



**HAL**  
open science

# Collecter et cartographier les données du bilan carbone d'un congrès

Chloé Friguet, François Husson

► **To cite this version:**

Chloé Friguet, François Husson. Collecter et cartographier les données du bilan carbone d'un congrès. Rencontres R 2024, Université Bretagne Sud, Jun 2024, Vannes (Bretagne), France. hal-04624860

**HAL Id: hal-04624860**

**<https://hal.science/hal-04624860v1>**

Submitted on 25 Jun 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Collecter et cartographier les données du bilan carbone d'un congrès

Cas des Rencontres R'24

Chloé FRIGUET & François HUSSON

# Introduction

Etablir le **bilan carbone** d'un événement scientifique : démarche globale éco-responsable

- quantifier les émissions de gaz à effet de serre (notamment de CO2) ainsi que d'autres paramètres comme la production de déchets ou la consommation alimentaire.
- charte SFdS, charte UBS, condition pour subventions, etc
- labo1.5

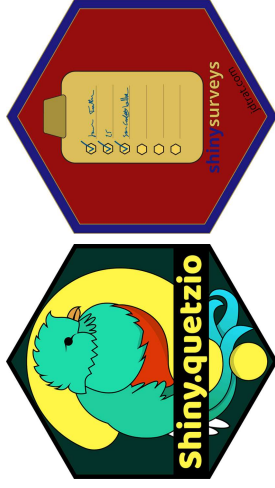
Contributions :

- une **application R-shiny** pour recueillir les informations auprès des participant-e-s
- des **dataviz** pour représenter les résultats obtenus

# Le questionnaire

# Construction d'un questionnaire avec R-shiny

Des solutions existent :



Mais peu de flexibilité pour des champs de saisie personnalisés :

- Développement d'une application R-shiny ad-hoc pour collecter les données ([shinydashboard](#))
- Mise à disposition du questionnaire sur le serveur [shinyapp.io](#)
- Recueil des données via une connexion [googledrive](#)

# Structure du questionnaire et champs de saisie (1)

- Profil participant-e : type d'employeur, expérience en R, âge, genre (`selectInput`, `numericInput`)



The image shows a web form titled "Votre profil" with a blue header. Below the header, there are several input fields:

- Votre profil** (header)
- Vous travaillez dans le domaine de...** (dropdown menu) with the selected value "Académique et institut de recherche".
- Votre expérience avec R...** (dropdown menu) with the selected value "Je pratique régulièrement".
- Avez vous déjà participé aux Rencontres R?** (dropdown menu) with the selected value "Oui, j'ai déjà participé à plusieurs précédentes éditions".
- Votre genre?** (dropdown menu) with the selected value "Féminin".
- Votre âge?** (numeric input field) with the value "40".

# Structure du questionnaire et champs de saisie (2)

- **Profil participant-e** : type d'employeur, expérience en R, âge, genre (`selectInput`, `numericInput`)
- **Habitudes** : alimentaires (`DT::dataTableOutput` + `DT::renderDataTable` avec `callback=JS()`), trajet domicile-travail

## Vos habitudes

### Vos habitudes alimentaires

Sur les 6 repas (midi + soir) prévus pendant les 3 jours des RR24, indiquez combien de repas vous auriez consommés en considérant les cas ci dessous. Indiquez 0 si vous ne consommez pas un type de repas. La somme doit faire 6 !

	0	1	2	3	4	5	6
repas uniquement végétarien	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
repas uniquement végétalien	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
repas avec de la viande blanche	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
repas avec de la viande rouge	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
repas avec du poisson	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

# Structure du questionnaire et champs de saisie (2)

- **Profil participant-e** : type d'employeur, expérience en R, âge, genre (`selectInput`, `numericInput`)
- **Habitudes** : alimentaires (`DT::dataTableOutput` + `DT::renderDataTable` avec `callback=JS()`), trajet domicile-travail (`numericInput`, `sliderInput`)

The image shows a digital questionnaire form with two main sections. The first section, titled "Vos habitudes de trajets", contains two input fields. The first field is labeled "Distance domicile-travail (pour UN trajet aller). Indiquez 0 si vous travaillez tout le temps à domicile." and has a numeric input with the value "10". The second field is labeled "Sur les 3 jours des Rencontres R, combien de jours seriez vous allé sur votre lieu de travail?" and has a numeric input with the value "3". The second section, titled "Répartition km par moyen de transport: 10 / 10 km.", contains a series of sliders for different transport modes. The instructions state: "Indiquer la distance (en km) pour chaque moyen de transport, pour UN trajet aller domicile-travail. La somme doit correspondre à la distance totale indiquée précédemment!". The sliders are: "à pied, à vélo, en trottinette" (value 10), "en voiture électrique" (value 0), "en bus, en métro, en tram" (value 100), "en train" (value 0), "en voiture thermique" (value 100), and "à moto, à scooter" (value 0). Each slider has a scale from 0 to 100.



# Structure du questionnaire et champs de saisie (3)

- **Profil participant-e** : type d'employeur, expérience en R, âge, genre (`selectInput`, `numericInput`)
- **Habitudes** : alimentaires (`DT::dataTableOutput` + `DT::renderDataTable` avec `callback=JS()`), trajet domicile-travail (`numericInput`, `sliderInput`)
- **Trajets** pour venir aux RR24: choix et visualisation (`textInput+renderUI+UIoutput`, `selectInput`, entrées conditionnelles, `carte leafletOutput`)

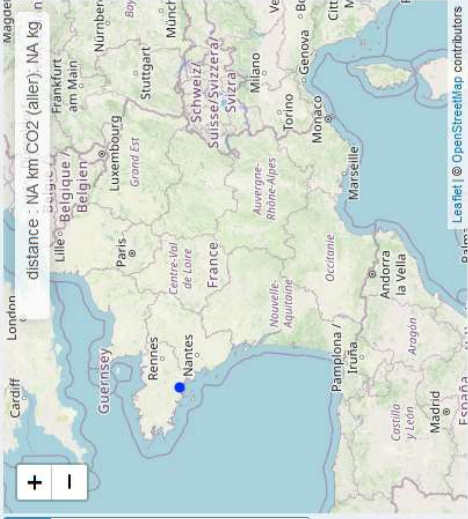
**Recherche de trajet**

Indiquer le nom de votre ville de départ ou d'étape en toutes lettres: commencer à taper le nom de la ville et sélectionner dans la liste. Laisser l'étape vide si non concerné.

Si la ville est à l'étranger, indiquer l'indicatif du pays entre parenthèse. Par exemple: Berlin (GE); Mons (BE); Luxembourg (LU); Cambridge (UK); Genève (SZ); Dakar (SG); etc. Si la ville n'est pas en métropole, indiquer l'indicatif FR entre parenthèse. Par exemple: Nouméa (FR); etc

distance : NA km CO2 (aller) : NA kg

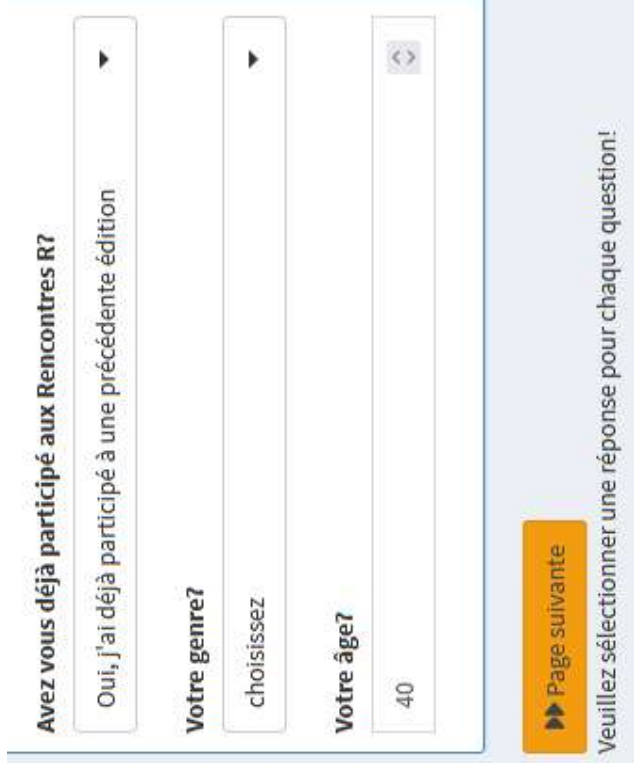
ville de départ  
Nom ou code postal  
train  
ville d'arrivée  
Vannes (56000)



Maguel  
London  
Lille  
Paris  
Guernsey  
Rennes  
Nantes  
France  
Belgique  
Luxembourg  
Grand Est  
Schweiz/Svizzera/Svizzera  
München  
Munich  
Milano  
Torino  
Monaco  
Genova  
Città del Vaticano  
Occitanie  
Andorra  
la Vella  
Castilla y León  
Aragón  
Madrid  
Fes  
Morocco  
Leaflet | © OpenStreetMap contributors

# Contrôle de saisie (1)

- **Bouton** pour passer d'une page à l'autre et bouton de validation du questionnaire : bloqués + message si incohérence



Avez vous déjà participé aux Rencontres R7

Oui, j'ai déjà participé à une précédente édition

▼

▼

Votre genre?

choisissez

▼

Votre âge?

40

▼

▶▶ Page suivante

Veillez sélectionner une réponse pour chaque question!

# Contrôle de saisie (2)

- Bouton pour passer d'une page à l'autre et bouton de validation du questionnaire : bloqués + message si incohérence
- Titres dynamiques

### Vos habitudes de trajets

Distance domicile-travail (pour UN trajet aller). Indiquez 0 si vous travaillez tout le temps à domicile.

Sur les 3 jours des Rencontres R, combien de jours seriez vous allé sur votre lieu de travail?

## Répartition km par moyen de transport: 10 / 10 km.

Indiquer la distance (en km) pour chaque moyen de transport, pour UN trajet aller domicile-travail. La somme doit correspondre à la distance totale indiquée précédemment!

à pied, à vélo, en trottinette	<input type="text" value="0"/>	<input type="text" value="10"/>	<input type="text" value="50"/>	<input type="text" value="100"/>
en bus, en métro, en tram	<input type="text" value="0"/>	<input type="text" value="100"/>	<input type="text" value="250"/>	<input type="text" value="100"/>
en voiture thermique	<input type="text" value="0"/>	<input type="text" value="100"/>	<input type="text" value="100"/>	<input type="text" value="100"/>
en voiture électrique	<input type="text" value="0"/>	<input type="text" value="100"/>	<input type="text" value="100"/>	<input type="text" value="100"/>
en train	<input type="text" value="0"/>	<input type="text" value="250"/>	<input type="text" value="100"/>	<input type="text" value="100"/>
à moto, à scooter	<input type="text" value="0"/>	<input type="text" value="100"/>	<input type="text" value="100"/>	<input type="text" value="100"/>

# Contrôle de saisie (3)

- **Bouton** pour passer d'une page à l'autre et bouton de validation du questionnaire : bloqués + message si incohérence
- **Titres dynamiques**
- **Choix des villes** (départ, étape, arrivée) du trajet: Liste des villes avec affichage en cours de saisie, tri par population décroissante

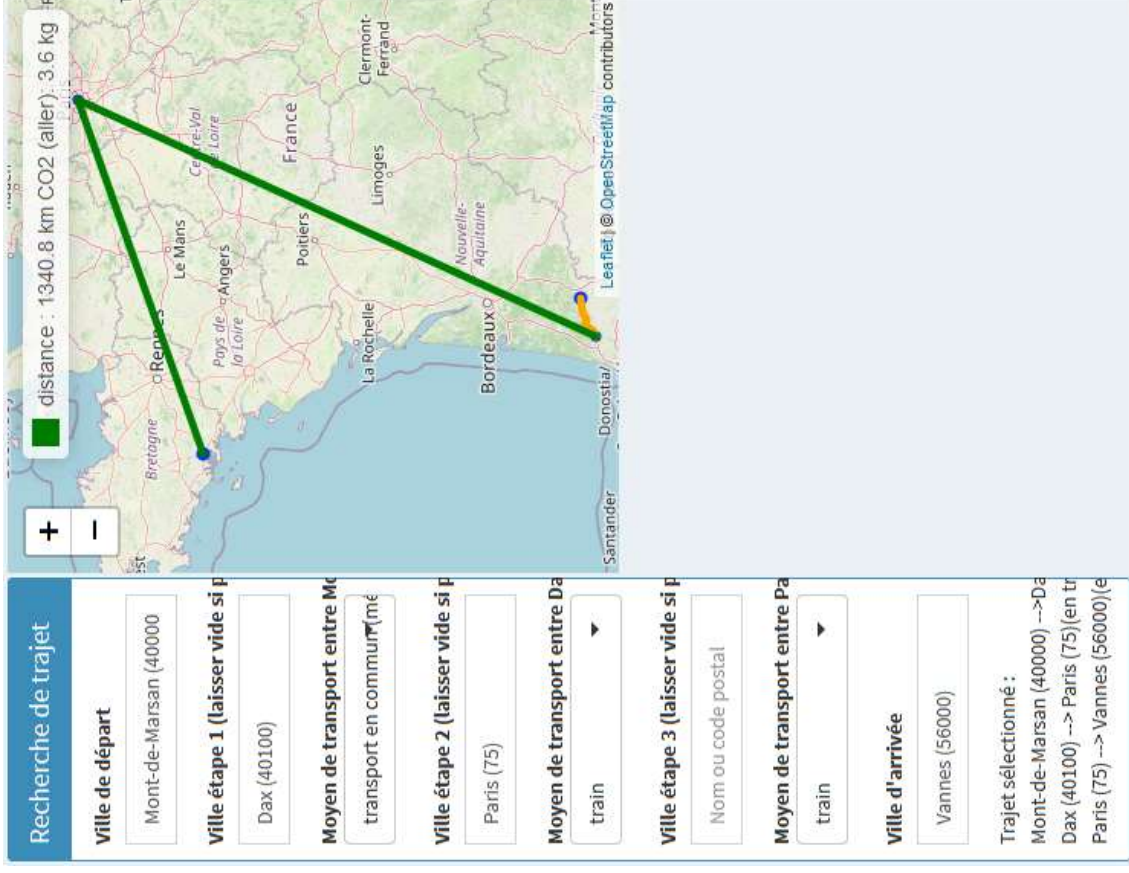
Recherche de trajet

Ville de départ

- Aurillac (15000)
- Auray (56400)
- Auriol (13390)
- Aureilhan (65800)
- Aurec-sur-Loire (43110)
- Dun-sur-Auron (18130)
- Auribeau-sur-Siagne (06810)
- Saint-Ferréol-d'Auroure (43330)

# Contrôle de saisie (4)

- Visualisation du trajet: villes de départ et étape + moyen de transport
- **geonames** : :GNsearch: latitude et longitude d'une ville en France et à l'étranger



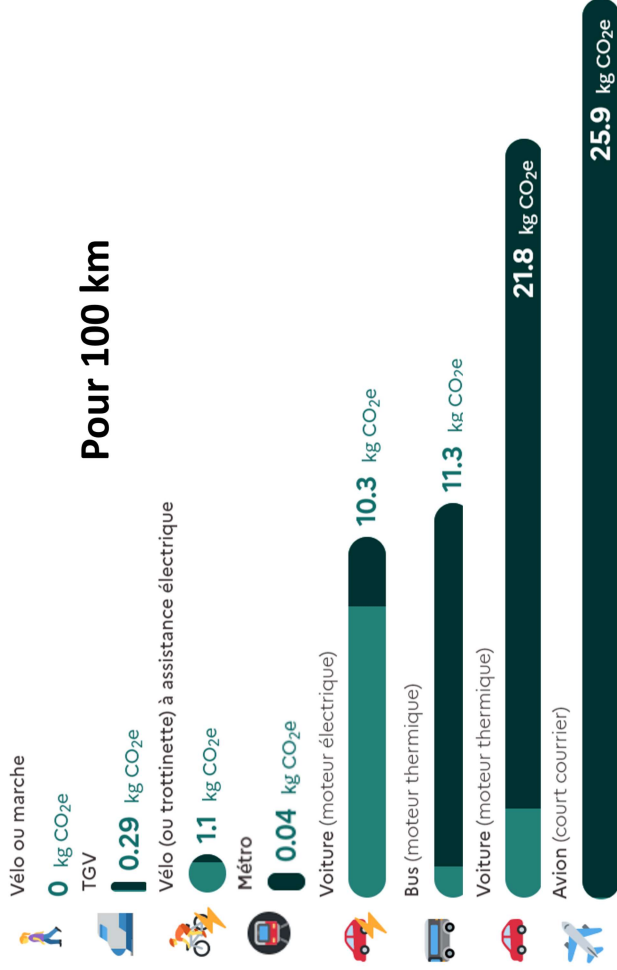
The screenshot displays a web application for calculating travel routes. It features a map of France with a green line connecting three cities: Mont-de-Marsan, Dax, and Vannes. A legend indicates a distance of 1340.8 km CO2 (aller) and 3.6 kg CO2. Below the map is a form with the following fields:

- Recherche de trajet**
- Ville de départ**: Mont-de-Marsan (40000)
- Ville étape 1 (laisser vide si possible)**: Dax (40100)
- Moyen de transport entre Mont-de-Marsan et Dax**: transport en commun (mètre)
- Ville étape 2 (laisser vide si possible)**: Paris (75)
- Moyen de transport entre Dax et Paris**: train
- Ville étape 3 (laisser vide si possible)**: Nom ou code postal
- Moyen de transport entre Paris et Vannes**: train
- Ville d'arrivée**: Vannes (56000)

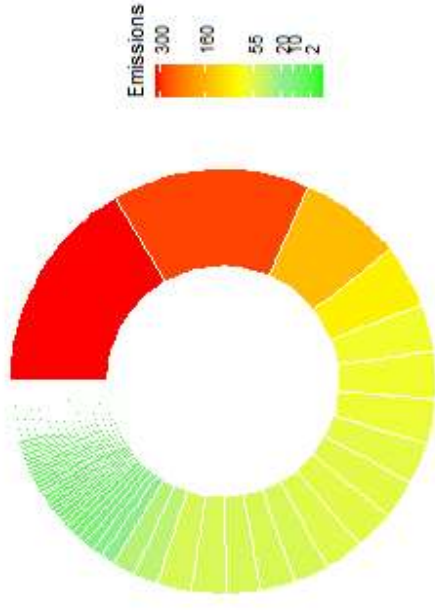
**Trajet sélectionné :**  
Mont-de-Marsan (40000) --> Dax (40100) --> Paris (75) (en train)  
Paris (75) --> Vannes (56000) (en train)

# Analyse des résultats

# Bilan CO2 par participant

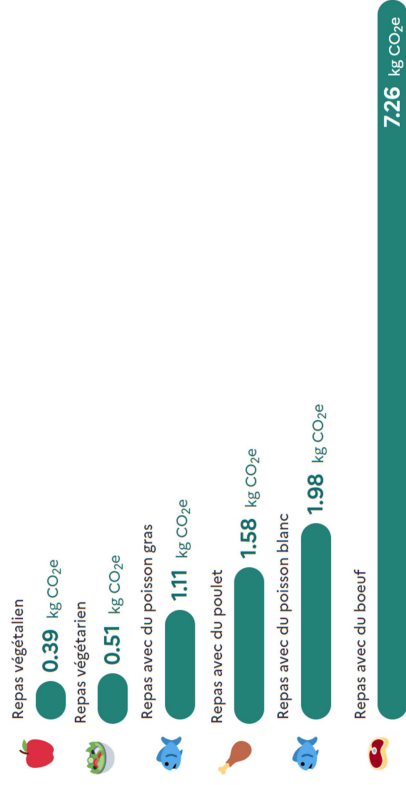


Répartition des émissions de CO<sub>2</sub> pour les trajets



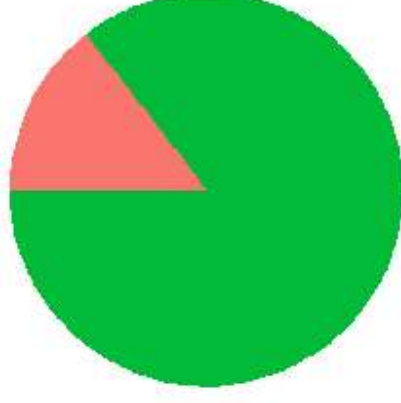
Source : ADEME

# Bilan CO2 par participant



Source : ADEME

Présentiel (24.99 kg CO<sub>2</sub> e)



Distanciel (15.3 kg CO<sub>2</sub> e)



Repas Trajet Visio

	Repas (5)	Trajets	Visio	Total
<b>quotidien</b>	10.61	3.38	0.96/2.35	15.3
<b>congrès</b>	3.62	21.37	0	24.99

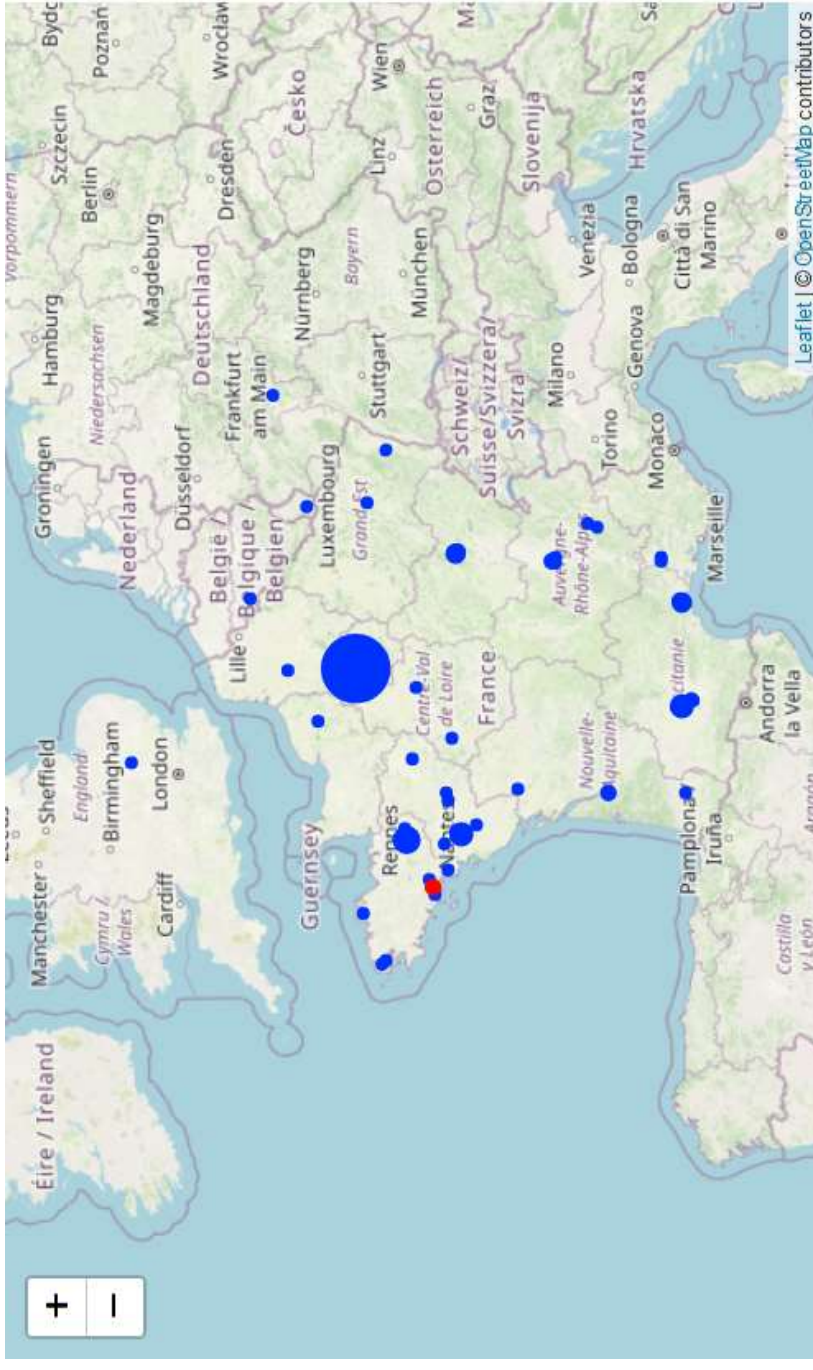
Moyenne des émissions de CO<sub>2</sub> par participant

Remarque : une donnée aberrante supprimée : trajet = 8734 kg CO<sub>2</sub> e !!!



# Villes d'origine des 91 participants

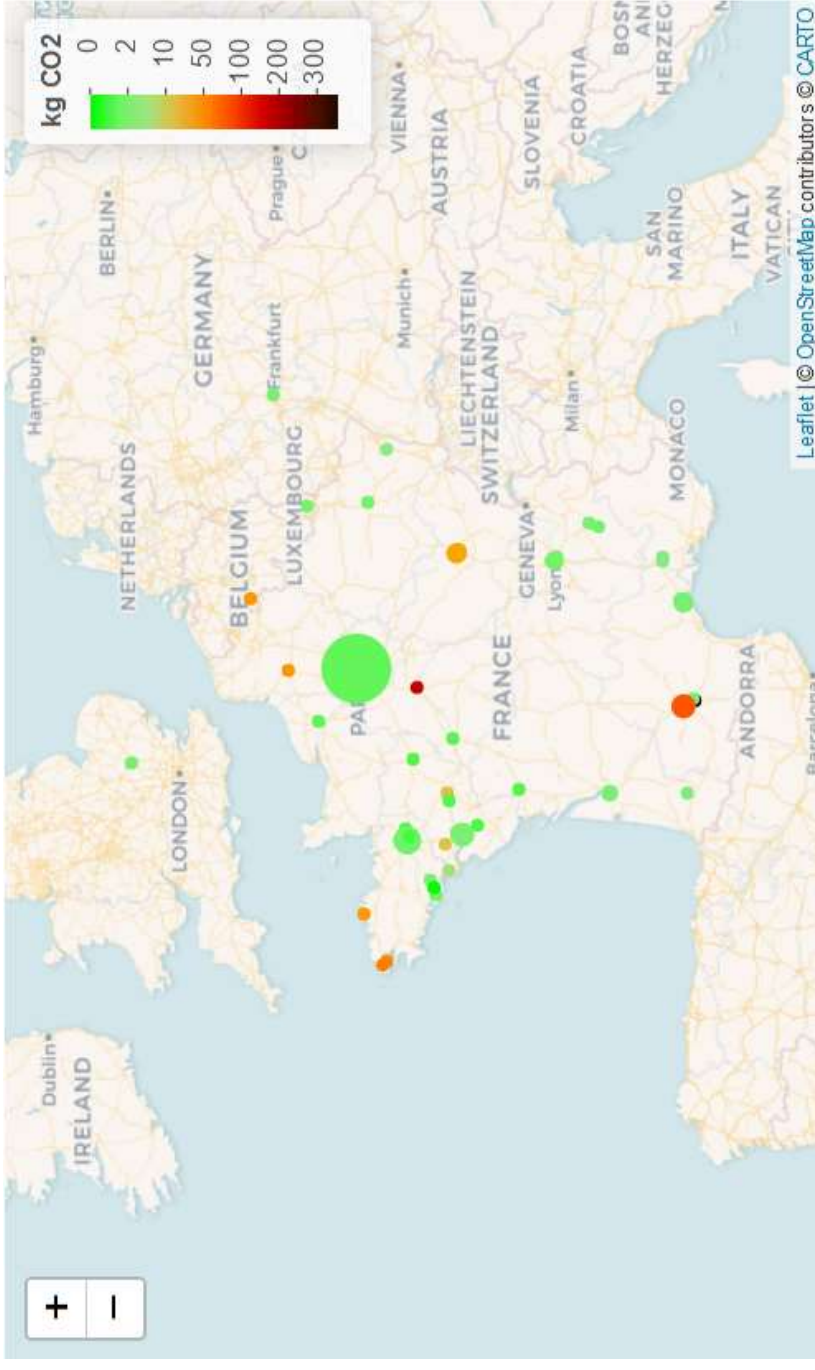
- carte leaflet
- fonctions `addProviderTiles`, `addCircleMarkers`
- utilisation de latitude et longitude des villes ainsi que de l'effectif



```
1 map <- leaflet() %>% addProviderTiles("OpenStreetMap.Mapnik") %>%
2   addCircleMarkers(data = tab_villes, lng = ~lng, lat = ~lat, radius = ~Freq,
3     fillOpacity = 1, opacity=1) %>%
4   addCircleMarkers(lng = vanne$lng, lat = vanne$lat, radius=2,
5     fillOpacity = 1, opacity=1, col="red")
```

# Bilan du CO2 moyen par ville d'origine

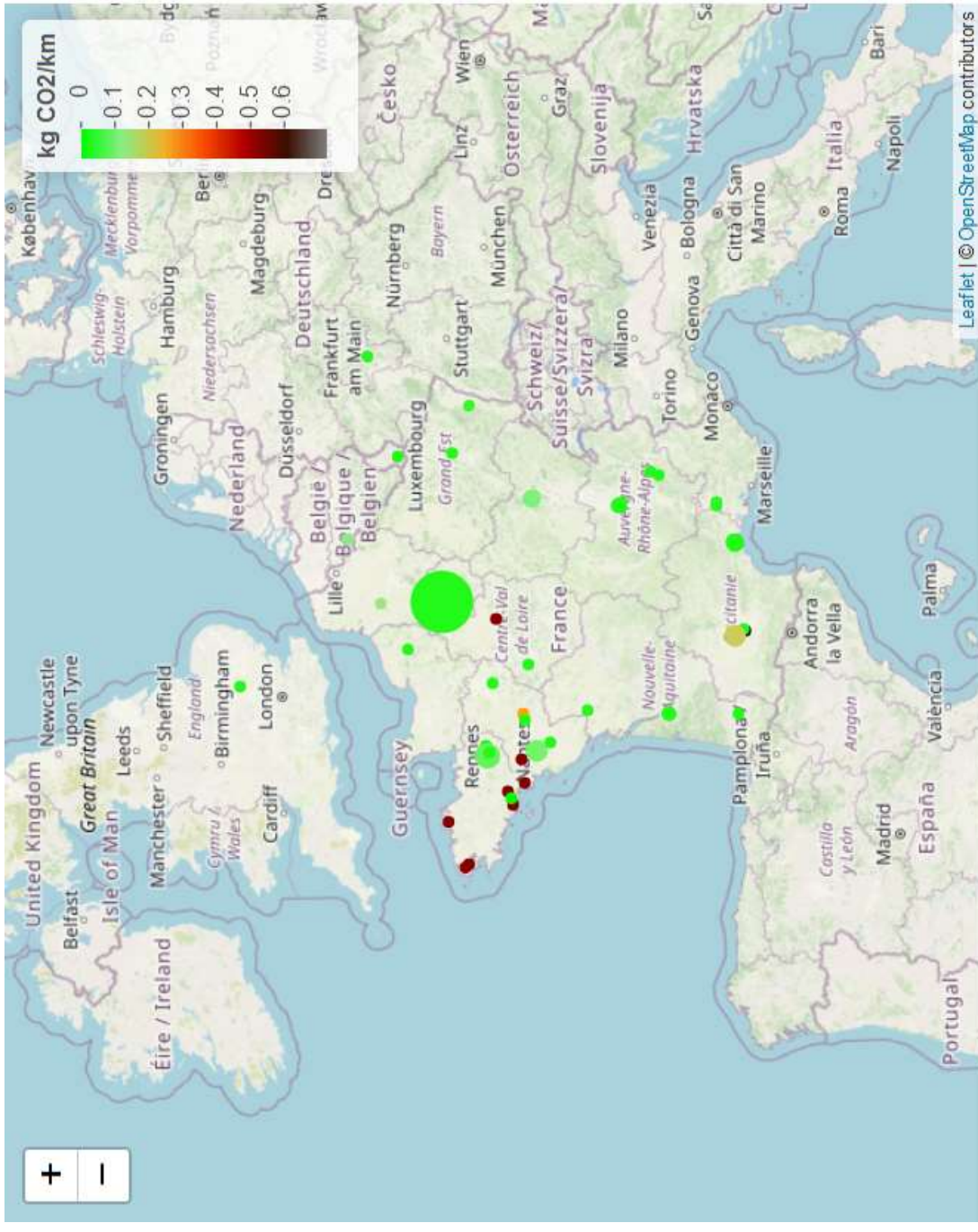
- carte `leaflet`
- fonctions `addProviderTiles`, `addCircleMarkers`
- utilisation de latitude et longitude des villes ainsi que de l'effectif
- coloration des points en fonction du bilan CO2 moyen des habitants d'une ville



```
1 pal2 <- colorNumeric(palette=c("green", "lightgreen", "orange", "red", "darkred", "black"),
2   domain = c(0, sqrt(max(dt$meanco2))))
3 map <- leaflet() %>% addProviderTiles("OpenStreetMap.Mapnik") %>%
4   addCircleMarkers(data = dt, lng = ~lng, lat = ~lat, radius = ~Freq,
5     color=~pal2(sqrt(meanco2)), fillOpacity = 1, opacity=1) %>%
6   addLegend(position = "topright", pal = pal2, values = sqrt(legend va
```

# Bilan du CO<sub>2</sub>/km moyen selon la ville d'origine

- carte `leaflet`
- fonctions `addProviderTiles`, `addCircleMarkers`
- utilisation de latitude et longitude des villes ainsi que de l'effectif
- coloration des points en fonction du bilan CO<sub>2</sub>/km moyen des habitants d'une ville



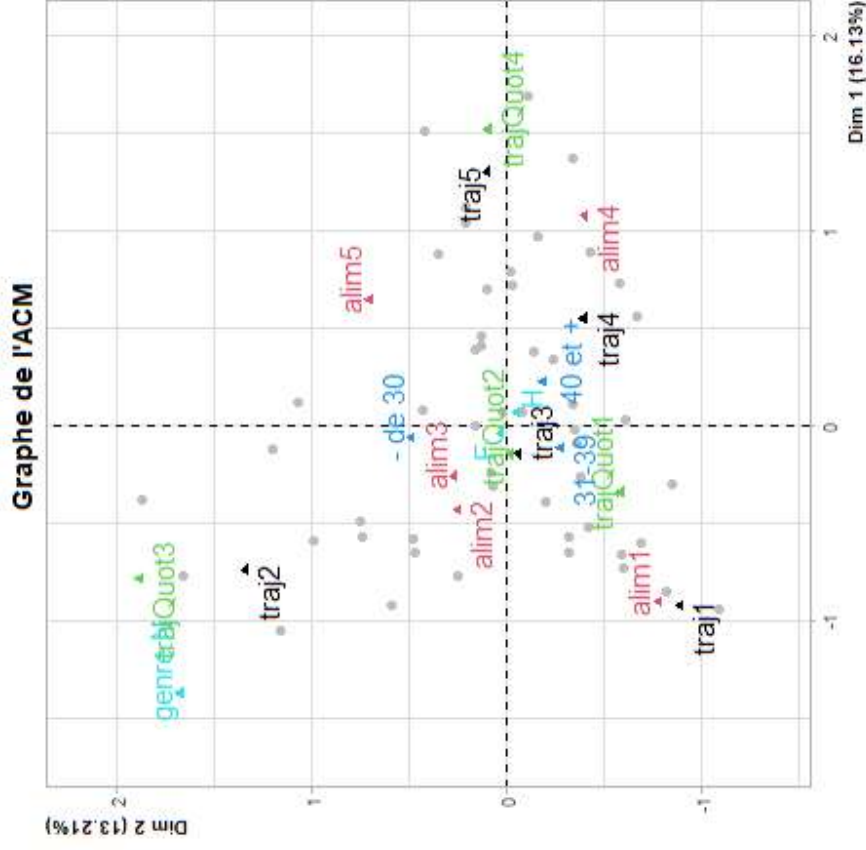
# Analyse de l'enquête par ACM

- fonction MCA de FactoMineR
- var actives : alimentation, trajet quotidien, trajet congès; illustratives : age, genre

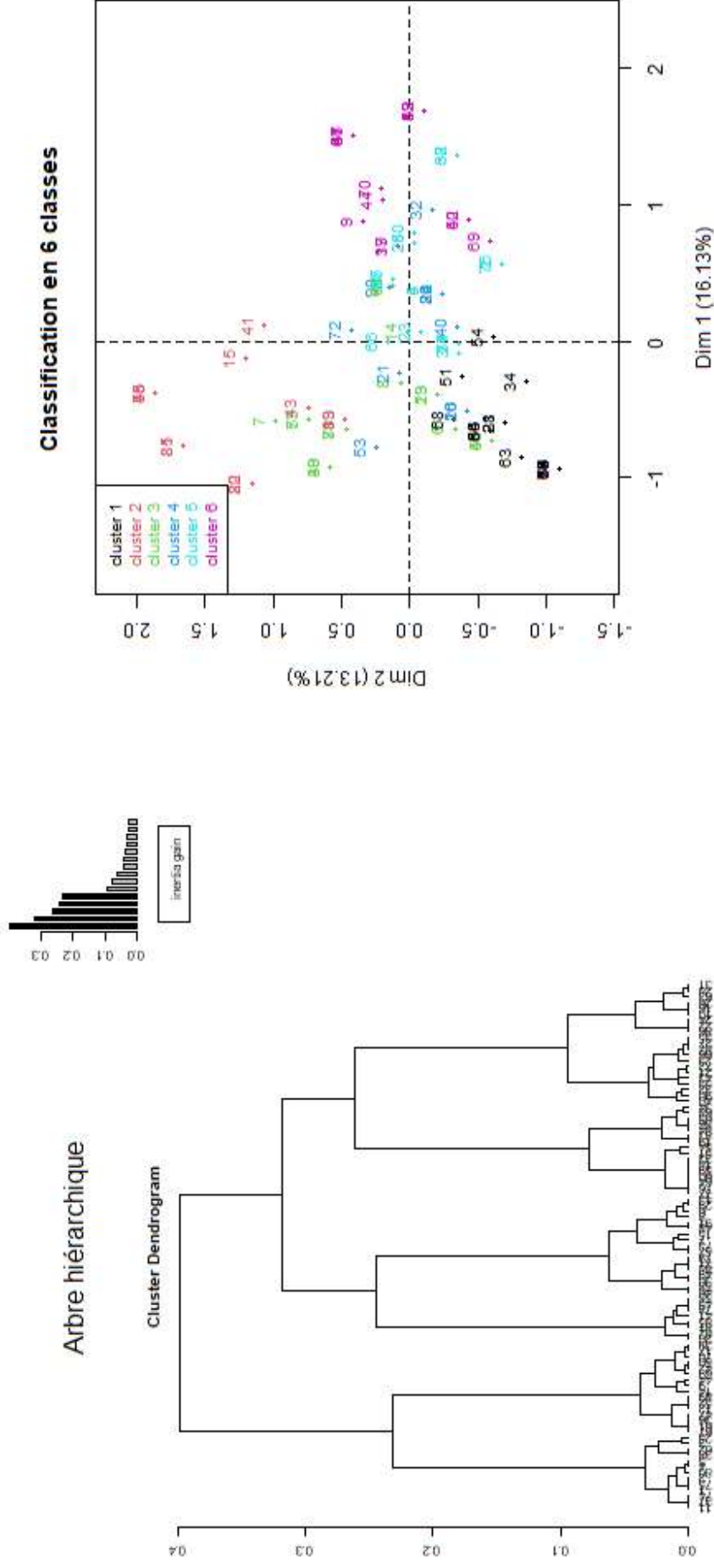
```
1 library(FactoMineR)
2 res.MCA<-MCA(dta, quali.sup=4:5)
```

ou

```
1 library(Factoshiny)
2 Factoshiny(dta)
```



# Classification des 91 participants



1 res.HCPC<-HCPC(res.MCA)

## Factoshiny(dta)

# Caratérisation des classes

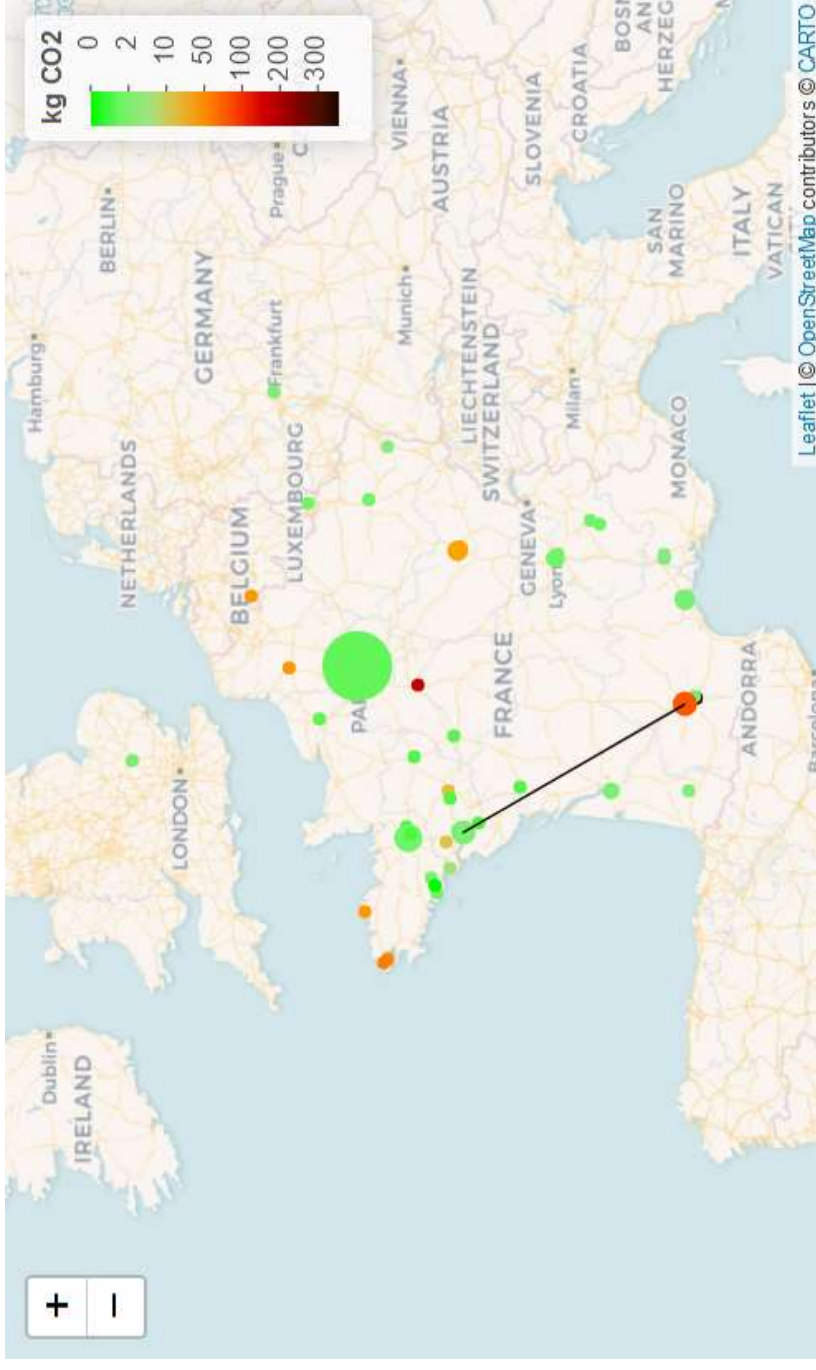
- fonction HCPC de **FactoMiner**
- caractérisation des classes par fonction **catdes**

```
$`Classe 1 (Effectif = 17)`  
traj=traj1      Cla/Mod  Mod/Cla  Global      p.value      v.test  
alim=alim1      68.421053  76.47059  20.87912  2.551566e-08  5.569715  
trajquotidien=trajQuot1  63.157895  70.58824  20.87912  6.633352e-07  4.971802  
ageQ=31-39      38.461538  88.23529  42.85714  3.251304e-05  4.155110  
ageQ=- de 30    31.428571  64.70588  38.46154  1.840147e-02  2.357439  
                6.451613  11.76471  34.06593  3.027185e-02  -2.166515  
  
$`Classe 2 (Effectif = 12)`  
trajquotidien=trajQuot3  Cla/Mod  Mod/Cla  Global      p.value      v.test  
traj=traj2      75.00000  75.00000  13.18681  5.643690e-08  5.429738  
ageQ=- de 30    55.55556  83.33333  19.78022  3.793299e-07  5.079048  
alim=alim3      29.03226  75.00000  34.06593  2.717341e-03  2.998026  
                33.33333  50.00000  19.78022  1.336824e-02  2.473806  
  
$`Classe 3 (Effectif = 17)`  
alim=alim2      Cla/Mod  Mod/Cla  Global      p.value      v.test  
traj=traj3      84.21053  94.11765  20.87912  6.101291e-14  7.505889  
                38.88889  41.17647  19.78022  2.587274e-02  2.228117  
  
$`Classe 4 (Effectif = 12)`
```

```
1 res.HCPC$desc.var
```

# Tracé d'un trajet sur une carte

- fonction `addPolylines`
- épaisseur des traits proportionnelle au nb d'itinéraires identiques
- très bien ... pour les avions



```
1 avion <- aggregate(Effectif ~ ., data = avion, FUN = sum)
2 for (i in 1:nrow(avion)) {
3   map <- addPolylines(map, lng=c(avion$lng.d[i], avion$lng.a[i]),
4     lat=c(avion$lat.d[i], avion$lat.a[i]), weight=avion$Effectif)
5 }
```

# Trouver un itinéraire et la distance associée

- en avion : `geosphere::distHaversine` calcule les distances à vol d'oiseau (→ utile pour les avions seulement!)

```
1 distHaversine(traj[,c("lng.d", "lat.d")], traj[,c("lng.a", "lat.a")]) / 1000
```

- en voiture, vélo, piéton : `osrm::osrmRoute` retourne un objet `sf` avec itinéraire vélo, piéton ou voiture et distance réelle

```
1 osrmRoute(src = traj[,c("lng.d", "lat.d")], dst = traj[,c("lng.a", "lat.a")],  
2   osrm.profile="car")
```

- en train : réseau inexistant ou avec les horaires de tous les trains → trop lourd  
Récupérer les données depuis l'API : <https://trainmap.ntag.fr/> :

```
1 url <- paste0("https://trainmap.ntag.fr/api/route?dep=", lng.d, ", ", lat.d,  
2   "&arr=", lng.a, ", ", lat.a)  
3 tra <- content(GET(url), "text")  
4 routes <- st_cast(st_boundary(st_make_valid(st_read(tra)), "LINESTRING"))
```

Paris et Bordeaux posent pb → modifier lat et long pour mettre le centre de Paris



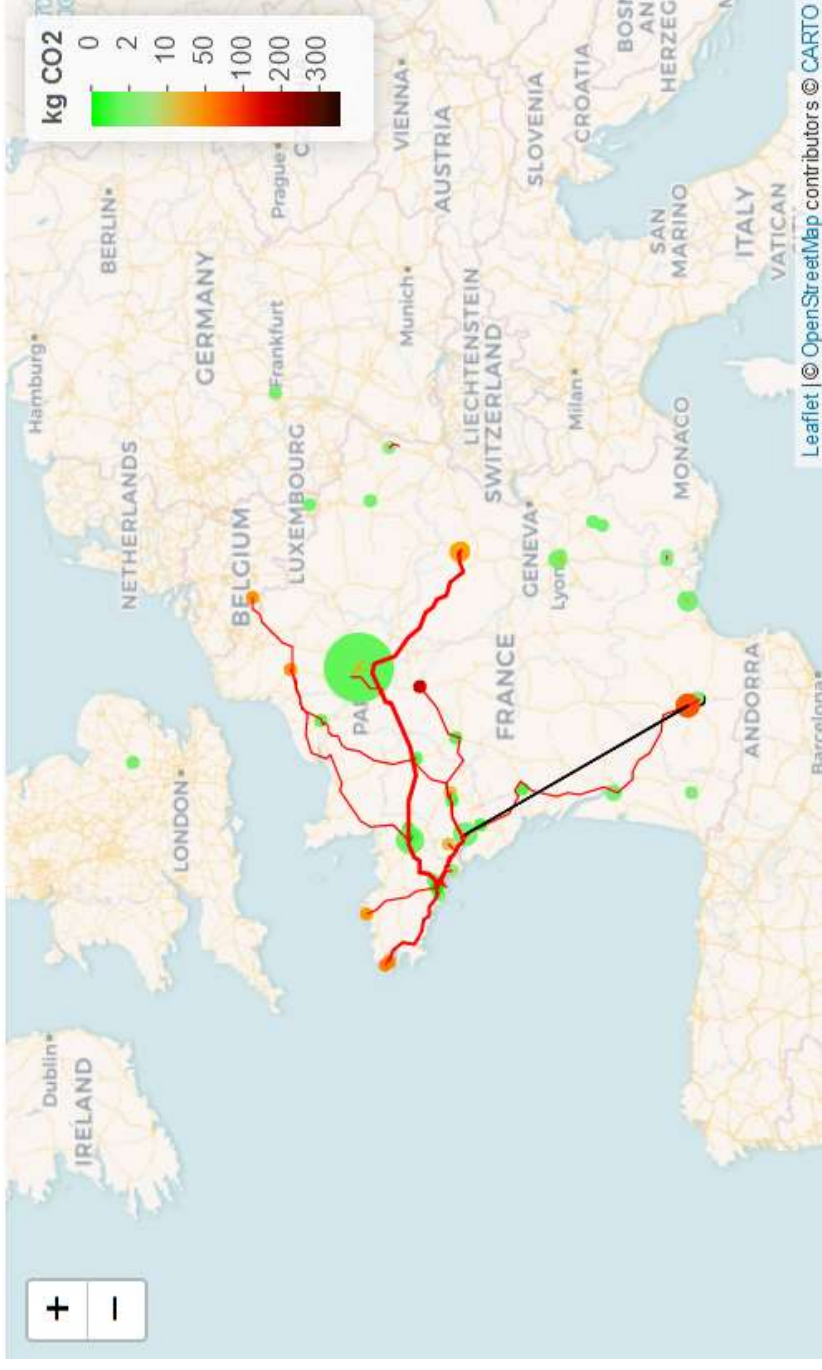
# Construire un réseau à partir d'itinéraires

- Création d'un réseau pour voiture, un autre pour réseau ferré, un autre pour vélo
- Comptage du nombre de personnes parcourant une même partie de trajet
- Superposition des trajets et bouts de trajets
- `stplanr::overLine`

```
1 tab_routes <- NULL
2 for (i in 1:nrow(traj_car)) {
3   tab_routes <- rbind(tab_routes,
4                       osrmRoute(src = traj_car[i,c("lng.d", "lat.d")],
5                                     dst = traj_car[i,c("lng.a", "lat.a")],
6                                       osrm.profile="car"))
7 }
8 tab_routes$total = rep(1,nrow(tab_routes))
9 reseau_car <- overline(tab_routes, attrib = "total")
```

# Carte des déplacements par la route

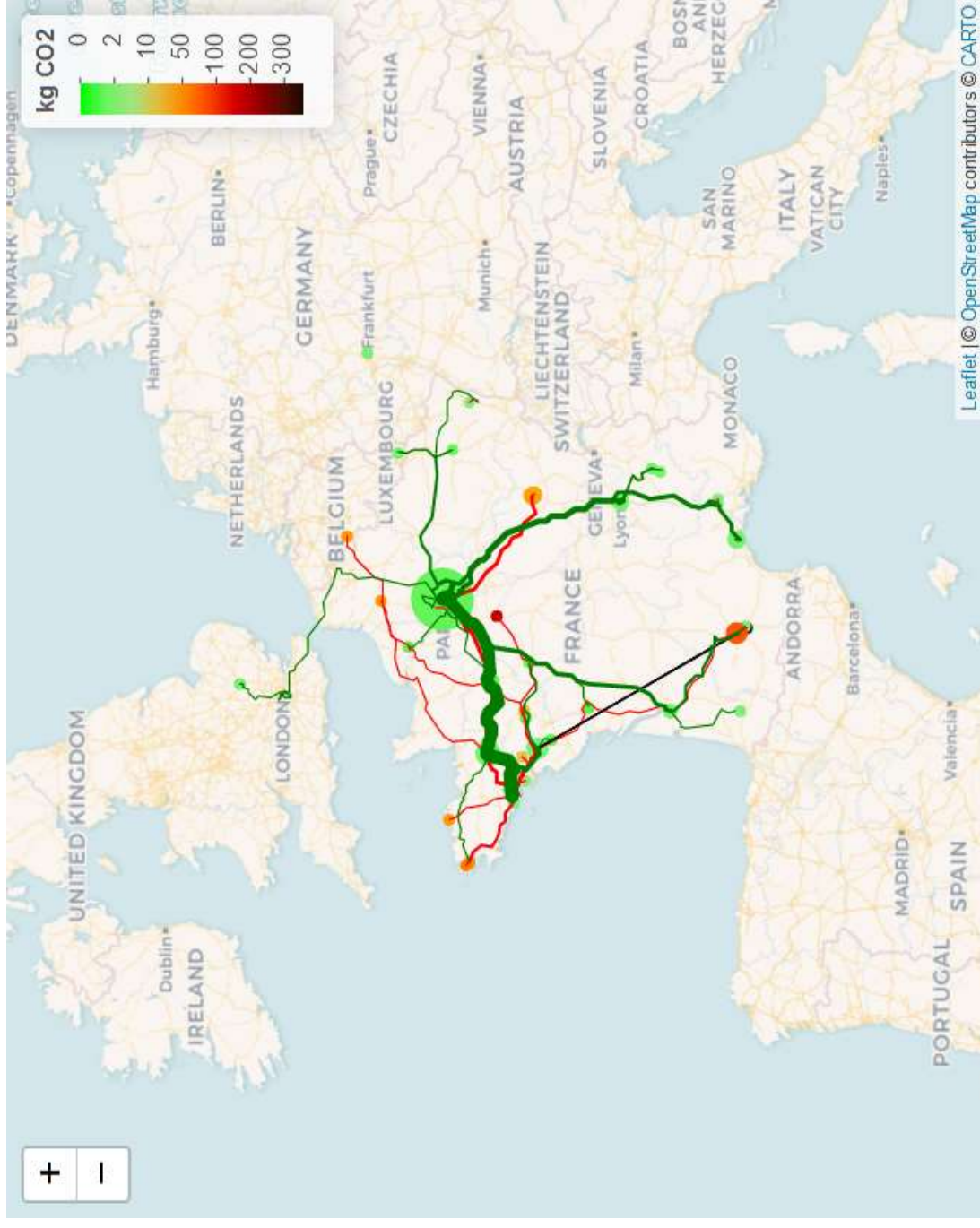
- chaque route du réseau est convertie en coordonnées puis tracée comme une suite de petites lignes droites



```
1 for (i in 1:nrow(reseau_car)) {
2   route <- reseau_car[i, ]
3   coords <- st_coordinates(route$geometry)
4   map <-addPolylines(map, lng=coords[,1], lat=coords[,2],
5                       weight=sqrt(route$total), color="red")
6 }
```

# Carte des déplacements du congrès Rencontres R

- Superposer tous les réseaux (voiture, vélo, train)
- pour les trains ... qq petits pb à régler



# Conclusion - perspectives

- R permet :
  - la construction de questionnaire avec shiny
  - le traitement du questionnaire
  - la visualisation par des cartes ... même si parfois ce n'est pas pas immédiat
- L'empreinte carbone est une mesure relativement facile à obtenir, il faudrait évaluer d'autres indicateurs (gestion ressources, provenance locale, plastique, etc.)
- **Quantifier/mesurer est indispensable - visualiser est nécessaire pour faire passer les idées et faire prendre conscience**
  - rôle important à jouer en tant que data scientists

**Merci pour votre participation au  
questionnaire et merci pour  
votre attention**

