IUT de Paris - Rives de Seine
Université Paris Cité

SciencesPo.

CONSTRUCTIONS, VISUALISATIONS ET ANALYSES STATISTIQUES DE RESEAUX SOCIAUX ET ORGANISATIONNELS

STAGE DE 2E ANNÉE DE BUT SD

Eva BERTRAND

Maître de stage : Emmanuel Lazega

08 Avril 2024 – 21 Juin 2024

17 JUIN 2024

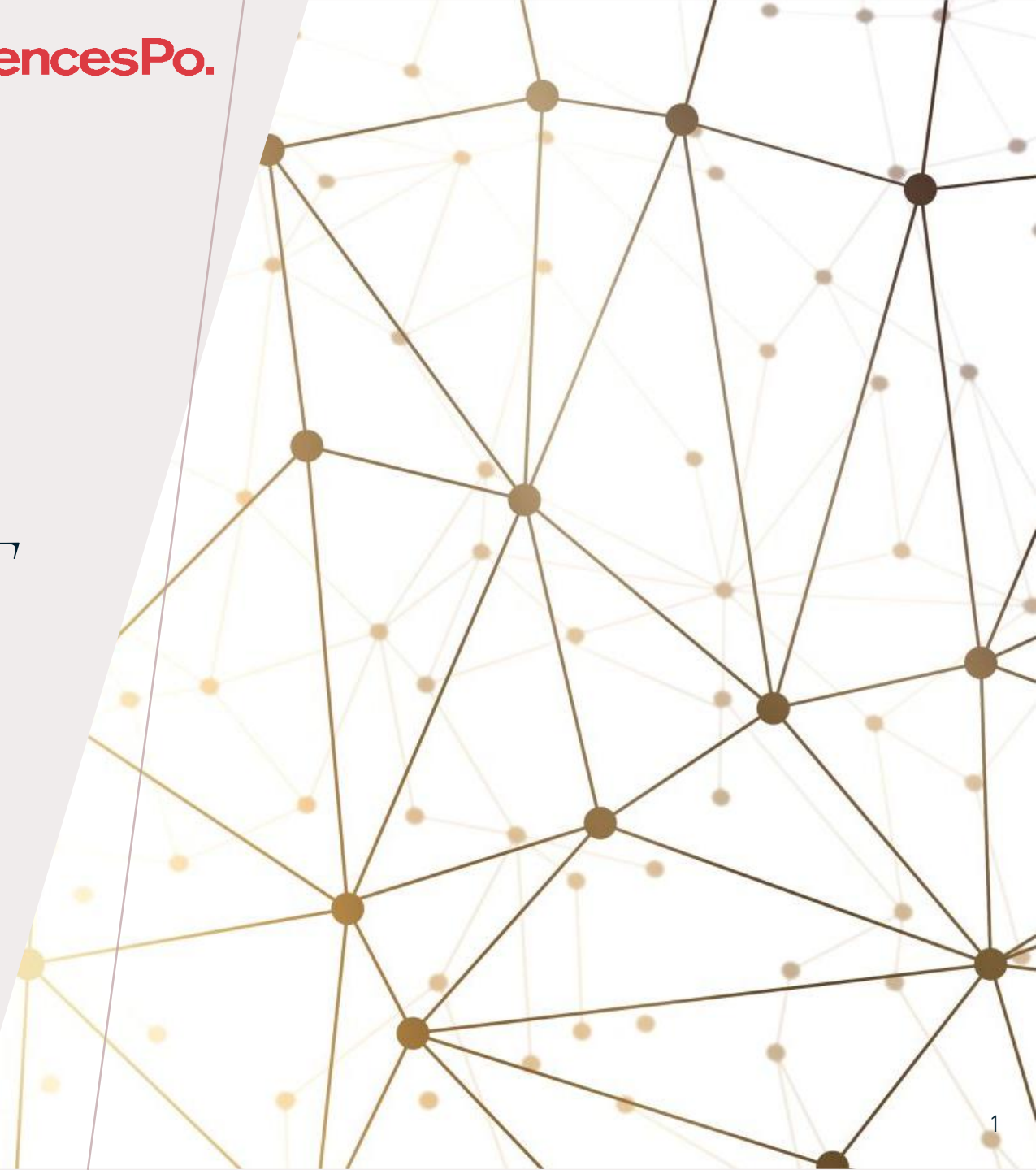
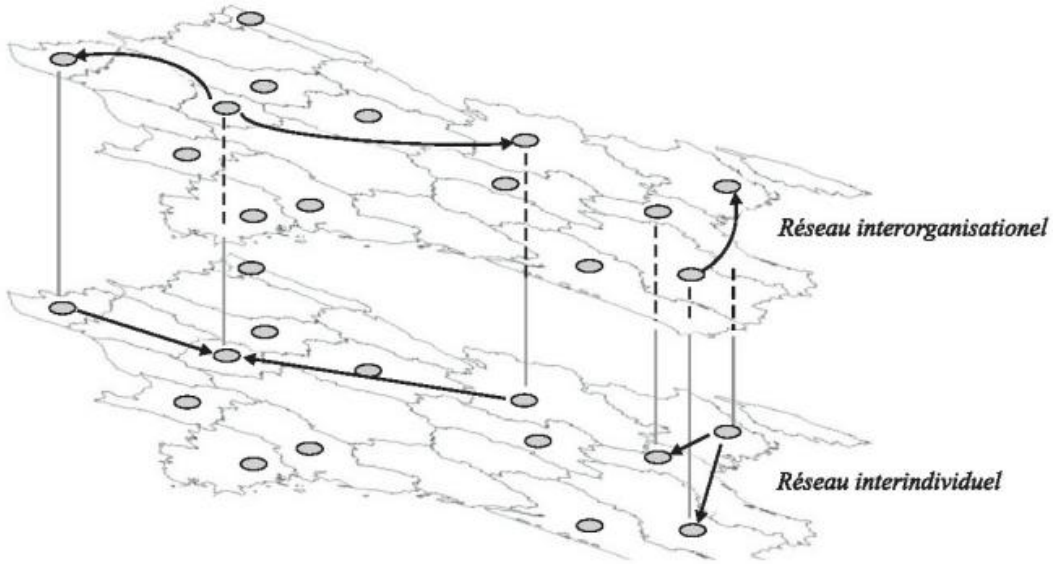


TABLE DES MATIÈRES :



CONTEXTE

- Contexte du stage :
  - Cadre formation BUT SD : 2eme année FI
  - Offre reçue de madame Bonnot
  - Découverte de la sociologie, mise en pratique

- L'entreprise :
  - Institut d'études Politiques de Paris
  - Créé en 1872 par Emile Boutmy
  - 7 écoles : Dijon, Havre, Menton, Nancy, Poitiers, Reims et Paris
  - Sciences politiques, droit, histoire, sociologie et économie

- Le service :
  - Centre de Sociologie des Organisations : CSO
  - Fondé en 1964 par Michel Crozier -> recherche sur l'administration française
  - S'étant sur : sociologie économique, organisations, action publique …

MISSION 1 : ETUDES SUR LES DONNÉES ARC

- Bases de données :
  - Chercheurs : 14 bases sur les réseaux des chercheurs
  - Laboratoires : 16 bases sur les réseaux des laboratoires

- Missions :
  - Création des analyses multi-niveaux entre les organisations et les individus,
  - Aide à la documentation et à l'archivage sur le site de datasciencespo des données, scriptes et publications,
  - Rédaction et conception de la documentation explicative

- Livrables :
  - Scripts R
  - Visualisations sortantes de R
  - Rapport explicatif

	U	V	W	X	Y	Z	AA
54	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0
63	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0
65	0	0	1	0	0	1	1	1
66	0	0	0	0	0	0	0	0
67	0	0	0	0	0	0	0	0
68	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0
71	0	0	0	0	0	1	0	0
72	0	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	0
75	0	0	0	0	0	0	0	0
76	0	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	1	0
79	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0

MISSION 1 : ETUDES SUR LES DONNÉES ARC

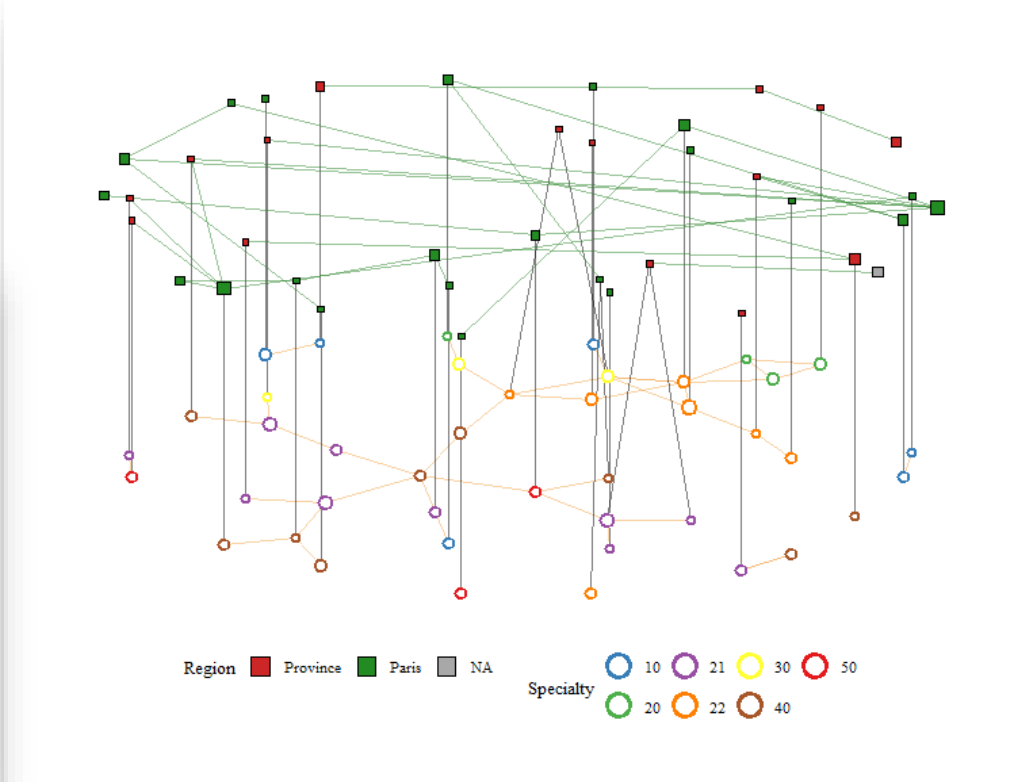
```r
#CHARGEMENT DES DONNEES
  #La fonction ci-dessus permet quand l'on rentre le chemin de données de charger le chemin du fichier .xls
load_excel = function(file_path) {
  data = read_xls(file_path, sheet = 1)
  | #permet de supprimer la première colonne de la base et de garder toutes les autres.
  data = data %>% select(-1)
   #permet de renommer les colonnes restantes de la base de 1 à n (n = nb total de colonne)
  colnames(data) = 1:ncol(data)
  return(data)
}

# Chargement des données et nettoyage
rs1 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-1.xls")
rs2 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-2.xls")
rs3 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-3.xls")
rs4 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-4.xls")
rs5 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-5.xls")
rs6 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-6.xls")
rs7 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-7.xls")
rs8 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-8.xls")
rs9 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-9.xls")
rs10 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-10.xls")
rs11 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-11.xls")
rs12 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-12.xls")
rs13 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-13.xls")
rs14 = load_excel("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/Chercheurs répondants 128/V1RTR-14.xls")
```

```r
#CHARGEMENT DES DONNEES
load_and_clean_excel <- function(file_path) {
  # Charger le dataframe depuis le fichier Excel
  data = read_xls(file_path, sheet = 1)

  # Supprimer les lignes qui contiennent des NA (cad n'ayant pas répondu)
  data = data[complete.cases(data), ]

  # Parcourir chaque valeur de la première colonne
  for (i in 1:nrow(data)) {
    # Extraire une sous-chaîne à partir du 2ème caractère
    data[i, 1] = substr(data[i, 1], start = 2, stop = nchar(data[i, 1]))
  }

  # Création d'une colonne pour stocker les résultats
  data$nom = NA

  # Boucle pour traiter chaque ligne
  for (i in 1:nrow(data)) {
    data$nom[i] = substr(data[i, 1], start = 4, stop = nchar(data[i, 1]))
  }

  # Réorganiser les colonnes pour placer la colonne "nom" en première position
  data = data[, c("nom", names(data)[-1])]

  # Supprimer la dernière colonne 'nom'
  data = data[, -ncol(data)]

  # Récupérer les noms des variables
  noms_variables = names(data)

  # Initialiser un vecteur pour stocker les colonnes à conserver
  colonnes_a_conserver = c()
  # Récupérer les 13 premiers caractères de la première colonne
  premieres_valeurs_colonne = substr(data$nom, start = 1, stop = 13)

  # Parcourir les noms des colonnes du dataframe
  for (variable in noms_variables) {
    # Extraire les 13 premiers caractères de la variable
    debut_variable = substr(variable, start = 1, stop = 13)

    # Vérifier si les 13 premiers caractères sont dans la première colonne
    if (!is.na(match(debut_variable, premieres_valeurs_colonne))) {
      colonnes_a_conserver = c(colonnes_a_conserver, variable)
    }
  }

  # Sélectionner uniquement les colonnes à conserver
  data = data[, colonnes_a_conserver]
  # Retourner le dataframe nettoyé
  return(data)
}
```

MISSION 1 : ETUDES SUR LES DONNÉES ARC

```
graph::degree(g)
' Supprimer les nœuds avec un degré inférieur à 2
odes_to_remove <- which(igraph::degree(g) <= 2)
 <- delete_vertices(g, nodes_to_remove)

' Afficher le nombre total de nœuds après suppression des isolats
um_nodes_deletion <- vcount(g)
:at("Nombre total de nœuds après suppression des isolats:", num_nodes_deletion, "\n")

' Identifier les nœuds isolés restants après suppression
solated_nodes <- which(igraph::degree(g) < 2)

' Supprimer ces nœuds isolés et leurs arêtes correspondantes
f (length(isolated_nodes) > 0) {
 g <- delete_vertices(g, v = isolated_nodes)


' Mettre à jour le nombre total de nœuds après suppression des isolats
um_nodes_total <- vcount(g)
:at("Nombre total de nœuds après fusion:", num_nodes_total, "\n")
```

```
# #fix1
# # xy3 utilise une fonction de disposition (layout_as_backbone) pour organiser les noeuds, en ignorant les noeuds isomorphes.
xy3 <- layout_as_multilevel(as.undirected(g),"fix1",
                            FUN1 = layout_with_fr,
                            ignore_iso = FALSE,
                            alpha = 45,beta = 90)


ecols <- c("#f4a548", "#2D882D")

# Tracé avec ggraph
ggraph(g, layout = 'manual', x = xy3[, 1], y = xy3[, 2]) +
  geom_edge_link0(aes(filter = (node1.lvl == 1 & node2.lvl == 1)), edge_width = 0.3, edge_colour = ecols[1], alpha = 0.5) +
  geom_edge_link0(aes(filter = (node1.lvl != node2.lvl)), alpha = 0.3, edge_width = 0.1, edge_colour = "black") +
  geom_edge_link0(aes(filter = (node1.lvl == 2 & node2.lvl == 2)), edge_width = 0.3, edge_colour = ecols[2], alpha = 0.5) +
  geom_node_point(shape = 21, aes(filter = (lvl == 1), size = nsize, col = as.factor(recrutement_laboratoires_type), stroke = 1.2),
                  fill = "white") +
  geom_node_point(shape = 22, aes(filter = (lvl == 2), size = nsize, fill = as.factor(recrutement_chercheurs_type))) +
  scale_size_continuous(range = c(1.5, 3.5), guide = FALSE) +
  scale_fill_manual(values = c("firebrick3", "forestgreen", "grey25"), na.value = "grey66", name = "Region",
                    labels = c("Province", "Paris")) +
  scale_color_manual(values = c("#377EB8", "#4DAF4A", "#984EA3", "#FF7F00", "#FFFF33", "#A65628", "#E41A1C"),
                     na.value = "grey66", name = "Specialty") +
  theme_graph(base_family = "serif", base_size = 10) +
  coord_cartesian(clip = "off", expand = TRUE) +
  theme(legend.position = "bottom", legend.box = "horizontal", legend.justification = "center") +
  guides(colour = guide_legend(override.aes = list(size = 5, stroke = 2))) +
  guides(fill = guide_legend(override.aes = list(size = 5), nrow = 1))

# Sauvegarde du graphique
ggsave("C:/Users/gevab/OneDrive/Documents/BUT/Stage/Archivage ARC/graphiques_multi/arc_45.pdf", width = 12, height = 10)
```



MISSION 2 : ETUDES SUR LES DONNÉES UPC

- Bases de données :
  - Events_Attributes : informations des événements
  - Actors_Attributes : informations des acteurs
  - Events_to_Actors : liste de participants par événements

- Missions :
  - Construction des visualisations (réseaux bipartites et des analyses multi-niveaux) entre les événements et individus,
  - Analyses descriptives de variables sur les événements et individus,
  - Rédaction et conception de la documentation explicative

- Livrables :
  - Scripts R
  - Visualisations sortantes du R
  - Rapport explicatif

event_city	event_cnt	latitude	longitude
San Servolo	IT	45.416665	12.34986
	IT		
London	UK	51.5074456	-0.1277653
Brussels	BL	50.8465573	4.351697
Warsaw	PL	52.2337172	21.0714322
Brussels	BL	50.8465573	4.351697
Brussels	BL	50.8465573	4.351697
Brussels	BL	50.8465573	4.351697
Strasbough	FR	48.584614	7.7507127
Strasbough	FR	48.584615	7.7507128
Strasbough	FR	48.584616	7.7507129
Munich	DE	48.1371079	11.5753822
Munich	DE	48.1371080	11.5753823

	id_eve_xx	list_actors
1	1	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,2
2	2	4,5,7,10,11,13,15,17,18,19,22,25,27,28,29,30,31,32,33,34,35,36,
3	3	28,42,43,39,44,45,46,47,48,49,50,51,52,53,54,55
4	4	5,19,40,47,48,56,57
5	5	7,58,59,60,61,62,63
6	6	4,5,7,23,40,47,48,56,57,64,65,66,67,68,69
7	7	56
8	8	5,46,56,57,66,71,72,73,74,75
9	9	
10	10	5,7,12,19,25,42,47,57,76,77,78,79,80,81,82,83
11	11	44,48,70,84,85,86,87,88
12	12	44,48,52,70,85,86,87
13	13	42,70,89,90,91,92,93,94,95,96,97,98
14	14	5,28,47,48,70,76,95,99,100,101,102
15	15	5,28,47,48,51,95,99,100,101,102,103,104
16	16	7,28,47,48,70,99,100,101,103,105,106,107
17	17	2,9,25,56,71,72,76,96,108,109,110,111,112,113,114,115,116,117
18	18	28,36,44,56,70,72,76,96,114,124,125,126,127,128,129,130,131,1
19	19	5,7,31,36,44,46,56,67,72,76,79,124,125,126,127,128,139,140,14
20	20	44,47,56,70,72,76,125,127,140,143,154,155,156,173,174,175,17
21	21	5,6,7,13,14,15,16,18,22,23,26,29,39,158,159,160,161,162,163,16
22	22	1,2,3,7,9,19,22,25,26,27,29,30,124,169,184,185,186,187,188,189
23	23	1,2,4,5,7,12,16,19,22,29,30,33,34,39,40,73,76,79,124,187,188,18
24	24	2,3,4,5,7,9,12,13,15,17,20,22,25,27,29,30,36,39,40,59,60,61,73,
25	25	7,64,73,328
26	26	5,7,36,44,64,70,71,76,78,109,123,264,329,330,331,332,333,334
27	27	44,70,76,79,335,336
28	28	31
29	29	70,338,340,341
30	30	151

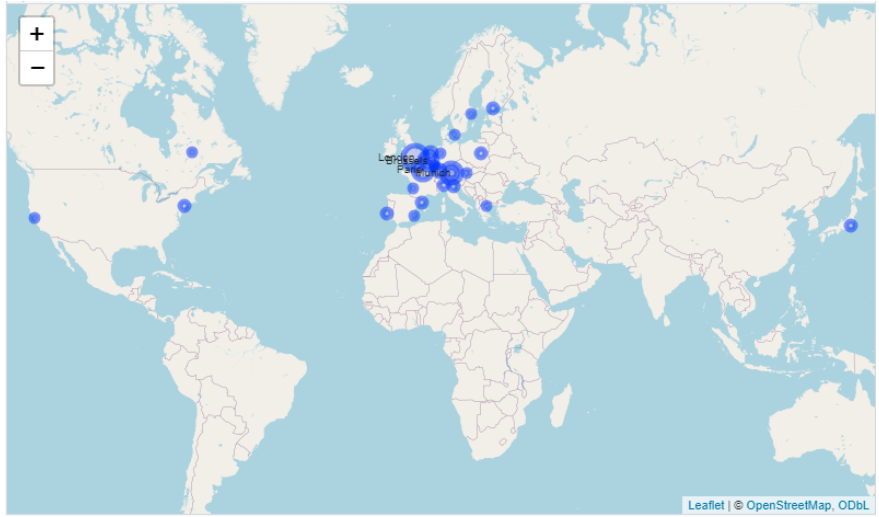
MISSION 2 : ETUDES SUR LES DONNÉES UPC

Events_Attributes

```
########## TRAITEMENT ##########
#-- Years --#
  #Fonction pour extraire les années
extract_years = function(date_string) {

    #initialisation des variables pour années de début et de fin
  year_start = NA
  year_end = NA

    #extraction de toutes les années
  years = str_extract_all(date_string, "\\d{4}")

    #si la chaine contient "from"
  if (grepl("from", date_string, fixed = TRUE)) {
      #si "from", l'année de début : première année extraite
    year_start = years[[1]]
      #année de fin : année actuelle
    year_end = as.numeric(format(Sys.Date(), "%Y"))
  }
    #si la chaine contient "till"
  else if (grepl("till", date_string, fixed = TRUE)) {
      #année de début : première année extraite
    year_start = years[[1]][1]
      #année de fin : deuxième année extraite
    year_end = years[[1]][2]
  }
   #si la chaine contient au moins une année
  else if (length(years[[1]]) > 0) {
      #année de début : première année extraite
    year_start = years[[1]][1]
     #si deuxième année, elle est définie comme année de fin
    if (length(years[[1]]) > 1) {
      year_end = years[[1]][2]
    }
     #année de fin = année de début
    else {
      year_end = year_start
    }
  }
   #si aucune année extraite
  if (is.na(year_start)) {
      #valeurs debut et fin sont "_"
    year_start = "_"
    year_end = "_"
  }
  return(data.frame(year_start, year_end))
}
```

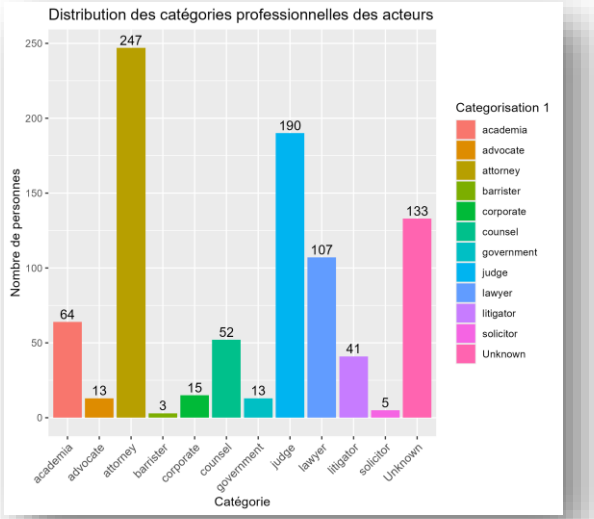
Mission : analyses descriptives

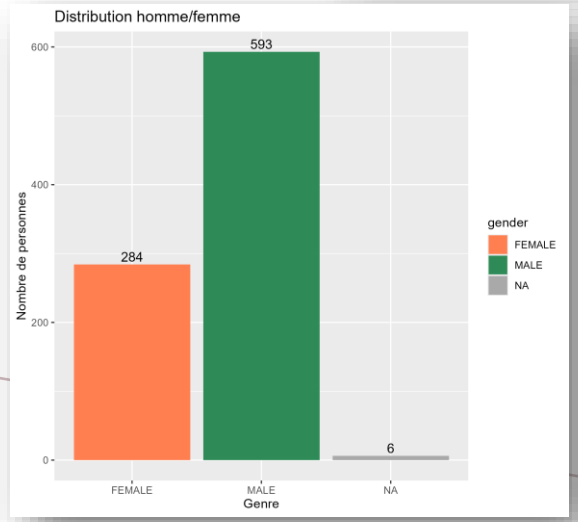
Variables utilisées : id, year, city, latitude, longitude, organizers



MISSION 2 : ETUDES SUR LES DONNÉES UPC

Actors_Attributes





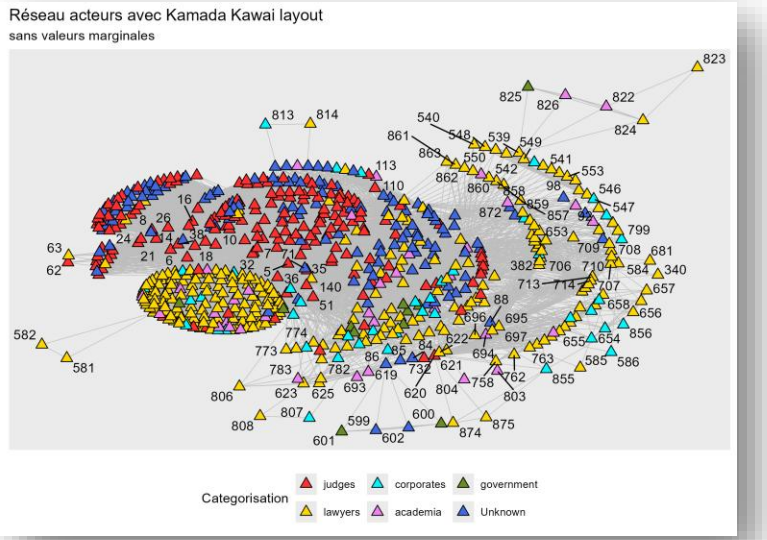
Mission : analyses descriptives

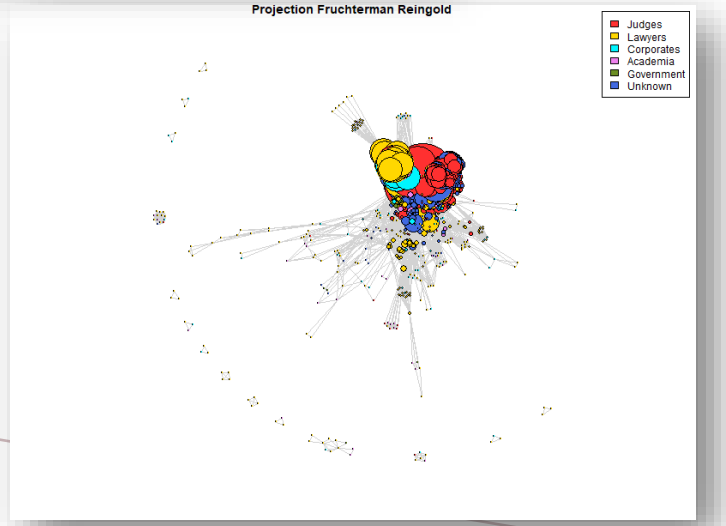
Variables utilisées : id, gender, cnt, Categorisation

```r
#-- Catégories Professionnelle des participants--#
  #on garde les 3 premières, la 117ème et 11 dernières colonnes
act_categ = actatt %>% select(1:3, 117, (ncol(.) - 10):ncol(.))

  #transformation des valeurs 'blank' et NA en Unknown
act_categ = act_categ %>% mutate(`Categorisation 1` = case_when(
  `Categorisation 1` == "blank" ~ 'Unknown',
  is.na(`Categorisation 1`) ~ 'Unknown',
  TRUE ~ `Categorisation 1`
))

  #count des catégories professionnelles
freq_categ = act_categ %>% count(`Categorisation 1`)

  #création du graphique pour visualiser la distribution des catégories
ggplot(freq_categ, aes(x=`Categorisation 1`, y=n, fill=`Categorisation 1`)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=n), vjust= -0.3, hjust= 0.5, color= "black", size= 4) +
  labs(x='Catégorie', y='Nombre de personnes', title='Distribution des catégories professionnelles des acteurs') +
  theme(axis.text.x=element_text(angle= 45, hjust= 1, size= 10))

  #sauvegarde
ggsave(file.path(actors_path, "11.Visualizations/Actors_Attributes_Visualizations/CategPro_Distribution.png"))

#---#
#-- Rassemblement des catégories professionnelles --#
act_categ$group = ifelse(act_categ$`Categorisation 1` == "judge", "Judges",
                         ifelse(act_categ$`Categorisation 1` == "academia", "Academia",
                                ifelse(act_categ$`Categorisation 1` == "gouvernment", "Gouvernment",
                                       ifelse(act_categ$`Categorisation 1` %in% c("advocate", "barrister", "litigator", "solicitor", "lawyer", "attorney"), "Lawyers",
                                              ifelse(act_categ$`Categorisation 1` %in% c("corporate", "counsel"), "Corporates",
                                                     ifelse(act_categ$`Categorisation 1` == "Unknown", "Unknown", act_categ$`Categorisation 1`)))))))

group_categ = act_categ %>% count(group)

  #graphique
ggplot(group_categ, aes(x= group, y= n, fill= group)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=n), vjust= -0.3, hjust= 0.5, color= "black", size= 4) +
  labs(x='Catégorie', y="Nombre d'individus", title='Distribution des groupes des catégories professionnelles des acteurs') +
  theme(axis.text.x=element_text(angle= 45, hjust= 1, size= 10))

  #sauvegarde
ggsave(file.path(actors_path, "11.Visualizations/Actors_Attributes_Visualizations/GroupCategPro_Distribution.png"))
```

MISSION 2 : ETUDES SUR LES DONNÉES UPC

Events_to _Actors





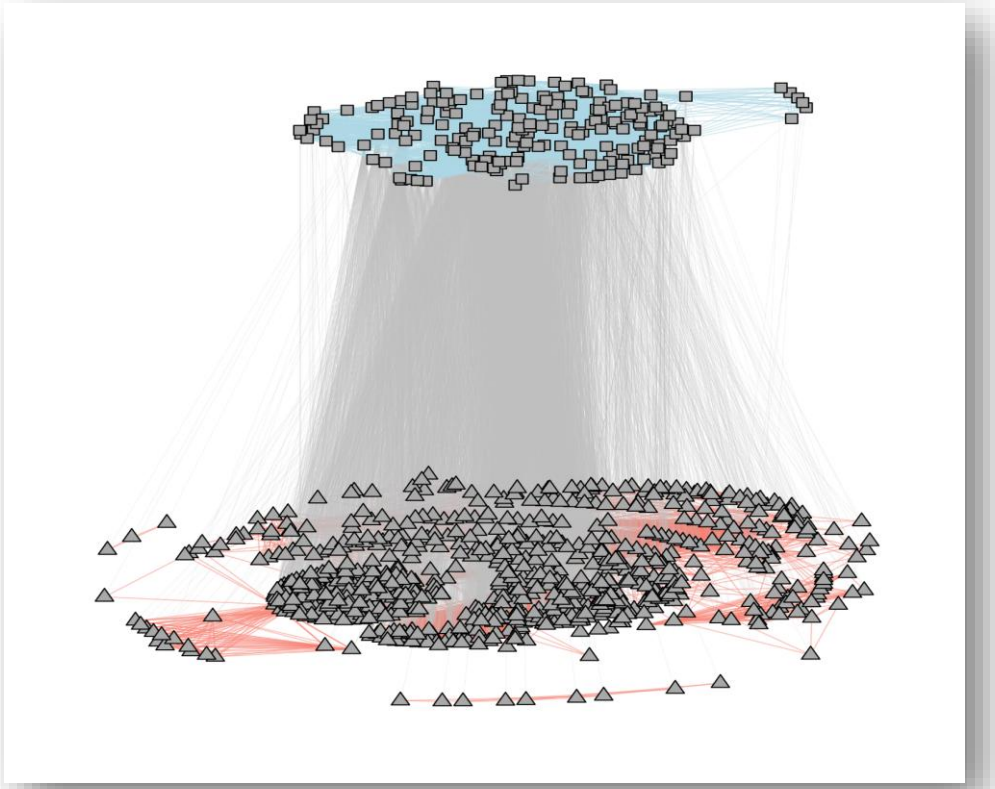
Mission : construction de visualisations du réseau bipartite et multi-niveau

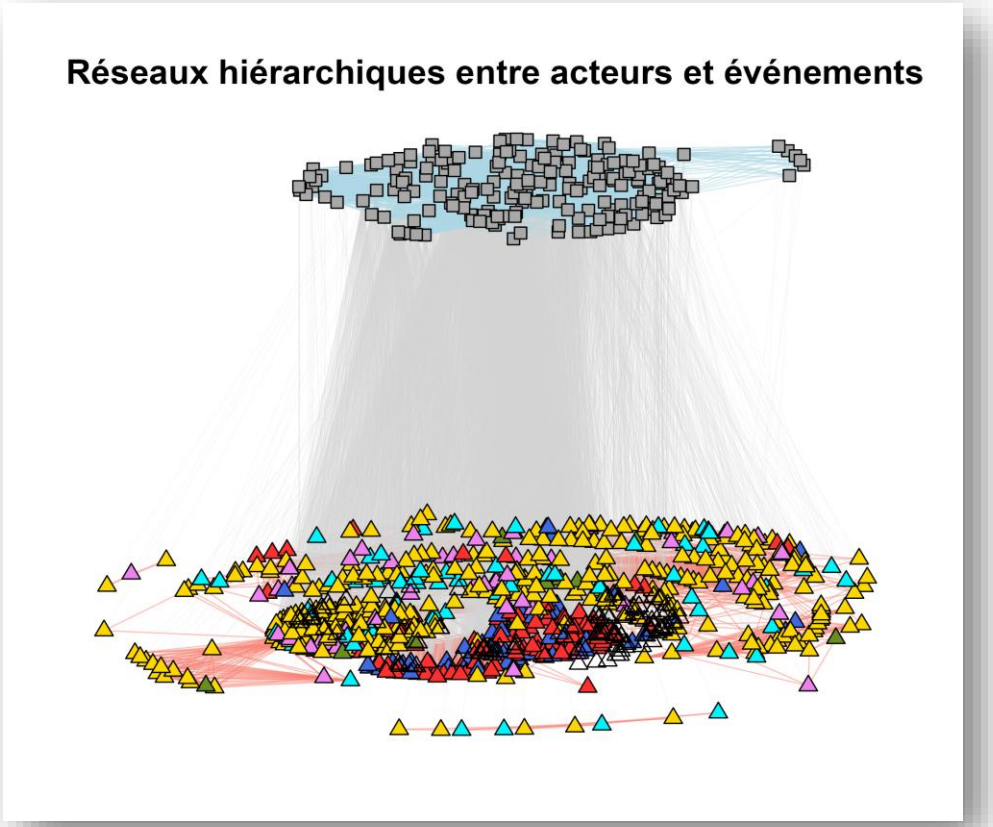
Variables utilisées : id (événements et acteurs), Categorisation, gender, ctn/city

```
#-- réseau des acteurs --#
#indentation des id
V(Actor_Graph)$id = Actors_Attributes$id_ind_xx[match(names(V(Actor_Graph)), Actors_Attributes$id_ind_xx)]

#mappage des couleurs
Actors_Attributes = Actors_Attributes %>%
  mutate(color = case_when(
    `Categorisation 1` == "judge" ~ "firebrick1",
    `Categorisation 1` %in% c("advocate", "barrister", "litigator", "solicitor", "lawyer", "attorney") ~ "gold",
    `Categorisation 1` %in% c("corporate", "counsel") ~ "turquoise1",
    `Categorisation 1` == "academia" ~ "violet",
    `Categorisation 1` == "government" ~ "olivedrab",
    `Categorisation 1` == "blank" ~ "royalblue",
    is.na(`Categorisation 1`) ~ "royalblue"
  ))
V(Actor_Graph)$color = Actors_Attributes$color[match(names(V(Actor_Graph)), Actors_Attributes$id_ind_xx)]
V(Actor_Graph)$categorisation = Actors_Attributes$`Categorisation 1`[match(names(V(Actor_Graph)), Actors_Attributes$id
V(Actor_Graph)$color = factor(V(Actor_Graph)$color,
                              levels = c("firebrick1", "gold", "turquoise1", "violet", "olivedrab", "royalblue"),
                              labels = c("judges", "lawyers", "corporates", "academia", "government", "Unknown"))

#-global visualizations-#
#Fruchterman Reingold
rep_act_global_fr = ggraph(Actor_Graph, layout = "fr") +
  geom_edge_link(edge_colour = "grey", width = 0.2) +
  geom_node_point(aes(fill = color), size = 3, shape = 24, color = "black") +
  labs(title = "Réseau acteurs", subtitle= "Fruchterman Reingold layout") +
  geom_node_text(aes(label = id), repel = TRUE) +
  scale_fill_manual(values = c("firebrick1", "gold", "turquoise1", "violet", "olivedrab", "royalblue"),
                    name = "Categorisation") +
  theme(legend.position = "bottom")
```

MISSION 2 : ETUDES SUR LES DONNÉES UPC

Events_to _Actors





CONCLUSION GÉNÉRALE

Les apports du stage :

- Techniques :
  - Renforcement dans l'écriture de mes scripts
  - Meilleure analyse des attentes et besoins
  - Renforcement dans les étapes de visualisations

- Savoir-être :
  - Meilleure organisation
  - Autonomie
  - Plus de persévérance et de confiance
  - De la curiosité

Découverte du monde professionnel:

- Quelques appréhensions mais vite dissipées (charges de travail, compétences)
- Nouveau rythme de travail et nouvel environnement

Découverte du domaine de la sociologie :

- Domaine très intéressant et enrichissant
- Découvrir les différentes possibilités

DÉMARCHE PORTFOLIO

- Traiter des données à des fins décisionnelles :
  - Nécessité de tester, documenter et corriger son programme
  - Intervention à toutes les étapes de vie de la donnée
  - Inscription dans une démarche de documentation des réalisations adaptée au public visé

- Analyser statistiquement des données :
  - Mise en évidence les grandes tendances et les informations principales
  - Intérêt des analyses multivariées pour synthétiser et résumer l'information portée par plusieurs variables

- Valoriser une production dans un contexte professionnel :
  - Expression correcte, en français ou dans une langue étrangère, à l'oral et à l'écrit
  - Choix des indicateurs pertinent pour la communication des résultats
  - Utilisation de la forme de restitution adaptée

- Développer un outil décisionnel :
  - Mise en œuvre d'une structuration des données adaptée à leurs caractéristiques
  - Réalisation de solutions de visualisation spécifiques aux données métier
  - Compréhension les enjeux de l'automatisation et de l'interopérabilité d'un ensemble de tâches

MERCI DE M'AVOIR ÉCOUTÉE !!!