Bio2041 : Séance informatique 5

ANOVA à un critère de classification et test statistique de χ^2

Département de sciences biologiques
Université de Montréal

Marie-Hélène Ouellette, Guillaume Blanchet, Daniel Borcard,
Sébastien Durand et Guillaume Bourque
27/04/08 10:32

Objectifs : *Cette séance informatique à pour but de vous montrer comment effectuer une analyse de variance à un critère de classification (ANOVA) et un test de χ^2 dans le langage **R**.*

ANOVA à un critère de classification

Une ANOVA se calcule avec la fonction « `aov( )`». On fournit les variables dépendantes et explicatives (critère de classification) à la fonction sous la même forme que dans une régression linéaire simple. Voici un exemple utilisant des données prêtes à êtres utilisées.

1. Pour bien comprendre le fonctionnement de la fonction « `aov( )` », importez dans la console R le petit jeu de données appelé « **insuline.txt** ». Cette série de données présente le niveau d'insuline dans le pancréas de souris ayant grandi dans 5 environnements différents (des salles de 5 couleurs différentes). (Ces données sont modifiées de *An introduction to biostatistics* par Glover et Mitchell 2002).
2. Appelons l'objet dans lequel se trouve le tableau « `insuline` ».

```
Insuline <- read.table("insuline.txt", header=TRUE)
```

Dans ce tableau, il y a deux séries d'informations: la première colonne présente la couleur des différentes salles (*le critère de classification*) alors que la seconde présente les niveaux d'insuline dans le pancréas des différentes souris échantillonnées (*la variable dépendante*). Chaque ligne correspond à une souris (*l'élément*).

3. Avant de faire l'ANOVA, vous devez vous assurer que l'objet décrivant le critère de classification (*ici c'est la première colonne du fichier*) soit un facteur. Un facteur est, en soi, un vecteur définissant des classes. En général, si les noms des classes sont du texte (comme ici la couleur de la salle), R aura automatiquement transformé la colonne en facteur lors de l'importation des données à l'aide de read.table(). Cependant, si les noms des classes sont des chiffres (par exemple, si vous avez effectué votre expérience à trois niveaux de température, 15, 20 et 25 °C), R n'aura probablement **PAS** transformé la colonne en facteur : vous devez donc le faire vous-même. Utilisez la fonction « as.factor() » pour transformer une colone en facteur. En tout temps, vous pouvez vérifier si une colone ou un vecteur est un facteur avec la fonction « is.factor() ».

```
# Vérifie si la colonne de critères de classification est un facteur
is.factor(Insuline$Couleur.Salle)
# Transforme la colonne de critères de classification en facteur
Insuline[,1] <- as.factor(Insuline[,1])
```

ATTENTION : Si votre colonne de critères de classification n'est pas un facteur, la fonction « aov » vous donnera un résultat erroné, mais aucun message d'erreur.

4. Vous pouvez afficher les différents niveaux du facteur; ces niveaux sont en fait le nom des classes qu'utilise le facteur.

```
# Affiche les niveaux du facteur
levels(Insuline$Couleur.Salle)
```

5. L'ANOVA peut ensuite être calculée de la façon suivante :

```
# Stocke le résultat de l'ANOVA dans « resultats »
resultats <- aov(Concentration.Insuline ~ Couleur.Salle, data=Insuline)
```

   - N'oubliez pas, le « ~ » veut dire "en fonction de…". Ici on demande le modèle linéaire avec la colonne **y** (*variable dépendante*) **en fonction** de la colonne **x** (*variable explicative ou indépendante, ici le critère de classification*).

6. Pour avoir la totalité de l'information sur l'ANOVA, utilisez la fonction « `summary()` » de votre objet contenant le résultat de votre ANOVA.

```
# Affiche le sommaire de l'objet stocké dans « resultats »
summary( resultats )
```

La sortie de cette fonction est :

```
                 Df Sum Sq Mean Sq F value Pr(>F)
Couleur.Salle  4  49.84   12.46  1.4624 0.2508
Residuals     20 170.40    8.52
```

`Df :` Les degrés de liberté.
`Sum Sq :` Somme des carrés d'écarts à la moyenne (dispersion).
`Mean Sq :` Variance due au facteur « `Couleur.salle` » et variance des résidus « `Residuals` » (Sum sq/Df).
`F value :` La valeur de la statistique de Fisher (variance due au facteur/variance des résidus)
`Pr (>F) :` Probabilité qu'on puisse rencontrer la valeur calculée de la statistique de Fisher « `F value` » ou une valeur plus grande si H_0 est vraie.

Test statistique de χ^2

La fonction permettant de faire un test statistique de χ^2 est « `chisq.test( )` ». Cette fonction permet d'obtenir la valeur de la statistique calculée ainsi qu'un test de signification paramétrique ou par permutation.

1. Pour bien comprendre le fonctionnement de la fonction « `chisq.test( )` », importez dans la console R le petit jeu de données appelé « **Couleur des yeux.txt** ». Cette série de données présente le sexe et la couleur des yeux de 28 personnes.

2. Appelons « `donnees` » l'objet dans lequel se trouvera le tableau.

```
donnees <- read.table("Couleur des yeux.txt", header=TRUE)
```

   - La colonne « `Couleur.yeux` » est une variable qui a deux états : brun et bleu.
   - La colonne « `Sexe` » est une variable qui a deux états : Homme et Femme.
   - Chaque ligne du tableau représente une observation. Dans cet exemple, on a 14 observations au total. La première est un homme aux yeux bruns, la deuxième est une femme aux yeux bleus, etc.

3. Nous pouvons créer un tableau de contingence à l'aide de la fonction « `table( )` ».

```
Tableau.yeux <- table(donnees$Sexe, donnees$Couleur.yeux)
```

   Le tableau de contingence que l'on obtient se nomme « Tableau.yeux » :

```
        bleu brun
Femme   11    3
Homme    5    9
```

4. Sachez que l'on peut fournir le tableau de contingence « Tableau.yeux » OU les variables « donnees$Couleur.yeux » et « donnees$Sexe » directement comme arguments à la fonction « `chisq.test()` »; dans le second cas, la fonction va elle-même construire le tableau de contingence. Dans tous les cas, la fonction va ensuite calculer la statistique de χ^2, ainsi que la probabilité associée à cette statistique calculée. Ces deux éléments permettent de conclure ou non au rejet de H_0.

```
# Pour utiliser les données originales
Resultats <- chisq.test(donnees$Sexe, donnees$Couleur.yeux)
# ou utiliser directement le tableau de contingence que nous avons créé
Resultats <- chisq.test(Tableau.yeux)
```

   N. B.: L'argument « data= » n'est pas disponible dans la fonction « chisq.test() »

5. Et le tour est joué !! Pour faire afficher le résultat, appelez simplement l'objet par son nom :

```
> Resultats
      Pearson's Chi-squared test with Yates' continuity correction

data:  Tableau.yeux
X-squared = 3.6458, df = 1, p-value = 0.05621
```

N.B. : La fonction « summary() » ne vous donnera rien d'intéressant pour le test du χ^2, puisque tout est déjà fourni dans la sortie de base (voir à la page suivante pour une description de la sortie).

`Pearson's… :` Nom exact de la statistique utilisée pour faire le test.

`data :` Nom des séries de données (vecteurs colonnes) ou du tableau de contingence (matrice) utilisé.

`X-squared :` La valeur de la statistique de χ^2.

`df :` Le nombre de degrés de liberté.

`p-value :` Probabilité qu'on puisse rencontrer la valeur calculée de la statistique de χ^2 « `X-squared` » ou une valeur plus grande si H_0 est vraie.

6. Si vous voulez accéder à un élément précis de la sortie, ou à d'autres éléments calculés par la fonction « chisq.test() », vous pouvez les appeler à utilisant le nom de l'objet de sortie (ici « Resultats ») suivit du « $ » et du nom de l'élément souhaité (dont vous avez la liste dans la section « Value » de l'aide de la fonction « chisq.test() »).

   Par exemple, si vous voulez accéder à la statistique de χ^2, tapez :

```
Resultats$statistic
```

   Voici une description des différents éléments qui se trouvent dans l'objet de sortie produit par « `chisq.test( )` » :

`statistic`	donne la valeur de la statistique de χ^2
`parameter`	donne le nombre de degrés de liberté
`p.value`	donne la valeur de la probabilité calculée pour le χ^2 obtenu.
`method`	donne le nom exact de la statistique utilisée pour faire le test (regardez, vous pourriez être surpris !!)
`data.name`	donne le nom des séries de données (vecteurs) ou du tableau de contingence (matrice) utilisé.
`observed`	donne le tableau de contingence des données utilisées.
`expected`	donne le tableau de contingence des valeurs attendues si l'hypothèse nulle était vraie.
`residuals`	tableau de contingence des résidus calculés avec la formule: **(observed – expected) / sqrt(expected)**

Test de χ^2 par permutation

Si on veut réaliser un test de χ^2 par permutation, il suffit d'utiliser quelques arguments supplémentaires dans la fonction « chisq.test() » :

`simulate.p.value=TRUE`	donne la directive à la fonction de faire un test par permutation.
`B=xxx`	donne le nombre de permutations (où « `xxx` » est le nombre de permutations)

On obtient alors une fonction s'écrivant comme suit pour un test de χ^2 avec 999 permutations:

```
ResPerm <-chisq.test(Tableau.yeux, simulate.p.value = TRUE, B = 999)
```

Toutes les sorties sont les mêmes, sauf pour la probabilité qui a été calculée par permutations.

Passez un bon examen final et un bel été !!!