Cas d'étude Cyclistic Bike Share

Haseeb MOHAMMAD

20/09/2021

Introduction

Cette étude de cas est le projet Capstone du **certificat professionnel Google Data Analytics**. Les **6 étapes de l'analyse des données** sont utilisées pour présenter cette analyse.

Cyclistic Bike Share : Comment Un Partage De Vélo Peut-Il Réussir Rapidement?

ÉTAPE 1 : DEMANDER

1.0 Contexte

Cyclistic est une société de vélos-actions à Chicago. En 2016, Cyclistic a lancé avec succès une offre de partage de vélos. Depuis, le programme est devenu une flotte de 5824 vélos qui sont géolocalisés et verrouillés dans un réseau de 692 stations à travers Chicago. Les vélos peuvent être déverrouillés d'une station et retournés à n'importe quelle autre station du système à tout moment.Le directeur de marketing pense que le succès futur de l'entreprise dépend de la maximisation du nombre d'adhésions annuelles.

1.2 Tâche Opérationnel

L'idée est de concevoir des stratégies de marketing visant à convertir les cyclistes occasionnels en membres annuels. Afin de faire cela, l'équipe d'analystes marketing doit mieux comprendre comment les membres annuels et les cyclistes occasionnels diffèrent, pourquoi les cyclistes occasionnels achèteraient une adhésion, et comment les médias numériques pourraient affecter leurs tactiques de marketing. Nous nous intéresserons à analyser les données historiques sur les déplacements cyclistes pour dégager des tendances.

1.3 Objectifs Opérationnels

1. Comment les membres annuels et les cyclistes occasionnels utilisent-ils les vélos cyclistes différemment?
2. Pourquoi les cyclistes occasionnels achèteraient-ils des adhésions annuelles?
3. Comment les cyclistes peuvent-ils utiliser les médias numériques pour inciter les cyclistes occasionnels à devenir membres?

1.4 Produits Livrables :

1. Un énoncé clair de la tâche opérationnelle
2. Une description de toutes les sources de données utilisées
3. Documentation de tout nettoyage ou manipulation de données
4. Un résumé de votre analyse
5. Visualisation et principales constatations à l'appui
6. Vos trois principales recommandations fondées sur votre analyse

1.5 Principaux Intervenants

1. Lily Moreno : Le directeur du marketing et manager. Moreno est responsable du développement des campagnes et des initiatives visant à promouvoir le programme de partage de vélos, notamment le courriel, les médias sociaux et d'autres canaux.

2. Équipe d'analyse du marketing cycliste : Une équipe d'analystes de données chargés de recueillir, d'analyser et des données qui aident à orienter la stratégie de marketing cycliste.

3. Équipe de direction cycliste : L'équipe de direction axée sur les détails notoirement décidera d'approuver ou non le programme de marketing recommandé.

ÉTAPE 2 : PREPARER

2.1 Renseignements Sur La Source De Données

1. Les données sont accessibles ici : https://divvy-tripdata.s3.amazonaws.com/index.html stockées dans 12 fichiers csv.

2. Les données ont été mises à disposition par Motivate International Inc. sous cette licence : https://www.divvybikes.com/data-license-agreement.

3. Les données disponibles sont entre Avril 2020 et Mars 2021.
4. Les données recueillies comprennent :

- 1. ride_id - un identifiant unique par trajet
- 2. rideable_type : le type de vélo utilisé
- 3. started_at : la date et l'heure de la sortie du vélo
- 4. ended_at : la date et l'heure d'enregistrement du vélo

- 5. start_station_name : le nom de la station au début du trajet
- 6. start_station_id : un identifiant unique pour la station de démarrage
- 7. end_station_name : le nom de la station à la fin du trajet
- 8. end_station_id : un identifiant unique pour la station finale
- 9. Start_lat : la latitude de la station de départ
- 10. start_lng : la longitude de la station de départ
- 11. end_lat : la latitude de la station terminale
- 12. end_lng : la longitude de la station finale
- 13. member_casual : un champ indiquant si la bicyclette a été prise par un membre ou un occasionnel

2.2 Limites De L'Ensemble De Données

Les questions de confidentialité des données qui vous interdisent d'utiliser les informations personnelles identifiables des cyclistes. Ce qui signifie que nous ne serons pas en mesure de relier les achats de laissez-passer aux numéros de carte de crédit pour déterminer si les cyclistes occasionnels vivent dans la zone de service cycliste ou s'ils ont acheté plusieurs laissez-passer individuels.

2.3 Les Données Sont-Elles ROCCC ?

Une bonne source de données est ROCCC qui signifie fiable, original, complet, actuel et cité.

1. Fiable - ELEVE - Fiable, car il compte des informations sur un réseau de 692 stations à travers Chicago

2. Original - ELEVE - Fournisseur exclusif (Company Cyclist Bike SHare)

3. Complet - MOYEN - Les paramètres correspondent à la plupart des paramètres des produits Cyclist

4. Current - MOYEN - Données qui date de l'année 2020 et début 2021

5. Cité - ELEVE - Données recueillies auprès de l'entreprise elle même

Dans l'ensemble, l'ensemble de données est considéré comme des données de bonne qualité.

2.4 Sélection Des Données

- 202004-divvy-tripdata.zip
- 202005-divvy-tripdata.zip
- 202006-divvy-tripdata.zip
- 202007-divvy-tripdata.zip
- 202008-divvy-tripdata.zip
- 202009-divvy-tripdata.zip
- 2020010-divvy-tripdata.zip
- 2020011-divvy-tripdata.zip
- 2020012-divvy-tripdata.zip
- 2021001-divvy-tripdata.zip

- 2021002-divvy-tripdata.zip
- 2021003-divvy-tripdata.zip

ÉTAPE 3 : PHASE DE PROCESSUS

3.1 Utilisation De Différents Outils

- Google Sheet : Utilisé pour le nettoyage initial et le traitement des fichiers csv individuels, cet outil fourni un moyen rapide et simple de transformer les données et de créer les nouvelles colonnes dont nous avons besoin. Je l'ai également utilisé pour analyser la durée moyenne des trajets par mois, le mode et le nombre de trajets par mois, et créer une visualisation unique montrant le pourcentage d'utilisateurs dans chaque catégorie.

- Tableau : Utilisé pour créer deux visualisations à partir de feuilles de calcul, car la plateforme prête lui-même à la fonctionnalité « glisser-déposer » et permet de créer des visuels simples mais clairs et de rejoindre données provenant de diverses sources.

- RStudio : Utilisé RStudio pour l'essentiel de la manipulation, l'analyse et la visualisation en raison de sa capacité de traiter une grande quantité de données et c'est une approche très logique des fonctions et de la syntaxe. La bibliothèque *sqldf* est un bonne outil, car il nous a permis d'utiliser des requêtes SQL sur la grande base de données.

3.2 Nettoyage Et Manipulation Des Données

- Google Sheet : Les données ont été livrées dans un format .csv, j'ai utilisé la fonction « text to columns » pour formater les données dans des lignes et des colonnes. J'ai ensuite supprimé tous les doublons des données. J'ai converti les colonnes « started_at » et « ended_at » qui était au format date-heure dans le format jj/mm/aaaa - hh:mm. J'ai créé une colonne ride_length avec une formule pour soustraire la valeur started_at de ended_at valeur. J'ai formaté les données dans la colonne comme hh:mm:ss. J'ai ensuite copié et collé les valeurs dans une nouvelle colonne ride_length. Ensuite, j'ai trié la feuille par longueur de trajet dans l'ordre croissant, et supprimé plusieurs rangées où la longueur du trajet était négative (ce qui indique un problème où le « ended at » valeur était avant la valeur « started at ».

) J'ai envisagé de supprimer les parcours où la longueur = 0.
J'ai créé une colonne « day_of_week » et j'ai utilisé la fonction WEEKDAY pour créer une représentation numérique du jour de la semaine où chaque vélo a été vérifié, où le dimanche était représenté par 1.
Trier la feuille par « started at date » en ordre croissant. J'ai ensuite créé des tableaux croisés dynamiques pour calculer le nombre de trajets et la durée moyenne des trajets par mois pour les membres et les employés occasionnels. Afin de calculer et de représenter la durée moyenne mensuelle du trajet, j'ai créé une formule pour convertir le temps en minutes ***(=HOUR(A2)x60+MINUTE(A2)+SECOND(A2)/60.)***

- RStudio : Tout d'abord, j'ai lus tous les fichiers csv dans des bases de données séparées dans RStudio, et j'entrevois des bases de données pour voir si les types de données que j'ai définis dans Google Sheet ont été préservés.
  Etant donné que les types de données n'étaient pas préservés, j'ai décidé de fusionner toutes les bases de données dans un, après quoi je convertirais certains types de données et puis nettoyer le reste comme nécessaire, une fois que j'aurais progressé à travers mon analyse.

Installer les librairies

```
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(lubridate)

##
## Attachement du package : 'lubridate'

## Les objets suivants sont masqués depuis 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(markdown)
library(sqldf)

## Le chargement a nécessité le package : gsubfn

## Le chargement a nécessité le package : proto

## Le chargement a nécessité le package : RSQLite

library(maps)

##
## Attachement du package : 'maps'

## L'objet suivant est masqué depuis 'package:purrr':
##
##     map

library(rgdal)

## Le chargement a nécessité le package : sp

## Please note that rgdal will be retired by the end of 2023,
## plan transition to sf/stars/terra functions using GDAL and PROJ
## at your earliest convenience.
##
## rgdal: version: 1.5-27, (SVN revision 1148)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.2.1, released 2020/12/29
## Path to GDAL shared files: C:/Users/Asus/Documents/R/win-
library/4.1/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 7.2.1, January 1st, 2021, [PJ_VERSION: 721]
## Path to PROJ shared files: C:/Users/Asus/Documents/R/win-
library/4.1/rgdal/proj
## PROJ CDN enabled: FALSE
## Linking to sp version:1.4-5
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading sp
or rgdal.
## Overwritten PROJ_LIB was C:/Users/Asus/Documents/R/win-
library/4.1/rgdal/proj

library(ggrepel)
```

Définir le répertoire de travail et créer des bases de données pour chaque fichier csv

```
apr_20 <- read.csv("202004-divvy-tripdata.csv", sep=";")
may_20 <- read.csv("202005-divvy-tripdata.csv", sep=";")
```

```
jun_20 <- read.csv("202006-divvy-tripdata.csv", sep=";")
jul_20 <- read.csv("202007-divvy-tripdata.csv", sep=";")
aug_20 <- read.csv("202008-divvy-tripdata.csv", sep=";")
sep_20 <- read.csv("202009-divvy-tripdata.csv", sep=";")
oct_20 <- read.csv("202010-divvy-tripdata.csv", sep=";")
nov_20 <- read.csv("202011-divvy-tripdata.csv", sep=";")
dec_20 <- read.csv("202012-divvy-tripdata.csv", sep=";")
jan_21 <- read.csv("202101-divvy-tripdata.csv", sep=";")
feb_21 <- read.csv("202102-divvy-tripdata.csv", sep=";")
mar_21 <- read.csv("202103-divvy-tripdata.csv", sep=";")
```

Aperçu d'une base de données, pour voir si les types de données d'Excel ont été préservés (ils ne l'étaient pas)

```
glimpse(dec_20)

## Rows: 131,573
## Columns: 15
## $ ride_id            <chr> "70B6A9A437D4C30D", "158A465D4E74C54A",
"5262016E0F~
## $ rideable_type      <chr> "classic_bike", "electric_bike",
"electric_bike", "~
## $ started_at         <chr> "2020-12-27 12:44:29", "2020-12-18 17:37:15",
"2020~
## $ ended_at           <chr> "2020-12-27 12:55:06", "2020-12-18 17:44:19",
"2020~
## $ start_station_name <chr> "Aberdeen St & Jackson Blvd", "", "", "", "",
"", "~
## $ start_station_id   <chr> "13157", "", "", "", "", "", "", "", "", "",
"", ""~
## $ end_station_name   <chr> "Desplaines St & Kinzie St", "", "", "", "",
"", ""~
## $ end_station_id     <chr> "TA1306000003", "", "", "", "", "", "", "", "",
"",~
## $ start_lat          <dbl> 41.87773, 41.93000, 41.91000, 41.92000,
41.80000, 4~
## $ start_lng          <dbl> -87.65479, -87.70000, -87.69000, -87.70000, -
87.590~
## $ end_lat            <dbl> 41.88872, 41.91000, 41.93000, 41.91000,
41.80000, 4~
## $ end_lng            <dbl> -87.64445, -87.70000, -87.70000, -87.70000, -
87.590~
## $ member_casual      <chr> "member", "member", "member", "member",
"member", "~
## $ ride_length        <chr> "00:10:37", "00:07:04", "00:06:55", "00:05:53",
"00~
```

```
## $ day_of_week        <int> 1, 6, 3, 3, 3, 3, 5, 5, 7, 6, 7, 7, 7, 7, 6, 6,
2, ~
```

Fusion de toutes les bases de données d'abord

Créer la liaison

```
df_1 <- do.call("rbind", list(apr_20, may_20, jun_20, jul_20, aug_20, sep_20,
oct_20, nov_20, dec_20, jan_21, feb_21, mar_21))
```

Avant de fusionner les dataframes, j'ai calculé le nombre de lignes dans chaque dataframe et j'ai écrit une fonction pour comparer la somme de toutes les lignes à la base de données finale. Une fois que j'ai fusionné les bases de données, j'ai exécuté cette fonction pour vérifier si toutes les lignes ont été préservées. , ce qui était le cas.

Calculer le nombre total de lignes pour vérifier la fusion

```
tot_rows <- nrow(apr_20) + nrow(may_20) + nrow(jun_20) + nrow(jul_20) +
nrow(aug_20) + nrow(sep_20) + nrow(oct_20) + nrow(nov_20) + nrow(dec_20) +
nrow(jan_21) + nrow(feb_21) + nrow(mar_21)
```

Vérifier que le nombre de lignes correspond

```
if (tot_rows == nrow(df_1)){
  print("Binding complete, data verified.")
} else{
  print("Error, please verify your data.")
}
```

```
## [1] "Binding complete, data verified."
```

J'ai ensuite muté les colonnes started_at et ended_at au format datetime en utilisant la fonction as_datetime, et j'ai muté la colonne ride_length dans un format de différence de temps en utilisant as.difftime qui a retourné le temps en secondes.

Modifier les types de données started_at, ended_at en datetime et ride_length en time pour toutes les bases de données.

```
df_1 <- mutate(df_1, started_at = as_datetime(df_1$started_at))
df_1 <- mutate(df_1, ended_at = as_datetime(df_1$ended_at))
df_1 <- mutate(df_1, ride_length = as.difftime(df_1$ride_length, "%H:%M:%S"))
```

Je suis ensuite passé à la création d'autres bases de données pour creuser un peu plus profondément dans les données.

En utilisant la fonction sqldf, j'ai créé deux requêtes qui rendaient le nom, le nombre de voyages par station et la latitude et la longitude des cinq premières stations de départ pour les membres et les employés occasionnels, respectivement. Remarquant que cela renvoyait initialement une valeur nulle dans les noms de station, j'ai reformulé la requête à exclure les valeurs sans noms, puis regrouper les données par nom de station, les classer par trajet et ensuite limité à 5. J'ai ensuite lié ces deux bases de données en une seule.

Une analyse rapide pour trouver la moyenne de la colonne ride_length, et la longueur de trajet maximale.

```
mean_r_length <- as.numeric(mean(df_1$ride_length, na.rm=TRUE))/60
cat("The average ride length over the year is:",mean_r_length,"minutes")

## The average ride length over the year is: 24.11659 minutes

max_r_length <- as.numeric(max(df_1$ride_length, na.rm=TRUE))/3600
cat("The longest ride for the year was:",max_r_length,"hours")

## The longest ride for the year was: 23.99833 hours
```

Maintenant, je vais créer une nouvelle base de données avec les données que je veux pour une visualisation. J'utiliserai sqldf pour démontrer certaines de mes capacités SQL.

Création de deux bases de données avec les 5 premières stations de départ et d'arrivée + nombre de voyages par mem/cas. Top 5 des géolocalisations de départ pour les membres.

```
mem_start_geo <- sqldf("SELECT member_casual, start_station_name AS Start,
                start_lat AS Starting_Latitude,
                start_lng As Starting_Longitude, count(start_station_name) AS
Num_Trips
                FROM df_1
                WHERE start_station_name IS NOT ''
                AND member_casual = 'member'
                GROUP BY start_station_name
                ORDER BY count(start_station_name) DESC
                LIMIT 5", method = "auto")
```

Top 5 des géolocalisations de départ pour les occasionnels.

```
cas_start_geo <- sqldf("SELECT member_casual, start_station_name AS Start,
                start_lat AS Starting_Latitude, start_lng As
Starting_Longitude,
                count(start_station_name) AS Num_Trips
                FROM df_1
                WHERE start_station_name IS NOT ''
                AND member_casual = 'casual'
                GROUP BY start_station_name
                ORDER BY count(start_station_name) DESC
                LIMIT 5", method = "auto")
```

Ensuite, j'ai fait exactement la même chose pour les stations finales.

Top 5 des géolocalisations finales pour les membres.

```
mem_end_geo <- sqldf("SELECT member_casual, end_station_name AS End,
                end_lat AS Ending_Latitude,
                end_lng As Ending_Longitude, count(end_station_name) AS
Num_Trips
                FROM df_1
                WHERE end_station_name IS NOT ''
                AND member_casual = 'member'
                GROUP BY end_station_name
```

```
                ORDER BY count(end_station_name) DESC
                LIMIT 5", method = "auto")
```

Top 5 des géolocalisations finales pour les employés occasionnels.

```
cas_end_geo <- sqldf("SELECT member_casual, end_station_name AS End,
                end_lat AS Ending_Latitude, end_lng As Ending_Longitude,
                count(end_station_name) AS Num_Trips
                FROM df_1
                WHERE end_station_name IS NOT ''
                AND member_casual = 'casual'
                GROUP BY end_station_name
                ORDER BY count(end_station_name) DESC
                LIMIT 5", method = "auto")
```

Les bases de données séparées ont ensuite été combinées pour le tracé, mais après avoir tenté quelques tracés sans succès, je me suis rendu compte que la latitude et les longitudes devaient être converties en nombres, mais mes tentatives initiales rendaient des valeurs NULL jusqu'à ce que je réalise que les virgules dans les chaînes causaient le problème, j'ai donc utilisé la fonction gsub pour remplacer tous les points par des virgules et combiné avec la fonction as.numeric et cela a fait l'affaire.

Lier les deux tables dans une base de données et les visualiser.

```
start_geo <- rbind(mem_start_geo, cas_start_geo)
View(start_geo)
```

Changement du type de données des coordonnées en nombres réels à utiliser pour les tracés.

```
start_geo$Starting_Latitude =
as.numeric(gsub(",",".",start_geo$Starting_Latitude,fixed=TRUE))
start_geo$Starting_Longitude =
as.numeric(gsub(",",".",start_geo$Starting_Longitude,fixed=TRUE))
```

Lier les deux tables dans une base de données et les visualiser.

```
end_geo <- rbind(mem_end_geo, cas_end_geo)
View(end_geo)
```

Changement du type de données des coordonnées en nombres réels à utiliser pour les tracés.

```
end_geo$Ending_Latitude = as.numeric(gsub(",",".",end_geo$Ending_Latitude,
fixed=TRUE))
end_geo$Ending_Longitude = as.numeric(gsub(",",".",end_geo$Ending_Longitude,
fixed=TRUE))
```

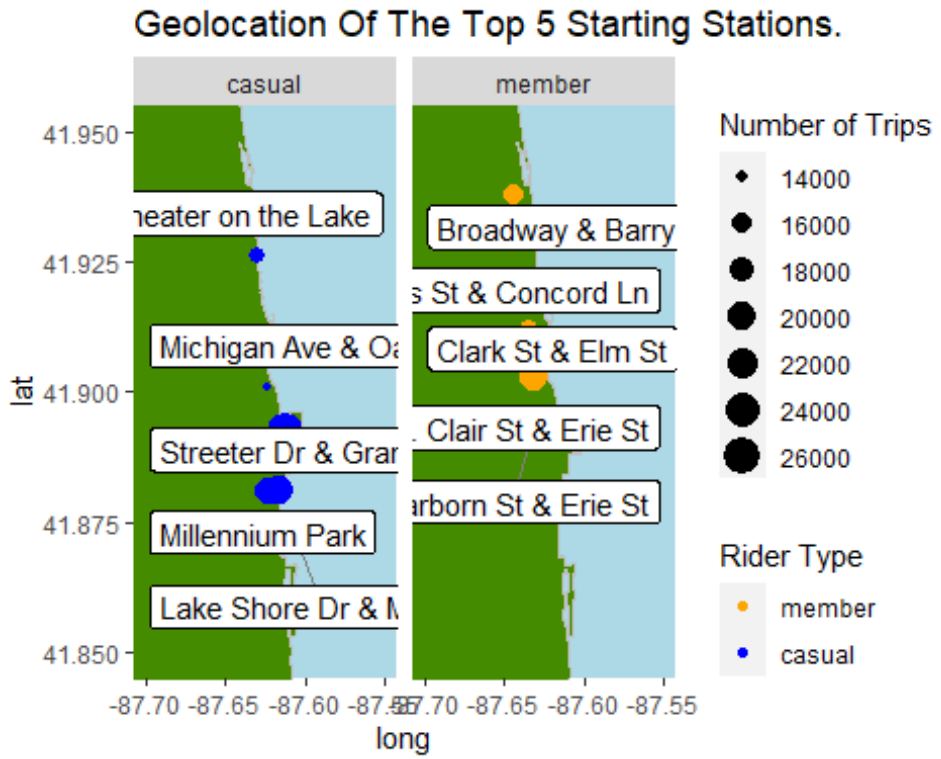
Création d'une carte de géolocalisation des 5 premières stations de départ et de fin.

Obtenir un fichier Shapefile de Chicago, et le fortifier dans une base de données.

```
chi_map <- readOGR(dsn = "C:/Users/Asus/Downloads/project/project", layer =
"geo_export_f0375d57-0e9c-4a83-b39d-e03fc8e9ab4a")

## Warning in OGRSpatialRef(dsn, layer, morphFromESRI = morphFromESRI,
dumpSRS =
## dumpSRS, : Discarded datum WGS84 in Proj4 definition: +proj=longlat
+ellps=WGS84
## +no_defs

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Asus\Downloads\project\project", layer:
"geo_export_f0375d57-0e9c-4a83-b39d-e03fc8e9ab4a"
## with 1 features
## It has 4 fields

chi_df = fortify(chi_map)

## Regions defined for each Polygons
```

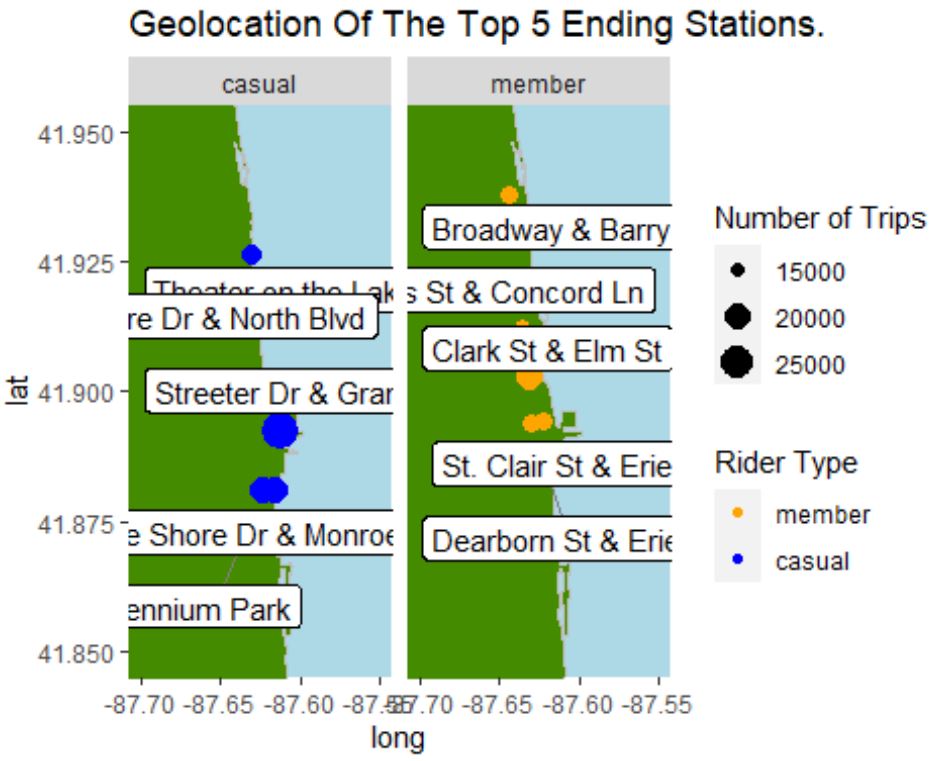
Tracer la géolocalisation de la station de départ.

```
ssgmap <-ggplot() +
    geom_polygon(data = chi_df, aes(x = long, y=lat , group = group), colour
= 'grey',
    fill = 'chartreuse4', size = .7) +
    geom_point(data = start_geo,
               aes(x = Starting_Longitude, y = Starting_Latitude, size =
Num_Trips, color = member_casual),
               alpha = 1) +
    geom_label_repel(data = start_geo,
                     aes(x = Starting_Longitude, y = Starting_Latitude, label =
Start),
                     box.padding   = 0.4,
                     point.padding = 0.65,
                     segment.color = 'gray50') +
  scale_colour_manual(values=c(member = 'orange',  casual= 'blue'))+
  facet_wrap(~member_casual) +
  labs(title = "Geolocation Of The Top 5 Starting Stations.", size = 'Number
of Trips',
       color = 'Rider Type') +
  coord_cartesian(xlim = c(-87.7, -87.55), ylim = c(41.85, 41.95))+
  theme(panel.background = element_rect(fill = "lightblue")) +
        theme(panel.border = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
ssgmap
```



Tracer la géolocalisation de la station finale.

```
esgmap <- ggplot() +
    geom_polygon(data = chi_df, aes(x = long, y=lat , group = group), colour
= 'grey',
    fill = 'chartreuse4', size = .7) +
  geom_point(data = end_geo,
             aes(x = Ending_Longitude, y = Ending_Latitude, size = Num_Trips,
color = member_casual),
             alpha = 1) +
  geom_label_repel(data = end_geo,
                   aes(x = Ending_Longitude, y = Ending_Latitude, label =
End),
                   box.padding   = 0.4,
                   point.padding = 0.65,
                   segment.color = 'gray50') +
  scale_colour_manual(values=c(member = 'orange',  casual= 'blue')) +
  facet_wrap(~member_casual) +
  labs(title = "Geolocation Of The Top 5 Ending Stations.", size = 'Number of
Trips',
       color = 'Rider Type') +
  coord_cartesian(xlim = c(-87.7, -87.55), ylim = c(41.85, 41.95)) +
    theme(panel.background = element_rect(fill = "lightblue")) +
    theme(panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank())
esgmap
```



J'ai ensuite créé une base de données avec les données de mon fichier shapefile, afin que je puisse l'utiliser pour tracer mes visualisations. J'ai ensuite tracé les cinq premiers emplacements de départ et de fin, en utilisant un facetwrap pour séparer les parcelles par member_casual. Pour le mode annuel, j'ai lancé une autre requête sqldf, pour retourner le jour de la semaine, un décompte des jours de la semaine et regroupés par member_casual, et classés par ordre décroissant de jour de la semaine. J'ai ensuite remplacé chaque jour numérique de la semaine par les noms du jour de la semaine, Parce que je voulais les ploter.

Requêtes SQL pour le Mode annuel de day_of_week (total, members, casuals).

```
mode_t <- sqldf("SELECT day_of_week, member_casual, COUNT(day_of_week) AS
Total
                 FROM df_1
                 GROUP BY member_casual, day_of_week
                 ORDER BY day_of_week DESC", method = "auto")
```

Remplacement des valeurs numériques par les noms des jours de semaine.

```
mode_t$day_of_week[mode_t$day_of_week == "1"] <- "Sunday"
mode_t$day_of_week[mode_t$day_of_week == "2"] <- "Monday"
mode_t$day_of_week[mode_t$day_of_week == "3"] <- "Tuesday"
mode_t$day_of_week[mode_t$day_of_week == "4"] <- "Wednesday"
mode_t$day_of_week[mode_t$day_of_week == "5"] <- "Thursday"
mode_t$day_of_week[mode_t$day_of_week == "6"] <- "Friday"
mode_t$day_of_week[mode_t$day_of_week == "7"] <- "Saturday"
```

J'ai utilisé une fonction pour attribuer des niveaux de facteurs aux variables du jour de la semaine, de sorte que RStudio n'a pas à les réorganiser quand je les ai tracés sur l'axe X.

Traçage des modes.

Cette fonction se verrouille dans l'ordre que j'ai établi pour que l'axe x ne soit pas trié.

```
mode_t$day_of_week <- factor(mode_t$day_of_week, levels =
rev(unique(mode_t$day_of_week)), ordered=TRUE)
```

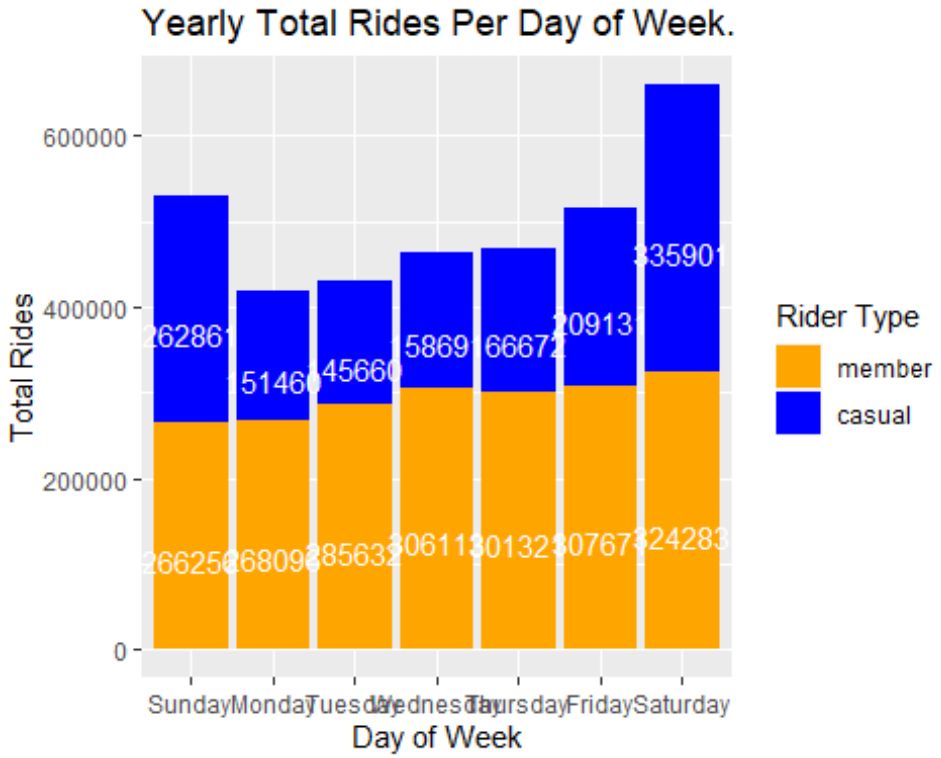
Cette fonction trouve la somme des cyclistes occasionnels et membres, à utiliser pour tracer des étiquettes au milieu de chaque barre.

```
mode_t <- mode_t %>%
  arrange(day_of_week, rev(member_casual)) %>%
  group_by(day_of_week) %>%
  mutate(GTotal = cumsum(Total) - 0.5 * Total)
```

Un graphique à barres empilées avec les modes annuels pour tous les cyclistes.

```
Mode_plot <- ggplot(data = mode_t, aes(x = day_of_week, y = Total, fill =
member_casual)) +
             scale_fill_manual(values=c(member = 'orange',  casual= 'blue'))
+
  geom_col() +
```

```
  geom_text(aes(y = GTotal, label = Total), vjust = 1.5, colour = "white") +
  labs(title = "Yearly Total Rides Per Day of Week.", x = "Day of Week",
       y = "Total Rides", fill = "Rider Type") +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
Mode_plot
```



J'ai ensuite tracé le mode du jour de la semaine avec un graphique à barres empilées, et inclus les totaux pour membres/employés occasionnels. Enfin, j'ai examiné les types de vélos utilisés par les membres et les employés occasionnels. Encore une fois, j'ai utilisé un sqldf requête, qui renvoyait les types de vélos avec un compte des types de vélos comme le champs member_casual, puis regroupés par member_casual puis par les types de vélos, et les a ordonné par le compte. J'ai changé les noms des types de vélos juste pour supprimer les underscores, puis créé un plot côte à côte.

Une requête pour retourner les résultats liés aux types de vélos utilisés par les membres

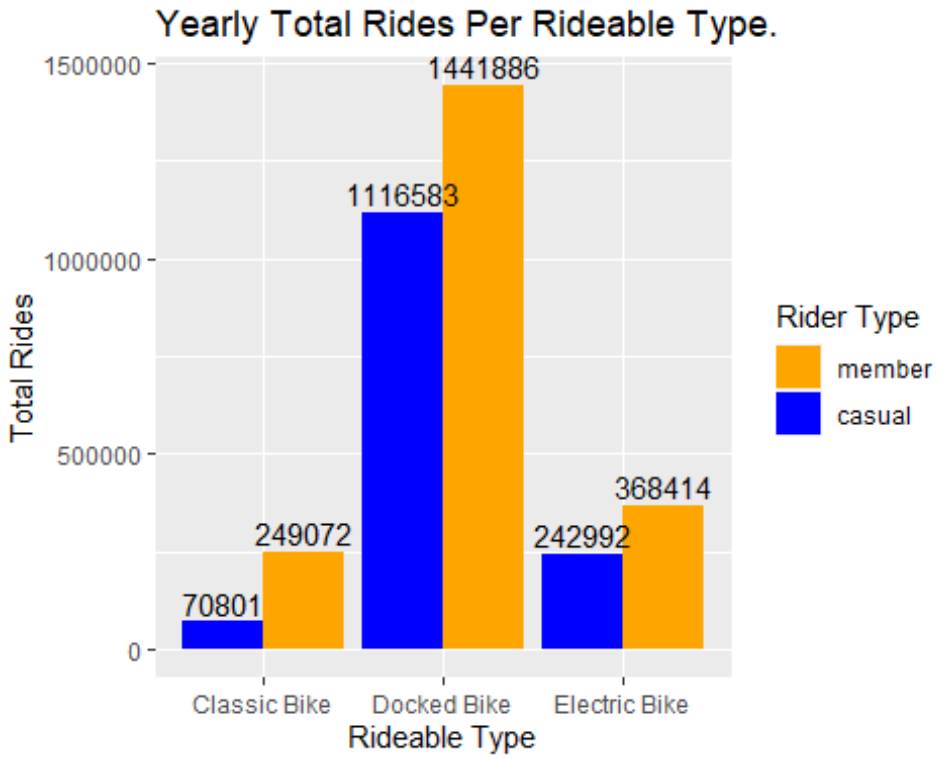
```
bike_df <- sqldf("SELECT rideable_type, member_casual, count(rideable_type)
as number_of_uses
                 FROM df_1
                 GROUP BY member_casual, rideable_type
                 ORDER BY count(rideable_type) DESC", method = "auto" )
```

Afficher les noms du type de vélo pour supprimer le trait de soulignement

```
bike_df$rideable_type[bike_df$rideable_type == "classic_bike"] <- "Classic
Bike"
bike_df$rideable_type[bike_df$rideable_type == "docked_bike"] <- "Docked
Bike"
bike_df$rideable_type[bike_df$rideable_type == "electric_bike"] <- "Electric
Bike"
```

Un graphique de bar côte à côte avec le compte annuel de vélo pour tous les cyclistes

```
bike_plot <- ggplot(data = bike_df, aes(x = rideable_type, y =
number_of_uses, fill = member_casual)) +
  scale_fill_manual(values=c(member = 'orange',  casual= 'blue')) +
  geom_col(position = "dodge") +
  geom_text(aes(label = number_of_uses),  vjust = -0.3 ,colour = "black",
            position = position_dodge(.9)) +
  labs(title = "Yearly Total Rides Per Rideable Type.", x = "Rideable Type",
       y = "Total Rides", fill = "Rider Type") +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
bike_plot
```



ÉTAPE 4 : ANALYSER

Nombre total de voyages :

Dans l'ensemble, il s'agit du nombre total de déplacements valides pour l'année, calculé à l'aide des tableaux croisés dynamiques dans EXCEL :

- Membres : 2052077
- Employés occasionnels : 1427121
- Total : 3479198

59 % de tous les voyages ont été effectués par des membres, tandis que les 41 % restants ont été effectués par des cyclistes occasionnels. Le mois où le nombre de voyages a été le plus élevé a été août 2020, alors que le mois où le nombre de voyages a été le plus faible a

été Février 2021. Il n'est pas surprenant que les mois d'été aient été plus nombreux que les mois d'hiver.

Durée des trajets :

- Durée moyenne du trajet pour l'année : 24 minutes
- Durée de trajet maximale pour l'année était : 23,99 heures (peut-être quelqu'un a laissé leur vélo pendant la nuit sans le vérifier de nouveau?)

Les distances moyennes par mois ont également été calculées à l'aide des tableaux croisés dynamiques, mais ce serait mieux de les montrer visualisés dans la section suivante.

Voici quelques points importants à considérer à partir de ces données :
Dans l'ensemble, les voyages des cyclistes occasionnels étaient plus longs en moyenne que ceux des membres, souvent deux fois en moyenne. Cela indique que les cyclistes occasionnels utilisaient le service plus longtemps, des trajets tranquilles, tandis que les membres semblent l'utiliser davantage pour les déplacements. Cette hypothèse devrait être n'a pas été oublié dans la section suivante.

Juillet est le mois où la durée moyenne du voyage est la plus longue, et janvier est le mois où la moyenne est la plus courte. durée du voyage : cela correspondait à la tendance des voyages plus longs pendant les mois les plus chauds et voyages plus courts pendant les mois les plus froids.

Stations les plus fréquentées:

- Les 5 meilleures stations de départ pour les membres :

Clark St. &. Elm St.
Broadway & Barry Ave.
St. Clair St. & Erie St.
Dearborn St. & Erie St.
Wells St. & Concord Ln.

- Les 5 meilleures stations de départ pour les occasionnels :

Streeter Dr. & Grand Ave.
Millennium Park
Lake Shore Dr. & Monroe St.
Theater on the Lake.
Michigan Ave. & Oak St.

- Les cinq principales stations d'arrivées pour les membres :

Clark St. &. Elm St.
Broadway & Barry Ave.
St. Clair St. & Erie St.
Dearborn St. & Erie St.
Wells St. & Concord Ln.

- Les 5 meilleures stations d'arrivées pour les occasionnels :

Streeter Dr. & Grand Ave.
Millennium Park
Lake Shore Dr. & Monroe St.
Theater on the Lake.
Lake Shore Dr.. & North Blvd.

Il semblerait que la plupart des voyages ont eu lieu dans le centre de la ville, bien que les voyages par les occasionnels cyclistes étaient concentrés dans une zone légèrement au sud d'où la plupart des voyages par les membres étaient.

Jours la de semaine les plus occupés (mode) :

Au cours de l'année, le samedi a été le jour le plus occupé de la semaine pour les membres et les cyclistes occasionnels, alors que les lundis ont connu le moins de balades de la part des membres et que mardi a connu le moins de balades de la part des cyclistes occasionnels.

Les fins de semaine ont été plus occupées que les jours de semaine, et il semble qu'il y a beaucoup moins de service en semaine, peut-être qu'ils préfèrent l'utiliser pour les loisirs plutôt qu'une course, tandis que les membres étaient beaucoup plus susceptibles d'utiliser le service pour se rendre au travail. Néanmoins, il y a un certain nombre de membres occasionnels qui utilisent le service pendant la semaine, et cela pourrait représenter un secteur de croissance.

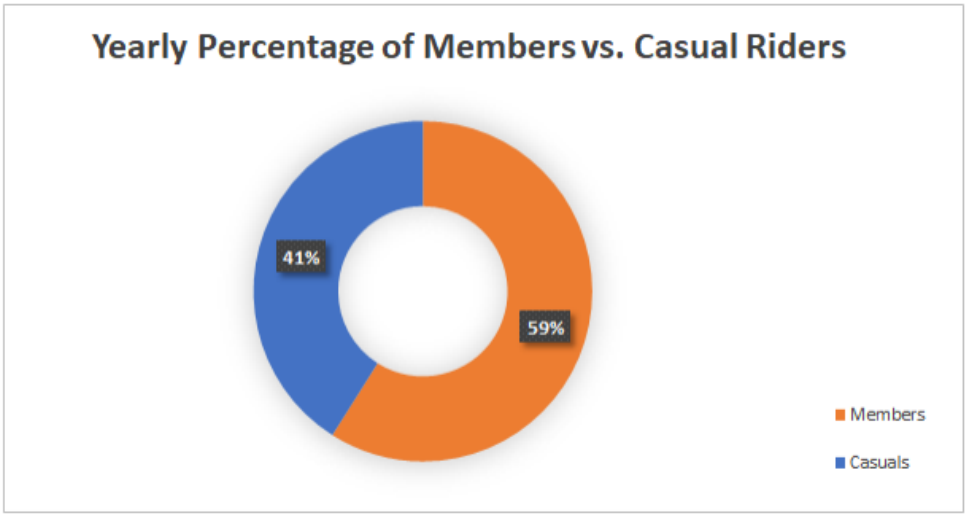
Préférence pour le type de vélo :

Enfin, l'analyse des types de vélos transportables a indiqué que les vélos avec un cadenas étaient les plus populaires autour, suivie par des vélos électriques et des vélos classiques en dernière place. Cette tendance était cohérente à travers les membres et les cyclistes occasionnels.

ÉTAPE 5 : PARTAGER

Nombre total de voyages :

Étant donné que l'objectif principal de l'analyse est de trouver des moyens de convertir les cyclistes occasionnels en membres, il est important pour comprendre la composition actuelle de notre clientèle. Comme vous pouvez le voir ci-dessous, les membres représentaient 59 % du nombre total de voyages effectués au cours de l'année. Les voyageurs occasionnels représentaient 41 % des voyages restants, ce qui indique la possibilité de convertir des cyclistes occasionnels en membres, et toute analyse ultérieure devrait prendre cette proportion en compte.



En ce qui concerne les voyages mensuels, il y a une relation claire entre le nombre de voyages et la saisonnalité; il y a beaucoup moins de trajets pendant les mois plus froids que dans les mois plus chauds, et cette tendance est particulièrement prononcée lorsque l'on compare les membres aux cyclistes occasionels comme par le graphique ci-dessous. Cependant, alors que les membres sont beaucoup plus susceptibles de braver le froid que les cyclistes occasionnels, il y a encore quelques milliers de cyclistes occasionnels qui ne sont pas découragés par le froid.

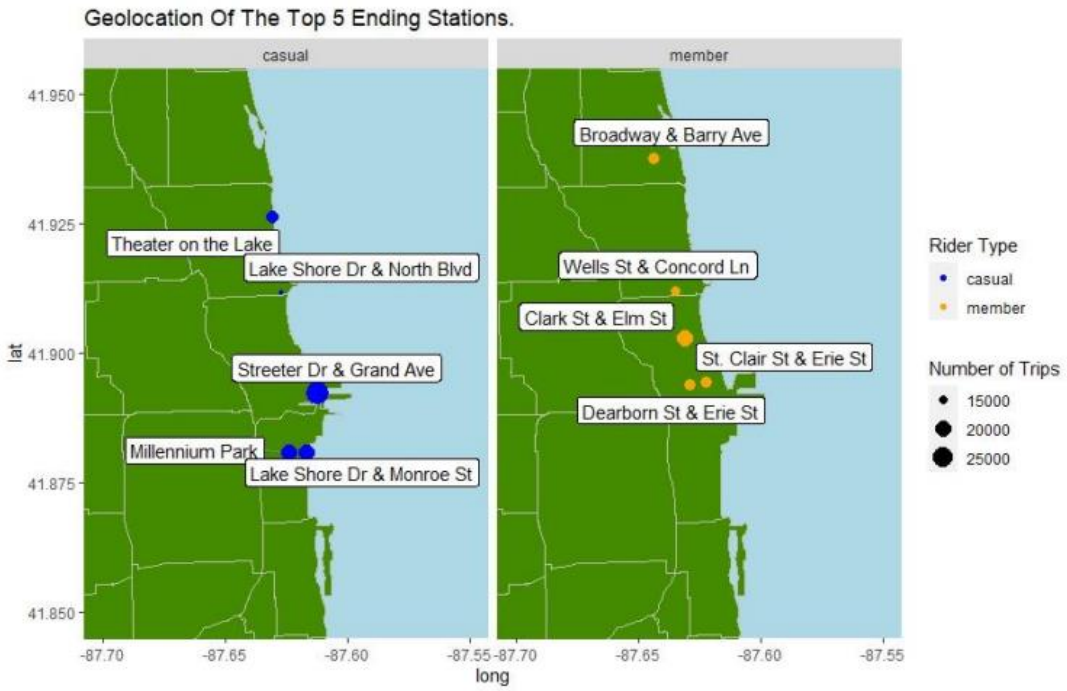
Durée des trajets :

La durée moyenne du trajet pour l'année était de 24 minutes, mais ce chiffre est inférieur à la durée moyenne pour les cyclistes occasionnels et plus que la durée de trajet moyenne pour les membres. Les employés occasionnels semblent prendre des trajets qui sont deux fois plus longs en moyenne que ceux des membres, indiquant que leurs voyages couvrent plus de distance, sont entrepris à un rythme tranquille, ou que peut-être les cyclistes occasionnels ont tendance à arriver à destination, garder leur vélo « vérifié » jusqu'à ce qu'ils aient terminé, puis les ramener. La durée moyenne du trajet des membres semble

pour répondre à l'hypothèse selon laquelle ils utilisent le service primaire pour se rendre au travail.

Stations les plus fréquentées :

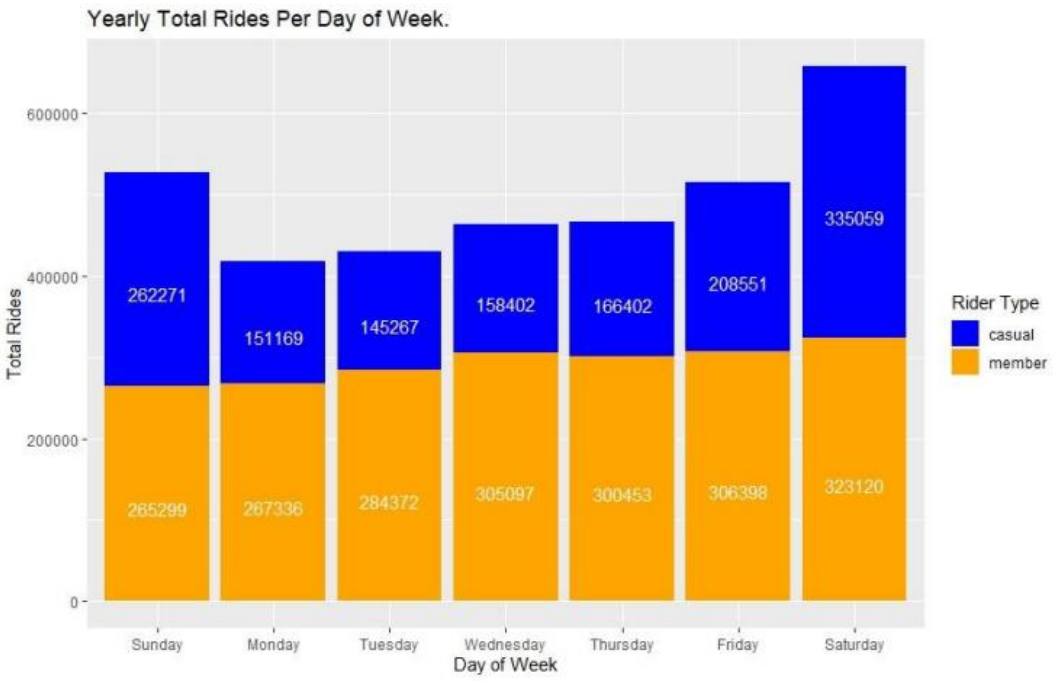
Il ressort clairement de ces visualisations que la station la plus occupée pour les cyclistes occasionnels est Streeter Drive et Grand Avenue, tandis que pour les membres, c'est Clark Street et Elm Street. Comme on l'a mentionné plus haut, les stations les plus fréquentées pour les cyclistes occasionnels sont concentrées légèrement au sud de ceux des membres, et compte tenu du fait que le parc Millenium pour les membres occasionnels, cela ajoute encore plus de poids à l'hypothèse qu'ils utilisent le service principalement pour les loisirs plutôt que pour les déplacements professionnels.



Jours de semaine les plus occupés (mode) :

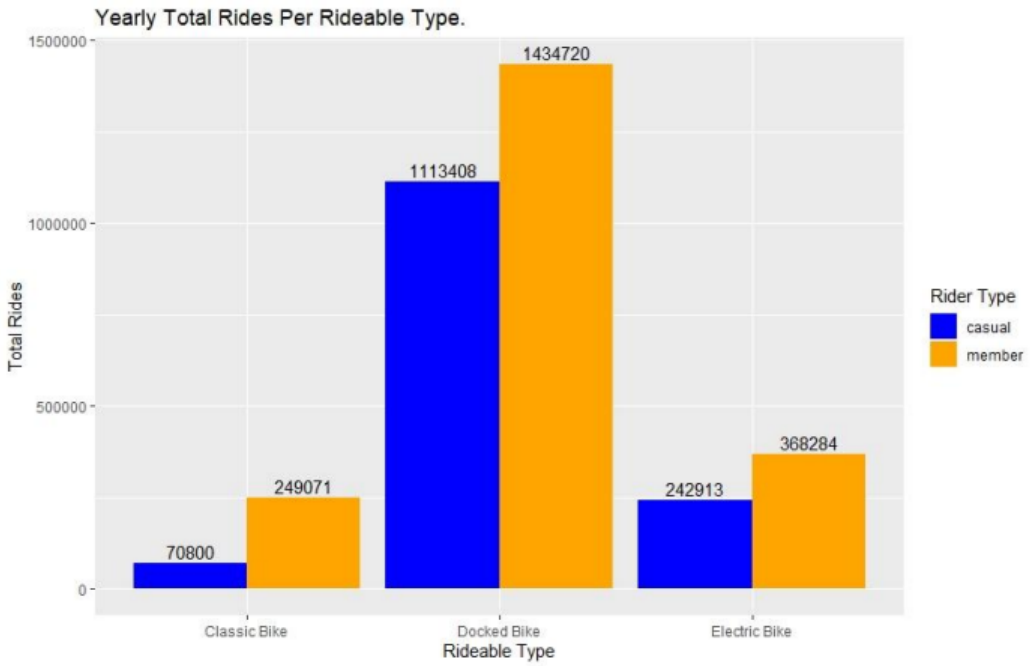
Les données indiquent que le samedi est le jour de la semaine le plus populaire pour les membres et les cyclistes occasionnels, et les week-ends ont été plus occupés dans l'ensemble que les jours de la semaine. Lundi est le jour le plus calme dans l'ensemble pour les membres, tandis que le mardi est la journée la plus calme pour les cyclistes occasionnels.

Sur le graphique ci-dessous, il est clair qu'il y a une baisse significative de l'utilisation par les cyclistes occasionnels pendant la semaine, alors que les membres semblent rouler plus uniformément tout au long. Encore une fois, cela correspond à l'hypothèse que les cyclistes occasionnels utilisent principalement le service pour les loisirs, bien qu'il y ait encore un nombre décent des voyages effectués au cours de la semaine.



Préférence pour le type de vélo :

Outre le fait que les vélos avec cadenas sont l'option la plus populaire de tous, il est également clair que les vélos classiques sont plutôt impopulaires avec les membres occasionnels.



ÉTAPE 6 : ACTION

D'après l'analyse ci-dessus, voici mes principales recommandations :

1. Afin d'attirer davantage les voyageurs occasionnels, toute campagne de publicité pourrait se concentrer sur les avantages de l'utilisation du service pour les déplacements au travail. Bien qu'il soit vrai que cyclistes occasionnels ont tendance à utiliser le service plus au cours des week-ends, une promotion ciblant ceux qui utilisent régulièrement le service pendant la semaine pourraient avoir plus de succès qu'un qui tente de convertir le week-end que les cyclistes occasionnels en membres.

2. Compte tenu de la nature saisonnière du service, il serait prudent de l'annoncer pendant les mois les plus chauds, lorsque les gens sont dehors à profiter. Considérant les volumes élevés de cyclistes le week-end, cela présente également une occasion pour l'équipe de marketing pour atteindre un plus grand nombre de personnes avec tout type de promotion « sur le terrain ».

3. Les promotions « sur le terrain » ou celles qui utilisent les médias imprimés devraient se concentrer sur les stations qui voient le plus grand nombre de départs

et d'arrivés. Une stratégie ciblée dans ces zones atteindront le nombre maximum de cyclistes, par opposition à une approche globale, plus large.

4. Tout appel à la population de cyclistes occasionnels devrait éviter l'utilisation du vélo classique dans leur matériel promotionnel, et devrait plutôt se concentrer sur les vélos avec cadenas ou électriques.

À l'avenir, il pourrait être judicieux d'attribuer à chaque cyclistes un numéro d'identification de cyclistes unique, car cela permet à l'entreprise de mieux suivre leurs habitudes de conduite afin de leur offrir d'autres services. Cela aidera à identifier les cyclistes occasionnels dont les modèles de conduite se rapprochent de ceux des membres, et permettrait des options de publicité plus ciblées. Des données supplémentaires sur les mois, sur les voyages pourrait également ouvrir la porte à des promotions visant à savoir combien un cyclistes pourrait économiser en passant d'un cycliste occasionnel à un membre.