TP2 : Distribution de deux variables couplées

Nous nous intéressons dans cette séance de travaux pratiques aux outils introduits en cours pour discuter d'un jeu de données associé à deux variables couplées. Comme à la séance précédente, nous utiliserons les données relatives aux vins blancs issues de l'article *Modeling wine preferences by data mining from physicochemical properties* de Cortez *et al.* paru dans *Decision Support Systems* en 2009.

```
blanc <- read.table(
  "http://www.math.univ-toulouse.fr/~xgendre/ens/data/WineWhite",
  header=TRUE,sep=";")
```

1 Covariance et corrélation

En vous inspirant de la fonction `variance` que vous avez écrite à la section 5 du TP précédent, écrivez une fonction `covariance` qui calcule et retourne la covariance de deux vecteurs `x` et `y` de même taille passés en paramètres. Dans un premier temps, les poids seront supposés uniformes.

- Adaptez votre fonction pour avoir le prototype suivant,

```
covariance <- function(x, y, normalise=FALSE)
{
  ...
}
```

  Si le paramètre `normalise` vaut `TRUE`, la fonction devra calculer la covariance entre les versions centrées réduites de `x` et `y`.

- Que font les commandes `cov` et `cor` ? Exécutez les commandes suivantes et commentez les résultats obtenus. En particulier, expliquez les différences lorsque vous en observez.

```
x <- blanc$residual.sugar
y <- blanc$alcohol

covariance(x, y)
cov(x, y)

covariance(x, y, normalise=TRUE)
cor(x, y)
```

- Ajoutez un paramètre `poids` à votre fonction pour permettre à l'utilisateur de choisir la pondération des observations. Par défaut, ces poids seront les poids uniformes. Est-ce que les fonctions `cov` et `cor` de R vous permettent de faire la même chose ? Quelles fonctions de R faudrait-il utiliser ?

2 Régression linéaire

Nous considérons ici les variables `residual.sugar` (la teneur en sucre) et `density` (la densité),

```
x <- blanc$residual.sugar
y <- blanc$density
plot(x, y)
```

A la vue de ce nuage de points, pensez-vous qu'une relation affine entre `x` et `y` soit envisageable ? Calculez le coefficient de corrélation linéaire de Pearson entre `x` et `y` et commentez.

Dans le cours, nous avons vu que l'équation de la droite de régression de y sur x est donnée par

$$y = \frac{\mathrm{Cov}(x,y)}{\mathrm{Var}(x)} \times x + \left(\bar{y} - \frac{\mathrm{Cov}(x,y)}{\mathrm{Var}(x)} \times \bar{x} \right) .$$

- Calculer la valeur des coefficients pour les données `x` et `y`.
- Grâce à l'aide sur la fonction `abline`, expliquez le rôle des paramètres `a` et `b` de cette fonction.
- Affichez la droite de régression de `y` sur `x` par dessus le nuage de points.
- Que vous donne les commandes suivantes ?

```
lm(y~x)
abline(lm(y~x), col="red")
```

- Calculez l'équation de la droite de régression de `x` sur `y` et affichez cette droite sur le graphique précédent (attention, vous devez faire un changement de repère). Commentez le résultat obtenu.

3 Distance du χ^2 à l'indépendance

Nous proposons maintenant d'étudier le lien entre le degré d'alcool du vin (variable `alcohol`) et la qualité ressentie (variable `quality`). La qualité est une variable qualitative correspondant à une note entière entre 0 et 10. Par contre, le degré d'alcool est une mesure physique et c'est donc une variable quantitative. Afin d'utiliser la distance du χ^2 à l'indépendance, il nous faut transformer la variable `alcohol` en une variable qualitative.

Une méthode simple pour faire cette transformation est de découper le jeu de données en catégories. Nous considérerons donc trois catégories de vins : les vins peu alcoolisés (moins de 10°), les vins moyennement alcoolisés (entre 10° et 12°) et les vins fortement alcoolisés (plus de 12°). Nous utiliserons donc la fonction suivante :

```
alc_categorie <- function(t) {
  if (t <= 10) {
    return("PEU")
  } else if (t <= 12) {
    return("MOY")
  } else {
    return("FORT")
  }
}
```

- Expliquez pourquoi la fonction `alc_categorie` retourne bien le bon résultat.
- Avec l'aide sur la fonction `sapply`, dites ce que font les commandes suivantes :

```
x <- blanc$quality
y <- sapply(blanc$alcohol, alc_categorie)
```

La table de contingence s'obtient alors très facilement avec la fonction `table`,

```
table(quality = x, alcohol = y)
```

Cet objet se manipule comme une matrice en utilisant le noms des lignes et des colonnes comme coordonnées. Attention, il s'agit de chaînes de caractères et non pas d'entiers.

```
t_obs <- table(quality = x, alcohol = y)

# Notez la différence
t_obs["3","FORT"]
t_obs[3,"FORT"]

# Les marges sont de simples sommes
sum(t_obs["5",])
sum(t_obs[,"PEU"])
```

Le calcul de la table de contingence des effectifs théoriques se fait comme nous l'avons vu en cours,

```
# Effectif total
n <- length(x)

# Récupération des modalités de x et de y
x_lev <- levels(factor(x))
y_lev <- levels(factor(y))

# Table de contingence théorique
t_th <- matrix(0, nrow=dim(t_obs)[1], ncol=dim(t_obs)[2],
               dimnames=list(x_lev, y_lev))
for (i in x_lev) {
  for (j in y_lev) {
    t_th[i,j] <- sum(t_obs[i,]) * sum(t_obs[,j]) / n
  }
}

# Affichage
t_th
```

- Prenez le temps de bien comprendre toutes les étapes et commandes utilisées.
- A partir de `t_obs` et de `t_th`, calculez la distance du χ^2 à l'indépendance.
- Quelle est la valeur maximale du χ^2 ? Comparez votre résultat à cette valeur et concluez.

- De la même façon, étudiez le lien entre la qualité ressentie du vin et la teneur en sucre (variable `residual.sugar`). Vous pourrez considérer quatre catégories :
  - `PEU` pour une teneur inférieure à 2 $g.l^{-1}$,
  - `MOY_PEU` pour une teneur entre 2 $g.l^{-1}$ et 5 $g.l^{-1}$,
  - `MOY_BCP` pour une teneur entre 5 $g.l^{-1}$ et 10 $g.l^{-1}$,
  - `BCP` pour une teneur supérieure à 10 $g.l^{-1}$.