

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université des Sciences et de la Technologie Houari Boumédiène

Faculté d'Electronique et d'Informatique
Département Informatique

Master 2 Systèmes Informatiques intelligents

Module : Data Mining

Rapport de projet, partie 1 Analyse de données

Réalisé par :

BOUROUINA Rania, 181831052716

CHIBANE Ilies, 181831072041

Année universitaire : 2022 / 2023

Table des matières

1	Manipulation du dataset	1
1.1	Importation et visualisation le contenu du dataset.	1
1.2	Description globale du dataset.	1
1.3	Description de chaque attribut du dataset.	2
1.4	Mise à jour et Enregistrement d'une instance ou d'une valeur du dataset.	2
2	Analyses des caractéristiques	4
2.1	Analyse par attribut	4
2.1.1	Calcule des mesures de tendance centrale et déduction des symétries	4
2.1.2	Calcule des mesures de dispersion et déduction des données aberrantes (outliers)	6
3	Visualisation de données	9
3.0.1	Construction une boîte à moustache et afficher les données aberrantes	9
3.0.2	Construction un histogramme et visualiser la distribution des données	10
3.1	Analyse Globale	12
3.2	Construction et affichage des diagrammes de dispersion des données avec déduction des corrélations entre les attributs	12
3.3	Notre Interface	13
3.4	Code Source du Notebook	

Introduction Générale

Le Data Mining est une technique qui consiste à examiner une grande structure de données pour trouver des modèles, des tendances, des idées cachées qui ne seraient pas possibles en utilisant des techniques plus simples basées sur des requêtes. Cette technique utilise des algorithmes mathématiques sophistiqués pour classer, diviser, segmenter l'ensemble des données, les prétraiter si nécessaire et évaluer la possibilité d'événements futurs.

Dans ce projet nous allons mettre en pratique toutes les notions vues dans le cours en passant par plusieurs phases :

1. Analyse des données
2. Prétraitement des données
3. Extraction de motifs fréquents, règles d'associations et corrélations
4. Classification et Prédiction
5. Clustering

Dans cette première partie qui est consacrée à l'analyse de données, nous avons commencé par effectuer des opérations basiques de manipulation de données. Nous avons ensuite procédé à la modification et la conservation d'une manière persistante. L'étape suivante était l'extraction et l'exploitation de connaissances tirées de notre dataset. Enfin, nous avons résumé toutes les notions importantes avec des graphiques personnalisables.

Manipulation du dataset

Introduction Dans ce chapitre, nous allons découvrir des opérations de manipulation de données qui nous permettent de les visualiser ainsi qu'en tirer des connaissances globales et particulières pour chaque donnée. La manipulation du dataset fourni nous permet l'extraction de connaissances utiles et la conservation de données.

1.1 Importation et visualisation le contenu du dataset.

Voici le résultat de l'importation et la visualisation des premières lignes de notre dataset.

```
1 df = pd.read_excel("Dataset1_HR-EmployeeAttrition.xlsx")
2 df.head()
```

✓ 0.5s

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...
0	55.0	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1.0	1	...
1	41.0	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1.0	2	...
2	55.0	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1.0	4	...
3	33.0	No	Travel_Frequently	1392	NaN	3	4	Life Sciences	1.0	5	...
4	33.0	No	Travel_Rarely	591	Research & Development	2	1	Medical	1.0	7	...

5 rows x 35 columns

FIGURE 1.1 – Aperçu du Dataset

NB : les commandes et le code source sont partagé dans l'Annexe afin d'éviter d'encombrer le rapport

1.2 Description globale du dataset.

Voici la Description globale de notre dataset.

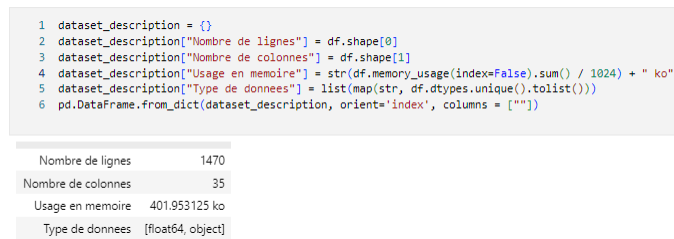


FIGURE 1.2 – Description globale du dataset

1.3 Description de chaque attribut du dataset.

Voici la Description de chaque attribut de notre dataset.

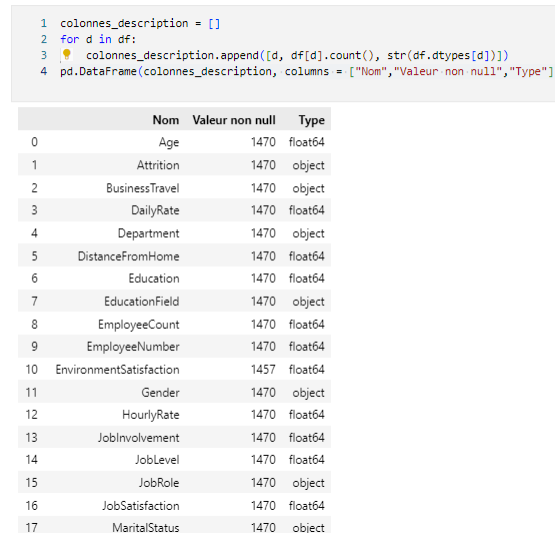


FIGURE 1.3 – Description globale du dataset

1.4 Mise à jour et Enregistrement d'une instance ou d'une valeur du dataset.

Pour mettre à jour n'importe quelle valeur dans notre dataset, l'utilisateur pourra écrire directement dans le tableau affiché dans notre interface.

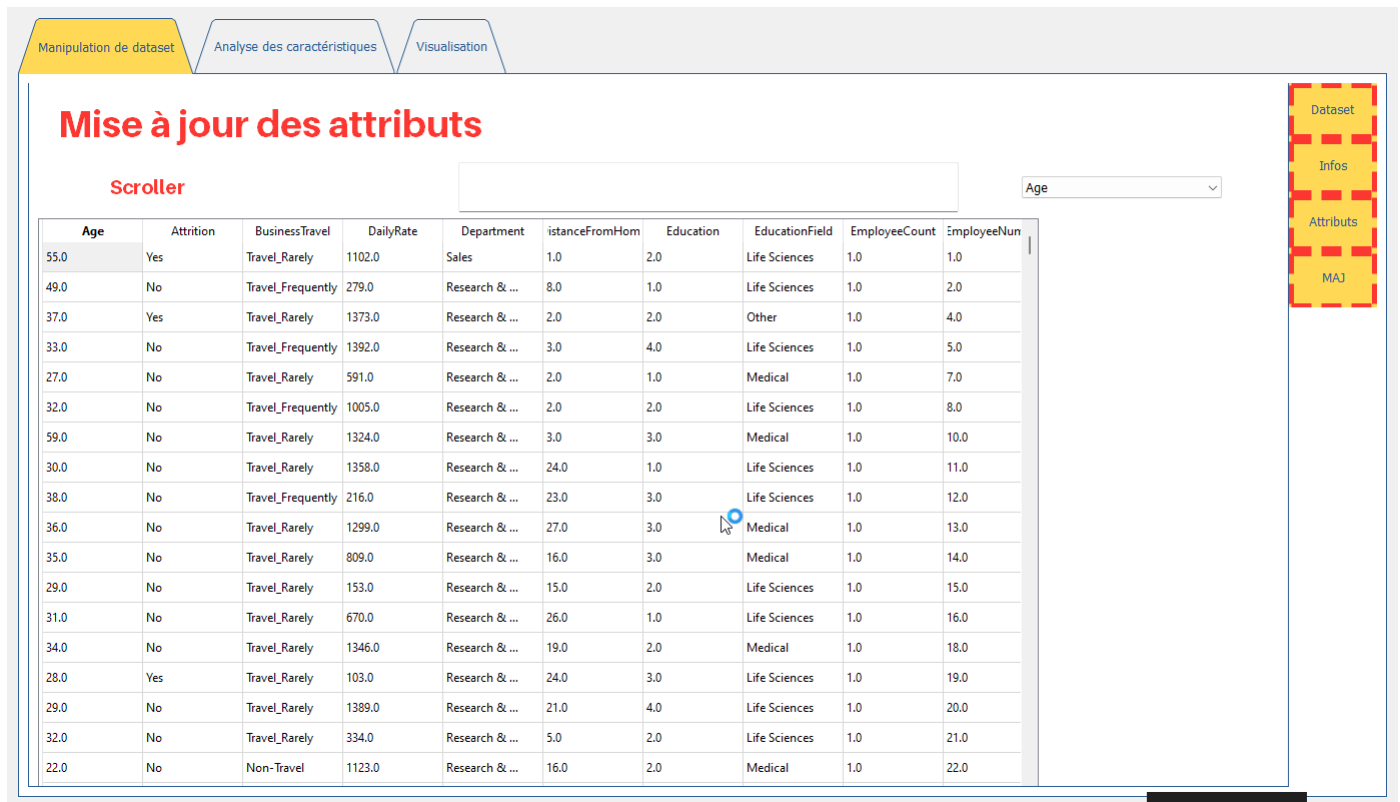


FIGURE 1.4 – Modification du dataset

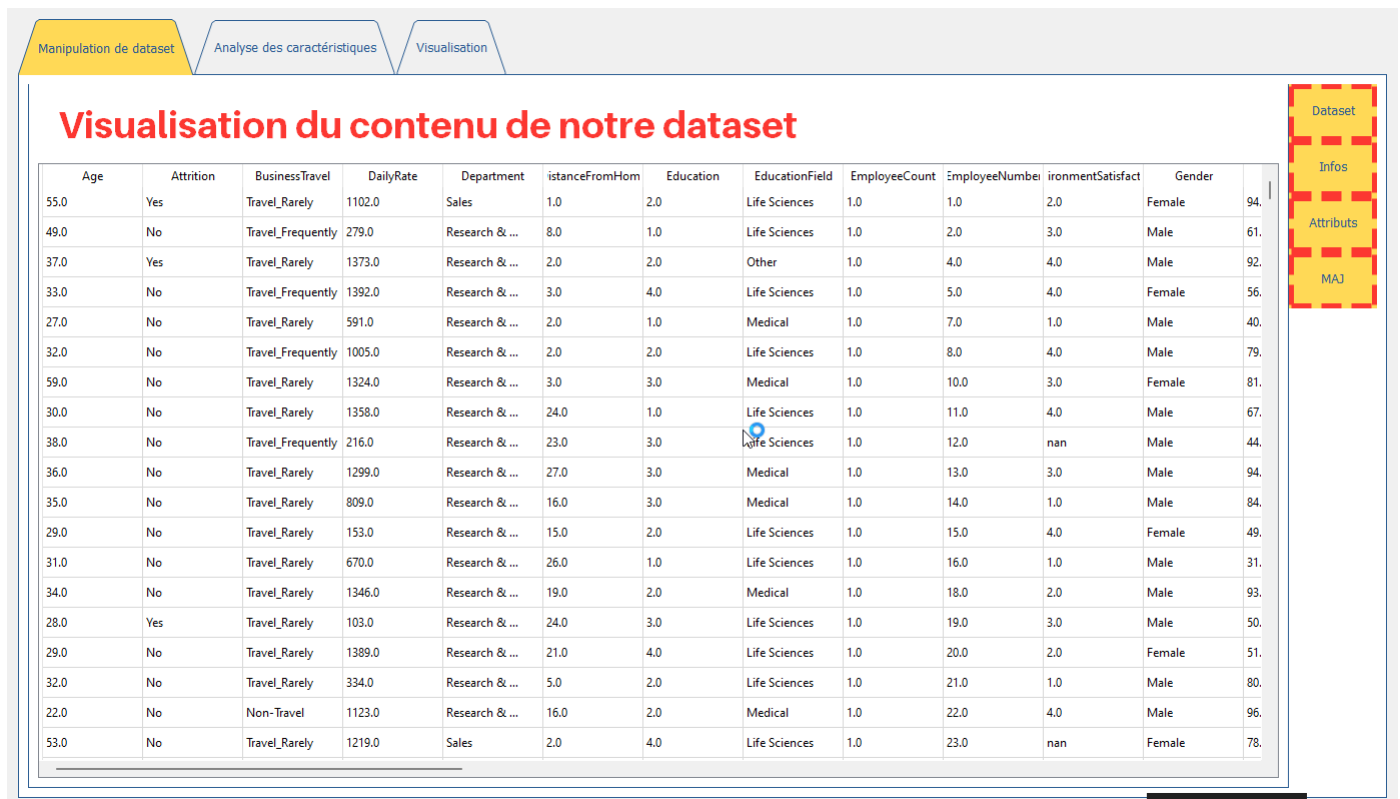


FIGURE 1.5 – Visualisation du dataset après modification

Analyses des caractéristiques

Introduction Ce chapitre nous introduit à l'analyse des caractéristiques des attributs d'un dataset. Ceci consiste en l'exploration des informations fournies par notre dataset afin de découvrir les tendances et faciliter par la suite la prise de décision.

2.1 Analyse par attribut

2.1.1 Calcule des mesures de tendance centrale et déduction des symétries

Un indicateur de tendance centrale est une valeur résumant un ensemble de données pour une variable quantitative ou ordinale. Il s'agit de résumer une série statistique par une seule valeur qui est la plus représentative autour de laquelle se concentrent les données d'une distribution. Il existe trois mesures de tendances centrales : La moyenne arithmétique, la médiane et le mode.

La moyenne est utilisée généralement pour avoir un portrait quantitatif global d'une distribution. Elle consiste à calculer la somme des valeurs et la diviser par le nombre de valeurs.

$$\bar{X} = \frac{(X_1 + X_2 + \dots + X_n)}{(n)} = \frac{1 \sum X_j}{n}$$

Concernant le mode, on y fait référence lorsqu'on cherche la ou les réponses les plus populaires dans la population étudiée. C'est la valeur la plus typique de l'ensemble, c'est-à-dire celle qui apparaît le plus souvent. Une distribution multimodale est celle où l'on peut trouver plusieurs modes.

$$Mediane = L1 + \frac{\frac{n}{2} + (\sum f)^l}{f_{Mediane}}$$

Pour ce qui est de la médiane est calculée lorsqu'on veut séparer un groupe en deux parties égales. C'est le point du milieu lorsque toutes les valeurs sont classées par ordre croissant.

$$Moyenne - Mediane = 3x(Moyenne - Mediane)$$

Par ailleurs, une distribution est dite symétrique si les valeurs observées sont réparties uniformément autour des trois mesures de tendances centrales vues auparavant.

Il existe trois types de distributions : distribution asymétrique étalée à droite, distribution symétrique, distribution asymétrique étalée à gauche.

Distribution étalée à droite

$$Mode < Mediane < Moyenne$$

Distribution symétrique

$$Mode = Mediane = Moyenne$$

Distribution étalée à gauche

$$Mode > Mediane > Moyenne$$

Voici le résultat de nos calculs des indicateurs de tendances centrales et de symétrie :

Attribut	Moyenne	Mediane	Mode	symetrie
Age	36.92	36.0	35.0	Distribution d'asymetrie positive
DailyRate	802.49	803.0	691.0	Distribution non identifie
DistanceFromHome	9.19	7.0	2.0	Distribution d'asymetrie positive
Education	2.91	3.0	3.0	Distribution symetrique
EmployeeCount	1.00	1.0	1.0	Distribution symetrique
EmployeeNumber	1024.87	1023.0	[1.0, 1391.0, 1389.0, 1387.0, 1383.0, 1382.0, ...]	Distribution non identifie
EnvironmentSatisfaction	2.72	3.0	3.0	Distribution symetrique
HourlyRate	65.89	66.0	66.0	Distribution symetrique
JobInvolvement	2.73	3.0	3.0	Distribution symetrique
JobLevel	2.06	2.0	1.0	Distribution d'asymetrie positive
JobSatisfaction	2.73	3.0	4.0	Distribution d'asymetrie negative
MonthlyIncome	6517.65	4961.5	2342.0	Distribution d'asymetrie positive
MonthlyRate	14313.10	14248.5	[4223.0, 9150.0]	Distribution d'asymetrie positive
NumCompaniesWorked	2.69	2.0	1.0	Distribution d'asymetrie positive
PercentSalaryHike	15.21	14.0	11.0	Distribution d'asymetrie positive
PerformanceRating	3.15	3.0	3.0	Distribution symetrique
RelationshipSatisfaction	2.71	3.0	3.0	Distribution symetrique
StandardHours	80.00	80.0	80.0	Distribution symetrique
StockOptionLevel	0.79	1.0	0.0	Distribution non identifie
TotalWorkingYears	11.28	10.0	10.0	Distribution non identifie
TrainingTimesLastYear	2.80	3.0	2.0	Distribution non identifie
WorkLifeBalance	2.76	3.0	3.0	Distribution symetrique
YearsAtCompany	7.01	5.0	5.0	Distribution non identifie
YearsInCurrentRole	4.23	3.0	2.0	Distribution d'asymetrie positive
YearsSinceLastPromotion	2.19	1.0	0.0	Distribution d'asymetrie positive
YearsWithCurrManager	4.12	3.0	2.0	Distribution d'asymetrie positive

2.1.2 Calcule des mesures de dispersion et déduction des données aberrantes (outliers)

Pour avoir une idée générale de l'ensemble des données de notre dataset, nous avons calculé les mesures de dispersion.

Il s'agit d'une caractérisation de l'étalement des valeurs présentes dans une distribution. Plus la distribution sera étalée, plus la valeur de la mesure de dispersion sera élevée.

Les mesures de dispersion sont : l'étendue, les quartiles, l'étendue interquartile, l'étendue des quarts, l'écart moyen, la variance et l'écart type.

L'étendue, représente la différence entre la valeur maximale et la valeur minimale de la distribution. Il s'agit donc d'une mesure qui permet de quantifier la longueur de l'intervalle dans lequel se situe les valeurs de la distribution.

$$E = X_{max} - X_{min}$$

Pour avoir une meilleure idée de la répartition des données d'une distribution, on peut la diviser en sous-groupes nommés quartiles. Ce sont trois valeurs qui séparent un ensemble de données placées

en ordre croissant en quatre sous-ensembles ayant le même nombre de données.

Pour le premier quartile, 25% des données lui sont inférieures.

Pour le second, 50% des données lui sont inférieures (c'est la médiane).

Finalement, 75% des données sont inférieures au troisième quartile.

L'étendue interquartile représente la dispersion du quart précédant la médiane et celui la succédant.

$$EtendueInterquartile = Q3 - Q1$$

Concernant les quarts, ce ne sont pas seulement les quartiles qui sont considérés, mais aussi le maximum et le minimum. Ceci permet de juger de la dispersion des données.

Voici des formules de calcul de quarts :

$$EQ1 = Q1 - X_{min}$$

$$EQ2 = Q2 - Q1$$

$$EQ3 = Q3 - Q2$$

$$EQ4 = X_{max} - Q3$$

L'écart moyen, est la moyenne des écarts à la moyenne des valeurs de la distribution. Cette mesure donne une idée exacte sur la dispersion de chacune des données avec la moyenne comme point de référence.

$$EM = \frac{\sum |x_i - \bar{x}|}{n}$$

La variance, représente la moyenne du carré des écarts à la moyenne des valeurs de la distribution. Elle nous permet de repérer les données aberrantes ou éloignées.

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1} \quad \text{ou} \quad \sigma^2 = \frac{\sum (x_i - \mu)^2}{N}$$

L'écart type, est une mesure autre de dispersion des données autour de la moyenne qui est utilisée pour comparer les données d'une même distribution entre elles. Plus l'écart type est grand, plus les données sont éloignées de chaque côté de la moyenne.

$$s = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

ou

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

Par ailleurs, une donnée éloignée aberrante est une valeur de la distribution qui est inférieure à 1.5 fois l'étendue interquartile par rapport à Q1, ou qui est supérieure à 1.5 fois l'étendue interquartile par rapport à Q3.

Une donnée est éloignée si :

$$x < Q1 - 1.5IQR \text{ ou bien } x > Q3 + 1.5IQR$$

avec $IQR = Q3 - Q1$

Voici le résultat de nos calculs des mesures de dispersion et données aberrantes :

Attribut	Ecart moyen	Variance	Ecart type	Minimum	Q1	Q3	Maximum	IQR	Donnees aberrantes
Age	[0.08, 23.08]	36.0	6.00	18.0	30.0	43.0	60.0	13.0	Aucune
DailyRate	[0.49, 700.49]	803.0	28.34	102.0	465.0	1157.0	1499.0	692.0	Aucune
DistanceFromHome	[0.19, 19.81]	7.0	2.65	1.0	2.0	14.0	29.0	12.0	Aucune
Education	[0.09, 2.09]	3.0	1.73	1.0	2.0	4.0	5.0	2.0	Aucune
EmployeeCount	[0.0, 0.0]	1.0	1.00	1.0	1.0	1.0	1.0	0.0	Aucune
EmployeeNumber	[0.13, 1043.13]	1023.0	31.98	1.0	491.0	1556.0	2068.0	1065.0	Aucune
EnvironmentSatisfaction	[0.28, 1.72]	3.0	1.73	1.0	2.0	4.0	4.0	2.0	Aucune
HourlyRate	[0.11, 35.89]	66.0	8.12	30.0	48.0	84.0	100.0	36.0	Aucune
JobInvolvement	[0.27, 1.73]	3.0	1.73	1.0	2.0	3.0	4.0	1.0	Aucune
JobLevel	[0.06, 2.94]	2.0	1.41	1.0	1.0	3.0	5.0	2.0	Aucune
JobSatisfaction	[0.27, 1.73]	3.0	1.73	1.0	2.0	4.0	4.0	2.0	Aucune
MonthlyIncome	[1.65, 13481.35]	4961.5	70.44	1009.0	2933.0	8500.0	19999.0	5567.0	18947.0, 17924.0, 19973.0, 17426.0, 19999.0, ...
MonthlyRate	[18.1, 12685.9]	14248.5	119.37	2094.0	8045.0	20462.0	26999.0	12417.0	Aucune
NumCompaniesWorked	[0.31, 6.31]	2.0	1.41	0.0	1.0	4.0	9.0	3.0	9.0
PercentSalaryHike	[0.21, 9.79]	14.0	3.74	11.0	12.0	18.0	25.0	6.0	Aucune
PerformanceRating	[0.15, 0.85]	3.0	1.73	3.0	3.0	3.0	4.0	0.0	4.0
RelationshipSatisfaction	[0.29, 1.71]	3.0	1.73	1.0	2.0	4.0	4.0	2.0	Aucune
StandardHours	[0.0, 0.0]	80.0	8.94	80.0	80.0	80.0	80.0	0.0	Aucune
StockOptionLevel	[0.21, 2.21]	1.0	1.00	0.0	0.0	1.0	3.0	1.0	3.0
TotalWorkingYears	[0.28, 28.72]	10.0	3.16	0.0	6.0	15.0	40.0	9.0	32.0, 33.0, 34.0, 35.0, 36.0, 37.0, 38.0, 40....
TrainingTimesLastYear	[0.2, 3.2]	3.0	1.73	0.0	2.0	3.0	6.0	1.0	0.0, 5.0, 6.0
WorkLifeBalance	[0.24, 1.76]	3.0	1.73	1.0	2.0	3.0	4.0	1.0	Aucune
YearsAtCompany	[0.01, 32.99]	5.0	2.24	0.0	3.0	9.0	40.0	6.0	32.0, 33.0, 34.0, 36.0, 37.0, 40.0, 19.0, 20....
YearsInCurrentRole	[0.23, 13.77]	3.0	1.73	0.0	2.0	7.0	18.0	5.0	16.0, 17.0, 18.0, 15.0
YearsSinceLastPromotion	[0.19, 12.81]	1.0	1.00	0.0	0.0	3.0	15.0	3.0	8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0
YearsWithCurrManager	[0.12, 12.88]	3.0	1.73	0.0	2.0	7.0	17.0	5.0	16.0, 17.0, 15.0

Visualisation de données

Introduction Ce chapitre est consacré à la visualisation de données où nous avons représenté les données fournies dans des graphiques de différents types afin de déceler et comprendre les notions qui sont difficilement interprétables dans un dataset sous forme d'un tableau. La visualisation de données est une approche qui consiste à analyser des ensembles de données pour en résumer les principales caractéristiques dans un graphique statistique.

3.0.1 Construction une boîte à moustache et afficher les données aberrantes

La boîte à moustaches, ou bien le diagramme de quartiles permet d'obtenir un résumé visuel sur plusieurs informations à propos de la dispersion des données d'un ensemble.

Entre autres, il montre les données minimale et maximale, les quartiles, la médiane et d'un ensemble de données. Il nous permet également d'évaluer la symétrie d'une distribution et de repérer les données aberrantes.

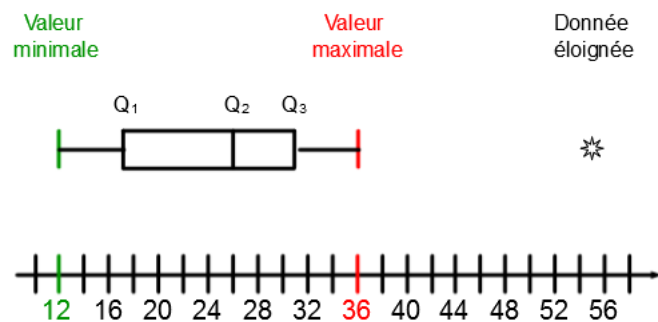


FIGURE 3.1 – Définition d'une boîte à moustaches

Voici les boites à moustaches des attributs de notre dataset :

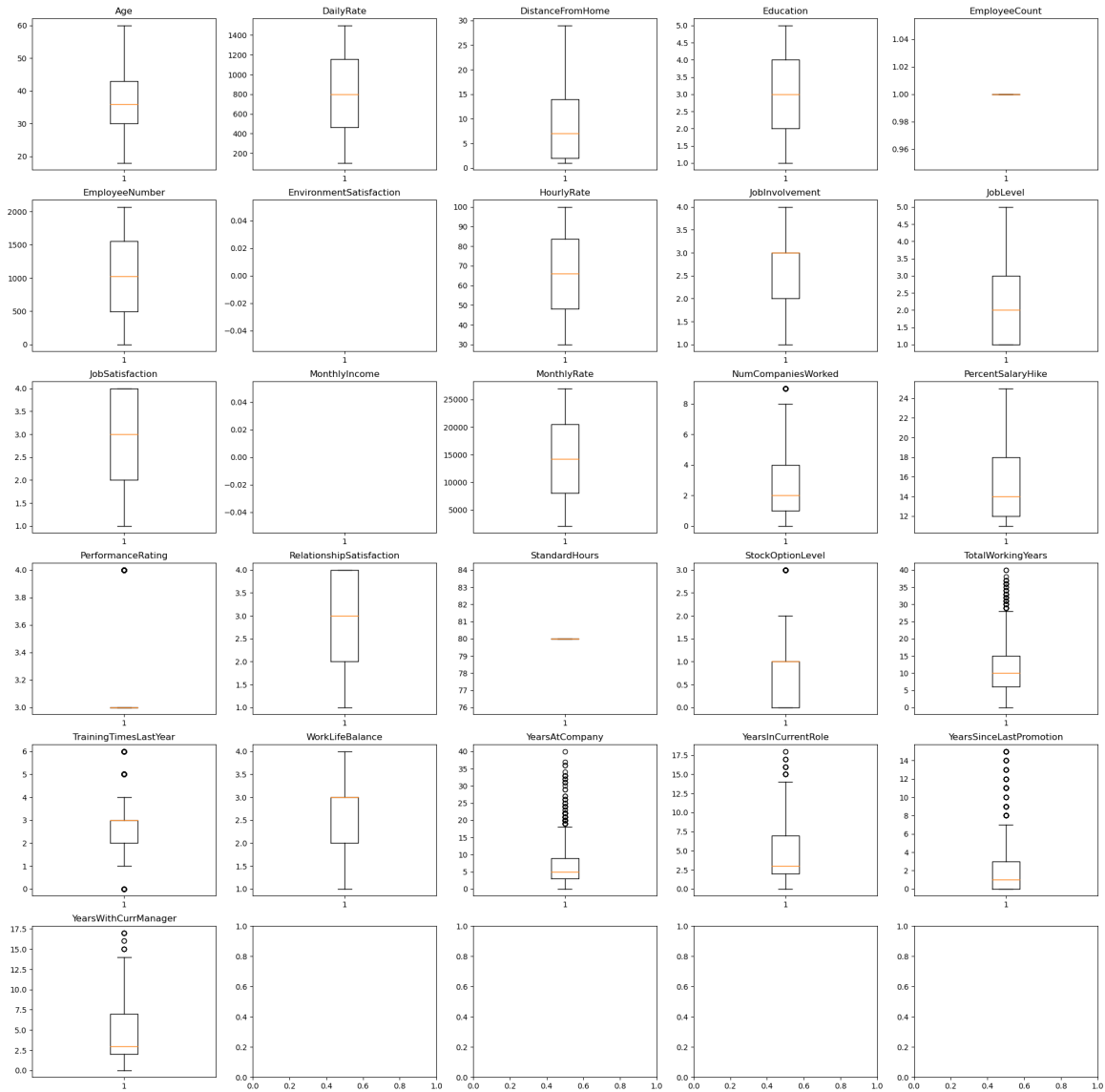


FIGURE 3.2 – Boîtes à moustaches des attributs de notre dataset

NB : Si les données sont catégoriques, il n'est pas possible de les représenter sur un boxplot, car celui-ci mesure des données quantitatives. Ceci va de même pour les attributs qui contiennent des valeurs NaN.

3.0.2 Construction un histogramme et visualiser la distribution des données

Les histogrammes sont généralement utilisés pour représenter des variables quantitatives continues qui sont regroupées en classes.

Voici les histogrammes des attributs de notre dataset :

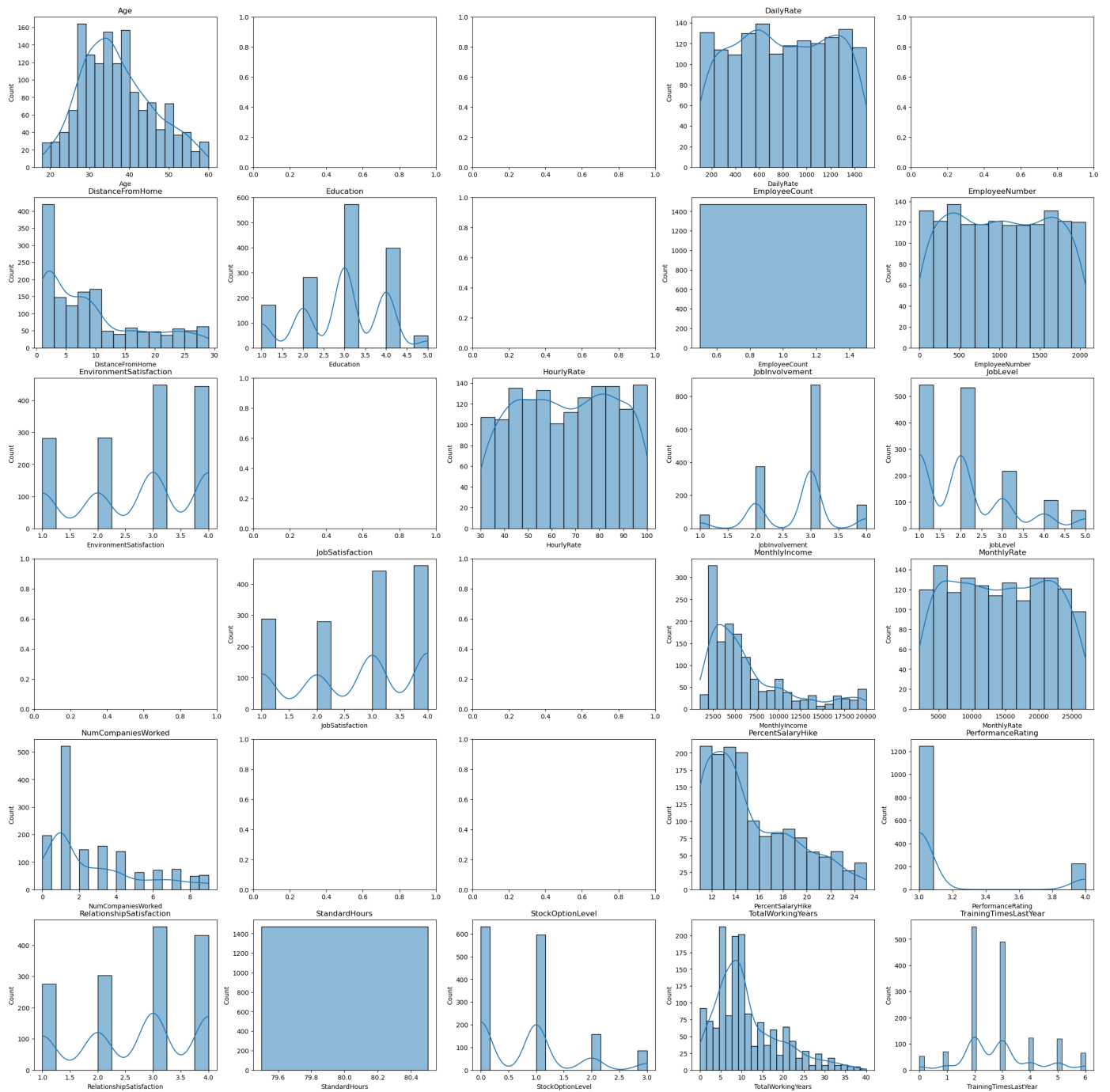


FIGURE 3.3 – Histogrammes des attributs de notre dataset

NB : Il est impossible de créer des Histogrammes pour les attributs qui contiennent des valeurs NaN.

3.1 Analyse Globale

3.2 Construction et affichage des diagrammes de dispersion des données avec déduction des corrélations entre les attributs

Le diagramme de dispersion est utilisé pour représenter des points de données sur un axe horizontal et un axe vertical afin de montrer la corrélation entre les deux. Chaque ligne du tableau de données est représentée par un marqueur dont la position dépend de ses valeurs dans les colonnes définies sur les axes X et Y.

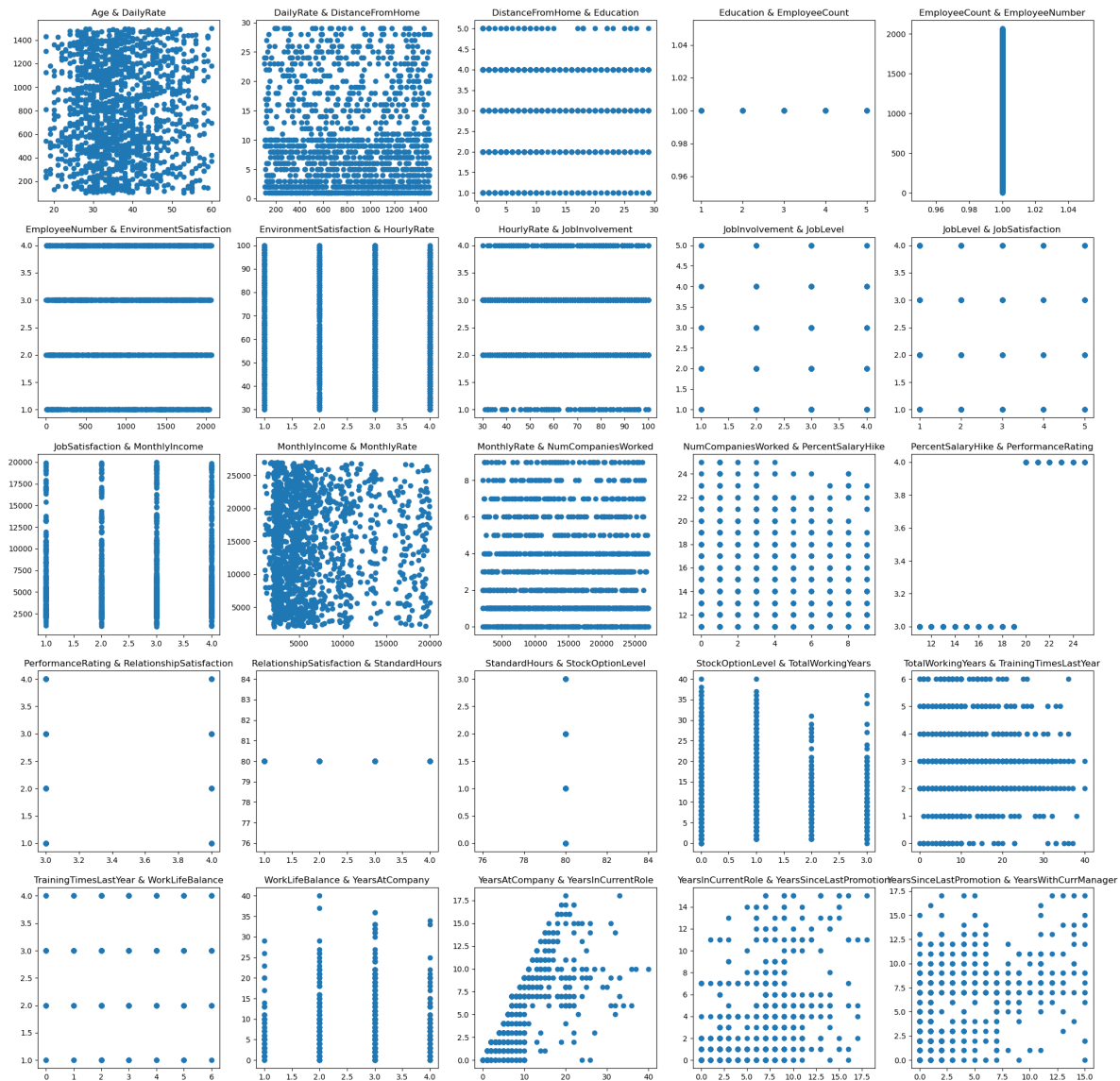


FIGURE 3.4 – Diagramme de dispersion des attributs de notre dataset

Correlation : Souvent, nous pourrions distinguer relations dans les diagrammes de dispersion. Lorsque la variable y a tendance à augmenter lorsque la variable x augmente, on dit qu'il existe une corrélation positive entre les variables. Exemple : les attributs yearsInCompany et yearsInCurrentRole.

Lorsque la variable y tend à diminuer lorsque la variable x augmente, on dit qu'il y a une corrélation négative entre les variables. Exemple : les attributs Age et DailyRate.

Lorsqu'il n'y a pas de relation claire entre les deux variables, on dit qu'il n'y a pas de corrélation entre les deux variables. Exemple : les attributs JobLevel et JobSatisfaction.

3.3 Notre Interface



FIGURE 3.5 – Notre interface

NB : Les fonctionnalités seront présentées lors de la démonstration pour éviter d'encombrer le rapport.

Annexe

3.4 Code Source du Notebook

```
1 from PyQt5 import QtCore, QtGui, QtWidgets
2 from PyQt5.QtGui import QStandardItemModel
3 from PyQt5.QtWidgets import QTreeWidgetItem
4 import sys, res
5 import pandas as pd
6 import numpy as np
7 from PyQt5.QtCore import Qt, QItemSelectionModel, QSortFilterProxyModel
8 from pandas.api.types import is_numeric_dtype
9 from PyQt5.QtCore import Qt, QSortFilterProxyModel, QAbstractTableModel,
   QModelIndex
10 import matplotlib.pyplot as plt
11 from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as
   FigureCanvas
12 from PyQt5.QtWidgets import QApplication, QWidget, QAbstractItemView
13 import sys
14 from PyQt5.QtWidgets import QDialog, QApplication, QPushButton, QVBoxLayout
15 from matplotlib.backends.backend_qt5agg import NavigationToolbar2QT as
   NavigationToolbar
16 import matplotlib.pyplot as plt
17 import seaborn as sns
18 from openpyxl import load_workbook
19
20
21 class Ui_Form(object):
22     app = QApplication([])
23     DF =pd.read_excel(
24         moustaches,scat,histo = QWidget(),QWidget(),QWidget()
25
26     '''def search(self,textEdit, table):
27         data = pd.read_excel("Dataset1_ HR-EmployeeAttrition.xlsx")
28         #sech = combo.currentIndex()
29         model = pandasModel(data)
30         filter_proxy_model = QSortFilterProxyModel()
31         filter_proxy_model.setFilterKeyColumn(-1)
32         filter_proxy_model.setSourceModel(model)
33         filter_proxy_model.setFilterCaseSensitivity(Qt.CaseInsensitive)
34         table.setModel(filter_proxy_model)'''
35
```

```

36     def plottingMoustaches(self,tab,verticalLayout,plot= ,combo1= ,combo2= )
    :
37         verticalLayout.removeWidget(self.moustaches)
38         # creating a window object
39         self.moustaches = Window(tab)
40         self.moustaches.setMinimumSize(QtCore.QSize(800, 500))
41         self.moustaches.setMaximumSize(QtCore.QSize(800, 500))
42         self.moustaches.setObjectName( )
43
44         self.moustaches.button.clicked.connect(lambda: self.moustaches.
Moustaches(combo1))
45
46         verticalLayout.addWidget(self.moustaches)
47         # showing the window
48         self.moustaches.show()
49
50     def plottingHist(self,tab,verticalLayout,plot= ,combo1= ,combo2= ):
51         verticalLayout.removeWidget(self.histo)
52         # creating a window object
53         self.histo = Window(tab)
54         self.histo.setMinimumSize(QtCore.QSize(800, 500))
55         self.histo.setMaximumSize(QtCore.QSize(800, 500))
56         self.histo.setObjectName( )
57
58         self.histo.button.clicked.connect(lambda: self.histo.hist(combo1))
59
60
61         verticalLayout.addWidget(self.histo)
62         # showing the window
63         self.histo.show()
64
65     def plottingScatter(self,tab,verticalLayout,plot= ,combo1= ,combo2= ):
66         verticalLayout.removeWidget(self.scats)
67         # creating a window object
68         self.scats = Window(tab)
69         self.scats.setMinimumSize(QtCore.QSize(800, 500))
70         self.scats.setMaximumSize(QtCore.QSize(800, 500))
71         self.scats.setObjectName( )
72
73         self.scats.button.clicked.connect(lambda: self.scats.dispersion(combo1,
combo2))
74         verticalLayout.addWidget(self.scats)
75         # showing the window
76         self.scats.show()
77

```

```

78     def tendances(self, view):
79         df = self.DF
80         tendance_centrale = {}
81         for d in df:
82             tc = {}
83             if is_numeric_dtype(df[d]):
84                 tc[ ] = self.moyenne(df, d)
85                 tc[ ] = self.median(df, d)
86                 tc[ ] = self.mode(df, d)
87                 tc[ ] = self.coeff_asy(tc)
88             tendance_centrale[d] = tc
89         self.fill_widget(view, tendance_centrale)
90         view.show()
91
92     def lookup(self, combo, text, table):
93         column=0
94         column = combo.currentIndex()
95         model = table.model()
96         start = model.index(0, column)
97         matches = model.match(
98             start, QtCore.Qt.DisplayRole,
99             text, 1, QtCore.Qt.MatchContains)
100         if matches:
101             index = matches[0]
102             table.selectionModel().select(
103                 index, QItemSelectionModel.Select)
104             table.scrollTo(index)
105
106     def scroll_tree(self, treeWidget, text):
107         newItem = []
108         newItem = treeWidget.findItems(text.currentText(), Qt.
MatchRegularExpression, 0)
109         treeWidget.scrollToItem(newItem[0])
110         print(newItem[0])
111         treeWidget.setCurrentItem(newItem[0])
112
113     def find(self, text, table, column=0 ):
114         model = table.model()
115         start = model.index(0, column)
116         matches = model.match(
117             start, QtCore.Qt.DisplayRole,
118             text, 1, QtCore.Qt.MatchContains)
119         if matches:
120             index = matches[0]
121             table.selectionModel().select(

```

```

122         index, QItemSelectionModel.Select)
123         table.scrollTo(index)
124
125     def ecart_moyen(self, df, d):
126         em = []
127         m = self.moyenne(df, d)
128         for d in df[d]:
129             em.append(float(
130                 .format(np.abs(d-m))))
131         return em
132
133     def variance(self, df, d):
134         em_sum = []
135         m = self.moyenne(df, d)
136         n = len(df[d])
137         for d in df[d]:
138             em_sum.append(np.power((d - m), 2))
139         return float(
140             .format(sum(em_sum) / n))
141
142     def ecart_type(self, var):
143         return float(
144             .format(np.sqrt(var)))
145
146     def quartilles(self, df, d):
147         colonne_sorted = list(df[d].sort_values())
148         n = len(df[d])
149         return [min(colonne_sorted), colonne_sorted[n // 4], colonne_sorted[(n
150 * 3) // 4], max(colonne_sorted)]
151
152     def iqr(self, quart):
153         return quart[2] - quart[1]
154
155     def highlight(self, combo, table):
156         high = combo.currentText()
157         self.find(high, table)
158
159     def attributs(self, view):
160         colonnes_description = []
161         df = self.DF
162         for d in df:
163             colonnes_description.append([d, df[d].count(), str(df.dtypes[d
164 ])])
165
166         dff=pd.DataFrame(colonnes_description, columns = [
167             ,
168             ])
169         model = pandasModel(dff)
170         view.setModel(model)
171         view.resize(500, 200)

```

```

164         view.show()
165
166     def load_combo(self, combo):
167         df = self.DF
168         items = df.columns
169         combo.clear()
170         combo.addItem(items)
171
172     def infos(self, widget):
173         dataset_description = {}
174         df = self.DF
175         dataset_description[ ] = df.shape[0]
176         dataset_description[ ] = df.shape[1]
177         dataset_description[ ] = str(df.memory_usage(index=
False).sum() / 1024) +
178         dataset_description[ ] = list(map(str, df.dtypes.unique
()).tolist()))
179         self.fill_widget(widget, dataset_description)
180         widget.show()
181
182     def fill_item(self, item, value):
183         item.setExpanded(True)
184         if type(value) is dict:
185             for key, val in sorted(value.items()):
186                 child = QTreeWidgetItem()
187                 child.setText(0, str(key))
188                 item.addChild(child)
189                 self.fill_item(child, val)
190         elif type(value) is list:
191             for val in value:
192                 child = QTreeWidgetItem()
193                 item.addChild(child)
194                 if type(val) is dict:
195                     child.setText(0, )
196                     self.fill_item(child, val)
197                 elif type(val) is list:
198                     child.setText(0, )
199                     self.fill_item(child, val)
200                 else:
201                     child.setText(0, str(val))
202                 child.setExpanded(True)
203         else:
204             child = QTreeWidgetItem()
205             child.setText(0, str(value))
206             item.addChild(child)

```

```

207
208     def fill_widget(self, widget, value):
209         widget.clear()
210         self.fill_item(widget.invisibleRootItem(), value)
211
212     def visualiser_dataset(self, view):
213         df = self.DF
214         model = pandasModel(df)
215         view.setModel(None)
216         view.setModel(model)
217         view.resize(1000, 700)
218         view.show()
219
220     def mesures(self, view):
221         mesures_de_dispersion = {}
222         df = self.DF
223         for d in df:
224             mdd = {}
225             if is_numeric_dtype(df[d]):
226                 em = self.ecart_moyen(df, d)
227                 mdd[ ] = [min(em), max(em)]
228                 mdd[ ] = self.median(df, d)
229                 mdd[ ] = self.ecart_type(mdd[ ])
230                 quart = self.quartilles(df, d)
231                 quart_name = [ , , , ]
232                 for q, qm in zip(quart, quart_name):
233                     mdd[qm] = q
234                 mdd[ ] = self.iqr(quart)
235                 Outliers = [x for x in df[d] if (x > (mdd[ ] + 1.5 * mdd[
236 ])) or (x < (mdd[ ] - 1.5 * mdd[ ]))]
237                 mdd[ ] = if len(Outliers) == 0 else
238 set(Outliers)
239
240         mesures_de_dispersion[d] = mdd
241
242         data= pd.DataFrame.from_dict(mesures_de_dispersion, orient= )
243         data = data.reset_index(level=0)
244         model = pandasModel(data)
245         view.setModel(model)
246         view.resize(600, 500)
247         view.show()
248
249     def setupUi(self, Form):
250         Form.setObjectName( )
251         Form.setWindowModality(QtCore.Qt.NonModal)
252         Form.resize(1000, 800)

```

```

250     Form.setStyleSheet( )
251     Form.setWindowIcon(QtGui.QIcon( ))
252     QtWidgets.QApplication.setStyle(ProxyStyle())
253     self.gridLayout_3 = QtWidgets.QGridLayout(Form)
254     self.gridLayout_3.setObjectName( )
255     self.gridLayout_2 = QtWidgets.QGridLayout()
256     self.gridLayout_2.setObjectName( )
257     self.tabWidget = QtWidgets.QTabWidget(Form)
258     self.tabWidget.setStyleSheet(
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282 )
283
284     self.tabWidget.setTabShape(QtWidgets.QTabWidget.Triangular)
285     self.tabWidget.setDocumentMode(False)
286     self.tabWidget.setTabsClosable(False)
287     self.tabWidget.setMovable(True)
288     self.tabWidget.setTabBarAutoHide(False)
289     self.tabWidget.setObjectName( )
290     self.tab = QtWidgets.QWidget()
291     self.tab.setObjectName( )
292     self.horizontalLayout_2 = QtWidgets.QHBoxLayout(self.tab)
293     self.horizontalLayout_2.setObjectName( )
294     self.horizontalLayout = QtWidgets.QHBoxLayout()

```

```
295         self.horizontalLayout.setObjectName(
296         self.tabWidget_2 = QtWidgets.QTabWidget(self.tab)
297         self.tabWidget_2.setTabBar(TabBar1(self.tabWidget_2))
298         self.tabWidget_2.setStyleSheet(
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316     )
317
318     self.tabWidget_2.setTabPosition(QtWidgets.QTabWidget.East)
319     self.tabWidget_2.setTabShape(QtWidgets.QTabWidget.Rounded)
320     self.tabWidget_2.setElideMode(QtCore.Qt.ElideLeft)
321     self.tabWidget_2.setMovable(True)
322     self.tabWidget_2.setTabBarAutoHide(True)
323     self.tabWidget_2.setObjectName(
324     self.tab_5 = QtWidgets.QWidget()
325     self.tab_5.setObjectName(
326     self.verticalLayout_2 = QtWidgets.QVBoxLayout(self.tab_5)
327     self.verticalLayout_2.setObjectName(
328     self.label = QtWidgets.QLabel(self.tab_5)
329     self.label.setStyleSheet(
330     self.label.setObjectName(
331     self.verticalLayout_2.addWidget(self.label)
332     self.verticalLayout = QtWidgets.QVBoxLayout()
333     self.verticalLayout.setObjectName(
334     self.tableView = QtWidgets.QTableView(self.tab_5)
335     self.tableView.setObjectName(
336     self.tableView.resizeColumnToContents(0)
337     self.verticalLayout.addWidget(self.tableView)
338     self.verticalLayout_2.addLayout(self.verticalLayout)
339     self.tabWidget_2.addTab(self.tab_5,
```



```

340     self.tab_7 = QtWidgets.QWidget()
341     self.tab_7.setObjectName(
342     self.verticalLayout_4 = QtWidgets.QVBoxLayout(self.tab_7)
343     self.verticalLayout_4.setObjectName(
344     self.verticalLayout_3 = QtWidgets.QVBoxLayout()
345     self.verticalLayout_3.setObjectName(
346     self.label_2 = QtWidgets.QLabel(self.tab_7)
347     self.label_2.setObjectName(
348     self.verticalLayout_3.addWidget(self.label_2)
349     self.tableView_2 = QtWidgets.QTreeWidget(self.tab_7)
350     self.tableView_2.setObjectName(
351     self.verticalLayout_3.addWidget(self.tableView_2)
352     self.verticalLayout_4.addLayout(self.verticalLayout_3)
353     self.tabWidget_2.addTab(self.tab_7,
354     self.tab_8 = QtWidgets.QWidget()
355     self.tab_8.setObjectName(
356     self.verticalLayout_6 = QtWidgets.QVBoxLayout(self.tab_8)
357     self.verticalLayout_6.setObjectName(
358     self.verticalLayout_5 = QtWidgets.QVBoxLayout()
359     self.verticalLayout_5.setObjectName(
360     self.label_3 = QtWidgets.QLabel(self.tab_8)
361     self.label_3.setObjectName(
362     self.verticalLayout_5.addWidget(self.label_3)
363     self.formLayout = QtWidgets.QFormLayout()
364     self.formLayout.setRowWrapPolicy(QtWidgets.QFormLayout.DontWrapRows)
365     self.formLayout.setFormAlignment(QtCore.Qt.AlignCenter)
366     self.formLayout.setObjectName(
367     self.tableView_3 = QtWidgets.QTableView(self.tab_8)
368     self.tableView_3.setMinimumSize(QtCore.QSize(500, 200))
369     self.tableView_3.setObjectName(
370     self.tableView_3.setMidLineWidth(100)
371     self.formLayout.setWidget(1, QtWidgets.QFormLayout.LabelRole, self.
tableView_3)
372     self.comboBox = QtWidgets.QComboBox(self.tab_8)
373     self.comboBox.setMaximumSize(QtCore.QSize(300, 16777215))
374     self.comboBox.setObjectName(
375     self.comboBox.activated.connect(lambda : self.highlight(self.comboBox,
self.tableView_3))
376     self.formLayout.setWidget(1, QtWidgets.QFormLayout.FieldRole, self.
comboBox)
377     self.label_5 = QtWidgets.QLabel(self.tab_8)
378     self.label_5.setMaximumSize(QtCore.QSize(350, 200))
379     self.label_5.setStyleSheet(
380     self.label_5.setText(

```

```

381         self.label_5.setObjectName(
382         self.formLayout.addWidget(0, QtWidgets.QFormLayout.FieldRole, self.
label_5)
383         self.label_21 = QtWidgets.QLabel(self.tab_8)
384         self.label_21.setMinimumSize(QtCore.QSize(150, 0))
385         self.label_21.setStyleSheet(
386
387         self.label_21.setText(
388         self.label_21.setObjectName(
389         self.formLayout.addWidget(2, QtWidgets.QFormLayout.LabelRole, self.
label_21)
390         self.label_22 = QtWidgets.QLabel(self.tab_8)
391         self.label_22.setMaximumSize(QtCore.QSize(150, 16777215))
392         self.label_22.setStyleSheet(
393
394         self.label_22.setText(
395         self.label_22.setObjectName(
396         self.formLayout.addWidget(2, QtWidgets.QFormLayout.FieldRole, self.
label_22)
397         self.verticalLayout_5.addLayout(self.formLayout)
398         self.verticalLayout_6.addLayout(self.verticalLayout_5)
399         self.tabWidget_2.addTab(self.tab_8,
400         self.tab_9 = QtWidgets.QWidget()
401         self.tab_9.setObjectName(
402         self.verticalLayout_8 = QtWidgets.QVBoxLayout(self.tab_9)
403         self.verticalLayout_8.setObjectName(
404         self.verticalLayout_7 = QtWidgets.QVBoxLayout()
405         self.verticalLayout_7.setObjectName(
406         self.label_4 = QtWidgets.QLabel(self.tab_9)
407         self.label_4.setObjectName(
408         self.verticalLayout_7.addWidget(self.label_4)
409         self.horizontalLayout_5 = QtWidgets.QHBoxLayout()
410         self.horizontalLayout_5.setObjectName(
411         self.label_6 = QtWidgets.QLabel(self.tab_9)
412         self.label_6.setMaximumSize(QtCore.QSize(300, 50))
413         self.label_6.setStyleSheet(
414
415         self.label_6.setObjectName(
416         self.horizontalLayout_5.addWidget(self.label_6)
417         self.textEdit = QtWidgets.QTextEdit(self.tab_9)
418         self.textEdit.setMaximumSize(QtCore.QSize(500, 50))
419         self.textEdit.setStyleSheet(
420

```

```
421         self.textEdit.setObjectName(
422
423         self.horizontalLayout_5.addWidget(self.textEdit)
424         self.comboBox_2 = QtWidgets.QComboBox(self.tab_9)
425         self.comboBox_2.setMaximumSize(QtCore.QSize(200, 16777215))
426         self.comboBox_2.setObjectName(
427         self.horizontalLayout_5.addWidget(self.comboBox_2)
428         self.verticalLayout_7.addLayout(self.horizontalLayout_5)
429         self.tableView_4 = QtWidgets.QTableView(self.tab_9)
430         self.tableView_4.setMinimumSize(QtCore.QSize(0, 500))
431         self.tableView_4.setStyleSheet(
432         self.tableView_4.setObjectName(
433         self.verticalLayout_7.addWidget(self.tableView_4)
434         self.verticalLayout_8.addLayout(self.verticalLayout_7)
435         self.tabWidget_2.addTab(self.tab_9,
436         self.horizontalLayout.addWidget(self.tabWidget_2)
437         self.horizontalLayout_2.addLayout(self.horizontalLayout)
438         self.tabWidget.addTab(self.tab,
439         self.tab_2 = QtWidgets.QWidget()
440         self.tab_2.setObjectName(
441         self.horizontalLayout_4 = QtWidgets.QHBoxLayout(self.tab_2)
442         self.horizontalLayout_4.setObjectName(
443         self.horizontalLayout_3 = QtWidgets.QHBoxLayout()
444         self.horizontalLayout_3.setObjectName(
445         self.tabWidget_3 = QtWidgets.QTabWidget(self.tab_2)
446         self.tabWidget_3.setTabPosition(QtWidgets.QTabWidget.East)
447         self.tabWidget_3.setTabShape(QtWidgets.QTabWidget.Rounded)
448         self.tabWidget_3.setObjectName(
449         #self.tabWidget_3.setTabBar(TabBar2(self.tabWidget_3))
450         self.tabWidget_3.setStyleSheet(
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
```

```

466
467
468 )
469     self.tab_6 = QtWidgets.QWidget()
470     self.tab_6.setObjectName(
471     self.verticalLayout_10 = QtWidgets.QVBoxLayout(self.tab_6)
472     self.verticalLayout_10.setObjectName(
473     self.verticalLayout_9 = QtWidgets.QVBoxLayout()
474     self.verticalLayout_9.setObjectName(
475     self.label_7 = QtWidgets.QLabel(self.tab_6)
476     self.label_7.setObjectName(
477     self.verticalLayout_9.addWidget(self.label_7)
478     self.horizontalLayout_6 = QtWidgets.QHBoxLayout()
479     self.horizontalLayout_6.setObjectName(
480     self.label_8 = QtWidgets.QLabel(self.tab_6)
481     self.label_8.setMaximumSize(QtCore.QSize(300, 50))
482     self.label_8.setStyleSheet(
483
484     )
485     self.label_8.setObjectName(
486     self.horizontalLayout_6.addWidget(self.label_8)
487     self.comboBox_3 = QtWidgets.QComboBox(self.tab_6)
488     self.comboBox_3.setMaximumSize(QtCore.QSize(300, 30))
489     self.comboBox_3.setObjectName(
490     self.horizontalLayout_6.addWidget(self.comboBox_3)
491     self.comboBox_3.activated.connect(lambda: self.scroll_tree(self.
tableView_5,self.comboBox_3))
492     self.label_9 = QtWidgets.QLabel(self.tab_6)
493     self.label_9.setMaximumSize(QtCore.QSize(90, 50))
494     self.label_9.setStyleSheet(
495
496     )
497     self.label_9.setText(
498     self.label_9.setObjectName(
499     self.horizontalLayout_6.addWidget(self.label_9)
500     self.verticalLayout_9.addLayout(self.horizontalLayout_6)
501     self.verticalLayout_10.addLayout(self.verticalLayout_9)
502     self.tableView_5 = QtWidgets.QTreeWidget(self.tab_6)
503     self.tableView_5.setMinimumSize(QtCore.QSize(0, 500))
504     self.tableView_5.setObjectName(
505     self.verticalLayout_10.addWidget(self.tableView_5)
506     self.tabWidget_3.addTab(self.tab_6,
507     self.tab_11 = QtWidgets.QWidget()
508     self.tab_11.setObjectName(
509     self.verticalLayout_12 = QtWidgets.QVBoxLayout(self.tab_11)
510     self.verticalLayout_12.setObjectName(

```

```

509     self.verticalLayout_11 = QtWidgets.QVBoxLayout()
510     self.verticalLayout_11.setObjectName(
511     self.label_10 = QtWidgets.QLabel(self.tab_11)
512     self.label_10.setObjectName(
513     self.verticalLayout_11.addWidget(self.label_10)
514     self.horizontalLayout_7 = QtWidgets.QHBoxLayout()
515     self.horizontalLayout_7.setObjectName(
516     self.label_11 = QtWidgets.QLabel(self.tab_11)
517     self.label_11.setMaximumSize(QtCore.QSize(16777215, 50))
518     self.label_11.setStyleSheet(
519
520     )
521     self.label_11.setObjectName(
522     self.horizontalLayout_7.addWidget(self.label_11)
523     self.comboBox_4 = QtWidgets.QComboBox(self.tab_11)
524     self.comboBox_4.setMaximumSize(QtCore.QSize(300, 30))
525     self.comboBox_4.setObjectName(
526     self.horizontalLayout_7.addWidget(self.comboBox_4)
527     self.label_12 = QtWidgets.QLabel(self.tab_11)
528     self.label_12.setMaximumSize(QtCore.QSize(150, 50))
529     self.label_12.setStyleSheet(
530
531     )
532     self.label_12.setText(
533     self.label_12.setObjectName(
534     self.horizontalLayout_7.addWidget(self.label_12)
535     self.verticalLayout_11.addLayout(self.horizontalLayout_7)
536     self.verticalLayout_12.addLayout(self.verticalLayout_11)
537     self.tableView_6 = QtWidgets.QTableView(self.tab_11)
538     self.tableView_6.setObjectName(
539     self.comboBox_4.activated.connect(lambda : self.highlight(self.
540     comboBox_4, self.tableView_6))
541     self.verticalLayout_12.addWidget(self.tableView_6)
542     self.tabWidget_3.addTab(self.tab_11,
543     self.tab_12 = QtWidgets.QWidget()
544     self.tab_12.setObjectName(
545     self.verticalLayout_14 = QtWidgets.QVBoxLayout(self.tab_12)
546     self.verticalLayout_14.setObjectName(
547     self.verticalLayout_13 = QtWidgets.QVBoxLayout()
548     self.verticalLayout_13.setObjectName(
549     self.label_13 = QtWidgets.QLabel(self.tab_12)
550     self.label_13.setMaximumSize(QtCore.QSize(16777215, 100))
551     self.label_13.setObjectName(
552     self.verticalLayout_13.addWidget(self.label_13)
553     self.horizontalLayout_8 = QtWidgets.QHBoxLayout()
554     self.horizontalLayout_8.setObjectName(

```

```

552     self.label_14 = QtWidgets.QLabel(self.tab_12)
553     self.label_14.setMaximumSize(QtCore.QSize(16777215, 50))
554     self.label_14.setStyleSheet(
555
556         )
557     self.label_14.setObjectName(
558         )
559     self.horizontalLayout_8.addWidget(self.label_14)
560     self.comboBox_5 = QtWidgets.QComboBox(self.tab_12)
561     self.comboBox_5.setMinimumSize(QtCore.QSize(200, 30))
562     self.comboBox_5.setMaximumSize(QtCore.QSize(500, 30))
563     self.comboBox_5.setObjectName(
564         )
565     self.horizontalLayout_8.addWidget(self.comboBox_5)
566     self.verticalLayout_13.addLayout(self.horizontalLayout_8)
567     self.verticalLayout_14.addLayout(self.verticalLayout_13)
568     self.tabWidget_3.addTab(self.tab_12,
569                             )
570     self.tab_10 = QtWidgets.QWidget()
571     self.tab_10.setObjectName(
572         )
573     self.verticalLayout_16 = QtWidgets.QVBoxLayout(self.tab_10)
574     self.verticalLayout_16.setObjectName(
575         )
576     self.verticalLayout_15 = QtWidgets.QVBoxLayout()
577     self.verticalLayout_15.setObjectName(
578         )
579     self.label_15 = QtWidgets.QLabel(self.tab_10)
580     self.label_15.setMaximumSize(QtCore.QSize(16777215, 100))
581     self.label_15.setObjectName(
582         )
583     self.verticalLayout_15.addWidget(self.label_15)
584     self.horizontalLayout_9 = QtWidgets.QHBoxLayout()
585     self.horizontalLayout_9.setObjectName(
586         )
587     self.label_16 = QtWidgets.QLabel(self.tab_10)
588     self.label_16.setMaximumSize(QtCore.QSize(16777215, 50))
589     self.label_16.setStyleSheet(
590
591         )
592     self.label_16.setObjectName(
593         )
594     self.horizontalLayout_9.addWidget(self.label_16)
595     self.comboBox_6 = QtWidgets.QComboBox(self.tab_10)
596     self.comboBox_6.setMinimumSize(QtCore.QSize(200, 30))
597     self.comboBox_6.setMaximumSize(QtCore.QSize(500, 16777215))
598     self.comboBox_6.setObjectName(
599         )
600     self.horizontalLayout_9.addWidget(self.comboBox_6)
601     self.verticalLayout_15.addLayout(self.horizontalLayout_9)
602     self.verticalLayout_16.addLayout(self.verticalLayout_15)
603     self.tabWidget_3.addTab(self.tab_10,
604                             )
605     self.horizontalLayout_3.addWidget(self.tabWidget_3)
606     self.horizontalLayout_4.addLayout(self.horizontalLayout_3)
607     self.tabWidget.addTab(self.tab_2,
608                             )

```

```

597     self.tab_13 = QtWidgets.QWidget()
598     self.tab_13.setObjectName(
599     self.verticalLayout_18 = QtWidgets.QVBoxLayout(self.tab_13)
600     self.verticalLayout_18.setObjectName(
601     self.verticalLayout_17 = QtWidgets.QVBoxLayout()
602     self.verticalLayout_17.setObjectName(
603     self.label_17 = QtWidgets.QLabel(self.tab_13)
604     self.label_17.setMaximumSize(QtCore.QSize(16777215, 100))
605     self.label_17.setObjectName(
606     self.verticalLayout_17.addWidget(self.label_17)
607     self.horizontalLayout_10 = QtWidgets.QHBoxLayout()
608     self.horizontalLayout_10.setObjectName(
609     self.label_18 = QtWidgets.QLabel(self.tab_13)
610     self.label_18.setMaximumSize(QtCore.QSize(16777215, 50))
611     self.label_18.setStyleSheet(
612
613     )
614     self.label_18.setObjectName(
615     self.horizontalLayout_10.addWidget(self.label_18)
616     self.verticalLayout_19 = QtWidgets.QVBoxLayout()
617     self.verticalLayout_19.setObjectName(
618     self.label_19 = QtWidgets.QLabel(self.tab_13)
619     self.label_19.setMaximumSize(QtCore.QSize(16777215, 50))
620     self.label_19.setStyleSheet(
621
622     )
623     self.label_19.setObjectName(
624     self.verticalLayout_19.addWidget(self.label_19)
625     self.comboBox_7 = QtWidgets.QComboBox(self.tab_13)
626     self.comboBox_7.setMinimumSize(QtCore.QSize(0, 30))
627     self.comboBox_7.setMaximumSize(QtCore.QSize(300, 16777215))
628     self.comboBox_7.setObjectName(
629     self.verticalLayout_19.addWidget(self.comboBox_7)
630     self.horizontalLayout_10.addLayout(self.verticalLayout_19)
631     self.verticalLayout_20 = QtWidgets.QVBoxLayout()
632     self.verticalLayout_20.setObjectName(
633     self.verticalLayout_21 = QtWidgets.QVBoxLayout()
634     self.verticalLayout_21.setObjectName(
635     self.label_20 = QtWidgets.QLabel(self.tab_13)
636     self.label_20.setMaximumSize(QtCore.QSize(16777215, 50))
637     self.label_20.setStyleSheet(
638
639     )
640     self.label_20.setObjectName(
641     self.verticalLayout_21.addWidget(self.label_20)

```

```

642     self.comboBox_8 = QtWidgets.QComboBox(self.tab_13)
643     self.comboBox_8.setMinimumSize(QtCore.QSize(0, 30))
644     self.comboBox_8.setMaximumSize(QtCore.QSize(300, 16777215))
645     self.comboBox_8.setObjectName(
646     self.verticalLayout_21.addWidget(self.comboBox_8)
647     self.verticalLayout_20.addLayout(self.verticalLayout_21)
648     self.horizontalLayout_10.addLayout(self.verticalLayout_20)
649     self.verticalLayout_17.addLayout(self.horizontalLayout_10)
650     self.verticalLayout_18.addLayout(self.verticalLayout_17)
651     self.tabWidget.addTab(self.tab_13,
652     self.gridLayout_2.addWidget(self.tabWidget, 0, 0, 1, 1)
653     self.gridLayout_3.addLayout(self.gridLayout_2, 0, 0, 1, 1)
654
655     self.textEdit.textChanged.connect(lambda: self.lookup(self.comboBox_2,
self.textEdit.toPlainText(), self.tableView_4))
656
657     self.retranslateUi(Form)
658     self.tabWidget.setCurrentIndex(0)
659     self.tabWidget_2.setCurrentIndex(3)
660     self.tabWidget_3.setCurrentIndex(3)
661     QtCore.QMetaObject.connectSlotsByName(Form)
662
663     def retranslateUi(self, Form):
664         _translate = QtCore.QCoreApplication.translate
665         Form.setWindowTitle(_translate(
666         self.label.setText(_translate(
667
668         self.tabWidget_2.setTabText(self.tabWidget_2.indexOf(self.tab_5),
_translate(
669         self.label_2.setText(_translate(
670
671         self.tabWidget_2.setTabText(self.tabWidget_2.indexOf(self.tab_7),
_translate(
672         self.label_3.setText(_translate(
673         self.tabWidget_2.setTabText(self.tabWidget_2.indexOf(self.tab_8),
_translate(
674         self.label_4.setText(_translate(
675         self.label_6.setText(_translate(
676         self.tabWidget_2.setTabText(self.tabWidget_2.indexOf(self.tab_9),
_translate(
677         self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab), _translate(
,
678         self.label_7.setText(_translate(
679         self.label_8.setText(_translate(

```



```

678         self.tabWidget_3.setTabText(self.tabWidget_3.indexOf(self.tab_6),
_translate(
,
))
679         self.label_10.setText(_translate(
,
))
680         self.label_11.setText(_translate(
,
))
681         self.tabWidget_3.setTabText(self.tabWidget_3.indexOf(self.tab_11),
_translate(
,
))
682         self.label_13.setText(_translate(
,
))
683         self.label_14.setText(_translate(
,
))
684         self.tabWidget_3.setTabText(self.tabWidget_3.indexOf(self.tab_12),
_translate(
,
))
685         self.label_15.setText(_translate(
,
))
686         self.label_16.setText(_translate(
,
))
687         self.tabWidget_3.setTabText(self.tabWidget_3.indexOf(self.tab_10),
_translate(
,
))
688         self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2),
_translate(
,
))
689         self.label_17.setText(_translate(
,
))
690         self.label_18.setText(_translate(
,
))
691         self.label_19.setText(_translate(
,
))
692         self.label_20.setText(_translate(
,
))
693         self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_13),
_translate(
,
))
694
695
696
697
698         self.tabWidget.currentChanged.connect(lambda: self.visualiser_dataset(
self.tableView))
699         self.tabWidget.currentChanged.connect(lambda: self.attributs(self.
tableView_3))
700         self.tabWidget_2.currentChanged.connect(lambda: self.attributs(self.
tableView_3))
701
702         self.tabWidget.currentChanged.connect(lambda: self.tendances(self.
tableView_5))
703         self.tabWidget.currentChanged.connect(lambda: self.mesures(self.
tableView_6))
704
705         self.comboBox_7.activated.connect(lambda: self.plottingScatter(self.
tab_13,self.verticalLayout_17,
, self.comboBox_7.currentText(),self.
comboBox_8.currentText()))

```

```

706         self.comboBox_8.activated.connect(lambda: self.plottingScatter(self.
tab_13,self.verticalLayout_17,
, self.comboBox_7.currentText(),self.
comboBox_8.currentText()))
707         self.comboBox_6.activated.connect(lambda: self.plottingHist(self.tab_10
,self.verticalLayout_16,
, self.comboBox_6.currentText()))
708         self.comboBox_5.activated.connect(lambda: self.plottingMoustaches(self.
tab_12,self.verticalLayout_13,
, self.comboBox_5.currentText()))
709
710
711
712
713     def moyenne(self,df, d):
714         return float(
.format(df[d].sum() / df[d].count()))
715
716     def median(self,df, d):
717         colonne_sorted = list(df[d].sort_values())
718         n = len(df[d])
719         return ((colonne_sorted[n//2] + colonne_sorted[n//2 + 1]) / 2) if n % 2
== 0 else colonne_sorted[(n+1)/2]
720
721     def mode(self,df, d):
722         l = []
723         l.append(df[d].value_counts().index)
724         l.append(list(df[d].value_counts()))
725         mod = [l[0][0]]
726         for i in range(1, len(l[0])):
727             if l[1][i] == l[1][0]:
728                 mod.append(l[0][i])
729             else:
730                 break
731         return mod if len(mod) > 1 else [] if len(mod) == len(df[d]) else mod
[0]
732
733     def coeff_asy(self,tc):
734         mode = max(tc[
]) if type(tc[
]) == list else tc[
]
735
736         if round(tc[
]) == round(tc[
]) == round(mode):
737             return
738         elif tc[
] < tc[
] < mode:
739             return
740         elif tc[
] > tc[
] > mode:
741             return
742         else:
743             return
744

```

```

745
746
747
748
749 class TabBar1(QtWidgets.QTabBar):
750     def tabSizeHint(self, index):
751         s = QtWidgets.QTabBar.tabSizeHint(self, index)
752         s.transpose()
753         return s
754
755     def paintEvent(self, event):
756         painter = QtWidgets.QStylePainter(self)
757         opt = QtWidgets.QStyleOptionTab()
758
759         for i in range(self.count()):
760             self.initStyleOption(opt, i)
761             painter.drawControl(QtWidgets.QStyle.CE_TabBarTabShape, opt)
762             painter.save()
763
764             s = opt.rect.size()
765             s.transpose()
766             r = QtCore.QRect(QtCore.QPoint(), s)
767             r.moveCenter(opt.rect.center())
768             opt.rect = r
769
770             c = self.tabRect(i).center()
771             painter.translate(c)
772             painter.rotate(270)
773             painter.translate(-c)
774             painter.drawControl(QtWidgets.QStyle.CE_TabBarTabLabel, opt);
775             painter.restore()
776
777
778
779 class ProxyStyle(QtWidgets.QProxyStyle):
780     def drawControl(self, element, opt, painter, widget):
781         if element == QtWidgets.QStyle.CE_TabBarTabLabel:
782             ic = self.pixelMetric(QtWidgets.QStyle.PM_TabBarIconSize)
783             r = QtCore.QRect(opt.rect)
784             w = 0 if opt.icon.isNull() else opt.rect.width() + self.
pixelMetric(QtWidgets.QStyle.PM_TabBarIconSize)
785             r.setHeight(opt.fontMetrics.width(opt.text) + w)
786             r.moveBottom(opt.rect.bottom())
787             opt.rect = r
788             QtWidgets.QProxyStyle.drawControl(self, element, opt, painter, widget)

```

```

789
790
791 class pandasModel(QAbstractTableModel):
792
793     def __init__(self, data):
794         QAbstractTableModel.__init__(self)
795         self._data = data
796
797     def rowCount(self, parent=None):
798         return self._data.shape[0]
799
800     def columnCount(self, parent=None):
801         return self._data.shape[1]
802
803     def data(self, index, role=Qt.DisplayRole):
804         if index.isValid():
805             if role == Qt.DisplayRole:
806                 return str(self._data.iloc[index.row(), index.column()])
807             return None
808
809     def headerData(self, col, orientation, role):
810         if orientation == Qt.Horizontal and role == Qt.DisplayRole:
811             return self._data.columns[col]
812         return None
813
814     def flags(self, index):
815         return Qt.ItemIsSelectable|Qt.ItemIsEnabled|Qt.ItemIsEditable
816
817     def setData(self, index, value, role):
818         if value== :
819             value = np.nan
820         if role == Qt.EditRole:
821             # Set the value into the frame.
822             self._data.iloc[index.row(), index.column()] = value
823             dffff = pd.read_excel(
824                 dffff.iloc[int(index.row()), int(index.column())] =value
825                 dffff.update(self._data)
826                 dffff.to_excel(
827                     , index =False)
828             return True
829         return False
830
831
832 class Window(QWidget):
833     DF =pd.read_excel(

```

```

834
835     # constructor
836     def __init__(self, parent=None):
837         super(Window, self).__init__(parent)
838
839         # a figure instance to plot on
840         self.figure = plt.figure()
841
842         # this is the Canvas Widget that
843         # displays the 'figure' it takes the
844         # 'figure' instance as a parameter to __init__
845         self.canvas = FigureCanvas(self.figure)
846
847         # this is the Navigation widget
848         # it takes the Canvas widget and a parent
849         self.toolbar = NavigationToolbar(self.canvas, self)
850
851         # Just some button connected to 'plot' method
852         self.button = QPushButton(
853
854         # creating a Vertical Box layout
855         layout = QVBoxLayout()
856
857         # adding tool bar to the layout
858         layout.addWidget(self.toolbar)
859
860         # adding canvas to the layout
861         layout.addWidget(self.canvas)
862
863         # adding push button to the layout
864         layout.addWidget(self.button)
865
866         # setting layout to the main window
867         self.setLayout(layout)
868
869     # action called by the push button
870     def Moustaches(self, combo):
871
872         # clearing old figure
873         self.figure.clear()
874         df = self.DF
875         axe = self.figure.add_subplot(111)
876         if is_numeric_dtype(df[combo]):
877             axe.boxplot(df[combo])
878             axe.set_title(combo)

```

```

879         # refresh canvas
880         self.canvas.draw()
881
882     def hist(self, combo):
883
884         # clearing old figure
885         self.figure.clear()
886         df = self.DF
887         ax = self.figure.add_subplot(111)
888         if is_numeric_dtype(df[combo]):
889             sns.histplot(data = df[combo], kde=True, ax=ax).set(title=
combo)
890         # refresh canvas
891         self.canvas.draw()
892
893
894     def dispersion(self, combo1, combo2):
895         # clearing old figure
896         self.figure.clear()
897         df = self.DF
898         ax = self.figure.add_subplot(111)
899         if is_numeric_dtype(df[combo1]) and is_numeric_dtype(df[combo2]):
900             ax.scatter(df[combo1], df[combo2])
901             ax.set_title(combo1 + ' ' + combo2)
902         # refresh canvas
903         self.canvas.draw()

```