

# TP Statistique

Manal El Karchouni, Roxane Gall, Jean Haberer

31 Mars 2020

Voici le plan de ce qui sera fait dans le TP.

## 0. Visualisation de chemins

Lecture du fichier des villes :

```
villes <- read.csv('./DonneesGPSvilles.csv',header=TRUE,dec='.',sep=';',quote="\")
str(villes)

## 'data.frame': 22 obs. of 5 variables:
## $ EU_circo : Factor w/ 7 levels "Île-de-France",...: 6 6 4 3 7 4 3 2 3 4 ...
## $ region : Factor w/ 22 levels "Alsace","Aquitaine",...: 22 10 19 11 2 5 9 3 6 17 ...
## $ ville : Factor w/ 22 levels "Ajaccio","Amiens",...: 11 1 2 3 4 5 6 7 8 9 ...
## $ latitude : num 45.7 41.9 49.9 47.2 44.8 ...
## $ longitude: num 4.847 8.733 2.3 6.033 -0.567 ...
```

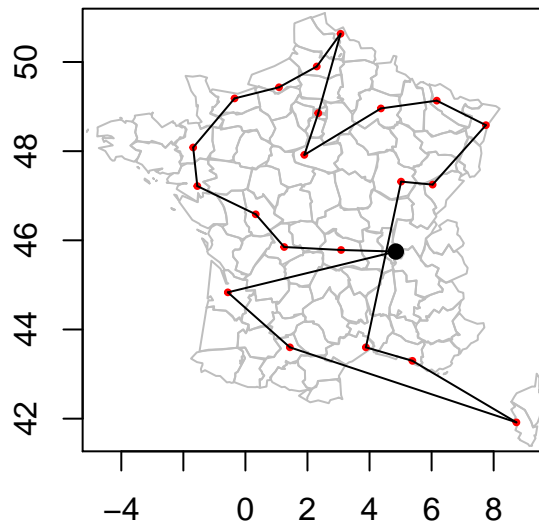
Représentation des chemins par plus proches voisins et du chemin optimal :

```
coord <- cbind(villes$longitude,villes$latitude)
dist <- distanceGPS(coord)
voisins <- TSPnearest(dist)

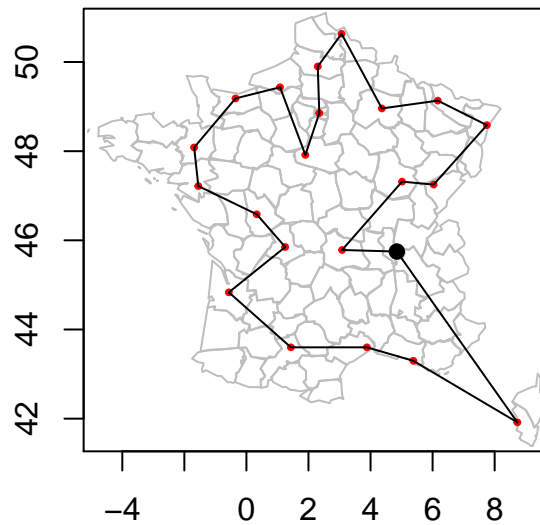
pathOpt <- c(1,8,9,4,21,13,7,10,3,17,16,20,6,19,15,18,11,5,22,14,12,2)

par(mfrow=c(1,2),mar=c(1,1,2,1))
plotTrace(coord[voisins$chemin,], title='Plus proches voisins')
plotTrace(coord[pathOpt,], title='Chemin optimal')
```

Plus proches voisins



Chemin optimal



Les longueurs des trajets (à vol d'oiseau) valent respectivement, pour la méthode des plus proches voisins :

```
## [1] 4303.568
```

et pour la méthode optimale :

```
## [1] 3793.06
```

Ceci illustre bien l'intérêt d'un algorithme de voyageur de commerce. Nous allons dans la suite étudier les performances de cet algorithme.

## 1. Comparaison d'algorithmes

Nombre de sommets fixes et graphes "identiques".

```
n <- 10
sommets <- data.frame(x = runif(n), y = runif(n))
couts <- distance(sommets)
```

### 1.1. Longueur des chemins

Comparaison des longueurs de différentes méthodes :

- boxplots
- test entre 'nearest' et 'branch'
- tests 2 à 2

## 1.2. Temps de calcul

Comparaison des temps à l'aide du package microbenchmark.

Exemple d'application de microbenchmark :

```
microbenchmark(sqrt(x), x^0.5, times=100, setup={x <- runif(1)})
```

```
## Unit: nanoseconds
##      expr min  lq   mean median    uq  max neval cld
##  sqrt(x) 300 402 603.97    501 601.0 4201   100   a
##    x^0.5 600 701 931.04    801 901.5 5601   100   b
```

## 2. Etude e la complexité de l'algorithme Branch and Bound

### 2.1. Comportement par rapport au nombre de sommets : premier modèle

Récupération du temps sur 10 graphes pour différentes valeurs de  $n$ .

Ajustement du modèle linéaire de  $\log(\text{temps})^2$  en fonction de  $n$ .

Analyse de la validité du modèle :

- pertinence des coefficients et du modèle,
- étude des hypothèses sur les résidus.

### 2.2. Comportement par rapport au nombre de sommets : étude du comportement moyen

Récupération du temps moyen.

Ajustement du modèle linéaire de  $\log(\text{temps.moy})^2$  en fonction de  $n$ .

Analyse de la validité du modèle :

- pertinence des coefficients et du modèle,
- étude des hypothèses sur les résidus.

### 2.3. Comportement par rapport à la structure du graphe

Lecture du fichier 'DonneesTSP.csv'.

Ajustement du modèle linéaire de  $\log(\text{temps.moy})^2$  en fonction de toutes les variables présentes. Modèle sans constante.

Mise en œuvre d'une sélection de variables pour ne garder que les variables pertinentes.

Analyse de la validité du modèle :

- pertinence des coefficients et du modèle,
- étude des hypothèses sur les résidus.