1]/n,
        c = c(45, 20),
        l = c(30, 80),
        power = 0.6)
    )(1- (count/max_count))
  )
```

Plotting the tree map

```
ggplot(data = q4_data_df2, aes(area = count, fill=colour, subgroup = category))+
  geom_treemap(colour = "white", size = 1*.pt, alpha = NA)  +
  #Defining text and properties of subgroups
  geom_treemap_text(aes(label = vehicle), colour = "black" , size =10,
                    place = "topleft",fontface = "bold",padding.x = grid::unit(1.5, "mm"),
                    padding.y = grid::unit(1.5, "mm"), grow = FALSE,  min.size = 0) +
  #Defining text of counts
  geom_treemap_text(aes(label = format(count, nsmall=0, big.mark=",",trim=TRUE)), color = "bl
ack", size = 8,
                    place = "topleft", min.size = 3, padding.x = grid::unit(1.5, "mm"),
                    padding.y = grid::unit(15, "points"), grow = FALSE)+
  #Defining the subgroup properties
  geom_treemap_subgroup_border(colour = "white", size =0.5) +

  geom_treemap_subgroup_text(grow = FALSE,  colour = "#FAFAFA", size = 42,
                             place ="bottomleft", fontface = "bold", alpha = 1.0, min.size =
0) +

  scale_fill_identity()+
  coord_cartesian(clip = "off") +
  guides(colour = "none", fill = "none")
```