Propos liminaire.

Lorsqu'on débute un travail d'analyse de données (ou de sciences des données), le premier reflèxe est de vérifier la distribution des données. Lors de cette étape, on s'intéresse notamment à détecter les **outliers**. Ces points sont connus sous diverses terminologies : points extrêmes, points aberrants, points atypiques, etc. Leur détection et leur traitement est très important car leur présence va induire un biais dans les conclusions qui seront faites à partir des données. Également, ces points peuvent impacter les performances de vos modèles (sur- ou sous-apprentissage). Il est donc primordiale de savoir le traiter. Pour cela, il faut les repérer et les qualifier.

J'attire votre attention sur ce dernier point : il est vrai que les outliers connaissent de nombreuses dénominations. Néanmoins, j'utilise deux termes distincts pour qualifier les deux types d'outliers que vous pouvez observer : les points atypiques et les points aberrants. Les premiers sont atypiques en cela qu'ils représentent une mesure correcte du phénomène étudié : par exemple, si vous vous intéressez à la taille des individus d'un échantillon, il est probable d'observer un individu mesurant 200cm. Certe, ce point est peu probable mais il n'en demeure pas moins une mesure correcte du phénomène étudié. À l'inverse, si pour ce même échantillon d'individus, vous observez une taille de 20cm, alors la mesure récolté n'a pas de sens avec le phénomène étudié : il s'agit donc d'un point aberrant, en cela qu'il n'a pas de sens avec le phénomène étudié.

Au long de l'exposé, je vais donc conserver cette terminologie : les outliers ayant du sens avec le phénomène étudié seront qualifiés de **points atypiques** ; les outliers qui n'ont pas de sens avec la mesure du phénomène étudié seront qualifiés de **points aberrants**.

Introduction.

Un **point extrême**, plus connu sous le nom d'**outlier** ou d'**anomalie**, est une observation distante des autres observations de la distribution. Ces points peuvent être de deux natures, soit ils sont le résultat de la variabilité propre au phénomène étudier, soit ils sont dus à une erreur de saisie. Dans le premier cas je parles de **points atypiques** puisqu'ils ont du sens pour le phénomène observé. Dans le second cas, je parlerais de **points aberrants** en cela qu'ils n'ont pas de sens vis-à-vis du phénomène observé.

Nous le savons, certains paramètres statistiques sont **sensibles aux valeurs extrêmes**. Ainsi, ne pas détecter, qualifier et traiter ces points peut conduire à modifier à tort les paramètres statistiques de la distribution. Dans la mesure où ces paramètres - la moyenne et l'écart-type, par exemple - sont utilisés dans de nombreuses analyses inférentielles, **nos résultats seront biaisés** puisque l'estimation faite s'éloignera significativement de la vrai valeur du paramètre à estimer. Également, ces valeurs extrêmes peuvent contraindre les modèles à s'ajuster à elles, **créant un phénomène de surapprentissage** (surajustement).

Dans ce qui suit nous allons voir **comment détecter, qualifier et traiter ces points extrêmes**. Pour détecter les outliers, nous aborderons différentes méthodes : numériques, graphiques et inférentielles. Puis, nous aborderons la qualification de la nature de l'outlier. En effet, savoir qualifier la nature de l'anomalie est important afin d'adapter notre **stratégie de traitement**. Entre autres, nous verrons qu'un point atypique peut être important pour les analyses statistiques et donc qu'il peut être utile de le conserver dans nos analyses - dans l'industrie, cela peut révéler une défaillance du processus de production ; en finance et en assurance, aider à la détection de fraudes. Nous développerons donc différentes stratégies de traitement des anomalies. À la fin de l'exposer vous trouverez un arbre de décision pour la

détection et le traitement des outliers en fonction de leur nature. Finalement, nous présenterons une étude de cas en nous appuyant sur les données *AirBnB* disponible sur le site Kaggle.Com.

Détection d'outliers.

La détection d'outlier est très importante avant la mise en place d'une analyse inférentielle ou d'un algorithme d'apprentissage automatique. En effet, la présence de points lointains peut biaiser le résultat des analyses ou impacter la performance d'un modèle (d'apprentissage automatique). Pour cette raison, leur détection ne doit pas être minorée et elle doit être réalisée dans les premières phases du projet - cf. Analyse des données exploratoire, **analyse univariée**.

Lors de l'analyse exploratoire des données, vous allez vous intéresser à la distribution de vos variables. Concernant les variables quantitatives, vous allez consulter les valeurs de certains paramètres statistiques clés tels que les paramètres de tendance centrale (*étendue, Q1, Q2, Q3, Q4, moyenne*) et ceux de dispertion (*variance, écart-type*). Cela vous donnera un premier aperçu de l'état de votre distribution, par exemple, si elle est asymétrique ($$moyenne - mediane \neq 0$$)[^1].

Également, vous allez utiliser un histograme et/ou un diagramme en boîte (a.k.a. boîte à moustaches ou boxplot) pour inspecter visuellement la distribution des données.

C'est très bien tout ça, mais quel rapport avec nos outliers ? C'est simple, ces paramètres de distribution et ces deux graphiques vont nous aider à détecter les outliers.

Pour illustrer l'exposé, je vais utiliser les données *Airbnb* disponible sur Kaggle. Cet ensemble de données sera également utilisé lors de l'étude de cas.

```
import pandas as pd
import numpy as np

df = pd.read_csv('airbnb.csv')
```

Méthodes graphiques.

Par exemple, si vous éditez un histogramme, vous avez un premier aperçu de la distribution des données et de la présence d'outliers. En outre, des barres peuvent être distantes du reste de la distribution ou vous pouvez simplement constater la présence de valeurs qui étirent les queues de l'histogramme. Observez plutôt l'histogramme de la variable "log_price".

```
import plotly.express as px

fig = px.histogram(df, x='log_price', nbins=40, title='Histogram of
Log(Price)',)
fig.update_layout(xaxis_title='Log Price', yaxis_title='Count')
fig.update_traces(texttemplate='%{y}', textposition='auto')
fig.show()
```

![Histogramme]({{ site.baseurl }}/assets/img/outliers_detection/histogram_logprice.png)

On remarque nettement la présence d'outliers dans la queue inférieure de la distribution. Il existe deux points qui se distingue particulièrement des autres. Qui plus est, les queues de la distribution semblent étirées par quelques observations. Nous allons vérifier ce point avec un diagramme en boîte.

> la bibliothèque *plotly* de Python est très utile et pratique, notamment pour créer des dashboards dynamiques avec Streamlit ou des graphiques interactifs pour votre portofolio.

Méthodes par critère/score.

Le **diagramme en boîte**, communément appelé boxplot, est une solution graphique qui permet de visualiser la distribution d'une variable quantitative et d'isoler les outliers. En effet, lorsque vous éditez un diagramme en boîte, vous observez parfois **des points qui sont à l'extérieur des moustaches**. Ces points sont des outliers. *Qu'est ce qui permet à un boxplot de définir la longueur des moustaches ?* C'est l'écart interquartile. Plus précisémment, **le critère de Tukey**. Ainsi, on considérera qu'un point est extrême dès lors qu'il se situe à plus ou moins 1.5 fois l'écart interquartile.

Vérifions l'analyse faite à la lecture de l'histogramme.

```
fig = px.box(df, x='log_price', title='Boxplot of Log(Price)')
fig.update_layout(xaxis_title='Log Price',)
fig.show()
```

![Diagramme]({{ site.baseurl }}/assets/img/outliers_detection/boxplot_logprice.png)

D'après le critère de Tukey, on peut confirmer la présence de nombreux outliers. D'une part, deux points sont extrêmes dans la partie inférieure de la distribution, et, d'autre part, de nombreux points étirent anormalement la distribution supérieure des données.

Le critère de Tukey.

Une fois votre boxplot éditer, vous observez un certain nombre d'outliers. Pour les retrouver, il vous suffit de rechercher l'ensemble des points situés à plus ou moins 1.5 fois l'écart interquatile de la distribution. Soit,

$$ cutoff = 1.5*(Q3-Q1) $$

on obtient alors, $$lower_bound = Q1 - cutoff$$ et $$upper_bound = Q3 + cutoff$$. Tous les points situés au-dessus de l'upper_bound ou en-dessous du lower_bound seront considérés comme outliers. Cette méthode présente l'avantage de s'appliquer à des échantillons dont la distribution n'est pas Normale.

Voici une fonction permettant de trouver les points extrêmes d'après ce critère.

```
def tukey_outliers(data = df, variable = "log_price"):
    outliers = []

    Q1, Q3 = np.quantile(df[variable], 0.25) , np.quantile(df[variable],
0.75)
    IQR = Q3 - Q1
```

```
    upper_bound = Q3 + 1.5*IQR
    lower_bound = Q1 - 1.5*IQR

    for i in df[variable]:
        if i < lower_bound:
            outliers.append("lower_outlier")
        elif i > upper_bound:
            outliers.append("upper_outlier")
        else:
            outliers.append("normal")
    return outliers
```

Cette fonction permet de cibler chaque point d'après le critère de Tukey ; elle retourne une liste contenant, pour chaque individu, son classement en tant que "lower_outlier", "normal" ou "upper_outlier".

Voyons le nombre d'outliers situés dans les parties inférieure et supérieure de la distribution.

```
df['tukey_outliers'] = tukey_outliers()
df.tukey_outliers.value_counts()

tukey_outliers
normal           72579
upper_outlier     1372
lower_outlier      160
Name: count, dtype: int64
```

Le **critère de Tukey** établit la présence de **1532 points extrêmes**. Si on souhaite mener des analyses sur ces individus, il sera facile d'accéder à leurs données. Par exemple, ces outliers sont-ils davantage présents dans une ville plutôt qu'une autre ?

```
(df[(df["tukey_outliers"]=="upper_outlier") |
(df["tukey_outliers"]=="lower_outlier")]
["city"]
.value_counts(normalize=True)
.rename_axis("Cities")
.rename("Proportion de points extrêmes par ville"))

Cities
LA         0.364230
DC         0.214752
NYC        0.209530
SF         0.150131
Chicago    0.039817
Boston     0.021540
Name: Proportion de points extrêmes par ville, dtype: float64
```

Les villes de Los Angeles, Washington, New-York et San Fransisco sont davantages concernées par la présence d'outliers.

Si vous souhaitez simplement **récupérer les indexes des outliers** pour ensuite interroger votre base de données :

```
tukey_outliers_index = df[df["tukey_outliers"] != "normal"].index.to_list()
df.iloc[tukey_outliers_index]
```

Outre la critère de Tukey, il existe une autre méthode numérique pour détecter les outliers : le **Z score**. Comme le critère de Tukey, il permet de définir une frontière numérique au_delà de laquelle un point sera jugé comme lointain du reste de la distribution.

Z score

Le fameux **z-score**! Il est partout décidément. En même temps, qu'est ce qu'il est pratique ! Pratique mais fragile... l'usage d'un tel score repose sur l'hypothèse d'une **distribution normale de vos données**. Aussi son usage doit être maitrisé pour ne pas conduire à des conclusions erronées. Pour rappel, le score Z permet de **décrire un individu statistique en fonction de son écart au centre de gravité de la distribution, et ce, en unité d'écart type**. Ainsi, puisque vos données sont normalement distribuées il vous indique où se situe le point sur une distribution normale, de sorte que si un point est égale à 2, un peu plus de 95% des valeurs sont inférieur à lui (en valeur absolu). On va donc pouvoir fixer un **seuil au-delà duquel les valeurs seront jugées comme lointaines** du reste de la distribution car peu probable d'être observées.

Mais quel valeur seuil choisir ? Puisque vos données sont normalement distribuées - au moins approximativement (cf. Théorème central limite ; pour un bref rappel, consultez cet article) - vous savez que **près de 68% (95%, 99.7%) des valeurs de votre distribution sont situées à plus ou moins un (deux, trois) écart-type de la moyenne**. Il semble donc judicieux de choisir un seuil au moins égal à deux pour juger qu'une valeur est lointaine de la distribution. Ce "lointain" se définit comme l'ensemble des valeurs dont la probabilité d'être observé est faible. En fixant un seuil de 2 écart-type, le "lointain" sera l'ensemble des valeurs ayant une probabilité d'être observé inférieur à 5%. Attention toutefois au placemeent de ce seuil, beaucoup d'article préconise de fixer une valeur de 3.

Formellement le score-Z est égal à,

$$ Z = \frac{X - \bar{x}}{\sigma} $$ où, $$\bar{x}$$ et $$\sigma$$ sont, respectivement, la moyenne et l'écart-type de l'échantillon.

Quelques prudences à adopter lors du calcul du score Z :

1. Vérifier la normalité de la distribution ;
2. Le choix du seuil est décisif et impactera forcément le résultat. Il dépend du niveau de sensibilité souhaité ;
3. Le score suppose un "lointain" par rapport à la moyenne. Cela peut être problématique puisque la moyenne est sensible aux outliers : c'est un peu le serpent qui se mord la queue cette histoire.

Vérifions la normalité de la distribution, avec un risque $$\alpha$$ de 5%. J'utilise le test de Kolmogorov-Smirnov pour deux raisons : la présence de nombreux outliers, et la taille de l'échantillon. En effet, l'utilisation du test de Shapiro-Wilk n'est pas possible pour les échantillons de plus de 5'000 individus.

Concernant le premier point, le test de Shapiro-Wilk est très sensible aux outliers. Aussi, combien même la taille de l'échantillon eu été raisonnable, il est préférable d'utiliser un test d'adéquation non-paramétrique tel que celui de Kolmogorov-Smirnov.

```
from statsmodels.stats.diagnostic import kstest_normal as kstest

statistic, pvalue = kstest(df.log_price, dist='norm', pvalmethod='table')
print(statistic, pvalue)

statistic : 0.06274151563260133, pvalue : 0.0009999999999998899
```

> pour réaliser un test de Kolmogorov-Smirnov, vous pouvez aussi utiliser la méthode *kstest* du module *scipy.stats*.

D'après le résultat du test, la p-value est inférieure au seuil de significativité de 5%. On peut donc rejeter l'hypothèse nulle au profit de l'hypothèse alternative. Il est peu probable d'obtenir de telles données en supposant qu'elles soient normalement distribuées. Pensez à toujours appuyer ce résultat avec un graphique quantile-quatile (**QQ-plot**).

```
import statsmodels.api as sm
qqplot = sm.qqplot(df["log_price"], fit=True, line="45")
qqplot.show()
```

![QQplot]({{ site.baseurl }}/assets/img/outliers_detection/qqplot_logprice.png)

On observe plusieurs défaut de normalité puisque les points s'éloignent de la bisectrice à plusieurs reprises. Qui plus est, deux points semblent particulièrement distant de la bisectrice.

Les données ne sont pas normalement distribuées. On peut donc conclure que l'usage de cette méthode n'est pas envisageable.

Pour réaliser ce test et ce graphique avec **R**, voici le code :

```
install.packages('dgof')
library(dgof)
library(stats)

ks.test(df$$log_price,"pnorm", mean(df$$log_price),sd(df$$log_price))
qqnorm(df$$log_price, main = "Normal Q-Q Plot",
       xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
       plot.it = TRUE,)
```

Robust Z score.

Ce dernier point m'amène à vous proposer une autre mesure du "lointain" d'une distribution : le score Z modifié/robuste.

Personnelement, je suis très prudent quant à l'usage de tehcnique quantitative conditionnelle à la distribution des données - i.e. **tests paramétriques**. En effet, ces techniques sont peu robustes car elles sont très sensible aux outliers (les revoilà ceux-là!) et aux défauts de normalités - i.e. asymétrie, par exemple. Pour cette raison, et puisqu'il existe très souvent (pour ne pas dire toujours) **une alternative robuste**, il peut être prudent d'utiliser une **alternative non-paramétrique** en substitution/complément. Par exemple, dans le cadre des tests d'homogénéité de deux moyennes, une alternative au test t de Student existe, le test de Wilcoxon-Mann-Whitney. Ce dernier est robuste puisqu'il ne suppose pas de distribution des échantillons et qu'il compare les distributions d'après les rangs de leurs observations. Aussi, il est souvent présenté comme un test d'homogénéité à la médiane.

Vous me voyez venir ? Le **score Z robuste** est simplement une alternative qui prend en compte la médiane plutôt que la moyenne. Mais, me direz-vous, si on prend en compte la médiane, mesurer l'écart à celle-ci en unité d'écart type n'a pas de sens ? Bien évidement ! On va donc définir une mesure de la variabilité de notre échantillon qui soit robuste, à savoir l'**écart absolu à la médiane** (**MAD**, Median Absolut Deviation). Il s'agit de la **distance médiane entre les données et la médiane**. Ainsi, on va pouvoir calculer l'écart individuel à la médiane en unité d'écart absolu médian à la médiane... complexe dit comme ca, voyons la formule.

Pour trouver l'**écart absolu à la médiane**, voici les étapes à suivre :

1. Calculer la médiane des données.
2. Soustraire la médiane à la valeur.
3. Prendre la valeur absolue de cette différence.
4. Calculer la médiane de cette différences absolue.

> La MAD est donc définie comme la médiane des écarts absolus par rapport à la médiane des données.

Formellement,

$$ MAD = med(| x_{i}-med(X) |) ,$$

avec $$x_{i}$$ la valeur d'un individu statistique pour la variable $$X$$. Nous avons le matériel nécessaire pour le calcul du score Z modifié.

$$Z = 1.4826 \frac{(x_{i} - med(X))}{MAD}.$$

Vous observez le coefficient $$1.4826$$ qui permet d'**approximer un équivalent médiane de l'écart-type**. Je m'explique : en multipliant l'écart à la médiane en unité de MAD par le coefficient $$\frac{1}{O.67449} \approx 1.4826$$ on s'assure que la MAD sera approximativement équivalente (au moins asymptotiquement) à l'estimateur standard de l'écart-type pour une distribution normale. Ce processus va nous permettre de **fixer un seuil** en nous appuyant sur les **quantiles de la distribution normale centrée-réduite** et d'interpréter le score Z modifié comme le score Z. Par exemple, si un individu, pour une variable donnée, obtient un score de 2, plus de 95% des individus seront inférieur à lui en valeur absolue.

Avantages du score Z modifié/robuste :

1. Les données peuvent ne pas être normalement distribuées ;

2. Les quantiles de la loi Normale centrée-réduite peuvent être utilisée pour fixer une valeur seuil ;
3. Le score Z robuste et le score Z sont comparables puisque définis sur la même échelle ;
4. Toutefois, les deux scores peuvent établir différent outliers. Cela est du au fait que la médiane est moins sensible aux valeurs extrêmes, donc il se peut que davantage de points se révèlent être des outliers avec le score Z modifié.

Calculons le score Z modifié des observations de la varible "log_price". Pour cela, on définit une fonction *robust_z_score*.

```
def robust_z_score(variable="log_price", df=df):
    med = df[variable].median()
    MAD = (np.abs(df[variable] - med )).median()
    coeff = 1/1.4826

    outliers = []

    for value in df[variable]:
        outliers.append((coeff * (value - med) / MAD))

    return outliers

df["modified_z_score"] = robust_z_score()

for i in range(3,5):
    anomalies = df[(df['modified_z_score'] < -i) | (df['modified_z_score']
> i)]
    nbr_outliers = anomalies.size
    print(f"Avec un seuil de {i} : {nbr_outliers} outliers.")

Avec un seuil de 3 : 60767 outliers.
Avec un seuil de 4 : 63 outliers.
```

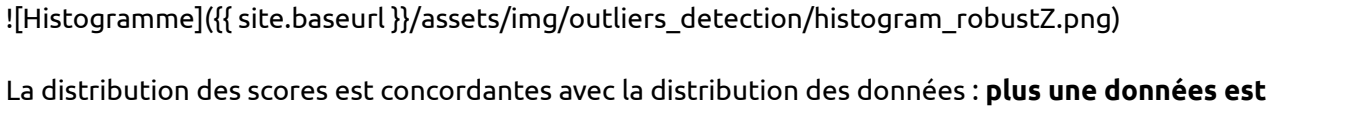
Cette fonction calcule, pour chaque observation d'une variable, le score Z modifié. L'ensemble des scores sont ajoutées à une liste. Cette liste permet d'implémenter la variable "*modified_z_score*" dans la base de données. Finalement, une boucle nous permet d'évaluer le nombre d'outliers potentiels avec une valeur seuil de *3* ou *4*. Avec un seuil de 3, le nombre d'outliers est très important. Cela est du au fait que la médiane est moins sensible aux valeurs extrêmes. Par contre, avec une **valeur seuil de 4**, on dénombre **63 outliers**. Vous pouvez accéder à leurs indices comme ceci :

```
indices = anomalies.index.to_list()
df.iloc[indices]
```

Juste pour le plaisir, observons la distribution des scores Z modifiés.

```
_ = px.histogram(df, x="robust_z_score", nbins=45, title="Histogram of
modified z score")
_.show()
```

```
```



La distribution des scores est concordantes avec la distribution des données : **plus une données est éloigné de la médiane, plus sont score (en valeur absolue) est important**.

Jusqu'alors nous avons évoqué différentes méthodes de détection d'outliers sans évoquer la réalisation de tests statistiques. La prochaine section présente un test statistique très connu en détection d'anomalies, le **test de Grubbs**.

Méthode inférentielle - le test de Grubbs.

La significativité statistique du score Z d'un point peut être évalué à l'aide d'un test d'hypothèse. Cependant, il se peut qu'un outlier soit présent à chacune des extrémités de la distribution. Aussi, il convient d'établir la **significativité statistique du score Z le plus élevé** en valeur absolue. Pour ce faire, on peut utiliser le test de Grubbs (1950) ; il est utilisé pour **détecter un unique outlier** à la fois. Notez que **ce test suppose l'échantillon normalement distribuées**.

Rappelons qu'un test d'hypothèse demande la mise en place d'un jeu d'hypothèses, d'un risque de première espèce et d'une statistique de test.

1. Le jeu des hypothèses : **Hypothèse nulle - H0** : Il n'y a pas de valeur aberrante dans l'ensemble de données. **Hypothèse alternative - H1** : Il y a exactement une valeur aberrante dans l'ensemble de données (le maximum ou le minimum).

2. Le risque d'erreur : On fixe le risque d'erreur à 5%.

3. La **statistique de test** : $$ Z = \frac{max_{i \in N}(x_i - \bar{x_i})}{\sigma}. $$

Cette statistique de test suit une **loi de Student à N-2 degrés de liberté**. Une loi de student est le rapport d'une variable normalement distribuée et la racine carré d'une variable distribuée d'après une loi du Khi-2. Une loi du Khi-2 est le rapport entre le carré d'une variable normalement distribuée et son nombre de degré de liberté $$k$$.

Règle de décision :

- Dans le cadre d'un test bilatéral. Si la probabilité d'observer la statistique de test, sous l'hypothèse nulle, est inférieure au seuil de significativité fixé $$\alpha$$, alors on pourra conclure que à la présence statistiquement significative d'un outlier - le maximum de la distribution en valeur absolu.

Le test de Grubbs - application.

Avant d'appliquer ce test, il faut vérifier la normalité de la distribution de l'échantillon. Nous savons que notre échantillon ne suit pas une loi Normale. Néanmoins, pour les besoins de l'exposé, nous allons développer ce test.

La statistique de test est la suivante : $$ G = \frac{max_{i\in N} |X_{i} - \bar{X}|}{s}$$ avec $$s$$ l'écart-type de l'échantillon et $$\bar{X}$$ sa moyenne.

Règle de décision :

Si la statistique $$G$$ se situe au-dessus de la valeur critique $$G_{critique}$$ pour un risque $$\alpha$$ donné, on concluera à la présence significative d'un outlier.

La valeur critique de G se calcul comme suit :

$$G_{critique} = \frac{N-1}{\sqrt{N}} \sqrt{\frac{t²_{\frac{\alpha}{2N}, N-2}}{N-2 + t²_{\frac{\alpha}{2N}, N-2}}}$$ avec $$t²_{\frac{\alpha}{2N}, N-2}$$ le valeur critique supérieure de la loi de student à N-2 degrés de liberté pour un risque $$\alpha$$ donné.

```
from collections import namedtuple
from scipy.stats import t

def statistic_test(x, iteration):
    """This functions calculate de G statistic to test with the Grubbs
test.

    Agrs:
    - A series, x.
    - iteration, int : the number of suspected outliers.
    Returns:
    - G, float : the statistic to test.
    - index_G, int : the index of the outlier.
    """

    mean = np.mean(x)
    std = np.std(x, ddof=0)

    abs_deviation = abs(x - mean)
    max_abs_deviation = max(abs_deviation)
    index_G = np.argmax(abs_deviation)

    score_G = max_abs_deviation / std

    print(f"Test number {iteration} has Statistics Value {score_G} which
corresponds to the {index_G} value.")
    return score_G, index_G


def critical_value(x, alpha=0.05):
    """ This function computes the critical value for comparison with the
test's statistic.

    Args:
    - A series, x.
    - A significance level alpha, float : by default, 5%.

    Returns:
    - G_critical, float : the critical value.
    """
    size = len(x)
    t_value = t.ppf(q = 1 - (alpha/(size*2)), df=size-2, loc=0, scale=1)
    G_critical = ((size-1) / np.sqrt(size)) * (np.sqrt(t_value**2 / (size-2
```

```
+ t_value**2)))

    print(f"The critical value is {G_critical}.")

    #Equivalently :
    #t = stats.t.isf(alpha/(2*n), n - 2) #equivalent to 1 - alpha/size
    #return ((n - 1) / sqrt(n)) * (sqrt(t**2 / (n - 2 + t**2)))

    return G_critical


def decision_rules(x, score_G, index_G, G_critical, iteration, verbose):
    """This function implements the decision rule for the Grubbs' test.
    It returns the decision with regard to the data point : wheter or not
    it can be considered as an outlier.

    Args:
    - A series, x.
    - A test statistic, float: G_score,
    - An index, int: index_G,
    - A critical value, float: G_critical,
    - The number of the test, int: interation.
    - A boolean : wheter to return the report for each outlier's test,
default True.

    Returns:
    - decision, str : {"Oulier", "Non-outlier"}
    """

    if score_G > G_critical:
        decision = "Outlier"
        if verbose:
            print(f"Test number {iteration} : individual {x[index_G]} is an
outlier. G > G_critical: {score_G:.4f} > {G_critical:.4f} \n")
        return decision
    else:
        decision = "Non-Outlier"
        if verbose:
            print(f"Test number {iteration} : individual {x[index_G]} is
not an outlier. G < G_critical: {score_G:.4f} < {G_critical:.4f} \n")
        return decision


def grubbs_test(x, alpha=0.05, nbr_outliers = 1, verbose=True):
    """ This function computes the iterative Grubbs test. First, it
calculates the test statistic G,
    then the critical value and compares both. Given a number of suspected
outliers, if an outlier is
    outputed, this point is removed from de sample and the test resume
until no outliers remain.
    To store results we use a namedtuple. This is very convenient to record
and append results to
    a dataframe.
```

```
    Args:
    - A series x.
    - A significance level, float : alpha, default value = 0.05.
    - A number of outlier, int : the number of suspected outliers to test.
    - Verbose, boolean : whether to return the report for each outlier's
test, default True.

    Returns:
    - results of the test, pd.DataFrame.
    """
    items = []
    result = namedtuple('result', 'indexes, value, stat_value,
critical_value,  decision')


    for iteration in range(1, nbr_outliers+1):
        score_G, index_G = statistic_test(x, iteration)
        G_critical = critical_value(x, alpha)
        decision = decision_rules(x, score_G, index_G, G_critical,
iteration, verbose)

        items.append(result(index_G, x[index_G], score_G, G_critical,
decision))

        x = np.delete(x, index_G)

    return pd.DataFrame(items)
```

Précédemment, le critère de Tukey et le score Z modifié ont permis de mettre en lumière l'exsistence de, respectivement, 1532 et 63 outliers. La fonction que nous venos de définir permet de réaliser un test de Grubbs itératif, c'est-à-dire de mettre en lumière un nombre de potentiels outliers. Nous allons donc réaliser ce test sur 1533 individus ; nous fixons le seuil de significativité à 5%.

```
outliers = grubbs_test(x = df["log_price"], alpha=0.05, nbr_outliers =
1533, verbose=False)
outliers.decision.value_counts()

decision
Non-Outlier    1532
Outlier           1
Name: count, dtype: int64
```

D'après les tests réalisés, seulement **une observation peut être qualifiée d'outliers**, au seuil de significativité de 5%. Notre test renvoi un dataframe pandas contenant l'index, la valeur de l'observation, la statistic de test, la valeur critique et la conclusion du test. Nous pouvons donc consulter l'index de l'outlier et sa valeur.

```
outliers[outliers["decision"] == "Outlier"]

    indexes     value   stat_value  critical_value  decision
0   11632       0.0     6.665936    4.968122        Outlier
```

Notre unique outlier est donc l'individu 11632, sa valeur est 0.0. Effectivement, une valeur de 1$ pour le prix d'une nuité airbnb semble aberrant.

Vérifions cela avec le test du module *outliers* de python. Ce module possède une fonction *grubbs* permettant d'implémenter ce test. Le test est effectué sur une série et recherche un unique outlier. Si le test détecte un outlier, il renvoi la série sans l'outlier.

```
from outliers import smirnov_grubbs as grubbs

outliers1 = grubbs.two_sided_test(df['log_price'], alpha=0.05)
outliers2 = grubbs.two_sided_test(outliers1, alpha=0.05)
print(f"Il y a {len(df)-len(outliers2)} point extrême.")

Il y a 1 point extrême.
```

Pour récupérer la valeur ou l'indice de l'observation, il faut employer une fonction particulière : *two_sided_test_outliers* ou *two_sided_test_indices*.

```
outlier_value = grubbs.two_sided_test_outliers(df['log_price'], alpha=0.05)
outlier_index = grubbs.two_sided_test_indices(df['log_price'], alpha=0.05)

outlier_value, outlier_index

Out: ([0.0], [11632])
```

D'après le test que nous avons conduit, on peut bien conclure à la présence d'**un outlier** dans notre ensemble de données, **avec un risque de première espèce de 5%**. Il s'agit du point minimum de la distribution.

> Dans notre cas, l'usage de ce test n'est pas pertinent puisque les données ne sont pas normalement distribuées.

Nous allons maintenant voir comment traiter un outlier.

Traitement des points lointains.

La présence de points lointains peut biaiser nos analyses. Particulièrement, ils étirent la moyenne (ou l'écart-type) à tort, apportant ainsi un biais au calcul des estimateurs d'intérêt. Il faut donc les traiter et ce traitement dépendra de la nature du point.

Soit le point est de nature **aberrante** -e.g. erreur de saisie-, soit il est de nature atypique -e.g. caractéristiques rare mais néanmoins observable. Dans le premier cas, le point peut connaitre trois traitements : **rectification**, **imputation** ou **suppression**. En effet, si la valeur n'a aucun sens vis-à-vis du phénomène étudié/mesuré alors il faut mener une analyse descriptive (univariée/multivariée) pour savoir si vous êtes en mesure de repérer la cause de cette absurdité et de la remplacer par une valeur adéquate (connaissance métier, connaissance de la base de données). Sinon, vous pourrez imputer la valeur via une méthode statistique ou la supprimer si les données perdues n'ont pas grande importance pour vos analyses.

Si le point est **atypique**, vous aurez deux solutions de traitement : l'**inclure dans vos analyses** ou le **retirer de votre analyse**. C'est votre connaissance métier et surtout les objectifs de votre analyse qui vous permettront de trancher. Si la particularité du phénomène observé n'a pas de sens dans votre analyse, vous pouvez la retirer. Sinon, vous pouvez mener l'analyse avec et sans puis évaluer l'impact de cette spécificité sur votre analyse - i.e. vérifier qu'elle n'introduit pas trop de bruit dans le modèle ou qu'elle n'entraine pas de surajustement.

Avant d'évoquer le traitement des points atypiques, nous allons aborder le traitement des points aberrants.

Traitement des points aberrants.

Rectifier les données grâce à la connaissance métier.

Dans de nombreux projets scientifiques nous sommes confrontés à la présence d'anomalies. Personnellement, j'ai rencontré différent cas lors de l'analyse des données de l'*Enquête Sociale Européenne*. J'aimerai revenir sur **deux exemples** qui permettront d'illustrer la nécessité de recourir à une analyse descriptive approfondie afin de qualifier la nature du point aberrant et le traiter.

Premièrement, trois individus déclarait employer $$7'777$$ personnes. Toutefois, après vérification ces individus étaient, pour certains, sans emplois ou retraités. Il s'est avéré que la valeur '777' permettait de cibler les individus n'ayant pas répondu à la question. En outre, la connaissance métier - la connaissance de la base de données - et le recours à une analyse descriptive approfondie ont permis d'établir la nature de ces points extrêmes : il s'agissait de points aberrants, dont la valeur n'avait aucun sens. Il a donc suffit de remplacer cette valeur par la valeur cible '777'.

Imputer les données grâce à un paramètre de tendance centrale ou un algorithme surpervisé.

Deuxième exemple, une variable décrivait le temps passé (en minute et par jour) à consulter les actualités et les journaux. Cette variable présentait aussi des valeurs anormalement hautes : plusieurs centaines d'individus déclaraient passer au moins 1000 minutes par jour à consulter les actualités et journaux de leur pays ; le maximum de la distribution était à 1428 minutes. Ce chiffre est un non sens complet : 1428 minutes équivaut à 23,8 heures. Soit ils ne sont pas humain, soit il s'agit d'une erreur de saisie. J'opte pour la seconde option. Il n'en demeure pas moins qu'il faut comprendre d'où peut provenir cette erreur de saisie pour pouvoir la traiter.

Après quelques recherches, et vu la généralisation du problème, j'en conclus qu'il s'agit d'une erreur de transfère de données entre les questionnaires et la base de données. En effet, dans les questionnaires, la réponse devait être complété sous le format "HH-MM". De plus, vu la quantité d'individus concernés et l'impossibilité de décrire la façon dont ces erreurs de saisies sont intervenues, je décide d'imputer les

données. Combien même la base de données comporte plus de 37 000 individus, il est préférable de ne pas perdre l'information que pourrait apporter ces données. Alors, ce pose la question du choix de la méthode d'imputation. Peut-on simplement utiliser un paramètre comme la moyenne, ou est-il préférable d'utiliser un algorithme comme les k plus proches voisins ? Nous reviendrons sur cette question dans la section suivante.

Suppression des outliers.

Dans le cas où les valeurs aberrantes ne seraient pas torp nombreuses et que les données fournies par les individus concernés n'apportent pas d'informations particulières, vous pouvez supprimer ces points de votre base de données. Toutefois, la suppression d'individus statistiques de votre base de données peut être délicate, notamment si vous n'avez pas beaucoup de données ou que les données concernées sont porteuses d'informations particulières pouvant être valorisées lors de vos analyses.

Dans le cas où les valeurs aberrantes seraient nombreuses, une méthode d'imputation peut être idéale afin de remplacer ces valeurs.

Take home messages.

Notez que dans le cadre d'une valeur aberrante il est parfois très difficile de comprendre la source de l'erreur pour imputer les données par nous même. Si nous ne parvenons pas à identifier la nature de l'erreur pour la rectifier, à l'aide d'une valeur adéquate, alors il faudra choisir : supprimer ou imputer la valeur à l'aide d'une méthode statistique -e.g. moyenne, médiane, etc ou d'un algorithme (non-)supervisé.

Le schéma suivant reprend le chemin décisionnel pour le traitement des erreurs de saisie.

Arbre de décision : qaulification et traitement des points aberrants.

Imputation des points aberrants.

Imputation par un paramètre de tendance centrale.

Il est courant d'observer des travaux (même scientifiques) dans lesquels les valeurs manquantes et/ou les valeurs aberrantes sont remplacé par des paramètres de position tel que la moyenne ou la médiane (variable quantitative) et le mode (variable catégorielle). Cette méthode est souvant critiquée. En effet, imputer une données par la moyenne ou la médiane de la série n'est pas forcément une bonne pratique. Tout d'abord, parce que la moyenne est impactée par les valeurs extrêmes, donc si la valeur aberrante est très élevée - e.g. 1428 minutes - la moyenne de la série ne reflète pas celle du phénomène étudié. Pour cette raison il peut être plus prudent de retirer cette valeur aberrante avant de calculer la moyenne (médiane) de la série pour l'imputer à votre observation. De surcroît, l'imputation par la moyenne a tendance à **gonfler** la médiane et l'**écart-type**. Il en va de même pour l'imputation par la médiane. Vous risquez donc, avec un trop grand nombre de valeurs aberrantes, d'introduire un nouveau biais dans vos analyses et donc d'apporter des conclusions erronées. En outre, les méthodes d'imputation par algorithmes (non)-supervisés sont souvent préférées à l'imputation par un paramètre de tendance centrale. Toutefois, nous allons voir que cela n'est pas toujours le cas.

Pour cette raison, je conseille d'évaluer la qualité de votre méthode d'imputation une fois réalisée. Par exemple, on peut comparer les paramètres tels que la moyenne, la médiane et l'écart-type avant et après

l'imputation. En règle générale, je choisis l'imputation qui est la plus protectrice à l'égard de la distribution et particulièrement de l'écart-type. Notamment, je compare toujours plusieurs méthodes d'imputation : médiane, moyenne, k plus proches voisins, entre autres.

> Cela vaut aussi pour l'imputation de données manquantes.

Pour illustrer ce point, nous allons comparer, pour chaque méthode d'imputation, la médiane, la moyenne et l'écart-type, avant et après imputation. On considèrera successivement, 50, 122 et 505 valeurs aberrantes et trois méthodes d'imputation des données : **médiane, moyenne et k plus proches voisins**.

Voici les résultats de l'imputation des valeurs aberrantes pour successivement, 50, 122 et 505 observations.

On observe que lorsque le nombre de valeurs aberrantes est faible, l'imputation par la moyenne (ou la médiane) est plus conservatrice (à l'égard de la distribution) que la méthode par le k plus proches voisins. Notamment, la **variation de l'écart-type est plus importante avec la méthode des k plus proches voisins**. Cependant, lorsque le nombre de valeurs aberrantes augmente (505 observations, soit à peine 1.3% de l'effectif total), l'imputation par les knn est bien plus protectrice à l'égard de l'écart-type de la distribution. En effet, alors que l'écart-type varie de 0.0062 unités avec une imputation par la moyenne (ou la médiane), il varie de seulement 0.0042 unités avec une imputation par les k plus proches voisins. Notez tout de même que l'impact sur la moyenne de la distribution est bien plus important avec l'algorithme des k plus proches voisins, quel que soit le nombre d'observations à imputer.

En conclusion, avant d'imputer vos données, vérifiez l'impact que la méthode aura sur votre distribution, sinon vous risquez de fausser les conclusions de vos analyses : en fonction des données et du nombre de valeurs à imputer et de l'analyse que vous souhaitez effectuer, les méthodes peuvent se révélées plus ou moins bonnes.

> **Pour résumer, trois méthodes sont à votre disposition pour traiter un point aberrant : trouver/comprendre la source de l'erreur, supprimer le point en question, imputer sa valeur via une méthode statistique ou d'apprentissage automatique.**

Voici un graphique interactif pour visualiser l'évolution de la distribution à mesure que le nombre de valeurs imputées augmente et que la méthode d'imputation varie.

Traitement des points atypiques.

En ce qui concerne le traitement des points extrêmes, vous ne devez pas les imputer par une autre valeur puisque celle observer, combien même atypique, est **une mesure réelle et exacte du phénomène étudié**. Aussi, compte tenu de votre connaissance métier, il vous faudra choisir entre retirer ces points de votre analyse ou les intégrer. Demandez-vous si ce point fait partie de la **population ciblée par votre question de recherche**.

Par exemple, si vous réaliser une régression pour comprendre l'**impact du sport sur le poids des étudiants**, il se peut que vous observiez des **sportifs professionnels** dont le temps consacré par semaine à la pratique est très élevé. Ces individus seront certainement jugés comme des points extrêmes mais ne doivent pas être remplacés. Supposons, qu'un des étudiants soit un sportif de haut niveau.

- Si il ne fait pas partie de votre population cible - e.g. les étudiants lambda - il peut être retiré de l'analyse ; il n'a pas d'intérêt pour répondre à la question de recherche spécifiée, il pourrait fausser vos conclusions sur la population cible. Pensez tout de même à le notifier dans votre rapport d'analyse.

- Si il fait partie de votre population cible, vous allez le conserver dans votre analyse, mais avec quelques points de prudence à adopter. Je conseille de **mener une analyse avec et sans l'outlier** pour vous assurer qu'il **n'affecte pas les hypothèses** et/ou **le résultat** du modèle. Il se peut que sa présence entrave les résultats - e.g. ce point crée une association inexistante entre X et Y : le coefficient de régression ne reflète pas vraiment l'effet de X sur Y ; i.e. ce point agit comme un levier sur votre modèle, il crée un phénomène de sur-ajustement. Enfin, si les hypothèses de votre modèle ne sont pas respectées, essayez une transformation log ou racine carrée pour diminuer l'impact de cette valeur extrême. Quoi qu'il en soit, si vous en arrivez à extraire ce point de votre étude, il faudra le notifier en expliquant pourquoi ce point à été exclu de l'analyse.

Pour aller plus loin dans la caractérisation des points atypiques, consultez cet article.

Take home messages.

Pour détecter et traiter les outliers, posez-vous les questions suivantes :

- Un (des) point est-il détaché du reste de la distribution ? Ou, un (des) point est-il à l'extérieur des moustaches du boxplot ?
- Ce (ces) point a-t-il du sens vis-à-vis du phénomène étudié/mesuré ?
    - Si oui, il s'agit d'un point atypique.
    - Si non, il s'agit d'un point aberrant, d'une erreur de saisie.
- Le point a du sens avec le phénomène étudié : présente-t-il un intérêt pour répondre à ma question de recherche/problématique métier ? Quel impact a-t-il sur mon modèle (point levier / sur-ajustement) ?
- Le point est une erreur de saisie : son absence aura-t-elle un impact sur la réponse apportée à la question de recherche/problématique métier ?
    - Si oui, imputation : il faut évaluer la méthode d'imputation pour conserver celle qui sera la plus protectrice vis-à-vis de la distribution ;
    - Si non, suppression.

La recherche et le traitement d'outliers n'est pas une science exacte, ni un processus linéaire ; n'appliquez pas des méthodes sans évaluer leur impact.

Merci d'avoir lu cette note ! À bientot !

Références.

- Seo, Songwon. A review and comparison of methods for detecting outliers in univariate data sets. Diss. University of Pittsburgh, 2006.
- Singh, Karanjit, and Shuchita Upadhyaya. "Outlier detection: applications and techniques." International Journal of Computer Science Issues (IJCSI) 9.1 (2012): 307.
- Tukey, J.W.: Exploratory Data Analysis. Addison-Wesley, Reading, MA (1977)
- Grubbs, Frank E. "Procedures for detecting outlying observations in samples." Technometrics 11.1 (1969): 1-21.
- Tietjen, Gary L., and Roger H. Moore. "Some Grubbs-type statistics for the detection of several outliers." Technometrics 14.3 (1972): 583-597.

- Rosner, Bernard. "Percentage points for a generalized ESD many-outlier procedure." Technometrics 25.2 (1983): 165-172.

Sitographie.

https://cedric.cnam.fr/vertigo/Cours/ml/coursDonneesManquantes.html

https://lemakistatheux.wordpress.com/category/tests-statistique-indices-de-liaison-et-coefficients-de-correlation/les-tests-pour-la-detection-doutliers/

https://www.claret.ca/fr/publications/utiliser-des-echelles-logarithmiques-pour-les-graphiques-de-prix-dactions/

http://www.parisschoolofeconomics.eu/docs/tenand-marianne/modeles-en-log.pdf

https://www.jmp.com/fr_fr/statistics-knowledge-portal/exploratory-data-analysis/histogram.html

[^1]: Pour rappel, si le coefficient est inférieur à $$0$$, asymétrie à gauche ; si le coefficient est supérieur à $$0$$, asymétrie à droite.