Projet : Analyse de données

Réalisé par: El Ghamraoui Hafsa-Jerrari Khaoula-Moutalattif Hind-Hattabi Rabie

1.Exploration des données explicatives

+On a 1030 observations sur 9 Variables

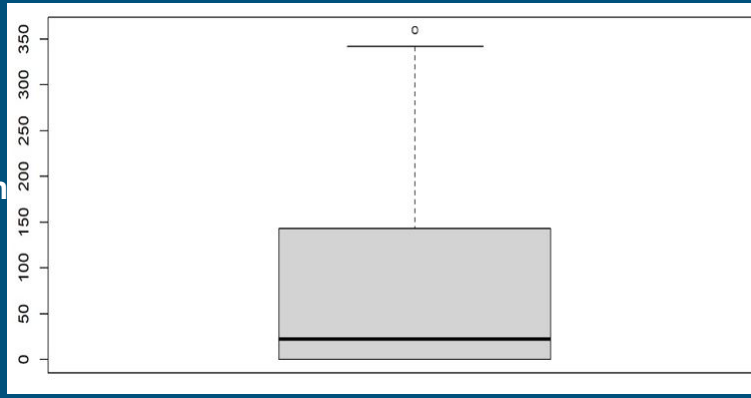
+On a considéré la variable « $Age » qualitative car elle possède le minimum de modalités par rapport aux autres variables.

Variables	Qualitatives	Quantitatives	Niveaux(as.factor)
Ciment		Oui	278
Laitier de haut fourneau		Oui	185
Cendres volantes		Oui	156
Eau		Oui	195
Superplastifiant		Oui	111
Granulats grossiers		Oui	284
Granulats fins		Oui	302
Age	oui		14
Resistance		oui	845

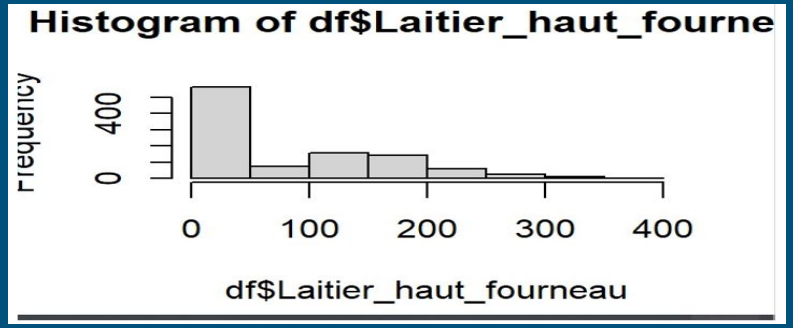
```
> df$Ciment=as.factor(df$Ciment)
> df$Laitier_haut_fourneau=as.factor(df$Laitier_haut_fourneau)
> df$Cendres_volantes=as.factor(df$Cendres_volantes)
> df$Eau=as.factor(df$Eau)
> df$Superplastifiant=as.factor(df$Superplastifiant)
> df$Granulats_grossiers=as.factor(df$Granulats_grossiers)
> df$Granulats_fins=as.factor(df$Granulats_fins)
> df$Age=as.factor(df$Age)
> df$Resistance=as.factor(df$Resistance)
> srt(df)
```

Analyser et représenter les distributions de la variable

$Laitier_haut_fourneau :



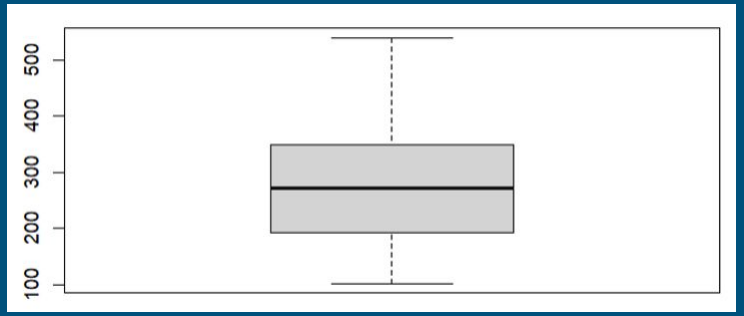
1. On remarque que la médiane est très écartée du centre de la boite du boxplot, donc les données sont hétérogènes, et 50% des observations on un laitier haut fourneau < 22 et 50% des observations ont un laitier haut fourneau > 22.On remarque aussi l'existence de valeurs aberrantes représentées par les points hors extrémités, on doit vérifier s'il s'agit de données valides ou non.

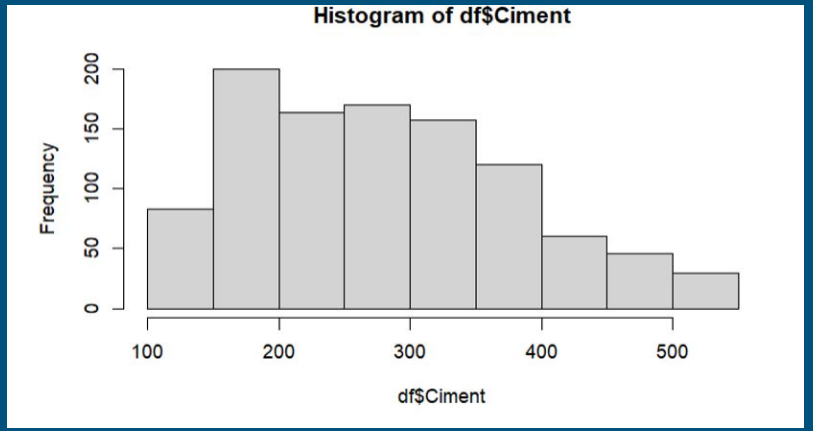


$Ciment

1. On remarque que la médiane est située à peu près au centre de la boite, donc les données sont symétriques c'est-à-dire, 50% des échantillons ont une quantité de ciment inférieure à 272.9 et 50% des échantillons ont une quantité de ciment supérieure à 272.9.

   On remarque aussi l'absence de valeurs aberrantes.

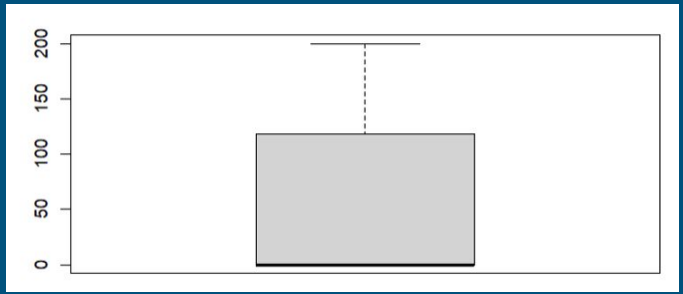




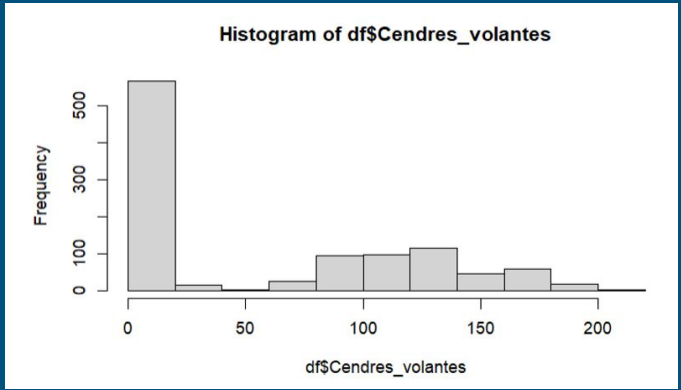
Cet histogramme représente la fréquence de chaque tranche de ciment avec un pas de 100, tel que la tranche entre 150 et 200 est la plus dominante avec une fréquence de 200 et donc un pourcentage de 19.41%

$Cendres_volantes

1 On remarque que la médiane est très écartée du centre de la boite donc les données sont extrêmement hétérogènes. En plus absence de valeurs aberrantes



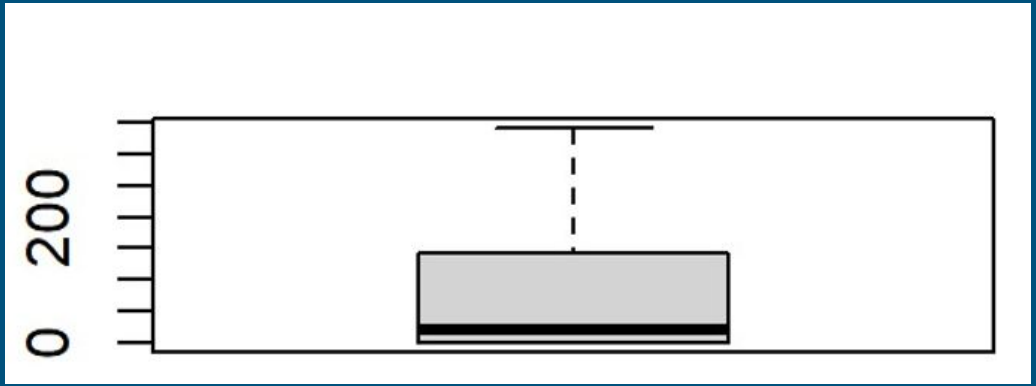
2 On remarque que les échantillons qui ont une composition de cendres volantes inférieure à 25 sont les plus dominantes.



2-Suppression de valeurs aberrantes

```
> Q1 <- quantile(df$Laitier_haut_fourneau, .25)
> Q3 <- quantile(df$Laitier_haut_fourneau, .75)
> IQR <- IQR(df$Laitier_haut_fourneau)
>
> #only keep rows in dataframe that have values within 1.5*IQR of Q1 a
nd Q3
> no_outliers <- subset(df$Laitier_haut_fourneau, df$Laitier_haut_four
neau> (Q1 - 1.5*IQR) & df$Laitier_haut_fourneau< (Q3 + 1.5*IQR))
>
> #view row and column count of new data frame
> dim(no_outliers)
NULL
> boxplot(no_outliers)
```

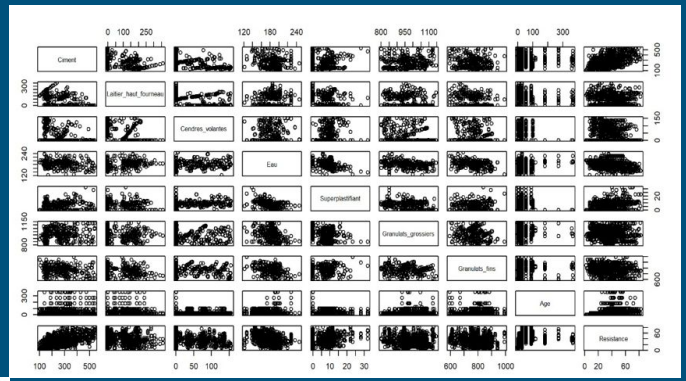
Vérification : affichage du boxplot de Laitier haut fourneau

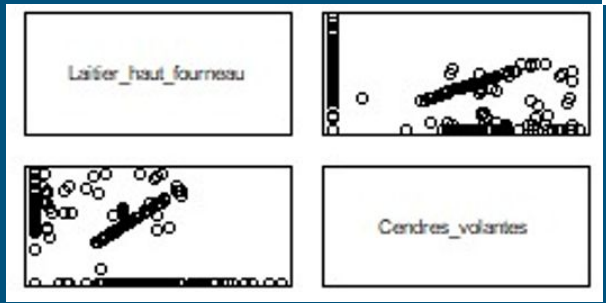


3-Description des liaisons entre les variables explicatives

1-On va croiser les jeux de données deux à deux à l'aide de « pairs », on obtient les résultats

chaque case représente le nuage de point après le croisement de deux variables, on remarque que le nuage de point « Laitier-haut-fourneau » et « Cendres-volantes » font apparaître une relation affine positive forte





+ Après on utilise la matrice de corrélation pour visualiser la relation entre ces deux variables :

+ Puis on diagonalise la matrice de corrélation :

```
                Ciment Laitier_haut_fourneau Cendres_volantes        Eau Superplastifiant
Ciment       1.00000000          -0.27521591     -0.397467341 -0.08158675       0.09238617
> eigen(autos.cor)
eigen() decomposition
$values
[1] 4.42085806 0.85606229 0.37306608 0.21392209 0.09280121 0.04329027

$vectors
           [,1]       [,2]        [,3]       [,4]       [,5]        [,6]
[1,] -0.4249360  0.1241911 -0.35361252  0.8077865 -0.1515800 -0.05889517
[2,] -0.4217944  0.4157739 -0.18492049 -0.3577920  0.2937346 -0.63303302
[3,] -0.4214599 -0.4118177  0.06763394 -0.2797523 -0.7305690 -0.19029153
[4,] -0.3869222 -0.4460870  0.60486812  0.2115694  0.4781901 -0.10956624
[5,] -0.4305120 -0.2426758 -0.48439601 -0.3017114  0.3045584  0.58081220
[6,] -0.3589443  0.6198626  0.48547226 -0.0735743 -0.1886551  0.45852167
```

+La diagonalisation de la matrice de corrélation a donné 9 valeurs propres écrites en ordre décroissant en haut avant la matrice, la somme de ces valeurs propre est 8.89 presque égale à 9 qui est la trace de la matrice de corrélation et qui représente la quantité totale d'information de la matrice de corrélation. Donc on peut dire qu'il n'y a pas eu de perte d'information lors de la diagonalisation.

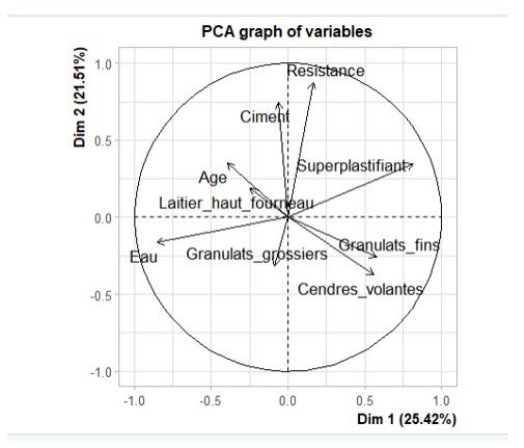
Ceci permet de réduire les dimensions de la matrice de corrélation pour avoir l'image la plus fidèle et donc selon les nouveaux vecteurs propres dont la valeur la plus importante est 2.28.

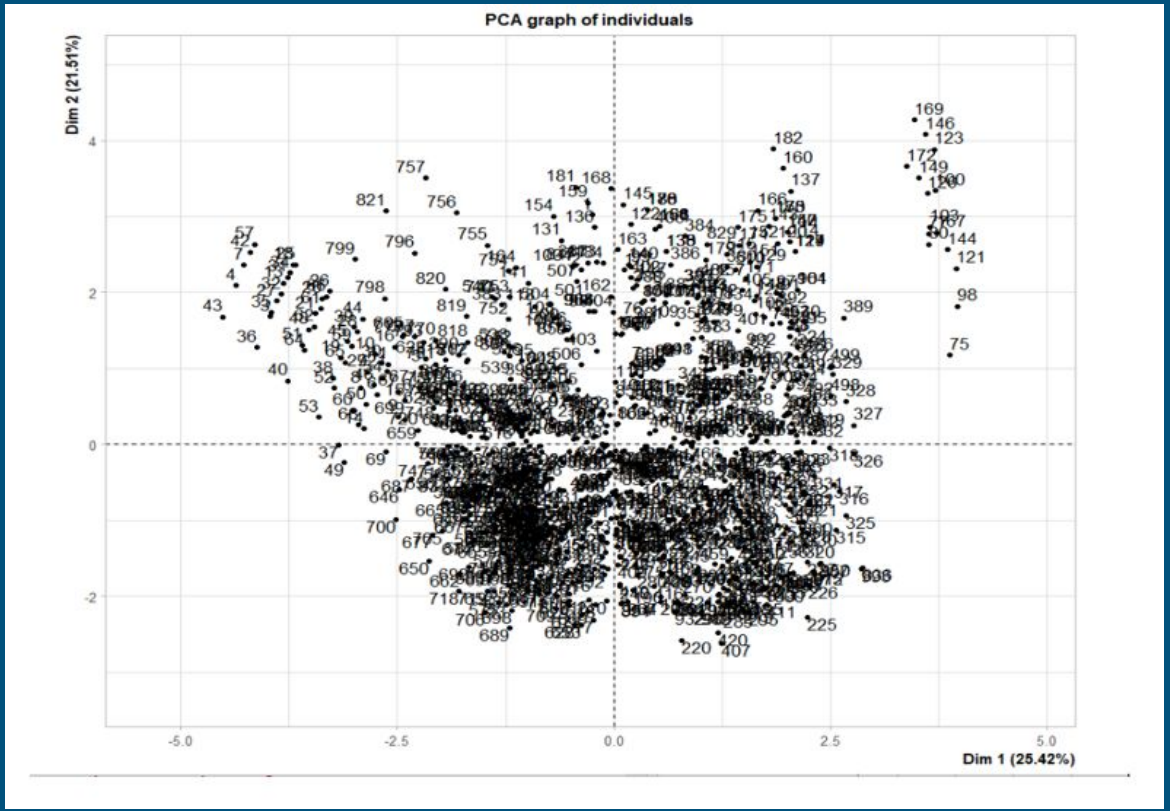
```
eigen() decomposition
$values
[1] 2.28789979 1.93634359 1.40889177 1.04282468 1.01415885
[6] 0.84737121 0.28685606 0.14690015 0.02875389

$vectors
            [,1]       [,2]        [,3]        [,4]
 [1,]  0.04132675  0.5364860  0.359693365  0.30981334
 [2,]  0.16304274  0.1361477 -0.699055619 -0.07599349
 [3,] -0.36989956 -0.2682705  0.019804327 -0.60068966
 [4,]  0.56402186 -0.1184577 -0.120189288 -0.04721492
 [5,] -0.53601839  0.2482488 -0.187966815 -0.16592948
 [6,]  0.06027386 -0.2247610  0.549480030 -0.22142114
 [7,] -0.38176315 -0.1868602  0.001258126  0.52753184
 [8,]  0.26194127  0.2517681  0.169505840 -0.35998236
 [9,] -0.10704127  0.6302352  0.033325802 -0.22526941
              [,5]        [,6]        [,7]        [,8]
 [1,] -0.0544413593 -0.38982482 -0.133851817 -0.2982539
 [2,] -0.3626502265  0.27015632  0.004947988 -0.2287693
 [3,]  0.2272376403 -0.32038502  0.247325097 -0.2553472
 [4,]  0.2960917614 -0.30601602 -0.010089669  0.5856528
 [5,] -0.0370769171 -0.08265085 -0.614074852  0.4473323
 [6,] -0.5456748109  0.34766017 -0.059695902  0.2429397
```

Et selon le graphe de PCA de la représentation des variables :

On peut garder les deux dimensions les plus dominantes.





4-les hypothèses d’application de la régression linéaire multiple.

1- -l’existence d’une variable continue dépendante.

2-l’existence d’au moins deux variables indépendantes soient continues soit catégoriques.

3-absence de valeurs aberrantes.

4-une relation linéaire entre la variable dépendante et les autres variables indépendantes.

5-avoir une distribution normale des résidus.

6-ne pas avoir une grande corrélation entre les variables indépendantes.

5-Modèle de régression :

Avant de créer le modèle de régression on a divisé les données en deux sous-ensembles : un ensemble « train » sur lequel le modèle sera appliqué et un ensemble « test » qu'on va utiliser pour tester le modèle et valider sa qualité.

Pour créer le modèle on a utilisé la commande « lm » sur la partie « train » des données.

Intercept représente la valeur minimale de la Résistance dans le cas où toutes les variables sont nulles ou absentes, dans ce cas Intercept=-30.55

```
> train<-df[0:721,]
> dim(train)
[1] 721   9
> test<-df[722:1030,]
> dim(test)
[1] 309   9
> lmModel <- lm(Resistance ~ . , data = train)
> # Printing the model object
> print(lmModel)

Call:
lm(formula = Resistance ~ ., data = train)

Coefficients:
          (Intercept)                 Ciment  Laitier_haut_fourneau
            -30.55937                0.11531                0.09646
     Cendres_volantes                    Eau       Superplastifiant
              0.09033               -0.14475                0.38969
  Granulats_grossiers         Granulats_fins                    Age
              0.02801                0.01664                0.12347
```

6-Valider la qualité de la régression.

Pour s'assurer que notre modèle est de qualité, on va tester sa performance sur les données de test. Pour dire que notre modèle est de qualité on doit avoir des résultats très proches ou même identiques entre Résistance et la Resistance prédite.

```
> # Predicting Resistance in test dataset
> test$PreditedResistance <- predict(lmModel, test)
> # Priting top 6 rows of actual and predited price
> head(test[ , c("Resistance", "PreditedResistance")])
    Resistance PreditedResistance
722      11.85           19.92749
723      17.24           20.42135
724      27.83           23.01414
725      35.76           30.66905
726      38.70           34.37303
727      14.31           22.56339
>
```

+ On remarque qu’il y a une différence notable entre la Résistance et la Résistance prédite, donc le modèle n’est pas applicable.

+ Pour améliorer le modèle, on peut utiliser la fonction « step() » pour nettoyer le modèle en supprimant des coefficients sans que la régression perde sa qualité.