L'encadrant Mr Febrissy m'a confirmé que je pouvais rendre un travail personnel.

1. Partie 1

1.1. Blood transfusion

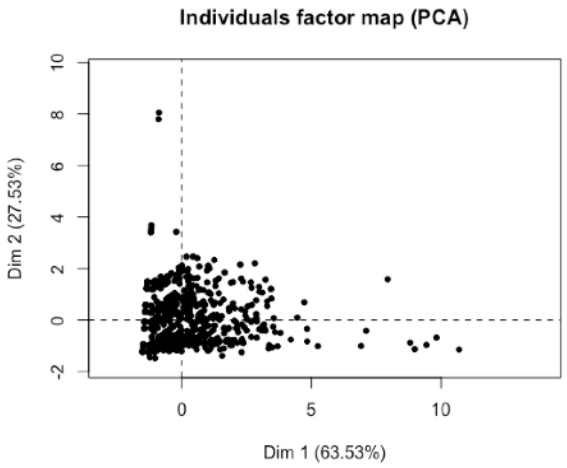
1.1.1. Présentation du jeu de données

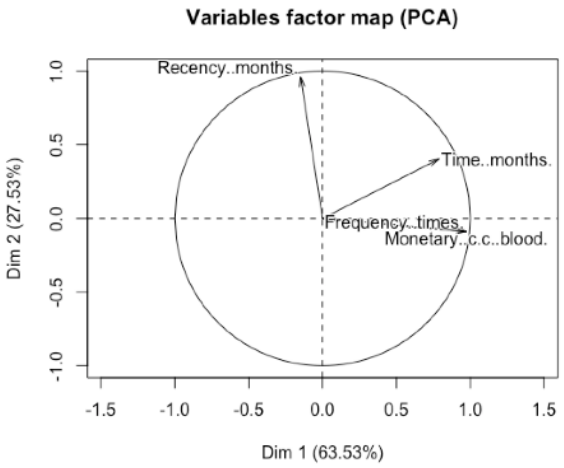
Le jeu de de données est tiré d'une base de donnée de *donneurs de sang* à Taiwan. Il est composé de **748 entrées** représentant des donneurs *extraits au hasard* du jeu de données initial. Les caractéristiques comprises sont tirées du model RFMT.

Les différents histogrammes nous montrent que certaines variables sont homogènes, mais quelques-unes ont une répartition plus uniforme. Dans le cas de variables non homogènes, il est intéressant de séparer le jeu de données en plusieurs classes.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Std
Recency	0	2.75	7	9.5	14	74	8.1
Frequency	1	2.00	4	5.5	7	50	5.8
Monetary	250	500.00	1000	1378.7	1750	12500	1459.8
Months	2	16.00	28	34.3	50	98	24.4

1.1.2. Réduction de dimensions





Cette réduction de dimensions via PCA a été réalisée en définissant les paramètres `scale.unit = TRUE`, afin de palier au différences d'échelles des variables, et `ncp = 2` afin de se concentrer sur une projection plane en minimisant la perte d'informations (étude du cercle des corrélations et de la positions des vecteurs). Le cercle des corrélations montre que *Recency* est très peu colérée de *Frequency* et *Monetary*. En revanche *Time* l'est plus, le cosinus est proche de 3/4. Toutes les variables sont bien représentées car très proches du cercle unité. Le plan des individus en montre certains bien plus représentés par la Dim1 que par la Dim2, et d'autre inversement. Ces individus restent cependant très minoritaires.

1.1.3. Classification

1.1.3.1. Hierarchical Clustering

			Classes	
	748		Positif	Négatif
			178	**570**
Prédictions	Positif	**740**	172	568
	Négatif	**8**	6	2

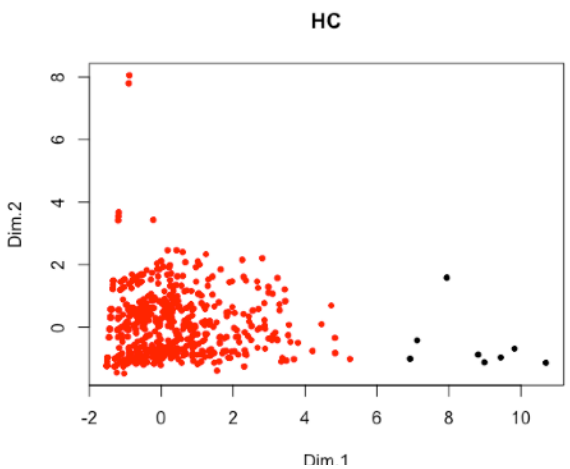
Cette méthode ne donne pas de bons résultats en regardant les faux positifs: elle est trop permissive et classe mal les objets.

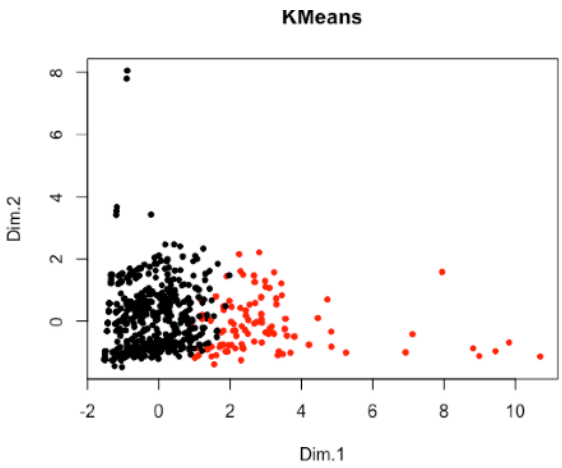
1.1.3.2. KMeans

			Classes	
	748		Positif	Négatif
			178	**570**
Prédictions	Positif	**631**	134	497
	Négatif	**117**	44	73

Cette méthode donne de meilleurs résultats, mais ils ne sont pas bons pour autant au regard de la table de confusion.

1.1.4. Projection des classes





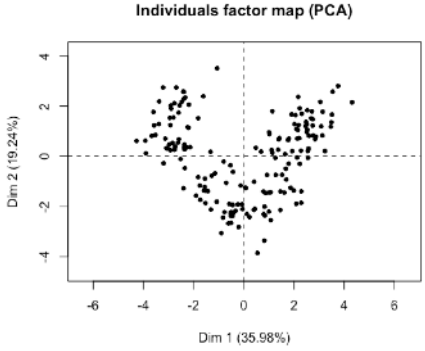
1.2. Wine recognition

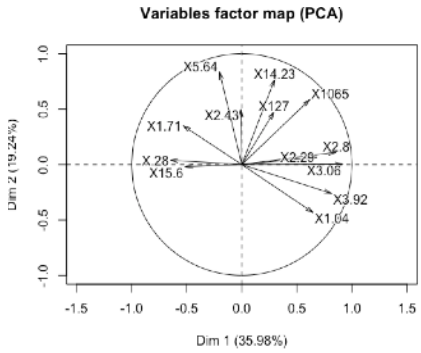
1.2.1. Présentation du jeu de données

Le jeu de de données est tiré d'une base de donnée de *vins italiens*. Il est composé de **177 entrées** représentant des vins issus de la même région par trois cultivateurs différents.

Les différents histogrammes nous montrent que les variables sont **homogènes**, car les valeurs sont proches de la moyenne. Dans le cas de variables non homogènes, il est intéressant de séparer le jeu de données en plusieurs classes. Nous remarquons aussi que la variable `Monetary` est très hétérogène: les valeurs maximum sont très éloignées de la moyenne comme le montre la `boxplot` associé.

1.2.2. Réduction de dimensions





Cette réduction de dimensions via PCA a été réalisée en définissant les paramètres `scale.unit = TRUE`, afin de palier au différences d'échelles des variables, et `ncp = 3` afin de minimiser la perte d'informations (étude du cercle des corrélations et de la positions des vecteurs). Le cercle des corrélation montre que les dimensions sont majoritairement bien représentées (assez proches du cercle unité). Certaines sont très corrélées (éventuellement négativement), et d'autres pas du tout. Il faut pour cela évaluer le cosinus entre les vecteurs. Le plan des individus fait apparaitre trois types de vins, assez bien corrélés avec les deux dimensions (leurs scores sont élevés en valeur absolue).

1.2.3. Classification

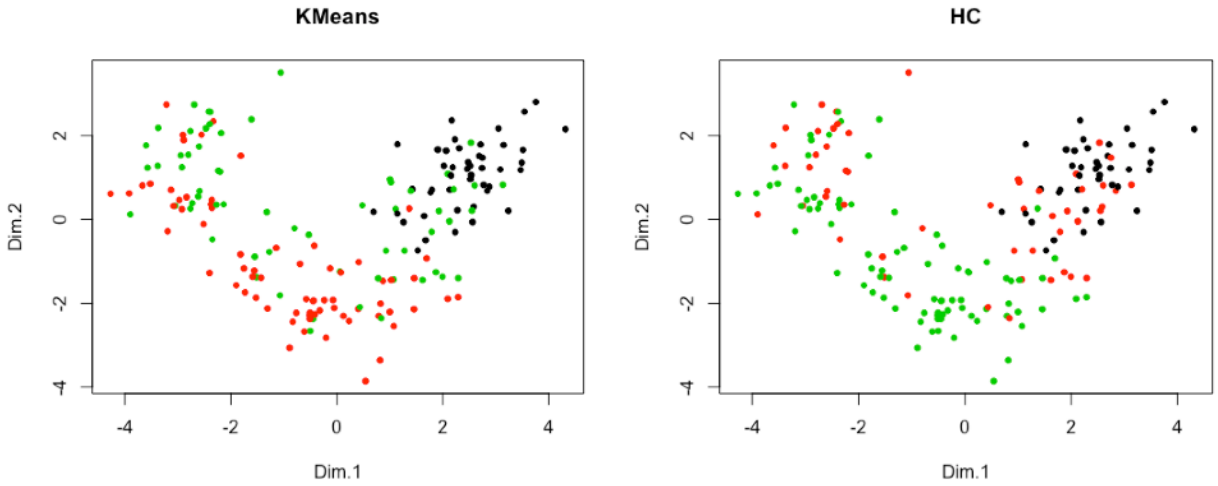
1.2.3.1. Hierarchical Clustering

Contrairement à l'analyse des donneurs de sang, la classification de ce jeu de données est mieux réussie: la classification hiérarchique ascendante parvient à classifier correctement **67,2%** (58 erreurs pour 119 succès) les vins du jeu de données.

1.2.3.2. KMeans

De la même manière, la méthode KMeans répond bien au problème en affichant **70%** de taux de réussite sur la classification (53 erreurs pour 124 succès).

1.2.4. Projection des classes



Nous observons cette fois que la classification est mieux effectuée par rapports aux classes apparues lors de la réduction de dimensions. Ceci est en accord avec les résultats obtenus, plus probants que lors de l'étude du jeu de données *transfusion*.

2. Partie 2

2.1. Classification

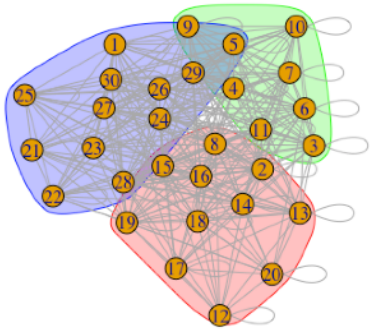
En appliquant une classification par KMeans au jeu de données, nous pouvons observer que le résultat n'est pas cohérent avec les labels fournis par le jeu de données. En effet, alors que chaque classe du jeu regroupe 10 éléments, l'algorithme de classification attribue 26 des 30 éléments à une même classe.

2.2. Graphe d'adjacence

Nous effectuons un produit matriciel de la D.T.M sur sa propre transposée. Ainsi nous lions chaque document avec ceux auxquels il est lié: c'est la matrice d'adjacence. La fonction `graph_from_adjency_matrix` du package igraph permet de construire le graphe d'adjacence.

2.3. Détection de communauté

Comme nous l'avons remarqué au début de la partie, l'algorithme KMeans ne classifie pas bien les objets. En effet, une classe est beaucoup plus présente que les deux autres, alors que les valeur dans classid montrent une répartition uniforme. En revanche, comparer les communautés trouvées aux labels du jeu de données montre **5 erreurs et 25 succès.** Nous pouvons donc affirmer que la classification est réussie avec 80%, mettant en avant la performance d'une étude de graphe par rapport à une classification KMeans dans notre cas.



3. Partie 3

3.1. Analyse descriptive

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Std
AT	1.81	13.5100	20345	19.65123	25.72	37.11	7.452473
V	25.36	41.7400	52080	54.30580	66.54	81.56	12.707893
AP	992.89	1009.1000	1012.940	1013.25908	1017.26	1033.30	5.938784
RH	25.56	63.3275	74975	73.30898	84.83	100.16	14.600269
PE	420.26	439.7500	451550	454.36501	468.43	495.76	17.066995

Les variables AP et VE sont homogènes, leur variance est basse par rapport à leur moyenne et les histogrammes confirmes une distribution normale. Les autres variables ne sont pas homogènes, sans avoir de distribution uniforme. Le tableau ci dessus montre aussi une différence d'ordre de grandeur, qui peut nécessiter une mise à l'échelle pour les algorithmes étant sensibles à ce paramètre.

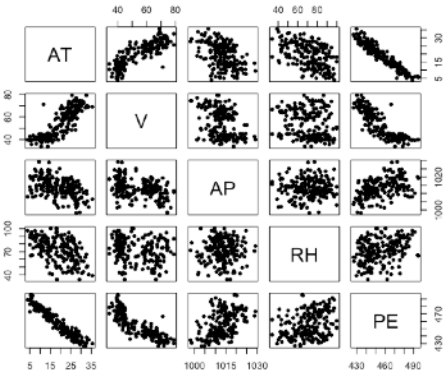
3.2. Affichage graphique

3.3. Corrélation des variables

	AT	V	AP	RH	PE
AT	1	**0.84**	-0.51	-0.54	**-0.95**
V	**0.84**	1	-0.41	-0.31	**-0.87**
AP	-0.51	-0.41	1	0.1	0.52
RH	-0.54	-0.31	0.1	1	0.39
PE	**-0.95**	**-0.87**	0.52	0.39	1

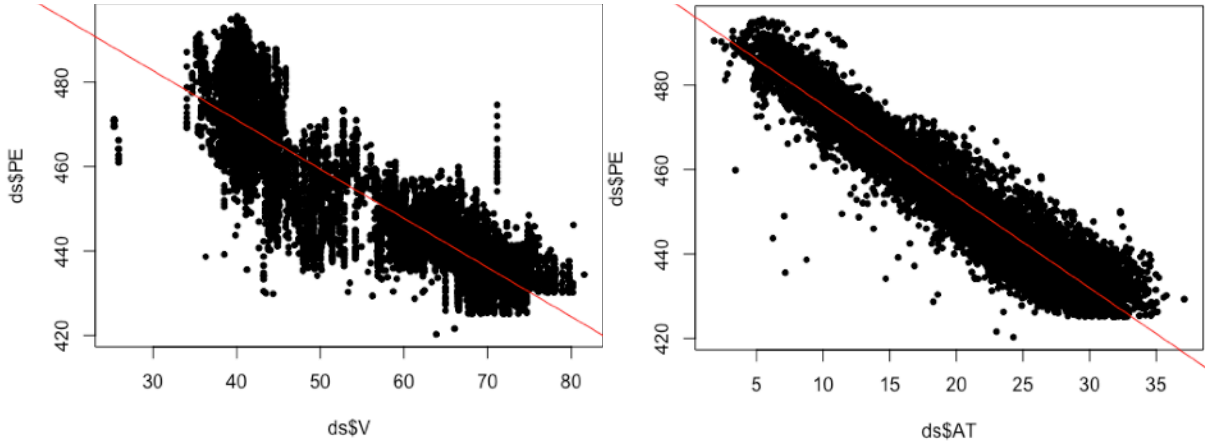
Nous observons que AT et PE sont très fortement corrélées (négativement). Les variables AT et V sont aussi très corrélées. Enfin, les variables V et PE sont très corrélées négativement.

3.4. Nuages de points



Nous observons facilement que les variables AT et V permettent de prédire la variable PE. En effet, les nuages de points montrent une directe corrélation en ayant deux valeurs propres très différentes (le nuage de poins est très « étiré »). Nous pouvons aussi confirmer les observations ci dessus concernant la corrélation entre les variables AT et V.

3.5. Régressions linéaires



Nous observons que le coefficient directeur de la droite de régression est négatif, donc les variable AT et V sont corrélées négativement à PE. De plus, nous nous apercevons graphiquement que la variable V n'est pas très bien représentée par une droite: le nuage de point semble suivre une loi plus complexe par rapport à PE.

3.6. Fonction de coût

	MSE	MAE
V	70.91	6.58
AT	29.43	4.29

En observant les erreurs des deux régressions, nous pouvons affirmer que la variable AT est mieux représentée par le modèle. En effet, les valeurs d'erreurs sont bien plus faibles dans le cas de la variable AT. Ceci confirme l'intuition visuelle dans les figure précédentes.