# TP 1 – Analyse en Composantes Principales

AUTEUR·RICE

B. Charlier, Marie Chavent

Environnement de travail : Pour réaliser ce TP, vous êtes libres de choisir l'un des trois outils suivants :

1. **RStudio** : avec un script `.R`, un document R Markdown (`.Rmd`) ou un fichier Quarto (`.qmd`).
2. **VS Code** : avec l'extension Quarto pour éditer et exécuter vos fichiers `.qmd`.
3. **Jupyter Notebook** : avec un noyau (*kernel*) R activé.

***Note : l'usage de Quarto ou R Markdown est vivement conseillé pour combiner code et rapport.***

On considère le tableau de données brutes suivant :

| | intensite.bulles | saveur.salee | appreciation.globale |
|---|---|---|---|
| **St Yorre** | 3.9 | 6.4 | 2.9 |
| **Vichy** | 1.4 | 6.0 | 2.8 |
| **Quezac** | 5.1 | 4.7 | 3.5 |
| **Salvetat** | 2.9 | 4.1 | 3.4 |
| **Perrier** | 8.2 | 4.9 | 2.8 |

Ces données sont disponibles ici.

## Exercice 1.

### I. Calcul de l'ACP du nuage des individus

Le but de cette première partie est de calculer « à la main », avec les fonctions de base de `R`, une ACP pour le nuage de points des individus.

On peut charger les données avec la commande

```
> eaux <- read.table("https://bcharlier.github.io/KMAK7AC1/tp/1_ACP/data/eaux.txt", header =
      TRUE, row.names = 1, sep = ";")
```

1. Calculez la matrice des corrélations $R$ à partir de la matrice des données centrées-réduites $Z$, en pondérant les lignes par $1/n$. *Indication : définissez un vecteur `moy` et un vecteur `ecartype` contenant respectivement la moyenne et l'écart-type biaisé des colonnes de la matrice de données $X$. Utilisez ensuite la fonction `scale()` pour normaliser les données. Voir aussi : `?mean`, `?sd` et `?apply`.*

```
> ###### Importation du tableau de données ######
>
> X <- eaux
> n <- nrow(X)
> p <- ncol(X)
>
> ###### Calcul de la matrice centrée-réduite avec la fonction `scale()` ######
>
> moy <- apply(X, 2, mean)
> ecartype <- apply(X, 2, sd) * sqrt((n-1)/n) # écart-type biaisé (diviseur n)
>
> ?scale
> Z <- scale(X, center = moy, scale = ecartype)
>
> apply(Z, 2, mean) # Vérification de la nullité des moyennes
```

```
   intensite.bulles          saveur.salee appreciation.globale
      -4.436555e-17          4.107663e-16         -5.772943e-16
```

```
> apply(Z, 2, sd) * sqrt((n-1)/n) # Vérification des écarts-types égaux à 1
```

```
   intensite.bulles          saveur.salee appreciation.globale
                  1                     1                     1
```

```
> ###### Calcul de la matrice des corrélations ######
>
> N <- diag(rep(1/n, n))
> R <- t(Z) %*% N %*% Z
```

```
> round(unname(Z[,]), 3)
```

```
       [,1]   [,2]   [,3]
[1,] -0.174  1.385 -0.588
[2,] -1.263  0.916 -0.915
[3,]  0.348 -0.610  1.373
[4,] -0.610 -1.315  1.046
[5,]  1.698 -0.376 -0.915
```

```
> N
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.2  0.0  0.0  0.0  0.0
[2,]  0.0  0.2  0.0  0.0  0.0
[3,]  0.0  0.0  0.2  0.0  0.0
[4,]  0.0  0.0  0.0  0.2  0.0
[5,]  0.0  0.0  0.0  0.0  0.2
```

```
> round(unname(R), 3)
```

```
       [,1]   [,2]   [,3]
[1,]  1.000 -0.289 -0.091
[2,] -0.289  1.000 -0.705
[3,] -0.091 -0.705  1.000
```

2. Effectuez la décomposition spectrale de $R$ et déduisez-en la matrice $V$ des vecteurs propres de $R$ triés par ordre décroissant des valeurs propres. Vérifiez que les vecteurs propres sont bien normés à 1 et orthogonaux (i.e., orthonormés). *Indication : utiliser la fonction `eigen()` de `R`. Voir donc `?eigen`.*

```
> ###### Décomposition spectrale avec la fonction `eigen()` ######
>
> e <- eigen(R)
> e$values
```

```
[1] 1.7332693 1.0635340 0.2031967
```

```
> e$vectors
```

```
           [,1]        [,2]       [,3]
[1,]  0.2008816 -0.92432968 -0.3244398
[2,] -0.7187537  0.08596119 -0.6899303
[3,]  0.6656123  0.37178664 -0.6470973
```

```
> t(e$vectors) %*% e$vectors # Les vecteurs propres sont bien orthonormés
```

```
              [,1]         [,2]          [,3]
[1,]  1.000000e+00 1.086934e-16 -5.219227e-16
[2,]  1.086934e-16 1.000000e+00  1.339940e-16
[3,] -5.219227e-16 1.339940e-16  1.000000e+00
```

```
> sum(e$values)
```

```
[1] 3
```

3. On peut maintenant projeter les individus sur les axes factoriels.

   a. Calculez la matrice $\psi$ des coordonnées des 5 eaux sur les trois axes factoriels.

```
> ###### Matrice des coordonnées factorielles des individus (scores) ######
> Psi<-Z %*% e$vectors
```

   b. Calculez les moyennes et les variances des trois colonnes de $\psi$. Comparez-les aux valeurs propres de $\boldsymbol{R}$.

```
> ###### Matrice des coordonnées factorielles des individus (scores) ######
> Psi<-Z %*% e$vectors
>
> apply(Psi, 2, mean) # Vérification de la nullité des moyennes
```

```
[1] -6.182815e-16 -1.179612e-16  8.882868e-17
```

```
> apply(Psi, 2, var) * (n - 1) / n
```

```
[1] 1.7332693 1.0635340 0.2031967
```

```
> e$values
```

```
[1] 1.7332693 1.0635340 0.2031967
```

```
> round(unname(Psi), 3)
```

```
      [,1]   [,2]   [,3]
[1,] -1.422  0.061 -0.519
[2,] -1.521  0.905  0.370
[3,]  1.422  0.136 -0.580
[4,]  1.519  0.839  0.428
[5,]  0.002 -1.942  0.301
```

4. Représentez graphiquement les 5 eaux sur le premier plan factoriel (1-2) des individus et sur le plan (1-3).

```
> ###### Matrice des coordonnées factorielles des individus (scores) ######
> Psi <- Z %*% e$vectors
> apply(Psi, 2, mean) # Vérification de la nullité des moyennes
```

```
[1] -6.182815e-16 -1.179612e-16  8.882868e-17
```

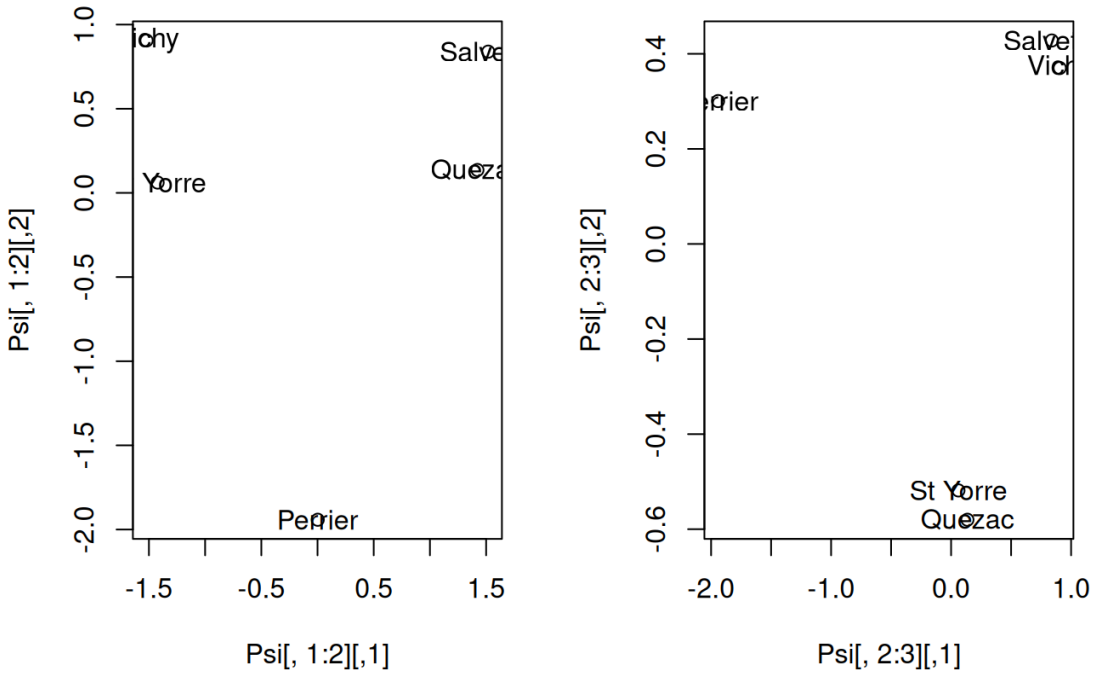
```
> apply(Psi, 2, var) * (n-1) / n
```

```
[1] 1.7332693 1.0635340 0.2031967
```

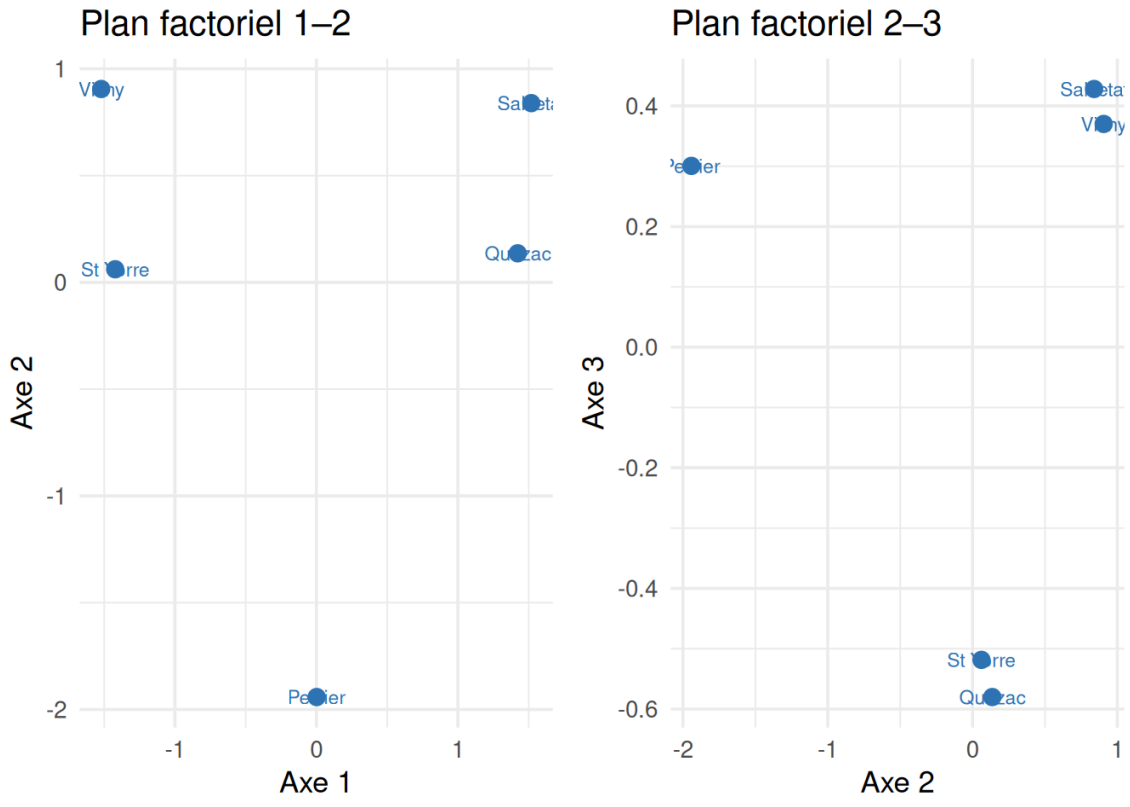
```
> e$values
```

```
[1] 1.7332693 1.0635340 0.2031967
```

```
> ###### Représentation graphique des individus sur les plans factoriels ######
> par(mfrow=c(1,2))
> plot(Psi[,1:2])
> text(Psi[,1:2],rownames(X))
>
> plot(Psi[,2:3])
> text(Psi[,2:3],rownames(X))
```



```
> library(ggplot2)
>
> scores_12 <- data.frame(
+   Eau = rownames(X),
+   x = Psi[, 1],
+   y = Psi[, 2]
+ )
>
> plot_12 <- qplot(x, y, data = scores_12, label = Eau, geom = c("point", "text"),
+   color = I("#2e74b5"), size = I(3)) +
+   labs(title = "Plan factoriel 1–2", x = "Axe 1", y = "Axe 2") +
+   theme_minimal(base_size = 13)
>
> scores_23 <- data.frame(
+   Eau = rownames(X),
+   x = Psi[, 2],
+   y = Psi[, 3]
+ )
>
> plot_23 <- qplot(x, y, data = scores_23, label = Eau, geom = c("point", "text"),
+   color = I("#2e74b5"), size = I(3)) +
+   labs(title = "Plan factoriel 2–3", x = "Axe 2", y = "Axe 3") +
+   theme_minimal(base_size = 13)
>
> gridExtra::grid.arrange(plot_12, plot_23, ncol = 2)
```




5. On sait que l'inertie satisfait la relation $I(\psi) = p = I(Z)$.
   a. Quelle est l'inertie du nuage des 5 eaux calculée à partir des coordonnées factorielles $\psi$ ?

```
> # Inertie totale du nuage des 5 eaux minérales
> sum(e$values)
```

```
[1] 3
```

b. Quel est le pourcentage d'inertie expliquée par chaque composante principale ? Quel est le pourcentage cumulé ? Combien de composantes retiendriez-vous ?

```
> e$values / p * 100 # Pourcentages d'inertie expliquée par chaque composante
```

```
[1] 57.775643 35.451135  6.773222
```

## II. On va maintenant calculer l'ACP avec `R` et le package `FactoMineR`.

1. Les deux fonctions de base sous `R` qui réalisent des ACP sont `prcomp()` et `princomp()`.

a. En vous aidant de l'aide, retrouvez les résultats du I, en particulier la matrice $\Psi$ et les valeurs propres.

```
> ###### ACP avec les fonctions `prcomp()` et `princomp()` ######
> ?prcomp
> res<-prcomp(X,center=TRUE,scale.=TRUE)
> res$sdev^2 # Carrés des racines des valeurs propres
```

```
[1] 1.7332693 1.0635340 0.2031967
```

```
> res$x
```

```
                  PC1          PC2         PC3
St Yorre -1.272107726  0.05483305 -0.4637780
Vichy    -1.360376543  0.80986396  0.3310178
Quezac    1.272316877  0.12162931 -0.5189193
Salvetat  1.358455596  0.75062871  0.3828634
Perrier   0.001711796 -1.73695502  0.2688161
```

```
> ?princomp
> res<-princomp(X, cor = TRUE)
> res$sdev^2 # Carrés des racines des valeurs propres
```

```
   Comp.1    Comp.2    Comp.3
1.7332693 1.0635340 0.2031967
```

```
> res$scores
```

```
               Comp.1      Comp.2     Comp.3
St Yorre -1.422259675 -0.06130521  0.5185196
Vichy    -1.520947213 -0.90545543 -0.3700892
Quezac    1.422493513 -0.13598570  0.5801695
Salvetat  1.518799529 -0.83922841 -0.4280543
Perrier   0.001913846  1.94197475 -0.3005456
```

b. Laquelle des deux fonctions utilise $n$ et $n-1$ au dénominateur pour le calcul des variances ? Expliquez les sorties suivantes.

```
> # Comparaison des valeurs propres selon la normalisation utilisée
> valeurs_ecart_type_n <- prcomp(
+   X,
+   center = TRUE,
+   scale. = apply(X, 2, sd) * sqrt((n - 1) / n)
+ )$sdev^2
>
> valeurs_ecart_type_n_1 <- prcomp(
+   X,
+   center = TRUE,
+   scale. = apply(X, 2, sd)
+ )$sdev^2
```

Valeurs propres selon la convention d'écart-type

| Écart-type avec diviseur n | Écart-type avec diviseur n-1 | Rapport des valeurs propres |
|---|---|---|
| **2.167** | 1.733 | 1.25 |
| **1.329** | 1.064 | 1.25 |
| **0.254** | 0.203 | 1.25 |

```
> # Comparaison des variances-covariances sans réduction des variables
> valeurs_princomp <- princomp(X, cor = FALSE)$sdev^2
> valeurs_prcomp <- prcomp(X, center = TRUE, scale. = FALSE)$sdev^2
```

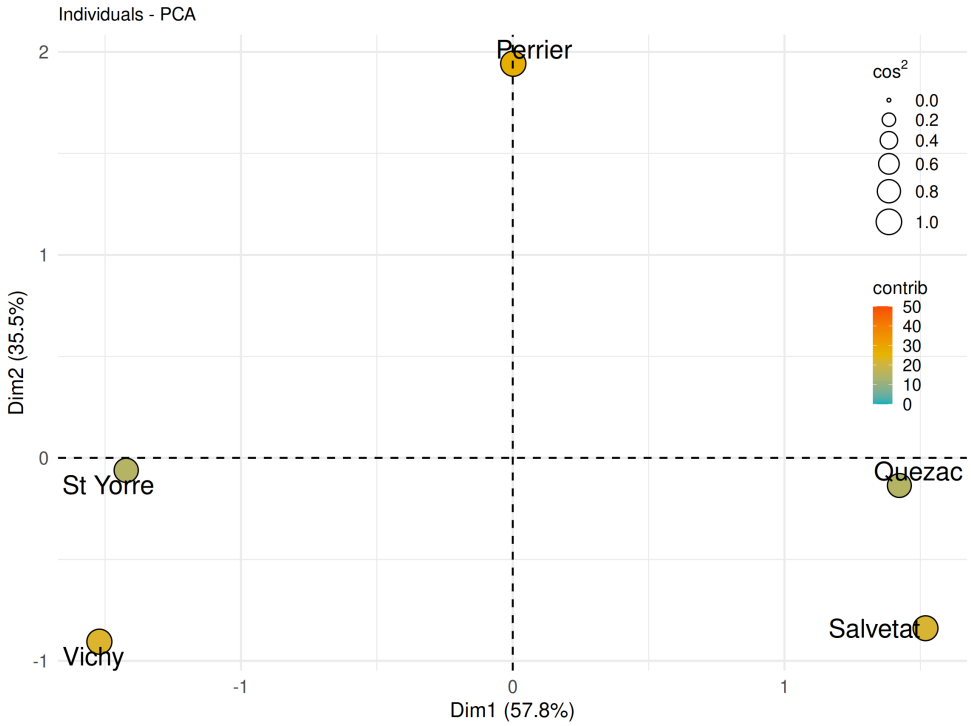
Valeurs propres des matrices de covariance

| | princomp sans réduction | prcomp sans réduction | Rapport covariance |
|---|---|---|---|
| **Comp.1** | 5.346 | 6.682 | 1.25 |
| **Comp.2** | 0.714 | 0.893 | 1.25 |
| **Comp.3** | 0.035 | 0.044 | 1.25 |

2. Installez le package `FactoMineR` et utilisez la fonction `PCA()` pour retrouver les résultats de la partie I.
   a. ***Indication : exécutez*** `summary(eaux.pca)`.
   b. Combien d'axes factoriels peut-on conserver ? Quelle est la qualité de représentation ? Quelles sont les contributions aux axes ? Quelle est l'interprétation du plan factoriel (1-2) des individus ?

```
> library(FactoMineR)
> eaux.pca <-PCA(eaux)      # Tous les calculs de l'ACP sont stockés dans l'objet
      "res"
```

3. Pour faire des graphiques plus avancés, on installera le package `factoextra`. Reproduisez la figure suivante :



## Exercice 2.

### I. Analyse du nuage des points-variables à la main

L'idée est toujours de retrouver les quantités définies dans le cours.

1. Calculez la matrice des produits scalaires des 5 eaux $NZZ^\top$ avec $N = \mathrm{Diag}(\frac{1}{n}, \ldots, \frac{1}{n})$.

```
> PS <- N %*% Z %*% t(Z)
```

2. Effectuez la décomposition spectrale de cette matrice et déduisez-en la matrice $U$ de ses vecteurs propres, triés par ordre décroissant des valeurs propres et $N$-normés. Comparez ces valeurs propres à celles de la matrice des corrélations $R$, puis vérifiez que $U$ est bien $N$-orthonormée.

```
> ###### Décomposition spectrale avec la fonction `eigen()` ######
> e <- eigen(PS)
> e$values
```

```
[1]  1.733269e+00  1.063534e+00  2.031967e-01  3.726754e-17 -7.256027e-17
```

```
> U <- sqrt(n) * e$vectors[,1:3] # Pour obtenir des vecteurs de norme N égale à 1
> t(U) %*% N %*% U
```

```
              [,1]          [,2]          [,3]
[1,]  1.000000e+00 1.148253e-16 -2.628806e-17
[2,]  8.252034e-17 1.000000e+00  2.447297e-16
[3,] -5.513964e-17 2.823855e-16  1.000000e+00
```

```
> t(e$vectors) %*% e$vectors # Les vecteurs propres sont bien orthonormés
```

```
              [,1]          [,2]          [,3]          [,4]          [,5]
[1,]  1.000000e+00 6.579292e-17 -7.149352e-17  2.531121e-18 -1.361449e-16
[2,]  6.579292e-17 1.000000e+00  2.409191e-16  1.128875e-16  2.367644e-17
[3,] -7.149352e-17 2.409191e-16  1.000000e+00 -2.182259e-16 -3.043859e-17
[4,]  2.531121e-18 1.128875e-16 -2.182259e-16  1.000000e+00  2.031204e-16
[5,] -1.361449e-16 2.367644e-17 -3.043859e-17  2.031204e-16  1.000000e+00
```

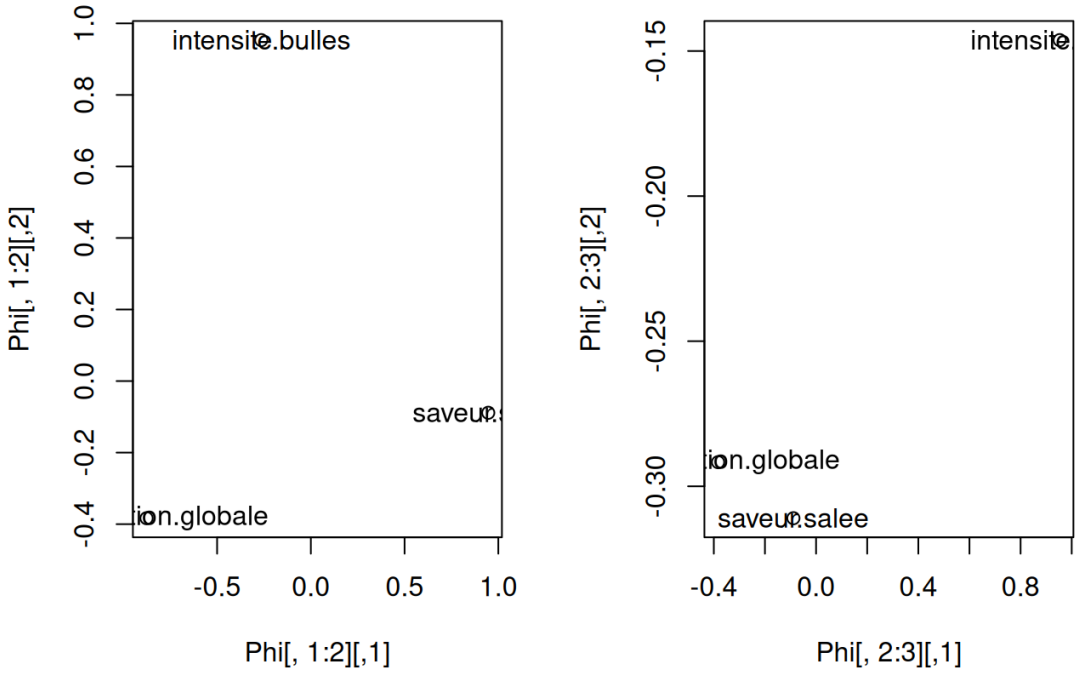
3. Calculez la matrice $\varphi$ des coordonnées factorielles des 3 variables sur les trois axes factoriels. Représentez graphiquement les 3 variables sur le premier plan factoriel (1-2).

```
> ###### Matrice des coordonnées factorielles des variables (charges factorielles)
      ######
> Phi<- t(Z) %*% N %*% U
```

```
> round(unname(Phi), 3)
```

```
       [,1]   [,2]   [,3]
[1,] -0.264  0.953 -0.146
[2,]  0.946 -0.089 -0.311
[3,] -0.876 -0.383 -0.292
```

```
> ###### Représentation graphique des variables sur les plans factoriels ######
> par(mfrow = c(1, 2))
> plot(Phi[, 1:2])
> text(Phi[, 1:2], colnames(X))
>
> plot(Phi[, 2:3])
> text(Phi[, 2:3], colnames(X))
```




4. Effectuez la décomposition en valeurs singulières (SVD) de $Z$ avec les métriques $I_p$ et $N$. ***Indication : vous pouvez utiliser la fonction*** `FactoMineR::svd.triplet()`.
   a. Vérifiez que les valeurs singulières sont bien les racines des valeurs propres déjà trouvées. Retrouvez, à partir de cette SVD, les matrices $U$ et $V$ (déjà trouvées aux questions précédentes).
   b. Vérifiez les formules de passage $\psi_{:\alpha} = \sqrt{\lambda_\alpha} u_\alpha$ et $\varphi_{:\alpha} = \sqrt{\lambda_\alpha} v_\alpha$.
   c. Calculez la corrélation entre l'intensité des bulles et la première composante principale $\psi_{:1}$. Vérifiez que vous retrouvez bien $\varphi_{11}$.

```
> ###### SVD classique ######
> svd_vanilla<-svd(N^0.5 %*% Z)
>
> svd_vanilla$d^2 # On retrouve bien les valeurs propres
```

```
[1] 1.7332693 1.0635340 0.2031967
```

```
> e$values
```

```
[1]  1.733269e+00  1.063534e+00  2.031967e-01  3.726754e-17 -7.256027e-17
```

```
> ###### SVD pondérée ######
>
> ?svd.triplet()
> row.w <- diag(N)
> col.w <- rep(1, p)
> svd_weighted <- svd.triplet(Z, row.w, col.w)
>
> svd_weighted$vs^2 # On retrouve bien les valeurs propres
```

```
[1] 1.7332693 1.0635340 0.2031967
```

```
> e$values
```

```
[1]  1.733269e+00  1.063534e+00  2.031967e-01  3.726754e-17 -7.256027e-17
```

```
> ###### SVD classique ######
>
> U <- sqrt(n) * svd_vanilla$u
> V <- svd_vanilla$v
>
> U %*% diag(svd_vanilla$d) # Première formule de passage vérifiée
```

```
             [,1]        [,2]       [,3]
[1,] -1.422259675  0.06130521 -0.5185196
[2,] -1.520947213  0.90545543  0.3700892
[3,]  1.422493513  0.13598570 -0.5801695
[4,]  1.518799529  0.83922841  0.4280543
[5,]  0.001913846 -1.94197475  0.3005456
```

```
> unname(Psi)
```

```
             [,1]        [,2]       [,3]
[1,] -1.422259675  0.06130521 -0.5185196
[2,] -1.520947213  0.90545543  0.3700892
[3,]  1.422493513  0.13598570 -0.5801695
[4,]  1.518799529  0.83922841  0.4280543
[5,]  0.001913846 -1.94197475  0.3005456
```

```
> V %*% diag(svd_vanilla$d) # Seconde formule de passage vérifiée
```

```
           [,1]        [,2]       [,3]
[1,]  0.2644680 -0.95324075 -0.1462489
[2,] -0.9462657  0.08864987 -0.3110022
[3,]  0.8763031  0.38341534 -0.2916943
```

```
> unname(Phi)
```

```
           [,1]        [,2]       [,3]
[1,] -0.2644680  0.95324075 -0.1462489
[2,]  0.9462657 -0.08864987 -0.3110022
[3,] -0.8763031 -0.38341534 -0.2916943
```

```
> ###### SVD pondérée ######
>
> U <- svd_weighted$U
> V <- svd_weighted$V
>
> U %*% diag(svd_weighted$vs) # Première formule de passage vérifiée
```

```
             [,1]        [,2]       [,3]
[1,] -1.422259675 -0.06130521  0.5185196
[2,] -1.520947213 -0.90545543 -0.3700892
[3,]  1.422493513 -0.13598570  0.5801695
[4,]  1.518799529 -0.83922841 -0.4280543
[5,]  0.001913846  1.94197475 -0.3005456
```

```
> unname(Psi)
```

```
             [,1]        [,2]       [,3]
[1,] -1.422259675  0.06130521 -0.5185196
[2,] -1.520947213  0.90545543  0.3700892
[3,]  1.422493513  0.13598570 -0.5801695
[4,]  1.518799529  0.83922841  0.4280543
[5,]  0.001913846 -1.94197475  0.3005456
```

```
> V %*% diag(svd_weighted$vs) # Seconde formule de passage vérifiée
```

```
           [,1]        [,2]      [,3]
[1,]  0.2644680  0.95324075 0.1462489
[2,] -0.9462657 -0.08864987 0.3110022
[3,]  0.8763031 -0.38341534 0.2916943
```

```
> unname(Phi)
```

```
           [,1]        [,2]       [,3]
[1,] -0.2644680  0.95324075 -0.1462489
[2,]  0.9462657 -0.08864987 -0.3110022
[3,] -0.8763031 -0.38341534 -0.2916943
```

```
> Phi[1, 1]
```

```
intensite.bulles
       -0.264468
```

```
> cor(X[, 1], Psi[, 1]) # Corrélation entre la variable et la composante
```

```
[1] 0.264468
```

## II. Avec `R` et le package `FactoMineR`

1. Retrouvez les coordonnées factorielles des variables (les charges factorielles), c'est-à-dire la matrice $\varphi$, avec les deux fonctions de base `prcomp()` et `princomp()`.
2. Retrouvez ensuite $\varphi$ avec la fonction `FactoMineR::PCA()`.

```
> eaux.pca$var$coord
```

```
                         Dim.1       Dim.2     Dim.3
intensite.bulles      0.2644680  0.95324075 0.1462489
saveur.salee         -0.9462657 -0.08864987 0.3110022
appreciation.globale  0.8763031 -0.38341534 0.2916943
```

```
> Phi
```

```
                          [,1]        [,2]       [,3]
intensite.bulles     -0.2644680  0.95324075 -0.1462489
saveur.salee          0.9462657 -0.08864987 -0.3110022
appreciation.globale -0.8763031 -0.38341534 -0.2916943
```

3. Retrouvez la représentation graphique des trois variables sur le premier plan factoriel et interprétez-la. Interprétez ensuite le premier plan factoriel des eaux à partir de ce graphique.

