

The cover features a teal background on the left side, which transitions into a white area on the right. A large, light blue circle is positioned in the lower-left quadrant, partially overlapping the teal area. The title is centered in the white area.

# **MINIPROYECTO VISUALIZACION DE DATOS**

Pablo Sanjuan, Alex Llorens,  
Jose María Camarasa

# Índice

<b>1. Importación Qgis</b>	<b>3</b>
<b>2. RStudio</b>	<b>5</b>
<b>2.1. Librerías</b>	<b>5</b>
<b>2.2 Código</b>	<b>6</b>
<b>3. Shiny</b>	<b>9</b>
<b>4. Interfaz Shiny</b>	<b>12</b>

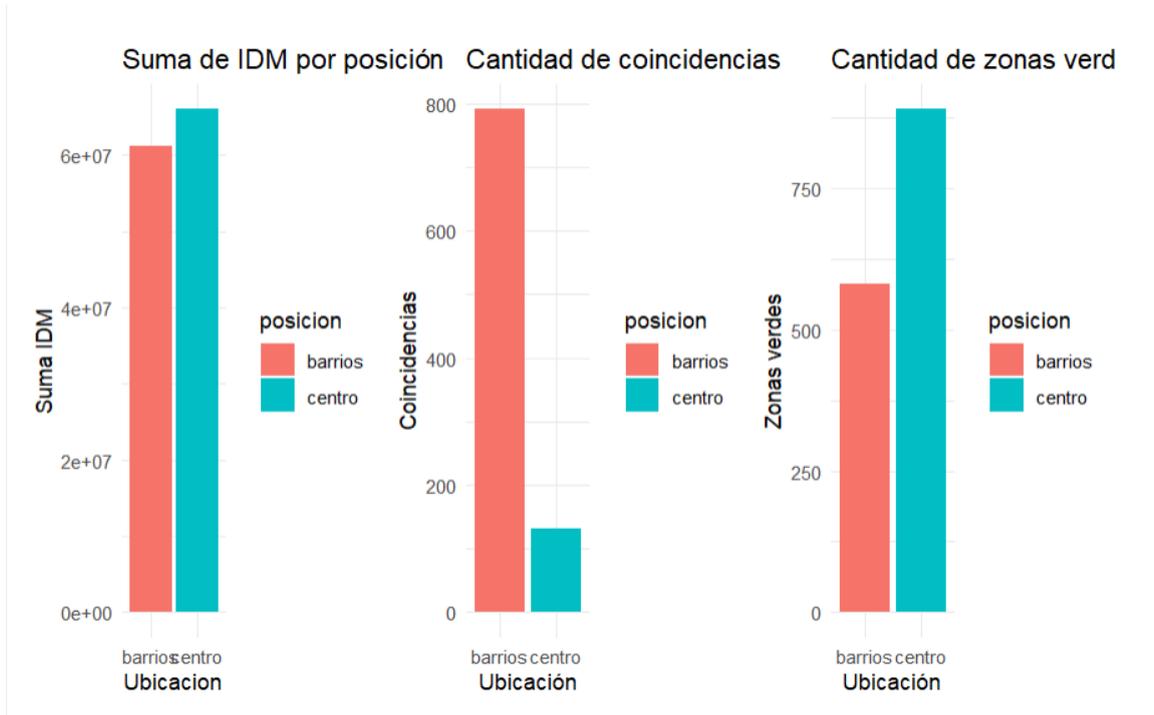
# 1. Importación Qgis

Iniciamos el proyecto importando los datos sobre las zonas verdes y la situación del tráfico y movilidad en Valencia en QGIS. Al analizar los datos, surgió la idea de explorar la coincidencia entre las zonas verdes y los puntos de congestión. Realizamos una operación de intersección para visualizar esta correlación, obteniendo como resultado un mapa que muestra tanto las áreas verdes como los puntos de tráfico, así como las zonas donde ambas características se superponen.

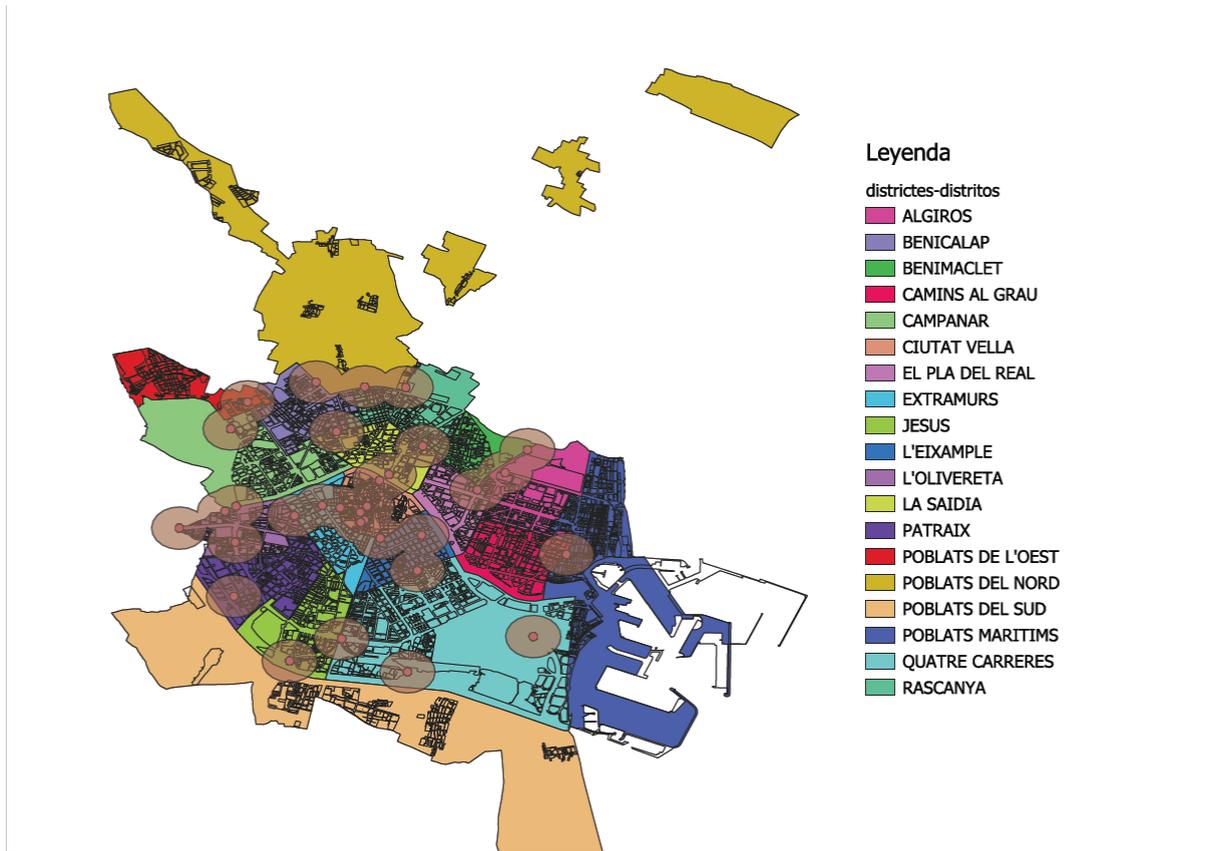


Lo que observamos en este mapa es que hay mucho tráfico en el centro y en algunos barrios de alrededor. También observamos que hay muchas más zonas verdes en el centro

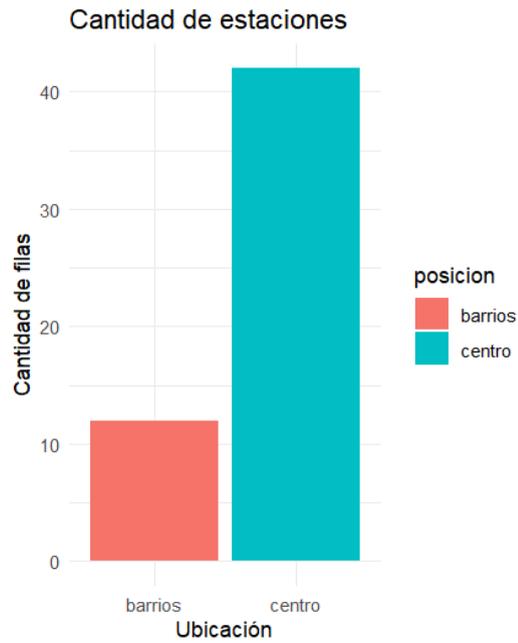
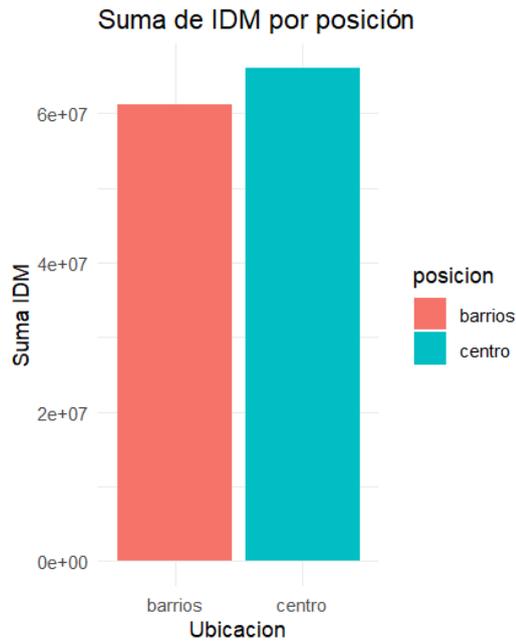
de la ciudad que en los barrios externos, en cambio observamos más coincidencias en los barrios. En los gráficos siguientes podemos confirmar nuestra certeza anterior.



Como podemos observar las zonas de los barrios hay ausencia de estaciones de bicis. Aunque el tráfico se concentra en el centro y en los barrios sin saber donde hay más, en cambio los puntos de bicis solo se encuentran en gran abundancia en el centro. El gráfico confirma nuestro razonamiento.



Después, podemos crear otro mapa para identificar las áreas con mayor tráfico y determinar si sería conveniente ubicar estaciones de bicicletas en esas zonas. Para hacer esto, primero importamos los datos de movilidad y los ajustamos a la misma proyección para poder delinear un área de influencia. Luego, superponemos una capa que muestra los distritos de Valencia para tener una referencia clara. Seguidamente, agregamos los puntos que indican la ubicación de las estaciones de bicicletas.



## 2. RStudio

### 2.1. Librerías

```

{r}
library(readr)
library(dplyr)
library(sf)
library(rgdal)
library(sp)
library(ggplot2)
library(dplyr)
library(gridExtra)
library(RColorBrewer)
library(leaflet)
library(tidyr)

```

### 2.2 Código

Primero, en R importamos los datos desde el archivo CSV y eliminamos las entradas vacías (NA). Para lograr esto, convertimos los datos de tipo carácter que deberían ser numéricos a su formato correspondiente y asignamos un valor de 0 a los NA. También, eliminamos la columna "fecha.baja" de aquellos dataframes que la contienen, ya que no aporta

información relevante. Por último, eliminamos las filas que aún contienen NA para evitar que interfirieran en nuestro análisis; sin embargo, estas son escasas.

```

library(readr)
estaciones_contaminacion <- read_csv2("datos/estacions-contaminacio-atmosferiques-estaciones-contaminacion-atmosfericas.csv")

rvvcca <- read_csv2("datos/rvvcca.csv")

bicis <- read_csv2("datos_compl/punts-mesura-bicis-espires-electromagnetiques-puntos-medida-bicis-espiras-electr.csv")
estaciones_trafico <- read_csv2("datos_compl/punts-mesura-traffic-espires-electromagnetiques-puntos-medida-traffic-espiras-ele.csv")

trafico <- read_csv2("datos_compl/qlik_datos_imd_mobilitat_enero_2016-diciembre-2022_coord.csv")
zonas_verdes <- read_csv2("datos_compl/zonas-verdes.csv")
...

```

```

library(dplyr)
#Transformacion de caracter a numerico
estaciones_contaminacion <- estaciones_contaminacion %>% mutate(co = as.numeric(co),no2 = as.numeric(no2))

rvvcca <- rvvcca %>% mutate_at(vars(6:32), as.numeric)

zonas_verdes <- zonas_verdes %>% mutate_at(vars(6:17), as.numeric)

#Eliminacion de fecha baja que siempre es NA
rvvcca <- rvvcca[, -34]
trafico <- trafico[, -13]
...

```

```

#Eliminacion de NA
estaciones_contaminacion <- data.frame(lapply(estaciones_contaminacion, function(x) ifelse(is.numeric(x) & is.na(x), 0, x)))

rvvcca <- data.frame(lapply(rvvcca, function(x) ifelse(is.numeric(x) & is.na(x), 0, x)))

estaciones_trafico <- data.frame(lapply(estaciones_trafico, function(x) ifelse(is.numeric(x) & is.na(x), 0, x)))
estaciones_trafico <- estaciones_trafico[complete.cases(estaciones_trafico), ]

bicis <- data.frame(lapply(bicis, function(x) ifelse(is.numeric(x) & is.na(x), 0, x)))
bicis <- bicis[complete.cases(bicis), ]

trafico <- data.frame(lapply(trafico, function(x) ifelse(is.numeric(x) & is.na(x), 0, x)))
trafico <- trafico[complete.cases(trafico), ]
trafico$IMD <- gsub(" ", "", trafico$IMD)
trafico$IMD <- as.numeric(trafico$IMD)

zonas_verdes <- data.frame(lapply(zonas_verdes, function(x) ifelse(is.numeric(x) & is.na(x), 0, x)))
zonas_verdes <- zonas_verdes[complete.cases(zonas_verdes), ]
...

```

Luego cargamos los mapas de qgis. Eliminamos también de la capa valencia los datos que no nos sirven.

```

buffer <- readOGR(dsn = "qgisminiproyecto/buffer500.shp",)
interseccion <- read_sf("qgisminiproyecto/interseccion.shp")
capa_zonas_verdes <- readOGR(dsn = "datos_compl/zonas-verdes/zonas-verdes.shp", )
capa_trafico <- readOGR(dsn = "datos_compl/qlik_datos_imd_mobilitat_enero_2016-diciembre-2022_coord/qlik_datos_imd_mobilitat_enero_2016-diciembre-2022_coord.shp", )
capa_bicis <- readOGR(dsn = "datos_compl/punts-mesura-bicis-espires-electromagnetiques-puntos-medida-bicis-espiras-electr/punts-mesura-bicis-espires-electromagnetiques-puntos-medida-bicis-espiras-electr.shp", )
capa_valencia=readOGR(dsn = "datos_compl/ayu_Val_manzanas/MANZANAS_POB.shp", )
capa_valencia<- capa_valencia[, c(-2,-3,-4,-5)]
...

```

Convertimos tanto la capa de Valencia como la capa de buffer de objetos espaciales a un formato de datos espaciales para que puedan ser integrados fácilmente con las demás capas. Además, las ajustamos a la misma proyección que las otras capas para poder visualizarlas conjuntamente en el gráfico.

```

{r}
capa_valencia <- st_as_sf(capa_valencia)
capa_valencia <- st_transform(capa_valencia, st_crs(capa_trafico))

```

```

{r}
buffer <- st_as_sf(buffer)
buffer <- st_transform(buffer, st_crs(capa_trafico))

```

Finalmente, para generar los gráficos, utilizamos tanto los datos del archivo .csv como la capa de intersección. Previamente, actualizamos las columnas de longitud y latitud del archivo .csv, convirtiéndolas de formato carácter a numérico, para garantizar su correcta representación.

```

for(i in 1:nrow(trafico)) {
  lat_lon <- trafico$Geo.Point[i]
  lat_lon_split <- strsplit(lat_lon, ",")
  latitud <- as.numeric(trimws(lat_lon_split[[1]][1]))
  longitud <- as.numeric(trimws(lat_lon_split[[1]][2]))
  trafico$Latitud[i] <- latitud
  trafico$Longitud[i] <- longitud
}

for(i in 1:nrow(bicis)) {
  lat_lon <- bicis$geo_point_2d[i]
  lat_lon_split <- strsplit(lat_lon, ",")
  latitud <- as.numeric(trimws(lat_lon_split[[1]][1]))
  longitud <- as.numeric(trimws(lat_lon_split[[1]][2]))
  bicis$Latitud[i] <- latitud
  bicis$Longitud[i] <- longitud
}

for(i in 1:nrow(zonas_verdes)) {
  lat_lon <- zonas_verdes$Geo.Point[i]
  lat_lon_split <- strsplit(lat_lon, ",")
  latitud <- as.numeric(trimws(lat_lon_split[[1]][1]))
  longitud <- as.numeric(trimws(lat_lon_split[[1]][2]))
  zonas_verdes$Latitud[i] <- latitud
  zonas_verdes$Longitud[i] <- longitud
}

interseccion$latitud=as.numeric(interseccion$latitud)
interseccion$longitud=as.numeric(interseccion$longitud)

```

Posteriormente creamos otra nueva columna dónde ubicar los datos en el centro de valencia o en las partes de los barrios.

```

[[r]]
trafico$posicion=NA
trafico$posicion <- ifelse(trafico$Latitud < 39.49 & trafico$Latitud > 39.44 & trafico$Longitud > -0.41 & trafico$Longitud <
-0.35, 'centro', trafico$posicion )
trafico$posicion<-ifelse(trafico$Latitud > 39.49 | trafico$Latitud < 39.44 | trafico$Longitud < -0.41 | trafico$Longitud >
-0.35, 'barrios', trafico$posicion )

bicis$posicion=NA
bicis$posicion <- ifelse(bicis$Latitud < 39.49 & bicis$Latitud > 39.44 & bicis$Longitud > -0.41 & bicis$Longitud <
-0.35, 'centro', bicis$posicion )
bicis$posicion<-ifelse(bicis$Latitud > 39.49 | bicis$Latitud < 39.44 | bicis$Longitud < -0.41 | bicis$Longitud >
-0.35, 'barrios', bicis$posicion )

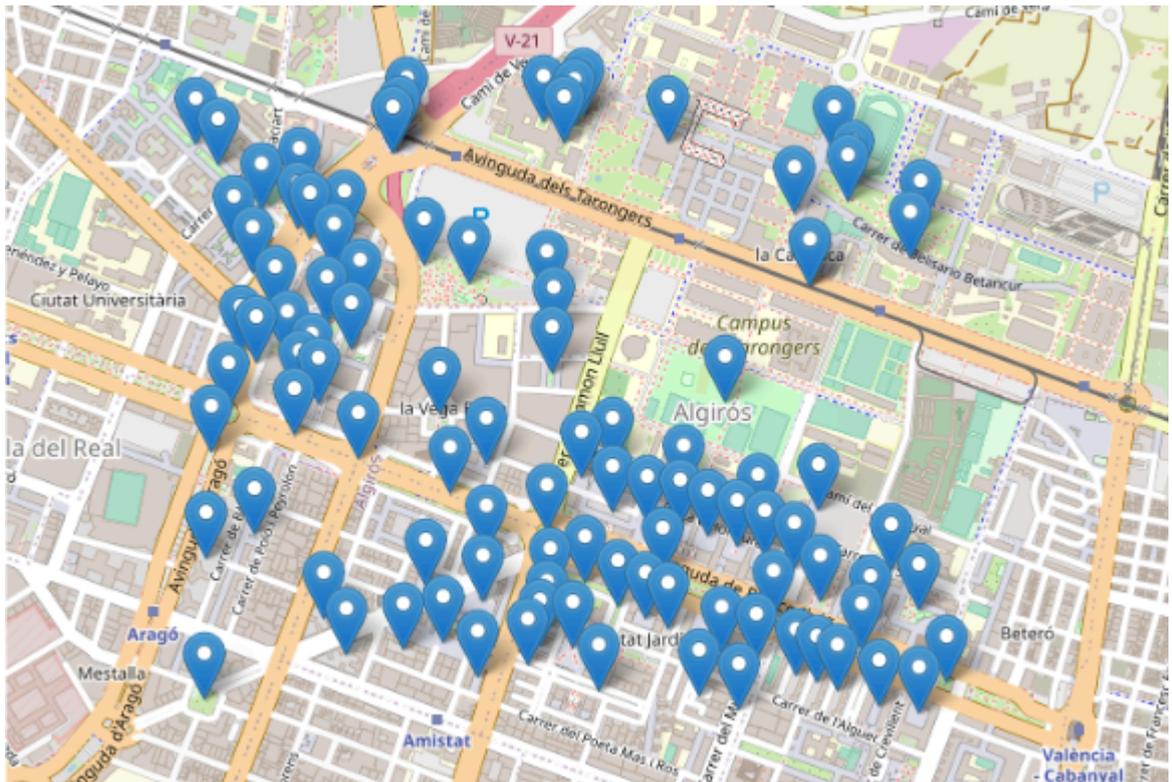
zonas_verdes$posicion=NA
zonas_verdes$posicion <- ifelse(zonas_verdes$Latitud < 39.49 & zonas_verdes$Latitud > 39.44 & zonas_verdes$Longitud > -0.41
& zonas_verdes$Longitud < -0.35, 'centro', zonas_verdes$posicion )
zonas_verdes$posicion<-ifelse(zonas_verdes$Latitud > 39.49 | zonas_verdes$Latitud < 39.44 | zonas_verdes$Longitud < -0.41 |
zonas_verdes$Longitud > -0.35, 'barrios', zonas_verdes$posicion)

interseccion$posicion=NA
interseccion$posicion <- ifelse(interseccion$latitud < 39.49 & interseccion$latitud > 39.44 & interseccion$longitud > -0.41
& interseccion$longitud < -0.35, 'centro', interseccion$posicion )
interseccion$posicion<-ifelse(interseccion$latitud > 39.49 | interseccion$latitud < 39.44 | interseccion$longitud < -0.41 |
interseccion$longitud > -0.35, 'barrios', zonas_verdes$posicion)

```

### 3. Shiny

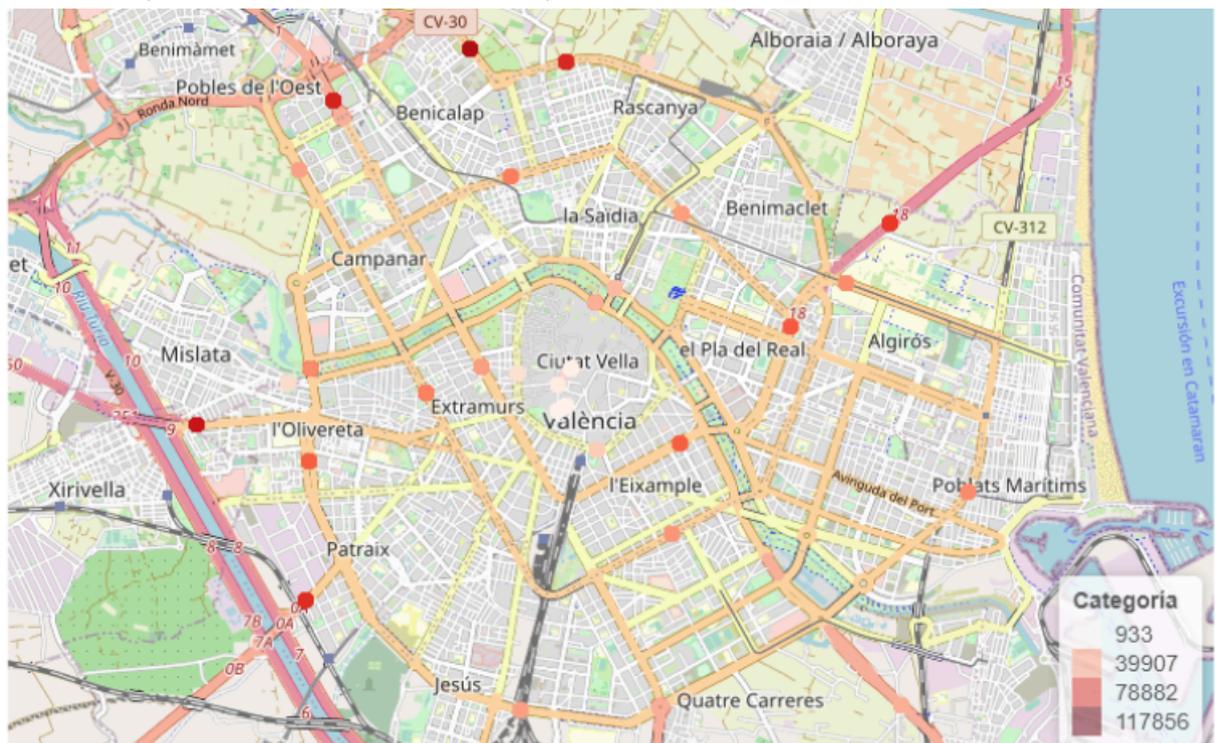
En esta parte hemos decidido hacer 2 mapas interactivos. Uno para saber si hay más zonas verdes o red viarias en la zona universitaria. Y otro mapa para ver donde hay más concentración de tráfico.



En este primer mapa, notamos una densa concentración de marcadores en el área de Tarongers, representando áreas verdes. Al mismo tiempo, al consultar el tercer mapa, donde se muestran los puntos de muestreo junto con los datos de tráfico de esa región, confirmamos lo observado anteriormente a través de los histogramas. Se evidencia que los barrios periféricos al centro muestran una mayor correspondencia entre zonas verdes y tráfico intenso.



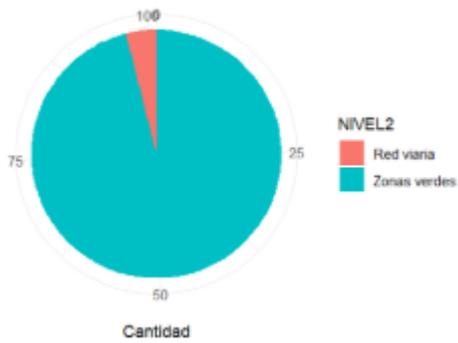
En nuestra área seleccionada, la avenida de Tarongers, observamos que los dos puntos de muestreo se encuentran dentro de los dos niveles más altos de tráfico, tal como se detalla en la leyenda del mapa. Hemos utilizado una escala de colores para representar la cantidad de tráfico percibida en cada punto de muestreo, donde los tonos de rojo más intensos indican un mayor nivel de tráfico en la zona.



Viendo así que en Ciutat Vella no hay casi tráfico ya que está muy bien establecida para los peatones, y con difícil accesibilidad para los transportes.

También nos ha parecido útil hacer un conteo de los distintos tipos de zonas (verdes, viarias). Hemos hecho un gráfico representando esto en la zona universitaria.

Tipos de zonas en la zona universitaria

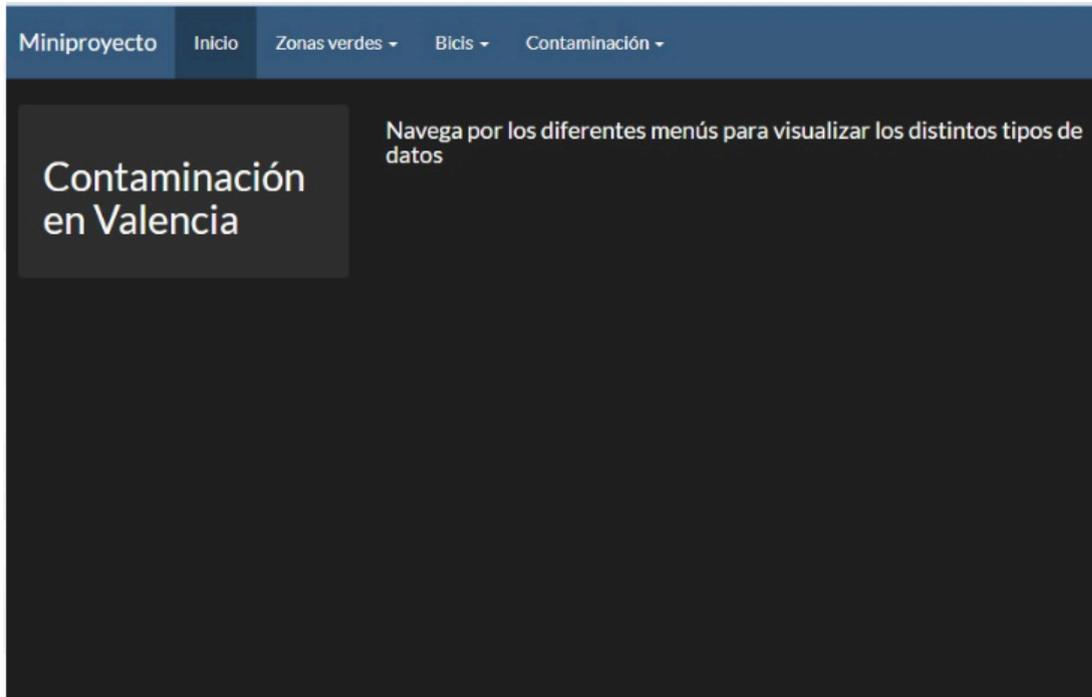


En el siguiente histograma, hemos continuado refinando la ubicación de las zonas verdes. Ahora, cada barra representa una de las divisiones definidas en el mapa de Valencia, las cuales están almacenadas en la variable "Tipo.Dato" del dataframe de tráfico que se creó previamente. Hemos asignado un color diferente a cada división para facilitar la visualización, y en el eje y se muestran las cantidades de cada división. Por lo tanto, podemos observar que la división "Entrada/Salida" muestra una cantidad significativamente mayor en comparación con las demás.



## 4. Interfaz Shiny

Finalmente después de crear todos los gráficos y los mapas interactivos, hay que reunirlos e implementarlos en un cuadro de mandos de Shiny, para hacerlo de manera interactiva y más sencilla.



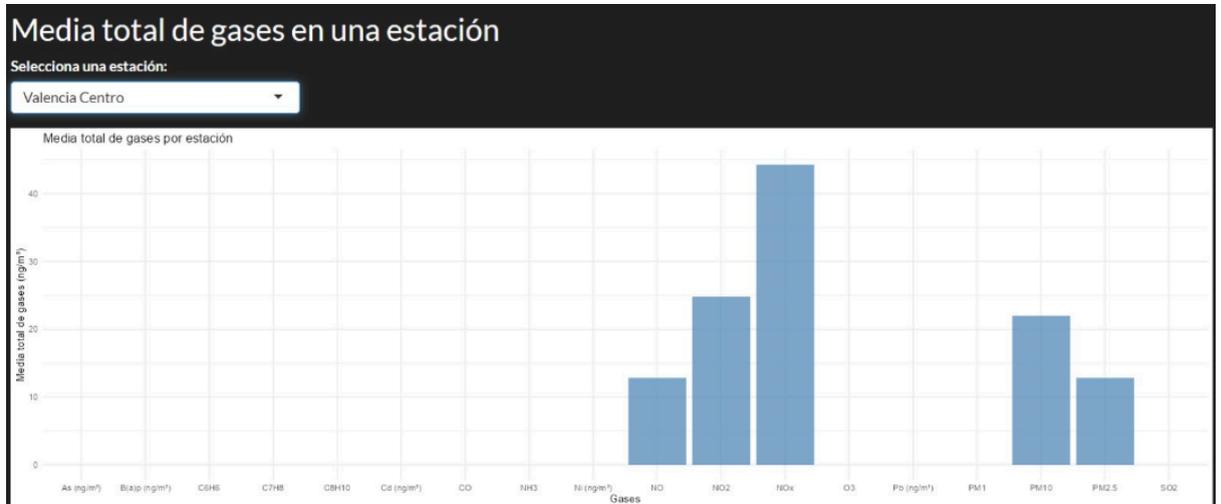
En este cuadro de mandos tenemos en la barra superior diferentes desplegables para navegar por las diferentes pantallas, en las que hemos puesto diferentes tipos de gráficos, mapas y formas de visualizar los datos.



Este mapa es de las zonas coincidentes en las que hay zonas verdes y tráfico.

Para finalizar hemos creado unos gráficos interactivos donde seleccionando la zona nos dan una gráfica media de los gases por estación, y la otra es la misma pero podemos elegir de que día queremos esta información, seleccionandolo en un desplegable.

Por la estación de valencia:



Por fecha en la misma zona que la anterior:

