

TP : Analyse en Composantes Principales

Pour éviter de surcharger le compte-rendu avec du code, on fournit le fichier R utiliser pour obtenir tous les résultats présents.

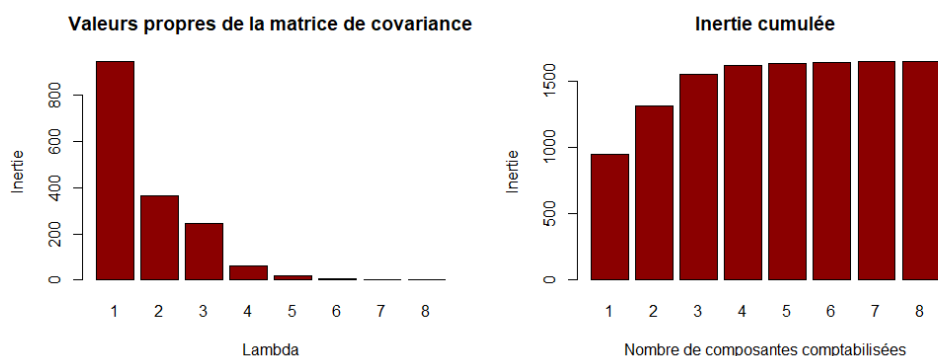
Partie I : ACP : Principes

1) On commence donc par importer les données sur R, ce qui nous donne une matrice. Cependant, cette matrice est formée de 10 colonnes, car elle contient l'indice de chaque ligne ainsi qu'une colonne à la fin qui ne nous est pas utile. On supprime donc les colonnes inutiles. On obtient donc une matrice contenant nos données qui se présente de la forme suivante :
Pour ce qui concerne les indicateurs qu'on nous demande de construire, on implémente les fonctions correspondantes.

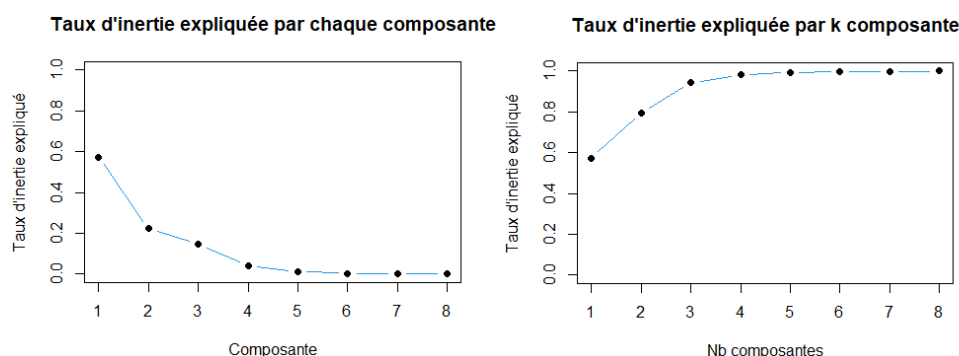
| | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 |
|----|--------|-------|------|------|---------|--------|----------|---------|
| 1 | 303.09 | 24.19 | 0.00 | 3.29 | 179.990 | 8.090 | 360.9000 | 120.330 |
| 2 | 281.88 | 38.59 | 4.29 | 1.06 | 192.000 | 10.500 | 353.5000 | 117.000 |
| 3 | 277.06 | 34.79 | 0.00 | 6.85 | 183.770 | 38.890 | 343.9500 | 114.650 |
| 4 | 276.38 | 32.43 | 4.14 | 2.04 | 190.790 | 38.530 | 341.1700 | 113.910 |
| 5 | 253.80 | 39.50 | 3.04 | 1.00 | 173.800 | 19.334 | 382.1100 | 127.373 |
| 6 | 243.56 | 34.39 | 2.79 | 3.43 | 166.670 | 27.590 | 391.1450 | 130.380 |
| 7 | 277.00 | 34.70 | 0.00 | 6.85 | 183.780 | 38.800 | 343.9400 | 114.650 |
| 8 | 294.80 | 28.29 | 1.85 | 1.83 | 182.290 | 10.290 | 360.2000 | 120.000 |
| 9 | 303.00 | 24.20 | 0.00 | 3.30 | 180.000 | 8.100 | 361.0000 | 120.340 |
| 10 | 269.38 | 36.89 | 2.99 | 1.03 | 197.700 | 12.590 | 359.4711 | 119.820 |
| 11 | 283.61 | 28.00 | 9.30 | 0.00 | 186.600 | 13.200 | 359.4180 | 119.806 |
| 12 | 290.32 | 23.20 | 0.80 | 2.34 | 172.400 | 39.400 | 353.6300 | 117.870 |
| 13 | 285.09 | 25.90 | 0.93 | 7.78 | 180.100 | 39.000 | 345.6900 | 115.230 |
| 14 | 265.48 | 40.39 | 0.95 | 5.14 | 184.390 | 38.830 | 348.5800 | 116.194 |
| 15 | 261.87 | 41.49 | 2.33 | 2.89 | 187.270 | 38.690 | 349.0620 | 116.345 |
| 16 | 274.38 | 29.79 | 6.69 | 0.00 | 183.580 | 38.890 | 349.9700 | 116.650 |
| 17 | 257.90 | 37.20 | 2.96 | 1.10 | 170.900 | 18.890 | 383.3000 | 127.769 |
| 18 | 238.20 | 29.80 | 2.60 | 0.80 | 166.705 | 14.140 | 410.1400 | 136.930 |
| 19 | 235.98 | 33.39 | 5.60 | 0.39 | 166.680 | 15.230 | 406.9700 | 135.650 |
| 20 | 247.77 | 36.69 | 5.03 | 1.79 | 166.680 | 27.090 | 386.2500 | 128.750 |
| 21 | 266.57 | 36.40 | 0.00 | 2.90 | 166.680 | 30.680 | 365.6000 | 121.880 |
| 22 | 264.79 | 24.19 | 1.19 | 5.60 | 166.680 | 39.690 | 391.9000 | 130.650 |
| 23 | 235.68 | 24.99 | 0.99 | 4.29 | 166.680 | 39.690 | 391.9700 | 130.650 |
| 24 | 239.58 | 47.89 | 0.40 | 4.19 | 166.680 | 39.690 | 376.0700 | 125.350 |
| 25 | 233.09 | 46.59 | 2.29 | 4.83 | 166.693 | 39.690 | 380.0800 | 126.690 |
| 26 | 241.37 | 34.50 | 5.15 | 0.39 | 166.683 | 39.690 | 383.7600 | 127.920 |

2) On implémente maintenant deux fonctions, qui vont pour la première centrer, et pour la seconde centrer et réduire notre nuage de points. On construit ensuite notre matrice de variance/covariance à partir de notre matrice d'individus centrée réduite. On diagonalise cette matrice en base orthornormée, et ainsi, les vecteurs propres et valeurs propres associées nous renseignent sur les hyperplans pour lesquels l'inertie projetée est maximale.

3) On s'intéresse maintenant à l'inertie expliquée de chacune des composantes principales, c'est-à-dire les valeurs propres de notre matrice de variance/covariance. On représente tout d'abord l'inertie expliquée par chacune des composantes (i.e. les valeurs propres), ainsi que l'inertie cumulée :



Pour avoir une meilleure vision de l'inertie expliquée, on représente également le taux d'inertie expliqué ainsi que le taux d'inertie expliqué par k composantes :



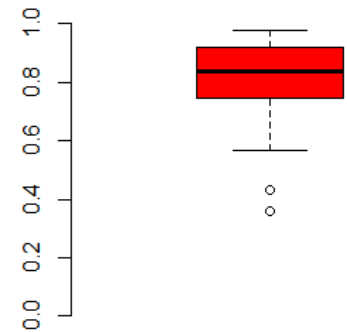
Pour déterminer le nombre de composantes qu'on va conserver, on va essayer de combiner plusieurs critères. Tout d'abord, on remarque que les trois premières composantes fournissent un taux d'inertie expliqué de 79.62 %. De plus, on peut chercher les cassures sur la courbe d'inertie expliquée par chaque composante (règle de Catell). On en remarque

une première sur le taux d'inertie expliquée par chaque composante entre la première et la seconde valeur propre, puis une seconde entre la troisième et quatrième. Ensuite, on peut également appliquer la règle de Kaiser-Guttman. On observe que les trois premières valeurs propres sont supérieures à 1, contrairement aux autres. Enfin, d'après la règle de Karlis-Saporta-Spinaki, le seuil des valeurs propres acceptables est de 1.06. Une fois encore, ce sont les trois premières valeurs qui correspondent.

Toutes ces raisons nous amènent à retenir les trois premières composantes principales.

4) Une fois les nouvelles coordonnées calculées, on calcule la qualité de chaque projection avec la formule donnée. Voici une boîte à moustache résumant la qualité de chaque projection :

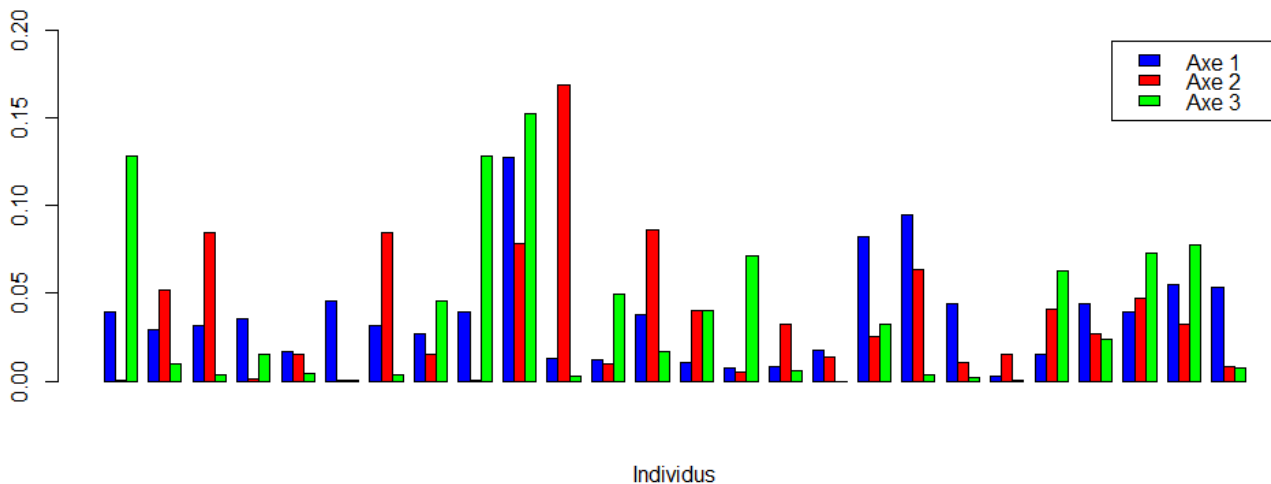
Qualité de la projection



Ainsi, la projection paraît relativement bonne : plus de 0.8 en moyenne, avec cela dit quelques valeurs pour lesquelles la qualité de projection est assez faible.

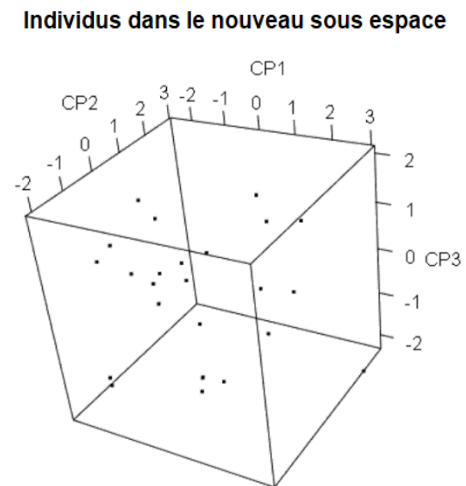
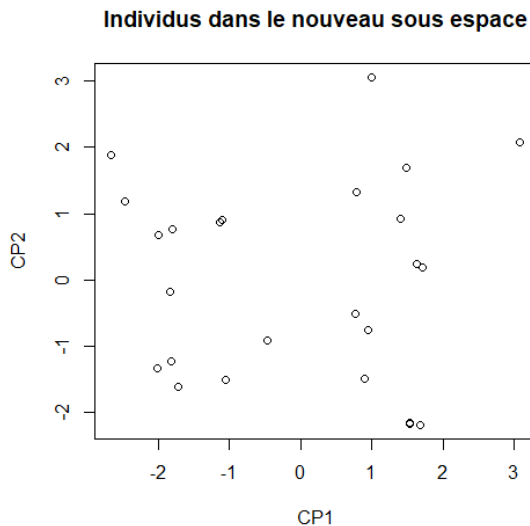
5) On calcule maintenant la contribution de chaque individu à chacun des trois axes factoriels retenus (avec la formule donnée). Voici les résultats obtenus (chaque groupement de trois barres correspond à un individu).

Contribution des individus par rapport aux trois axes



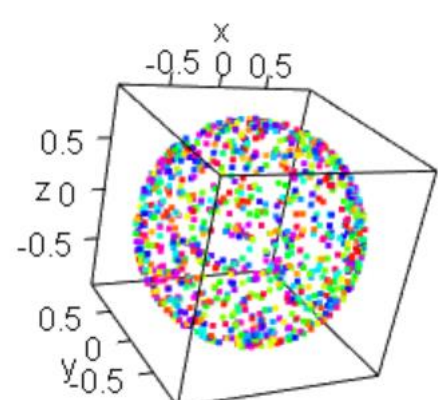
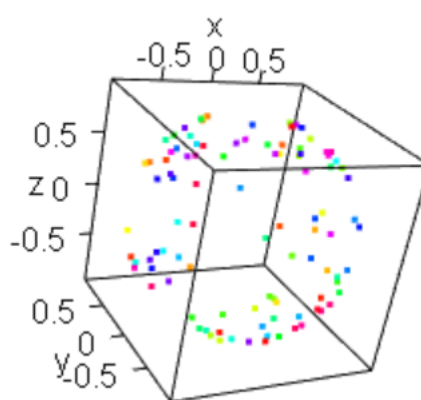
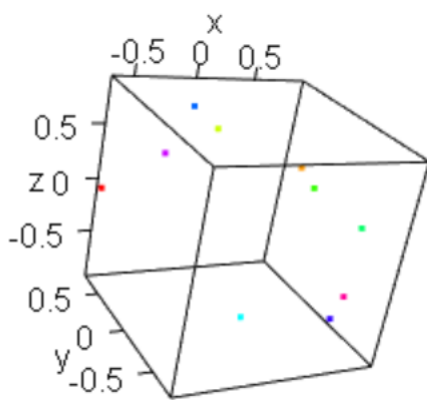
6) On va maintenant vérifier que la librairie R *ade4* effectue les mêmes opération que ce qu'on a programmé. Pour cela, on utilise la fonction *dudi.pca*. Afin de comparer notre projection à celle de cette fonction, on calcul la différence entre l'hyperplan de projection choisi précédemment et celui de la fonction. Le résultat obtenue étant très proche de 0, on conclut que notre premier plan factoriel est correcte.

7) On va maintenant représenter les individus dans le nouveau sous espace formé dans un premier temps des deux premiers axes principaux. On représente également les individus en 3 dimensions, dans l'espace de l'hyperplan retenu pour la projection.

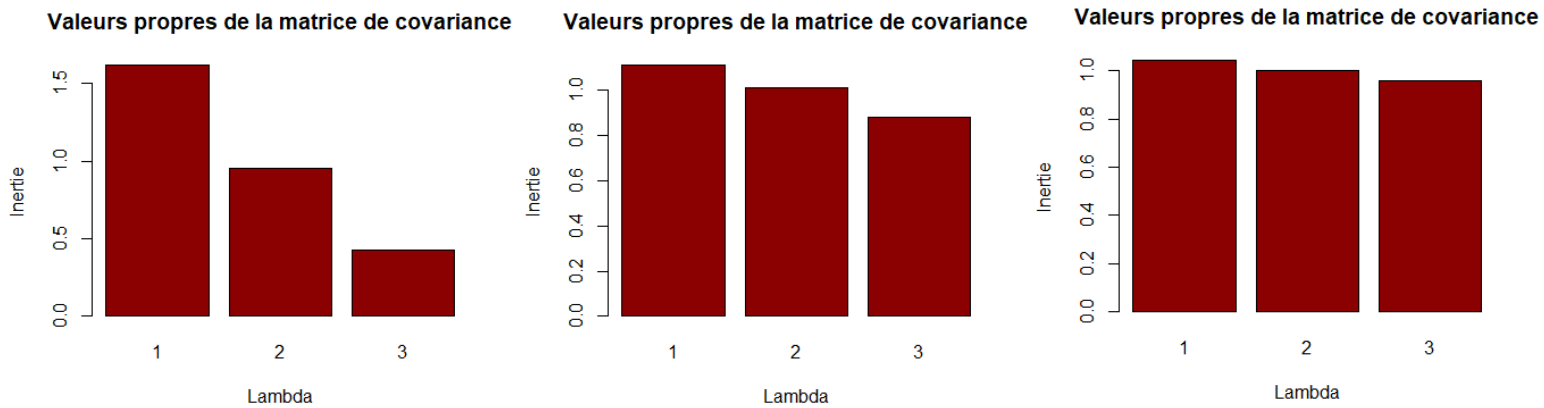


Partie II : ACP et étude de nuage de point

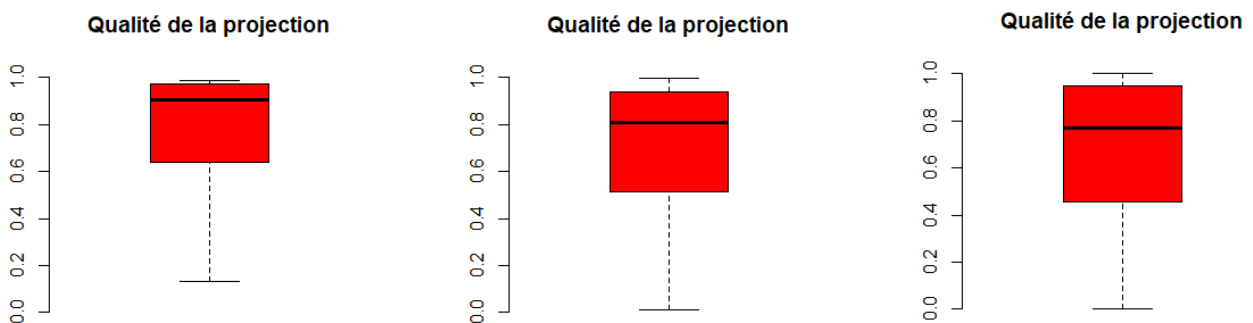
1) On commence par générer les données, en simulant trois vecteurs Gaussien de loi $N(0,1)$ comme cela est conseillé. On représente les données obtenus pour $n = 10$, $n = 100$ et $n = 1000$.



On réalise ensuite l'ACP. Voici les diagrammes représentant les valeurs propres pour les trois jeux de données précédentes :

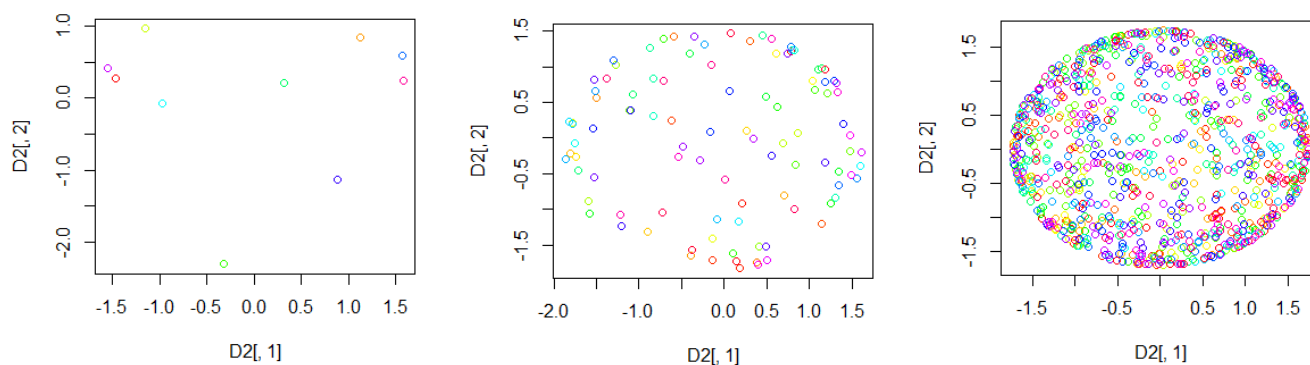


On observe que plus n augmente, plus les valeurs propres prennent des valeurs proches : cela indique que chaque composantes principales ont peu à peu des inerties équivalentes, ce qui signifie que l'ACP ne sera pas efficace. Dans les trois cas, on retient les deux premières composantes principales pour faire l'ACP. Ainsi, voici les boîtes à moustache des qualités de chaque projection :



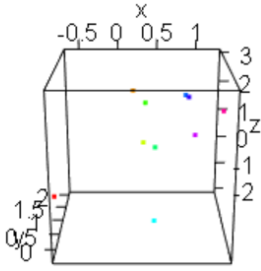
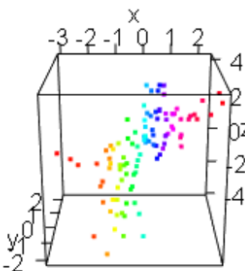
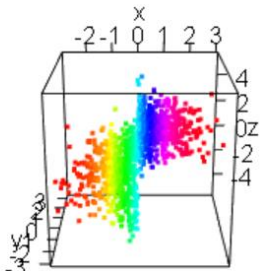
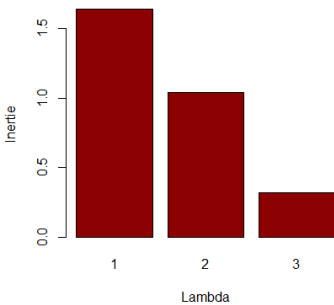
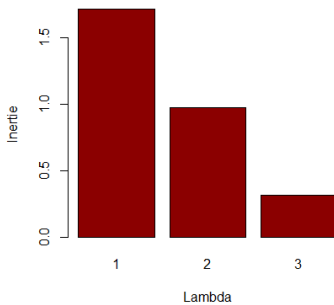
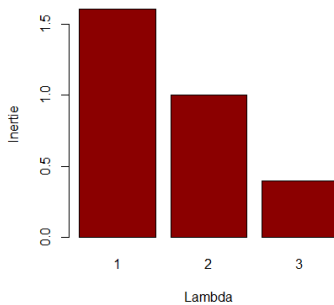
On remarque que, comme anticipé, la qualité de la projection a tendance à baisser lorsque n augmente. Ensuite, on voit que certaines valeurs ont des qualité de projection proche de 0 : il s'agit sans doute des valeurs dont la seule composante était selon la troisième composante principale (puisque l'on modélise une sphère, il est censé y avoir des points selon chaque composantes de l'espace).

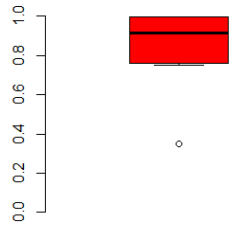
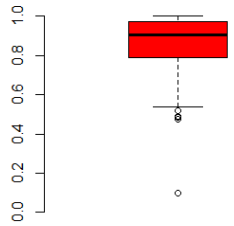
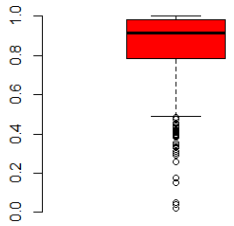
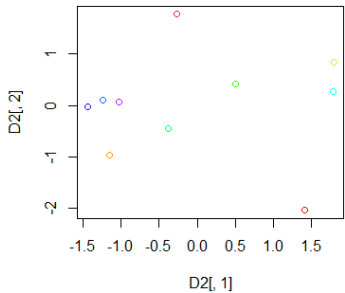
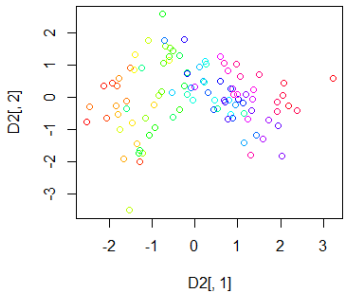
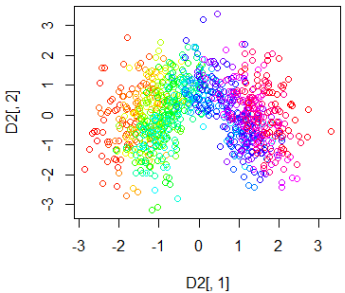
Enfin, voici le résultat de l'ACP, soit les points dans leurs nouvelles coordonnées



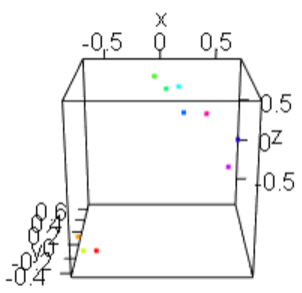
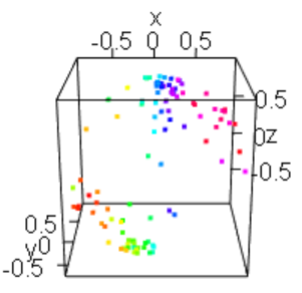
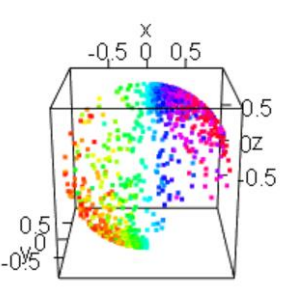
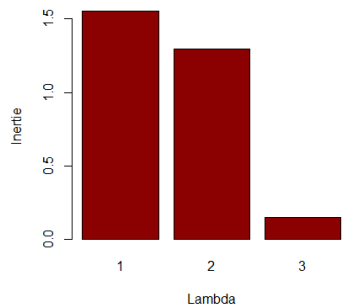
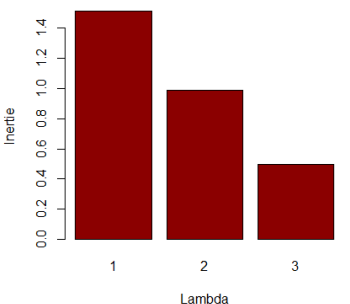
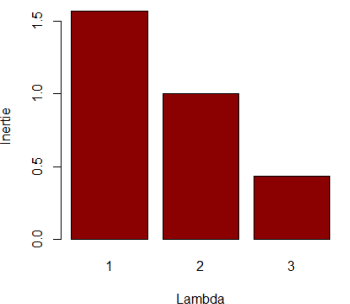
(On remarque qu'une sphère projetée est bien un disque, ce qui est rassurant).

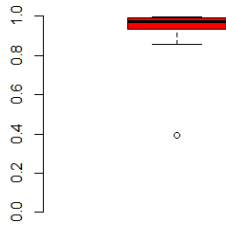
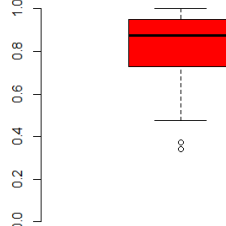
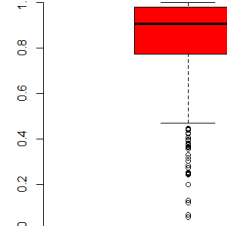
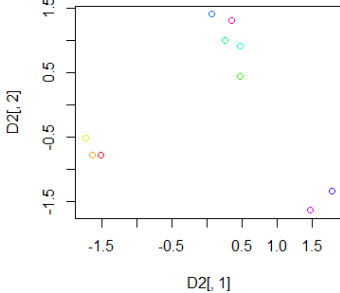
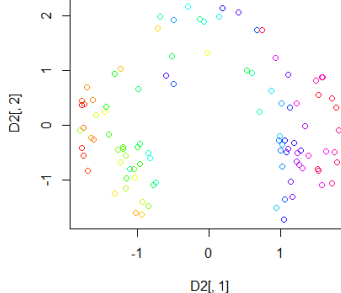
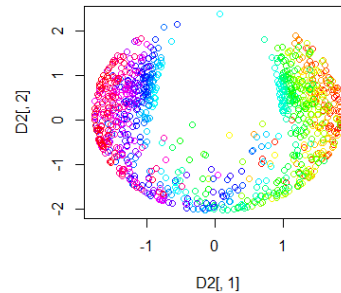
2) Après avoir écrit une fonction qui génère les données comme pour la question précédente, on va une fois de plus comparer les résultats pour $n = 10$, $n = 100$ et $n = 1000$ sans normer les données. Pour cela, on présente les mêmes figures que pour la question précédente :

| | $n = 10$ | $n = 100$ | $n = 1000$ |
|---|--|---|--|
| Données avant traitement |  |  |  |
| Valeurs propres de la matrice de covariance | <p>Valeurs propres de la matrice de covariance</p>  | <p>Valeurs propres de la matrice de covariance</p>  | <p>Valeurs propres de la matrice de covariance</p>  |

| | | | |
|--------------------------|--|--|--|
| Qualité de la projection | <p>Qualité de la projection</p>  | <p>Qualité de la projection</p>  | <p>Qualité de la projection</p>  |
| Données après traitement |  |  |  |

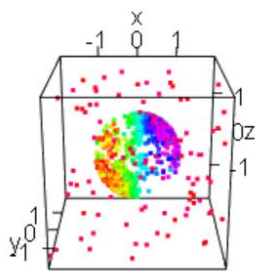
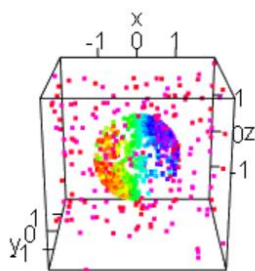
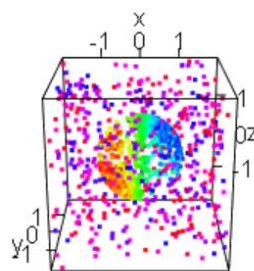
On réalise maintenant le même tableau, mais en normant les données avant traitement :

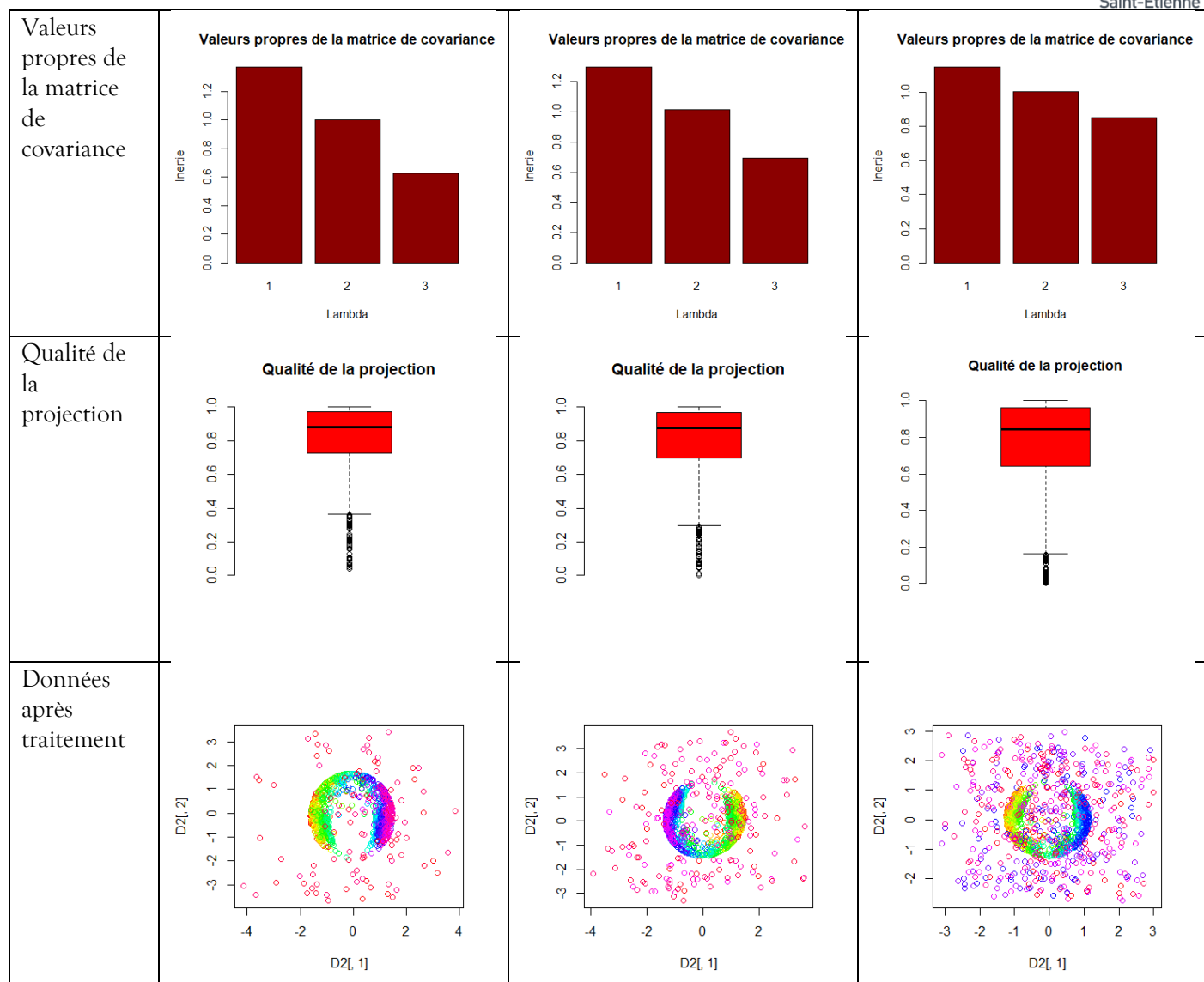
| | | | |
|---|---|--|---|
| | $n = 10$ | $n = 100$ | $n = 1000$ |
| Données avant traitement |  |  |  |
| Valeurs propres de la matrice de covariance | <p>Valeurs propres de la matrice de covariance</p>  | <p>Valeurs propres de la matrice de covariance</p>  | <p>Valeurs propres de la matrice de covariance</p>  |

| | | | |
|--------------------------|---|--|---|
| Qualité de la projection |  |  |  |
| Données après traitement |  |  |  |

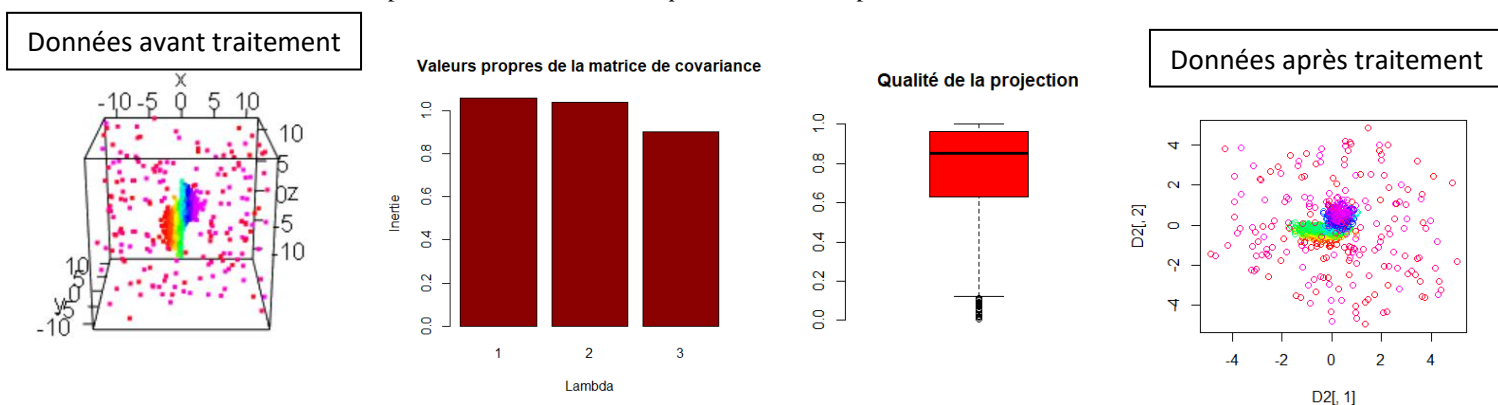
On remarque dans un premier temps que lorsque n augmente, la qualité de la projection baisse. Ensuite, la principale différence entre le fait de normer les données ou non semble résider dans la forme de la courbe finale : lorsqu'on norme les données, elles semblent suivre une forme plus précise que lorsqu'on ne les norme pas.

3) On va commencer par ajouter des points extrémaux sur les données de la question précédente (on reste avec $n = 1000$). On choisit donc d'ajouter 100, 200 et 500 points, qui prennent des valeurs extrémales. On représente comme précédemment les résultats de l'ACP :

| | | | |
|--------------------------|---|---|---|
| | <i>nextr</i> = 100 | <i>nextr</i> = 200 | <i>nextr</i> = 500 |
| Données avant traitement |  |  |  |



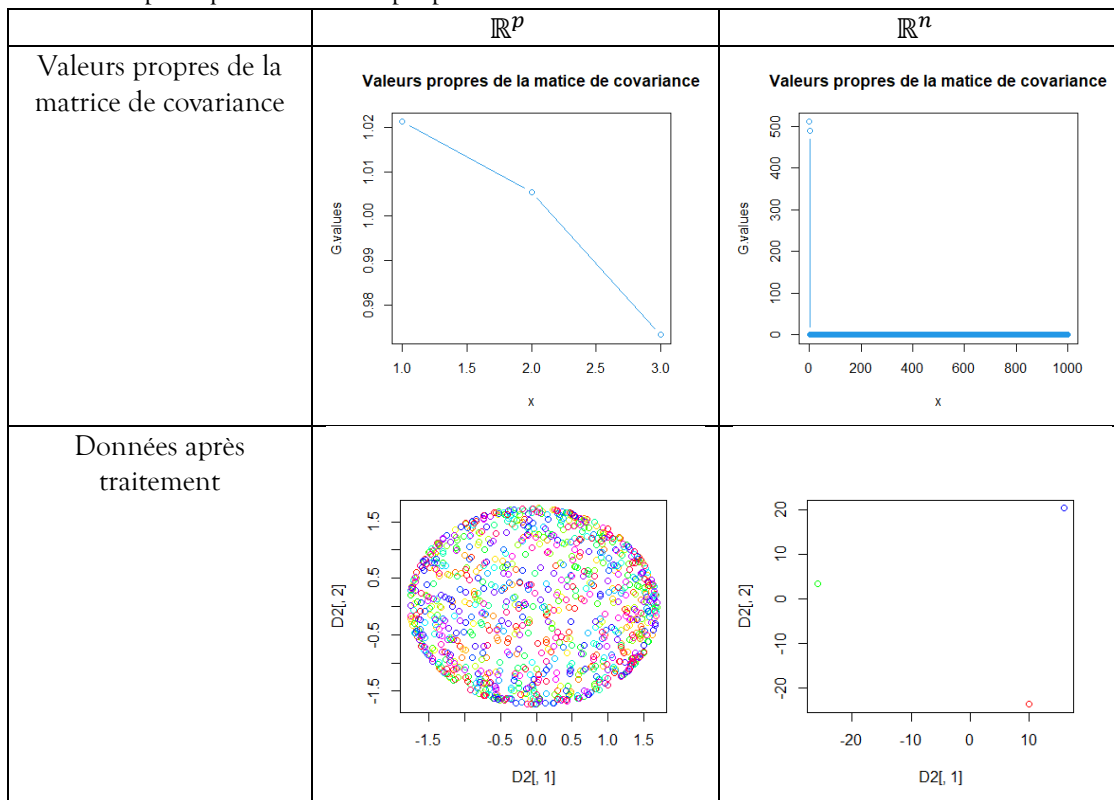
Voici les résultats de l'ACP pour $nextr = 200$ lorsqu'on ne norme pas les données :



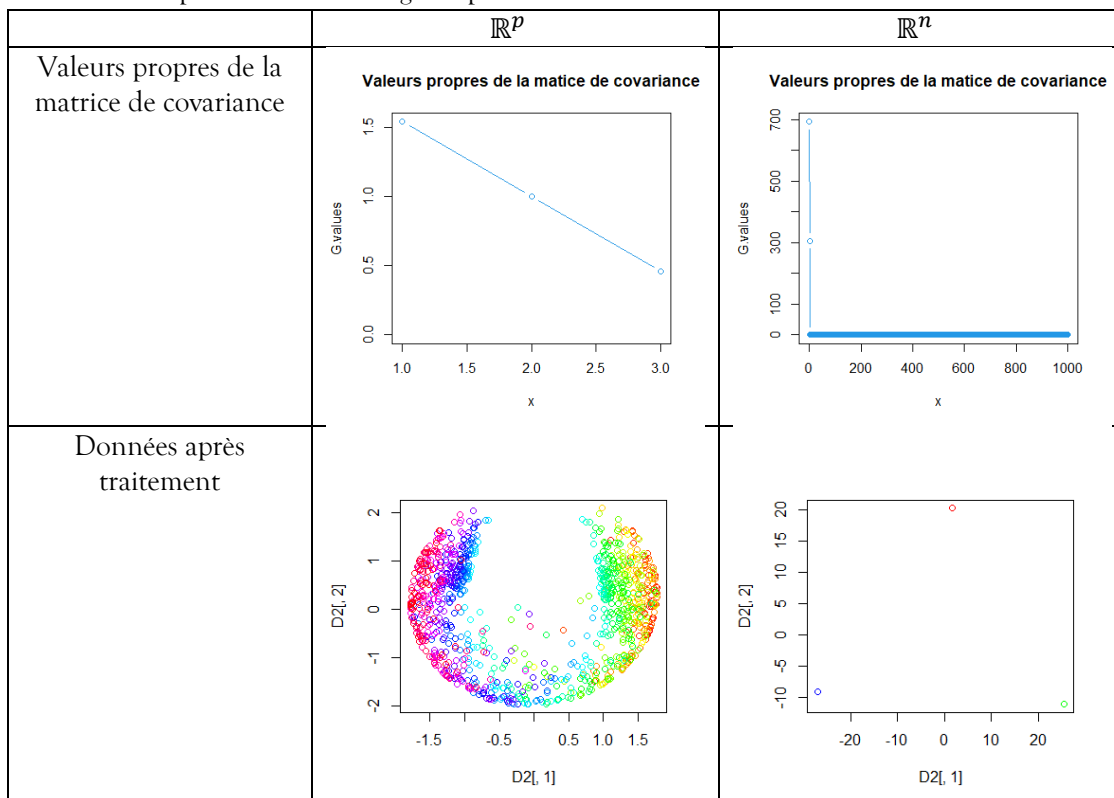
On observe tout d'abord que plus on ajoute de points extrémaux, plus les valeurs propres tendent à prendre les mêmes valeurs. Ensuite, on remarque que le fait de normer les données améliore grandement la qualité du résultat, que ça soit en terme d'écart entre les valeurs propres ou en terme de qualité de projection. De plus, les forme des données normées est bien plus précise que les données non normées.

Partie II : Etude de la forme du nuage initiale sur la réduction de dimension dans les deux espaces

1) On commence par comparer les résultats de l'ACP normé dans \mathbb{R}^p et \mathbb{R}^n pour le premier nuage de point (avec $n = 1000$). On ne compare que les valeurs propres de la matrice de covariance et le résultat final.



On réalise le même travail pour le second nuage de points :



2) Dans les deux cas, on remarque que seuls deux valeurs propres de la matrice de covariance sont supérieurs à 0, ce qui indique que la projection sera très performante.

3) On va maintenant vérifier les relations de passage suivantes :

$$v_\alpha = \frac{1}{\sqrt{\lambda_\alpha}} X u_\alpha \text{ et } u_\alpha = \frac{1}{\sqrt{\lambda_\alpha}} X' v_\alpha$$

avec v_α un vecteur propre normé de XX' , et u_α un vecteur propre normé de $X'X$, tout deux relatifs à la valeur propre non nulle λ_α .

Pour cela, on génère des données (on choisit celles qui correspondent au second nuage de points pour $n = 1000$). On vérifie tout d'abord que les valeurs propres non nulles de XX' et $X'X$ sont bien les mêmes ; pour cela on utilise le code suivant :

```
314 x <- données.gen2(1000)
315 lbda1 <- eigen(X%*%t(X))$values
316 lbda1 <- lbda1[which(lbda1 > 10**(-5))]
317 lbda2 <- eigen(t(X)%*%X)$values
318 print(lbda1-lbda2)
```

On obtient les résultats suivants, qui confirment le résultat théorique.

```
[1] -2.501110e-12  2.842171e-13  6.536993e-13
```

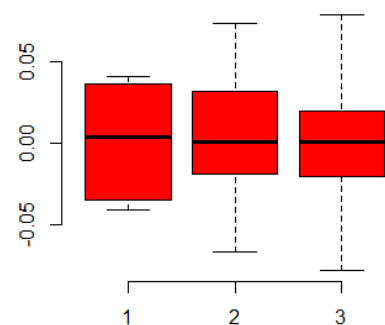
On va maintenant vérifier les relations de passage grâce au code suivant :

```
320 u <- eigen(t(X)%*%X)$vector
321 v <- eigen(X%*%t(X))$vector
322 v <- v[,which(lbda1 > 10**(-5))]
323
324 result <- matrix(0, nrow = dim(v)[1], ncol = dim(v)[2])
325 for (i in 1:length(lbda1))
326 {
327   result[,i] <- v[,i] - X%*%u[,i]/(lbda1[i])
328 }
329 boxplot(result,
330         main = "Vérification relations de passages" , col="red" , frame=F)
```

On obtient trois boîtes à moustaches qui (correspondant aux trois valeurs propres non nulles pour lesquels on a utilisé les relations de passage) représentent les valeurs des coefficients des trois vecteurs : $v_i - \frac{1}{\sqrt{\lambda_i}} X u_i$ pour $i \in \{1,2,3\}$

On observe que les coefficients des trois vecteurs obtenus sont relativement faibles, ce qui nous amène à considérer les relations de passages comme vérifiées (pour vérifier la seconde relation il suffit d'adapter le code).

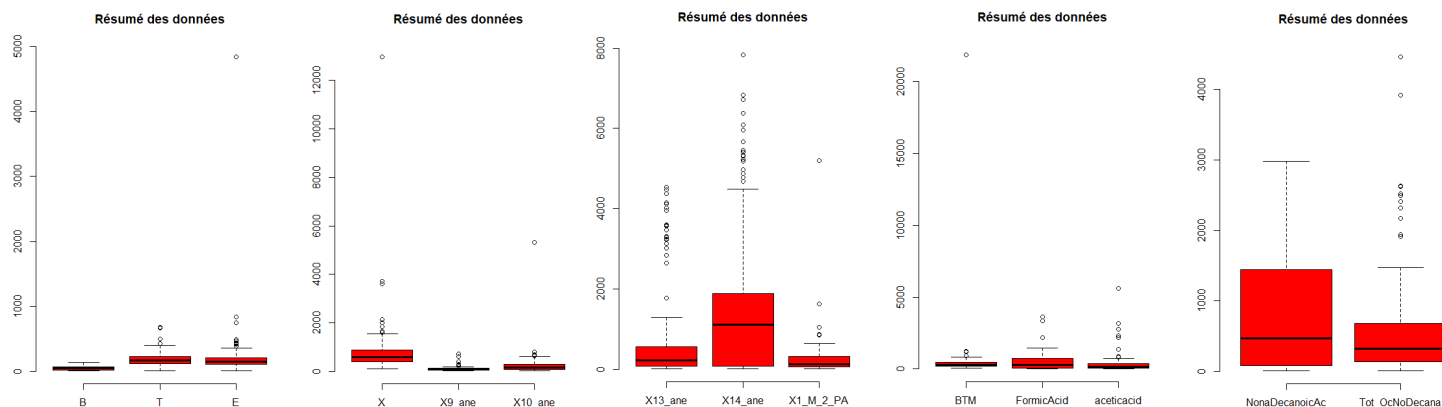
Vérification relations de passages



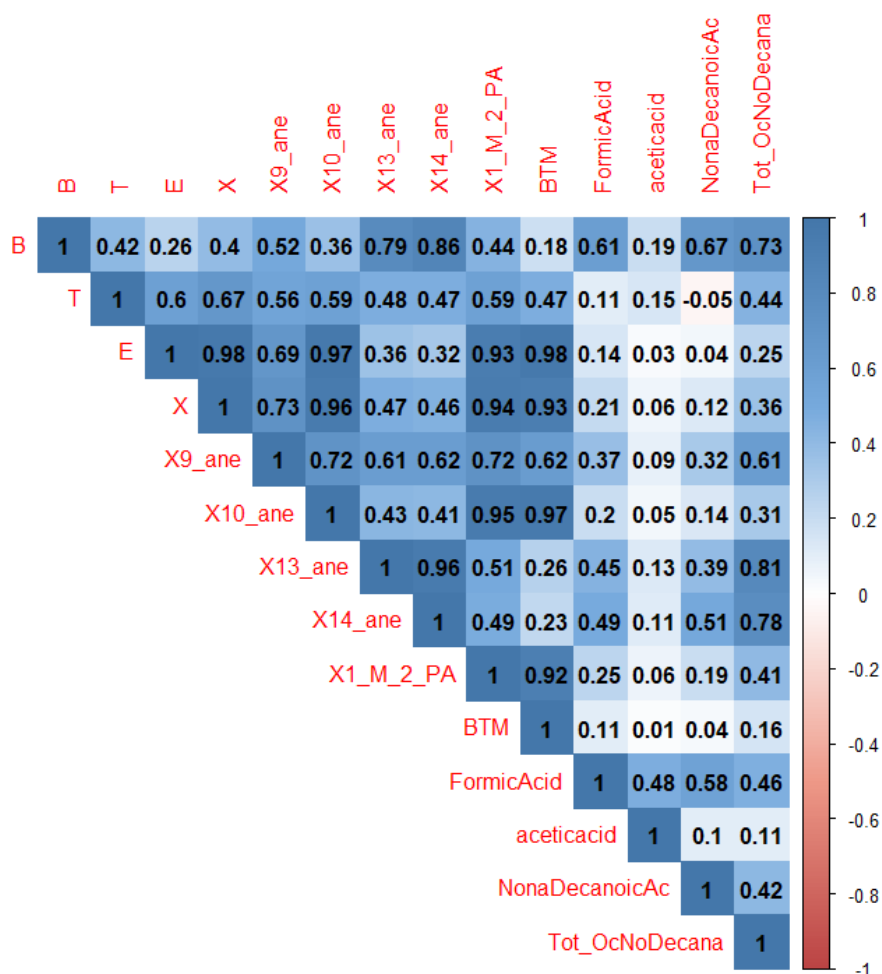
On remarque qu'en ouvrant les données, les accents sur le mot « été » ne sont pas bien gérés par R ; on modifie donc tous les « été » en « ete » directement sur le fichier source des données.

1) Traitement des données

Pour commencer à se familiariser avec les données mis à disposition, on va tout d'abord observer nos différentes données à l'aide des boîtes à moustaches (on étudie d'abord l'échantillon dans sa globalité) :



On va maintenant essayer de d'identifier des corrélations entre les différents facteurs. Pour cela, on représente la matrice de corrélation :



On observe que certaines des variables sont très fortement corrélées. Rigoureusement, il faudrait reproduire ces résultats en séparant nos données en fonctions des campagnes, de la saison, du moment d'ouverture des campagnes mais cela rendrait ce compte rendu encore plus lourd qu'il ne l'est. On va plutôt, pour chaque cas énoncé, comptabiliser dans une matrice le nombre de couples de facteurs pour lesquels le coefficient de corrélation est soit supérieur à 0.9, soit inférieur à 0.1 (en valeur absolue).

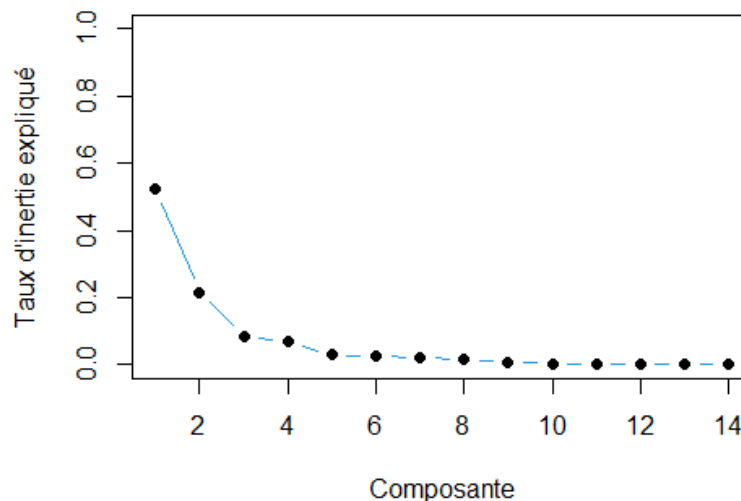
| Population prise compte | Nombre de couples fortement corrélés ($\text{abs}(\text{coef corrélation}) > 0.9$) | Nombre de couples peu corrélés ($\text{abs}(\text{coef corrélation}) < 0.1$) |
|-------------------------|--|--|
| Population totale | 11 | 9 |
| BF2 | 19 | 2 |
| BF3 | 6 | 5 |
| CA1 | 8 | 33 |
| CA2 | 10 | 33 |
| CA3 | 3 | 14 |
| CA4 | 0 | 30 |
| Été | 10 | 22 |
| Hiver | 4 | 7 |
| BF | 6 | 12 |
| CA | 7 | 23 |

On a maintenant plus d'idées sur les regroupement de données qui engendrent des corrélations forte.

2) Réduction de dimension et ACP

Le premier enjeu de l'ACP est de déterminer quelles et combien de composantes on pourrait considérer pour effectuer une projection. Pour répondre à ces question, on étudie les valeurs propres de la matrice de variance/covariance de nos données.

Taux d'inertie expliquée par chaque composante

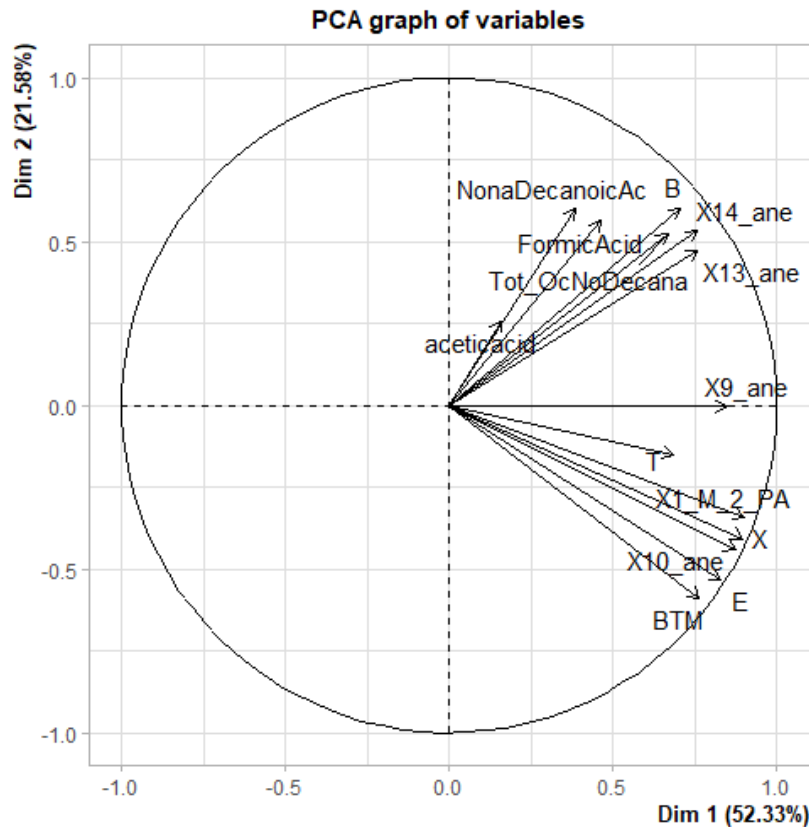


Ghcggh

vhjggkj

Numériquement, les deux premières composantes principales permettent d'expliquer 73.92 % de l'inertie, tandis que les quatre premières 89.27 %. Cependant, on observe une cassure assez marquée entre la seconde et la troisième composante. De plus, puisque les troisième et quatrième composantes ont des inerties très proches, il ne paraît pas justifié de n'en prendre qu'une des deux. Pour des simplifications de représentation, on opte pour les deux premières.

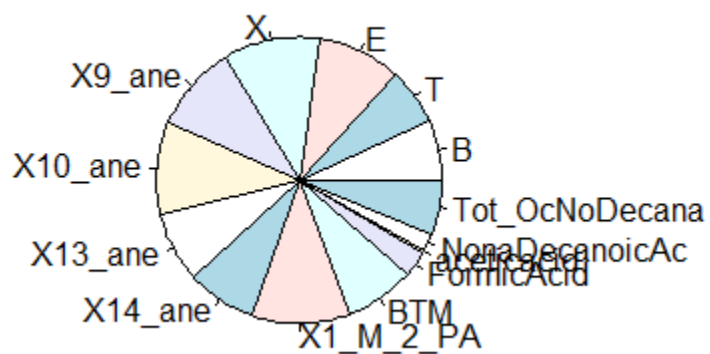
Pour déterminer quelles variables sont le mieux expliquées dans notre nouveau sous espace, on représente le cercle de corrélation des variables.



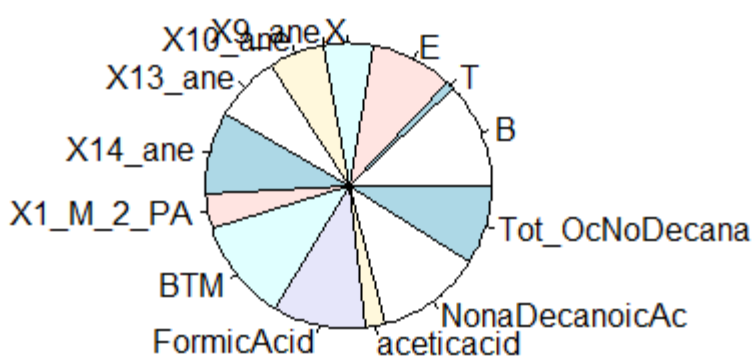
Sur ce cercle, chaque flèche correspond à une variable. Si deux flèches sont voisines, cela signifie que les variables correspondantes sont fortement corrélées positivement, et si deux flèches sont opposées, cela signifie qu'elles sont fortement corrélées négativement. Pour cette ACP on a utilisé la totalité de nos données, et d'après l'analyse des corrélations entre variables donnée plus tôt, on confirme qu'aucune variable n'est corrélée négativement avec une autre (pas de flèches opposées). De plus, plus une flèche est longue et proche du cercle, plus cela signifie que le facteur associé est bien représenté dans le nouveau sous espace. Ainsi, on conclut que ce sont les facteurs *aceticacid* et *T* qui sont le moins bien représentés.

Pour savoir quelles variables sont regroupées dans les différentes composantes, on s'intéresse aux contributions de chaque variable. Pour cela, on représente, pour chacun de nos deux composantes principales retenues, leur compositions :

Composante 1



Composante 2



Enfin, on représente les individus dans le nouveau sous espace :

