le cnam
école d'ingénieur·e·s

Rapport Diamonds

Arthur BARIBEAUD, Etienne MONTHIEUX

Introduction

L'analyse que nous avons réalisé s'est portée sur l'étude de Diamants. Pour ce faire, nous avons utilisé un jeu de données composé de 54 000 individus (des diamants) et pour lesquels nous avions connaissance de 10 variables d'études.

```
display(diamonds.shape)
>> (53940, 10)
```

```
diamonds.head(4)
```

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63

L'objectif de ce projet a ainsi été, dans un premier temps, d'étudier les différentes données en notre possession, puis de tenter de comprendre les variations de la variable "price" en fonction des autres variables.

Première partie : Étude des données

Avant de pouvoir essayer de trouver des corrélations expliquant le prix d'un diamant, il a donc été nécessaire d'étudier les données en notre possession. La première chose qu'il est ainsi possible de faire est d'étudier la nature de nos variables. Il est ainsi possible de classifier nos variables en deux catégories : qualitative et quantitative.

Variable quantitative :
Carat : Poids du diamant
Depth: Pourcentage de profondeur du diamant
Table: Taille de la partie supérieure du diamant
Price: Prix du diamant (en $)
x: Longueur
y: Largeur
z: Profondeur

Variable qualitative :
Cut : Qualité de la coupe du diamant
Color : Couleur du diamant
Clarity : Clarté du diamant

Nous nous retrouvons dès lors avec deux familles de variables à étudier et pour lesquels il est nécessaire d'effectuer une analyse différente. Pour ce faire, deux procédures d'analyse ont été développé afin d'en apprendre plus sur chaque variable.

Étude des variables quantitatives

La procédure dédiée à l'analyse des variables quantitative permet ainsi de récupérer 3 types d'informations et prend en paramètre le nom de la variable à étudier :

```
def descQuanti(variable):
    """
        Description statistique de la variable quantitative passée
         en paramètre de la procèdure

        Un premier tableau sur des indicateurs classiques : mean, std, ...

        Un histogramme de la distribution de la variable avec une
        comparaison à une Gaussienne + calcule de la p-valeur

        Une boîte à moustache pour étudier le profil de la variable

        :param variable: Nom de la variable quantitative à étudier
        :type variable: str
    """
```

Premièrement, une analyse statistique permet de récupérer des indicateurs classiques tel que la moyenne, l'écart type ainsi que les écarts inter-quartiles. Ces indicateurs sont automatiquement calculés et récupérés à l'aide de la méthode *describe()* de la librairie pandas.

```
#Statistique classique
print("Information sur la variable %s : " % variable)
describe = diamonds[variable].describe()
df = pd.DataFrame(describe)
display(df.T)
```

```
Information sur la variable carat :
```

	count	mean	std	min	25%	50%	75%	max
carat	53940.0	0.79794	0.474011	0.2	0.4	0.7	1.04	5.01

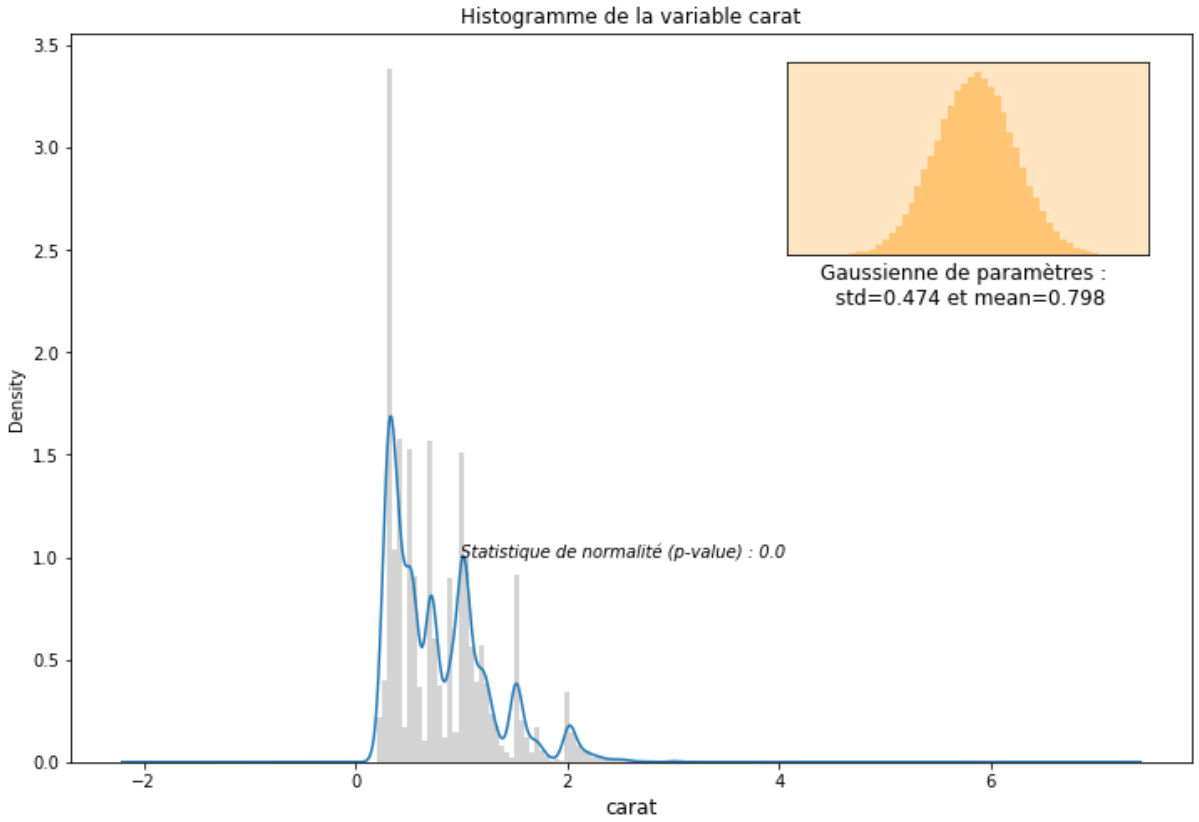
Deuxièmement, une analyse de la répartition des valeurs est mise en place à l'aide, notamment, d'une représentation graphique réalisée à l'aide de la librairie Matplotlib. Il a ainsi été mis en place deux histogrammes superposés pour observer la répartition des valeurs par bins pour le premier et la répartition sous forme de densité pour le second.

```
# Figure 1 : Histogramme
  plt.figure(figsize=(12,8))

  # Histogramme en bar
  diamonds[variable].plot(kind = "hist",
                          density = True,
                          color = "lightgrey",
                          bins=100)
  # Densité
  diamonds[variable].plot(kind = "kde")

  plt.title("Histogramme de la variable %s" % variable)
  plt.xlabel(variable, fontsize=12)
```

Pour finir, il a été rajouté un dernier histogramme comparatif avec une gaussienne de paramètre égale à ceux de notre variable. Cette représentation permet de nous donner un ordre d'idée "graphique" si oui, ou non, notre variable semble suivre une loi normale. Enfin, et afin de compléter cet axe d'analyse d'une Gaussienne, un test de normalité est effectué.

```
# Test de normalité de notre variable
plt.text(1,
         1,
         'Statistique de normalité (p-value) : %s' % normaltest(diamonds[variable])[1],
         style='italic')


# Gaussian de paramètre mean et std de notre variable
gaussian = np.random.normal(describe["std"], describe["mean"], 54000)

# Figure 1.1 : Histogramme d'une Gaussienne de même paramètres que notre variable
plt.axes([0.62,0.65,0.25,0.20], facecolor="#ffe5c1");
plt.xticks([])
plt.yticks([])
plt.xlabel("Gaussienne de paramètres : \n std=%s et mean=%s" % (round(describe["std"],3),
           round(describe["mean"], 3)),
           fontsize=12)

plt.hist(gaussian,
         bins=50,
         color="#FFC575");
```

Ce deuxième axe d'analyse nous permet ainsi d'obtenir le graphique ci-dessous :



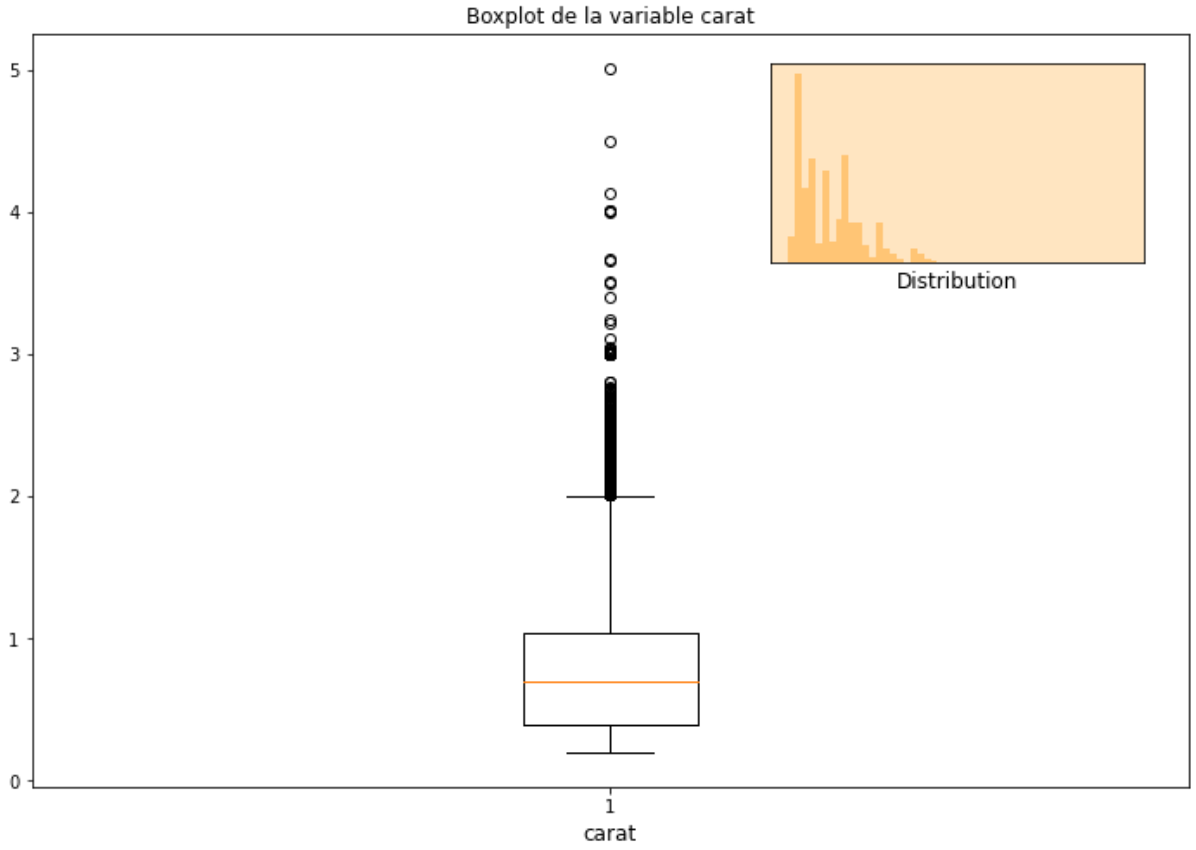
Enfin, le troisième axe d'analyse s'est porté sur l'étude du profil de la variable. Pour ce faire, il a été réalisé une boîte à moustache permettant d'observer, d'une manière différente à notre histogramme, la répartition de nos différentes valeurs.

```
# Figure 2 : BoxPlot
plt.figure(figsize=(12,8))

# BoxPlot de notre variable
plt.boxplot(diamonds[variable])
plt.xlabel(variable, fontsize=12)
plt.title("Boxplot de la variable %s" % variable)
```

Là encore, et afin d'améliorer la lisibilité du graphique, l'histogramme de notre variable est rajouté en haut à droite du graphique :

```
# Figure 2.1 : Histogramme de la variable
plt.axes([0.62,0.65,0.25,0.20], facecolor="#ffe5c1");
plt.xticks([])
plt.yticks([])
plt.xlabel('Distribution', fontsize=12)
plt.hist(diamonds[variable], bins=50, color="#FFC575")
```

Cette troisième et dernière analyse de nos données quantitative nous permet ainsi d'obtenir le graphique ci-dessous :



Étude des variables qualitative

La procédure dédiée à l'analyse des variables qualitative, qui prend en paramètre le nom de la variable à étudier, a ensuite été mise en place afin de récupérer 2 types d'informations :

```
def descQuali(variable):
    """
        Description statistique de la variable qualitative
        passée en paramètre de la procèdure

        Un premier tableau représentant la table d'effectif
        et de pourcentage de la variable
        Un barplot de la réprensativité de chaque modalitée de la variable

        :param variable: Nom de la variable qualitative à étudier
        :type variable: str
    """
```

Le premier axe d'analyse est ainsi effectué sur les différentes modalités de la variable. Étant dans un contexte qualitatif, il est en effet possible d'avoir une vision macro de la répartition des différentes modalités (dénombrés et finis). Pour ce faire, il a été réalisé un tableau d'effectif et de pourcentage permettant d'observer cette répartition dans les modalités :

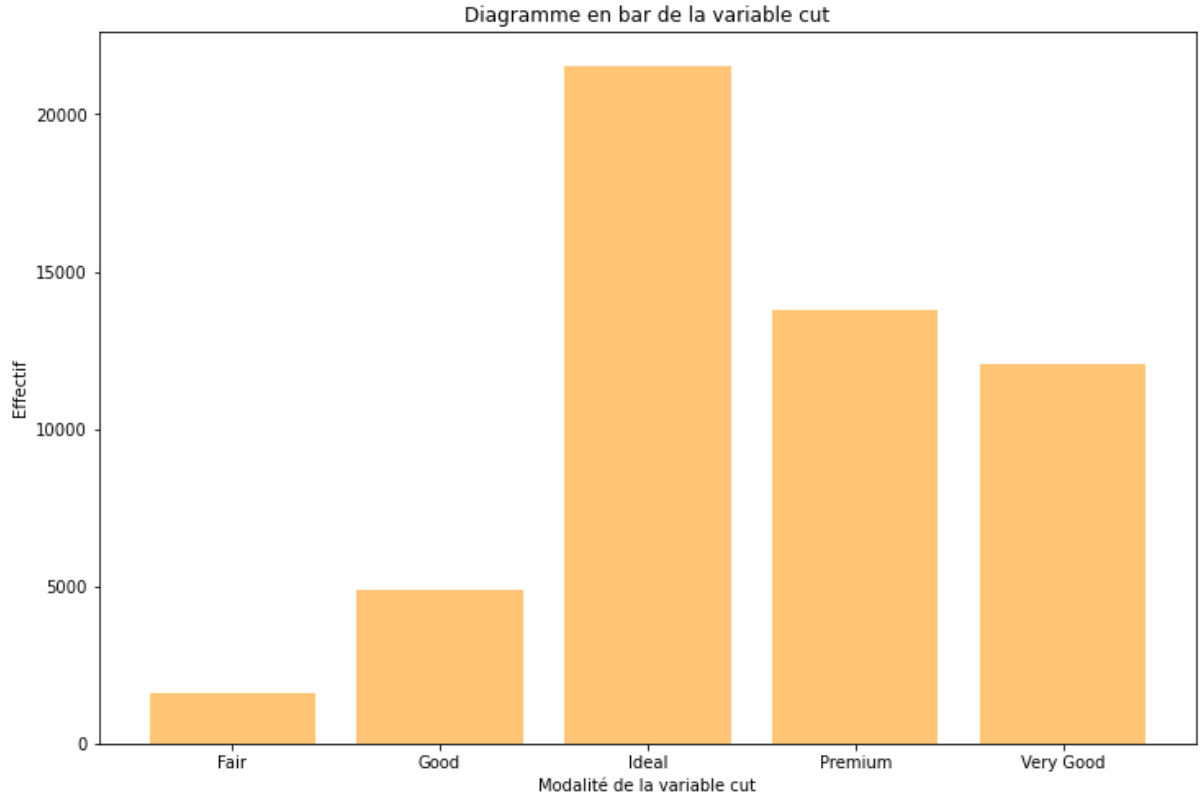
```
# Table d'effectif et de pourcentage
eff_table = pd.DataFrame(diamonds.groupby(variable).count()[diamonds.columns[0]])
eff_table.columns = ["count"]
effe_table_pivot = eff_table.T

percentage = [modality/sum(eff_table["count"].tolist()) for modality in eff_table["count"].tolist()]

effe_table_pivot.loc[len(effe_table_pivot)] = percentage

display(effe_table_pivot)
```

Le tableau affiché permet ainsi de rassembler à la fois la répartition d'effectif et de pourcentage tel que présenté ci-dessous :

cut	Fair	Good	Ideal	Premium	Very Good
count	1610.00	4906.0	21551.00	13791.00	12082.0
1	2.98	9.1	39.95	25.57	22.4

La deuxième analyse portée sur les données qualitative à ensuite été la mise en place d'un diagramme en bâton pour observer graphiquement la table d'effectif de la variable pour chaque modalité :

```
# Figure 1 : Barplot de la représentativité des modalitées de la variables
plt.figure(figsize=(12,8))
plt.bar(eff_table["count"].index.tolist(), eff_table["count"].tolist(), color="#FFC575")
plt.title("Diagramme en bar de la variable %s" % variable)

plt.xlabel("Modalité de la variable %s" % variable)
plt.ylabel("Effectif")
```



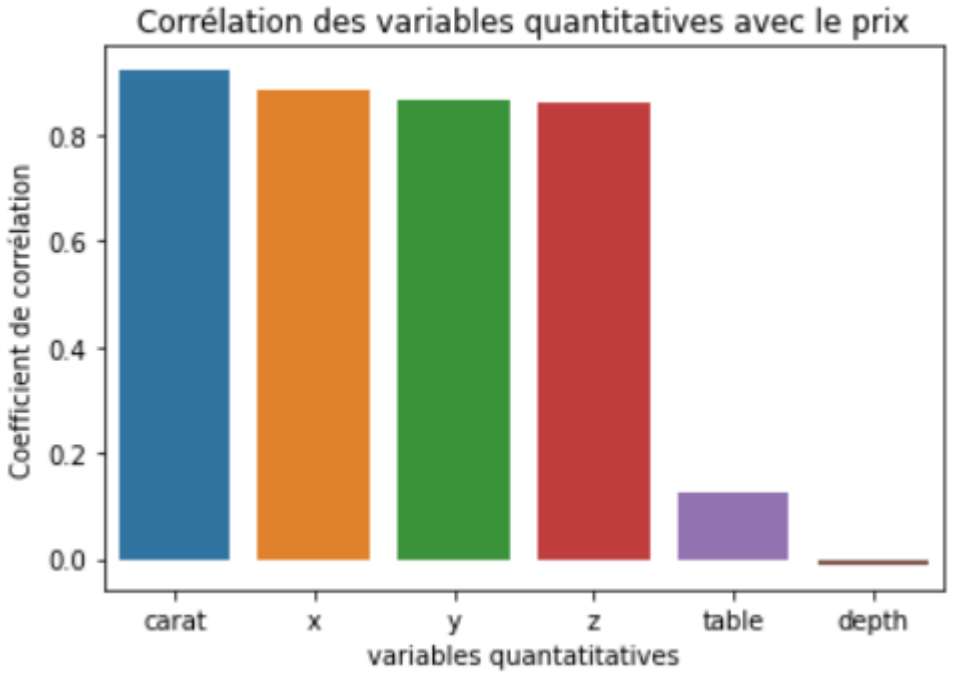
Seconde Partie : Influence des variables sur le prix

Dans cette partie, nous cherchons à trouver quelles sont les variables ou les classes au sein des variables qui influencent le prix des diamants :

Influence des variables quantitatives

Pour connaitre le lien entre les variables quantitatives et le prix, nous regardons la corrélation entre les variables et nous regardons la répartition de ces variables en fonction du prix.

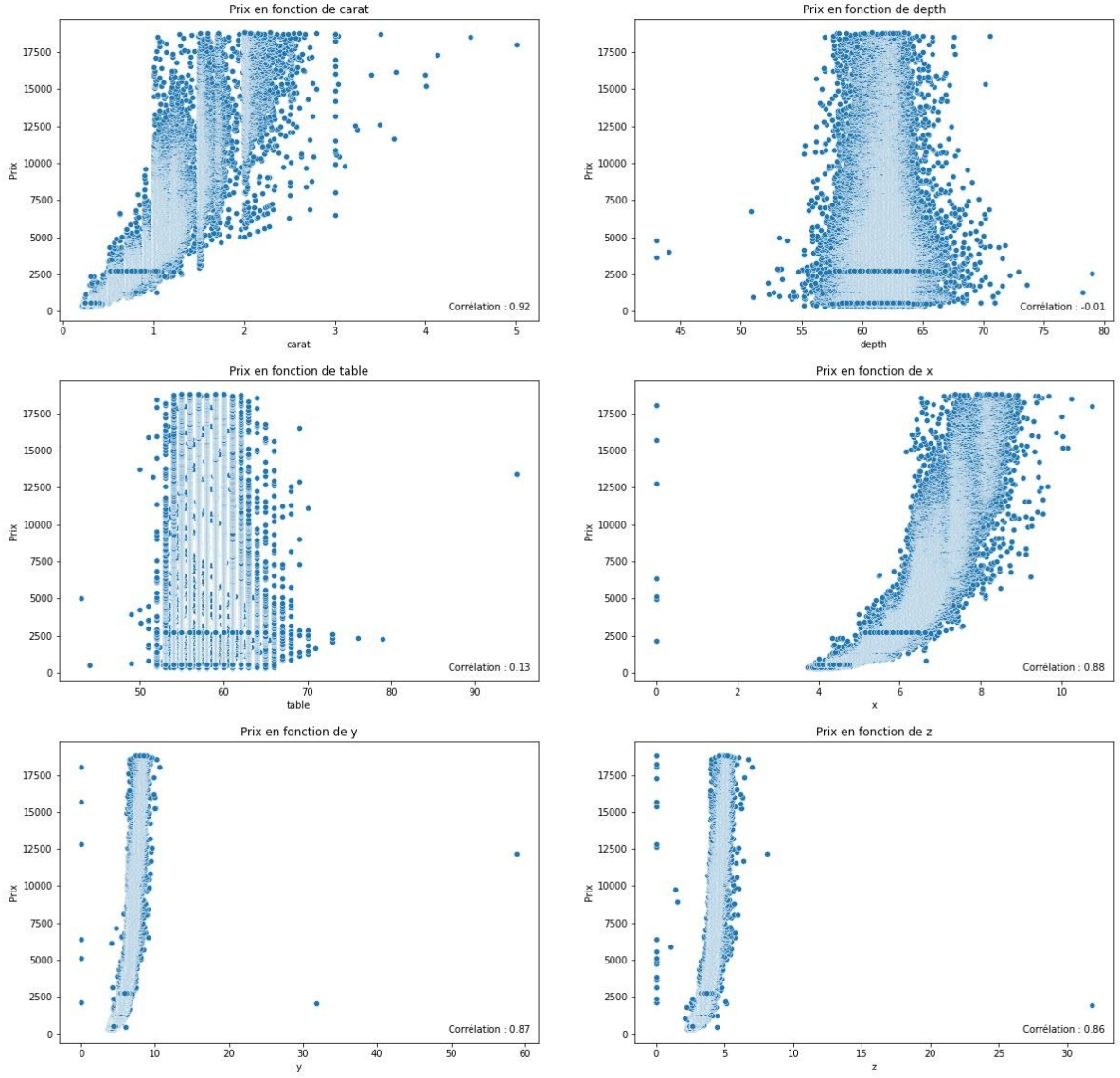
```
Corrélation du prix avec les autres variables quantitatives :

carat    0.921591
x        0.884435
y        0.865421
z        0.861249
table    0.127134
depth   -0.010647
```



Le tableau et le graphique ont été générés avec le code suivant :

```python
# Corrélation entre le prix et les autres variables
corr = diamonds.corr()

# Je retire la corrélation prix/prix
corr_price = corr["price"].sort_values(ascending=False).drop("price", axis=0)

print("Corrélation du prix avec les autres variables quantitatives :\n")

# Affichage de la table de corrélation
display(corr_price)

print("\nLes variables les plus corrélées avec le prix sont le carat, et les mesures x, y et z\n")

# graphique en barres de corr
p = sns.barplot(x=corr_price.index, y=corr_price.values);
p.set(title='Corrélation des variables quantitatives avec le prix'
      , xlabel = "variables quantatitatives"
      , ylabel = "Coefficient de corrélation");
```

Matrice de répartition des variables quantitatives en fonction du prix



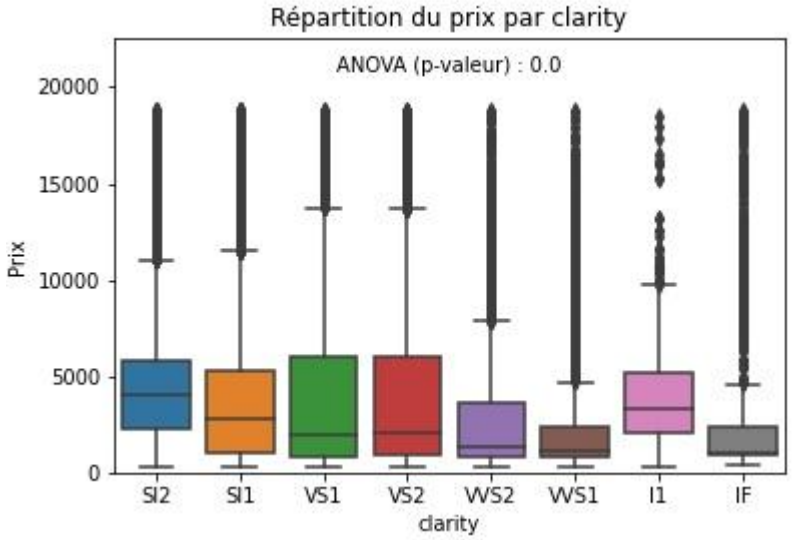
Pour générer cette matrice, nous avons créé le code suivant :

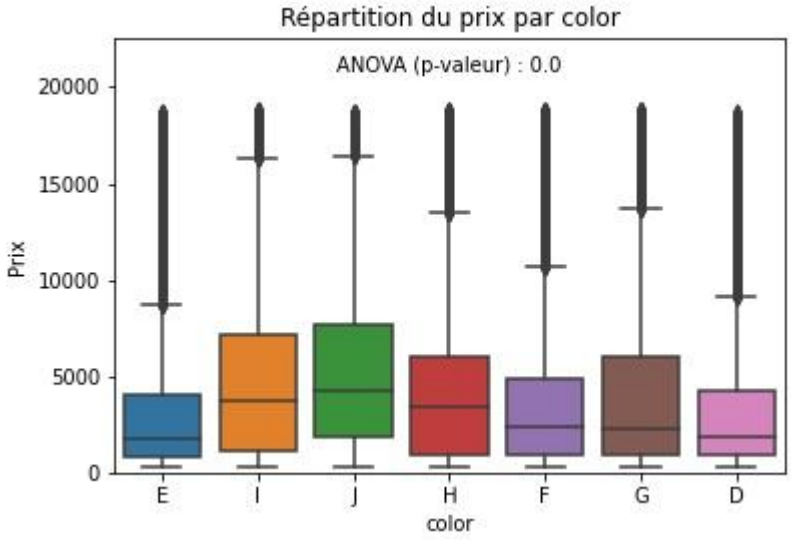
```
# tableau de graphiques scatterplot des variables quantitatives en fonction du prix

# Figure de dimension 3,2
fig, ax = plt.subplots(3,2, figsize=(20,20));

# Sélection des variables qualitatives
variables_quantitatives = diamonds.select_dtypes(include=['int64', 'float64']).columns.drop("price")

# Compteur pour la variable à sélectionner
var_i = 0

# Je parcours la figure
for i in range(3):
    for j in range(2):

        # Je récupère la variabe
        var = variables_quantitatives[var_i]

        # Je place le nuage de points dans la matrice
        p = sns.scatterplot(x=var, y="price", data=diamonds, ax=ax[i,j]);
        p.set(title="Prix en fonction de {}".format(var), xlabel=var, ylabel="Prix");
        anc = AnchoredText("Corrélation : {}".format(round(corr[var]["price"], 2)), loc="lower right", frameon=False)
        ax[i,j].add_artist(anc)

        # J'incrémente le compteur pour la variable suivante
        var_i = var_i + 1
```

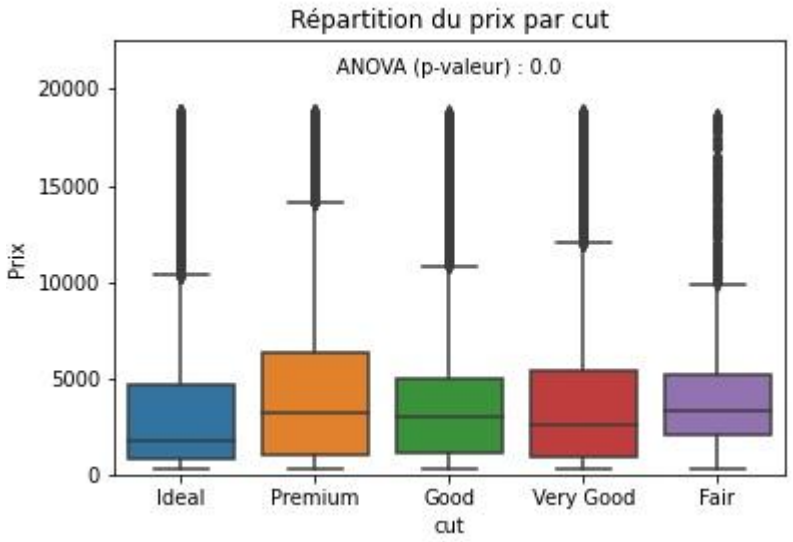
Les variables les plus corrélés avec le prix sont donc le carat, et les mesures x, y et z. Nous en concluons que le poids et la taille du diamant et son les variables qui influencent le plus son prix. Plus un diamant est lourd ou volumineux et plus il aura de la valeur.

Influence des classes des variables qualitatives

Pour connaitre le lien entre les variables qualitatives et le prix, nous regardons la distribution de ces variables en fonction du prix avec des boîtes à moustaches et une ANOVA.







Les ANOVAs ont étés réalisés grâce à ce code :

```
# Anova des variables quantitatives

# Sélection des variables qualitatives
diamonds_qual = diamonds.select_dtypes(include=['object'])

# Boucle sur chaque variable
for i in diamonds_qual :

    # Formatage, puis calcul de l'Anova entre le prix et la variable
    anova = [diamonds["price"][diamonds[i] == s] for s in list(diamonds[i].unique())]
    # F1 score et p-valeur
    F, p_val = stats.f_oneway(*anova)

    # Graphique
    p = sns.boxplot(x=i, y="price", data=diamonds);
    p.set(title='Répartition du prix par {}'.format(i), xlabel=i, ylabel="Prix");
    p.set_ylim(bottom=0, top=max(diamonds["price"]) + max(diamonds["price"]) * 0.2)
    anc = AnchoredText("ANOVA (p-valeur) : {}".format(round(p_val,2)), loc="upper center", frameon=False);
    p.axes.add_artist(anc);
    plt.savefig("Prix_{}.jpg".format(i))
    plt.show()
```

Nous observons que les ANOVA ont toutes une p-valeur de 0, nous pouvons conclure que toutes les classes au sein des variables influences le prix. L'importance des classes est visible avec les boîtes à moustaches.

Pour automatiser l'analyse des variables, nous avons créé deux fonctions *lienPrixQuanti()* et *lienPrixQuali()*.
Voici comment elles fonctionnent :

```
# lienPrixQuanti(variable) prend en paramètre une chaîne de charactères correspondant à une variable quantitative
# et retourne les coefficients de corrélation et un nuage de points de la variable quantitative en fonction du prix
def lienPrixQuanti(variable : str):

    # Matrice de corrélation

    corr = diamonds.corr()
    corr_variable = corr[variable].sort_values(ascending=False).drop(variable, axis=0)
    print("\nCorrélation de la variable {} avec les autres variables quantitatives :\n".format(variable))
    print(corr_variable)

    print("\n")

    # Graphique en barres des coefficients de corrélation

    p = sns.barplot(x=corr_variable.index, y=corr_variable.values);
    p.set(title='Corrélation de la variable {} avec les autres variables quantitatives'.format(variable),
            xlabel=variable, ylabel="Coefficient de corrélation");
    plt.show()

    print("\n")

    # Nuage de points de la variable en fonction du prix

    fig, ax = plt.subplots(1)

    p = sns.scatterplot(x=variable, y="price", data=diamonds);
    p.set(title="Nuage de points de la variable {} en fonction du prix".format(variable), xlabel=variable, ylabel="Prix");
    anc = AnchoredText("Corrélation : {}".format(round(corr[var]["price"], 2)), loc="lower right", frameon=False)
    ax.add_artist(anc)
    plt.show()
```

```
# lienPrixQuanti(variable) prend en paramètre une chaîne de charactères correspondant à une variable qualitative
# et retourne une Anova et une boîte à moustaches de la variable en fonction du prix
def lienPrixQuali(variable) :

    # ANOVA

    anova = [diamonds["price"][diamonds[variable] == s] for s in list(diamonds[variable].unique())]
    F, p_val = stats.f_oneway(*anova)


    # boîte à moustaches de la variable en fonction du prix

    p = sns.boxplot(x=variable, y="price", data=diamonds);
    p.set(title='Répartition du prix par '.format(variable), xlabel=variable, ylabel="Prix");
    p.set_ylim(bottom=0, top=max(diamonds["price"]) + max(diamonds["price"]) * 0.2);
    anc = AnchoredText("ANOVA (p-valeur) : {}".format(round(p_val,2)), loc="upper center", frameon=False);
    p.axes.add_artist(anc);
    plt.show()
```

La fonction *help()* de chacune des fonctions retourne :

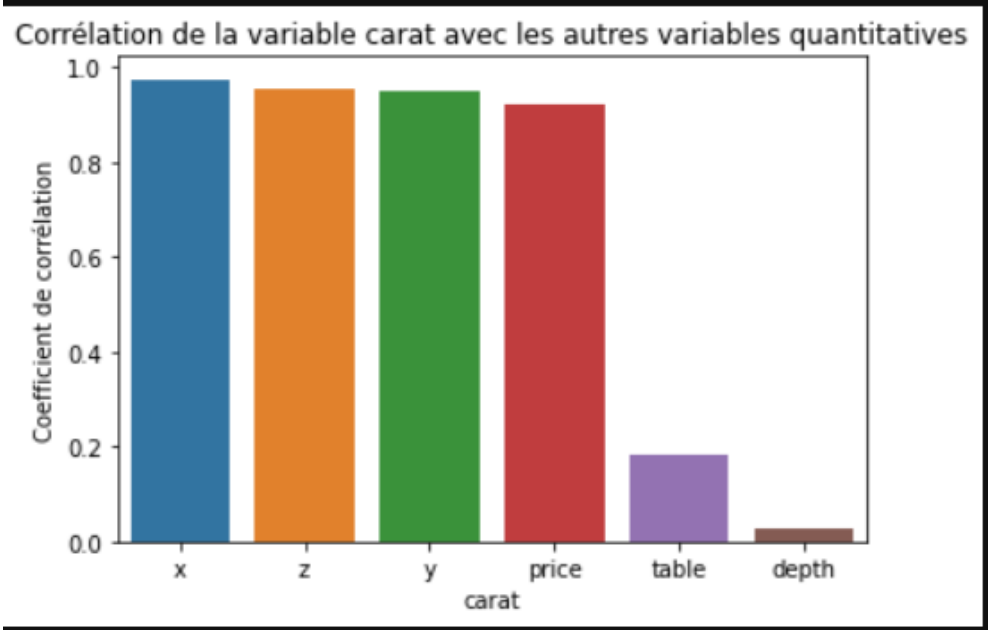
```
Help on function lienPrixQuanti in module __main__:

lienPrixQuanti(variable: str)
    # lienPrixQuanti(variable) prend en paramètre une chaîne de charactères correspondant à une variable quantitative
    # et retourne les coefficients de corrélation et un nuage de points de la variable quantitative en fonction du prix

Help on function lienPrixQuali in module __main__:

lienPrixQuali(variable)
    # lienPrixQuanti(variable) prend en paramètre une chaîne de charactères correspondant à une variable qualitative
    # et retourne une Anova et une boîte à moustaches de la variable en fonction du prix
```

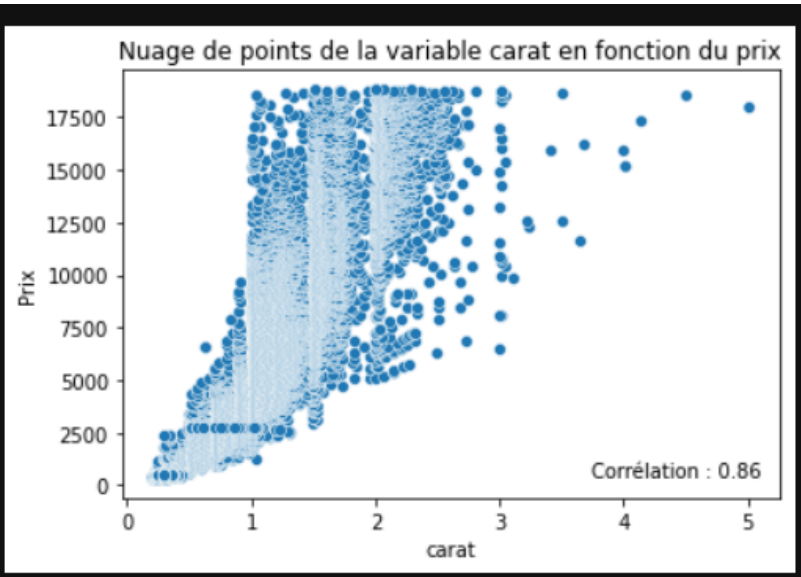
Enfin, pour les utiliser, nous bouclons sur toutes les variables et nous appelons la fonction correspondant à la nature de cette dernière :

```python
# Pour chaque variable de diamonds, si la variable est quantitative,
# on lance la fonction lienPrixQuanti(variable), sinon on lance la fonction lienPrixQuali(variable)
for variable in diamonds.columns :

    print("Analyse de la variable {} :\n".format(variable).upper())

    if diamonds[variable].dtype == "int64" or diamonds[variable].dtype == "float64" :
        lienPrixQuanti(variable)
    else :
        lienPrixQuali(variable)

    if variable != diamonds.columns[-1] :
        print("\n" + '-' * 60)
```

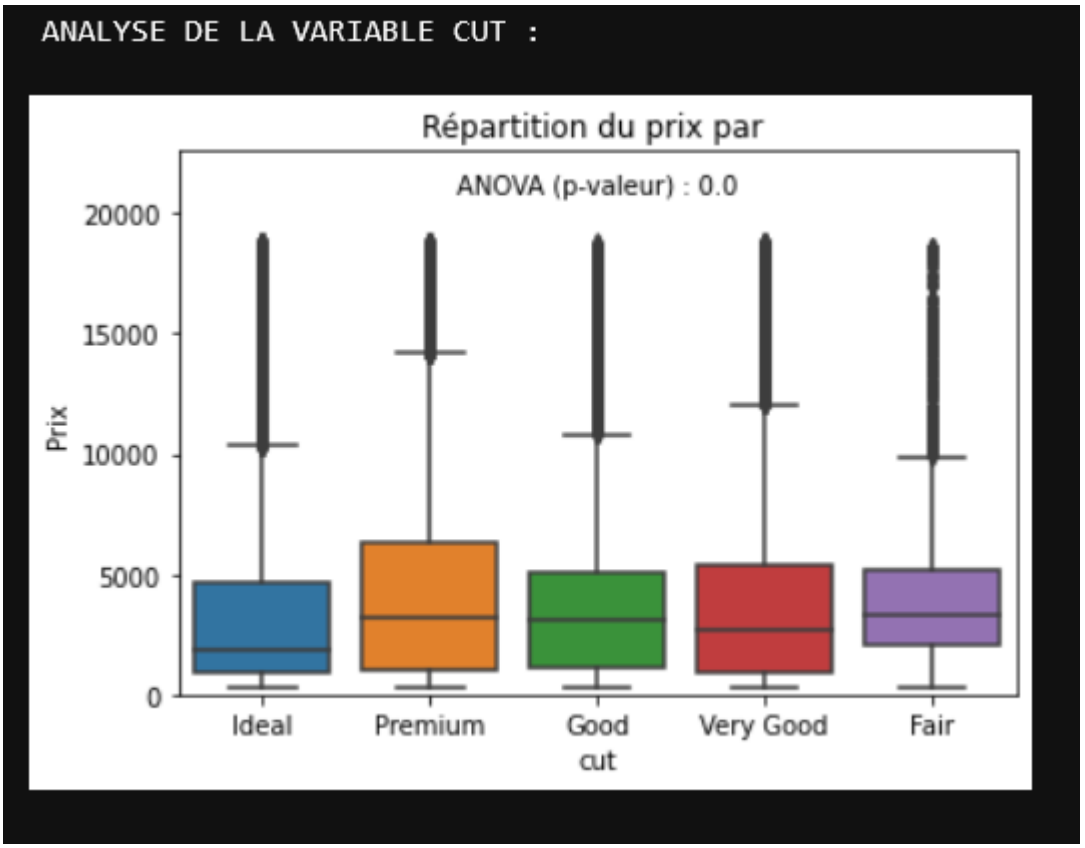
Pour les variables quantitatives la sortie est :

```
ANALYSE DE LA VARIABLE CARAT :


Corrélation de la variable carat avec les autres variables quantitatives :

x         0.975094
z         0.953387
y         0.951722
price     0.921591
table     0.181618
depth     0.028224
Name: carat, dtype: float64
```





Et pour les variables qualitatives le résultat est :



Conclusion

Après ces observations, nous pouvons en déduire que le carat a une forte influence sur le prix lorsqu'il augmente, car les carats les plus élevés sont plus rares. De plus, un carat élevé signifie un poids et un volume élevé du diamant, et permet de se distinguer socialement. Aussi, la coupe du diamant a un intérêt certain. Bien que la majorité des coupes soient idéales ou mieux, les coupes plus qu'idéale restent plus rares et justifient une augmentation du prix.