Le vin optimal

Dorian Hervé et Wael Ben Hadj Yahia

Enseignant : Louis Capitaine

Projet réalisé dans le cadre de l'UE Statistique non paramétrique
2019-2020

Contents

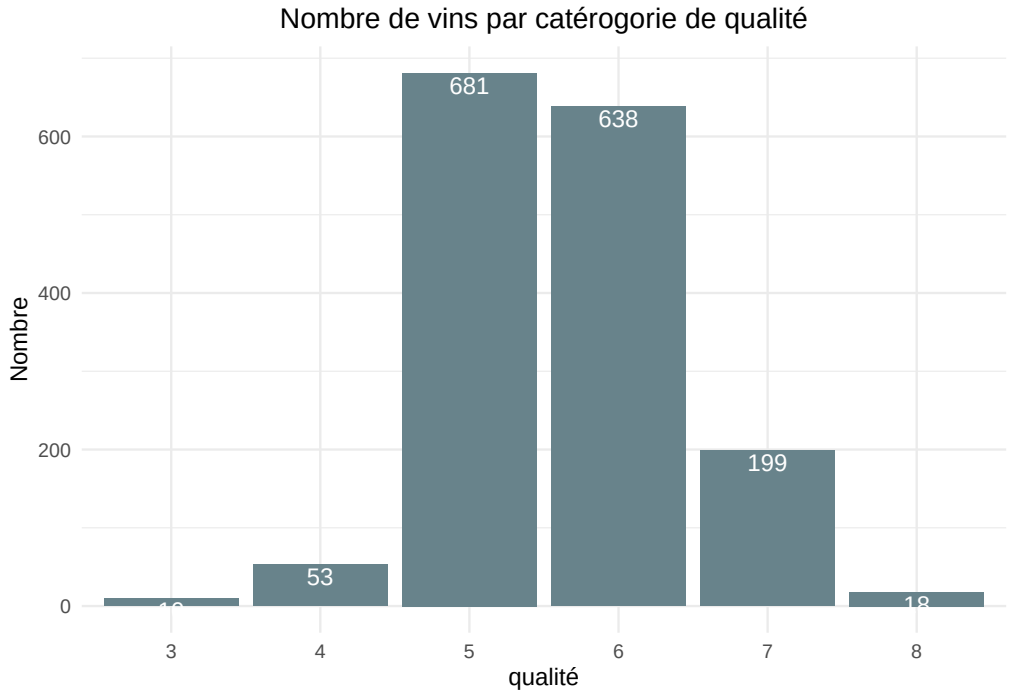
1 Introduction

Ce projet a été réalisé dans le cadre de l'UE Statistique non paramétrique. Le jeu de données utilisé est une base de données de vins qui comprend des teneurs en certains constituants ainsi qu'une variable prenant des valeurs sur une échelle de 1 à 10. Celles-ci représentent la qualité de ce vin (1 étant mauvais et 10 excellent). Les données sont disponibles à l'adresse suivante : https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009/

L'objectif de ce projet est de faire une analyse complète de cette base de données, notamment d'identifier les variables les plus influentes sur la qualité du vin, ainsi que de proposer des classifieurs qui permettent de catégoriser les vins.

Ce rapport a été réalisé par Wael Ben Hadj Yahia et Dorian Hervé. Nous tenons à remercier Louis Capitaine pour nous avoir encadré lors de la réalisation de ce projet.

Avant de commencer notre étude, afin de se familiariser avec les données qui seront utilisées par la suite, nous proposons d'abord un histogramme destiné à montrer la répartition des vins dans les différentes catégories :



Cet histogramme nous permet d'observer une très forte hétérogénéité dans la idstribution des qualités des vins. En effet, nous avons 1319 des 1599 vins de qualité 5 ou 6, soit 82.5%, par opposition aux 28 vins de qualité 3 ou 8.

2 Mise en évidence des variables explicatives importantes

Dans cette section, l'objectif est d'identifier les variables explicatives influentes sur la qualité du vin.

2.1 ACP

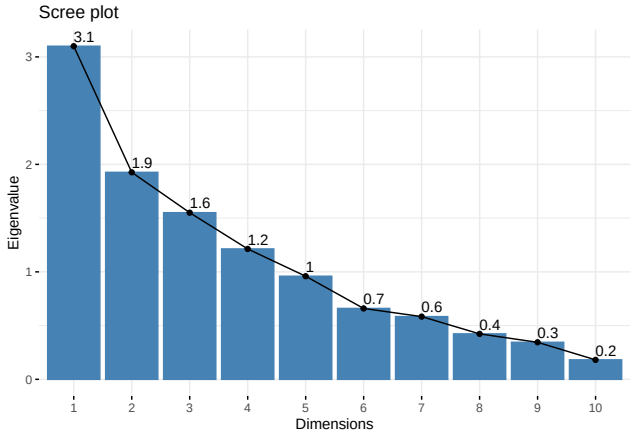
En ce qui concerne cette première partie, nous proposons une ACP afin de faire un premier travail de mise en évidence des variables clés.

Il s'agit d'abord de créer notre objet PCA en ACP normée, c'est à dire sur la matrice des corrélations (matrice Z obtenue en centrant et réduisant la matrice des données initiales X). La variable réponse "quality" est mise en variable supplémentaire dans toute la partie.

Affichons d'abord la matrice des corrélations afin de faire une première analyse :

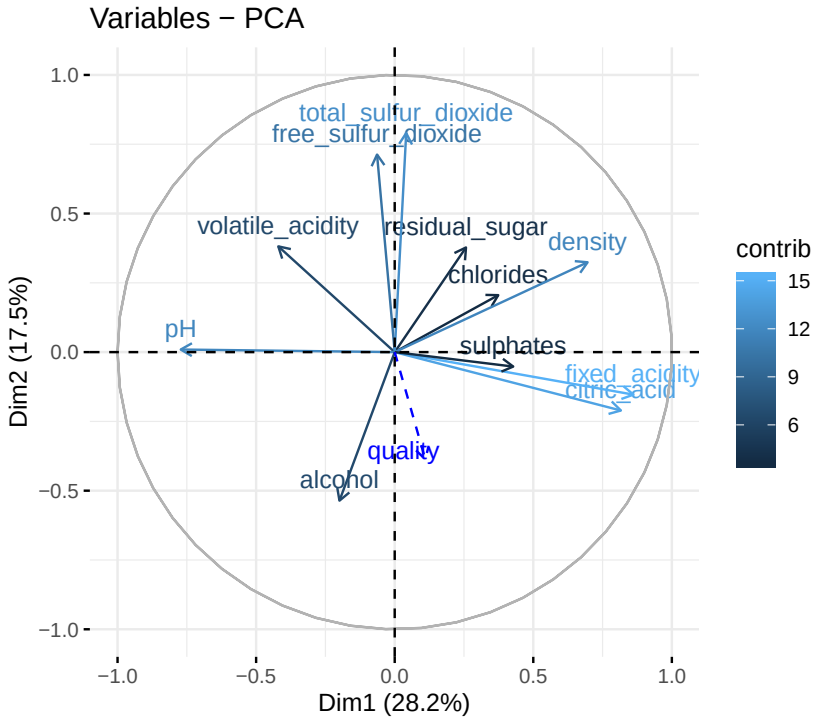
Les variables "fixed acidity" & "density", "fixed acidity" & "citric acid", "total sulfur dioxide" & "total sulfur dioxide" sont positivement corrélées entre elles. Ceci veut dire qu'un vin qui est riche en l'un de ces constituants a tendance à être riche en les autres aussi, et inversement (une faible teneur en l'un de ces constituants a tendance à se traduire par une faible teneur des autres). De même, nous avons naturellement une corrélation négative entre "pH" & "fixed acidity".

Intéressons nous maintenant aux valeurs propres afin de déterminer la dimension de projection que nous conserverons. L'éboulis des valeurs propres montre qu'après la cinquième valeur propre, la hauteur des barres décroît de manière homogène, (donc de la forme d'un "pli de coude"), donc nous devrions conserver 5 axes pour notre analyse.



La règle de Kaiser est une autre méthode plus formelle (et moins approximative) ayant le même objectif que la méthode de l'éboulis. Elle consiste à garder les valeurs propres plus grandes que la quantité InertieProjetée/p. Or comme nous sommes dans le cas d'une ACP normée, l'inertie projetée (qui vaut la somme des variances des colonnes, et celles-ci étant centrées et réduites, chaque colonne est de variance 1) vaut p avec p étant le nombre de colonnes de X, la matrice des données), donc cela revient à conserver les valeurs propres supérieures à 1. Nous devrions donc conserver les 5 premiers axes de projection en toute rigueur. Par souci de redondance, nous en conserverons uniquement 4 pour la suite.

Intéressons-nous au cercle de corrélation par rapport aux axes 1 et 2 :



Seules les variables “fixed acidity”, “citric acid”, “total sulfur dioxide”, “free sulfur dioxide” et “pH” sont assez bien projetées, nous ne prenons donc pas en compte les autres.

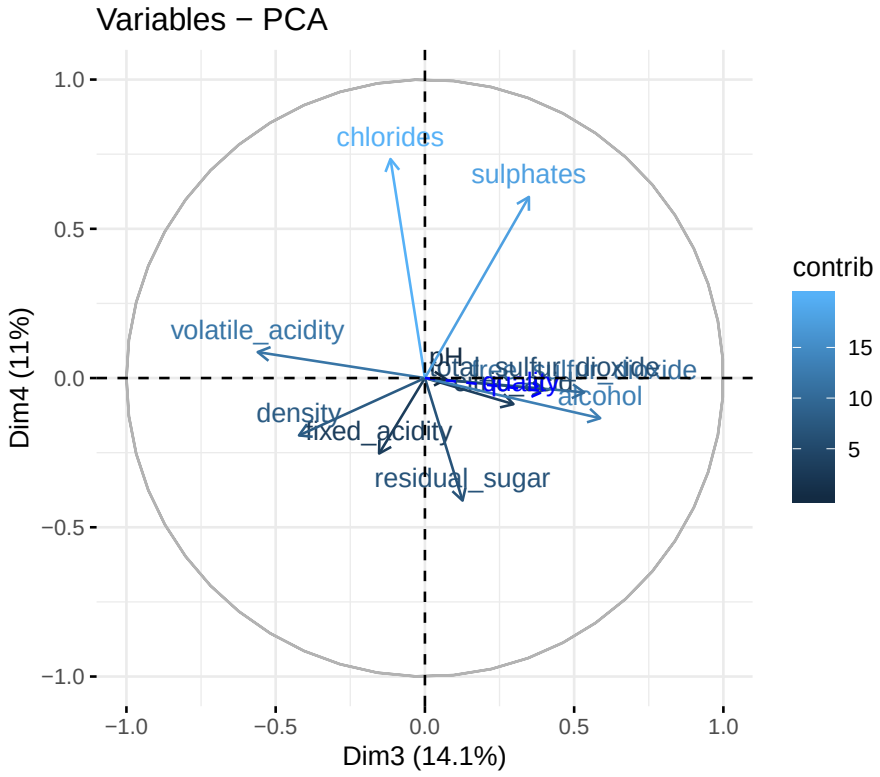
Nous pouvons séparer nos variables en 3 groupes : “fixed acidity” et “citric acid” qui sont fortement corrélées à l’axe 1 et décorrélées à l’axe 2, “pH” qui est négativement corrélé à l’axe 1 et décorrélé à l’axe 2, et enfin “total sulfur dioxide” et “free sulfur dioxide” qui sont fortement corrélées à l’axe 2 et décorrélées à l’axe 1.

En ce qui concerne la corrélation entre ces 3 groupes : nous pouvons remarquer qu’ils forment entre-eux quasiment un angle droit ce qui ne laisse présager aucune corrélation entre les variables qui les forment.

Ces observations appuient nos constatations faites lors de l’analyse de la matrice de corrélation.

Le plan engendré par ces deux axes (1 et 2) expliquerait environ 45% (28+17) de la variabilité des données initiale. L’axe 1 semble être “l’axe de l’acidité” et l’axe 2 semble plutôt être expliqué par les sulfites.

Intéressons-nous au cercle de corrélation par rapport aux axes 3 et 4 :

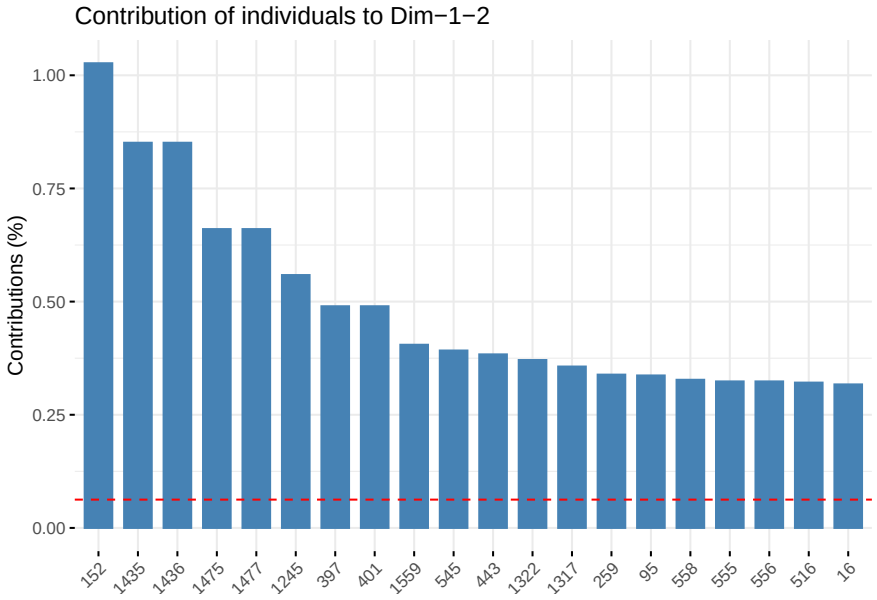


Contrairement au dernier cercle de corrélation, nous avons ici que les “chlorides” et les “sulphates” sont les deux seules variables à être relativement bien projetées.

Nous pouvons remarquer que les “chlorides” et les “sulphates” sont toutes deux positivement corrélées avec l’axe 4. En revanche nous pouvons voir que les “chlorides” sont quasiment décorrélés de l’axe 3 alors que les “sulphates” sont légèrement positivement corrélés avec l’axe 3. Remarquons que ces deux variables sont légèrement corrélées entre elles (angle assez petit entre leurs flèches de projection).

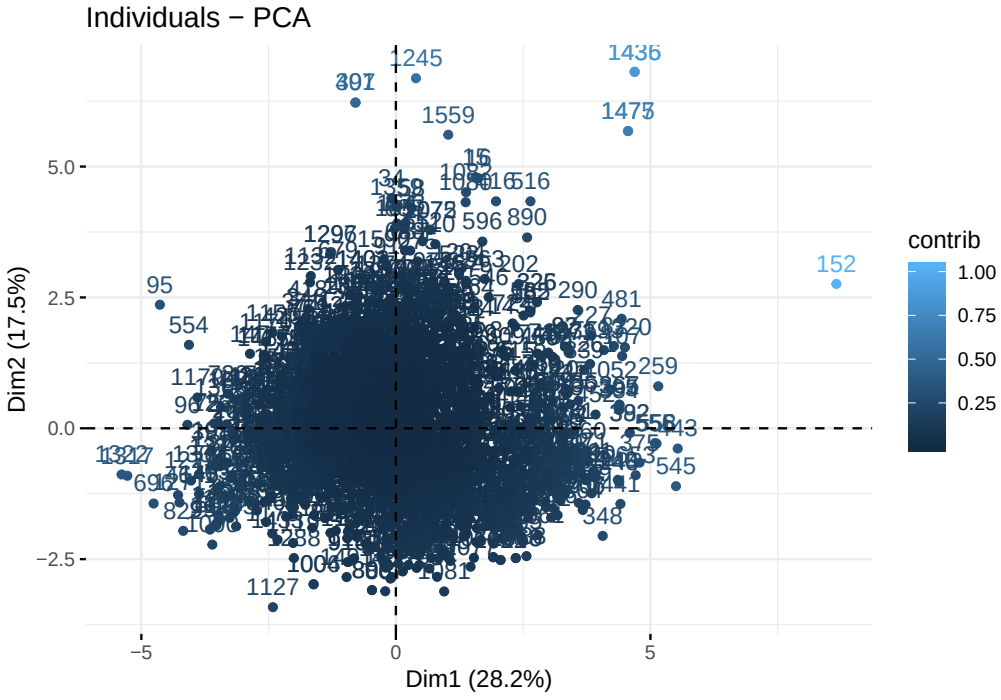
Le plan engendré par les axes 3 et 4 expliquerait 25% (14+11) de la variabilité des données initiales. L’axe 3 semble être plutôt l’axe des “sulphates”, et de même pour l’axe 4 à la différence que celui-ci serait également expliqué par les “chlorides”.

Regardons les vins les plus contribuants au plan (1,2) :



Les vins 152, 1435 et 1436 sont les 3 vins les plus contribuants du plan (1,2), avec le vin 152 qui contribue à plus de 1% à lui tout seul (ce qui est considérable sur une base de données de 1600 entrées). En vérifiant cette entrée, on constate que le vin 152 a une valeur de 1 à la colonne “citric acid” (c’est la plus grande valeur) alors que la moyenne de cette colonne est à 0,27 .

Intéressons nous maintenant à la projection des vins (points-individus) sur le premier plan factoriel engendré par les axes 1 et 2 :

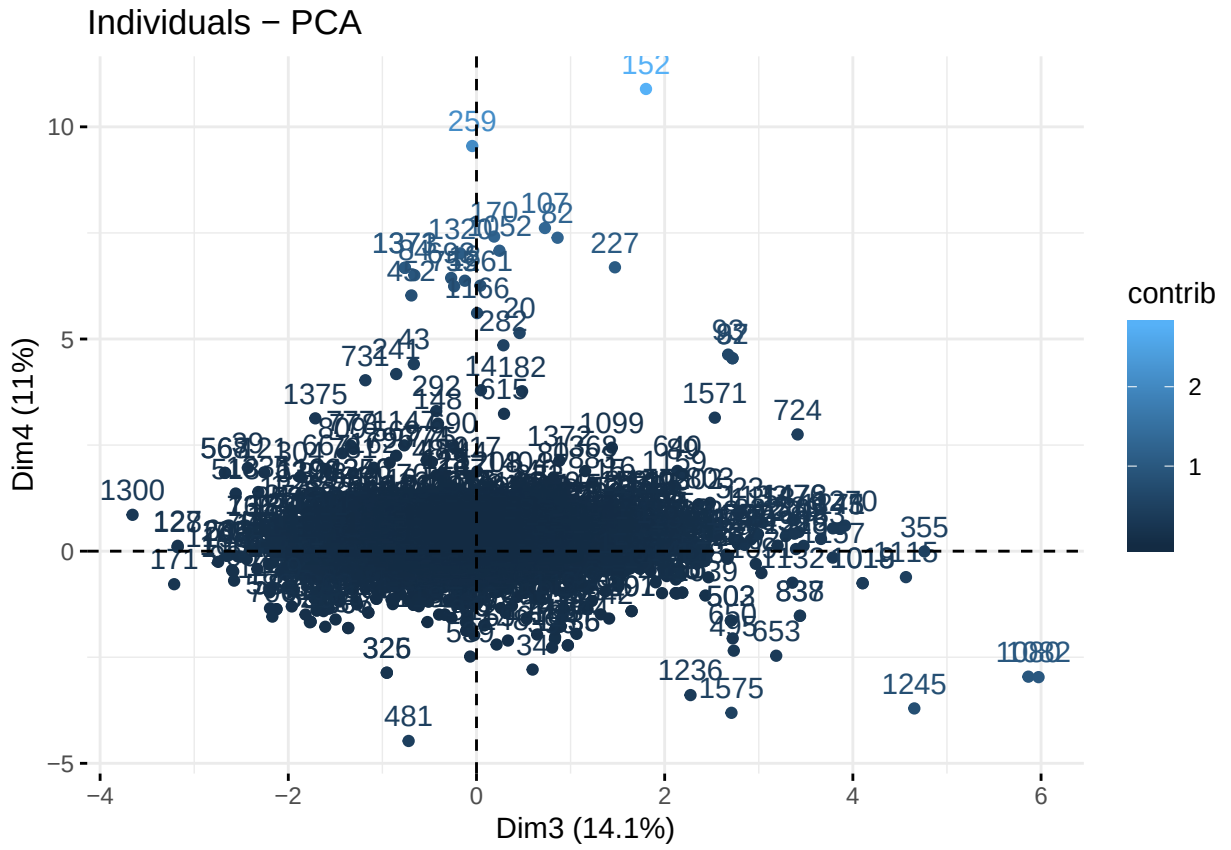


On s'intéresse aux points qui se distinguent des autres et qui s'éloignent du centre du nuage, tels que par exemple les points 152 et 1436.

Le vin 152 a une assez grande valeur sur le premier axe et la plus grande valeur sur l'axe 2, on s'attend donc à ce que ce vin soit très riche en acides et faible en pH (c'est bien le cas avec une teneur en acide citrique à 1 et un pH à 2.74, qui est le plus faible de la base de données).

Le vin 1436 a une assez grande valeur sur le deuxième axe et la plus grande valeur sur l'axe 1, ce qui est aussi corroboré en s'intéressant à ses teneurs en sulfites.

Intéressons-nous maintenant à la projection des plats (points-indidividus) sur le second plan factoriel engendré par les axes 3 et 4.



Une analyse similaire peut etre faite pour le plan (3,4) (notamment avec le vin 152 qui est le plus riche en "sulphates" et "chlorides"), mais ceci n'est pas très interessant dans notre cas ici.

En effet, cette analyse en ACP n'est pas concluante ici car la variable d'interet "quality" n'a pas été assez bien projetée sur les plans que nous avons considéré, et donc nous ne pouvons pas établir de corrélation entre la qualité d'un vin et d'autres composantes (remarquons tout de meme que l'alcool est la variable la plus corrélée à la qualité, mais cette corrélation n'est pas certaine à cause de la mauvaise projection). Nous proposons donc de continuer cette recherche en adoptant d'autres méthodes.

2.2 Régression

Dans cette section, le but est de construire le meilleur modèle de régression possible avec les données que l'on dispose, et d'en étudier les propriétés afin de mettre en évidence (ou non) des liens significatifs entre des variables explicatives et la variable réponse quality.

Nous proposons d'abord l'utilisation du critère de l'AIC afin de détérminer les variables que nous conserverons dans notre modèle optimal.

```
## Start:  AIC=-1375.49
## dat$quality ~ fixed_acidity + volatile_acidity + citric_acid +
##     residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     density + pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
## - density               1     0.287 666.70 -1376.8
## - fixed_acidity         1     0.389 666.80 -1376.5
## - residual_sugar        1     0.498 666.91 -1376.3
## - citric_acid           1     0.646 667.06 -1375.9
## <none>                              666.41 -1375.5
## - free_sulfur_dioxide   1     1.694 668.10 -1373.4
## - pH                    1     1.957 668.37 -1372.8
## - chlorides             1     8.391 674.80 -1357.5
## - total_sulfur_dioxide  1     8.427 674.84 -1357.4
## - sulphates             1    26.971 693.38 -1314.0
## - volatile_acidity      1    33.620 700.03 -1298.8
## - alcohol               1    45.672 712.08 -1271.5
##
## Step:  AIC=-1376.8
## dat$quality ~ fixed_acidity + volatile_acidity + citric_acid +
##     residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
## - fixed_acidity         1     0.108 666.81 -1378.5
## - residual_sugar        1     0.231 666.93 -1378.2
## - citric_acid           1     0.654 667.35 -1377.2
## <none>                              666.70 -1376.8
## - free_sulfur_dioxide   1     1.829 668.53 -1374.4
## - pH                    1     4.325 671.02 -1368.5
## - total_sulfur_dioxide  1     8.728 675.43 -1358.0
## - chlorides             1     8.761 675.46 -1357.9
## - sulphates             1    27.287 693.98 -1314.7
## - volatile_acidity      1    35.000 701.70 -1297.0
## - alcohol               1   119.669 786.37 -1114.8
##
## Step:  AIC=-1378.54
## dat$quality ~ volatile_acidity + citric_acid + residual_sugar +
##     chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
## - residual_sugar        1     0.257 667.06 -1379.9
## - citric_acid           1     0.565 667.37 -1379.2
```

```
## <none>                              666.81 -1378.5
## - free_sulfur_dioxide   1     1.901 668.71 -1376.0
## - pH                    1     7.065 673.87 -1363.7
## - chlorides             1     9.940 676.75 -1356.9
## - total_sulfur_dioxide  1    10.031 676.84 -1356.7
## - sulphates             1    27.673 694.48 -1315.5
## - volatile_acidity      1    36.234 703.04 -1295.9
## - alcohol               1   120.633 787.44 -1114.7
##
## Step:  AIC=-1379.93
## dat$quality ~ volatile_acidity + citric_acid + chlorides + free_sulfur_dioxide +
##     total_sulfur_dioxide + pH + sulphates + alcohol
##
##                         Df Sum of Sq    RSS     AIC
## - citric_acid            1     0.475 667.54 -1380.8
## <none>                               667.06 -1379.9
## - free_sulfur_dioxide    1     2.064 669.13 -1377.0
## - pH                     1     7.138 674.20 -1364.9
## - total_sulfur_dioxide   1     9.828 676.89 -1358.5
## - chlorides              1     9.832 676.89 -1358.5
## - sulphates              1    27.446 694.51 -1317.5
## - volatile_acidity       1    35.977 703.04 -1297.9
## - alcohol                1   122.667 789.73 -1112.0
##
## Step:  AIC=-1380.79
## dat$quality ~ volatile_acidity + chlorides + free_sulfur_dioxide +
##     total_sulfur_dioxide + pH + sulphates + alcohol
##
##                         Df Sum of Sq    RSS     AIC
## <none>                               667.54 -1380.8
## - free_sulfur_dioxide    1     2.394 669.93 -1377.1
## - pH                     1     7.073 674.61 -1365.9
## - total_sulfur_dioxide   1    10.787 678.32 -1357.2
## - chlorides              1    10.809 678.35 -1357.1
## - sulphates              1    27.060 694.60 -1319.2
## - volatile_acidity       1    42.318 709.85 -1284.5
## - alcohol                1   124.483 792.02 -1109.4
```

Le modèle optimal prendrait donc en compte les variables suivantes : “free sulfur dioxide”, “pH”, “total sulfur dioxide”, “chlorides”, “sulphates”, “volatile acidity”, et “alcohol”. Il s’agirait des variables qui expliquent au mieux la variabilité de la qualité des vins.

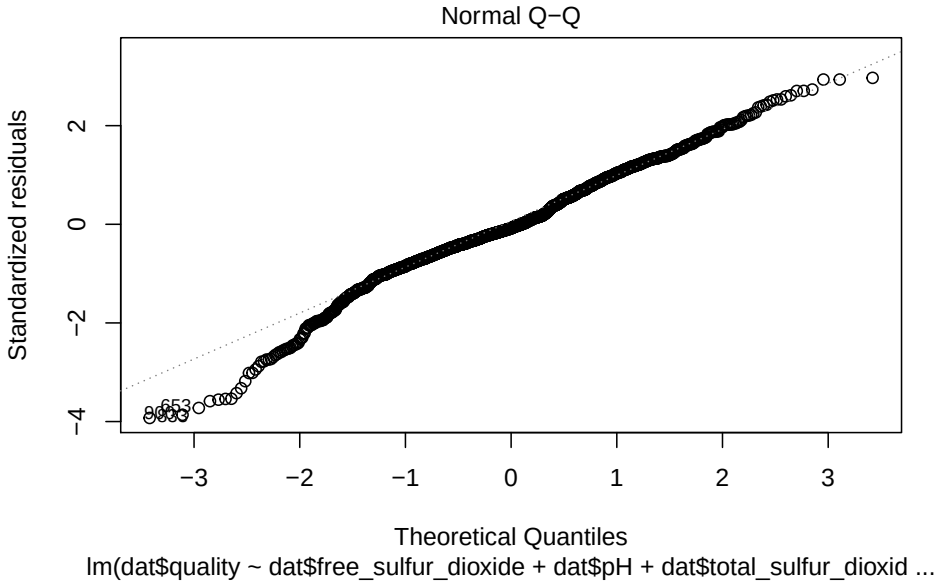
Etudions ce modèle :

```
##
## Call:
## lm(formula = dat$quality ~ dat$free_sulfur_dioxide + dat$pH +
##     dat$total_sulfur_dioxide + dat$chlorides + dat$sulphates +
##     dat$alcohol, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.61472 -0.37680 -0.05245  0.46128  1.97551
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 4.595638   0.415013  11.073  < 2e-16 ***
## dat$free_sulfur_dioxide     0.007332   0.002179   3.365 0.000784 ***
## dat$pH                     -0.822355   0.116064  -7.085 2.08e-12 ***
## dat$total_sulfur_dioxide -0.004302   0.000703  -6.120 1.17e-09 ***
## dat$chlorides            -2.744717   0.402970  -6.811 1.37e-11 ***
## dat$sulphates             1.158157   0.109717  10.556  < 2e-16 ***
## dat$alcohol               0.318974   0.017045  18.714  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6677 on 1592 degrees of freedom
## Multiple R-squared:  0.3189, Adjusted R-squared:  0.3163
## F-statistic: 124.2 on 6 and 1592 DF,  p-value: < 2.2e-16
```

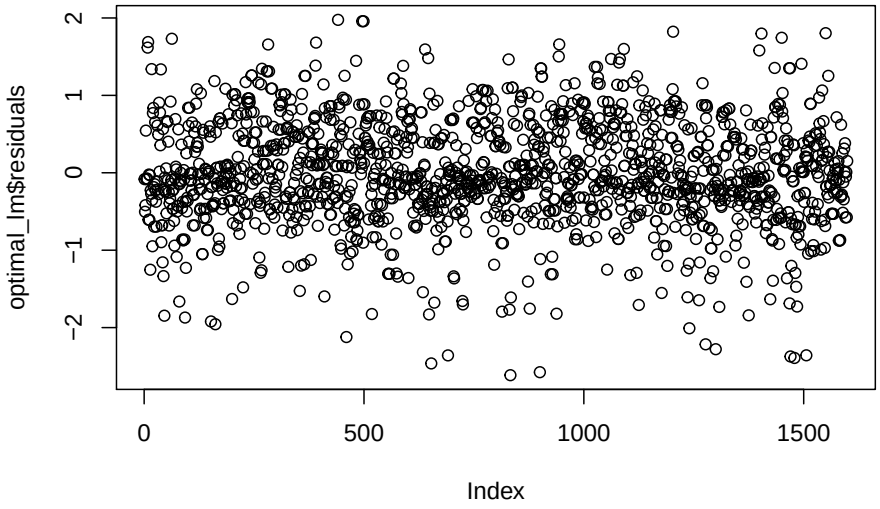
Il permet d'expliquer environ 32% de la variabilité de la qualité des vins, ce qui n'est pas très satisfaisant.

Nous avons un coefficient linéaire négatif entre la qualité d'un vin et : son pH, sa teneur en chlorides, et sa teneur en sulfites totales, ainsi qu'un coefficient linéaire positif entre la qualité du vin et sa teneur en sulfates, en dioxyde de sulfites et en alcool.

Néanmoins, vérifions que notre modèle possède de bonnes qualités :

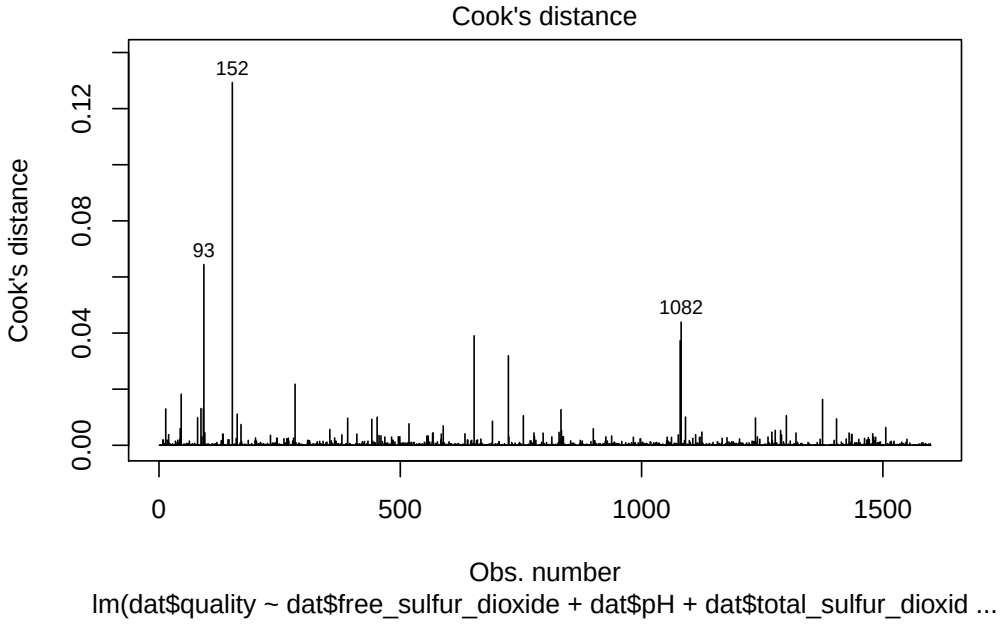


Le QQ-plot suggère des erreurs normales puisque les quantiles observés et les quantiles théoriques (obtenus si la distribution est normale) forment une droite. L'hypothèse de normalité semble être vérifiée.



De plus, la sortie ci-dessus montre que les résidus ne prennent aucune forme particulière, ce qui corrobore le fait que les résidus suivent bien une distribution normale.

Regardons maintenant le graphe des distances de Cook :



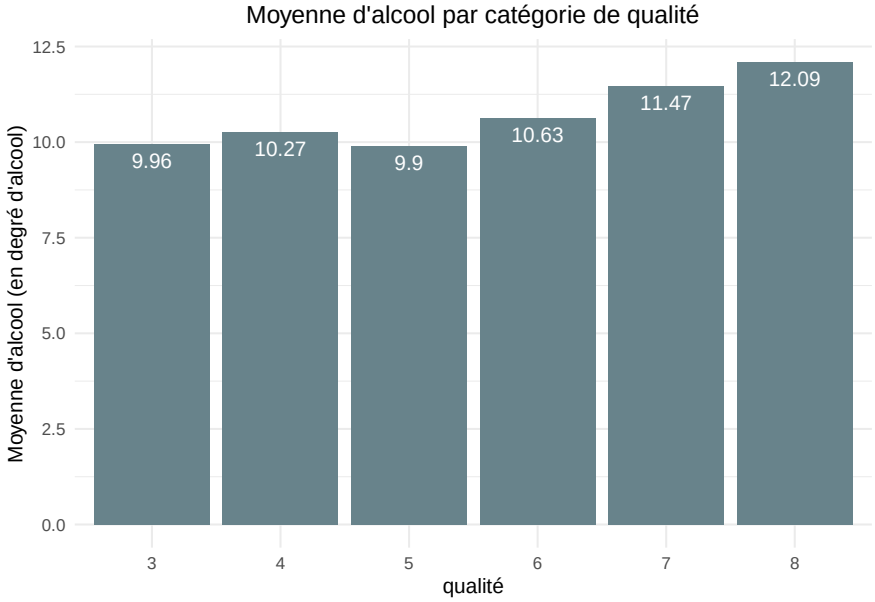
Les courbes de niveaux de leverage de distances de Cook ne sont pas égales, et 3 vins sont trop contributifs (on y retrouve notamment le vin 152). Dans de prochaines estimations, ils devraient etre supprimés de l'étude afin d'avoir une meilleur qualité du modèle.

En conclusion, le modèle de régression linéaire n'est pas approprié ici (il n'explique que 32% de la variabilité de la qualité des vins). Ceci peut etre lié au fait que nous essayons d'estimer le lien entre des variables explicatives avec une variable réponse qualitative (la qualité des vins) par un lien linéaire. Or ici, il ne s'agit pas d'un problème de régression, mais plutot d'un problème de classification. Nous nous penchons donc par la suite à d'autres méthodes.

2.3 Statistiques descriptives

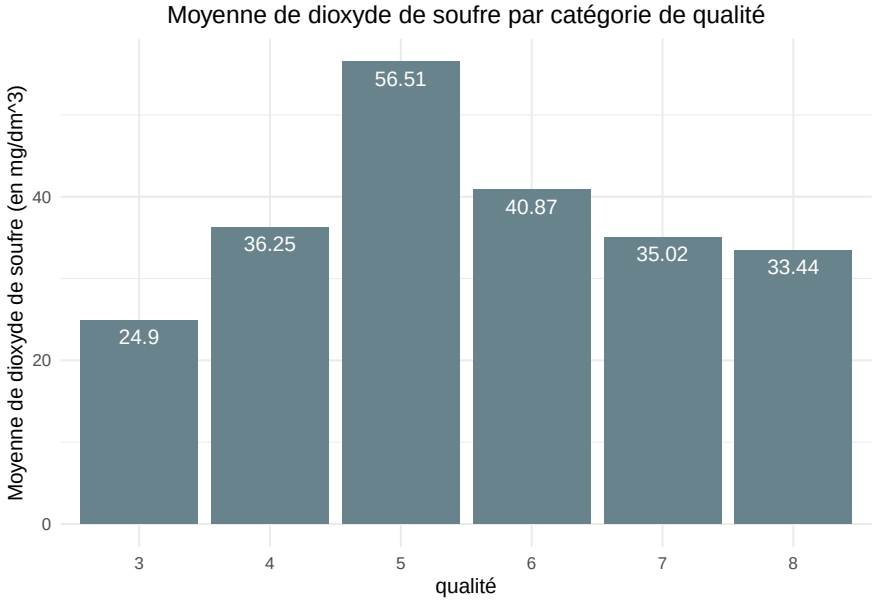
Grâce aux sections précédentes, nous avons pu avoir une première intuition sur les variables influentes sur la qualité du vin. Observons donc la teneur moyenne de ces composantes influentes en distinguant les catégories de qualité du vin. Le but de cette section est d'avoir un apercu général, et n'apporte pas de conclusions formelles.

Nous nous intéressons d'abord à la teneur en alcool par catégorie.



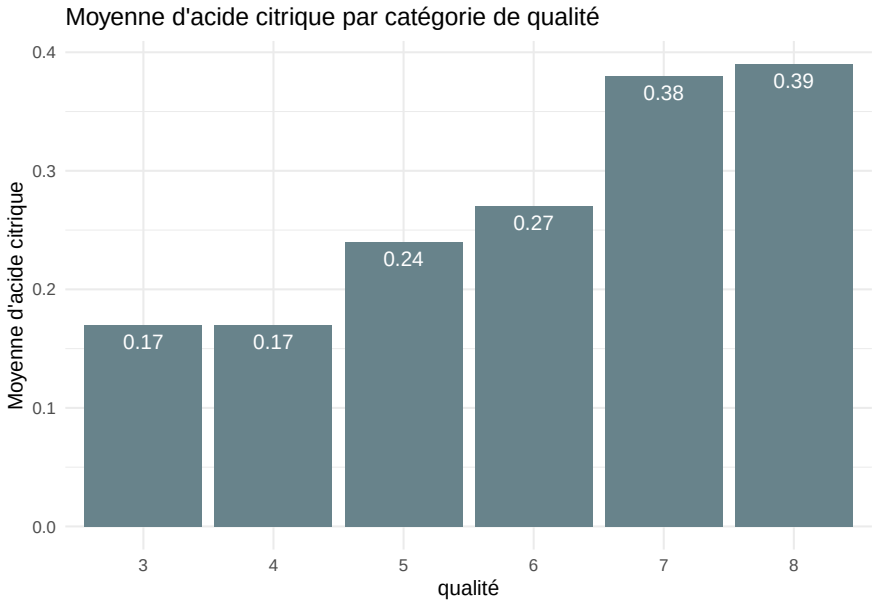
Nous observons un lien quasi linéaire entre la teneur en alcool et la qualité d'un vin : plus la teneur en alcool est élevée plus la qualité du vin semble être élevée aussi avec une moyenne de 12.09% dans la catégorie optimale.

Intéressons nous maintenant à la teneur en dioxyde de soufre par catégorie de qualité.



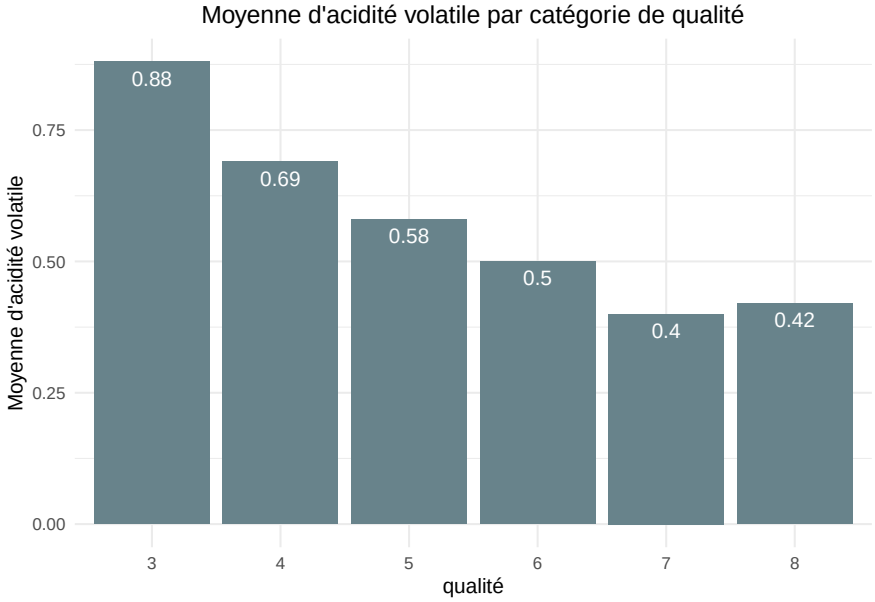
A priori ce graphe n'est pas très concluant mis à part le fait que les vins moyens soient relativement riches en dioxyde de soufre.

Observons le niveau moyen en acide citrique :



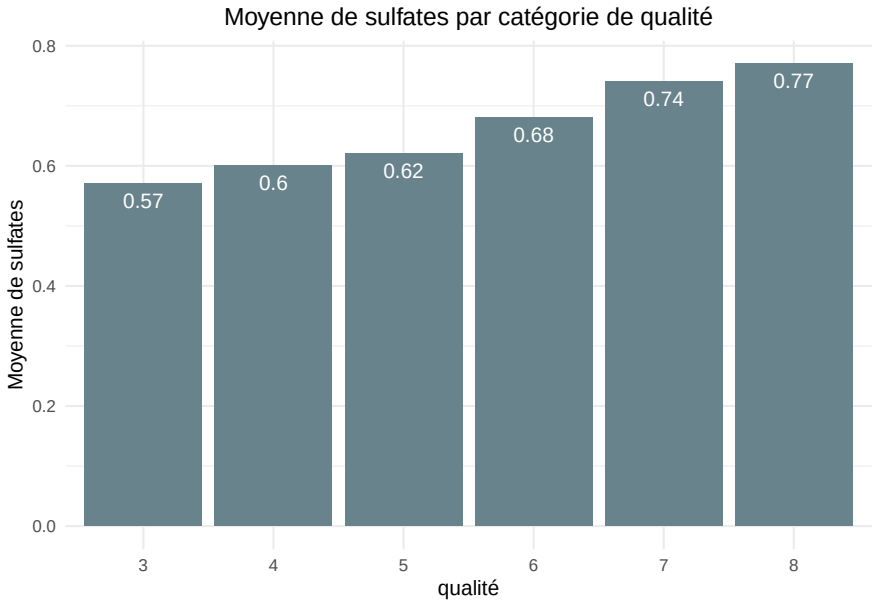
Il semble assez clair que la teneur en acide citrique et la qualité du vin sont positivement corrélées.

Observons maintenant l'acidité volatile moyenne par catégorie.



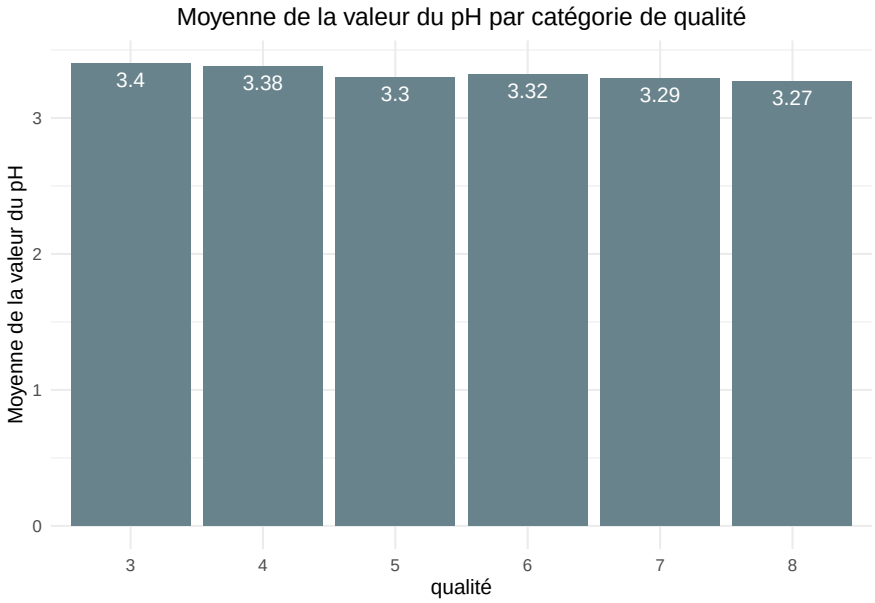
La qualité du vin semble avoir une tendence à augmenter lorsque l'acidité volatile a tendance à baisser.

Intéressons nous au niveau moyen de sulphates.



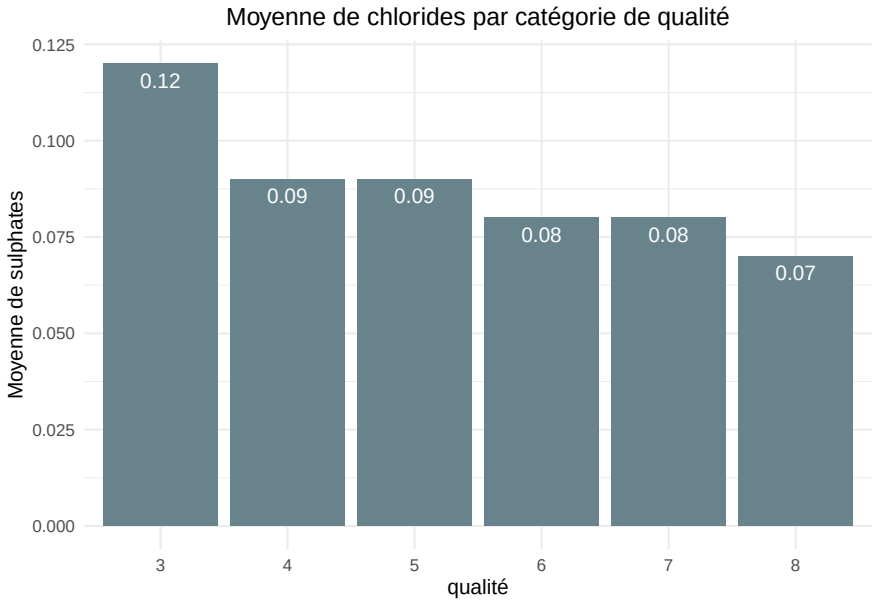
Plus la qualité du vin est haute, plus la teneur moyenne en sulfates semble augmenter de manière sensible.

Observons le niveau moyen du pH.



Il semble y avoir un équilibre entre acide citrique et acide volatile qui fait de sorte que le pH reste assez stable entre les catégories.

Enfin, intéressons nous à la teneur moyenne en chlorides.



Nous observons une tendance décroissante entre la teneur en chlorides et la qualité du vin.

2.4 Tests d'hypothèse

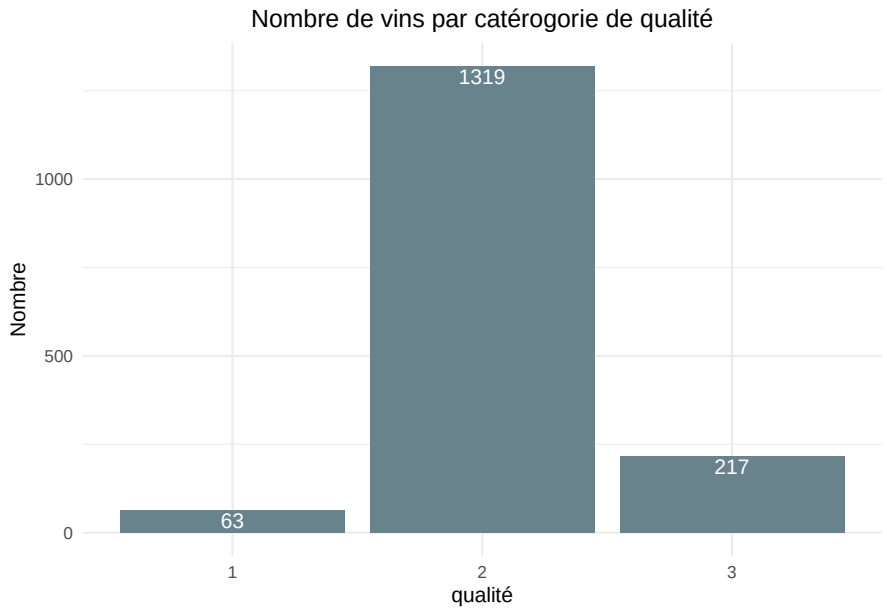
Dans cette section, nous allons effectuer des tests d'hypothèse afin de corroborer/refuter rigoureusement les hypothèses que nous venons de faire au vu des résultats graphiques précédents.

Pour la suite, nous allons réaliser des tests d'hypothèse avant de nous intéresser à la construction de classifieurs de vins.

Avant d'y procéder, nous choisissons de regrouper nos qualités de vins en 3 différentes catégories : 1 pour mauvais, 2 pour moyen et 3 pour bon. Dans la catégorie 1, nous regroupons les vins de qualité 3-4, les vins de qualité 5-6 pour moyen et 7-8 pour les bons vins.

En effet, ce regroupement permet de diluer les disparités entre les proportions des groupes. Nous avions 10 vins sur 1600 pour la catégorie minoritaire contre 63 sur 1600 après relabélisation de ce champ. Par ailleurs nous estmimons qu'il est plus intuitif de parler d'un bon vin plutôt que d'un vin de qualité 7 ou 8.

Voici donc la nouvelle répartition des vins par catégorie.



Effectuons à présent un test afin de voir s'il n'y a pas de différence entre la teneur en alcool selon le fait qu'un vin soit de catégorie 1 ou 3.

```
##
##  Two Sample t-test
##
## data:  sample_c1 and sample_c3
## t = -7.2427, df = 98, p-value = 1.007e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.760661 -1.003339
## sample estimates:
## mean of x mean of y
##    10.080    11.462
```

La p-valeur associée à ce test est de $1.0072034 \times 10^{-10}$, ce qui veut dire que notre réalisation de statistique de test a $1.0072034 \times 10^{-8}\%$ de chances d'avoir une valeur au moins aussi extreme que celle qui est observée, et ce sous l'hypothèse selon laquelle il n'y aurait pas de différence significative dans la teneur en alcool selon la catégorie. Au risque 5%, nous rejetons donc cette hypothèse, et nous concluons sur le fait que la teneur en alcool influe sur la qualité du vin.

Réalisons maintenant un test pour la variable "sulphates".

```
##
##  Two Sample t-test
##
## data:  sample_c1 and sample_c3
## t = -6.0903, df = 98, p-value = 2.209e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.218233 -0.110967
## sample estimates:
## mean of x mean of y
##    0.5752    0.7398
```

Au risque 5%, nous rejetons donc l'hypothèse nulle, et nous concluons sur le fait que la teneur en sulfates influe sur la qualité du vin.

A présent, réalisons un test sur la variable "volatile acidity" :

```
##
##  Two Sample t-test
##
## data:  sample_c1 and sample_c3
## t = 7.5072, df = 98, p-value = 2.811e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2274659 0.3909341
## sample estimates:
## mean of x mean of y
##    0.7058    0.3966
```

Encore une fois, nous rejetons l'hypothèse selon laquelle il n'y a pas de différence en teneur d'acide volatile suivant la catégorie des vins.

Enfin, intéressons nous à la variable dioxyde de soufre.

```
##
##  Two Sample t-test
##
## data:  sample_c1 and sample_c3
## t = -0.45676, df = 98, p-value = 0.6489
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -17.31663  10.83663
## sample estimates:
## mean of x mean of y
##     34.76     38.00
```

Cette fois-ci, avec une p-valeur de 0.6488526 nous ne rejettons pas l'hypothèse selon laquelle la teneur en dioxyde de soufre n'est pas significativement différente suivant les catégories de vins.

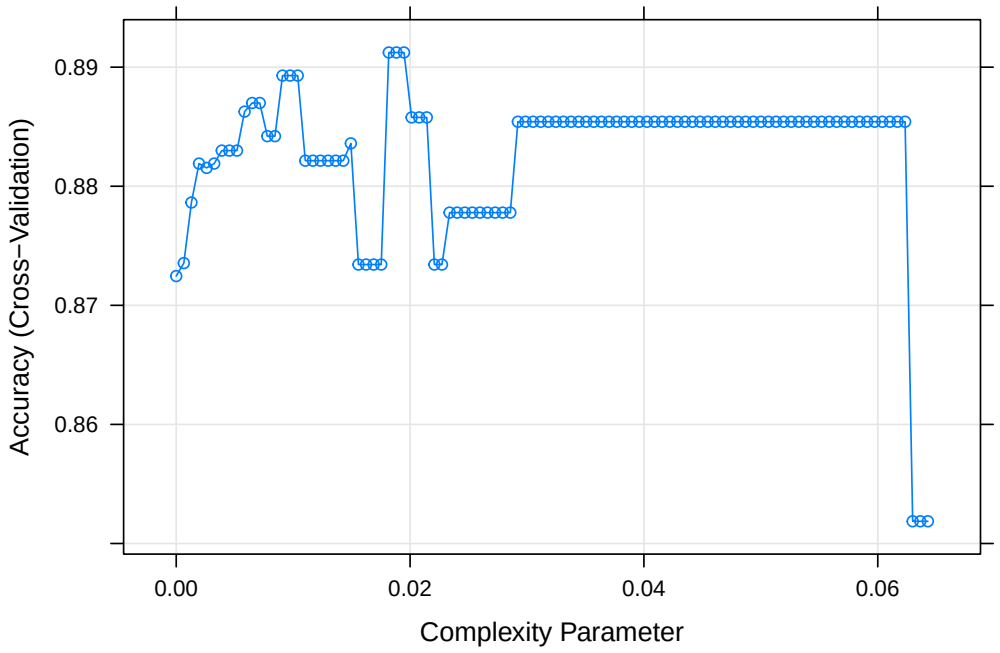
En conclusion de cette partie, les méthodes considérées précédemment nous permettent de définir les variables alcool, sulfates et acidité volatile comme étant les variables importantes. Ainsi la qualité d'un vin pourrait être expliqué à travers ces variables. Il semblerait qu'un bon vin est un vin riche en alcool et en sulfates, et faible en acidité volatile.

3 Classification

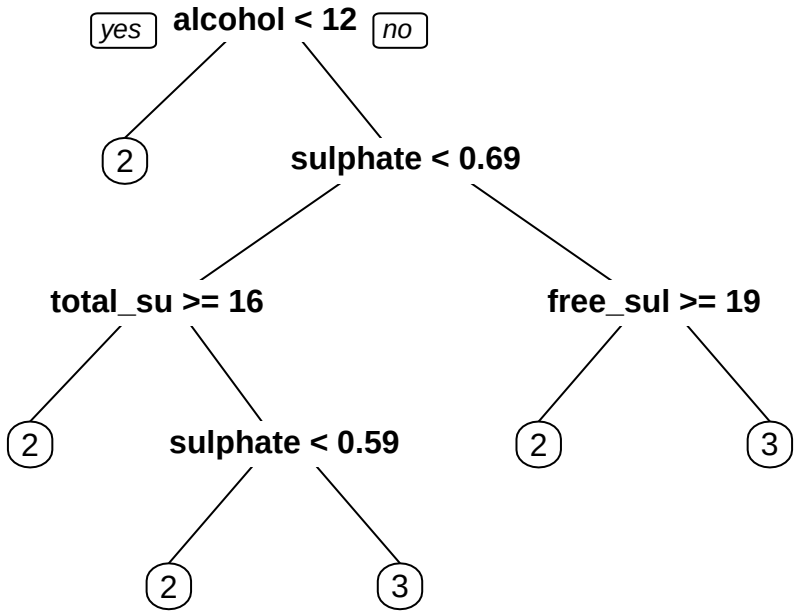
Dans cette section, le but est de proposer des classifieurs de vin et d'en étudier les performances afin de trouver la méthode la plus adaptée à notre problème.

3.1 Arbres CART

Avant de tracer notre arbre de décision, nous voulons trouver le paramètre *cp* (complexity paramter) optimal qui permet de réduire les chances de sur-apprentissage et d'obtenir un arbre suffisamment grand. Une trop petite valeur de *cp* mène à du sur-apprentissage tandis qu'une valeur trop grande de *cp* mène à un arbre trop petit. Une valeur optimale de *cp* peut être estimée en testant plusieurs valeurs de *cp* différentes et en utilisant des approches de validation croisée afin de déterminer la précision de prédiction du modèle correspondant. Le meilleur *cp* est ensuite défini comme étant celui qui maximise la précision de validation croisée.



Nous obtenons alors un $cp = 0.019$ et nous utilisons ce paramètre pour la construction de l'arbre ci-dessous.



Il semblerait donc qu'un vin fort en alcool et en sulfates et faible en dioxide de soufre est un vin de bonne qualité. Nous retrouvons donc une découpe cohérente avec la partie précédente. Remarquons tout de même l'absence de la classe 1 de vin, cela pourrait s'expliquer par son effectif trop faible.

Nous choisissons d'estimer l'erreur de classification empirique par la méthode du leave-one-out. Nous obtenons une erreur de 14.7%. Cependant notre arbre a classé aucun vin en qualité 1 mais cela impacte très peu le taux d'erreur puisque l'effectif de vins de classe 1 est trop faible. Cette méthode donne des résultats peu satisfaisants dans notre cas.

3.2 Forêts aléatoires

Dans cette section, nous allons nous intéresser au classifieur par forêt aléatoire.

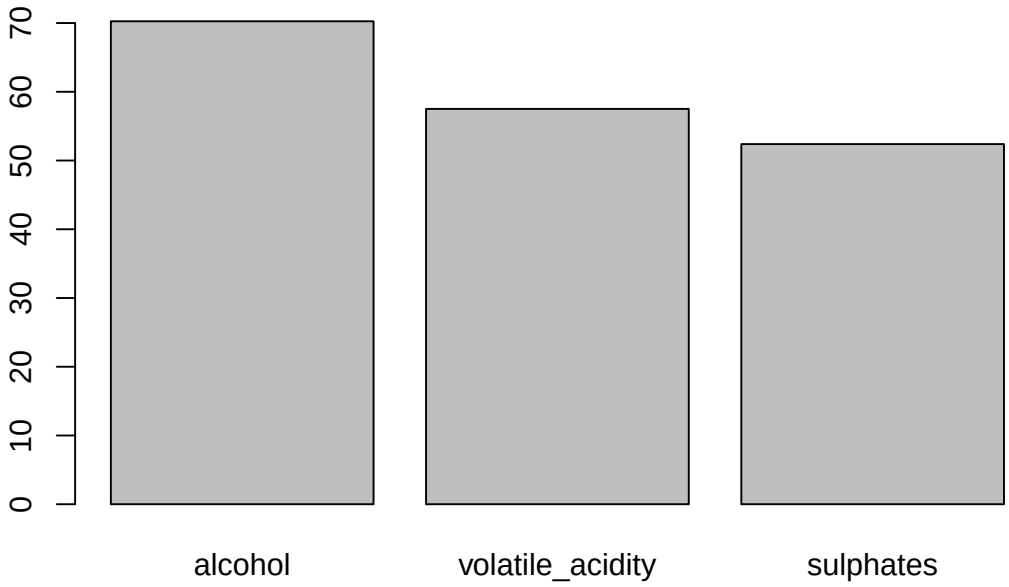
Construisons une première forêt aléatoire afin d'expliquer la variable qualité en fonction des autres variables explicatives.

```
##
## Call:
##  randomForest(formula = quality ~ ., data = dat, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 12.2%
## Confusion matrix:
##   1    2   3 class.error
## 1 0   62   1   1.0000000
## 2 1 1285  33   0.0257771
## 3 0   98 119   0.4516129
```

Nous remarquons une nouvelle fois grâce à la matrice de confusion que le modèle classe très mal les vins de qualité 1. Certes nous avons une précision d'environ 87%, ce qui est plus satisfaisant qu'avec la classification par arbre CART. De plus, le modèle est capable de classifier correctement plus de la moitié des bons vins.

Pour stabiliser l'erreur, nous avons réalisé une moyenne de l'erreur OOB sur 10 forêts. Nous obtenons alors 12% d'erreurs.

Observons les variables explicatives les plus importantes dans le prédiction de la qualité du vin.



Il semblerait que les variables les plus importantes sont l'alcool, l'acidité volatile et les sulfates. Cela confirme les résultats précédents.

3.3 Méthodes classiques de machine learning

Dans le cadre de notre cours intitulé Analyse Classification et Indexation des Données du semestre précédent, nous avons étudié quelques algorithmes de machine learning classiques. Nous essaierons par la suite d'en implémenter quelques-uns afin d'en estimer leur performance sur notre jeu de données. Le fonctionnement des algorithmes ne sera pas détaillé, mais les notes de cours destinées à l'explication de ces algorithmes sont dans la bibliographie.

Le perceptron cherche à séparer linéairement un problème en faisant une descente de gradient sur une fonction J avec $J(w) = \sum_{y \in Y_M(w)} -w^T y$ avec Y_M l'ensemble des points mal classés. Voici la proportion de vins bien classés en utilisant la méthode du perceptron : `0.7708333333333334`

Le SVM est une technique d'apprentissage machine qui cherche à séparer les différentes classes par des hyperplans dans un espace multidimensionnel. Nous obtenons la précision suivante : `0.8195286195286196`

La méthode des k plus proches voisins consiste à classifier un point comme étant de la catégorie majoritaire de ses k plus proches voisins. Nous obtenons la précision suivante : `0.8409090909090909`

Nous avons aussi extrait aléatoirement des vins du jeu de données de chaque catégorie afin d'en faire la prédiction et de la comparer à la classe originale :

```
class=3, expected=1, Predicted=[2]
class=3, expected=1, Predicted=[2]
class=3, expected=1, Predicted=[2]
class=3, expected=1, Predicted=[2]
class=3, expected=1, Predicted=[2]

class=4, expected=1, Predicted=[2]
class=4, expected=1, Predicted=[2]
class=4, expected=1, Predicted=[2]

class=5, expected=2, Predicted=[2]
class=5, expected=2, Predicted=[2]
class=5, expected=2, Predicted=[2]

class=6, expected=2, Predicted=[2]
class=6, expected=2, Predicted=[2]
class=6, expected=2, Predicted=[2]
class=6, expected=2, Predicted=[2]

class=7, expected=3, Predicted=[2]
class=7, expected=3, Predicted=[2]
class=7, expected=3, Predicted=[2]
class=7, expected=3, Predicted=[2]
class=7, expected=3, Predicted=[2]

class=8, expected=3, Predicted=[2]
class=8, expected=3, Predicted=[2]
class=8, expected=3, Predicted=[2]
class=8, expected=3, Predicted=[2]
```

Nous obtenons cette même sortie pour ces 3 algorithmes : tous les vins choisis sont classifiés comme étant de qualité moyenne.

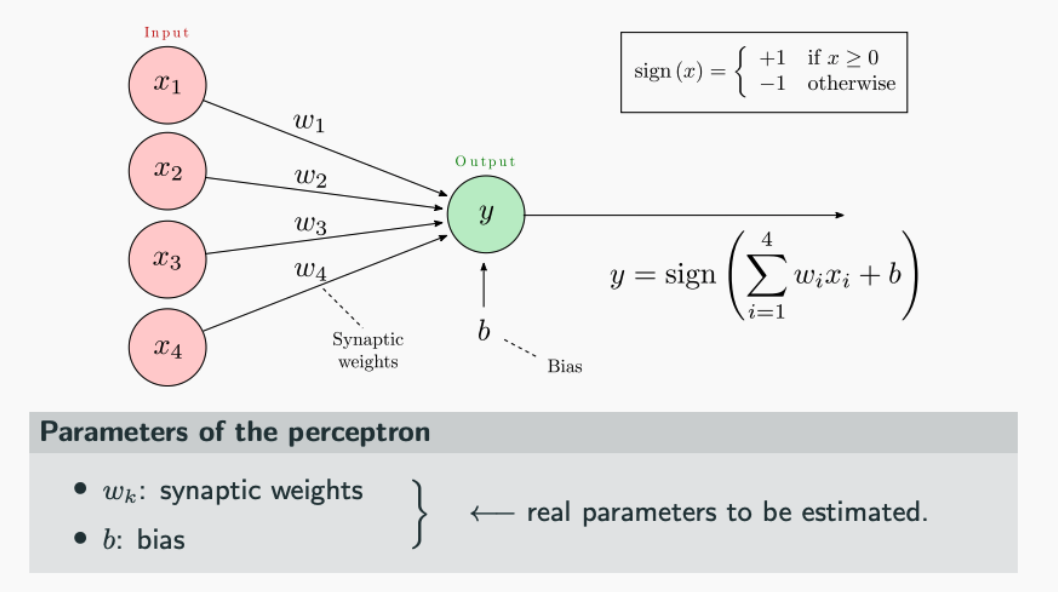
Avec une précision allant de 77% à 84%, l'ensemble de ces méthodes ne fournit pas un résultat assez satisfaisant. La nature de ces résultats peut s'expliquer par le fait que ces algorithmes sont peu efficaces dans ce cas, notamment à cause de la trop grande disparité dans la répartition des classes.

3.4 Réseau de neurones

Nous proposons de construire un classifieur basé sur un réseau de neurones profond. Puisque nous n'avons pas abordé ces notions dans un de nos cours, nous nous permettons d'insérer le texte explicatif associé qui fait partie de notre rapport de TER (réalisé par Wael Ben Hadj Yahia, Marie-Mathilde Garcia et Dorian Hervé).

Les réseaux de neurones artificiels sont inspirés du cerveau humain : dans notre cerveau nous avons des neurones, qui sont reliés entre eux et qui se transmettent des informations grâce à des impulsions électriques. En fonction de l'intensité de la pulsion électrique les neurones suivants s'activeront (ou pas si la pulsion électrique est trop faible). C'est selon ce même principe que les mathématiciens ont conçu les réseaux de neurones artificiels.

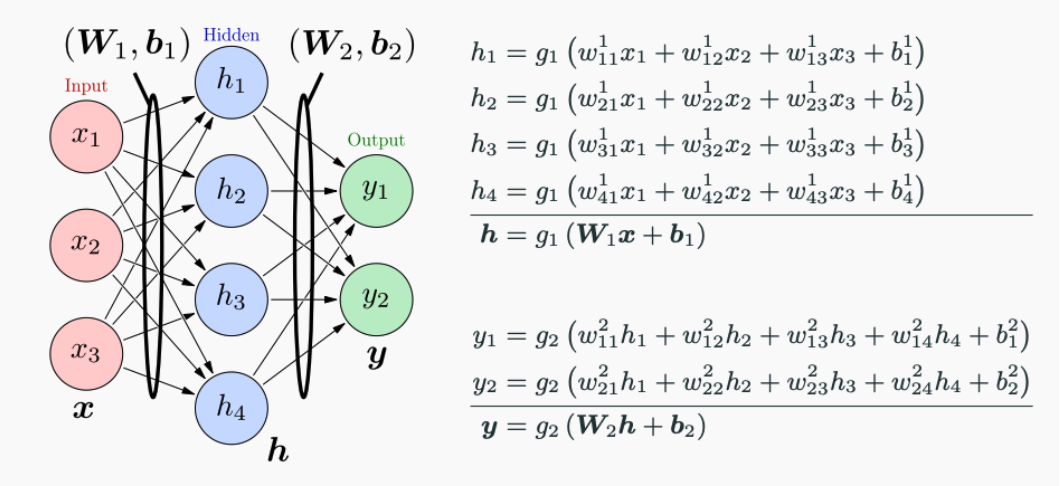
Warren McCulloch un neurologue américain et Walter Harry Pitts un mathématicien américain, sont les premiers à introduire les neurones artificiels en 1943. Un neurone artificiel est donc un modèle mathématique représentant un neurone biologique. En 1958, Frank Rosenblatt a introduit le perceptron, c'est le premier classifieur binaire utilisant l'apprentissage supervisé et les neurones artificiels. La figure ci-dessous nous donne un représentation de ce perceptron.



Il comporte plusieurs entrées, avec des poids sur chacune des arêtes. Le neurone effectue une sommation pondérée des entrées via les poids et déclenche un signal de sortie en fonction d'un certain seuil (le biais), comme nous pouvons le voir sur le schéma ci-dessus. Par exemple, si à la fin du perceptron $y = 1$, nous pouvons considérer que le signal est déclenché, alors que si $y = -1$ le signal n'est pas déclenché. L'entraînement du modèle consiste alors à ajuster les poids et le biais pour optimiser les résultats en sortie. Le problème de l'algorithme du perceptron est qu'il ne converge que pour des problèmes linéairement séparables.

Pour résoudre ce problème, les réseaux de neurones artificiels sont introduits. Ces réseaux sont aussi désignés par perceptron multicouche car ce type de réseau est le plus répandu.

Un réseau de neurones artificiels est un ensemble de neurones artificiels connectés. Ils vont être répartis en trois familles : les entrées (inputs), les sorties (outputs), et les "couches cachées" (hidden layers). La figure ci-dessous est un exemple d'un tel réseau.



Nous noterons, x le vecteur des entrées, h_1,...,h_n les vecteurs d'une couche cachée (car dans cet exemple nous n'en avons qu'une), y le vecteur des sorties et W la matrice des poids de chacune des connexions entre les neurones. Par rapport aux calculs de la figure ci-dessus nous avons aussi :

- w_{ij}^k qui représente le poids sur l'arête entre le noeud j et le noeud i à la k-ième couche cachée
- g_k la fonction d'activation appliquée aux x d'entrée.

Dans l'exemple de la figure, nous pouvons voir les calculs à chaque étape de ce réseau de neurones, le but est d'optimiser les valeurs des matrices W_1, W_2 et des vecteurs b_1 et b_2.

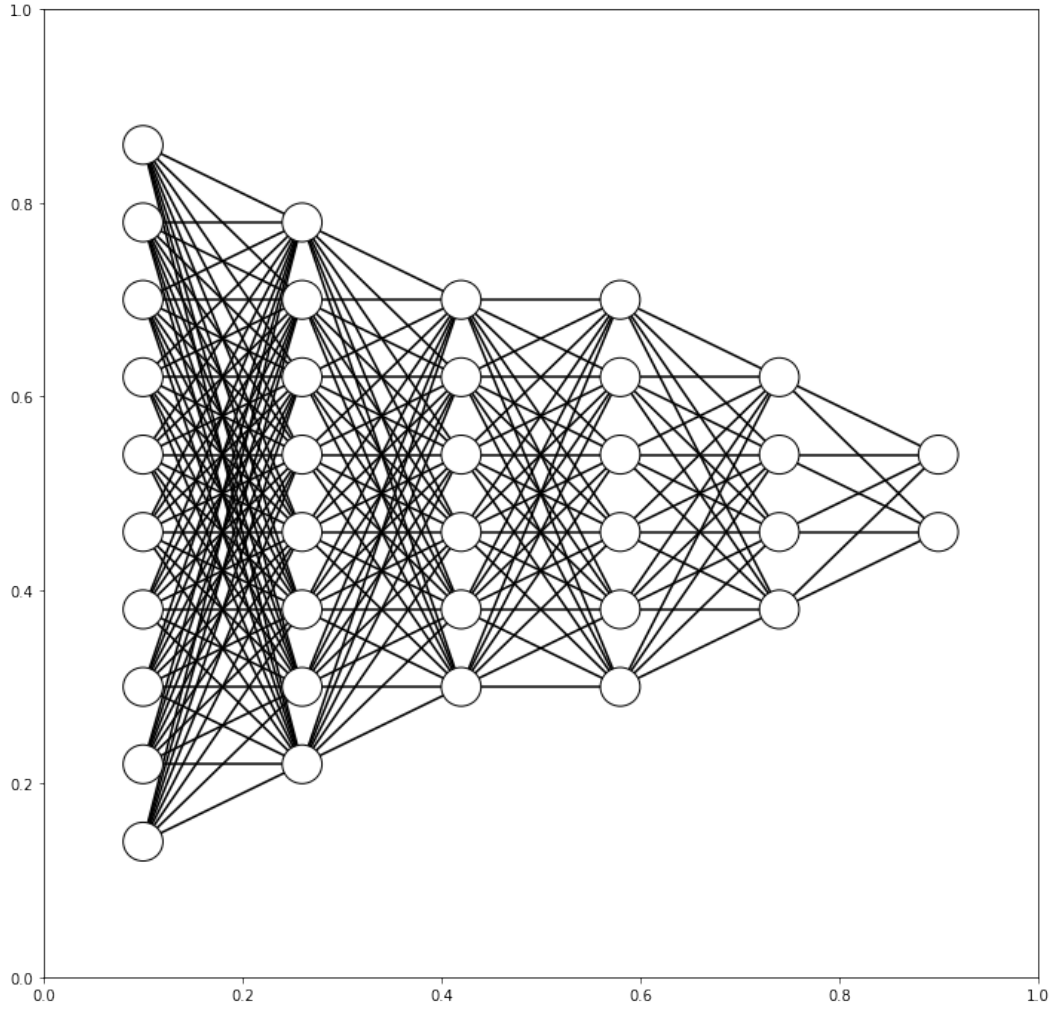
Dans cet exemple, nous avions une unique couche cachée mais dans ce que nous appelons les réseaux de neurones profonds, il peut y en avoir beaucoup plus.

L'une des applications des réseaux de neurones profonds est la classification d'images. Dans ce cas, en entrée sont donnés les pixels d'une image, et en sortie nous allons avoir 1 si l'image est un chien ou 0 si ce n'en est pas un (si par exemple nous cherchons à déterminer si l'image donnée en entrée est un chien).

Il existe plusieurs fonctions d'activation, ce sont ces fonctions qui conditionnent les sorties de notre réseau. Nous verrons plus tard laquelle nous déciderons d'utiliser pour notre cas.

Dans notre cas de classification des vins, nous avons construit un réseau de neurones assez simple en se basant sur un modèle et nous l'avons adapté afin qu'il puisse construire un classifieur avec nos données et nos nouvelles classes de vin.

Voici un schéma représentant la structure du réseau de neurones que nous avons construit :



Ce réseau est donc constitué de 6 couches. C'est un réseau "fully-connected" (autrement dit tous les neurones d'une couche sont reliés à tous les neurones des couches adjacentes). Notre fonction d'activation est ReLu définie de cette manière :

$$f(x) = \begin{cases} x & \text{si} \quad x > 0 \\ 0 & \text{sinon} \end{cases}$$

sauf pour la dernière couche avec un softmax qui sort un vecteur $\sigma(z)$ composé de K nombres réels strictement positifs tels que $\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e_k^z}$ pour tout $j \in \{1, ..., K\}$.

Nous obtenons des résultats qui s'approchent des 92% de précision en exécutant sur 2500 epochs.

```
Epoch 2500/2500
543/543 [==============================] - 0s 206us/step - loss: 0.2588 - accuracy: 0.9134
<keras.callbacks.callbacks.History at 0x7f86881bad68>
```

Notons que si nous utilisions le critère de la classe majoritaire, nous obtiendrons $1319/1599 = 0.825$ donc environ 82% de précision. Notre réseau de neurones classe donc de manière plus précise les classes non majoritaires.

Voici la prédiction des mêmes vins considérés que précédemment.

```
class=3, expected=1, Predicted=[1]
class=3, expected=1, Predicted=[1]
class=3, expected=1, Predicted=[2]
class=3, expected=1, Predicted=[1]
class=3, expected=1, Predicted=[1]

class=4, expected=1, Predicted=[2]
class=4, expected=1, Predicted=[1]
class=4, expected=1, Predicted=[1]

class=5, expected=2, Predicted=[2]
class=5, expected=2, Predicted=[2]
class=5, expected=2, Predicted=[2]

class=6, expected=2, Predicted=[2]
class=6, expected=2, Predicted=[2]
class=6, expected=2, Predicted=[2]
class=6, expected=2, Predicted=[2]

class=7, expected=3, Predicted=[3]
class=7, expected=3, Predicted=[3]
class=7, expected=3, Predicted=[3]
class=7, expected=3, Predicted=[2]
class=7, expected=3, Predicted=[3]

class=8, expected=3, Predicted=[3]
class=8, expected=3, Predicted=[3]
class=8, expected=3, Predicted=[3]
class=8, expected=3, Predicted=[3]
```

Nous constatons que le modèle est capable de classer assez efficacement les vins de qualité extrême alors que nous pensions que le classe 1 allait être moins bien prédite du au faible effectif, de la même manière que les classifieurs précédents. Le modèle semble moins bien prédire la classe d'un vin légèrement au-dessus ou au-dessous de la classe moyenne et semble ne pas se tromper pour les vins de cette classe.

En conclusion, le meilleur classifieur pour ce jeu de données est le réseau de neurones qui a une précision de 92%. Étant la grande disparité dans la répartition des classes, les autres algorithmes de machine learning semblaient tendre vers un classifieur de classe majoritaire, voire moins efficaces car ce dernier aurait une précision de 83% environ. Par exemple, le perceptron était à 77% d'efficacité (du au fait que le problème ne soit pas linéairement séparable ici).

4 Conclusion

En conclusion, l'analyse complète de la base de données indiquerait qu'un bon vin serait un vin avec une forte teneur en alcool et en sulfates, ainsi qu'une faible teneur en acidité volatile et en dioxyde de soufre. En ce qui concerne le classifieur le plus adapté à ce jeu de données, nous trouvons que le réseau de neurones est le classifieur optimal avec une précision de 92%. En axe de recherche complémentaire, nous proposons d'enrichir la base de données avec des vins de toutes qualités afin d'obtenir des résultats plus affinés.

5 Bibliographie

- CART: http://www.sthda.com/english/articles/35-statistical-machine-learning-essentials/141-cart-model-decision-tree-e
#pruning-the-tree
- Machine learning : https://www.kaggle.com/vanshjatana/applied-machine-learning
- Deep Learning : https://github.com/jg-fisher/wineNeuralNetwork
- Cours ACID : https://www.labri.fr/perso/domenger/Cours/ACID-Web.pdf
- Cours Deep Learning : https://www.charles-deledalle.fr/pages/files/ucsd_kpuw/3_deep.pdf
- Modèle de régression et tests d'hyporthèse : Pierre-André Cornillon et Eric Matzner-Løber, Régression avec R, Springer 2010
- Le logiciel R - 2e édition - Pierre Lafaye de Micheaux, Rémy Drouilhet, et Benoit Liquet