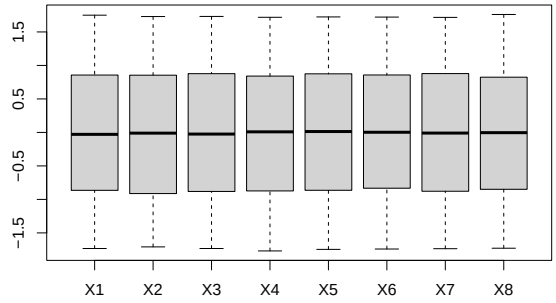
Robotics

`Robotics` dataset

Le jeu de données d'apprentissage de ce problème représente la cinématique du bras d'un robot. Les predicteurs ainsi la réponse données sont toutes de type numérique. Le but est de trouver la relation de la réponse `y` et les 8 prédicteurs `X1, X2, X3, X4, X5, X6, X7, X8`. Il s'agit bien d'un problème de régression.

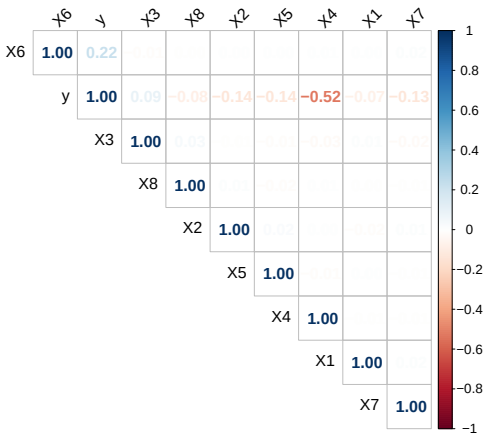
Analyse exploratoire

Dans un premier temps, on essaie de regarder la plage de toutesl es données.

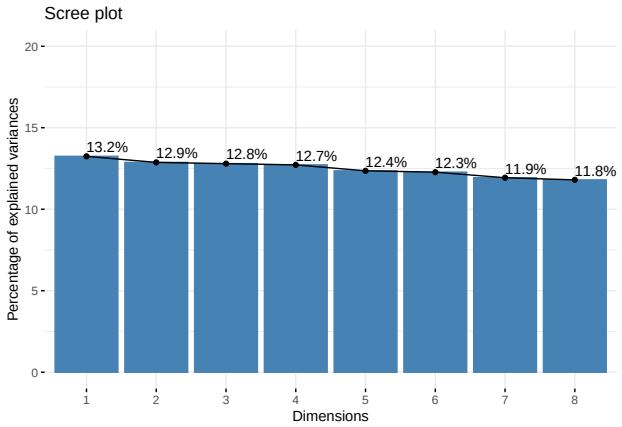


On observe que la plage de données est très homogène, ce qui nous donne la possibilité d'explorer nos données sans faire un scaling.

Ensuite on vérifie la corrélation entre les variables.



On constate qu'il existe pas de corrélations significatives entre les variables, seule de faibles corrélations entre la réponse et les prédicteurs. Ce graphe exclut le besoin d'enlever certaines variables puisqu'elles sont peu corrélées. On confirme cette observation en faisant une ACP



On constate avec ce graphe que la variance est expliquée par toutes les variables. Et on en déduit que toutes les méthodes concernant **Subset Selection** eront pas utiles vu que les variables devraient toutes être incluses.

Sélection de modèle

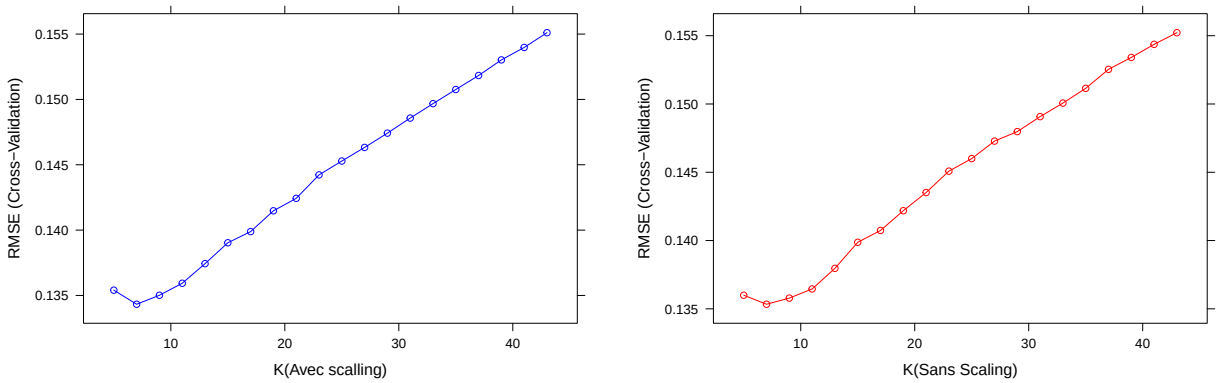
On test sur le jeu de données `Robotics` les méthodes suivantes, que l'on a vues en cours:

1. Modèle linéaire
2. La méthode des **KNN**
3. Les méthode de régularization **Ridge Regression, Lasso Regression, Elastic Regression**
4. Les méthode **Subset Selection**
5. La méthode des **Splines**
6. **SVM**
7. **L'Arbre de Décision**

Dans les différents modèles que l'on a utilisé, on a calculé l'erreur quadratique en appliquant la méthode **Cross validation** avec 10-folds. Dans certains cas, on l'a également utilisé pour trouver l'hyper-paramètre.

Pour commencer on a fait une linéaire régression. En regardant le `Summary`, on a observé que toutes les variables étaient significatives, c'est ce à quoi on s'attend puisque elles sont peu corrélées. Néanmoins, la `Adjusted R-squared` est inférieure à 0.5, ce qui rélève qu'elles n'expliquent pas beaucoup la volatilité de la variable `y` à prédire. Le modèle linéaire donne une erreur en moyenne 0.04185.

On a ensuite effectué une régression en utilisant **KNN**. En utilisant **Cross validation**, on a trouvé le meilleur nombre de voisin 7 pour les données avec/sans scaling. On obtient une erreur en moyenne 0.016 pour les deux modèles, mais il n'existe pas un grand écart entre le modèle avec scaling et celui sans scaling.



On a ensuite appluqué les méthodes de régularisation Ridge, Lasso et ElasticNet. La régression Ridge ne permet pas de séléctionner les prédicteurs et elle les inclut tous. La régression Lasso, quant à elle, permet de réduire le coefficient de certains prédicteurs à 0, elle est donc une méthode de sélection de variables. La méthode ElasticNet est comprise entre les deux dernière. Elle permet d'effectuer une sélection de modèle et réduire le coefficient des variables corrélées. On a estimé le paramètre de cette méthode en le faisant varier entre (0, 1).

SPLINES A FAIRE

On a appliqué de différentes méthodes concernant l'arbre des décision. Dans un premier temps, on a appliqué l'arbre de décicion classique. On pourrait savoir l'importance de chaque prédicteur en affichant l'arbre complet. Pour que les résultats soient plus précises, on a ainsi utilisé **Prune** et **Bagging**. L'arbre de décision reste inchangé avec **Prune**

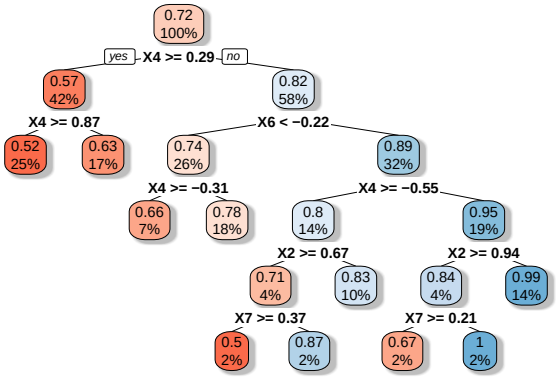


Figure 1: A gauche : l'arbre initial; A droite : l'arbre élagué

Enfin, on a appliqué le modèle SVM en utilisant de différent noyaux. Les noyaux le plus performants qu'on a observé sont le noyau laplacien et le noyeau gaussien. Pour estimer la valeur de l'hyper-paramètre C, on a utilisé **Cross validation**

Modèle retenu

```
boxplot(err.lin.mse, CV.noscale, err.las.mse, err.rid.mse, err.ela.mse, err.tree.mse, err.tree.mse.prune
```

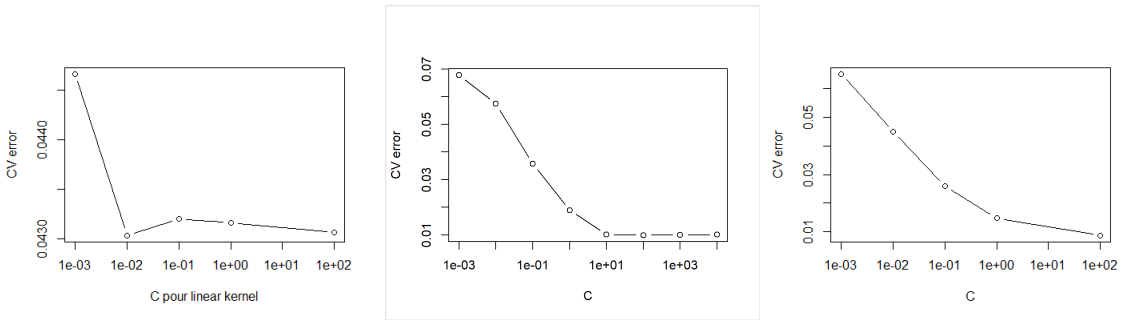


Figure 2: Le choix de l'hyper-paramètre pour Linear Kernel, Laplace Kernel, Gaussian Kernel