Projet ModIA

Analyse de données - Eléments de modélisation statistique

B. Aussel, A. Cintas, B. Drai

Contents

1 Introduction

Ces dernières décennies, la prise de conscience collective sur le respect de l'environnement a fait prendre au gouvernement certaines mesures. Le diagnostic de performance énergétique (DPE) des bâtiments est une de ces mesures. Elle permet d'évaluer les bâtiments selon leur consommation énergétique et leur émission de gazs à effet de serre. Ainsi, le DPE classe les bâtiments selon sept classes, de A à G, où A représente un bâtiment avec une très bonne performance énergétique et une faible émission de CO_2.

Mis à part le côté environmental lié à ce diagnostic, il y a aussi un réel impact financier. En effet, dans certaines villes de France, les logements classés A coûtent en moyenne 68% plus cher que ceux classés E.

Il y a donc une certaine nécessité à construire dans le futur des logements verts afin de répondre aux normes environnementales mais aussi à la demande des acheteurs qui considèrent de plus en plus ce critère dans leur choix de logement.

L'objectif de notre étude est de prédire la classe énergétique des bâtiments à partir de certaines de leurs caractéristiques comme les différentes superficies, l'orientation ou encore la surface vitrée.

Pour cela, nous utiliserons les données de quelques centaines de bâtiments. Nous serons tout d'abord amenés à analyser ces données afin de les comprendre et de distinguer les liens pré-existants entre variables. Nous progresserons au long de l'étude en comparant 2 approches différentes de modélisation : la régression et la classification de l'énergie émise par les bâtiments. A cette fin, nous expliquerons cette variable avec des modèles linéaires et non linéaires. Nous utiliserons donc des régréssions linéaires, des analyses de covariance, des arbres de classification et de régression mais aussi des fôrets aléatoires.

Nous analyserons ces modèles en détails, tenterons de les simplifier et de les optimiser afin que la prédiction qui en découle soit la meilleure possible.

Finalement, nous les comparerons entre eux dans le but d'identifier le modèle qui prédit le mieux l'énergie émise et qui est le plus simple à mettre en oeuvre.

2 Analyse des données

L'objectif de cette première partie est de comprendre les données que nous modéliserons. Pour cela, nous ferons une analyse unidimensionnelle de chacune de nos variables de notre jeu de données. Puis dans un deuxième temps, nous analyserons les variables par paires afin de voir s'il existe des liens entre notre variable. Nous examinerons également les variables qui ont l'air d'influer le plus sur notre variable de sortie. Finalement, nous ferons du clustering sur nos données afin de les regrouper en classes.

```
df <- read.csv("DataEnergy_Student_V2.csv")
```

On observe un jeu de données contenant 768 individus décrits par 10 variables.

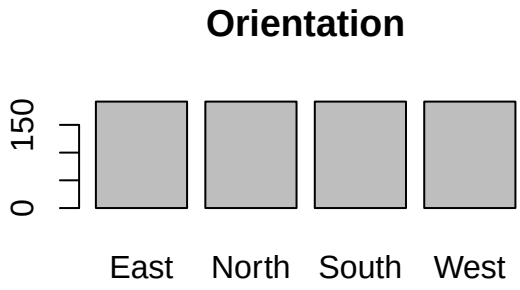
Quatre d'entre elles sont qualitatives: *Orientation*, *Energy.efficiency* , *Overall.height* et *Glazing.area.distr.*

Les six autres sont quantitatives : *Surface.area*, *Wall.area*, *Roof.area* *Glazing.area*, *Energy* et *Relative.compactness.*

2.1 Analyse des variables qualitatives

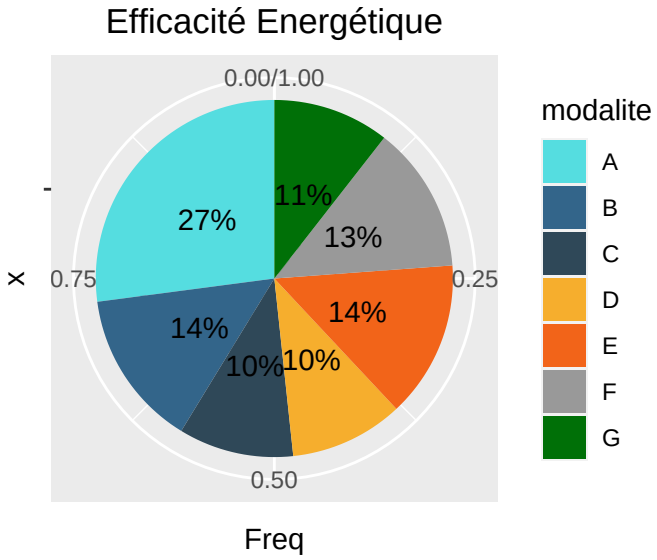
Dans cette sous-partie, nous étudierons les variables qualitatives une à une.

2.1.1 Orientation



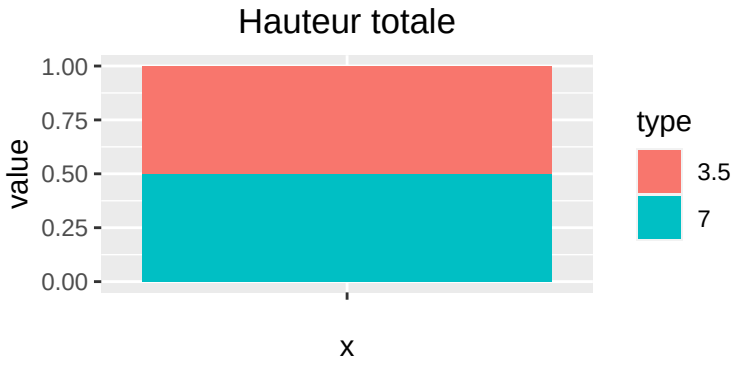
La variable *Orientation* est équitablement répartie entre ses 4 directions (Nord, Sud, Est, Ouest). Il y a donc 192 individus orientés dans chaque direction.

2.1.2 Energy.Efficiency



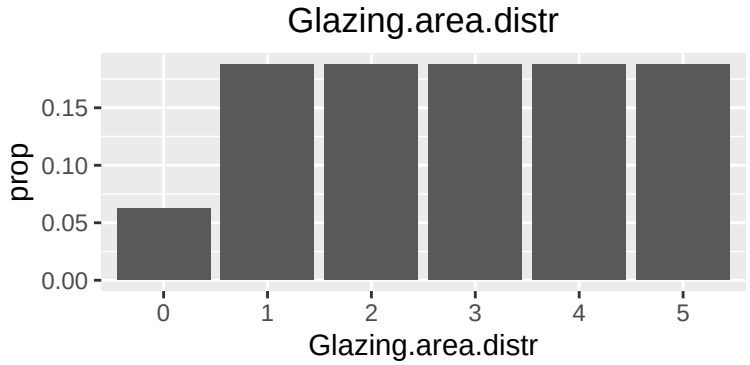
La variable *Energy.efficiency* possède 8 “notes” de dépense en énergie entre A et G. On remarque qu’environ 51% des appartements ont une très bonne dépense d’énergie (A,B ou C), tandis que 10% appartiennent à la catégorie D qui représente une dépense d’énergie moyenne. Enfin, environ 39% d’entre eux ont une mauvaise dépense en énergie (E, F ou G).

2.1.3 Overall.Height



La variable *Overall.height* contient 2 mesures différentes : 3.5m et 7m. Elle est équitablement répartie : 384 bâtiments sont de hauteur 3.5m et 384 sont de hauteur 7m.

2.1.4 Glazing.area.distr



La variable *Glazing.area.distr* représente 6 catégories de disposition différentes de fênetres :

- 0: il n'y a pas de fenêtre.
- 1: c'est une répartition uniforme 25% de chaque coté.
- 2: 55% des fenetres au nord et 15% sur les autres cotés.
- 3: 55% des fenetres à l'est et 15% sur les autres cotés
- 4: 55% des fenetres au sud et 15% sur les autres cotés.
- 5: 55% des fenetres à l'ouest et 15% sur les autres cotés

On constate qu'il y a 48 appartements qui n'ont pas de fênetre soit une proportion de 6.25%. Les autres catégories sont équitablement réparties : il y a 144 individus différents dans chaque catégorie soit 18.75%.

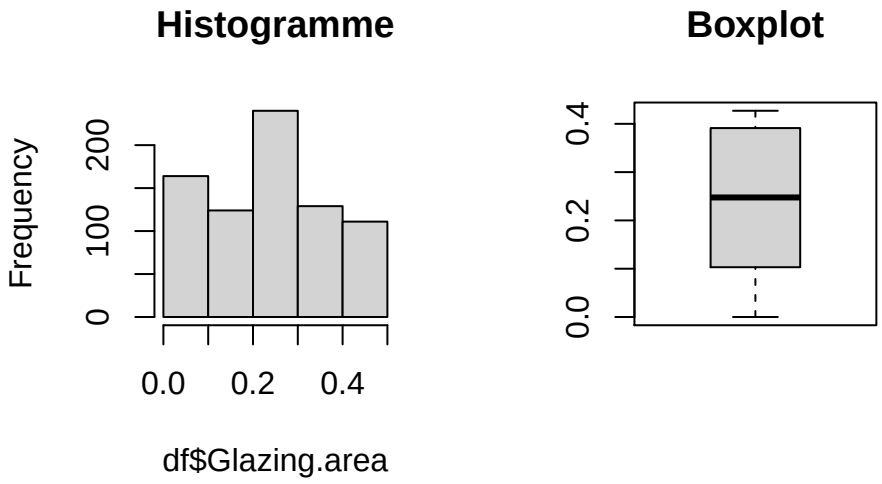
2.2 Analyse des variables quantitatives

Dans cette sous-partie, nous étudierons les variables quantitatives une à une.

2.2.1 Glazing.area

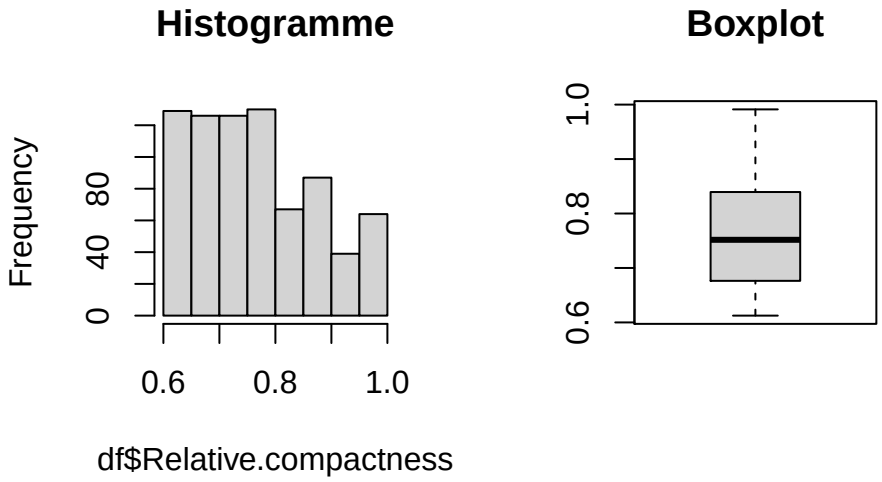
La variable *Glazing.area* représente la proportion de surface vitrée par rapport à la surface du sol.

Nous avons pu remarquer dans l'affichage des données que la variable *Glazing.area* contient des valeurs très proches de 0. Il se trouve que ces individus ont un *Glazing.area.distr* nul. On décide, par soucis de précision, de mettre à 0 ces lignes là pour avoir une variable plus homogène.



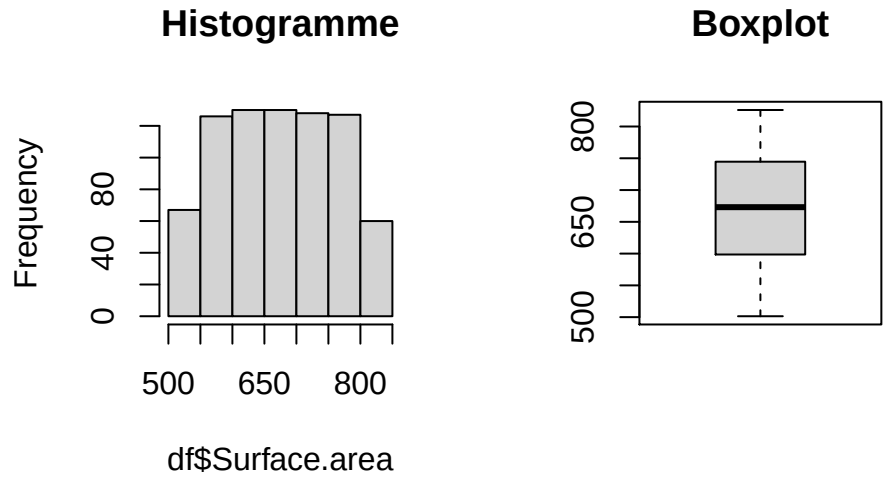
La proportion de surface vitrée par rapport à la surface du sol varie donc entre 0% et 42% avec une majorité des individus ayant cette proportion comprise entre 20% et 30%.

2.2.2 Relative.Compactness



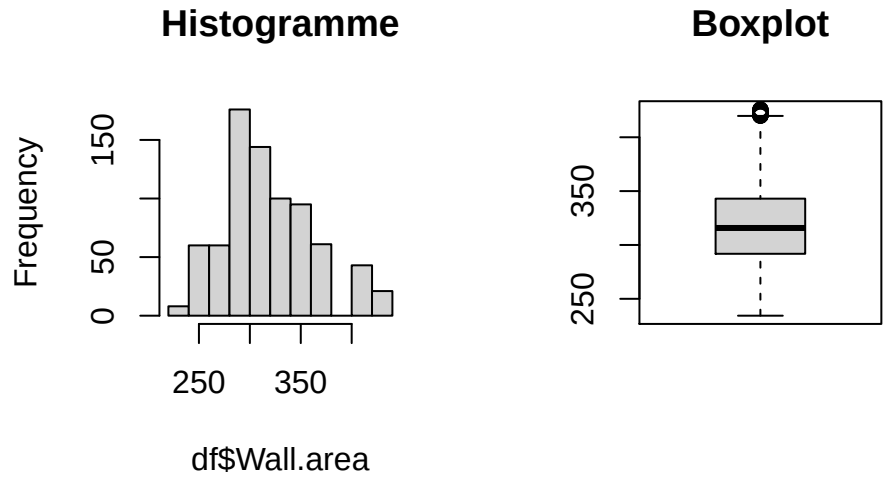
La compacité des matériaux varie entre environ 0.6 et 0.99, avec une majorité des batiments ayant une compacité comprise entre 0.6 et 0.8.

2.2.3 Surface.Area



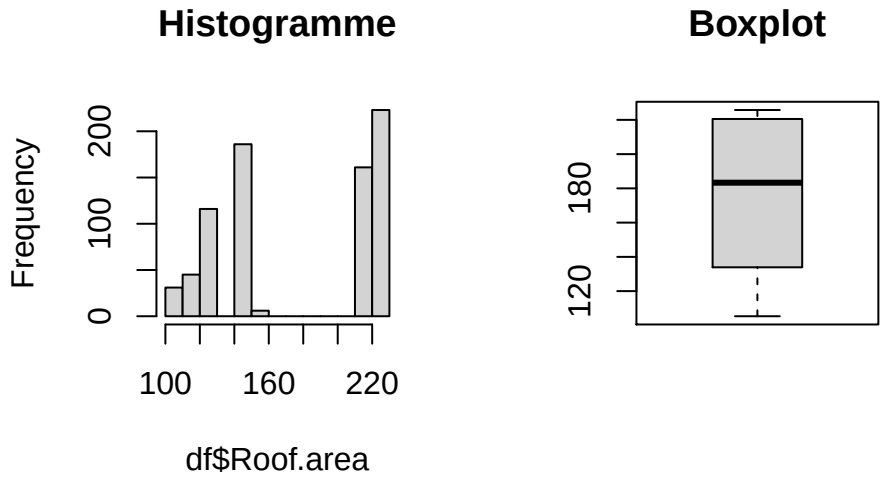
La **surface** du sol des batiments semble être équirépartie autour de la moyenne de $671.3m^2$ avec des valeurs allant d'environ $500m^2$ à $826m^2$.

2.2.4 Wall.Area



On remarque ici que la surface des murs, en excluant les quelques outliers, est principalement répartie autour de la médiane qui vaut $315.8m^2$, avec une majorité de batiments ayant une surface comprise entre $300m^2$ et $350m^2$.

2.2.5 Roof.Area

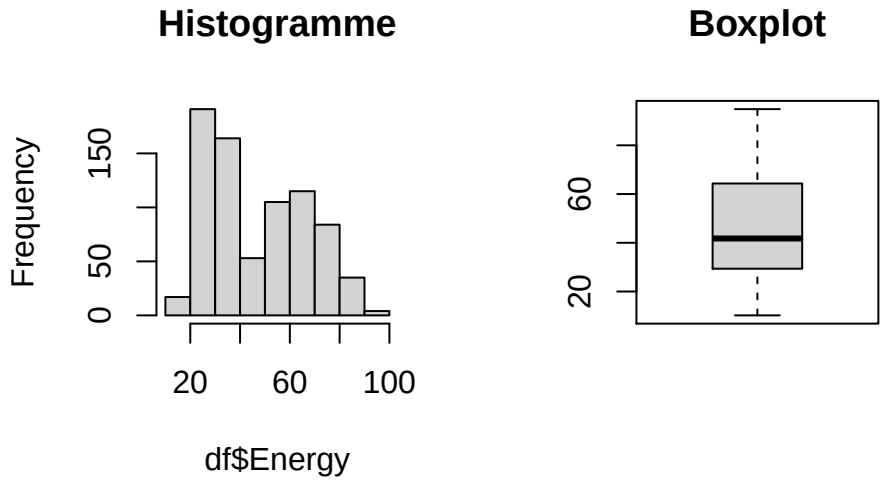


On ne remarque aucune symétrie ici. Deux modes distincts semblent se détacher :

- Des toits d'aire comprise entre $100m^2$ et $160m^2$.
- Des toits d'aire comprise entre $210m^2$ et $230m^2$.

On a plus précisément : 384 bâtiments avec une surface de toit supérieure à $180m^2$ et donc 384 batiments avec une surface de toit inférieure à $180m^2$.

2.2.6 Energy



La charge énergétique semble répartie de manière bi-modale :

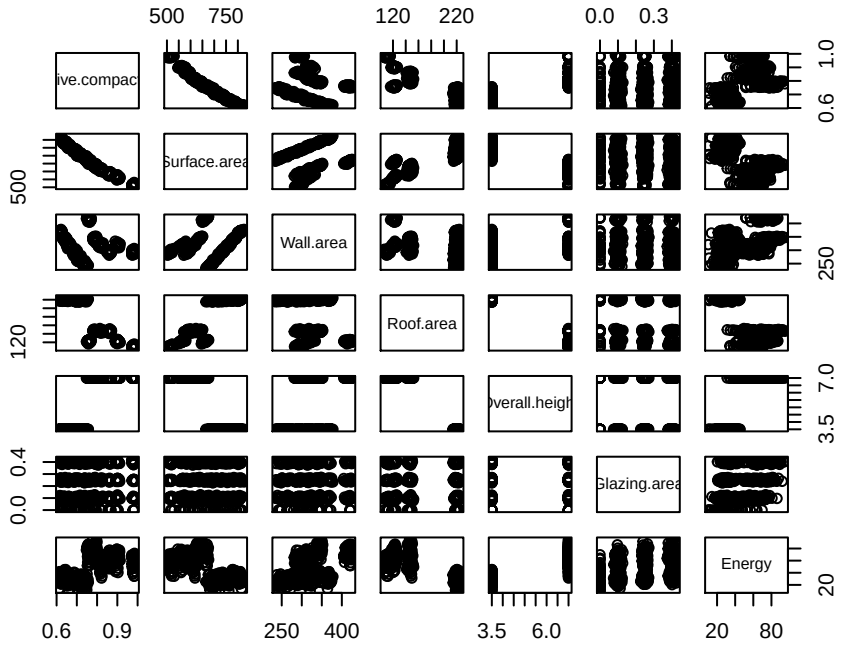
- un premier mode entre 20 et 40
- un deuxième entre 50 et 80

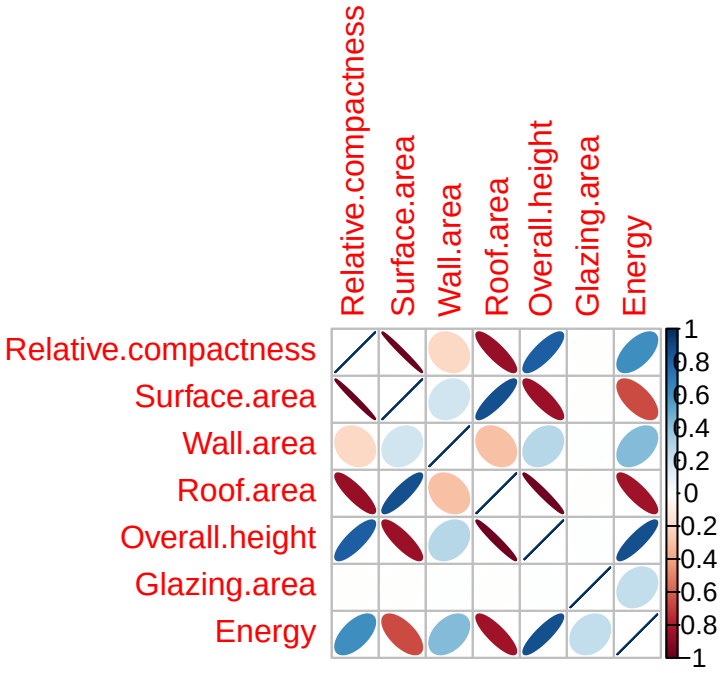
2.3 Analyse Bidimensionnelle

Nous allons à présent nous intéresser à l'étude bidimensionnelle des variables. Dans un premier temps, nous chercherons des relations entre toutes les variables quantitatives afin de simplifier notre jeu de données. Dans un second temps, nous étudierons plus précisément le poids des variables sur notre variable de sortie *Energy*.

2.3.1 Entre toutes les variables

Commençons par visualiser les tracés par paire de chacune des variables ainsi que le diagramme des correlations entre les variables quantitatives.





Nous pouvons relever que *Glazing.area* est la seule variable qui n'a aucune corrélation avec les autres variables. Les tracés bidimensionnelles de cette variable montrent qu'elle est divisée en facteur avec les autres variables.

De plus, il semble y avoir une forte relation linéaire entre les variables *Relative.compactness* et *Surface.area*. En effet, le tracé bidimensionnelle se rapproche d'une droite de pente négative et leur corrélation est très proche de -1.

Nous pouvons aussi remarquer une forte corrélation entre les variables de surface, à savoir *Surface.area*, *Wall.area* et *Roof.area*. Il est donc judicieux de chercher une relation linéaire entre ces variables. Nous pouvons intuitionner d'après les significations de ces trois variables que :

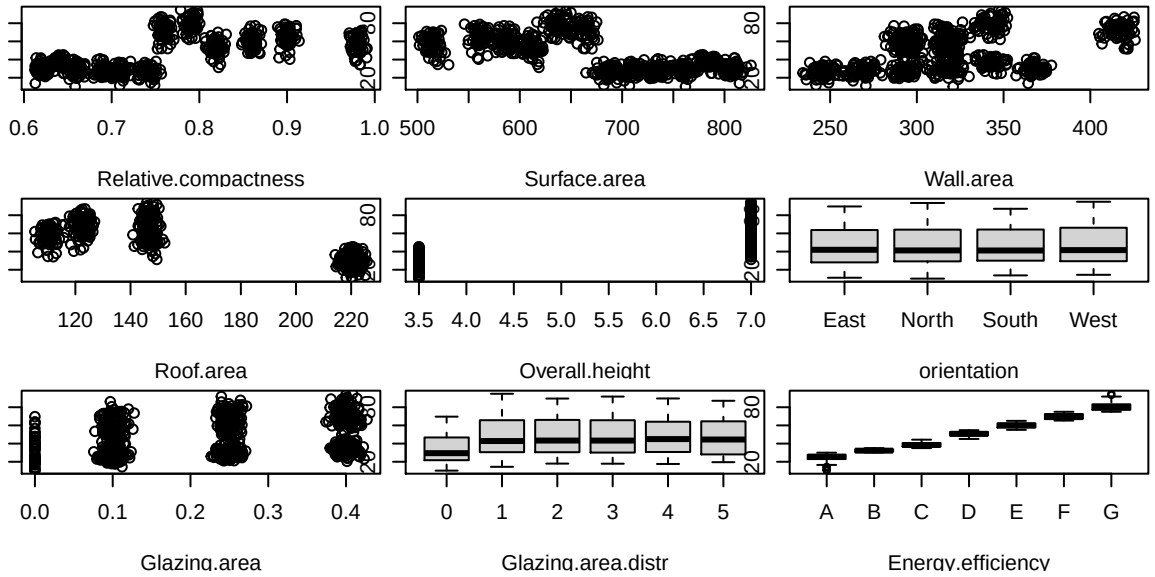
$$Surface.area = 2 * Roof.area \ + \ Wall.area$$

Vérifions le : `sum(abs(2*df$Roof.area + df$Wall.area - df$Surface.area))` $= 4.0569148 \times 10^{-10}$

En sommant les données selon cette formule, on a bien un résultat très proche de 0. On en conclut que nous pouvons déduire la variable *Surface.area* des deux autres.

2.3.2 Avec la sortie Energy

Intéressons nous plus particulièrement à l'impact des différentes variables sur notre variable d'output *Energy*.



On remarque que l'orientation n'a aucun impact sur l'efficacité energétique. En effet, le graphique obtenu montre que pour les différentes orientations, l'efficacité energétique ne présente aucune différence.

La variable *Overall.height* a visiblement un lien avec *Energy*. En effet, pour la modalité $3.5m$, *Energy* varie entre 5 et 45 alors que pour la modalité $7.0m$ elle varie entre 35 et 90.

De même, il semblerait que les variables *Relative.compactness* et *Surface.area* impactent notre sortie car on peut remarquer que les valeurs sont divisées en deux paliers. Pour *Relative.compactness*, ces niveaux sont autour de 30 et 60 et pour *Surface.area* ils sont autour de 30 et 70.

Notons que le lien entre *Energy* et *Energy.efficiency* est biaisé car la variable *Energy.efficiency* est la version qualitative de *Energy* en la divisant en différentes classes energétiques.

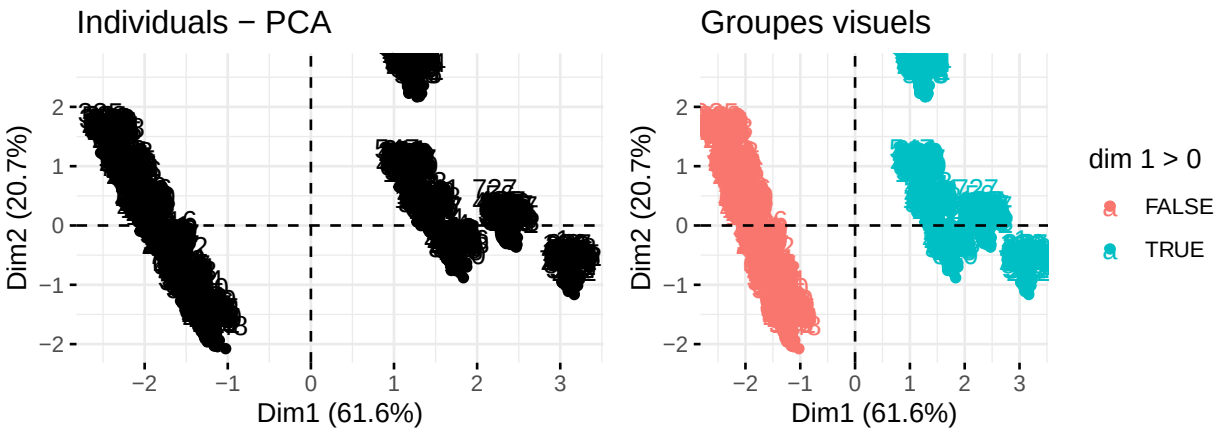
Nous pouvons aussi remarquer que la variable *Glazing.area.distr* a le même effet sur *Energy* pour les valeurs $1, 2, 3, 4, 5$. Nous avons donc tout interêt à transformer cette variable en une variable bi-modale prenant les valeurs :

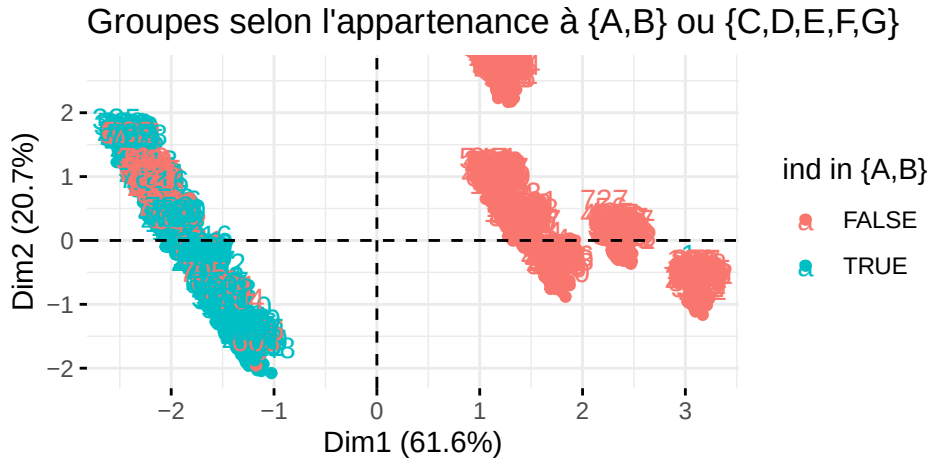
- 1 lorsque le bâtiment possède des fenêtres;
- 0 lorsqu'il n'y en a pas.

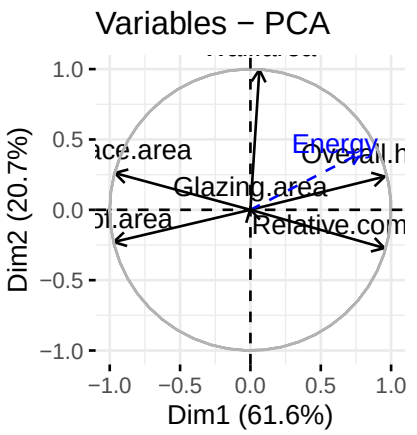
2.4 Clustering

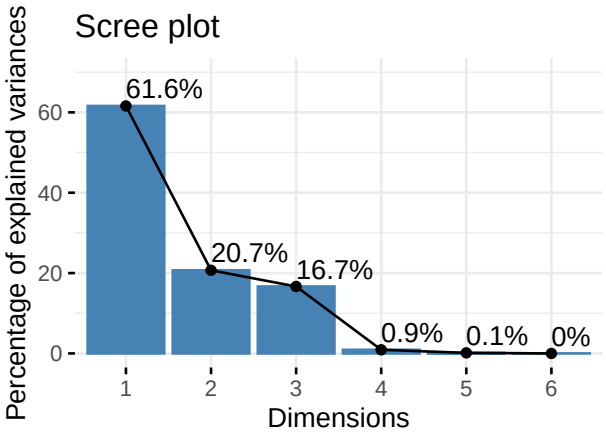
Ici, nous utiliserons l'analyse en composantes principales (ACP) et le classement ascendant hiérarchique dans le but de découvrir des clusters dans nos données.

Il est intéressant ici de tenter de se représenter notre nuage de points initialement dépendant de 8 variables en seulement 2 variables. Pour cela on va utiliser la technique de l'ACP qui consiste à trouver les meilleures combinaisons linéaires de variables pré-existantes, au sens d'une maximisation de l'inertie projetée :





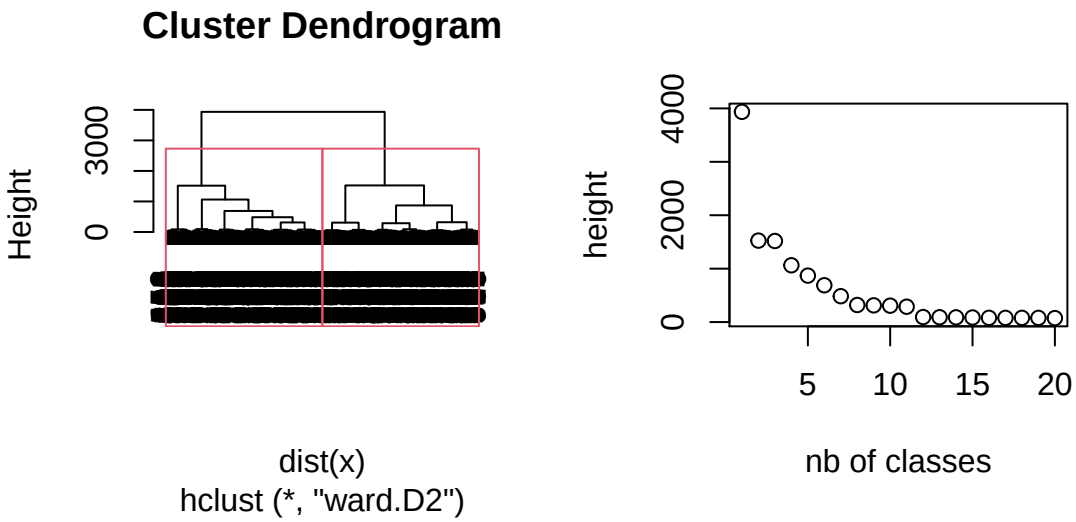




On voit que 82% de l'information est considérée en ne gardant que les deux premières composantes principales. On voit de plus visuellement l'apparition de deux groupes distincts. En coloriant les points en fonction de leur appartenance à la classe énergétique {A,B} ou {C,D,E,F,G}, il semble y avoir une corrélation avec cette classification. Le cercle des corrélations nous confirme les corrélations précédemment trouvées dans le corrplot. De plus on voit bien que l'information relative aux variables *Relative.compactness*, *Overall.height*, *Surface.area* et *Roof.area* est portée principalement par la première composante principale, et la variable *Wall.area* est principalement portée par la deuxième composante principale.

On va donc tenter de vérifier par hierarchical clustering qu'il est possible de construire deux classes à partir

de nos données :



Cette méthode de classification hiérarchique montre qu'on peut diviser nos données en deux classes distinctes. En effet, on distingue ici que le plus gros saut a lieu lorsque l'on sépare nos données en 2 classes.

Ceci conforte notre résultat précédemment évoqué lors de notre Analyse en Composantes Principales.

3 Modèles Linéaires

Dans cette section, nous allons tenter de trouver le modèle linéaire le plus approprié pour modéliser nos données.

Pour cela, nous chercherons dans un premier temps le meilleur modèle linéaire pour expliquer la variable *Energy*. Puis dans un second temps, nous nous intéresserons à la variable de sortie *Energy.efficiency* en écrivant un modèle linéaire généralisé.

Tous les modèles étudiés ne comprennent pas la variable *Surface.area* car nous avons montré que : $Surface.area = 2 * Roof.area \ + \ Wall.area$.

3.1 Avec les variables quantitatives

Dans un premier temps, nous allons comparer des modèles sans les variables qualitatives.

Pour commencer, prenons un modèle linéaire avec toutes les interactions des variables.

On obtient un R^2 ajusté de 0.90. Nous allons essayer de simplifier ce modèle en utilisant le critère AIC.

Le modèle proposé par le critère AIC est le suivant :

$$\begin{aligned} \text{Energy} \sim & Relative.compactness + Wall.area + Roof.area + Overall.height + Glazing.area \\ & + Relative.compactness * (Roof.area + Overall.height + Glazing.area) \\ & + Wall.area * (Roof.area + Overall.height + Glazing.area) + Roof.area * Overall.height \end{aligned}$$

Testons ce modèle :

```r
anova(mod_quanti_int_simplifie, mod_quanti_int)
```

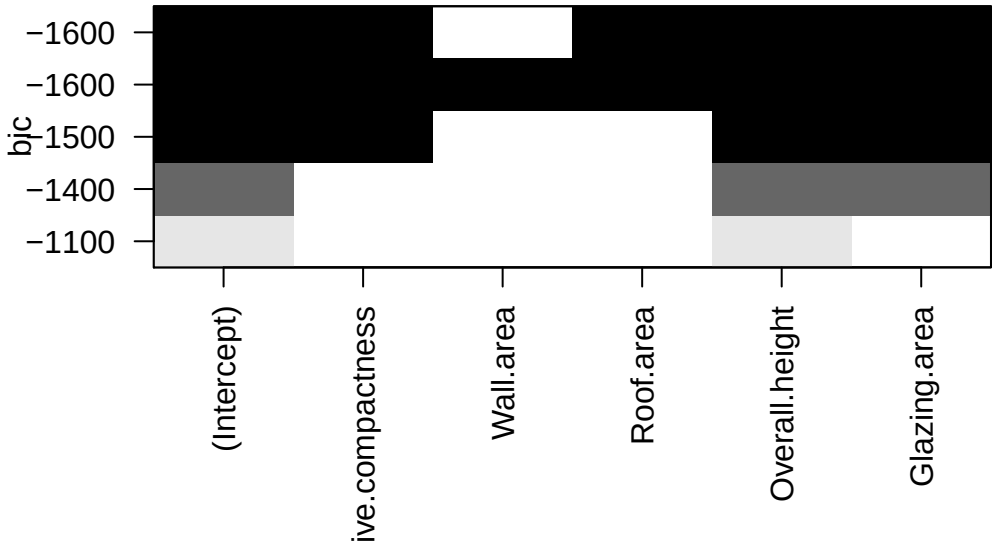
Le test de sous-modèle nous donne une p-value de 0.9896. On ne rejette donc pas H0 et on garde le sous-modèle donné par stepAIC. Avec ce modèle simplifié, on garde le même R^2 ajusté mais en ayant enlevé trois interactions entre les variables. En revanche, nous avons tout de même un modèle lourd avec beaucoup d'intéractions.

Les interactions entre variables rendent le modèle pesant. A présent, nous allons enlever les intéractions et prendre un modèle additif et voir si les intéractions sont nécessaires à la fiabilité de notre modèle.

```
##
## Call:
## lm(formula = Energy ~ . - Energy.efficiency - Glazing.area.distr -
##     orientation - Surface.area, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.9520  -4.4175  -0.0396   4.4977  25.4280
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           7.785e+01  3.267e+01   2.383   0.0174 *
## Relative.compactness -7.899e+01  1.775e+01  -4.450 9.88e-06 ***
## Wall.area            -1.313e-04  2.211e-02  -0.006   0.9953
## Roof.area            -1.690e-01  5.961e-02  -2.835   0.0047 **
## Overall.height        9.662e+00  6.976e-01  13.849  < 2e-16 ***
## Glazing.area          3.677e+01  1.902e+00  19.332  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.01 on 762 degrees of freedom
## Multiple R-squared:  0.8763, Adjusted R-squared:  0.8755
## F-statistic:  1079 on 5 and 762 DF,  p-value: < 2.2e-16
```

Ce modèle a un $R^2 = 0.8755$ et ne contient que 5 variables. Nous pouvons remarquer que le test de nullité de la variable *Wall.area* nous donne une p-value proche de 1, ce qui nous permettrait d'accepter la nullité de cette variable. Regardons plus en détail si d'autres variables sont susceptibles d'être supprimées du modèle.

Pour cela, utilisons une méthode de séléction de variable type BIC :



On en conclut qu'on pourrait à priori n'enlever que la variable *Wall.area*.

Verifions cela par un test de sous-modèle de Fisher :

```
anova(mod_quanti_simplifie, mod_quanti)
```

On obtient une p-value de 0.9953 donc on accepte le sous-modèle. Pour ce modèle simplifié sans interactions, on obtient un R^2 ajusté de 0.88. Ce résultat est certes légèrement plus faible que ce que nous avions obtenu avec les intéractions mais notre nouveau modèle contient bien moins de variables que le précédent.

Donc finalement, le modèle retenu dans cette partie est :

$$Energy \sim Relative.compactness + Roof.area + Overall.height + Glazing.area$$

Testons maintenant l'efficacité de la prédiction de ce modèle. Pour cela, nous divisons notre jeu de données en un jeu d'entraînement de notre modèle et un jeu de test. Nous entrainons ensuite notre modèle sur le jeu d'entrainement, et nous le testons sur notre jeu de test.

Ce premier modèle a un taux de bonnes prédictions de 61.69%.

3.2 Ajout des variables qualitatives

A présent, nous allons tenter de compléter notre modèle en y ajoutant les variables qualitatives.

De la même manière que dans la première partie, nous allons utiliser le critère AIC pour simplifier le modèle avec toutes les intéractions entre les variables.

Le modèle proposé par le critère AIC est le suivant :

$$\begin{aligned} Energy \sim & Wall.area * Glazing.area.distr + Relative.compactness * Glazing.area.distr \\ & + Roof.area * Overall.height + Relative.compactness * Overall.height + Roof.area * Glazing.area \\ & + +Relative.compactness * Overall.height + Wall.area * Roof.area + Wall.area * Overall.height \end{aligned}$$

Faisons un test de sous-modèle de Fisher pour voir la validité de ce sous-modèle :

```
anova(mod_complet_aic, mod_complet_int)
```

On obtient une $pvalue = 2.2 * 10^{-16}$ ce qui nous amène à rejeter ce modèle.

Essayons de prendre un modèle additif avec toutes les variables de notre jeu de données et utilisons le critère AIC pour faire notre sélection de variables:

Le modèle proposé est :

$$Energy \sim Glazing.area.distr + Roof.area + Glazing.area + Overall.height + Relative.compactness$$

Testons ce modèle :

```
anova(mod_complet_aic2, mod_complet)
```

Le test de sous-modèle nous donne une $pvalue = 0.6774$. Nous pouvons donc accepter ce modèle qui nous permet d'avoir un R^2 ajusté de 0.8801.

Cette étude de modèle linéaire associant variables quantitatives et qualitatives nous amène à considérer le modèle suivant :

$$Energy \sim Glazing.area.distr + Roof.area + Glazing.area + Overall.height + Relative.compactness$$

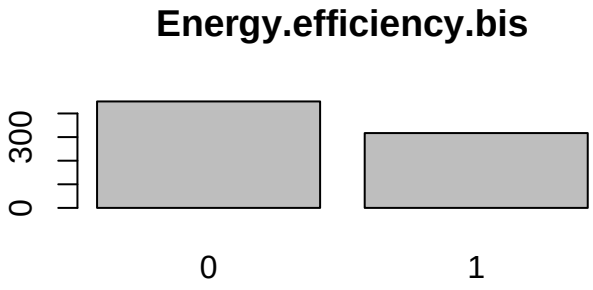
De même que précédemment, nous allons tester la prédiction de ce modèle.

Ce deuxième modèle plus complet a un taux de bonnes prédictions de 59.74%. Celui-ci est plus faible que le modèle précédent avec seulement les variables quantitatives. Ceci peut s'expliquer par un phénomène d'overfitting dû au nombre plus élevé de variables dans ce modèle.

3.3 Modèles Linéaires Généralisés

Nous avons pu montrer dans l'ACP que la variable de sortie *Energy_efficiency* se divise en deux classes majeures : la première contenant les modalités {A, B} et l'autre avec les modalités {C, D, E, F, G}.

Créons donc une nouvelle variable de sortie *Energy.efficiency.bis* binaire qui prend la valeur 1 lorsque les modalités sont dans {A, B} et 0 sinon :



Dans cette partie, nous écrirons un modèle de régression logistique multiple additif afin d'expliquer *Energy.efficiency.bis* en fonction de toutes les autres variables :

```
##
## Call:
## glm(formula = Energy.efficiency.bis ~ ., family = binomial(link = "logit"),
##     data = df_bis)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.26245  -0.00005  -0.00001   0.37803   2.08055
##
## Coefficients: (1 not defined because of singularities)
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              8.67895 1603.84546   0.005   0.9957
## Relative.compactness    28.32129   15.16249   1.868   0.0618 .
## Surface.area             0.03336    0.02682   1.244   0.2136
## Wall.area               -0.02913    0.01817  -1.603   0.1088
## Roof.area                    NA         NA      NA       NA
## Overall.height          -6.43860  229.09102  -0.028   0.9776
## orientationNorth        -0.20084    0.41656  -0.482   0.6297
## orientationSouth        -0.43445    0.41103  -1.057   0.2905
## orientationWest         -0.22924    0.41413  -0.554   0.5799
## Glazing.area            -6.85743    1.33942  -5.120 3.06e-07 ***
## Glazing.area.distr2    -17.53042  801.81278  -0.022   0.9826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1041.17  on 767  degrees of freedom
## Residual deviance:  305.94  on 758  degrees of freedom
## AIC: 325.94
##
## Number of Fisher Scoring iterations: 19
```

Nous pouvons remarquer dans le résumé de ce modèle additif que plusieurs variables ont une p-value de leur Z-Test bien supérieure à 5%. Il est donc judicieux d'utiliser des algorithmes de séléctions de variables pour simplifier ce modèle.

Pour cela, nous allons utiliser les méthodes de séléction de variables par critères BIC et AIC.

Le modèle proposé par le critère BIC est le suivant :

$$logit(\pi_i) = \theta_0 + \theta_1\ Relative.compactness_i + \theta_2\ Overall.height_i + \theta_3\ Glazing.area_i + \theta_4 \mathbb{1}_{Glazing.area.distr=2}$$

Le modèle proposé par le critère AIC est le suivant :

$$logit(\pi_i) = \theta_0 + \theta_1\ Relative.compactness_i + \theta_2\ Overall.height_i + \theta_3\ Glazing.area_i + \theta_4\ Roof.area_i + \theta_5 \mathbb{1}_{Glazing.area.distr=2}$$

Nous pouvons remarquer que la variable *Roof.area* est présente sur le modèle selectionné avec le critère AIC mais pas sur celui avec le critère BIC.

Dans une optique de simplification de notre modèle, on se propose de garder le modèle avec le moins de paramètres, donc celui généré avec le critère BIC.

Faisons un test d'ANOVA pour vérifier la validité de notre modèle :

```
anova(mod_log.bic, mod_log, test="Chisq")
```

On obtient une p-value de 0.59, on ne rejette pas le sous modèle. Le modèle simplifié séléctionné est donc :

$$logit(\pi_i) = \theta_0 + \theta_1\ Relative.compactness_i + \theta_2\ Overall.height_i + \theta_3\ Glazing.area_i + \theta_4\ \mathbb{1}_{Glazing.area.distr=2}$$

Afin de vérifier le pouvoir prédictif de notre modèle, comparons les valeurs de la variable réponse avec celles des valeurs prédites dans une table de contingence :

```
##
##     FALSE TRUE
##   0   398   53
##   1    16  301
```

Nous pouvons constater que la prédiction faite avec ce modèle permet de trouver dans 91.02% des cas la bonne classe énergétique.

Il y a tout de même encore 8.98% des cas où l'energie est dans la catégorie {C,D,E,F,G} mais est prédite dans la catégorie {A,B}, ce qui reste quand même raisonnable.

Remarque : nous avons essayé d'implémenter le modèle polytomique, mais sans succès.

4 Modèles Non-Linéaires

Dans cette section, nous allons écrire plusieurs modèles non-linéaires du type : arbre de classification et de régréssion, et forêts aléatoires.

4.1 Arbres

Dans cette partie, nous allons générer des arbres de régression afin d'expliquer notre variable quantitative de sortie *Energy*, ainsi que des arbres de classification pour expliquer la variable qualitative *Energy.efficiency*. Enfin, nous analyserons les résultats obtenus pour les deux arbres.

4.1.1 Arbre de régression

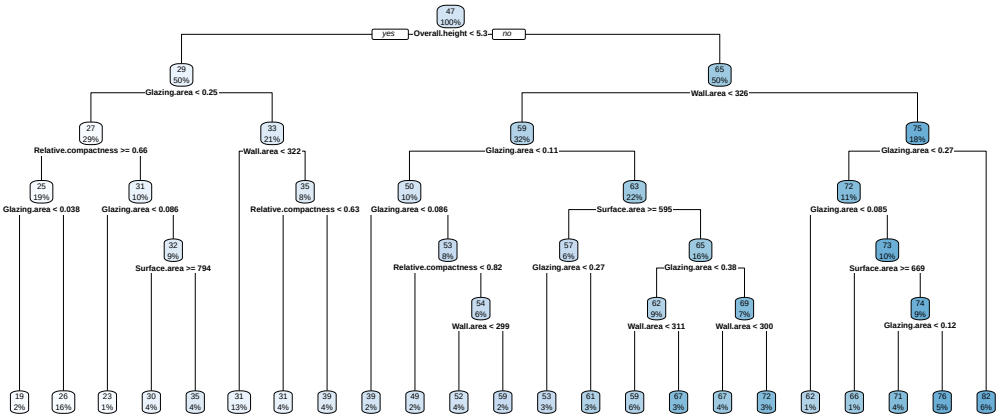
On génère ici un arbre de régression afin d'estimer la valeur de l'énergie émise de la variable *Energy*, et de pouvoir la classifier.

Le critère de pénalisation que nous avons choisi par défaut est une valeur prise arbitrairement, à savoir : $C_p = 0.001$.

Avant d'analyser plus en détail notre arbre de régréssion, on va chercher quel critère de pénalisation permet de réduire au maximum l'erreur.

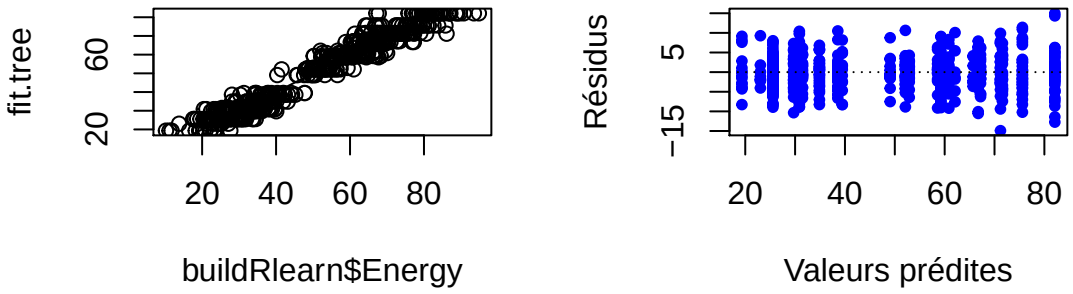
Après calcul, on obtient un critère de pénalisation optimale $Cp_{min} = 0.0010068$

On le ré-injecte dans l'arbre:



Nous pouvons remarquer que les variables qui reviennent essentiellement sont : $Glazing.area$, $Wall.area$, $Surface.area$ et $Relative.compactness$.

Etudions à présent les résidus ainsi que les valeurs ajustées. On se propose dans un premier temps d'afficher les valeurs prédites en fonction des vraies valeurs, ainsi que les résidus :



La structure en 'strates' est due au fait que les valeurs proches en feuille terminale de l'arbre sont réunies. Ici les valeurs prédites semblent suivre la droite de régression $y = x$, avec une dispersion assez faible (~5), ce qui conforte l'efficacité de cet arbre.

Les résidus sont centrés et il y a homoscédasticité ce qui exprime une stabilité de l'arbre relative à l'évolution des valeurs prédites. On retrouve la structure en 'strates' précédante due au fait qu'on ne prédit que des valeurs en sortie d'arbre. L'écart type de 19 est tout de même un peu élevé comparé à la grandeur des intervales correspondants aux classes d'énergie qui sont d'environ 10.

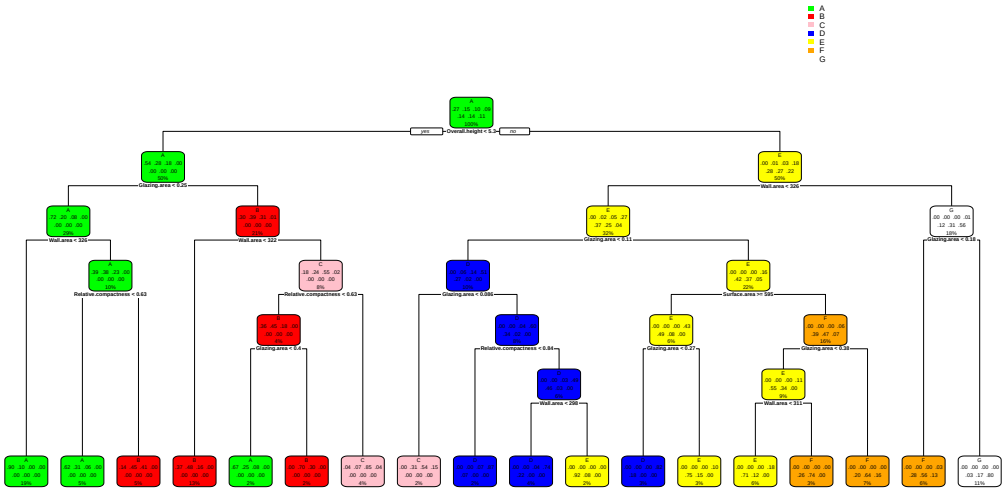
4.1.2 Arbre de Classification

On génère ici un arbre de classification afin de prédire la classe d'appartenance de l'énergie de la variable *Energy.efficiency*.

Nous faisons les même calculs que pour l'arbre de régression afin de trouver le critère de pénalité le plus petit possible.

On obtient un critère de pénalisation optimale $Cp_{min} = 0.0082198$ qui minimise au maximum l'erreur.

On le ré-injecte dans l'arbre :



4.1.3 Analyse des arbres :

Dans cette section, nous allons analyser le pouvoir de prédiction et l'erreur associés à nos deux arbres obtenus précédemment.

En s'aidant des seuils prédéfinis, on va classifier nos prédictions de notre régression dans les classes {A,B,C,D,E,F,G}.

Pour conclure, nous pouvons à présent établir les tables de contingence pour la classification et la régression:

```
##
## pred.treeq  A  B  C  D  E  F  G
##          A 33  4  4  0  0  0  0
##          B  8 14  5  0  0  0  0
##          C  0  1  7  5  0  0  0
##          D  0  0  1 10  4  0  0
##          E  0  0  0  6 14  2  0
##          F  0  0  0  2  5 13  3
##          G  0  0  0  0  0  3 10
```

```
##
## pred.treerClass  A  B  C  D  E  F  G
##               A 32  3  0  0  0  0  0
##               B  9 15  9  0  0  0  0
##               C  0  1  7  5  0  0  0
##               D  0  0  1 10  4  0  0
##               E  0  0  0  8 14  3  0
##               F  0  0  0  0  5 11  3
##               G  0  0  0  0  0  4 10
```

Pour l'arbre de régression, la proportion des bonnes prévisions est de 64.29%. En ce qui concerne l'arbre de classification, la proportion est de 65.58%.

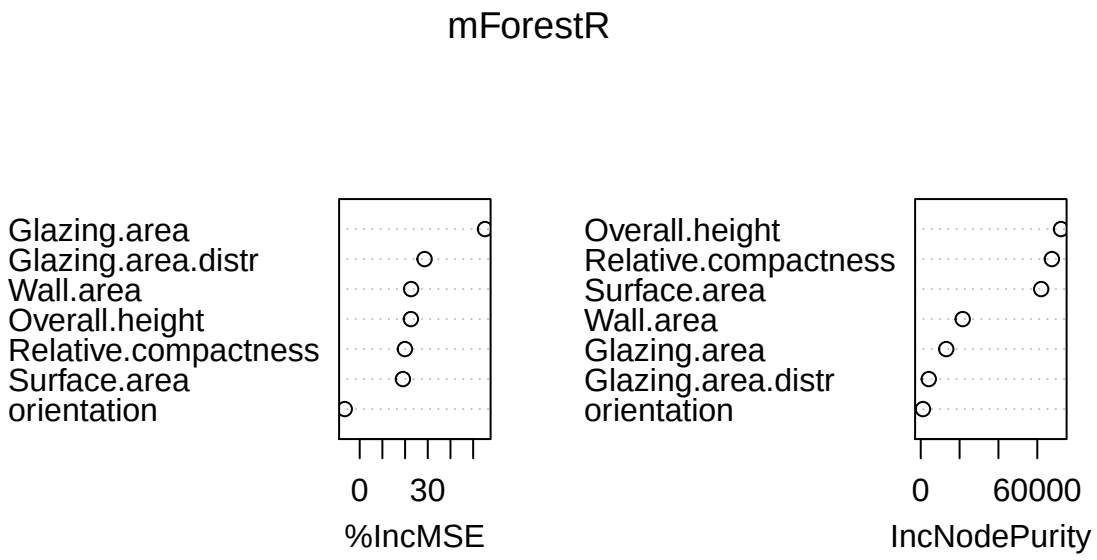
4.2 Random Forest

Afin d'étendre notre étude faite sur les arbres de décision, nous allons nous intéresser dans cette section aux forêts aléatoires. Comme pour les arbres, nous étudierons les forêts aléatoires de régression et de classification afin de travailler sur les sorties *Energy* et *Energy.efficiency*.

Une fois ces deux premiers points étudiés, nous veillerons à améliorer les performances de notre algorithme de Random Forest en choisissant le paramètre "mtry" optimal. Ce paramètre représente le nombre de variables échantillonnées aléatoirement comme candidats à chaque "split".

4.2.1 Forêt aléatoire de régression

Commençons par l'explication et la prédiction de la variable quantitative *Energy*.



A l'aide de ces deux graphiques, nous avons la possibilité de connaître, par deux méthodes différentes, quelles sont les variables qui impactent le plus notre modèle. Il est donc nécessaire de choisir judicieusement les variables qui se démarquent dans les deux méthodes.

Dans la première méthode `Increase MSE`, deux variables se démarquent : `Glazing.area` car elle est la plus précise et `orientation` car elle est la moins précise. Puis nous avons plusieurs variables qui sont quelques peu équivalentes.

La deuxième méthode `Increase Node Purity` distingue principalement trois variables : `Overall.height`, `Relative.compactness` et `Surface.area`.

Un modèle que nous pourrions proposé pour être en adéquation avec ces résultats serait le suivant :

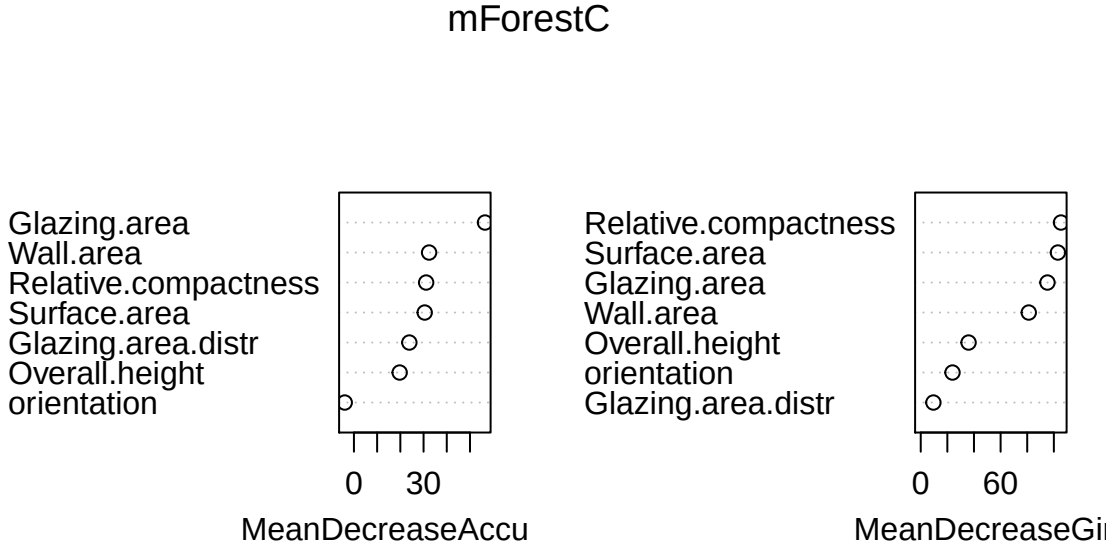
$$Energy \sim Glazing.area + Overall.height + Surface.area + Relative.compactness$$

Cet arbre de régression nous donne un pourcentage de bonnes prédictions de 70.13%.

4.2.2 Forêt aléatoire de classification

Voyons à présent si le fait d'avoir discrétiser la variable *Energy* en plusieurs classes énergétiques améliore le taux de bonnes prédictions.

```
##    A B  C  D  E  F  G class.error
## A 35 5  1  0  0  0  0   0.1463415
## B 10 7  2  0  0  0  0   0.6315789
## C  1 5 10  1  0  0  0   0.4117647
## D  0 0  3 11  8  1  0   0.5217391
## E  0 0  0  2 19  2  0   0.1739130
## F  0 0  0  0  3 10  5   0.4444444
## G  0 0  0  0  0  3 10   0.2307692
```



Nous pouvons remarquer que les deux graphiques ont le même "Top 4" mais dans un ordre différent. Nous pouvons donc nous limiter à ces 4 premières variables. L'algorithme de forêt aléatoire de classification nous permet donc d'écrire le modèle suivant :

$$Energy.efficiency \sim Glazing.area + Wall.area + Surface.area + Relative.compactness$$

Cette forêt aléatoire de classification nous donne un pourcentage de bonnes prédictions de 66.23%.

4.2.3 Optimisation des forêts

L'algorithme de Random Forest implémenté dans R possède des valeurs par défaut du paramètre "mtry" qui varient selon le choix de la méthode utilisée : classification ou régression. Nous verrons dans chaque partie

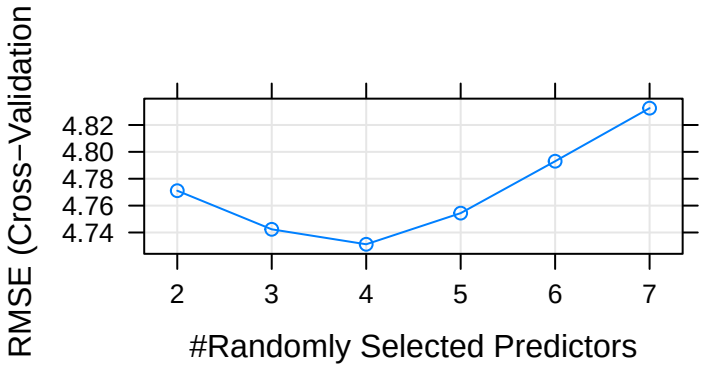
quelle est cette valeur et comment nous pouvons l'optimiser.

4.2.3.1 Régression

Lorsque nous faisons de la régression, la valeur par défaut choisie est $mtry = \frac{p}{3}$ où p est le nombre de variables de notre jeu de données. Dans notre étude, nous avons $p = 7$, donc étant donné que notre paramètre est un entier, on aurait par défaut : $mtry = 2$.

Calculons, par validation croisée, l'erreur quadratique moyenne RMSE selon les valeurs du mtry :

```
## note: only 6 unique complexity parameters in default grid. Truncating the grid to 6 .
```



L'objectif étant de minimiser l'erreur, nous pouvons remarquer sur la figure précédente que le mtry optimal est égal à 3.

Nous reconstruisons alors notre forêt de régression en précisant $mtry = 3$.

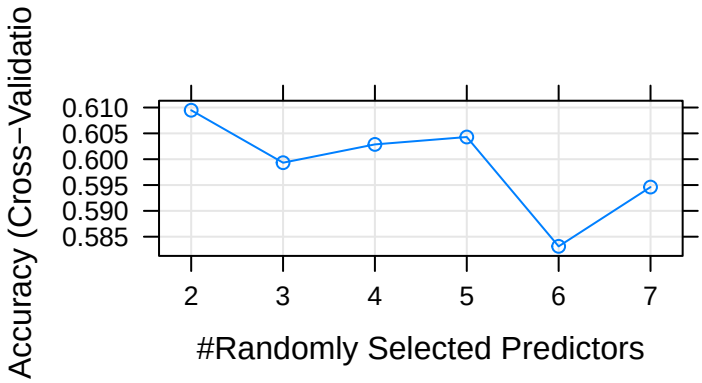
On trouve un pourcentage de bonnes prédictions de 71.43 %, ce qui est désormais notre meilleur prédicteur.

4.2.3.2 Classification

Lorsque nous faisons de la classification, la valeur par défaut choisie est $mtry = \sqrt{(p)}$. Nous aurions donc par défaut : $mtry = 3$.

Effectuons le même calcul que précédemment mais qui, pour une forêt de classification, nous donne la précision du modèle :

```
## note: only 6 unique complexity parameters in default grid. Truncating the grid to 6 .
```



Nous pouvons voir que la précision est maximale lorsque $mtry = 2$. Nous reconstruisons alors notre forêt avec cette nouvelle valeur du paramètre :

Cet arbre de classification avec amélioration du paramètre mtry nous donne un pourcentage de bonnes prédictions de 67.53% ce qui est légèrement mieux qu'avec le mtry par défaut.

4.3 Modèles non linéaires pour deux catégories : {A,B} et {C,D,E,F,G}

Nous avions vu dans la section "Analyse des données" que la variable *Energy.efficiency* peut être séparée en deux classes. Cherchons un sous-modèle qui repose sur ces deux classes.

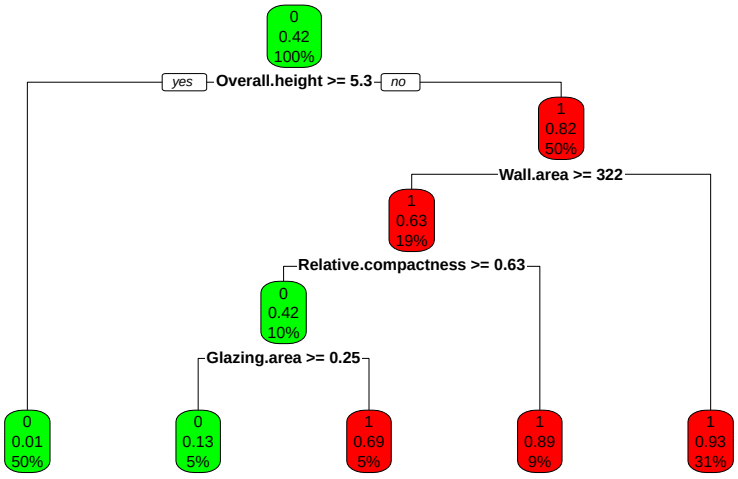
On sépare nos données en un ensemble de test et un ensemble d'entraînement.

On génère ici un arbre de classification afin de prédire la classe d'appartenance de l'énergie de la variable *Energy.efficiency* parmi les classes : + 1 = {A,B} + 2 = {C,D,E,F,G}

Nous faisons les même calculs que pour l'arbre de classification précédent afin de trouver le critère de pénalité le plus petit possible.

On obtient un critère de pénalisation optimal $Cp_{min} = 0.0166605$ qui minimise au maximum l'erreur.

On le ré-injecte dans l'arbre :



Nous avons ici une proportion de bonnes prévisions de 94.16%.

5 Conclusion

5.1 Récapitulatif des résultats de prédiction

	Classification en 2 classes	Classification en 7 classes	Régression (puis découpage)
Régression linéaire			61.69%
ANCOVA			59.74%
Régression logistique	91.02%		
Arbre	94.16%	65.58%	64.29%

	Classification en 2 classes	Classification en 7 classes	Régression (puis découpage)
Forêt aléatoire		66.23%	70.13%
Forêt aléatoire améliorée		67.53%	71.43%

5.2 Récapitulatif des modèles :

Régression linéaire :

$$Energy \sim Relative.compactness + Roof.area + Overall.height + Glazing.area$$

ANCOVA :

$$Energy \sim Glazing.area.distr + Roof.area + Glazing.area + Overall.height + Relative.compactness$$

Régréssion logistique : (*Energy.efficiency* binaire)

$$logit(\pi_i) = \theta_0 + \theta_1\ Relative.compactness_i + \theta_2\ Overall.height_i + \theta_3\ Glazing.area_i + \theta_4\ \mathbb{1}_{Glazing.area.distr=2}$$

Forêt aléatoire de régression :

$$Energy \sim Glazing.area + Overall.height + Surface.area + Relative.compactness$$

Forêt aléatoire de classification :

$$Energy.efficiency \sim Glazing.area + Wall.area + Surface.area + Relative.compactness$$

5.3 Lien entre les modèles et l'analyse de données

Dans notre analyse des données, nous avions relever certaines variables qui semblaient avoir un lien avec l'*Energy* et d'autres ne pas en avoir. Il est intéressant de comparer ces premières observations avec nos modèles obtenus. Nous avons conclu dans la première partie, que l'*Orientation* ne semblait pas avoir d'effet sur notre sortie. Cette première interprétation est en accord avec nos modèles, étant donné qu'aucun des modèles ne considère cette variable. De même, nous avions relevé l'impact des variables : *Overall.height*, *Relative.compactness* et *Surface.area*. La variable *Overall.height* est présente dans 4 modèles sur 5 et *Relative.compactness* dans tous nos modèles. Rappelons que *Surface.area* a été exprimée en fonction de *Wall.area* et *Roof.area*. On remarque qu'au moins une de ces trois variables de surface est présente dans 4 modèles. Il est aussi intéressant de voir que *Glazing.area* est présente dans tous les modèles, alors que notre analyse bidimensionnelle ne tirait aucune conclusion de cette variable.

De plus, lorsque nous avons effectué le clustering sur nos données, deux classes se sont principalement démarquées : {A,B} et {C,D,E,F,G}. Au cours de notre étude, nous avons effectué une régression logistique et un arbre sur cette nouvelle variable de sortie binaire. Nous pouvons remarquer sur le tableau ci-dessus que les meilleurs taux de prédictions obtenus dans les différents modèles sont avec cette nouvelle variable. Le clustering était donc efficace car nous pouvons prédire la bonne appartenance aux deux groupes dans plus de 90% des cas.

5.4 Conclusion générale

Au cours de ce projet nous avons donc cherché à construire un modèle de prédiction de l'impact énergétique des bâtiments en fonction de nombreuses de leurs caractéristiques. L'étude s'est divisée en deux axes : la régression et la classification. Au sein même de la classification on a séparé l'étude en deux parties : la classification en sept classes {A,B,C,D,E,F,G} et en deux classes ({A,B} et {C,D,E,F,G}). A la vue du tableau récapitulatif des performances des modèles, on a constaté que les modèles non-linéaires entraînent des résultats plus performants que les modèles linéaires. Et ainsi, après optimisation de nos modèles non-linéaires, on a finalement remarqué que procéder à de la régression puis découper en sept classes donnait de meilleurs résultats que de travailler directement avec les classes. En effet, on a abouti au final à une forêt aléatoire avec un taux de réussite de 75%.

Parallèlement on a étudié une classification en deux classes, et celle-ci nous a donné un taux de bonnes prédictions de 94%. On a ici donc un modèle très performant pour ce problème.

Pour conclure, on a donc réussi à construire des modèles qui permettent de classer avec 75% de réussite les bâtiments dans une des sept classes énergétiques, et avec 94% de réussite dans une des classes parmi {A,B} et {C,D,E,F,G}.

Ainsi, dans l'optique d'améliorer le DPE des bâtiments, ce projet permettra de prédire quelle sera l'empreinte énergétique d'un bâtiment qu'il faut construire ou rénover, en fonction de ses caractéristiques. Il sera alors possible de moduler ces caractéristiques, dans le but de réduire les consommations en énergie des habitants.