Travaux Dirigés de Statistiques – Polytech Grenoble – Année 3

Nicolas Glade, 2018-2019

Bonne pratique des TDs.

- Il s'agit de TD/TP, c'est à dire que vous allez pratiquer sur machine, mais l'enseignant sera là pour faire les rappels de cours nécessaires à la bonne interprétation des travaux pratiques réalisés. L'objectif est qu'au terme de cette année de statistiques, vous soyez opérationnels sous R en statistiques univariées.

- Les techniques et savoirs que vous allez accumuler sont, certes passionnants, fort utiles, mais aussi parfois complexes. Il est donc essentiel que vous posiez des questions. Aucune n'est stupide. Toutes permettront de progressivement mieux appréhender ce qui est l'objet des statistiques : comprendre la variance, la contrôler, et décider en fonction.

- Pour l'ensemble des TDs, les documents et fichiers seront disponibles à l'URL suivante (faites Google « nicolas glade » pour chercher la page) :
https://sites.google.com/site/nicolasglade/teaching

- Les données que vous avez saisies sont également disponibles là : http://k6.re/ZxNiX

- Vous enregistrerez, pour les TD, les fichiers sur le bureau ou dans l'espace de travail qui vous convient le mieux. Pensez à enregistrer vos documents, idéalement sur une clef USB, pour y avoir accès rapidement à chaque TD. Durant les travaux sous R, il arrive que les PC ou l'application seule subisse un crash. Pour éviter de perdre données et traitements, pensez à régulièrement enregistrer, en cours de TD, votre production.

- Copiez le code valide que vous écrivez dans un fichier texte (fenêtre de script, bloc note ...) que vous enregistrerez systématiquement en cours et en fin de session.

Note préalable au TD.

Les données traitées seront les données personnelles anonymes que vous avez saisis, concernant votre biométrie, vos études ...
La plupart des variables sont explicites, néanmoins voici quelques détails ainsi que des indications sur leur format :

- La dernière année d'études est calculée, puisque lors de la saisie, les étudiants n'avaient pas fini leur études.
- Les pointures sont en nombre entier.
- Les tailles et longueur des cheveux sont données en cm.
- Les modalités pour la couleur des yeux sont : Bleu, Vert, Marron, Noir.
- Celles des cheveux sont : Blond, Chatain, Noir, Roux.
- Les longueurs des doigts (index, majeur, annulaire) sont mesurée à partir de l'insertion du doigt sur la main, au sommet de la première articulation (bosse) et sont données en cm.
- L'écartement orbital (en cm) se mesure du centre d'une pupille à l'autre.
- Le type de pied : Carre, Egyptien ou Grec. Wiki : *On distingue les pieds selon la forme de leur extrémité distale : le type égyptien où l'extrémité du gros orteil est la plus distale et les autres orteils sont de taille décroissante ; le pied grec où cette fois c'est l'extrémité du deuxième orteil qui est la plus distale, avec un pied en forme de triangle ; le type carré ou romain où les extrémités des trois premiers orteils sont équivalentes.*
- Le nombre d'heures de jeux vidéos mensuelles
- L'animal de compagnie préféré (Chien, Chat, les NAC (reptiles, tortues comprises, araignées …), Autres (rats, souris, chèvre, cheval …))

- Le nombre d’animaux de compagnie
- La note de maths au S5
- Votre niveau de Pilosité (Faible, Moyenne, Forte)
- Le nombre d’heures de sport mensuelles
- Le système d’exploitation de votre téléphone mobile (IOS, Android, Autre)
- La taille de votre fratrie
- Le nombre de repas avec au moins une part de pizza, dans le mois
- Est ce que vous êtes Chocolatine ou Pain au chocolat
- Votre latéralisation (G/D)

Partie 1. Analyse de normalité, Comparaison de données quantitatives, Échantillonnage

L'objectif de la première séance est d'abord de commencer à prendre en main le logiciel R durant des exercices de statistiques descriptives. Il s'agira de décrire correctement les données proposées et d'en faire une première analyse en calculant certains de leurs paramètres et en représentant graphiquement ces données. On choisira des données pour lesquelles on souhaite faire des tests de comparaison. Nous nous attarderons sur leurs conditions de validité. Le principe de fonctionnement des tests de comparaison, paramétriques et non paramétriques, l'importance de la normalité, de l'effectif ... seront revus. Enfin, nous étudierons l'effet de l'échantillonnage.

Nous allons étudier une population d'étudiants. A titre d'exemple, nous chercherons dans le cadre de ce TD à quantifier l'effet du facteur sexe sur la taille des individus.

1.1. Chargement des données

Commençons par nous assurer que nous travaillons bien dans le bon dossier. Pour cela, il y a 2 solutions :

- A l'aide de la commande ***getwd()*** vous saurez quel est le dossier actif (*a priori* votre home, « *c:/Documents\ and\ settings/monlogin/* » par défaut). Avec la commande ***setwd("nouveauchemin")*** vous pouvez changer de dossier actif. Depuis votre home, faites ***setwd("./Bureau/")***.
- Ou bien, plus simple, si vous travaillez sous *R Studio*, vous pouvez changer le répertoire de travail depuis le menu fichier.

Chargez le fichier de données « *Donnees_Etudiants.tsv* » à l'aide de la commande
d.all <- ***read.csv****(" filename ", header=T, sep='\t').*

1.2. Préparation des données

Explorons rapidement les données. Pour voir le début du tableau, faites ***head(****d.all****)***, et pour la fin ***tail(****d.all****)***. Sélectionnez les colonnes *Sexe* et *Taille_en_cm* que vous copierez dans un nouveau tableau *d*.

Note 1. si vous ne vous intéressez qu'à certaines variables (par exemple les variables concernant le sexe et la taille des individus), il est recommandé de sélectionner les colonnes correspondantes dans le tableau contenant toutes les données et de les copier dans un nouveau tableau. Par exemple :

```
d <- d.all[ , c(4,6) ]        #   la variable Sexe est la 4ème colonne ; la variable Taille_en_cm la 6ème
```

Note 2. certaines valeurs sont notées **NA** (pour **N**ot **A**vailable) ; ce sont les valeurs manquantes. Pour les enlever, il y a 2 manières :

```
d <- d[ rowSums(is.na(d[,])) , ] # à la main
d <- na.omit(d)
```

Notez qu'il est recommandé de n'enlever les valeurs NA qu'après avoir sélectionné les variables d'intérêt.

Exercices.

Pour voir les données sur les hommes, ajoutons une condition sur la variable sexe : *d[d$Sexe=='H',]*ce qui signifie « je veux les données du tableau d, en sélectionnant les lignes telles que la variable Sexe vaut 'H' (hommes) et sans mettre de conditions sur les colonnes (on a effectivement **Tableau[Lignes , Colonnes]**).

Cherchez les femmes faisant plus de 1m70 !

Comptez *d.N* le nombre total de personnes à partir du fichier, *d.N.H* le nombre d'hommes, *d.N.F* le nombre de femmes. Pour déterminer d.N, vous devrez déterminer la 2eme dimension du tableau d à l'aide de la commande ***dim(****d****)[****2****]***. Pour déterminer d.N.H et d.N.F, vous devez faire la même opération mais en ajoutant une condition sur la variable sexe, telle que d$Sexe=='H' ou 'F' comme suit : ***dim(****d[d$Sexe=='H',]****)***

[2], ou bien en utilisant la commande ***length*** sur une colonne du tableau : ***length(****d[d$Sexe=='H',1]****)***.

En utilisant la commande ***unique(****x****)***, déterminez les différentes valeurs que peut prendre la variable *Sexe*. De quel type est-elle ? Pour les variables ayant un faible nombre de modalités (ex: le sexe), vous pouvez aussi utiliser la commande ***level(****x****)***. Cette commande présente l'avantage d'autoriser des changements de modalités de réponse. Par exemple, si l'on souhaite remplacer les modalités *'H'* et *'F'* de la variable *Sexe* par *'h'* et *'f'*, il suffit de faire ***level(****d$Sexe****)*** <- ***c(****'f','h'****)***

Vous pouvez, si vous souhaitez travailler plus aisément sur des groupes différents, séparer les données des hommes de celles de femmes, soit en utilisant le fonction ***split(****x****)***, mais je ne le conseille pas dans l'immédiat, soit (de façon plus simple) en isolant les donnés des hommes dans un tableau *d.H* et celles des femmes dans un autre *d.F*. De même pour des données spécifiques comme les tailles des hommes et celles des femmes que vous pouvez stocker si vous le souhaitez dans *d.T.H* et *d.T.F*.

1.3. Exploration des données et description.

Dans cette partie, vous explorerez les données et décrirez graphiquement les populations que vous étudierez (histogrammes, boxplots, qqplots ...).

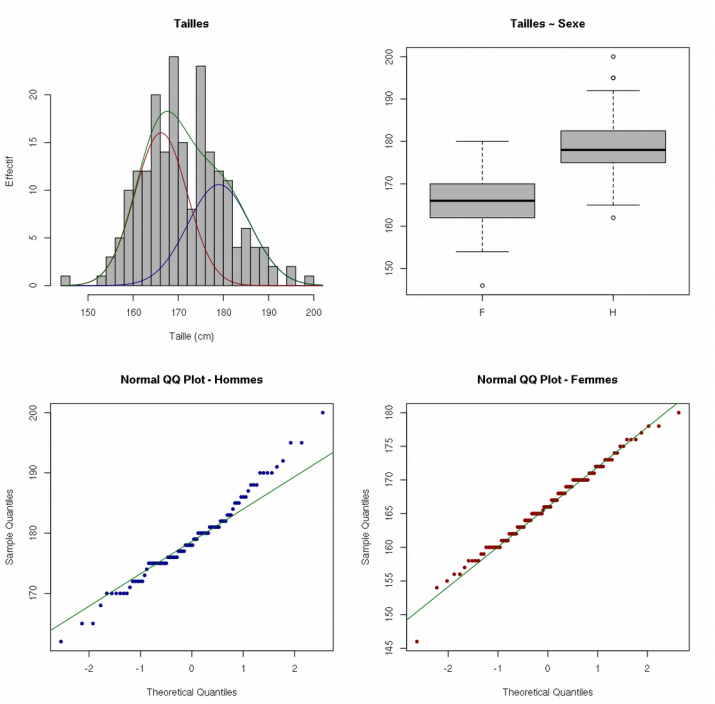
1.3.1. Paramètres

Déterminez pour la variable *Taille* et chaque catégorie (homme ou femme) la moyenne ***mean(****x****)***, la médiane ***median(****x****)*** et la variance ***var(****x****)*** ou l'écart type ***sd(****x****)*** des distributions. Vous pouvez aussi utiliser la commande ***summary(****x****)*** sur le tableau de données entier.

Que semblent indiquer ces valeurs concernant les tailles des hommes et des femmes ?

1.3.2. Graphiques

Réalisons maintenant une fonction (que nous nommerons *plot_tailles*) qui va nous permettre d'afficher plusieurs graphiques utiles en fonction d'un jeu de données fourni en paramètre. L'appel de la fonction *plot_tailles(d)* devrait afficher ceci :



Les étapes pour construire ce graphique vont être décrites pas à pas en TD. Néanmoins, voici les commandes utilisées :

Pour créer une fonction :
*mafonction <- **function(**paramètres**){***
*   # code de la fonction*
*   **}***
Pour mettre plusieurs graphiques sur la même page graphique on utilise
par(mfrow=c(**nblignes**,** nbcolonnes**))

Pour réaliser un histogramme avec une courbe gaussienne théorique :
T.by <- 2 # largeur des batons de l'histogramme
*T.breaks <- **seq(from**=min(d)-T.by**, to**=max(d)+T.by**, by**=T.by**)** # séquence de cassures*
hist(**d**, breaks**=T.breaks**, col**='grey70'**, xlab**="Taille (cm)"**, ylab**="Effectif"**, main**="Tailles"**)
*T.x <- **seq(from**=min(d)-T.by**, to**=max(d)+T.by**, length**=1000**)** # x de la courbe théorique*
*T.y <- **dnorm(x**=T.x**, mean**=mean(d)**, sd**=sd(d)**) * length(**d**) * T.by # y de la courbe théorique*
***lines(**T.x**,** T.y**, col**='darkblue'**)** # courbe théorique*

Faites en sorte que la courbe théorique des femmes, celle des hommes et la courbe totale soient affichées.

Précisons notre analyse en traçant la boite à moustache de la variable quantitative en fonction d'une variable qualitative comme suit :
boxplot(**d$MaVariableQuantitative ~ d$MaVariableQualitative**, main**="titre"**, ylab**="titre des y"**)

Faites aussi un quantile-quantile plot pour chaque sexe à l'aide de la commande :
qqnorm(**MaVariableQuantitative**, pch**=20, ...**)
qqline(**MaVariableQuantitative**, col**='red', ...**)
Comment fonctionne un quantile-quantile plot ? Comment l'analyse t-on ?

De quels renseignements supplémentaires dispose t-on ? Quelles études va t-on pouvoir ainsi réaliser. Comment ? Quels seront les critères importants pour pouvoir réaliser cette étude ?
On se posera notamment la question de l'effectif (combien y a t-il de mesures pour chaque variable?) et de la distribution des valeurs pour chacune des variables. En quoi ces questions sont elles importantes ?

1.4. Normalité

A quoi sert d'analyser la normalité d'une distribution ?

Faites une analyse quantitative de normalité pour chacun des groupes des données. Réalisez pour cela un test de normalité. Ces tests peuvent être réalisés à l'aide des commandes ***shapiro.test(**x**)*** pour le test de Wilk-Shapiro.

1.5. Echantillonnage - Bootstrap

Recommencez ce travail (graphique et quantitatif) d'analyse de normalité sur des sous-populations d'effectif ***n*** (***n<N*** l'effectif de la population complète) obtenues par tirage aléatoire dans ces populations en utilisant la commande ***sample(**x**, n, replace**=F**)***. Vous testerez des sous-effectifs de 50% et 25% de la population complète (et éventuellement d'autres proportions). Répétez les opérations plusieurs fois.

Vous pouvez aussi procéder de la même manière sur des échantillons de même taille en faisant un tirage aléatoire avec remise en utilisant la commande ***sample(**x,N,replace=T**)***. Cette technique se nomme *Bootstrap*.

Remarque importante : pour échantillonner dans un tableau *T* à *p* colonnes et d'effectif *n*, on ne peut pas

utiliser la commande *sample* directement sur le tableau. Pour faire cela, il faut tirer aléatoirement une séquence d'indices, puis utiliser ces indices pour sélectionner les lignes dans le tableau, comme suit :

```
d.indices <- sample(1:n, m, replace=F) # tirage de m valeurs sans remise
d.s <- d[d.indices, ]

# ou

d.indices <- sample(1:n, replace=T) # avec remise
d.s <- d[d.indices, ]
```

Utilisez votre fonction de tracé des tailles sur l'échantillon. Faites les plusieurs fois. Qu'observez vous ?

[Facultatif mais utile] Pour répéter les opérations plusieurs fois (pour les tests en particulier), une possibilité est de faire une boucle et de stocker les p-valeurs des tests dans un vecteur comme suit :

par exemple :

```
p <- vector("numeric")
n.s <- 30
n.rep <- 100
for (i in 1:n.rep){
    t <- shapiro.test( sample(d[,6], n.s) )
    p[i] <- t$p.value
    }
```

ou pour des graphiques :

```
n.s <- 30
n.rep <- 100
plot(x=NULL, y=NULL, xlim=c(0,n.s), ylim=c(-30,30))
for (i in 1:n.rep){
    s <- sample(d[,6], n.s)
    points( 1:n.s,  s-mean(s) , pch=20, cex=0.1)
    }
```

Expliquez à quoi sert cette analyse de normalité sur de tels échantillons. En quoi cela change-t-il selon l'effectif ***n*** du sous-échantillon ? Comment la normalité se comporte-t-elle lorsqu'on ré-échantillonne avec remise (même effectif N)?

1.6. Comparaison

Les sous-populations hommes et femmes sont-elles appariées ? Pourquoi ? Qu'est ce que ça change ?
Quelles comparaisons peut-on faire ? A quoi sert le test d'homoscedasticité ?

1.6.1. Homoscedasticité : tests sur les variances

Testez l'homoscédasticité des 2 groupes de données que vous avez choisis à l'aide de la commande ***var.test(****pop1,pop2****)***. Faites de même sur des populations réduites (en utilisant la commande ***sample(****x,n****)***.

Qu'en déduisez-vous quant au type de test de comparaisons de moyenne que vous allez pouvoir réaliser ?

1.6.2. Comparaison de moyennes

Comparez les moyennes des 2 groupes de données que vous étudiez. Attention, vous devrez choisir le test adapté et préciser, si le test est paramétrique (***t.test(****x,y****)***, ***var.equal=T*** en argument de cette fonction si l'homoscedasticité est vérifiée. Vous devez aussi faire attention à l'appariement de vos populations. Si vous devez la prendre en compte dans un test paramétrique, ajoutez l'argument ***paired=T*** dans la fonction de test.

Si le test à réaliser est non paramétrique (non respect de la normalité), utilisez la commande ***wilcox.test(****x,y****)***.

Dans les deux cas, vous pouvez aussi faire des tests orientés en précisant l'hypothèse alternative avec ***alternative=''greater''*** ou ***alternative=''less''***.

1.6.3. Robustesse

Recommencez cette étude en éprouvant la robustesse de ces tests. Pour cela:

- créez un vecteur vide de p- valeurs : **pv <- vector(**"numeric"**)**
- Dans une boucle ***for(****i in 1:1000****){ ....}*** récupérez la p-valeur d'un test fait sur des échantillons ré-échantillonnés avec remise. La p-valeur des tests peut être récupérée en stockant le résultat d'un test dans une variable (ex: *monTest<-t.test(x1,x2)*) puis en accédant à sa variable interne ***p.value*** du test et en la stockant dans le vecteur de p valeurs comme suit : ***pv[i]<-monTest$p.value***

Analysez le vecteur de p-valeurs.

1.6.4. Conclusion

Concluez sur la comparaison que vous avez faite. Que pensez-vous du test statistique en regard d'une simple comparaison graphique des populations ? Quels risques prend-on en effectuant ces comparaisons ? Comment améliorer la puissance du test ? ...

Partie 2. Comparaisons de données non normales, comparaisons de données qualitatives

2.1. Chargement des données - préparation des données

Après vous être assurés de travailler dans le bon répertoire, chargez le fichier de données « *Donnees_Etudiants.tsv* » à l'aide de la commande *d <-* ***read.table****(" filename ", header=T).*

Ce fichier contient à la fois des données pour une comparaison non paramétrique (variable qualitative non normale) et des données pour un test d"indépendance. En ce qui nous concerne, pour ce TP, nous réaliserons d'une part (section 2.2) une comparaison de taille de cheveux des hommes et des femmes, d'autre part (section 2.3), nous nous poserons la question de l'indépendance des variables *Sexe* et *Longueur_cheveux_en_cm*.

Le tableau contient des données manquantes (non renseignées) notées ***NA***.
Pour préparer vos données, procédez comme suit :

- Commencer par sélectionner les colonnes qui vous serviront pour chacun des tests dans les sections 2.2 et 2.3 (en les stockant par exemple dans 2 tableaux d2 et d3.
- Eliminez dans chaque tableau les lignes contenant des ***NA***, comme suit :
  d <- d[***rowsums(is.na(****d****))****== 0 ,]*

2.2. Tests de rang : Mahn & Whitney, Wilcoxon

Dans le problème suivant, vous voulez vous assurer que, statistiquement, les hommes ont des cheveux plus courts que les femmes.

Mettez ce problème en hypothèses (hypothèse vraie H0 et hypothèse alternative H1).

Réalisez les mêmes étapes que pour la comparaison des tailles pour éprouver la normalité de vos données.

Quelle distribution de probabilité décrit bien la distribution de cheveux des femmes ? des hommes ? (cf cours *1 - introduction*).

Après avoir montré que les tests paramétriques ne peuvent pas être utilisés (rappelez pourquoi), réalisez le test de comparaison adapté aux hypothèses que vous aurez posées en utilisant la commande ***wilcox.test(****x1,x2****)*** où x1 et x2 constituent les 2 échantillons à comparer.

Réalisez tout de même un test paramétrique de comparaison de moyennes et expliquez pourquoi il aboutit à un résultat comparable.

2.3. Test d'indépendance (du Chi2)

Pour apprendre à faire des tests du Chi2, nous allons tester l'indépendance des variables *Sexe* et *Longueur_cheveux_en_cm*.

De quel type sont ces variables ? Pourquoi va-t-il falloir refactoriser si l'on souhaite réaliser un test du Chi2 sur de telles variables ?

Quels autres tests du Chi2 aurait-on pu faire sur de telles variables ?!

Nous allons maintenant voir comment refactoriser une variable quantitative sous la forme d'une variable catégorielle. pour cela, nous allons utiliser la fonction ***cut(****d3$Longueur... ,* ***breaks****=... ,* ***labels****=...)*.
Exécutez cette fonction sur la variable *Longueur_cheveux_en_cm* pour la réduire à quelques 2 catégories : *courts* et *longs*. La moyenne de la longueur des cheveux déterminera le seuil séparant ces 2 catérories.

Nous allons maintenant ajouter la variable catégorielle sous la forme d'une nouvelle colonne dans le tableau

d3. Pour cela, utilisez la fonction ***cbind*** (column bind) comme suit :
d3 <- ***cbind(****d3****,*** *LC_cat2* = ***cut(****...,...,...****))*** où *LC_cat2* désigne le nom de la nouvelle colonne.

De même, ajoutez une nouvelle variable catégorielle *LC_cat3* pour la longueur des cheveux, avec cette fois 3 catégories : *courts* de 0 à 10 cm, *moyens* de 10 à 20 cm, *longs* de 20 cm à "+ l'infini" (défini par *Inf*).

Affichons les tableaux de contingence formés des variables *Sexe* et *LC_cat** en utilisant la commande ***table(****variable1****,*** *variable2****)***. Pour le tableau formé des variables *Sexe* et *LC_cat2*, calculez le tableau des effectifs théoriques. Vous pouvez calculer les sommes en marges à l'aide de la commande ***margin.table(****tableau****,*** *marge****)***. Calculez la valeur de *Zeta* à la main.

Pour chaque paire de variables catégorielles, réalisez un test du Chi2 en utilisant la commande ***chisq.test(****variable1****,*** *variable2****)*** et interprétez.

Recommencez en sous-échantillonnant avec des échantillons de très petite taille (5 à 10 individus par classe).

Partie 3. Régression - Prédiction - Efficacité et Robustesse des modèles

3.1. Chargement et préparation des données

Nous allons travailler sur la prédiction de la longueur des cheveux catégorielle (*LC_cat2)* par la taille des individus. Pour cela, nous allons créer cette variable catégorielle *LC_cat2* en utilisant la commande *cut* décrite à l'étape 2.3. On supposera 2 catégories de tailles de cheveux : « courts » et « longs » définies par un seuil (arbitraire, calculé sur la moyenne, calculé sur la médiane …).

Sélectionnez les colonnes utiles de ce tableau (le nouveau tableau stockant ces colonnes s'appellera par exemple *d.m.dat*) ; éliminez également les lignes contenant des données manquantes.
Profitons en pour renommer les variables avec des noms plus courts comme suit :

names(*d.m.dat****)*** *<- c("Taille", "LCC")*

3.2. Régression logistique

Nous allons travailler sur la prédiction de la longueur des cheveux catégorielle (*LCC)* par la taille des individus. Pour cela, nous allons réaliser une régression logistique entre la variable qualitative (catégorielle) transformée en facteur à l'aide de la fonction ***as.factor(****variable_qualitative****)*** et la variable quantitative *Taille* comme suit :
d.m.mod <***- glm(as.factor(****LCC****)*** *~ Taille****, family****="binomial"****, data****=d.m.dat****)***
et observons les résultats de ce modèle à l'aide de la commande ***summary(...)***. Stockez ce résultat dans une variable (par exemple *d.m.sum*). Vous pouvez accéder aux variables internes calculées par la commande ***summary*** à l'aide du ***$***, par exemple *d.m.sum$deviance*.

Calculez le coefficient de détermination à partir de la variance résiduelle (*deviance*) et totale (*null.deviance*) ! Notre modèle est il bon *a priori* ?

3.3. Prédiction

3.3.1. Prédiction et confusion sur les données d'origine

La fonction ***predict(object, type, newdata)*** permet de calculer la réponse (valeur relative à la variable expliquée par le modèle) fournie par un modèle (*object*) à partir d'un ensemble de descripteurs du modèle (valeurs *newdata* des variables d'entrée du modèle). Dans notre cas, nous cherchons à obtenir comme réponse la probabilité que la variable catégorielle soit égale à une certaine modalité, par exemple *P(LCC='longs')*. Pour obtenir cela, on spécifie ***type='r'*** (response). Enfin, ***newdata*** est un tableau contenant l'ensemble des variables (expliquée et explicative(s)) du modèle. Si ce paramètre n'est pas spécifié, c'est le jeu de données d'origine qui est utilisé, sinon, de nouvelles valeurs peuvent être calculées.

Commençons par calculer les valeurs prédites à partir des données d'origine et comparons les avec les réponses attendues.

On réalise d'abord la prédiction qu'on stocke dans un nouveau tableau, comme suit :
d.m.pred.v1 <***- predict(object****=d.m.mod****, type****='r'****)***

En utilisant la commande ***cut(x,breaks,labels)***, transformez les valeurs de probabilité prédites (*d.m.pred.v1*) en valeurs catégorielles (*d.m.pred.c1*).

Comparez les valeurs prédites avec les valeurs d'origine en utilisant la fonction *tab*<***- table(****catégories_origine, catégories_prédites****)***. Cette table s'appelle la matrice de confusion. Utilisez une combinaison des commandes ***rep(...)*** et ***margin.table(****tab****,1)*** pour transformer cette table en pourcentage de bonnes et mauvaises prédictions et faites vous une idée de la qualité de prédiction de ce modèle.

3.3.2. Prédiction à partir de nouvelles données

Pour réaliser une prédiction sur de nouvelles données, il suffit de fournir un nouveau tableau contenant une colonne *Taille* contenant de nouvelles valeurs de taille à partir desquelles on souhaite réaliser la prédiction et une colonne *LCC* mise à 0.

Pour cela, on construit un « data.frame », c'est à dire un tableau de données comme suit :
d.m.newdat <- ***data.frame(****LCC=0****,*** *Taille=* ***c(****180,165,171****))***

Utilisez ces 3 valeurs pour prédire les tailles de cheveux de ces individus.

3.3.3. Modélisation à partir d'un sous-échantillon et prédiction et confusion sur les données d'origine

L'idée ici est de créer un modèle logistique à partir d'un sous-échantillon (disons d'1/4 des valeurs initiales) de la population initiale(utilisation de la commande *indices <-* ***sample(****1:n, n/4****)***), puis de se servir de ce modèle pour prédire les valeurs sur le tableau entier et de comparer les résultats à ceux d'origine.
Qu'observez vous ?

3.3.4. Modélisation et prédiction à partir d'un échantillon de même taille, tiré aléatoirement avec remise (bootstrap)

Cette fois, on réalise la même opération, mais on tire aléatoirement les indices d'un échantillon de même taille n avec remise comme suit : *d.bs.ind <-* ***sample(****1:n, n****, replace=T)***.
Recommencez de nombreuses fois. Qu'observez vous ?
Essayez de cumuler les matrices de confusion sur un grand nombre (ex: 1000) bootstraps à l'aide d'une boucle ***for(****i* ***in*** *1:1000****){*** *...* ***}*** . Qu'en déduit-on sur la qualité de ce modèle logistique ?