Application : violation des hypothèses, hétéroscédasticité

Nous allons, dans cette application, nous intéresser aux déterminants du prix des logements.

Afin d'importer les données dans R, utilisez la commande suivante :

```
hprice1=read.csv2(file="https://raw.githubusercontent.com/guillaume-bourgeois92/-conom-trie_l3/main/hprice1.csv",
                  check.names=F,
                  sep=",",
                  dec=".")
```

Pour réaliser ce travail, vous aurez besoin des packages suivants (si besoin les installer dans un premier temps) :

```
library(stargazer)
library(magrittr)
library(lmtest)
library(whitestrap)
library(sandwich)
library(pastecs)
library(skedastic)
library(dplyr)
```

Notre variable à expliquer est ici *price = house price in thousand dollars*. Nos variables explicatives sont :

- *lotsize = size of lot in square feet*
- *sqrft = size of house in square feet*
- *bdrms = nb of bedrooms*

1. Donner les principales statistiques descriptives des variables *price, lprice (log du prix), lotsize, sqrft* et *bdrms*. *Indication : fonction* `stat.desc` *du package pastecs, ou fonction* `stargazer` *du package éponyme*

2. Réaliser une régression linéaire multiple de *price* sur les variables *lotsize, sqrft* et *bdrms* (fonction `lm(variable dépendante~variables indépendantes séparées par un "+", data=nom_data_frame)`). Enregistrer les résultats de cette régression dans un objet "*reg1*". Afficher les résultats de la régression avec la fonction `stargazer()`.

3. **Créer une nouvelle colonne** (nommée *ehat*) dans le *data frame* hprice1 correspondant aux résidus de la régressions *reg1* : `reg1$residuals` ainsi qu'une variable correspondant aux valeurs prédites de *price* `=reg1$fitted.values`

4. Pour chacune des variables explicatives, tracer un graphique des résidus en fonction de cette variable (aide , aide2). Il faut donc ici que vous obteniez un premier graphique (nuage de points) avec en ordonnées les résidus de *reg1* (variable *ehat*) et en abscisse la variable *lotsize*. Un deuxième graphique avec en ordonnées les résidus de *reg1* (variable *ehat*) et en abscisse la variable *sqrft.* Un troisième graphique avec en ordonnées les résidus de *reg1* (variable *ehat*) et en abscisse la variable *bdrms*.

   Interpréter ces graphiques : laissent-ils penser à une hétéroscédasticité des résidus? Si oui, quelle serait la variable la plus à même d'être à l'origine de cette hétéroscédasticité?

5. En utilisant les fonctions de R, réaliser un test de Breusch-Pagan : `bptest(reg1)`
   Prenez le temps de bien rédiger le test, en précisant les hypothèses, la procédure de test et donnez l'interprétation des résultats

6. En utilisant les fonctions de R, réaliser un test de White : `white_test(reg1)`
   Attention : ce test est légèrement différent que celui que nous avons présenté dans les diapos de cours (exécuter `?white_test` pour afficher l'aide et obtenir plus d'informations)
   Prenez le temps de bien rédiger le test, en précisant les hypothèses, la procédure de test et donnez l'interprétation des résultats

7. Réaliser le test de Breusch-Pagan "étape par étape" (en n'utilisant pas les fonctions de R). Pour cela vous devez :

   7.1. Créer une variable représentant le carré des résidus de *reg1* (le carré de *ehat*) , vous pouvez ajouter cette nouvelle variable au data frame *hprice1*

   7.2. Estimer ces résidus au carré sur l'ensemble des régresseurs (point 2 sur la diapo de cours). Enregistrer les résultats de cette régression dans un objet "*reg_BP*"

   7.3. Récupérer le coefficient de détermination de la régression "*reg_BP*" :
   ```
   r2 = summary(reg_BP)$r.squared
   ```

   7.4. Récupérer le nombre d'observations de la régression "*reg_BP*" :
   ```
   n = nobs(reg_BP)
   ```

   7.5. Calculer la statistique de Lagrange (BPc) égale au produit entre le R^2 et le nombre d'observations. On sait que cette statistique suit une loi du chi2 à *k* degrés de liberté, où k est le nombre de régresseurs dans l'estimation "*reg_BP*". Enregistrer cette valeur dans un objet *LM_stat*.

   7.6. Calculer la *p-value* associée au test :
   `pchisq(q = LM_stat, df = …, lower.tail = FALSE)`. Sans l'option `lower.tail=FALSE` on calcule *1-pvalue, df* représente le nombre de degrés de liberté (le nombre de régresseurs dans "*reg_BP*").

7.7. Conclure sur le résultat de ce test (normalement, votre conclusion doit être la même que celle formulée avec la fonction `bptest()`)

8. Réaliser le test de White “étape par étape” (en n’utilisant pas les fonctions de R). Pour cela vous devez :

   8.1. Créer les variables au carré des régresseurs, par exemple
   `hprice1$lotsizesq = hprice1$lotsize^2`

   8.2. Créer trois nouvelles variables, représentant les intéractions entre les régresseurs, par exemple :
   `hprice1$lotsizeXsqrft = hprice1$lotsize*hprice1$sqrft`

   8.3. Estimer les résidus au carré de *reg1* sur les régresseurs, les régresseurs au carré, et les intéractions entre les régresseurs. Enregistrer les résultats de cette régression dans un objet “*reg_white*”.

   8.4. Calculer la statistique de Lagrange (Wc) égale au produit entre le R^2 et le nombre d’observations. On sait que cette statistique suit une loi du chi2 à *k* degrés de liberté, où *k* est le nombre de régresseurs dans l’estimation “*reg_white*”. Enregistrer cette valeur dans un objet *LM_stat*.

   8.5. Calculer la *p-value* associée au test :
   `pchisq(q = LM_stat, df =..., lower.tail = FALSE)`. Sans l’option `lower.tail=FALSE` on calcule *1-pvalue, df* représente le nombre de degrés de liberté (le nombre de régresseurs dans “*reg_BP*”).

9. Calculer les écarts-types estimés des coefficients estimés, corrigés de l’hétéroscédasticité (avec la correction de White) et faire les nouveaux tests de student :
   `coeftest(reg1, vcov. = vcovHC(reg1, type = "HC1"))`. Voyez-vous des différences avec les tests de Student sans correction de l’hétéroscédasticité?

10. Réaliser une régression WLS, en considérant que la variable *lotsize* est celle qui cause l’hétéroscédasticité. Ainsi, nous avons la relation suivante pour la variance des erreurs : $var(\epsilon|x) = \sigma^2 \times lotsize$. Pour cela :

    10.1. Créer les nouvelles variables : régresseurs corrigés par $\frac{1}{\sqrt{lotsize}}$, en n’oubliant pas la nouvelle “constante” :
    ```
    hprice1$constantstar = 1/sqrt(hprice1$lotsize)
    hprice1$pricestar = hprice1$price/sqrt(hprice1$lotsize)
    hprice1$lotsizestar=hprice1$lotsize/sqrt(hprice1$lotsize)
    ```

    Il vous reste à créer *sqrftstar* et *bdrmsstar*

10.2. Réaliser l'estimation des moindres carrés pondérés, sur ces nouvelles variables corrigées :

```
reg_WLS = lm(pricestar ~ 0 + constantstar+ lotsizestar + sqrftstar + bdrmsstar, data=hprice1)
```

Le `0` après le ~ signale à R qu'on ne souhaite pas que R considère une constante dans le modèle

10.3. Réaliser un test de Breusch-Pagan et un test de White sur la régression reg_WLS de la question 10.2. (`bptest(reg_WLS)` pour Breusch-Pagan). Quelles conclusions en tirez-vous?

11. Considérons à présent qu'on ne connaît pas la source de l'hétéroscédasticité. Nous allons donc réaliser une régression FGLS. Il nous faut donc dans un premier temps estimer la fonction $h(x_i)$ notée $\widehat{h(x_i)}$

11.1. Calculer le logarithme des résidus au carré de la régression *reg1*

```
hprice1$ehat=reg1$fitted.values
hprice1$ehatsq=hprice1$ehat^2

hprice1$g=log(hprice1$ehatsq)
```

11.2. Régresser ce logarithme des résidus au carré sur les trois régresseurs de l'équation de base (*lotsize, sqrft* et *bdrms*) (fonction `lm()`). Enregistrer les résultats de cette régression dans un objet "*reg_g*"

11.3. Récupérer les valeurs prédites du logarithme des résidus au carré de l'estimation de la question 11.2. : `hprice1$ghat = reg_g$fitted.values`

11.4.Calculer les valeurs de la variable $\widehat{h}$, en prenant l'exponentielle de la variable *ghat* (fonction `exp()`)

11.5 Créer les nouvelles variables pour la régression FGLS, en les corrigeant par $\frac{1}{\sqrt{\widehat{h}}}$

, en n'oubliant pas de créer la nouvelle "constante" :

```
hprice1$constantstar1 = 1/sqrt(hprice1$hhat)
hprice1$pricestar1 = hprice1$price/sqrt(hprice1$hhat)
…
```

11.6. Réaliser l'estimation FGLS :

```
reg_FGLS2 = lm(pricestar1 ~ 0 + constantstar1 + lotsizestar1 + sqrftstar1 + bdrmsstar1, data=hprice1)
```

11.7. Vérifier avec un test de Breusch-Pagan et de White le comportement des résidus de l'estimation reg_FGLS2

12. Reprendre toutes les questions (sauf la 1.) avec pour variable dépendante le logarithme du prix (*lprice*). A-t-on toujours un problème d'hétéroscédasticité?