Statistique Numérique et Analyse de Données
Ecole des Ponts ParisTech, 2ème année

TP2 : ANALYSE DESCRIPTIVE MULTIVARIÉE

Analyse des données de peintures de Rembrandt et de Van Gogh

On se propose, à titre purement pédagogique, d'analyser un jeu de données composé de 40 œuvres de Rembrandt et de 44 œuvres de Van Gogh. Un rapide aperçu de ces œuvres est donné à la page 4 de ce document.

Afin d'effectuer une analyse statistique, on représente chaque image par son histogramme de couleurs. Dans le cas présent, cela consiste à partitionner l'espace des couleurs (ici, $[0,255]^3$) en k parties égales et, pour chaque image I, calculer la proportion p_j^I des pixels se trouvant dans la partie j, pour $j = 1,\ldots,k$. Après avoir effectué cette étape, on associe à chaque image I le vecteur numérique de taille k contenant les proportions $(p_1^I,\ldots,p_k^I)$. On peut, sans perte d'information, supprimer la dernière coordonnée de ce vecteur car elle est toujours égale à (1− la somme de toutes les autres coordonnées).

Pour les 84 images correspondantes aux peintures de Rembrandt et de Van Gogh, les vecteurs des histogrammes des couleurs avec $k = 8$ et $k = 64$ ont été calculés et stockés respectivement dans les fichiers `painting8.dat` et `painting64.dat`. Ces fichiers sont disponibles sur la page `http://imagine.enpc.fr/~dalalyan/StatNum.html`.

1. Créez un répertoire nommé `TP_2_Stat`.
2. Dans la fenêtre de `R`, définissez le répertoire créé comme répertoire courant.
   `Fichier --> changer le répertoire courant ...`
3. Téléchargez les fichiers `painting8.dat` et `painting64.dat`. Sauvez-les dans votre répertoire courant.
4. Chargez le contenu du fichier `painting8.dat` à l'aide de la commande
   ```
   paintings8=data.frame(read.table("painting8.dat",sep=","));
   ```
5. Pour le compte-rendu, répondez aux questions suivantes :
   (a) Que fait la fonction `data.frame(.)` dans la commande ci-dessus ?
   (b) A quoi l'option `sep=","` sert-elle ?
6. **Dans le tableau téléchargé, les 40 premières lignes correspondent aux peintures de Rembrandt et les 44 dernières lignes correspondent à celles de Van Gogh.**
   Afin de se convaincre que les données ont bien été chargées, on peut afficher les boxplots des différentes variables à l'aide des commandes
   ```
   par(bg="cornsilk",lwd=2,col="darkblue",fg="darkblue");
   boxplot(paintings8);
   ```
   Laquelle de ces 7 variables est la plus dispersée ? la moins dispersée ?

7. Pour visualiser la matrice des nuages des points bivariés, on saisit :

```
pairs(paintings8);
```

On peut également modifier les paramètres graphiques :

```
pairs(paintings8,fg="darkblue",bg="orange",pch=21,cex=1.5);
```

8. Pour le compte-rendu, répondez aux questions suivantes :
   (a) A quoi servent les options `bg="orange"` et `cex=1.5` ?
   (b) Y a-t-il des paires de variables qui sont (approximativement) reliées par une fonction affine ?

9. On effectue maintenant une ACP sur les données `paintings8` :

```
Z=prcomp(paintings8,retx=T,scale=F);
z=Z$x;
par(mfcol=c(1,2),bg="cornsilk",lwd=2,col="darkblue",fg="darkblue")
boxplot(z)
plot(z[,1:2],col="darkblue",pch=21,cex=1.5,bg="orange");
```

La plus importante commande ci-dessus est `prcomp`. C'est la fonction de `R` qui permet d'effectuer une ACP (en anglais, PCA, **pr**incipal **comp**onent analysis). Son fonctionnement est expliqué ci-dessous :

La commande `prcomp`

La commande `prcomp` permet d'effectuer une ACP. La syntaxe générale est

```
Z=prcomp( jeu de données , options )
```

- Les arguments
  - L'argument `jeu de données` est du type `data.frame`. Il contient la matrice des données multivariées que l'on souhaite analyser par une ACP.
  - L'argument `options` permet de spécifier un certain nombre d'options dont les plus utilisées sont `retx`, `center` et `scale`.
    - `retx` : paramètre logique (`TRUE/FALSE`) indiquant si oui ou non les projections sur les composantes principales doivent être calculées.
    - `center` : paramètre logique (`TRUE/FALSE`) indiquant si oui ou non les variables doivent être centrées. La valeur par défaut est `TRUE`.
    - `scale` paramètre logique (`TRUE/FALSE`) indiquant si oui ou non les variables doivent être réduites. Lorsque les variables sont mesurées dans des unités hétérogènes, il est primordial d'utiliser l'ACP sur les variables réduites.
- La sortie de la commande `prcomp` est une liste contenant les éléments suivants :
  - `Z$x` : la matrice des données projetées sur les composantes principales. Par exemple, la première colonne `Z$x[,1]` de la matrice `Z$x` est la projection de tous les individus sur la première composante principale (CP).
  - `Z$sdev` : les écarts-types des composantes principales (CP).
  - `Z$rotation` : les matrices contenant les coordonnées des CP.
- L'objet `Z` peut être passé comme argument aux fonctions `plot` et `summary`.

10. Le graph affiché par la commande `boxplot(z)` confirme-t-il que les composantes principales sont classées par ordre de décroissance de variance des données projetées ?
11. On affiche maintenant les projections des individus sur les deux premiers axes principaux, en différenciant les peintures de Rembrandt de celles de Van Gogh :
```
plot(z[,1:2],type="n");
points(z[1:40,1:2],col="darkblue",pch=21,cex=1.5,bg="lightblue");
points(z[41:84,1:2],col="darkred",pch=24,cex=1.5,bg="orange");
```
Y a-t-il une bonne séparation entre les points représentants les œuvres de Rembrandt et ceux qui représentent les œuvres de Van Gogh ?
12. Effectuer une ACP sur les données `painting64` et tracer les projections des individus sur les deux premiers axes principaux. La séparation entre les peintures de Rembrandt et de Van Gogh est-elle meilleure que celle de la question précédente ? Donner une explication intuitive de ce résultat.
13. On a vu en cours que les trois représentations graphiques les plus utilisées dérivées d'une ACP sont la projection des individus, le scree-graph et la projection des variables sur le disque des corrélations. Afin d'obtenir les deux derniers graphes, on utilise les commandes `screeplot(.)` et `s.corcircle(.)`, ce dernier étant disponible dans le package `ade4`. Dans le cas des données `painting64`, on peut afficher ces trois graphes à l'aide des commandes suivantes :
```
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))
par(bg="cornsilk")
plot(z[,1:2],type="n")
points(z[1:40,1:2],col="darkblue",pch=21,cex=1.5,bg="lightblue")
points(z[41:84,1:2],col="darkred",pch=24,cex=1.5,bg="orange")
text(z[1:40,1:2]-c(0.01,0.01),as.character(1:40),font=2,col="darkblue")
text(z[41:84,1:2]-c(0.01,0.01),as.character(1:44),font=2,col="darkred")
screeplot(PCC,xlab="Scree graph",main="")
cc=cor(paintings64,z[,1:2])
library(ade4)
s.corcircle(cc, lab = names(paintings64))
```
Lorsqu'on lance ces commandes, on remarque que la variable `PCC` utilisée ci-dessus n'existe pas. En utilisant l'aide en ligne pour la commande `screeplot`, essayez de comprendre par quelle variable existante `PCC` doit être remplacée.
Cela devrait conduire vers le résultat affiché dans la Figure 2 ci-après.
Question : que fait la commande `cc=cor(paintings64,z[,1:2])` ?
14. On appelle la part de l'inertie totale expliquée par les k premières composantes principales le rapport de la somme des variances des k premières CP sur la somme des variances de toutes les CP.
    (a) Ecrire une fonction qui prend comme argument un `data.frame` et qui affiche à la sortie la part de l'inertie totale expliquée par les deux premières CP. (Indication : on pourra se servir de la valeur `Z$sdev` produite par la commande `prcomp`.)
    (b) Quelle est la part de l'inertie totale expliquée par les deux premières CP pour les données `painting64` ?

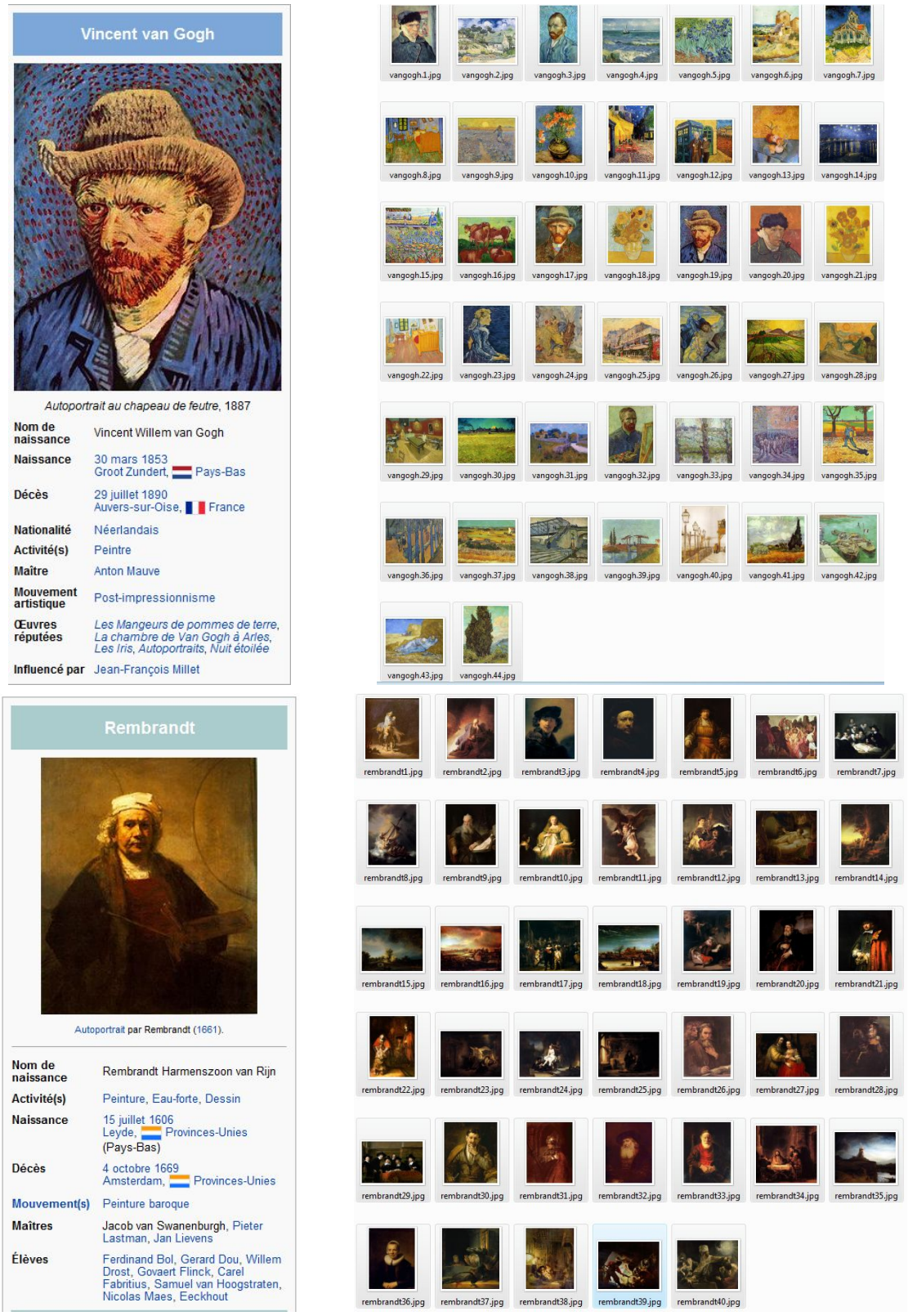

FIGURE 1 – Les œuvres de Van Gogh et de Rembrandt utilisées dans ce TP.

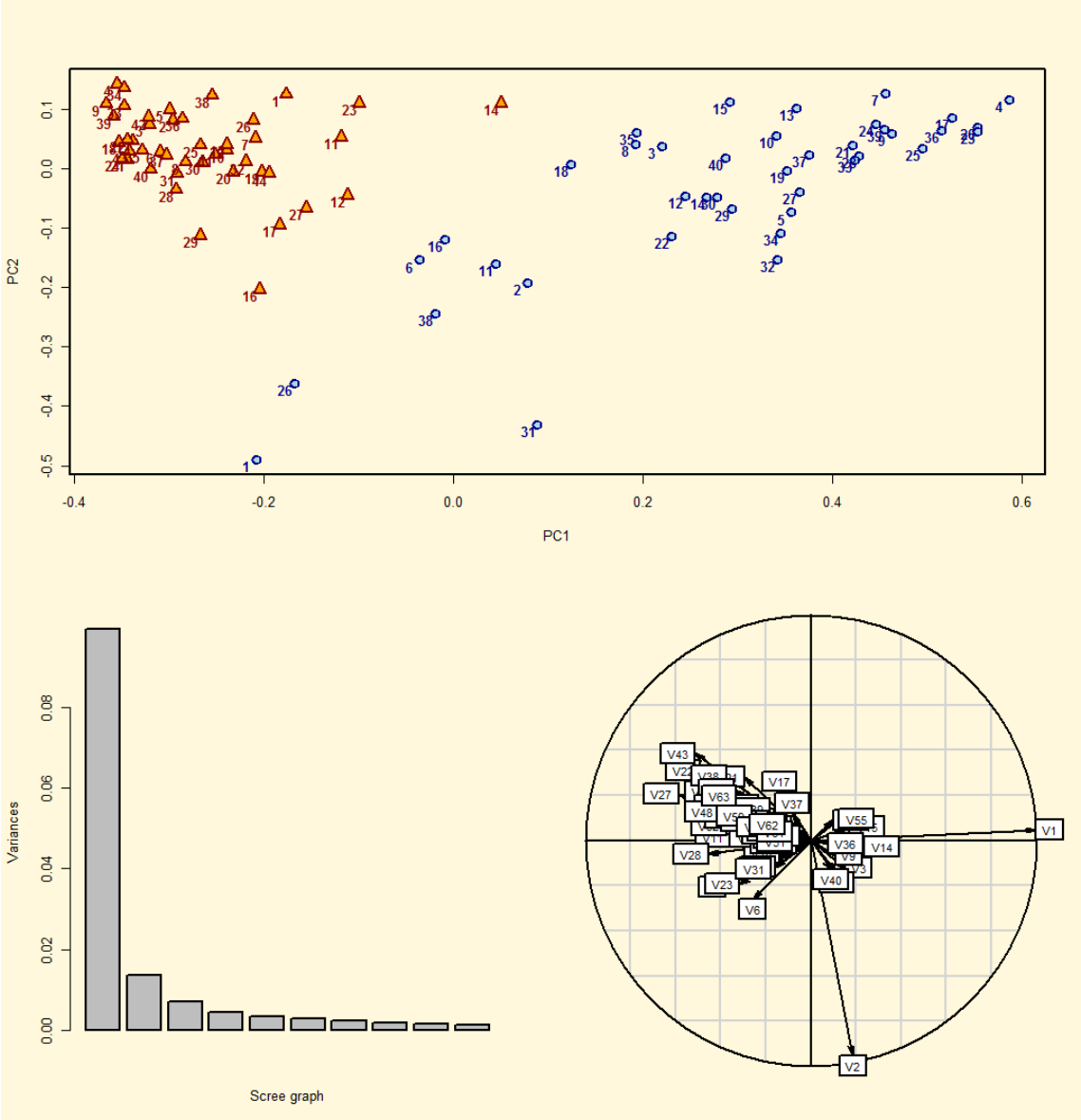

FIGURE 2 – Les 3 représentations graphiques les plus utilisées dérivées d'une ACP.