```
## Ask
## Three questions will guide the future marketing program:
## 1. How do annual members and casual riders use Cyclistic bikes differently?
# R.1 There are a strange difference between the usability by each member type.
# R.2 The annual members are using classic and eletric bikes a lot compared with casual users.
# R.3 But, the average time of rides shows us that casual users are using bikes (more classic)
than annual members.
# R.4 When we compare the distance between members and casuals, there is a slightly difference
in favor of annual members using eletric bikes.

## 2. Why would casual riders buy Cyclistic annual memberships?
# R.1 Of course, I believe that there are some benefits to annual members
# R.2 So, casual riders are using bikes for much more time than annual members.

## 3. How can Cyclistic use digital media to influence casual riders to become members?
# R.1 Ciclystic Co. should be implementing a campaingn to call casual users to become a annual
members according the advantages available.
# R.2 The campaingn should be inserting banners on Rider Points (Bike Share Points) inviting
them to know the benefits of to be a annual member.
# R.3 Such as: You, casual rider, how many time are you using the bike? Short distances? Long
distances? Do you know the benefits of to be a annual member?
# R.4 How would be good a rider in family using ours eletric bikes on weekends? Do you know the
benefits for being a annual member?


# Database Ciclystic has 748.962 rows and 13 columns (6 months - Jul-Dec)
# rideable column has Classic and Eletric bike types
# member_casual column has Casual user and Member user


#-------------------------------------------------------------------------------
## Ações executadas com sucesso
# Import all datasets to a new database called Ciclystic
ciclystic <- read.csv("202412-divvy-tripdata.csv")
ciclystic <- read.csv("202411-divvy-tripdata.csv")
ciclystic <- read.csv("202410-divvy-tripdata.csv")
ciclystic <- read.csv("202409-divvy-tripdata.csv")
ciclystic <- read.csv("202408-divvy-tripdata.csv")
ciclystic <- read.csv("202407-divvy-tripdata.csv")
ciclystic <- read.csv("202406-divvy-tripdata.csv") #ok - Ciclystic.6m
ciclystic <- read.csv("202405-divvy-tripdata.csv") #ok - Ciclystic.6m
ciclystic <- read.csv("202404-divvy-tripdata.csv") #ok - Ciclystic.6m
ciclystic <- read.csv("202403-divvy-tripdata.csv") #ok - Ciclystic.3m
ciclystic <- read.csv("202402-divvy-tripdata.csv") #ok - Ciclystic.3m
ciclystic <- read.csv("202401-divvy-tripdata.csv") #ok - Ciclystic.3m

#-------------------------------------------------------------------------------
##----- Ação 1
# Efetuada limpeza do dataset com registros (rows) NULL
ciclystic_clean <- ciclystic %>%
  drop_na()

##----- Ação 2
# Inclusão de campo de RIDE TIME e RIDE DISTANCE na database nova COM 3 MESES
# variável fixa da esfera terrestre /// # variável fixa de PI
r1 <- 6371
p1 <- (3.141592653/180)
ciclystic_3m <- ciclystic_clean %>%
  mutate(ciclystic_clean, ride_min = (ymd_hms(ciclystic_clean$ended_at)-
ymd_hms(ciclystic_clean$started_at))/60) %>%
  mutate(ciclystic_clean, ride_distance = r1 * (2 * asin(pmin(1,
sqrt(((sin(((ciclystic_clean$end_lat - ciclystic_clean$start_lat) * p1)/2)^2 +
cos(ciclystic_clean$start_lat * p1)) * cos(ciclystic_clean$end_lat * p1) *
sin(((ciclystic_clean$end_lng - ciclystic_clean$start_lng) * p1)/2)^2))))))
# CREATE NEW ROWS TO IDENTIFY THE NAME OF DAY OF THE WEEK  AND OTHERS ROWS TO 3 MONTHS
ciclystic_3m$day <- strptime(ciclystic_3m$started_at, "%Y-%m-%d")
ciclystic_3m$weekday <- weekdays(ciclystic_3m$day)

# Create a dataset with rows about non zero distance rode by users
```

```
#- Dataset Cleaned
ciclystic_3mc <- ciclystic_3m %>%
  select(ride_id, rideable_type, member_casual, weekday, ride_min, ride_distance) %>%
  filter(ride_distance != 0)

# Create a dataset with rows about zero distance rode by users
ciclystic_3m_z <- ciclystic_3m %>%
  select(ride_id, rideable_type, member_casual, weekday, ride_min, ride_distance) %>%
  filter(ride_distance == 0)
#------------------------
# Inclusão de campo de  RIDE TIME,RIDE DISTANCE AND WEEKDAY na database nova COM 6 MESES
# variável fixa da esfera terrestre // # variável fixa de PI
r1 <- 6371
p1 <- (3.141592653/180)
# CREATE NEW ROWS TO IDENTIFY THE NAME OF DAY OF THE WEEK  AND OTHERS ROWS TO 6 MONTHS
ciclystic_6m <- ciclystic_clean
ciclystic_6m$day <- strptime(ciclystic_6m$started_at, "%Y-%m-%d")
ciclystic_6m$weekday <- weekdays(ciclystic_6m$day)
ciclystic_6m$ride_min <- ymd_hms(ciclystic_6m$ended_at)-ymd_hms(ciclystic_6m$started_at)
ciclystic_6mc$ride_min <- (ciclystic_6mc$ride_min/60)
ciclystic_6m$ride_distance <- r1 * (2 * asin(pmin(1, sqrt(((sin(((ciclystic_6m$end_lat -
ciclystic_6m$start_lat) * p1)/2)^2 + cos(ciclystic_6m$start_lat * p1)) *
cos(ciclystic_6m$end_lat * p1) * sin(((ciclystic_6m$end_lng - ciclystic_6m$start_lng) *
p1)/2)^2)))))
#-------------------------------------------------------------------------------
##------ Ação 2.0
# Cleaning Dataset
# Create a dataset cleaned - rows with zero distance rode by users
ciclystic_6m_z <- ciclystic_6m %>%
  select(ride_id, rideable_type, member_casual, weekday, ride_min, ride_distance) %>%
  filter(ride_distance == 0)
# Create a dataset with rows about non zero distance rode by users
#- Dataset Cleaned
ciclystic_6mc <- ciclystic_6m %>%
  select(ride_id, rideable_type, member_casual, weekday, ride_min, ride_distance) %>%
  filter(ride_distance != 0)

#-------------------------------------------------------------------------------

##----- Ação 2.1
# sAVE BEFORE THE LOAD
save.image("~/Ciclystic_data.RData")

# Load the R Environment into R Markdown Environment
# to use all datasets created them
data_rm <- load("ciclystic_data.Rdata")

##----- Ação 3
# Calculate a number of rides by each member users and rideable type
rides_num3m <- ciclystic_3mc %>%
  count(member_casual, rideable_type)

# Calculate a number of rides by each member users and rideable type
rides_num6m <- ciclystic_6mc %>%
  count(member_casual, rideable_type)

# rename the column "n" and re-create the same dataframe with new column name
rides_num6m$num_rides <- rides_num6m$n
rides_num6m <- rides_num6m %>%
  select(member_casual,rideable_type, num_rides)
#-------------------------------------------------------------------------------
# PLOTS
##------ Ação 11-3
# Show graphics for Type of Rides and Members
plot_num_rides3m <- ggplot(ciclystic_3mc) +
  geom_bar(mapping = aes(x = member_casual)) +
  #  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Numbers of rides per Members") +
  labs(x = "Members and Casual Users", y = "Number of Rides")
```

```
# ------- Ação 11-6
plot_num_rides6m <- ggplot(ciclystic_6mc) +
  geom_bar(mapping = aes(x = member_casual)) +
  labs(title = "Ciclystic Company", subtitle = "Numbers of rides per Members") +
  labs(x = "Members and Casual Users", y = "Number of Rides") +
  scale_y_continuous(labels = scales::comma)

#-------------------------------------------------------------------------------
#-------- Ação 11.1
# Show graphics for Type of Rides and Members and Rideables Types
plot_num_rides13m <- ggplot(ciclystic_3mc) +
  geom_bar(mapping = aes(x = member_casual)) +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Numbers of rides per Members and Rideable
Types") +
  labs(x = "Members and Casual Users", y = "Number of Rides")
# Show graphics for Type of Rides and Members and Rideables Types
plot_num_rides16m <- ggplot(ciclystic_6mc) +
  geom_bar(mapping = aes(x = member_casual)) +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Numbers of rides per Members and Rideable
Types") +
  labs(x = "Members and Casual Users", y = "Number of Rides")
#-------------------------------------------------------------------------------

##------ Ação 3.0
# Calcula tempo de médio de uso da Bike por Member Type - 3 months
ride_avg_member3m <- ciclystic_3mc %>%
  group_by(member_casual) %>%
  summarize(ride_avg = mean(ride_min), Total_ride = ride_avg_member3m$ride_avg[1] +
ride_avg_member3m$ride_avg[2])

# Calcula tempo de médio de uso da Bike por Member Type - 6 months
ride_avg_member6m <- ciclystic_6mc %>%
  group_by(member_casual) %>%
#  summarize(ride_avg = mean(ride_min), Total_ride = ride_avg_member6m$ride_avg[1] +
ride_avg_member6m$ride_avg[2])
  summarize(ride_avg = mean(ride_min))

# to show the time in minutes
ciclystic_6mc$ride_min <- ciclystic_6mc$ride_min * 100
#------- PLOTS
##------ Ação 11.0
# Gráfico com Average Time Member Type - 3 Months
plot_ride_avg_member3m <- ggplot(ride_avg_member3m, aes(member_casual, ride_avg)) +
  geom_bar(stat = "identity", position = 'dodge') +
  labs(title = "Ciclystic Company", subtitle = "Ride Time per Member") +
  labs(x = "Member x Casual Users", y = "Ride Time Average (min)")
#  annotate("text", x = 1, y = 2, label = c("39332","42936"))     ,"108856","110167"))
# Gráfico com Average Time Member - Type 6 months
plot_ride_avg_member6m <- ggplot(ride_avg_member6m, aes(member_casual, ride_avg)) +
  geom_bar(stat = "identity", position = 'dodge') +
  labs(title = "Ciclystic Company", subtitle = "Ride Time per Member") +
  labs(x = "Member x Casual Users", y = "Ride Time Average (min)")


##----- Ação 3.1
# Calcula tempo de médio de uso da Bike por Member Type - 3 Months
ride_avg_member13 <- ciclystic_3mc %>%
  group_by(member_casual, rideable_type) %>%
  summarize(ride_avg = mean(ride_min))
# Calcula tempo de médio de uso da Bike por Member Type - 6 MOnths
ride_avg_member16 <- ciclystic_6mc %>%
  group_by(member_casual, rideable_type) %>%
  summarize(ride_avg = mean(ride_min))

##------ Ação 11.1
# Gráfico com Average Time Member by Rideable Type - 3 Months
plot_ride_avg_member13 <- ggplot(ride_avg_member13, aes(member_casual, ride_avg, fill =
```

```
rideable_type)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Ride Time per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Time Average (min)")

# Gráfico com Average Time Member by Rideable Type - 6 Months
plot_ride_avg_member16 <- ggplot(ride_avg_member16, aes(member_casual, ride_avg, fill =
rideable_type)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Ride Time per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Time Average (min)")
#-------------------------------------------------------------------------------


##----- Ação 4
# calcula a distância média percorrida por Member Type - 3 Months
ride_dist_member3 <- ciclystic_3mc %>%
  group_by(member_casual) %>%
  summarise(avg_dist = mean(ride_distance))

# calcula a distância média percorrida por Member Type - 6 Months
ride_dist_member6 <- ciclystic_6mc %>%
  group_by(member_casual) %>%
  summarise(avg_dist = mean(ride_distance))

#-------- Ação 15
# Group Member and Rideable Type with both information - Ride Time and Ride Distance - 3 Months
rides_avg_mix <- ciclystic_3mc %>%
  group_by(member_casual, rideable_type) %>%
  summarize(avg_time = mean(ride_min), avg_dist = mean(ride_distance))

# Group Member and Rideable Type with both information - Ride Time and Ride Distance - 6 MOnths
rides_avg_mix6 <- ciclystic_6mc %>%
  group_by(member_casual, rideable_type) %>%
  summarize(avg_time = mean(ride_min), avg_dist = mean(ride_distance))
#-------------------------------------------------------------------------------

##----- Ação 4.1-3
# calcula a distância média percorrida por Member Type and Rideable Type
ride_dist_member1 <- ciclystic_3mc %>%
  group_by(member_casual) %>%
  summarise(avg_dist = mean(ride_distance))
# calcula a distância média percorrida por Member Type and Rideable Type
ride_dist_member13 <- ciclystic_3mc %>%
  group_by(member_casual, rideable_type) %>%
  summarise(avg_dist = mean(ride_distance))

##----- Ação 4.1-6
# calcula a distância média percorrida por Member Type and Rideable Type
ride_dist_member6 <- ciclystic_6mc %>%
  group_by(member_casual) %>%
  summarise(avg_dist = mean(ride_distance))
# calcula a distância média percorrida por Member Type and Rideable Type
ride_dist_member16 <- ciclystic_6mc %>%
  group_by(member_casual, rideable_type) %>%
  summarise(avg_dist = mean(ride_distance))


##------Ação 13-3
# Gráfico com Average Distance Member by Rideable Type
plot_ride_dist_member1 <- ggplot(ride_dist_member1, aes(member_casual, avg_dist)) +
  geom_bar(stat = "identity", position = 'dodge') +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Distance Average (meters)")

##------ Ação 13.1-3
# Gráfico com Average Distance Member by Rideable Type
```

```
plot_ride_dist_member13 <- ggplot(ride_dist_member13, aes(member_casual, avg_dist, fill =
rideable_type)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Distance Average (meters)")
#----------------
##------Ação 13-6
# Gráfico com Average Distance Member by Rideable Type
plot_ride_dist_member6 <- ggplot(ride_dist_member6, aes(member_casual, avg_dist)) +
  geom_bar(stat = "identity", position = 'dodge') +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Distance Average (meters)")

##------ Ação 13.1-6
# Gráfico com Average Distance Member by Rideable Type
plot_ride_dist_member16 <- ggplot(ride_dist_member16, aes(member_casual, avg_dist, fill =
rideable_type)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Distance Average (meters)")

#-------------------------------------------------------------------------
##----- Ação 5 - 3 months
# plot sem mapa apenas com geo localização com pontos Member Type por Rideable Type
plot_gen_rides_point3 <- ggplot(ciclystic_3m) +
  geom_point(data = ciclystic_3m, aes(x = start_lng, y = start_lat, color = member_casual)) +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Geolocation Rides per Member and Rideable
Type") +
  labs(x = "Longitude", y = "Latitude")

# Plot inverted Member Type Only
# plot sem mapa apenas com geo localização com pontos Rideable Type per Member Type
plot_gen_rides_point3i <- ggplot(ciclystic_3m) +
#  geom_point(data = ciclystic_3m, aes(x = start_lng, y = start_lat, color = rideable_type)) +
  geom_point(data = ciclystic_3m, aes(x = start_lng, y = start_lat)) +
  facet_wrap(~member_casual) +
  labs(title = "Ciclystic Company", subtitle = "Geolocation Rides per Member Type") +
  labs(x = "Longitude", y = "Latitude")

# Plot inverted Rideable Type Only
# plot sem mapa apenas com geo localização com pontos Rideable Type per Member Type
plot_gen_rides_point3r <- ggplot(ciclystic_3m) +
  #  geom_point(data = ciclystic_3m, aes(x = start_lng, y = start_lat, color = rideable_type))
+
  geom_point(data = ciclystic_3m, aes(x = start_lng, y = start_lat)) +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Geolocation Rides per Rideable Type") +
  labs(x = "Longitude", y = "Latitude")

##----- Ação 5 - 6 months
# plot sem mapa apenas com geo localização com pontos Member Type por Rideable Type
plot_gen_rides_point6 <- ggplot(ciclystic_6m) +
  geom_point(data = ciclystic_6m, aes(x = start_lng, y = start_lat, color = member_casual)) +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Geolocation Rides per Member and Rideable
Type") +
  labs(x = "Longitude", y = "Latitude")

# Plot inverted Member Type Only
# plot sem mapa apenas com geo localização com pontos Rideable Type per Member Type
plot_gen_rides_point6i <- ggplot(ciclystic_6m) +
  #  geom_point(data = ciclystic_6m, aes(x = start_lng, y = start_lat, color = rideable_type))
+
  geom_point(data = ciclystic_6m, aes(x = start_lng, y = start_lat, color = rideable_type)) +
  facet_wrap(~member_casual) +
  labs(title = "Ciclystic Company", subtitle = "Geolocation Rides per Member Type") +
```

```
  labs(x = "Longitude", y = "Latitude")

# Plot inverted Rideable Type Only
# plot sem mapa apenas com geo localização com pontos Rideable Type per Member Type
plot_gen_rides_point6r <- ggplot(ciclystic_6m) +
  #  geom_point(data = ciclystic_6m, aes(x = start_lng, y = start_lat, color = rideable_type))
+
  geom_point(data = ciclystic_6m, aes(x = start_lng, y = start_lat, color = member_casual)) +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Geolocation Rides per Rideable Type") +
  labs(x = "Longitude", y = "Latitude")
#-------------------------------------------------------------------------------

##----- Ação 6 - 3 months
# Calcula tempo de médio de uso da Bike por Ride Type
ride_avg_type3 <- ciclystic_3mc %>%
  group_by(rideable_type) %>%
  summarize(ride_avg = mean(ride_min))

##----- Ação 6.1
# Calcula tempo de médio de uso da Bike por Ride Type and Member Type
ride_avg_type13 <- ciclystic_3mc %>%
  group_by(rideable_type, member_casual) %>%
  summarize(ride_avg = mean(ride_min))

##------ Ação 12
# Gráfico com Average Time Rideable Type by Member
plot_ride_avg_type1 <- ggplot(ride_avg_type1, aes(rideable_type, ride_avg, fill =
member_casual)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~member_casual) +
  labs(title = "Ciclystic Company", subtitle = "Ride Time per Rideable Type and Members") +
  labs(x = "RIdeable Type Bikes", y = "Ride Time Average (min)")

##----- Ação 6 - 6 months
# Calcula tempo de médio de uso da Bike por Ride Type
ride_avg_type6 <- ciclystic_6mc %>%
  group_by(rideable_type) %>%
  summarize(ride_avg = mean(ride_min))

##----- Ação 6.1
# Calcula tempo de médio de uso da Bike por Ride Type and Member Type
ride_avg_type16 <- ciclystic_6mc %>%
  group_by(rideable_type, member_casual) %>%
  summarize(ride_avg = mean(ride_min))

##------ Ação 12
# Gráfico com Average Time Rideable Type by Member
plot_ride_avg_type16 <- ggplot(ride_avg_type16, aes(rideable_type, ride_avg, fill =
member_casual)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~member_casual) +
  labs(title = "Ciclystic Company", subtitle = "Ride Time per Rideable Type and Members") +
  labs(x = "RIdeable Type Bikes", y = "Ride Time Average (min)")

#-------------------------------------------------------------------------------
##----- Ação 7 - 3 months
# calcula a distância média percorrida por Ride Type
ride_dist_type3 <- ciclystic_3mc %>%
  group_by(rideable_type) %>%
  summarise(avg_dist = mean(ride_distance))

##----- Ação 7.1
# calcula a distância média percorrida por Ride Type
ride_dist_type13 <- ciclystic_3mc %>%
  group_by(rideable_type, member_casual) %>%
  summarise(avg_dist = mean(ride_distance))

# Gráfico com Average Distance Rideable Type by Member
```

```
plot_ride_dist_type13 <- ggplot(ride_dist_type13, aes(rideable_type, avg_dist, fill =
member_casual)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~member_casual) +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Rideable Type and Members") +
  labs(x = "RIdeable Type Bikes", y = "Ride Distance Average (meters)")

##----- Ação 7 - 6 months
# calcula a distância média percorrida por Ride Type
ride_dist_type6 <- ciclystic_6mc %>%
  group_by(rideable_type) %>%
  summarise(avg_dist = mean(ride_distance))

##----- Ação 7.1
# calcula a distância média percorrida por Ride Type
ride_dist_type16 <- ciclystic_6mc %>%
  group_by(rideable_type, member_casual) %>%
  summarise(avg_dist = mean(ride_distance))

# Gráfico com Average Distance Rideable Type by Member
plot_ride_dist_type16 <- ggplot(ride_dist_type16, aes(rideable_type, avg_dist, fill =
member_casual)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~member_casual) +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Rideable Type and Members") +
  labs(x = "RIdeable Type Bikes", y = "Ride Distance Average (meters)")
#-------------------------------------------------------------------------------
##----- Ação 8
# Gráfico de numerosde RIDES por RIDEABLE TYPE
ciclystic_new %>%
  ggplot(mapping = aes(x = rideable_type)) +
  geom_bar()

##----- Ação 9
# Grafico com numero de RIDES por MEMBER-CASUAL
ciclystic_new %>%
  ggplot(mapping = aes(x = member_casual)) +
  geom_bar()

##------ Ação 10
# Calcula quantidade de viagens por cada member and rideable type
rides_num <- ciclystic_new %>%
  count(member_casual, rideable_type)


# Gráficos para insight --- novas ações a incluir
# ---------------------------------------------------------------




##------ Ação 11.0
# Gráfico com Average Time Member Type
plot_ride_avg_member3m <- ggplot(ride_avg_member3m, aes(member_casual, ride_avg)) +
  geom_bar(stat = "identity", position = 'dodge') +
  labs(title = "Ciclystic Company", subtitle = "Ride Time per Member") +
  labs(x = "Member x Casual Users", y = "Ride Time Average (min)")
#  annotate("text", x = 1, y = 2, label = c("39332","42936"))     ,"108856","110167"))

##------ Ação 11.1
# Gráfico com Average Time Member by Rideable Type
plot_ride_avg_member13 <- ggplot(ride_avg_member13, aes(member_casual, ride_avg, fill =
rideable_type)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Ride Time per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Time Average (min)")
```

```
##------Ação 13
# Gráfico com Average Distance Member by Rideable Type
plot_ride_dist_member1 <- ggplot(ride_dist_member1, aes(member_casual, avg_dist)) +
  geom_bar(stat = "identity", position = 'dodge') +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Distance Average (meters)")

##------ Ação 13.1
# Gráfico com Average Distance Member by Rideable Type
plot_ride_dist_member13 <- ggplot(ride_dist_member13, aes(member_casual, avg_dist, fill =
rideable_type)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Member and Rideable Type") +
  labs(x = "Member x Casual Users", y = "Ride Distance Average (meters)")

# Gráfico com Average Distance Rideable Type by Member
plot_ride_dist_type13 <- ggplot(ride_dist_type13, aes(rideable_type, avg_dist, fill =
member_casual)) +
  geom_bar(stat = "identity", position = 'dodge') +
  facet_wrap(~member_casual) +
  labs(title = "Ciclystic Company", subtitle = "Ride Distance per Rideable Type and Members") +
  labs(x = "RIdeable Type Bikes", y = "Ride Distance Average (meters)")

#--------- Ação 14
#-------------------------------------------------------------------------------
# Create a dataset with records about rides on weekend
cicly_weekends <- ciclystic_3mc %>%
  filter(weekday == "sábado" | weekday == "domingo")

# Create a dataset with records about rides on weekdays
cicly_weekdays <- ciclystic_3mc %>%
  filter(weekday != "sábado" & weekday != "domingo")
#-------------------------------------------------------------------------------
# Create a dataset with records about rides on weekend
cicly_weekends <- ciclystic_6mc %>%
  filter(weekday == "sábado" | weekday == "domingo")

# Create a dataset with records about rides on weekdays
cicly_weekdays <- ciclystic_6mc %>%
  filter(weekday != "sábado" & weekday != "domingo")
#-------------------------------------------------------------------------------

# How many rides Members & Casuals are doing on Weekdays per Bike Types
rides_weekday <- cicly_weekdays %>%
  count(member_casual, rideable_type)

# Show the plot regarding this tibble above
plot_rides_weekday <- ggplot(cicly_weekdays) +
  geom_bar(aes(x = member_casual, fill = member_casual)) +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Number of Rides per Rideable Type and Members
on Weekdays") +
  labs(x = "Members and Casuals", y = "Number of Rides") +
  scale_y_continuous(labels = scales::comma)

# How many rides Members & Casuals are doing on Weekends per Bike Types
rides_weekend <- cicly_weekends %>%
  count(member_casual, rideable_type)

# Show the plot regarding this tibble above
plot_rides_weekend <- ggplot(cicly_weekends) +
  geom_bar(aes(x = member_casual, fill = member_casual)) +
  facet_wrap(~rideable_type) +
  labs(title = "Ciclystic Company", subtitle = "Number of Rides per Rideable Type and Members
on Weekends") +
  labs(x = "Members and Casuals", y = "Number of Rides")
```

```
#-------------------------------------------------------------------------------
#-------- Ação 15
# Group Member and Rideable Type with both information - Ride Time and Ride Distance
rides_avg_mix <- ciclystic_3mc %>%
  group_by(member_casual, rideable_type) %>%
  summarize(avg_time = mean(ride_min), avg_dist = mean(ride_distance))

# LER NOVAMENTE O DESCRITIVO DO PROJETO
# COLOCAR NOVAS IDEIAS E OBJETIVOS
# -------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
# Calcula média de tempo de uso de Casual and Member
cicly_cleaned <- ciclystic %>%
  drop_na() %>%
  group_by(member_casual) %>%
  summarize(dta_time = mean(ymd_hms(ended_at)-ymd_hms(started_at)))


install.packages("maps")
# criação de mapa para geo localização dos pontos
USA_map <- map_data("world", region = "USA")

# plot com mapa - mas não ficou bom
ggplot() +
  geom_map(map = USA_map, data = ciclystic_new, aes(ciclystic_new$start_lng,
ciclystic_new$start_lat, map_id = region), color = "gray80", fill = "gray80", size = 0.3) +
  geom_point(data = ciclystic_new, aes(x = start_lng, y = start_lat))



## ---------------------------------------------------------------------

# Cria um mapa de qq cidade USA
map('county', 'Illinois', fill = TRUE, col = palette())

#Variáveis utilizadas no cálculo da distância entre Coordenadas
# variável fixa da esfera terrestre
r1 <- 6371
# variável fixa de PI
p1 <- (3.141592653/180)
# Inclusão do campo RIDE DISTANCE (distancia entre as coordenadas LAT e LONG)
# na database Ciclystic_Cleaned
ciclystic_clean <- ciclystic %>%
  mutate(ciclystic, ride_distance = r1 * (2 * asin(pmin(1, sqrt(((sin(((ciclystic$end_lat -
ciclystic$start_lat) * p1)/2)^2 + cos(ciclystic$start_lat * p1)) * cos(ciclystic$end_lat * p1)
* sin(((ciclystic$end_lng - ciclystic$start_lng) * p1)/2)^2))))))

## Funciona como mutate, cria variável que recebe calculo e guarda no dataset
ciclystic_new$z <- with(ciclystic_new,
sqrt((ciclystic_new$start_lng^2+ciclystic_new$start_lat^2)))



## ---------------------------------------------------------------------

#Mesma linha de código a qual não funciona por motivos de alguma limitação do R
ciclystic_clean <- ciclystic %>%
  # cálculos de seno e coseno sobre as variáveis de longitude e latitude
  dif_long <- ((ciclystic$end_lng - ciclystic$start_lng) * p1) + dif_lat <- ((ciclystic$end_lat
- ciclystic$start_lat) * p1) %>%
  r2 <- ((sin(dif_lat/2)^2 + cos(ciclystic$start_lat * p1)) * cos(ciclystic$end_lat * p1) *
sin(dif_long/2)^2) %>%
  r3 <- (2 * asin(pmin(1, sqrt(r2)))) %>%
  mutate(ciclystic, ride_distance = r1 * r3)

## Linhas de cálculo para determinar a distancia em km das coordenadas
r1 <- 6371 # variável fixa da esfera terrestre
p1 <- (3.141/180) # variável fixa de PI
long1 <- -87.67423
```

```
lat1 <- 41.96845
long2 <- -87.70805
lat2 <- 41.96669
# cálculos de seno e coseno sobre as variáveis de longitude e latitude
dif_long <- ((long2 - long1) * p1)
dif_lat <- ((lat2 - lat1) * p1)
a1 <- sin(dif_lat/2)^2 + cos(lat1 * p1) * cos(lat2 * p1) * sin(dif_long/2)^2
b1 <- 2 * asin(pmin(1, sqrt(a1)))
d1 <- r1 * b1
```