

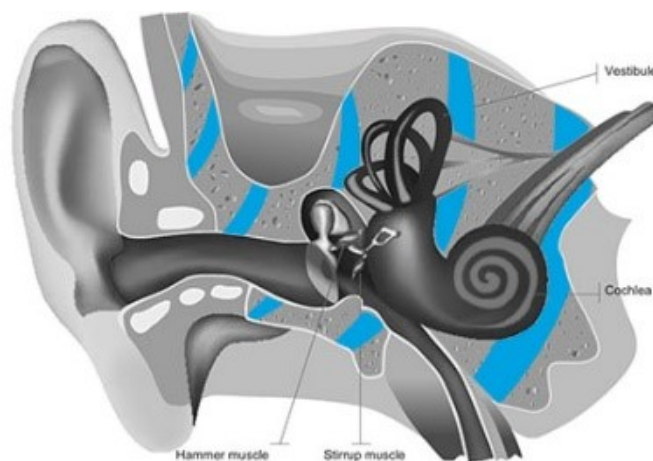
UNIVERSITÉ DE TOULOUSE III

RAPPORT

---

## Étude de la cochlée

---



*Auteurs :*

M. Max HALFORD

M. Axel BELLEC

*Responsable :*

M. Jean-Michel LOUBES

## Table des matières

<b>1</b>	<b>Objectif</b>	<b>2</b>
<b>2</b>	<b>Représentation initiale</b>	<b>3</b>
<b>3</b>	<b>Démarche</b>	<b>4</b>
<b>4</b>	<b>Code Python</b>	<b>5</b>
4.1	Modules requis . . . . .	5
4.2	Ouverture des données . . . . .	5
4.3	Normalisation des données . . . . .	5
4.4	Calcul des distances individuelles . . . . .	6
4.5	Calcul des distances moyennes . . . . .	6
4.6	Représentation des distances . . . . .	7
<b>5</b>	<b>Analyse avancée des distances</b>	<b>8</b>
<b>6</b>	<b>Application de la méthode <i>Multidimensional Scaling</i></b>	<b>13</b>
	<b>Conclusion</b>	<b>15</b>
	<b>Références</b>	<b>16</b>
	<b>Annexes</b>	<b>17</b>

# 1 Objectif

Nous disposons de données sur la *cochlée*<sup>1</sup> gauche de 22 individus (12 hommes et 10 femmes). Pour chaque individu, on dispose de 1000 points dans  $\mathbb{R}^3$ . On peut donc représenter chaque cochlée dans un plan cartésien. Notre but est de savoir si on peut établir une différence significative entre les cochlées des hommes et celles des femmes.

---

1. <http://fr.wiktionary.org/wiki/cochlée>

## 2 Représentation initiale

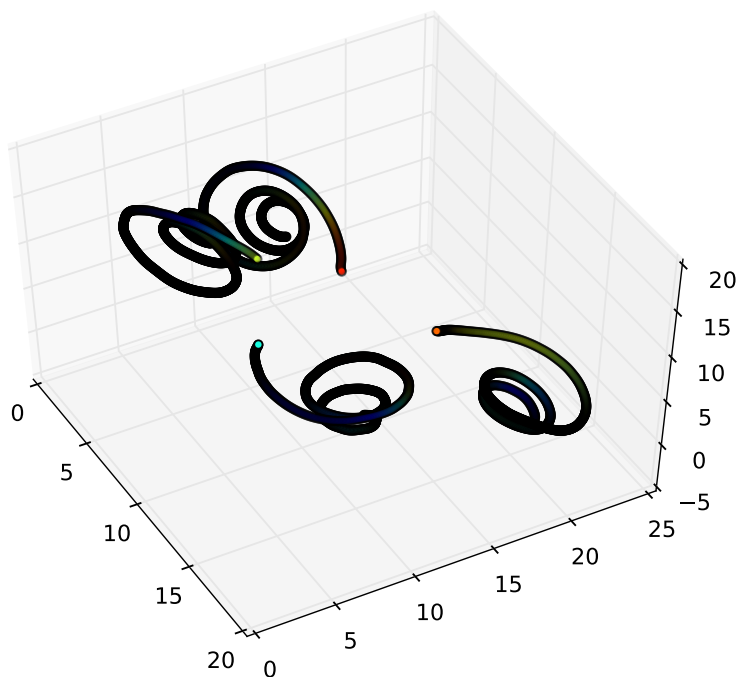


FIGURE 1 – Représentation de 4 cochlées

On se rend compte que les cochlées ne sont pas forcément situées au même endroit dans une oreille. En effet même si elles peuvent être exactement similaires, elles peuvent se situer autrepars. Pour résoudre ce problème on peut transformer les données pour que toutes les cochlées se trouvent dans un même repère normalisé. Par exemple en opérant le changement  $\frac{p-\min(p)}{\max(p)-\min(p)}$  sur chaque coordonnée les points seraient dans un repère  $[0, 1]^3$ . Cependant si l'on centre et l'on réduit les coordonnées on conserve plus de variance entre les points, ce qui est une bonne chose.

### 3 Démarche

On pourrait étudier plusieurs mesures. Par exemple on pourrait étudier l'orientation des cochlées ou bien l'étendue des points. On décide d'étudier la distance entre tous les points. C'est à dire qu'à chaque point on associe un vecteur  $\mathbb{R}^{1000}$  de distances. Il y'a 1000 points par individus donc on aura une matrice de distances qui prend ses valeurs dans  $\mathbb{R}^{1000 \times 1000}$ .

On cherche à étudier la différence entre les hommes et les femmes. Au lieu d'étudier les 22 matrices qui prennent leurs valeurs dans  $\mathbb{R}^{1000 \times 1000}$  on va plutôt calculer une matrice des distances moyennes pour chaque sexe et comparer ces deux matrices.

L'approche qu'on fait est assez intuitive, en effet on possède deux matrices qui nous donnent 1000000 de distances, la question est simplement : y'a-t-il des distances significatives ? Si oui lesquelles et à quelle amplitude ?

## 4 Code Python

### 4.1 Modules requis

```
import pandas as pd
import os
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
5 from scipy.spatial.distance import pdist, squareform
from sklearn.decomposition import PCA as sklearnPCA
import numpy as np
import scipy.stats as stats
```

Listing 1 – Modules

### 4.2 Ouverture des données

```
data = {'male' : [], 'female' : []}
maleFiles = [f for f in os.listdir(os.getcwd() + '/data/
male/')]
femaleFiles = [f for f in os.listdir(os.getcwd() + '/data/
/female/')]
c = ['x', 'y', 'z']
5 data['male'] = [pd.read_table('data/male/'+m, delimiter=r
"s+", names=c).dropna() for m in maleFiles]
data['female'] = [pd.read_table('data/female/'+f,
delimiter=r"s+", names=c).dropna() for f in
femaleFiles]
```

Listing 2 – Ouverture

On crée deux tableaux de données, un pour les hommes, l'autre pour les femmes.

### 4.3 Normalisation des données

```
normalize = lambda df: (df - df.mean()) / (df.var())
data['male'] = [normalize(indi) for indi in data['male']]
data['female'] = [normalize(indi) for indi in data['
female']]
```

Listing 3 – Normalisation

On normalise les données en les centrant et les réduisant, de cette façon elles conservent plus de variance qu'il n'en aurait eu si l'on avait utilisé  $\frac{p - \min(p)}{\max(p) - \min(p)}$ .

#### 4.4 Calcul des distances individuelles

```
maleDist = [squareform(pdist(male)) for male in data['  
male']]  
femaleDist = [squareform(pdist(female)) for female in  
data['female']]
```

Listing 4 – Distances individuelles

On utilise la fonction `squareform` de `scipy` qui fait exactement ce que fait la fonction `dist` dans R. On a maintenant 22 matrices  $1000 \times 1000$ .

#### 4.5 Calcul des distances moyennes

```
meanMaleDist = sum(maleDist) / len(maleDist)  
meanFemaleDist = sum(femaleDist) / len(femaleDist)  
difference = meanMaleDist - meanFemaleDist
```

Listing 5 – Distances moyennes

On calcule les matrices des distances moyennes de chaque sexe en les sommant et en les divisant par le nombre d'individus de chaque sexe.

## 4.6 Représentation des distances

On peut soustraire une des matrices à l'autre pour obtenir une matrice des différences. Dans notre cas nous soustrayons la matrice féminine à la masculine. On peut alors représenter la matrice obtenue graphiquement.

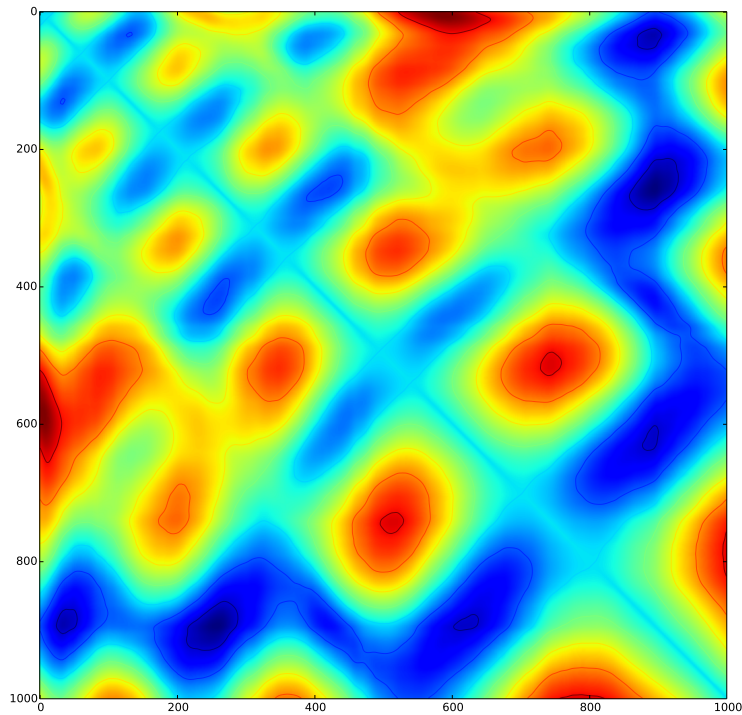


FIGURE 2 – Différences entre les distances moyennes des sexes

On constate qu'il y'a des zones "chaudes" et "froides" qui indiquent des différences conséquentes entre les cochlées des deux sexes. Évidemment la matrice que nous avons calculée est symétrique donc on ne prend en compte qu'un seul côté (on prend la diagonale nord-est). La façon d'interpréter le graphique est la suivante : si une zone est chaude cela indique que la distance entre les deux points est plus forte chez les hommes que chez les femmes et vice-versa. Cela provient de la construction de la matrice, on a soustrait celle des femmes à celle des hommes.



## 5 Analyse avancée des distances

Avec matplotlib on peut récupérer des coordonnées en survolant le graphique précédent avec la souris d'un ordinateur. Cette astuce n'est pas la plus précise mais elle a le mérite d'être rapide. Une piste pour obtenir les sommets de la matrice serait de faire une estimation de densité du noyau à deux dimensions, puis de faire un algorithme type "hill climbing" pour trouver les sommets. En survolant les zones les plus chaudes et les plus froides on obtient les points suivants :

- $(x_1, y_1) = (890, 521)$ , zone froide.
- $(y_2, y_2) = (343, 196)$ , zone chaude.
- $(x_3, y_3) = (416, 255)$ , zone froide.
- $(x_4, y_4) = (519, 348)$ , zone chaude.
- $(x_5, y_5) = (744, 512)$ , zone chaude.
- $(x_6, y_6) = (604, 12)$ , zone chaude.
- $(x_7, y_7) = (899, 255)$ , zone froide.
- $(x_8, y_8) = (893, 38)$ , zone froide.

Ces points correspondent aux distances entre deux points qui sont les plus significatifs entre la cochlée féminine moyenne et la cochlée masculine moyenne. On peut alors représenter les liens entre ces points sur les cochlées des individus. Sur le graphique suivant on compare une cochlée féminine à une cochlée masculine, le code couleur est le même pour chaque lien. C'est un peu difficile à voir sur une image statique mais la différence est nette, les liens plus ou moins grands selon la cochlée. Voici le code utilisé, le graphique produit est interactif si on exécute le script.

```
p1 = (890, 343, 416, 519, 744, 604, 899, 893)
p2 = (521, 196, 255, 348, 512, 12, 255, 38)

ax = plt.axes(projection='3d')
5 cochleas = [pData['male'][0], pData['female'][0]]
for d in cochleas:
    ax.scatter(d['x'], d['y'], d['z'], c = d['x']*d['y']*
              d['z'])
    10 for i in range(len(p1)):
        x = (d.iloc[p1[i]]['x'], d.iloc[p2[i]]['x'])
        y = (d.iloc[p1[i]]['y'], d.iloc[p2[i]]['y'])
        z = (d.iloc[p1[i]]['z'], d.iloc[p2[i]]['z'])
        ax.plot(x, y, z)
plt.show()
```

Listing 6 – Liens entres les points significatifs

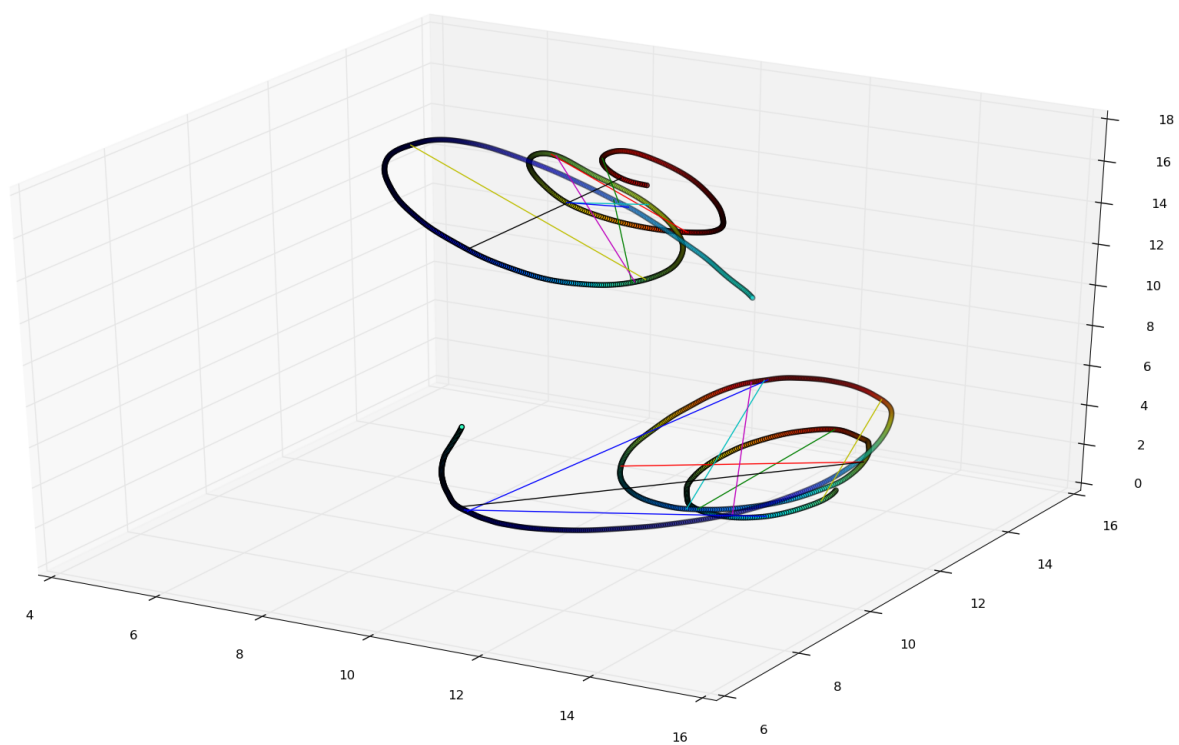


FIGURE 3 – Différences entre les distances significatives des cochlées de deux individus de sexes différents

Afin de gagner en précision on peut comparer les distances entre les points pour chaque individu. Celles-ci sont résumées dans le tableau de la page 12. Le tableau ci-dessous est simplement une transformation plus simple à interpréter du tableau de la page 12.

On voit bien que pour les points 1, 3, 7 et 8 (les points froids), les distances entre les points sont plus fortes chez les femmes que chez les hommes. Le contraire est vrai pour les points 2, 4, 5 et 6. Nous n'avons pas beaucoup de données mais nous pouvons essayer d'établir un intervalle de confiance pour chaque distance. En effet il suffit de calculer l'écart-type des distances pour chaque point, le quantile à 5% d'une loi de Student (car on estime la variance) et le tour est joué.

gender	distance	mean	lowerBound	upperBound
M	(890, 521)	1.62066	1.49418	1.74713
F	(890, 521)	1.6289	1.52591	1.73188
M	(343, 196)	1.26257	1.1635	1.36164
F	(343, 196)	1.11936	1.03747	1.20125
M	(416, 255)	1.21335	1.08208	1.34462
F	(416, 255)	1.26992	1.16845	1.37139
M	(519, 348)	1.69947	1.4726	1.92635
F	(519, 348)	1.51372	1.40298	1.62446
M	(744, 512)	2.11749	1.91147	2.3235
F	(744, 512)	1.91161	1.76409	2.05913
M	(604, 12)	1.39497	1.16596	1.62399
F	(604, 12)	1.16454	1.04637	1.2827
M	(899, 255)	2.05107	1.85157	2.25057
F	(899, 255)	2.17537	2.03231	2.31843
M	(893, 38)	2.08935	1.85069	2.32801
F	(893, 38)	2.19922	2.00062	2.39783

On constate que seuls les intervalles 5 et 6 sont ceux qui se chevauchent le moins. On pourrait peut-être prendre les distances de chaque individu et voir dans quel intervalle de confiance elles sont pour déterminer si c'est un homme ou une femme. Cependant nous ne le faisons pas ici car on ne peut pas appliquer le modèle aux données qui ont construit le modèle, les conclusions seraient biaisées.

Le code pour produire les deux tableaux est le suivant.

```

index = ['M' + str(i) for i in range(0, 12)] + \
        ['F' + str(i) for i in range(0, 10)] + \
        ['Mean_M', 'Mean_F']
columns = [p for p in zip(p1, p2)]
5 distances = pd.DataFrame(0.0, index=index, columns=
    columns)
for p in zip(p1, p2):
    for i, _ in enumerate(maleDist):
        distances[p]['M' + str(i)] = round(maleDist[i][p
            [0], p[1]], 5)
    for i, _ in enumerate(femaleDist):
10         distances[p]['F' + str(i)] = round(femaleDist[i][
            p[0], p[1]], 5)
        distances[p]['Mean_M'] = round(meanMaleDist[p[0], p
            [1]], 5)
        distances[p]['Mean_F'] = round(meanFemaleDist[p[0], p
            [1]], 5)
    distances.to_csv('distances')

15 def meanConfidenceInterval(data, confidence=0.95):
    a = 1.0 * np.array(data)
    n = len(a)
    m, se = np.mean(a), stats.sem(a)
    h = se * stats.t._ppf((1 + confidence)/2., n-1)
20     return round(m, 5), round(m-h, 5), round(m+h, 5)

maleIndex = index[:12]
femaleIndex = index[12:-2]
ciDf = pd.DataFrame()

25 for column in distances.columns:
    list = []
    for i in maleIndex:
        list.append(distances[column][i])
30     ci = meanConfidenceInterval(list)
    row = pd.Series(['M', str(column), ci[0], ci[1], ci
        [2]])
    ciDf = ciDf.append(row, ignore_index=True)
    list = []
    for i in femaleIndex:
35         list.append(distances[column][i])
        ci = meanConfidenceInterval(list)
        row = pd.Series(['F', str(column), ci[0], ci[1], ci
            [2]])
        ciDf = ciDf.append(row, ignore_index=True)

40 ciDf.columns = ('gender', 'distance', 'mean', 'lowerBound
    ', 'upperBound')
ciDf.to_csv('confidence_intervals', index=False)

```

Listing 7 – Récapitulatif et intervalle de confiance des distances pour chaque individu

	(890, 521)	(343, 196)	(416, 255)	(519, 348)	(744, 512)	(604, 12)	(899, 255)	(893, 38)
M0	1.9772	1.23013	1.56453	1.37227	1.82482	1.20458	2.52581	2.16332
M1	1.17528	1.53457	0.88412	1.31624	2.10429	2.04307	1.39848	1.3508
M2	1.65956	1.19116	1.39987	1.52213	1.87308	1.27328	2.36906	2.17414
M3	1.65777	1.10235	1.37031	1.46622	1.78874	1.30028	2.27705	2.44771
M4	1.49858	1.42524	1.01225	1.9508	2.39588	1.21812	1.85713	2.04835
M5	1.52083	1.1065	1.1516	1.4493	1.80795	1.06183	2.13423	2.21725
M6	1.58798	1.2429	1.43505	2.45886	2.57643	2.22189	2.16469	2.53833
M7	1.65627	1.38274	1.12011	1.88809	2.44465	1.34312	1.70537	1.58569
M8	1.83097	1.16834	1.26716	1.40135	1.83444	1.19887	2.03702	1.69655
M9	1.66439	1.01735	1.27359	1.54902	1.82785	1.35996	2.30348	2.53894
M10	1.73971	1.40469	1.11221	1.90786	2.41512	1.11122	1.97492	2.03945
M11	1.47937	1.34487	0.9694	2.11156	2.51657	1.40348	1.86559	2.27166
F0	1.83634	1.24096	1.23248	1.46654	1.9293	0.94079	1.99342	1.6874
F1	1.55294	1.16145	1.41188	1.69097	2.17748	1.39299	2.06042	1.89264
F2	1.40873	1.30376	1.06602	1.65388	2.14376	1.29122	1.83461	1.96812
F3	1.60163	0.969	1.31643	1.31152	1.61785	0.99583	2.28345	2.32455
F4	1.50586	1.01005	1.30521	1.46741	1.96137	1.13496	2.16205	2.181
F5	1.81196	1.12614	1.5077	1.51188	1.9063	1.19223	2.50924	2.58471
F6	1.50944	1.02942	1.13272	1.49102	1.67442	1.277	2.15361	2.38673
F7	1.7791	1.11093	1.38722	1.39111	1.74488	1.05552	2.43002	2.47505
F8	1.6046	1.23186	1.22153	1.79544	2.17469	1.37212	2.22891	2.29137
F9	1.67836	1.01	1.11802	1.35741	1.78603	0.9927	2.09799	2.20068
Mean_M	1.62066	1.26257	1.21335	1.69947	2.11748	1.39497	2.05107	2.08935
Mean_F	1.6289	1.11936	1.26992	1.51372	1.91161	1.16454	2.17537	2.19923

## 6 Application de la méthode *Multidimensional Scaling*

Une fois notre étude terminée, nous souhaiterions savoir si nous n'avons pas perdu trop d'information. Nous décidons alors d'appliquer le Multidimensional Scaling. Autrement appelée "positionnement multidimensionnel", la méthode MDS permet d'obtenir une meilleure représentation en dimension  $p$  par des distances euclidiennes. Il faut noter que si  $D = (||X_i - X_j||)$  est une distance euclidienne, alors la MDS est équivalente à l'ACP (Analyse en Composantes Principales).

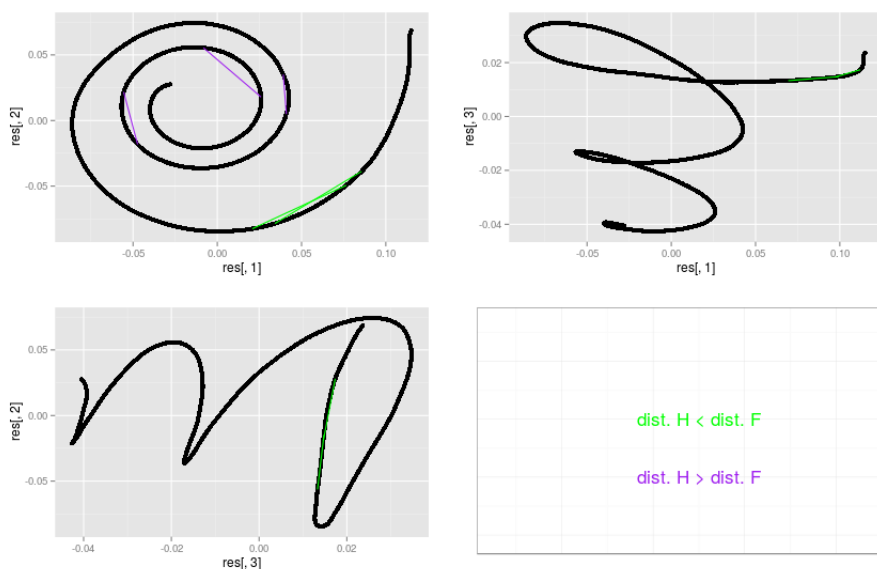


FIGURE 4 – Représentation d'une cochlée de femme (en moyenne)

La méthode MDS nous a permis de reconstituer les cochlées moyennes des hommes et des femmes de notre échantillon. On peut constater quelques légères différences entre les représentations des cochlées moyennes même si cela n'est pas flagrant.

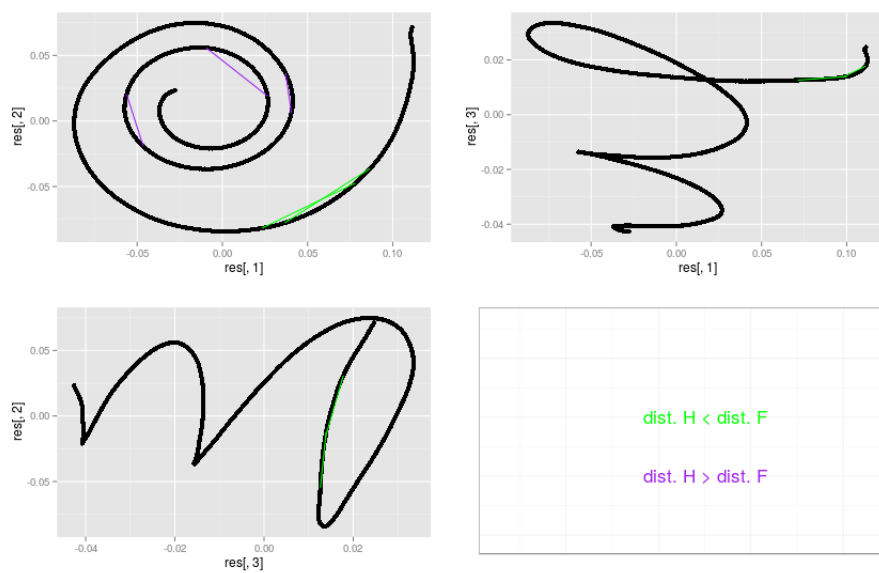


FIGURE 5 – Représentation d'une cochlée d'homme (en moyenne)

## Conclusion

L'étude de cas statistique que nous avons menée sur la cochlée nous a permis de pousser notre réflexion et de penser en tant que statisticien. Avant d'établir l'ensemble des scripts en langage R et Python il a d'abord fallu modéliser les problèmes rencontrés et les solutions que nous pouvions apporter. Nous avons donc dû à chaque étape réfléchir aux traitements que nous pouvions mettre en place. Cette étude a démontré la puissance des outils informatiques pour étudier un cas statistique dans un cadre médical. Nous avons initialement pu observer des cochlées dans un espace de dimension trois et après traitements nous sommes arrivés à visualiser des cochlées moyennes selon le genre. Nous sommes également parvenus à déceler certaines différences significatives concernant les distances entre deux points. En effet, certains intervalles de confiance pour les distances entre deux points sont relativement différents selon les sexes. Il aurait été intéressant d'accroître la taille de l'échantillon pour observer si ce constat était généralisable.



## Références

[WEB] <https://github.com/MaxHalford/Cochleas>

## Annexes

```
#~~~ PROGRAMME COCHLEE ~~~#

# Packages
install.packages("plot3D")
install.packages("vegan")
library("plot3D")
library("vegan")

# Importation des fichiers de donnees pour les Hommes et
# les Femmes
h1=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/002
  _Xt_Hs_EMBR136_6y_M_R_JB_Smooth_JB.am")
h2=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/003
  _Xt_Hs_EMBR383_5y4m_M_Lm_VP_Smooth_VP.am")
h3=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/006
  _Xt_Hs_EMBR473_3y4m_M_R_VP_Smooth_VP.am")
h4=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/008
  _Xt_Hs_EMBR205_3y_M_R_VP_Smooth_VP.am")
h5=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/029
  _Xt_Hs_EMBR168_0_M_R_VP_Smooth_VP.am")
h6=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/030
  _Xt_Hs_EMBR179_10y_M_R_VP_Smooth_VP.am")
h7=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/031
  _Xt_Hs_EMBR215_7m_M_R_VP_Smooth_VP.am")
h8=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/032
  _Xt_Hs_EMBR249_0_M_R_VP_Smooth_VP.am")
h9=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/033
  _Xt_Hs_EMBR277_6m_M_L_JB_Smooth.am")
h10=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/034
  _Xt_Hs_EMBR323_0_M_R_VP_Smooth_VP.am")
h11=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/035
  _Xt_Hs_EMBR388_12m_M_R_VP_Smooth_VP.am")
h12=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/038
  _Xt_Hs_EMBR479_1y3m_M_R_VP_Smooth_VP.am")

f1=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/018
  _Xt_Hs_EMBR384_2y1m_F_L_JB.am")
```

```

f2=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/020
  _Xt_Hs_EMBR121_5y_F_R_VP_Smooth_VP.am")
25 f3=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/023
  _Xt_Hs_EMBR212_5y_F_R_VP_Smooth_VP.am")
f4=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/024
  _Xt_Hs_EMBR281_1y10m_F_R_VP_Smooth_VP.am")
f5=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/036
  _Xt_Hs_EMBR382_9m_F_L_JB_Smooth.am")
f6=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/037
  _Xt_Hs_EMBR385_5m_F_L_VP_Smooth_VP.am")
f7=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/039
  _Xt_Hs_EMBR308_2m15d_F_R_VP_Smooth_VP.am")
30 f8=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/040
  _Xt_Hs_EMBR513_7m_F_R_JB_Smooth.am")
f9=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/041
  _Xt_Hs_EMBR576_7m_F_R_JB_Smooth_VP.am")
f10=read.table("/home/axelbellec/Bureau/L3 SID/
  EtudeCasStat/Donnees cours cas statistiques/042
  _Xt_Hs_EMBR583_2m_F_R_VP_Smooth_VP.am")

# Graphes 3D sur 3 h et 3 f
35 par(mfcol=c(3,2),pty="s",mar=rep(0.1,4))
scatter3D(h1[,1],h1[,2],h1[,3])
scatter3D(h2[,1],h2[,2],h2[,3])
scatter3D(h3[,1],h3[,2],h3[,3])
scatter3D(f1[,1],f1[,2],f1[,3])
40 scatter3D(f2[,1],f2[,2],f2[,3])
scatter3D(f3[,1],f3[,2],f3[,3])
par(mfrow=c(1,1))

# Construction des listes h et f
45 h=list(h1,h2,h3,h4,h5,h6,h7,h8,h9,h10,h11,h12)

f=list(f1,f2,f3,f4,f5,f6,f7,f8,f9,f10)

# Calcul des distances 2 a 2
50 dh=vector("list",12)
for (i in 1:12)
  dh[[i]]=as.matrix(dist(h[[i]]))

df=vector("list",10)
55 for (i in 1:10)
  df[[i]]=as.matrix(dist(f[[i]]))

# Graphes des distances
par(mfcol=c(3,2),pty="s",mar=rep(1,4))

```

```

60 image(dh[[1]])
   image(dh[[2]])
   image(dh[[3]])
   image(df[[1]])
   image(df[[2]])
65 image(df[[3]])
   par(mfrow=c(1,1))

   # Distances entre 2 points consecutifs
   par(mfcol=c(3,2))
70 hist(diag(dh[[1]][-1000,-1]),30)
   abline(v=median(diag(dh[[1]][-1000,-1])),col="red",lwd=2)
   hist(diag(dh[[2]][-1000,-1]),30)
   abline(v=median(diag(dh[[2]][-1000,-1])),col="red",lwd=2)
   hist(diag(dh[[3]][-1000,-1]),30)
75 abline(v=median(diag(dh[[3]][-1000,-1])),col="red",lwd=2)
   hist(diag(df[[1]][-1000,-1]),30)

   abline(v=median(diag(df[[1]][-1000,-1])),col="red",lwd=2)
   hist(diag(df[[2]][-1000,-1]),30)
80 abline(v=median(diag(df[[2]][-1000,-1])),col="red",lwd=2)
   hist(diag(df[[3]][-1000,-1]),30)
   abline(v=median(diag(df[[3]][-1000,-1])),col="red",lwd=2)
   par(mfrow=c(1,1))

85 # Normalisation... ou pas...
   for (i in 1:12)
       dh[[i]]=dh[[i]]/sum(diag(dh[[i]][-1000,-1]))

   for (i in 1:10)
90   df[[i]]=df[[i]]/sum(diag(df[[i]][-1000,-1]))

   # Calcul des matrices des distances moyennes
   Dh=matrix(0,nrow=1000,ncol=1000)
   for (i in 1:12)
95   Dh=Dh+dh[[i]]
   Dh=Dh/12

   Df=matrix(0,nrow=1000,ncol=1000)
   for (i in 1:10)
100  Df=Df+df[[i]]
   Df=Df/10

   # Graphe des distances moyennes
   par(mfrow=c(1,2),pty="s",mar=c(0,2,0,0.5))
105 image(Dh)
   image(Df)
   par(mfrow=c(1,1))

   # Tests de Student sur les distances 2 ? 2
110 ttests=matrix(1,nrow=1000,ncol=1000)
   for (i in 1:999)
       for (j in (i+1):1000) {
           les.h=c()

```

```

115     for (k in 1:12)
        les.h=c(les.h,dh[[k]][i,j])
        les.f=c()
        for (k in 1:10)
            les.f=c(les.f,df[[k]][i,j])
            ttests[j,i]=ttests[i,j]=t.test(les.h,les.f)$p.value
120     }
    dif=Dh-Df

    # Histogramme des differences : les distances des hommes
    # sont plus petites ???
    par(mar = c(6, 4, 4, 2))
125    hist(dif, main='Histogramme des differences')
    abline(v=median(dif),col="red",lwd=2)

    # Graphe des distances DE
    alpha=0.05
130    x11() #ouverture fenetre graphique
    par(pty="s")
    image(dif)
    contour(ttests<alpha,add=TRUE,nlevels=1,labels=alpha)
    legend("bottomright",col=heat.colors(12),legend=round(seq
        (min(dif),max(dif),,12),3),lwd=17,bg="white")
135    dev.off() #fermeture fenetre graphique

    # Extraction manuelle des points de difference
    x11() #ouverture fenetre graphique
    par(pty="s")
140    image(1:1000,1:1000,dif)
    contour(1:1000,1:1000,ttests<alpha,add=TRUE,nlevels=1,
        labels=alpha)

    # p1=locator(1)
    # p2=locator(1)
145    # p3=locator(1)
    # p4=locator(1)
    # p5=locator(1)
    # p6=locator(1)
    # p7=locator(1)
150

    dev.off() #fermeture fenetre graphique

    # p1x=round(p1$x*1000)
155    # p1y=round(p1$y*1000)
    # p2x=round(p2$x*1000)
    # p2y=round(p2$y*1000)
    # p3x=round(p3$x*1000)
    # p3y=round(p3$y*1000)
    # p4x=round(p4$x*1000)
160    # p4y=round(p4$y*1000)
    # p5x=round(p5$x*1000)
    # p5y=round(p5$y*1000)
    # p6x=round(p6$x*1000)

```

```

165 # p6y=round(p6$y*1000)
# p7x=round(p7$x*1000)
# p7y=round(p7$y*1000)

170 # Localisation des points de difference (points J.-M.
# Loubes)
#p1x=9;p1y=246
#p2x=13;p2y=613
#p3x=534;p3y=930
#p4x=388;p4y=609
175 #p5x=66;p5y=909
#p6x=62;p6y=376
#p7x=341;p7y=751

# 1\ Localisation des points de difference (points A.
# Bellec)
180 p1x=143;p1y=202
p2x=268;p2y=311
p3x=427;p3y=459
p4x=805;p4y=871
p5x=819;p5y=885
185 p6x=861;p6y=913
p7x=908;p7y=958

# 2\ Localisation des points de difference (points M.
# Halford)
p1x=890;p1y=521
p2x=343;p2y=196
190 p3x=416;p3y=255
p4x=519;p4y=348
p5x=744;p5y=512
p6x=899;p6y=255
195 p7x=893;p7y=38

x11() #ouverture fenetre graphique
par(pty="s")
image(dif)
200 contour(ttests<alpha,add=TRUE,nlevels=1,labels=alpha)
legend("bottomright",col=heat.colors(12),legend=round(seq
(min(dif),max(dif),,12),3),lwd=17,bg="white")
text(p1x/1000,p1y/1000,1,cex=1.5)
text(p2x/1000,p2y/1000,2,cex=1.5)
text(p3x/1000,p3y/1000,3,cex=1.5)
205 text(p4x/1000,p4y/1000,4,cex=1.5)
text(p5x/1000,p5y/1000,5,cex=1.5)
text(p6x/1000,p6y/1000,6,cex=1.5)
text(p7x/1000,p7y/1000,7,cex=1.5)
dev.off() #fermeture fenetre graphique

210 # Localisation des 6 points sur la cochlee d'un individu
x11() #ouverture fenetre graphique
ind=h9
scatter3D(ind[,1],ind[,2],ind[,3])

```

```

215 scatter3D(ind[c(p1x,p1y),1],ind[c(p1x,p1y),2],ind[c(p1x,
    p1y),3],type="l",add=TRUE,lwd=2,col="blue")
scatter3D(ind[c(p2x,p2y),1],ind[c(p2x,p2y),2],ind[c(p2x,
    p2y),3],type="l",add=TRUE,lwd=2,col="blue")
scatter3D(ind[c(p3x,p3y),1],ind[c(p3x,p3y),2],ind[c(p3x,
    p3y),3],type="l",add=TRUE,lwd=2,col="blue")
scatter3D(ind[c(p4x,p4y),1],ind[c(p4x,p4y),2],ind[c(p4x,
    p4y),3],type="l",add=TRUE,lwd=2,col="red")
scatter3D(ind[c(p5x,p5y),1],ind[c(p5x,p5y),2],ind[c(p5x,
    p5y),3],type="l",add=TRUE,lwd=2,col="red")
220 scatter3D(ind[c(p6x,p6y),1],ind[c(p6x,p6y),2],ind[c(p6x,
    p6y),3],type="l",add=TRUE,lwd=2,col="red")
scatter3D(ind[c(p7x,p7y),1],ind[c(p7x,p7y),2],ind[c(p7x,
    p7y),3],type="l",add=TRUE,lwd=2,col="red")
dev.off() #fermeture fenetre graphique

# Passage par cmdscale
225 # -> Application du Multi Dimensional Scaling
# = Meilleure representation en dimension p par des
# distances euclidiennes.
# Si  $D=(||X_i-X_j||)$  est une distance euclidienne,
# alors MDS  $\Leftrightarrow$  ACP

230 plotCMDScale=function(inDist) {
    res=cmdscale(inDist,3)
    rxyz=range(res)
    par(mfrow=c(2,2),mar=rep(2,4),pty="s")
    plot(res[,1],res[,2],xlim=rxyz,ylim=rxyz)
235 lines(res[c(p1x,p1y),1],res[c(p1x,p1y),2],lwd=2,col="
        blue")
    lines(res[c(p2x,p2y),1],res[c(p2x,p2y),2],lwd=2,col="
        blue")
    lines(res[c(p3x,p3y),1],res[c(p3x,p3y),2],lwd=2,col="
        blue")
    lines(res[c(p4x,p4y),1],res[c(p4x,p4y),2],lwd=2,col="
        red")
240 lines(res[c(p5x,p5y),1],res[c(p5x,p5y),2],lwd=2,col="
        red")
    plot(res[,3],res[,2],xlim=rxyz,ylim=rxyz)
    lines(res[c(p6x,p6y),3],res[c(p6x,p6y),2],lwd=2,col="
        red")
    lines(res[c(p7x,p7y),3],res[c(p7x,p7y),2],lwd=2,col="
        red")
    plot(res[,1],res[,3],xlim=rxyz,ylim=rxyz)
245 lines(res[c(p6x,p6y),1],res[c(p6x,p6y),3],lwd=2,col="
        red")
    lines(res[c(p7x,p7y),1],res[c(p7x,p7y),3],lwd=2,col="
        red")
    plot(1,type="n",bty="none",xaxt="none",yaxt="none")
    legend("center",lwd=3,col=c("red","blue"),legend=c("
        dist. H < dist. F","dist. H > dist. F"),cex=1.5)
}

250 plotCMDScale(dh[[1]])

```

```

plotCMDScale(dh[[2]])
plotCMDScale(dh[[3]])
plotCMDScale(dh[[4]])
255 plotCMDScale(dh[[5]])
plotCMDScale(dh[[6]])
plotCMDScale(dh[[7]])

plotCMDScale(df[[1]])
260 plotCMDScale(df[[2]])
plotCMDScale(df[[3]])
plotCMDScale(df[[4]])
plotCMDScale(df[[5]])
plotCMDScale(df[[6]])
265 plotCMDScale(df[[7]])

# Sur cochlees moyennes
x11()
plotCMDScale(Dh)
270 legend("bottom", legend="Cochlee Homme (moyenne)", cex=1,
      bty="n")
dev.off()

x11()
plotCMDScale(Df)
275 legend("bottom", legend="Cochlee Femme (moyenne)", cex=1,
      bty="n")
dev.off()

# cmdscale
280 # A 95% on represente les ecarts de distances entre les
      hommes et les femmes

dhf=c(dh,df)

dDist=matrix(0,nrow=22,ncol=22)
285
for (i in 1:21)
  for (j in (i+1):22)
    dDist[j,i]=dDist[i,j]=sqrt(sum((dhf[[i]]-dhf[[j]])^2)
    )

290 res=cmdscale(dDist,3,TRUE)
plot(res$eig,type="h")

par(mfrow=c(2,2))
plot(res$points[,1],res$points[,2],pch=c(rep(1,12),rep
(19,10)))
295 plot(res$points[,3],res$points[,2],pch=c(rep(1,12),rep
(19,10)))
plot(res$points[,1],res$points[,3],pch=c(rep(1,12),rep
(19,10)))
par(mfrow=c(1,1))

```



```

resh=cmdscale(dDist[1:12,1:12],3,TRUE)
300 plot(resh$eig,type="h")

par(mfrow=c(2,2))
plot(resh$points[,1],resh$points[,2],pch=rep(1,12))
plot(resh$points[,3],resh$points[,2],pch=rep(1,12))
305 plot(resh$points[,1],resh$points[,3],pch=rep(1,12))
par(mfrow=c(1,1))

resf=cmdscale(dDist[13:22,13:22],3,TRUE)
plot(resf$eig,type="h")
310

par(mfrow=c(2,2))
plot(resf$points[,1],resf$points[,2],pch=rep(19,10))
plot(resf$points[,3],resf$points[,2],pch=rep(19,10))
plot(resf$points[,1],resf$points[,3],pch=rep(19,10))
315 par(mfrow=c(1,1))

# hclust
res2=hclust(as.dist(dDist))
320 plot(res2,labels=c(rep("H",12),rep("F",10)))

# isomap
res3=isomap(dDist,k=7)
325 plot(res3$eig,type="h")

par(mfrow=c(2,2))
plot(res3$points[,1],res3$points[,2],pch=c(rep(1,12),rep(19,10)))
plot(res3$points[,3],res3$points[,2],pch=c(rep(1,12),rep(19,10)))
330 plot(res3$points[,1],res3$points[,3],pch=c(rep(1,12),rep(19,10)))
par(mfrow=c(1,1))

# CMDSCALE PLOT
#Cochlee moyenne femmes
335 resDf <- cmdscale(Df,3)
res=as.data.frame(resDf)
rxyz=range(res)

p1<-ggplot()+geom_point(data=res, aes(res[,1], res[,2]))+
340 geom_line(data=res, aes(res[c(p1x,p1y),1],res[c(p1x,p1y),2]), col="red")+
geom_line(data=res, aes(res[c(p2x,p2y),1],res[c(p2x,p2y),2]), col="red")+
geom_line(data=res, aes(res[c(p3x,p3y),1],res[c(p3x,p3y),2]), col="red")+
geom_line(data=res, aes(res[c(p4x,p4y),1],res[c(p4x,p4y),2]), col="blue")+
geom_line(data=res, aes(res[c(p5x,p5y),1],res[c(p5x,p5y),2]), col="blue")

```

```

345 p2<-ggplot()+geom_point(data=res, aes(res[,3], res[,2]))+
  geom_line(data=res, aes(res[c(p6x,p6y),3],res[c(p6x,p6y
    ),2]), col="blue")+
  geom_line(data=res, aes(res[c(p7x,p7y),3],res[c(p7x,p7y
    ),2]), col="blue")

350 p3<-ggplot()+geom_point(data=res, aes(res[,1], res[,3]))+
  geom_line(data=res, aes(res[c(p6x,p6y),1],res[c(p6x,p6y
    ),3]), col="blue")+
  geom_line(data=res, aes(res[c(p7x,p7y),1],res[c(p7x,p7y
    ),3]), col="blue")

p4 <- ggplot(data=res, aes(res[,1], res[,3]))+geom_blank
  ()+theme_bw()+
355 annotate("text", label = "dist. H < dist. F", x =
  0.025, y = 0, size = 6, colour = "blue")+
  annotate("text", label = "dist. H > dist. F", x =
  0.025, y = -0.02, size = 6, colour = "red")+
  theme(axis.title.x=element_blank(), axis.text.x=
    element_blank(), axis.ticks=element_blank())+
  theme(axis.title.y=element_blank(), axis.text.y=
    element_blank(), axis.ticks=element_blank())

360 multiplot(p1, p2, p3, p4, cols=2)

#Cochlee moyenne hommes
resDh <- cmdscale(Dh,3)
res=as.data.frame(resDh)
365 rxyz=range(res)

p1<-ggplot()+geom_point(data=res, aes(res[,1], res[,2]))+
  geom_line(data=res, aes(res[c(p1x,p1y),1],res[c(p1x,p1y
    ),2]), col="red")+
  geom_line(data=res, aes(res[c(p2x,p2y),1],res[c(p2x,p2y
    ),2]), col="red")+
370 geom_line(data=res, aes(res[c(p3x,p3y),1],res[c(p3x,p3y
    ),2]), col="red")+
  geom_line(data=res, aes(res[c(p4x,p4y),1],res[c(p4x,p4y
    ),2]), col="blue")+
  geom_line(data=res, aes(res[c(p5x,p5y),1],res[c(p5x,p5y
    ),2]), col="blue")

p2<-ggplot()+geom_point(data=res, aes(res[,3], res[,2]))+
375 geom_line(data=res, aes(res[c(p6x,p6y),3],res[c(p6x,p6y
    ),2]), col="blue")+
  geom_line(data=res, aes(res[c(p7x,p7y),3],res[c(p7x,p7y
    ),2]), col="blue")

p3<-ggplot()+geom_point(data=res, aes(res[,1], res[,3]))+
  geom_line(data=res, aes(res[c(p6x,p6y),1],res[c(p6x,p6y
    ),3]), col="blue")+
380 geom_line(data=res, aes(res[c(p7x,p7y),1],res[c(p7x,p7y
    ),3]), col="blue")

```

```
385 p4 <- ggplot(data=res, aes(res[,1], res[,3]))+geom_blank  
    ()+theme_bw()+  
    annotate("text", label = "dist. H < dist. F", x =  
      0.025, y = 0, size = 6, colour = "blue")+  
    annotate("text", label = "dist. H > dist. F", x =  
      0.025, y = -0.02, size = 6, colour = "red")+  
    theme(axis.title.x=element_blank(), axis.text.x=  
      element_blank(), axis.ticks=element_blank())+  
    theme(axis.title.y=element_blank(), axis.text.y=  
      element_blank(), axis.ticks=element_blank())  
  
multiplot(p1, p2, p3, p4, cols=2)
```

Listing 8 – Code R