projet realise par : soufiane sejjari

In [1]:

```
# import libraries
import gspread
import seaborn as sns
from oauth2client.service_account import ServiceAccountCredentials
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
```

In [2]:

```
# initialize variables for gspread
scope = ['https://spreadsheets.google.com/feeds',
'https://www.googleapis.com/auth/drive']
creds = ServiceAccountCredentials.from_json_keyfile_name('sheet.json', scope)
client = gspread.authorize(creds)
# define method to pull data from spreadsheet
def GetSpreadsheetData(sheetName, worksheetIndex):
    sheet = client.open(sheetName).get_worksheet(worksheetIndex)
    return sheet.get_all_values()[1:]
dataTest = GetSpreadsheetData('sheet1', 0)
finale=[]
```

In [5]:

```
def formatData(data,index):
    vare=[]
    vare.clear()
    for i in  range(len(data[1])):

        if i==1:
           if data[index][i]=="Femme":
                data[index][i]=0
           else:
                data[index][i]=1
           vare.append(data[index][i])

        if i==2:
           if data[index][i]=="moins de 18ans":
                data[index][i]=0
           elif data[index][i]=="entre 18 et 25ans":
                data[index][i]=1
           elif data[index][i]=="entre 25 et 35":
                data[index][i]=2
           else:
                data[index][i]=3
           vare.append(data[index][i])

        if i==3:
           vare.append(data[index][i])

        if i==9:
           data[index][i]=data[index][4]+data[index][5]+data[index][6]+data[index][7
           vare.append(data[index][i])

        if i==10:
           if data[index][i]=="plusieurs fois par année":
                data[index][i]=2
           elif data[index][i]=="des fois par année":
                data[index][i]=1
```

```
   elif data[index][i]=="rarement":
        data[index][i]=0
   vare.append(data[index][i])

if i==11:
   if data[index][i]=="très insatisfait":
        data[index][i]=0
   elif data[index][i]=="peu insatisfait":
        data[index][i]=0
   elif data[index][i]=="Ni satisfait ni insatisfait":
        data[index][i]=1
   elif data[index][i]=="Peu satisfait":
        data[index][i]=2
   else:
        data[index][i]=2
   vare.append(data[index][i])

if i==12:
    if 'la foule' in data[index][i]:
        vare.append(1)
    else:

        vare.append(0)
    if 'manque des employés' in data[index][i]:
        vare.append(1)
    else:
        vare.append(0)
    if 'Une mauvaise manière du traitement' in data[index][i]:
        vare.append(1)
    else:
        vare.append(0)



if i==13:
    if data[index][i]=="entre 9 et 11":
        data[index][i]=1
    elif data[index][i]=="entre 11 et 1":
         data[index][i]=2
    elif data[index][i]=="entre 1 et 3":
         data[index][i]=3
    elif data[index][i]=="aprés 3":
            data[index][i]=4
    vare.append(data[index][i])


if i==14:
    if data[index][i]=="lundi":
        data[index][i]=1
    elif data[index][i]=="mardi":
         data[index][i]=2
    elif data[index][i]=="mercredi":
         data[index][i]=3
    elif data[index][i]=="jeudi":
        data[index][i]=4
    elif data[index][i]=="vendredi":
        data[index][i]=5

    vare.append(data[index][i])

if i==15:
   vare.append(data[index][i])

if i==16:
```

```
            vare.append(data[index][i])
    return vare
```

In [6]:

```
for i in range(len(dataTest)):
    vare=[]
    finale.append(formatData(dataTest,i))
```

In [307…

```
t=test['satisfait_score'].index
print(t.values)
```

```
[  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35
  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53
  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71
  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89
  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107
 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179
 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233
 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269
 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287
 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303]
```

In [426…

```
pd.DataFrame(finale).to_csv('projetCommuneN.csv', index_label = "Index", header  = [
```

In [7]:

```
test=pd.read_csv('projetCommune.csv')
test=test.drop(['Index','bureauAvis','amileoration'],axis=1)
test.head()
```

Out[7]:

	sexe	age	province	bureau	visite_score	satisfait_score	la foule	Manque des employé	mauvaise maniere Traitement	les_l
0	0	1	ZOUGHA	ZOUGHA	0	1	1	0	0	
1	0	1	ZOUGHA	ZOUGHA	0	1	0	0	1	
2	1	3	JNAN EL WARD	SAHB LWARD	2	2	0	0	1	
3	1	1	FES-MEDINA	KARAOUIYINE	0	2	0	0	1	
4	0	1	ZOUGHA	ZOUGHA	0	1	1	0	0	

In [70]:

```
model=KNeighborsClassifier(n_neighbors=7)
```

In [63]:

```
y=test['satisfait_score']
x=test.drop(['bureau','province','satisfait_score'],axis=1)
```

In [71]:

```
model.fit(X,Y)
model.score(X,Y)
```

Out[71]:

```
0.94375
```

In [65]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
print('train set:',X_train.shape)
print('test set:',X_test.shape)
```

```
train set: (128, 8)
test set: (32, 8)
```

In [36]:

```
model.fit(X_train,y_train)
model.score(X_train,y_train)
```

Out[36]:

```
0.7396226415094339
```

In [37]:

```
model.score(X_test,y_test)
```

Out[37]:

```
0.5970149253731343
```

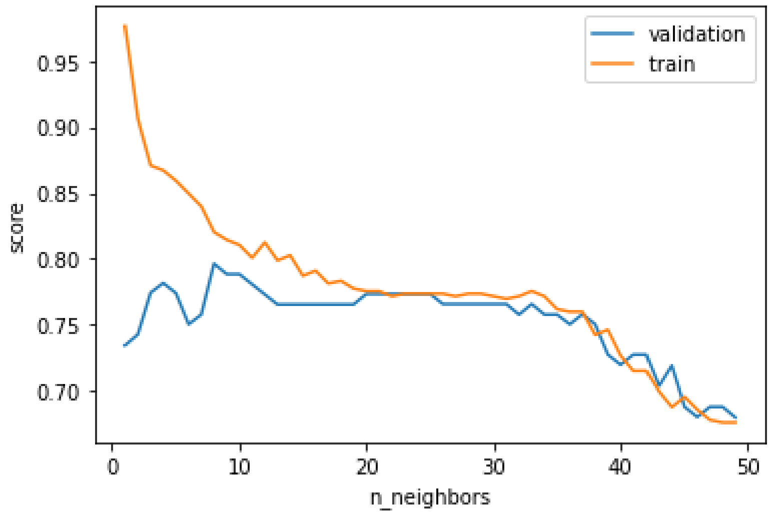
In [72]:

```
from sklearn.model_selection import validation_curve
modele=KNeighborsClassifier()
k=np.arange(1,50)
train_score, val_score=validation_curve(model,X_train,y_train,'n_neighbors', k,cv=5)
plt.plot(k,val_score.mean(axis=1),label="validation")
plt.plot(k,train_score.mean(axis=1),label="train ")
plt.ylabel('score')
plt.xlabel('n_neighbors')
plt.legend()
```

```
C:\Users\pc\anaconda3\lib\site-packages\sklearn\utils\validation.py:70: FutureWarnin
g: Pass param_name=n_neighbors, param_range=[ 1  2  3  4  5  6  7  8  9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
 49] as keyword args. From version 1.0 (renaming of 0.25) passing these as positiona
l arguments will result in an error
  warnings.warn(f"Pass {args_msg} as keyword args. From version "
```

Out[72]:

```
<matplotlib.legend.Legend at 0x214848392b0>
```



In [67]:

```
from sklearn.model_selection import GridSearchCV
estimator.get_params().keys()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9856/3315131551.py in <module>
      1 from sklearn.model_selection import GridSearchCV
----> 2 estimator.get_params().keys()

NameError: name 'estimator' is not defined
```

In [62]:

```
param_grid={'n_neighrbors':np.arange(1,20),
            'metric': ['euclidean','manhattan']}

grid=GridSearchCV(KNeighborsClassifier(),param_grid,cv=5)
grid.fit(X_train,y_train)
```

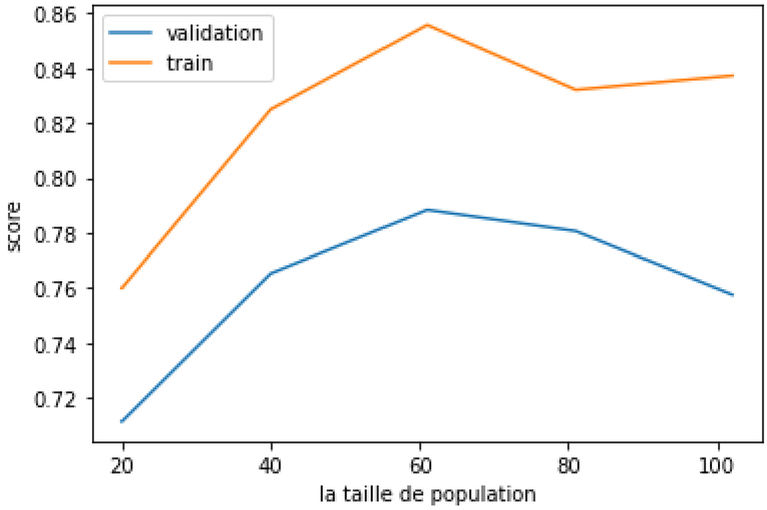
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9856/3653709204.py in <module>
      2             'metric': ['euclidean','manhattan']}
      3
----> 4 grid=GridSearchCV(KNeighborsClassifier(),param_grid,cv=5)
      5 grid.fit(X_train,y_train)

NameError: name 'GridSearchCV' is not defined
```

In [73]:

```
from sklearn.model_selection import learning_curve
N, train_score,val_score=learning_curve(model,X_train,y_train,train_sizes=np.linspac
plt.plot(N,val_score.mean(axis=1),label="validation")
plt.plot(N,train_score.mean(axis=1),label="train ")
plt.ylabel('score')
plt.xlabel('la taille de population')
plt.legend()
```

Out[73]:

```
<matplotlib.legend.Legend at 0x214848c5790>
```



In [57]:

```
from sklearn.feature_selection import VarianceThreshold
```

In [275…

```
X_train.var(axis=0).plot.bar()
```

```
---------------------------------------------------------------------------
```

```
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_22852/1388387244.py in <module>
----> 1 X_train.var(axis=0).plot.bar()

NameError: name 'X_train' is not defined
```

In [85]:

```
from sklearn.cluster import KMeans
modelk=KMeans(n_clusters=3)
```

In [102…

```
modelk.fit(X_train)
modelk.predict(X_train).plot.bar()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_24708/372100047.py in <module>
      1 modelk.fit(X_train)
----> 2 modelk.predict(X_train).plot.bar()

AttributeError: 'numpy.ndarray' object has no attribute 'plot'
```

In [87]:

```
modelk.cluster_centers_
```

Out[87]:

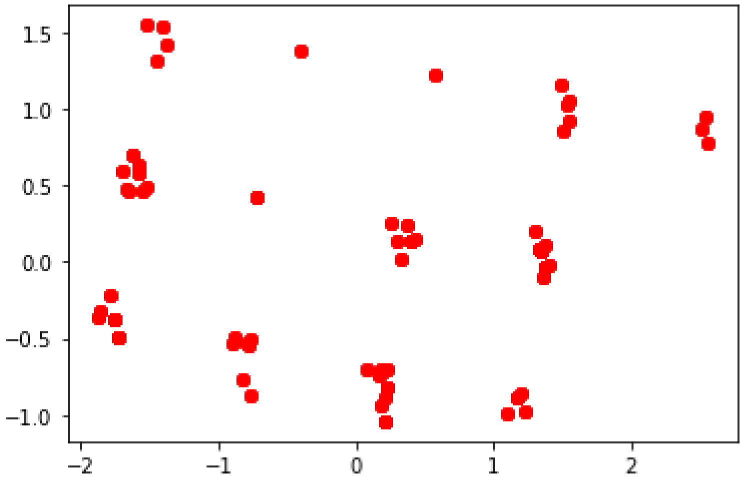
```
array([[0.63809524, 1.14285714, 0.71428571, 0.23809524, 0.36190476,
        0.48571429, 0.94285714],
       [0.57142857, 1.07142857, 0.18571429, 2.22857143, 0.61428571,
        0.25714286, 0.54285714],
       [0.44444444, 1.11111111, 1.32222222, 2.9       , 0.02222222,
        0.36666667, 0.8       ]])
```

In [109…

```
from sklearn.decomposition import PCA
pcaModel=PCA(n_components=0.80)
x_rd=pcaModel.fit_transform(x)
plt.scatter(x_rd[:,0],x_rd[:,1],c='red')
x_rd.shape
```

Out[109…

```
(332, 4)
```



info sur l'analyse

liste de base

- **target variable** : *satisfait_score*
- **type des variables** : qualitative 9: quantitative:4
- **valeur manquants** : 16 dans les horaire, et jourFoule

In [379…

```
r=pd.read_csv('exemple.csv',encoding = "ISO-8859-1")


r.head()
```

Out[379…

	Index	sexe	age	province	bureau	visite_score	satisfait_score	la foule	Manque des employÃ©	mauv man Traitem
0	0	0	1	ZOUGHA	ZOUGHA	0	2	1	0	
1	1	0	1	ZOUGHA	ZOUGHA	0	2	0	0	
2	2	1	3	JNAN EL WARD	SAHB LWARD	2	4	0	0	
3	3	1	1	FES-MEDINA	KARAOUIYINE	0	3	0	0	
4	4	0	1	ZOUGHA	ZOUGHA	0	2	1	0	

In [466…

```
x=test.drop(['province','jourFoule','les_horaires'],axis=1)
x.describe()
```

Out[466…

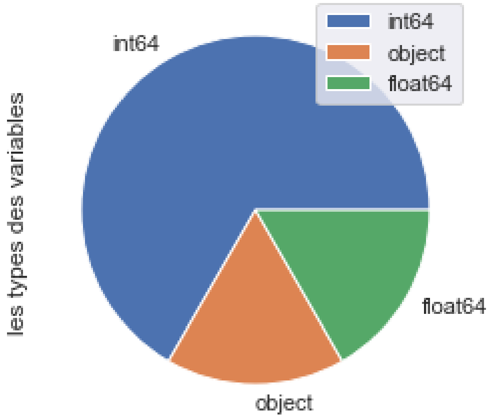
	sexe	age	visite_score	communication	satisfait_score	la foule	Manque des employé
count	336.000000	336.000000	336.000000	336.000000	336.000000	336.000000	336.000000
mean	0.535714	1.119048	0.750000	0.190476	1.154762	0.440476	0.380952
std	0.499467	0.391088	0.722764	0.393262	1.191954	0.497185	0.486345
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1.000000	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000
75%	1.000000	1.000000	1.000000	0.000000	2.000000	1.000000	1.000000
max	1.000000	3.000000	2.000000	1.000000	4.000000	1.000000	1.000000

In [457…

```
test.dtypes.value_counts().plot.pie(label='les types des variables',legend='legend')
```

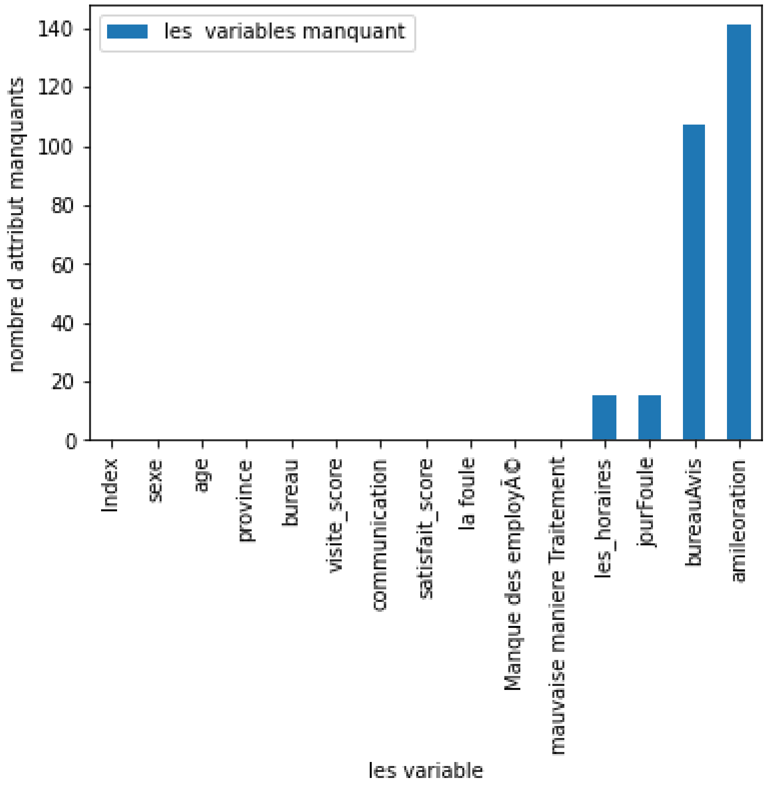
Out[457…

```
<AxesSubplot:ylabel='les types des variables'>
```



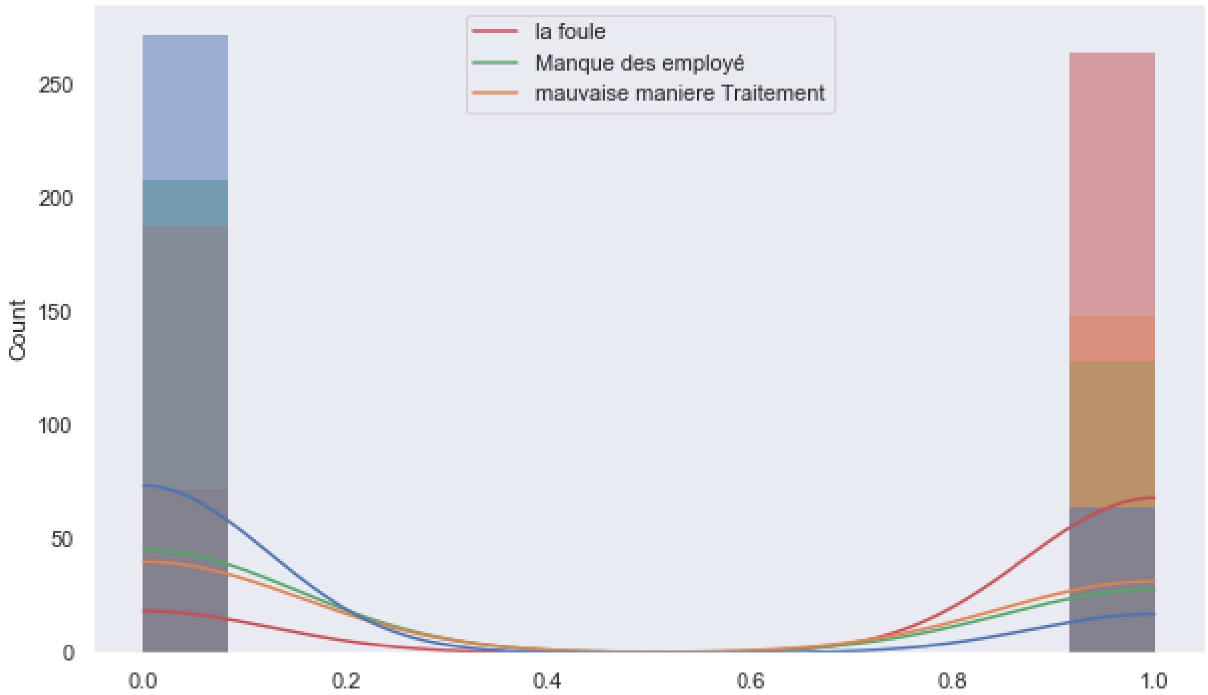
```
In [255…  test.isna().sum().plot.bar(label='les  variables manquant',
          ylabel='nombre d attribut manquants',
          xlabel='les variable',legend="true")
```

```
Out[255…  <AxesSubplot:xlabel='les variable', ylabel='nombre d attribut manquants'>
```



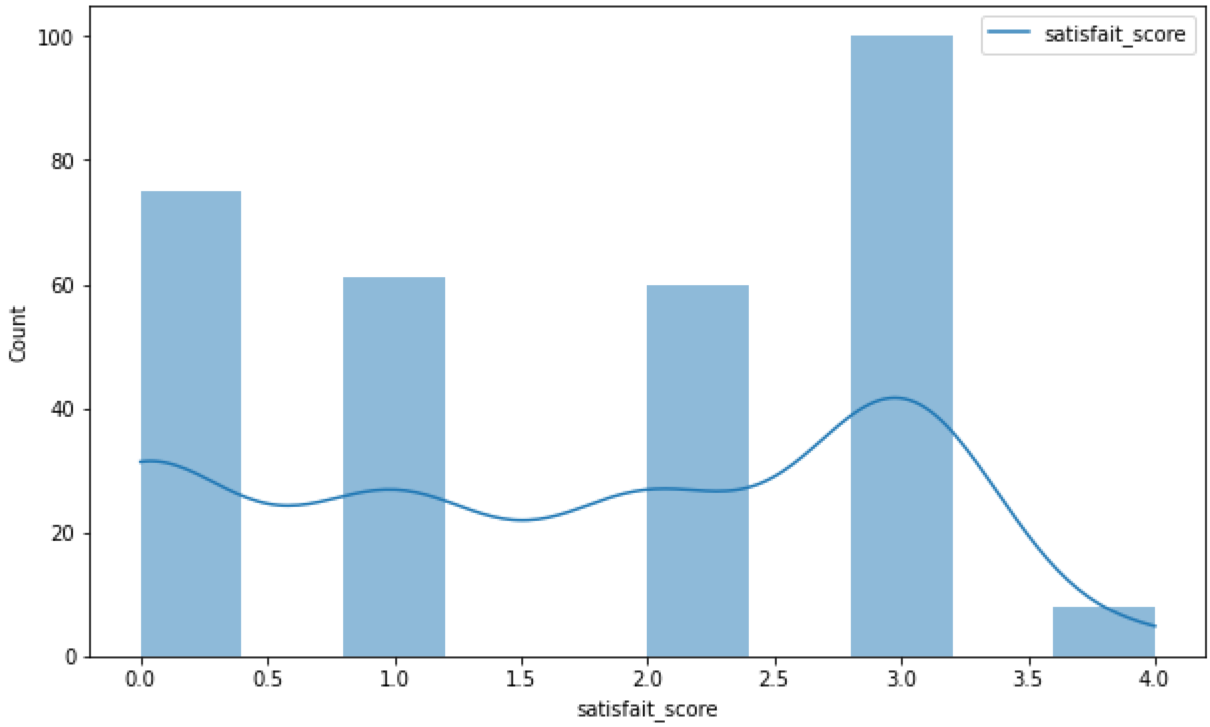
```
In [481…  r=test.drop(['sexe','age','province','bureau','visite_score','satisfait_score','les_
```

```
In [482…  fig = plt.figure(figsize=(10,6))
          sns.histplot(r, kde=True, stat="count",label='rfffffffff', linewidth=0)
          plt.legend(labels=['la foule','Manque des employé','mauvaise maniere Traitement'])
          plt.show()
```



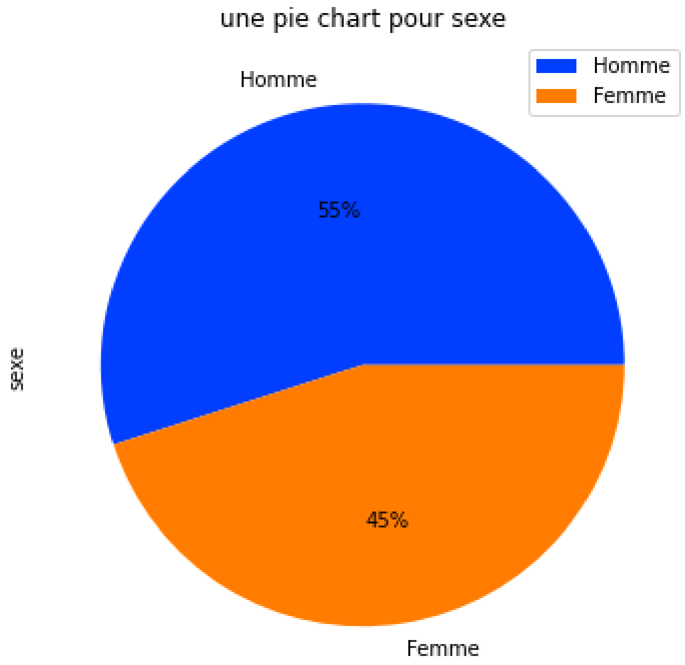
In [258…

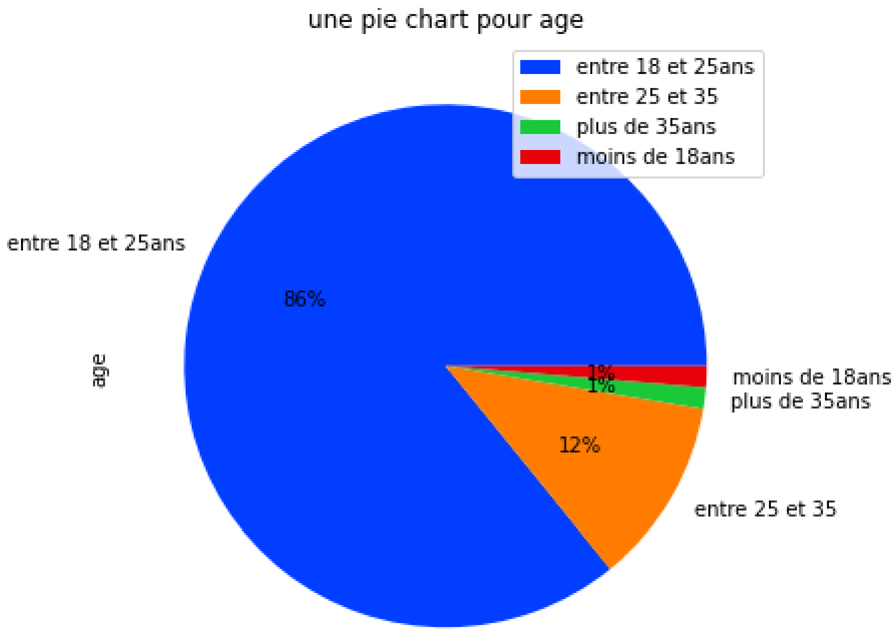
```
for col in test.select_dtypes( 'int') :
    fig = plt.figure(figsize=(10,6))
    sns.histplot(test[col], kde=True, stat="count",label='rfffffffff', linewidth=0)
    plt.legend(labels=[col])
    sns.color_palette("Paired")
```

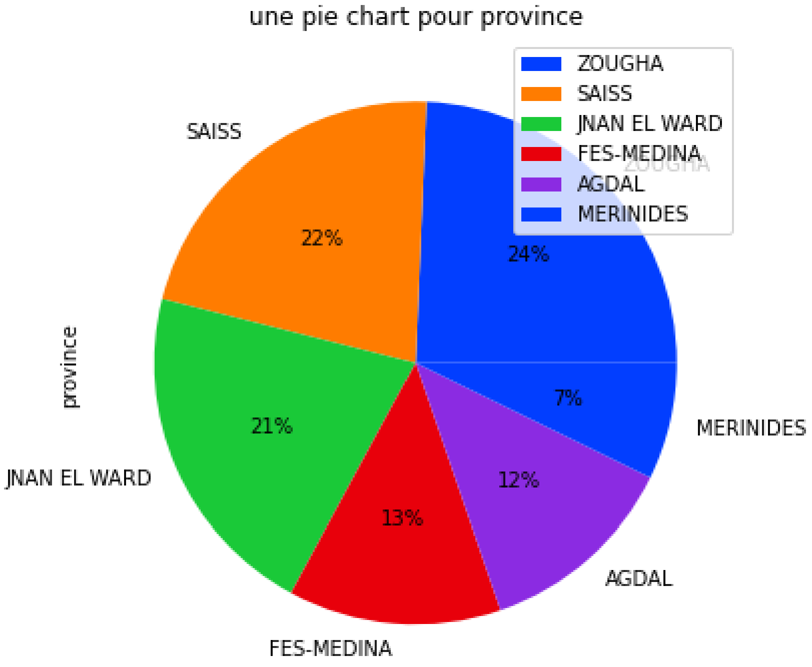


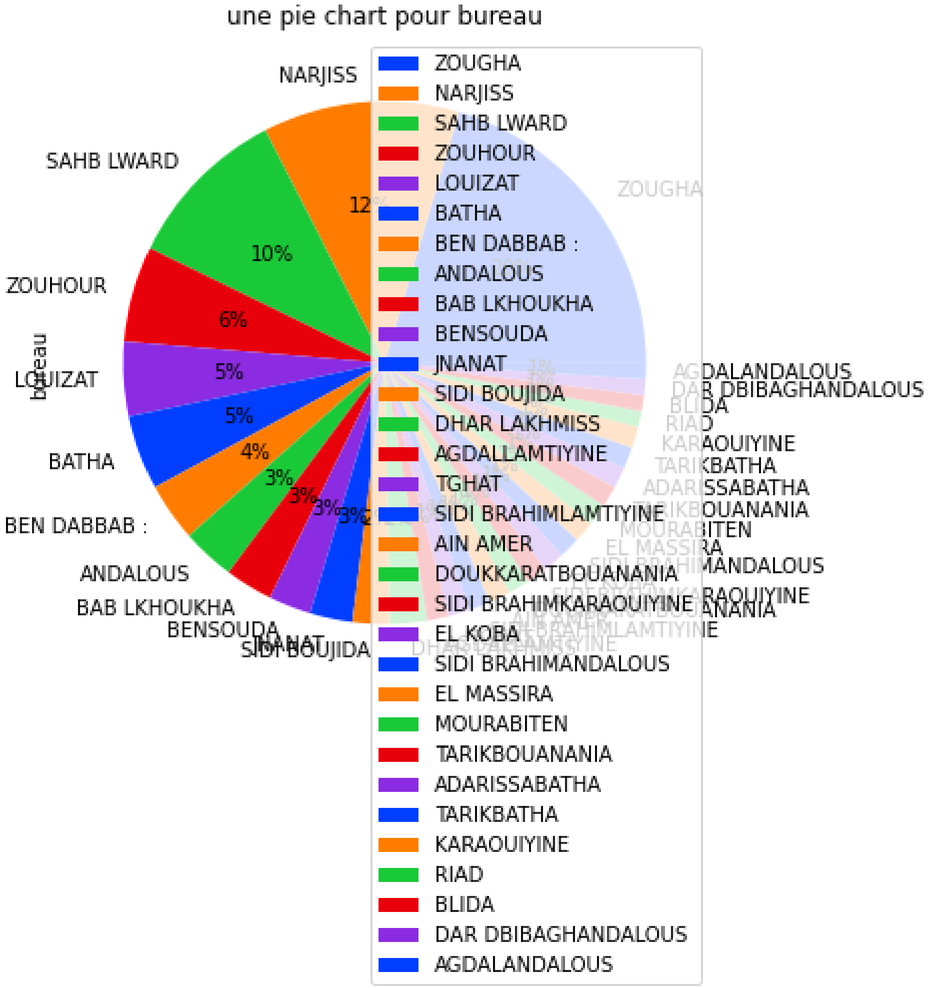
In [259…

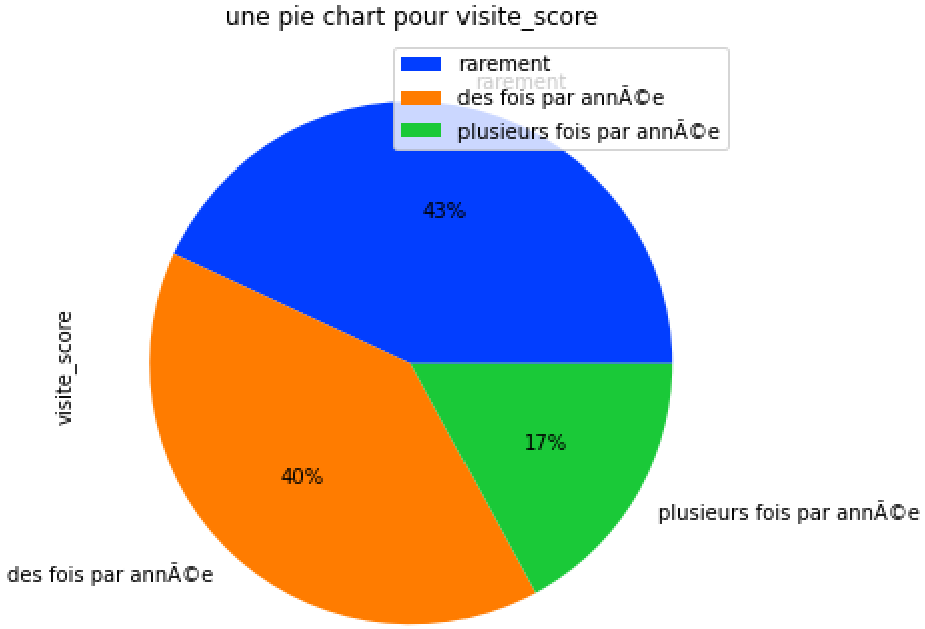
```
for col in test.select_dtypes('object') :
    fig = plt.figure(figsize=(10,6))
    colors = sns.color_palette('bright')[0:5]
    test[col].value_counts().plot.pie(colors=colors,autopct='%.0f%%').set(title="une
    plt.legend(labels=test[col].value_counts().index)
```

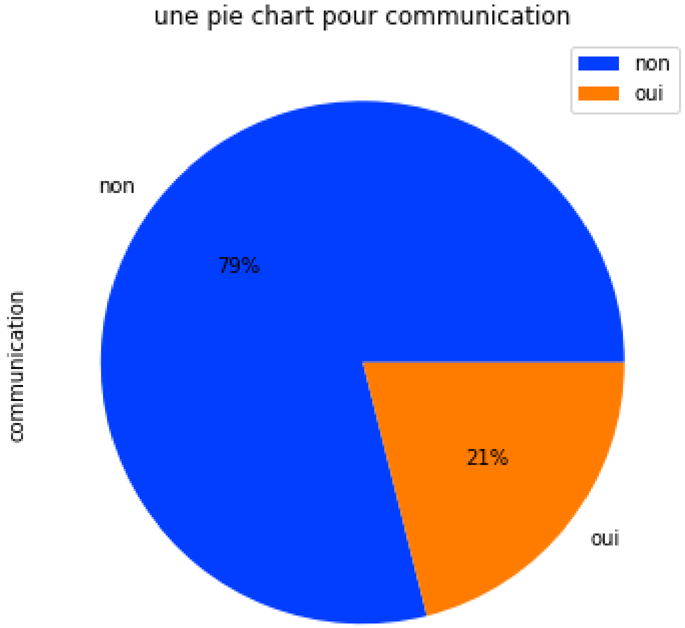

une pie chart pour sexe
Homme
Femme
Homme
55%
sexe
45%
Femme

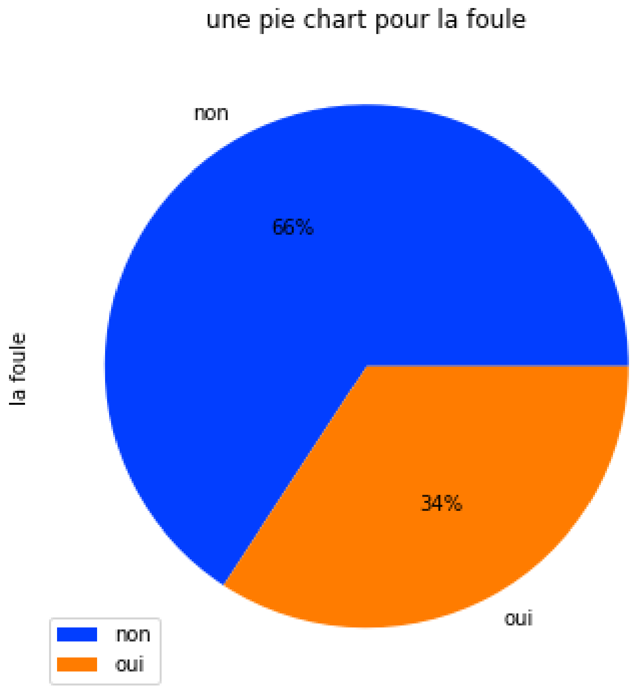

une pie chart pour age
entre 18 et 25ans
entre 25 et 35
plus de 35ans
moins de 18ans
entre 18 et 25ans
86%
age
1%
1%
moins de 18ans
plus de 35ans
12%
entre 25 et 35

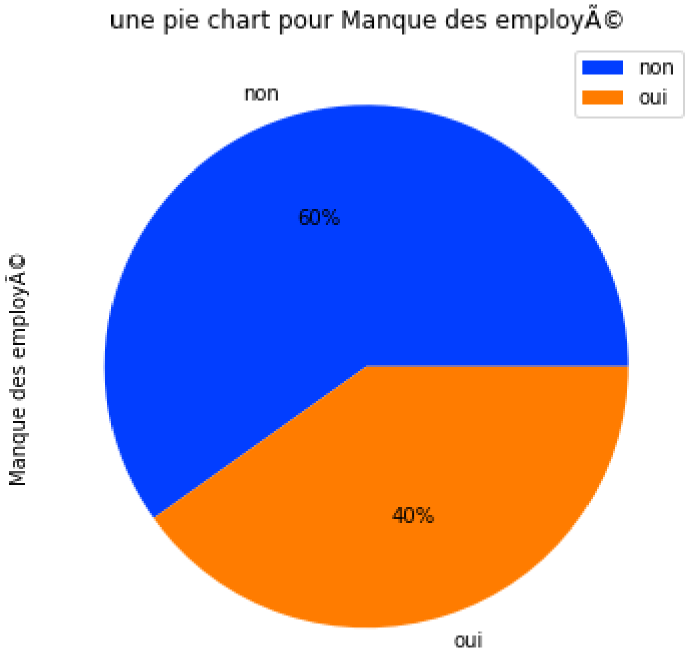

une pie chart pour province
province
ZOUGHA
SAISS
JNAN EL WARD
FES-MEDINA
AGDAL
MERINIDES
24%
22%
21%
13%
12%
7%
SAISS
JNAN EL WARD
FES-MEDINA
AGDAL
MERINIDES
ZOUGHA

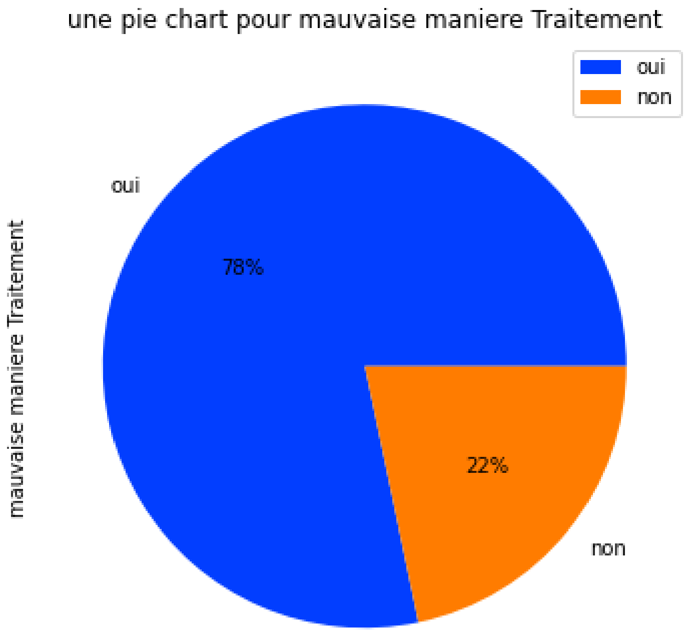

une pie chart pour bureau
bureau
ZOUGHA
NARJISS
SAHB LWARD
ZOUHOUR
LOUIZAT
BATHA
BEN DABBAB :
ANDALOUS
BAB LKHOUKHA
BENSOUDA
JNANAT
SIDI BOUJIDA
DHAR LAKHMISS
AGDALLAMTIYINE
TGHAT
SIDI BRAHIMLAMTIYINE
AIN AMER
DOUKKARATBOUANANIA
SIDI BRAHIMKARAOUIYINE
EL KOBA
SIDI BRAHIMANDALOUS
EL MASSIRA
MOURABITEN
TARIKBOUANANIA
ADARISSABATHA
TARIKBATHA
KARAOUIYINE
RIAD
BLIDA
DAR DBIBAGHANDALOUS
AGDALANDALOUS
12%
10%
6%
5%
5%
4%
3%
3%
3%
3%
NARJISS
SAHB LWARD
ZOUHOUR
LOUIZAT
BATHA
BEN DABBAB :
ANDALOUS
BAB LKHOUKHA
BENSOUDA
JNANAT
SIDI BOUJIDA
ZOUGHA
AGDALANDALOUS
DAR DBIBAGHANDALOUS
BLIDA
RIAD
KARAOUIYINE
TARIKBATHA
ADARISSABATHA
TARIKBOUANANIA
MOURABITEN
EL MASSIRA
SIDI BRAHIMANDALOUS
SIDI BRAHIMKARAOUIYINE
DOUKKARATBOUANANIA
AIN AMER
SIDI BRAHIMLAMTIYINE
DHAR LAKHMISS

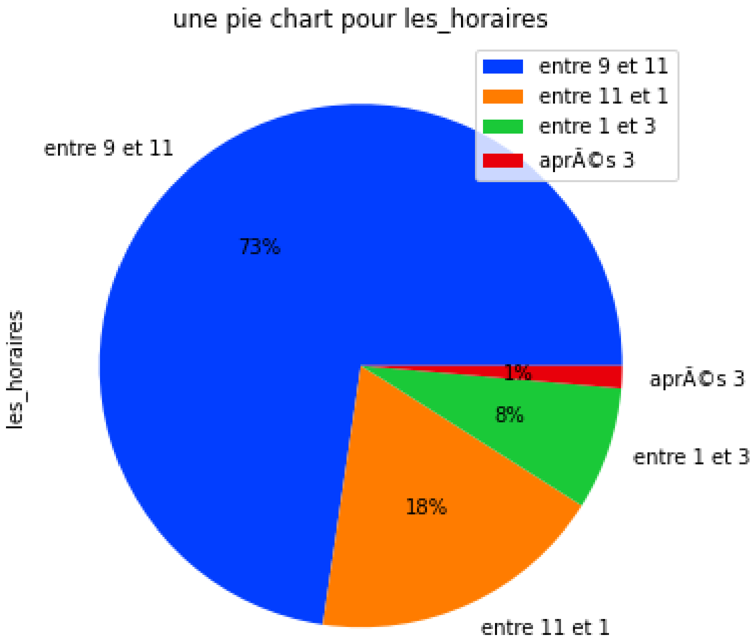

une pie chart pour visite_score
rarement
des fois par annÃ©e
plusieurs fois par annÃ©e
rarement
43%
visite_score
40%
17%
plusieurs fois par annÃ©e
des fois par annÃ©e

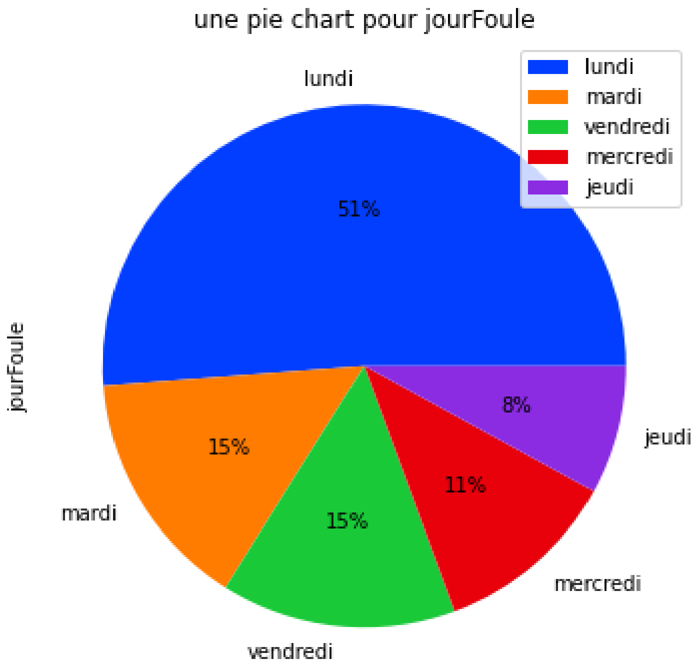

une pie chart pour communication
non
oui
non
79%
communication
21%
oui


une pie chart pour la foule
la foule
non
66%
34%
oui
non
oui


une pie chart pour Manque des employÃ©
Manque des employÃ©
non
oui
non
60%
40%
oui


une pie chart pour mauvaise maniere Traitement
oui
non
mauvaise maniere Traitement
oui
78%
22%
non


une pie chart pour les_horaires
entre 9 et 11
entre 11 et 1
entre 1 et 3
aprÃ©s 3
entre 9 et 11
73%
les_horaires
1%
aprÃ©s 3
8%
entre 1 et 3
18%
entre 11 et 1



In [321…

```
    satisfait_s=satisfait[col].value_counts()
    satisfait_is=insatisfait[col].value_counts()


    df = pd.DataFrame({
        'satisfait': satisfait_s,
        'insatisfait': satisfait_is
    })
    print(df.index['Index'])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_22852/3524556222.py in <module>
      7     'insatisfait': satisfait_is
      8 })
----> 9 print(df.index['Index'])

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in __getitem__(self, key)
   4602         if is_scalar(key):
   4603             key = com.cast_scalar_indexer(key, warn_float=True)
-> 4604             return getitem(key)
   4605 
   4606         if isinstance(key, slice):

IndexError: only integers, slices (`:`), ellipsis (`...`), numpy.newaxis (`None`) an
d integer or boolean arrays are valid indices
```

Relation

In [428…

In [261…

```
test['province'].value_counts().decreabe
```

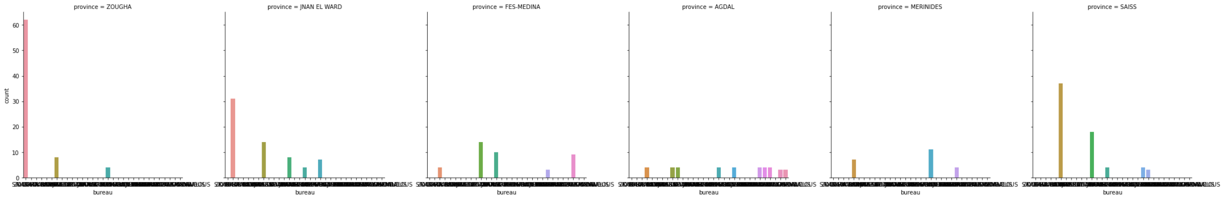
Out[261…

```
ZOUGHA          74
SAISS           66
JNAN EL WARD    64
```

```
FES-MEDINA      40
AGDAL           38
MERINIDES       22
Name: province, dtype: int64
```

In [291…

```
sss=sns.catplot(data=test,x="bureau",col="province",
             kind="count");
```



In [262…

```
ZOUGHA=test[test['province']=='ZOUGHA']


SAISS=test[test['province']=='SAISS']
jnan_el_ward=test[test['province']=='JNAN EL WARD']


FES_MEDINA=test[test['province']=='FES-MEDINA']
AGDAL=test[test['province']=='AGDAL']


MERINIDES=test[test['province']=='MERINIDES']
```

In [492…

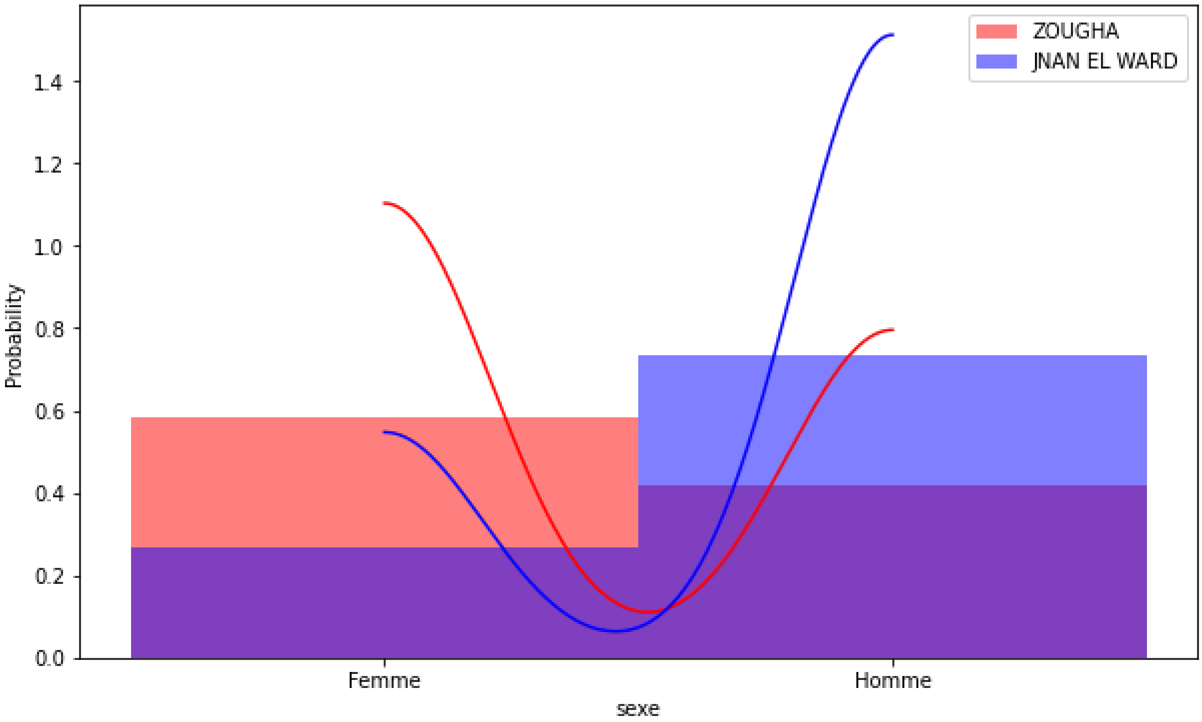
```
ZOUGHA.describe()
```

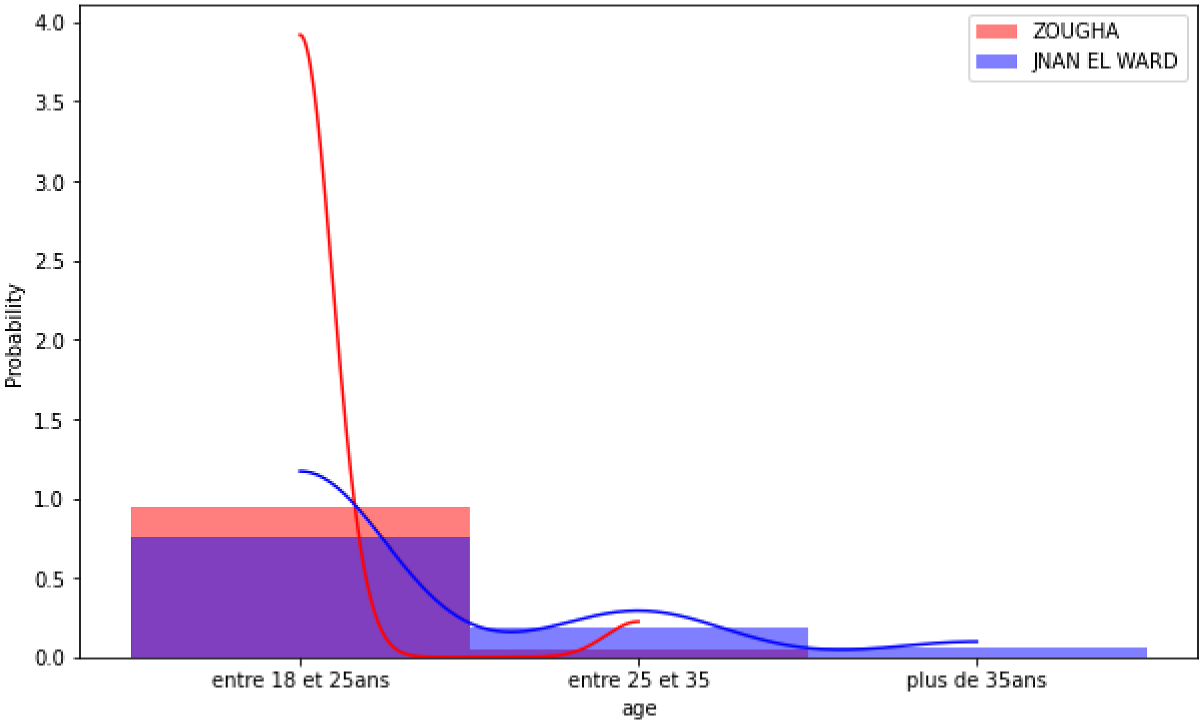
Out[492…

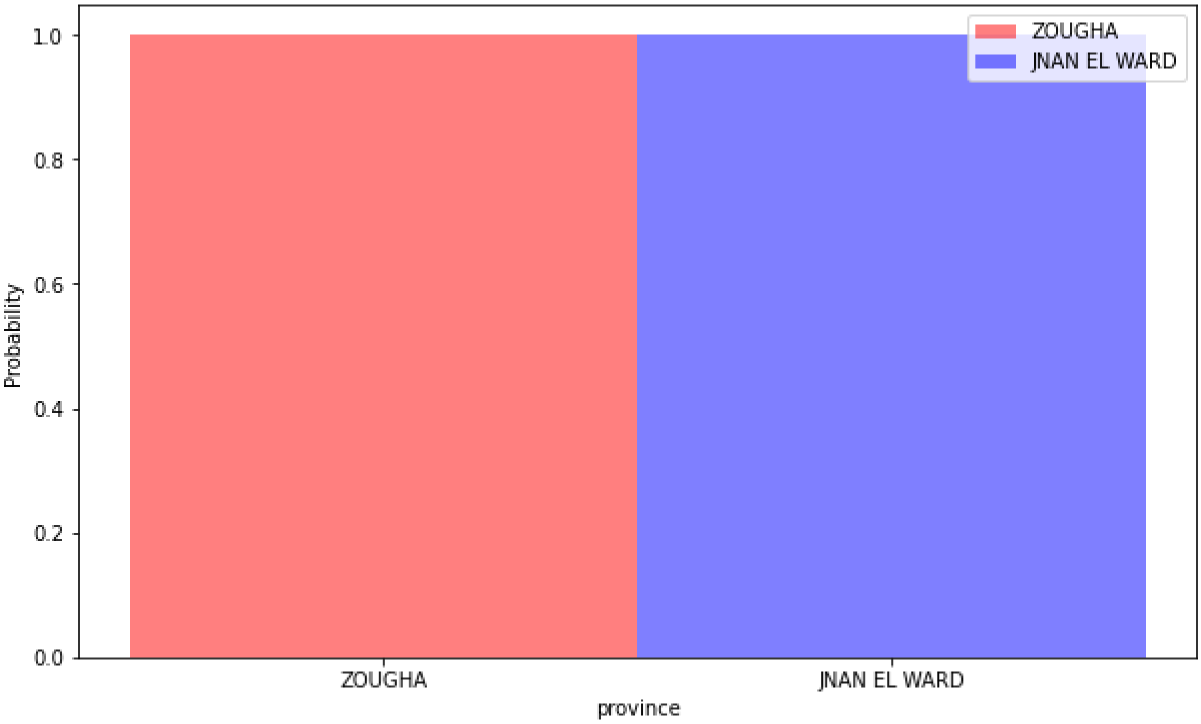
	satisfait_score
count	74.000000
mean	2.040541
std	1.243495
min	0.000000
25%	2.000000
50%	2.000000
75%	3.000000
max	4.000000

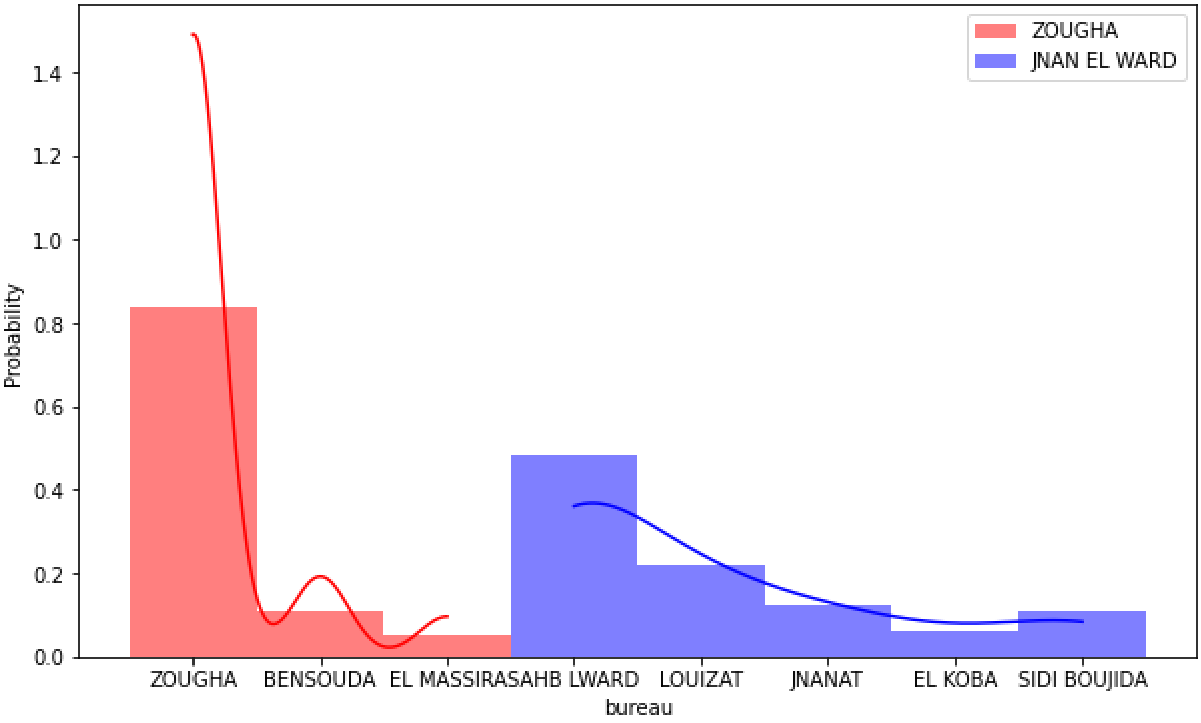
In [289…

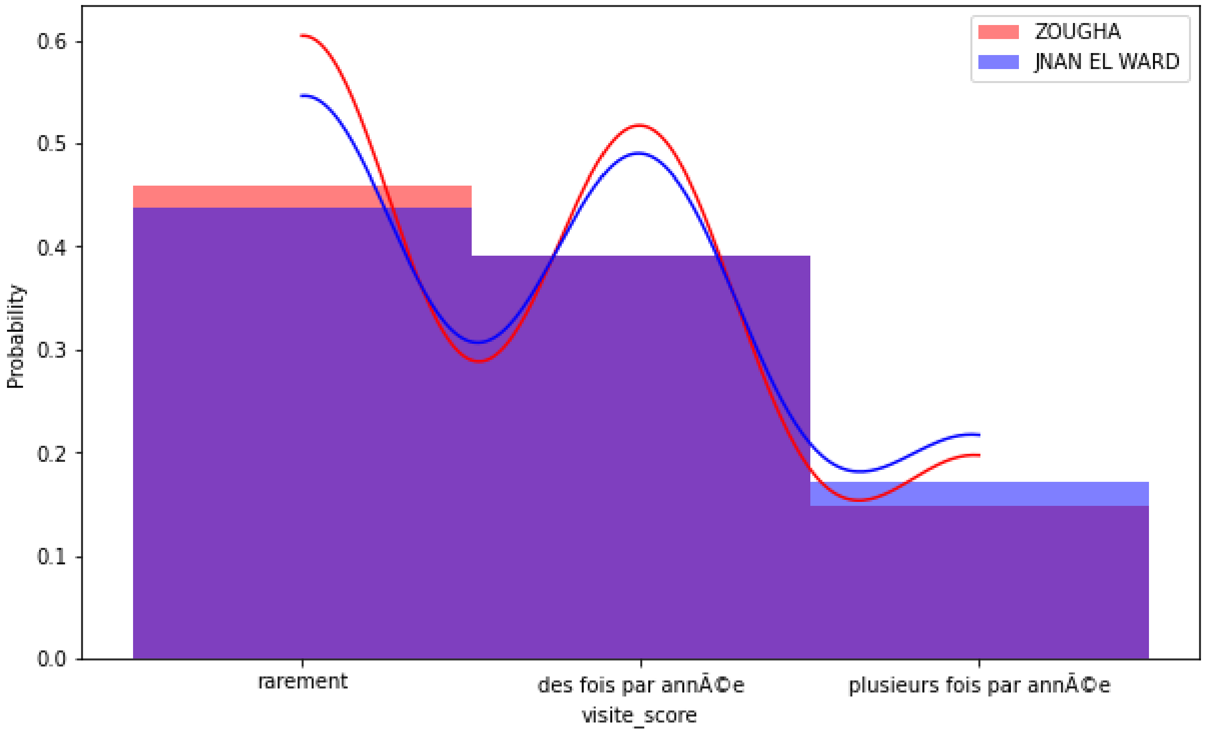
```
for col in test:
    fig = plt.figure(figsize=(10,6))
    colors = sns.color_palette('bright')

    sns.histplot(ZOUGHA[col], kde=True, stat="probability",label='ZOUGHA',linewidth=
    sns.histplot(jnan_el_ward[col], kde=True, stat="probability",label='JNAN EL WARD

    sns.set_palette("Paired")
    plt.legend()
```

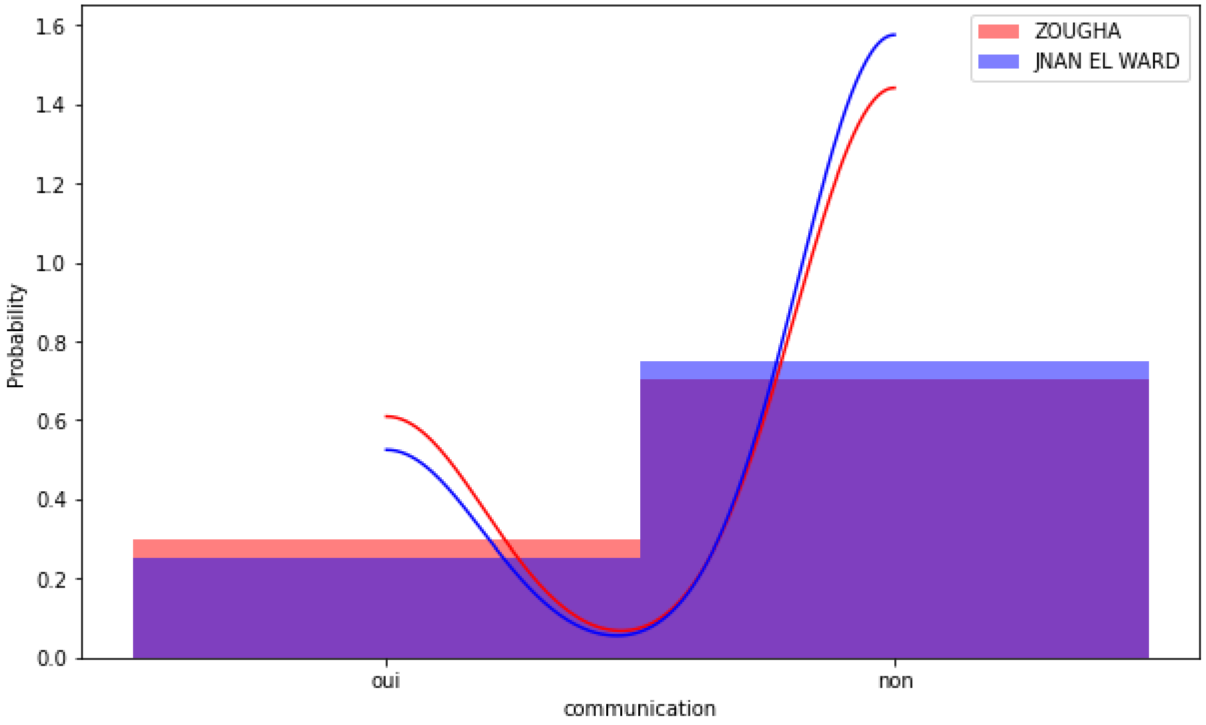

ZOUGHA
JNAN EL WARD
Probability
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Femme
Homme
sexe

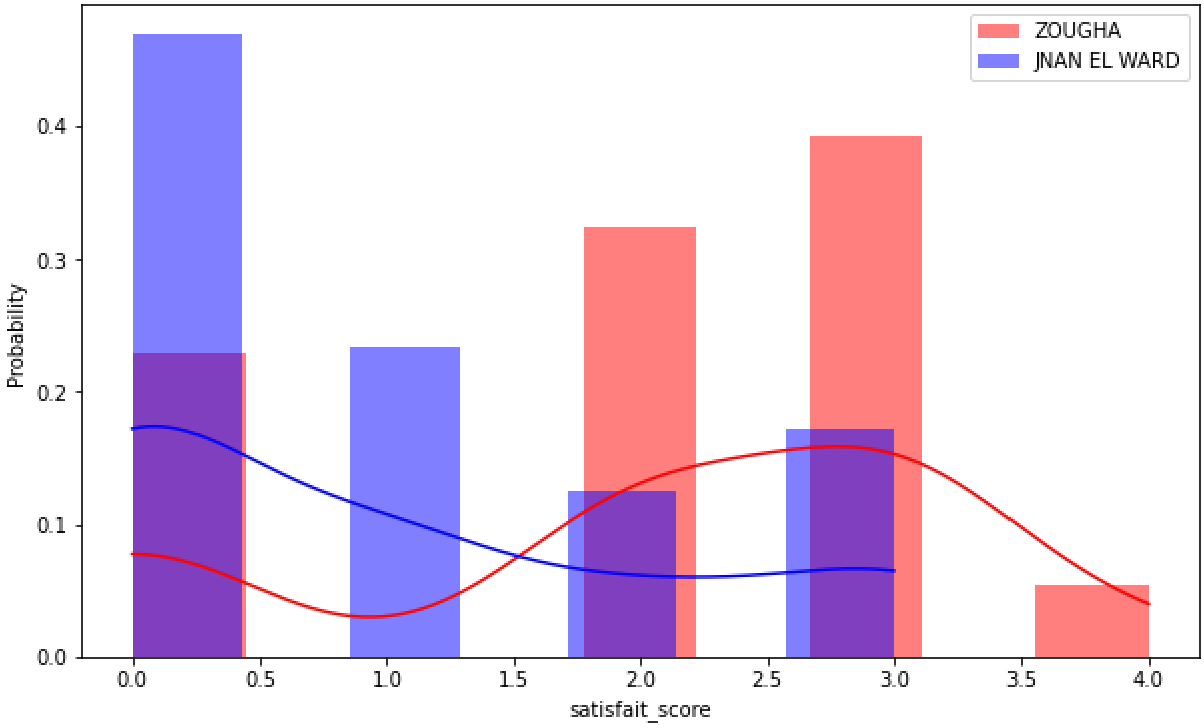

ZOUGHA
JNAN EL WARD
Probability
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
entre 18 et 25ans
entre 25 et 35
plus de 35ans
age

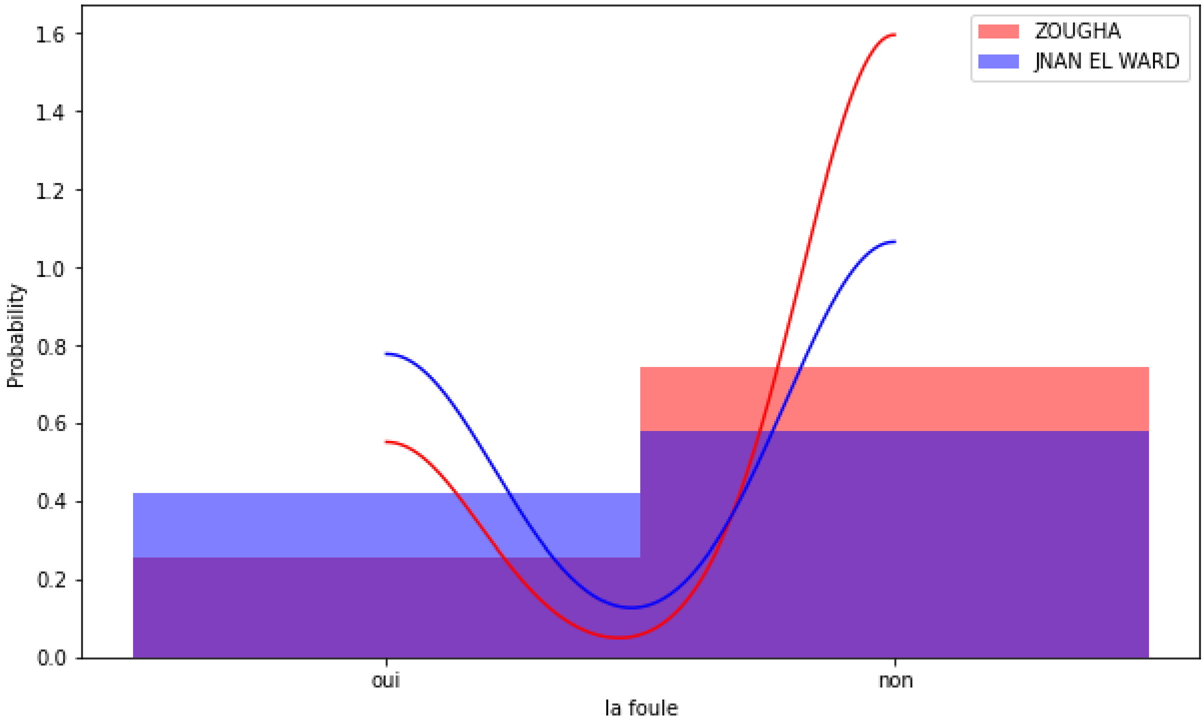

ZOUGHA
JNAN EL WARD
Probability
1.0
0.8
0.6
0.4
0.2
0.0
ZOUGHA
JNAN EL WARD
province

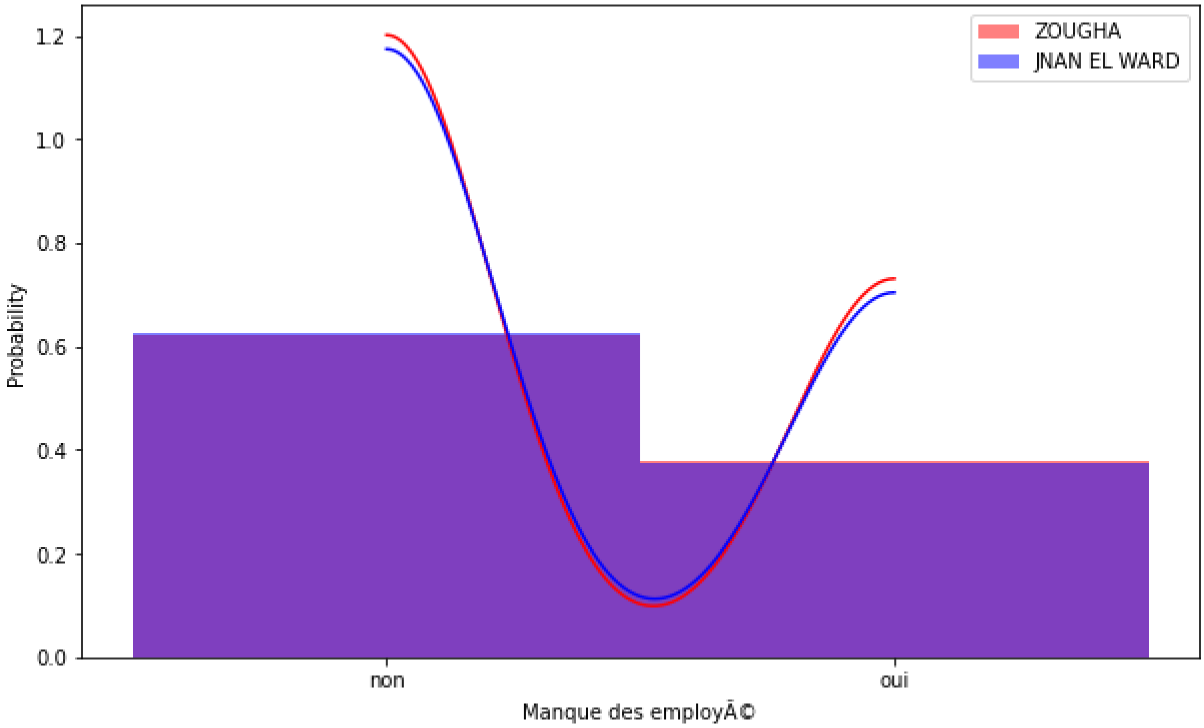

ZOUGHA
JNAN EL WARD
Probability
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
ZOUGHA
BENSOUDA
EL MASSIRASAHB LWARD
LOUIZAT
JNANAT
EL KOBA
SIDI BOUJIDA
bureau

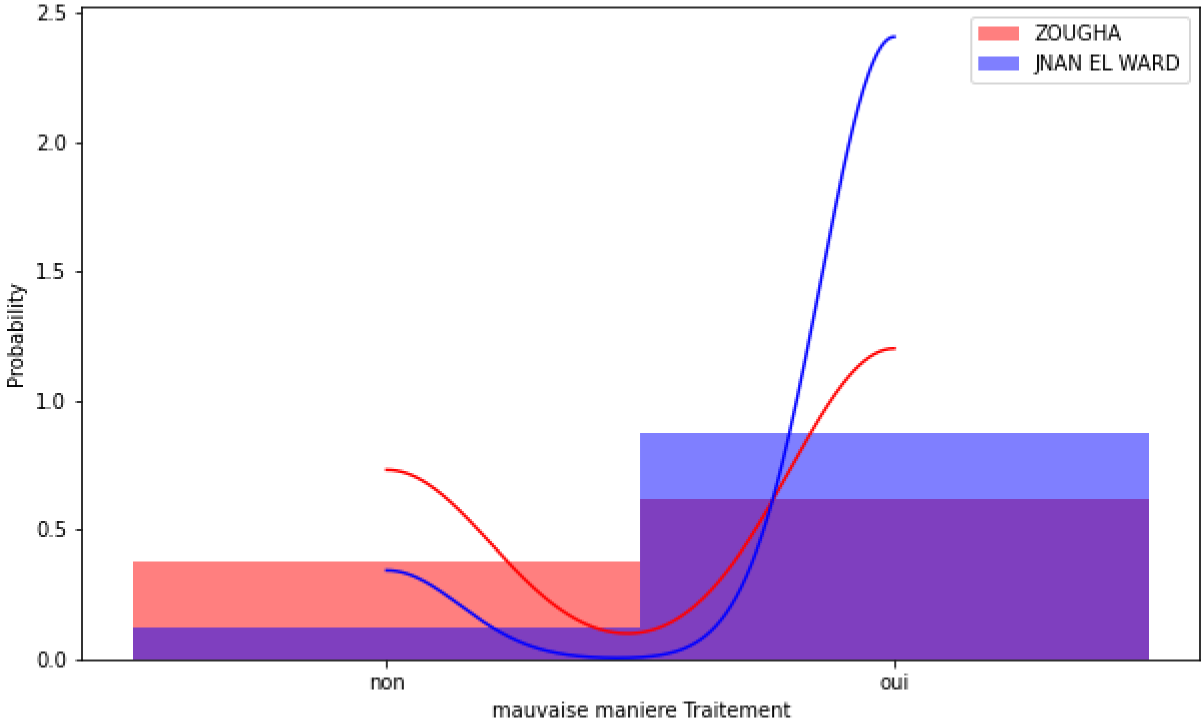

ZOUGHA
JNAN EL WARD
Probability
0.0
0.1
0.2
0.3
0.4
0.5
0.6
rarement
des fois par annÃ©e
plusieurs fois par annÃ©e
visite_score

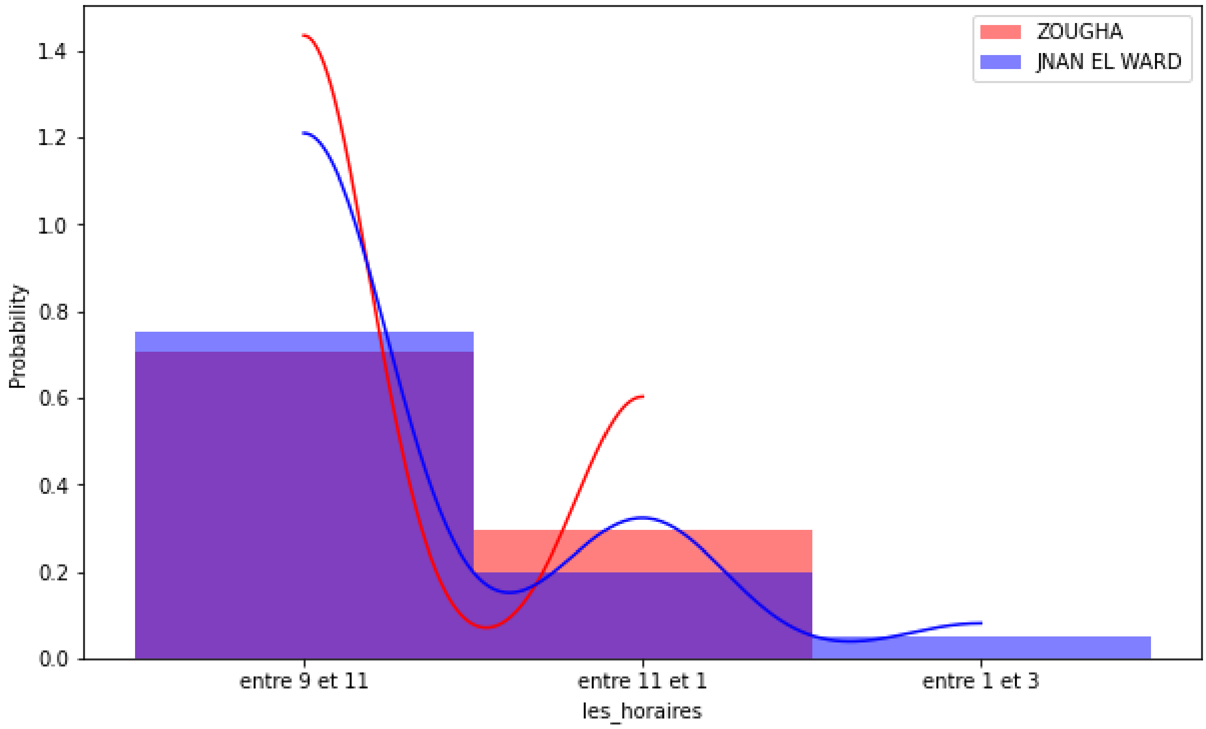

ZOUGHA
JNAN EL WARD
Probability
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
oui
non
communication

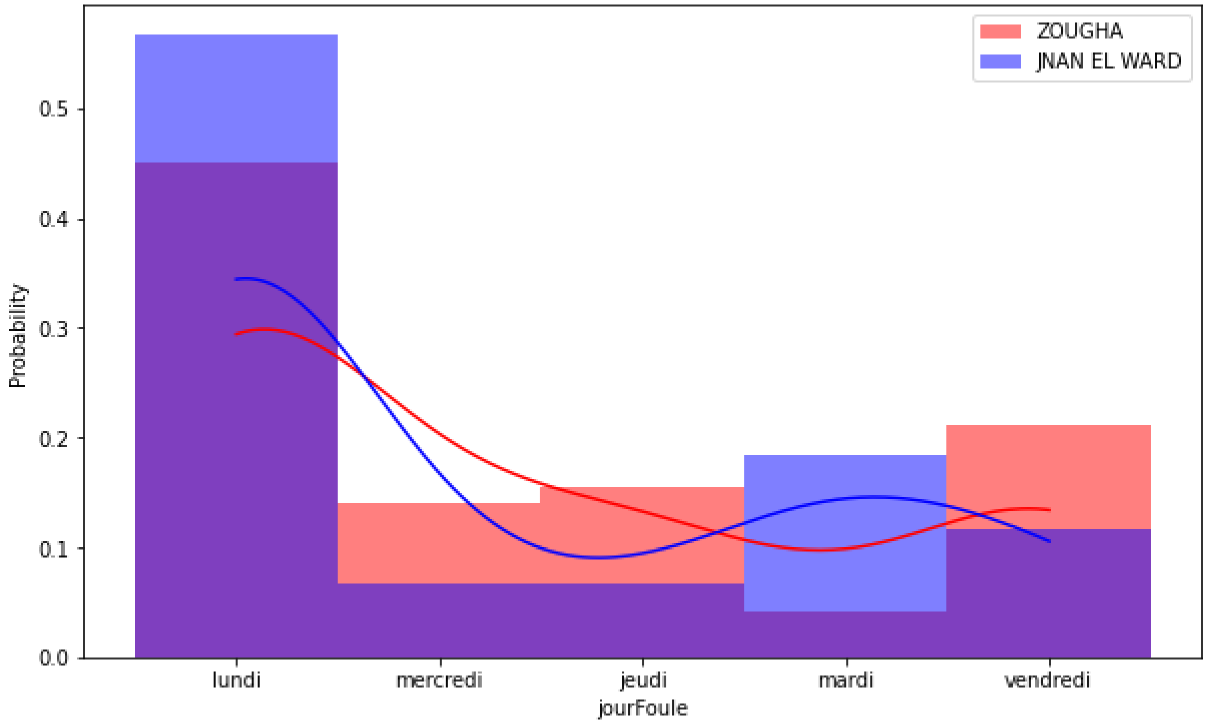

ZOUGHA
JNAN EL WARD
Probability
0.0
0.1
0.2
0.3
0.4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
satisfait_score


ZOUGHA
JNAN EL WARD
Probability
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
oui
non
la foule


ZOUGHA
JNAN EL WARD
Probability
0.0
0.2
0.4
0.6
0.8
1.0
1.2
non
oui
Manque des employÃ©


ZOUGHA
JNAN EL WARD
Probability
0.0
0.5
1.0
1.5
2.0
2.5
non
oui
mauvaise maniere Traitement





In [476…

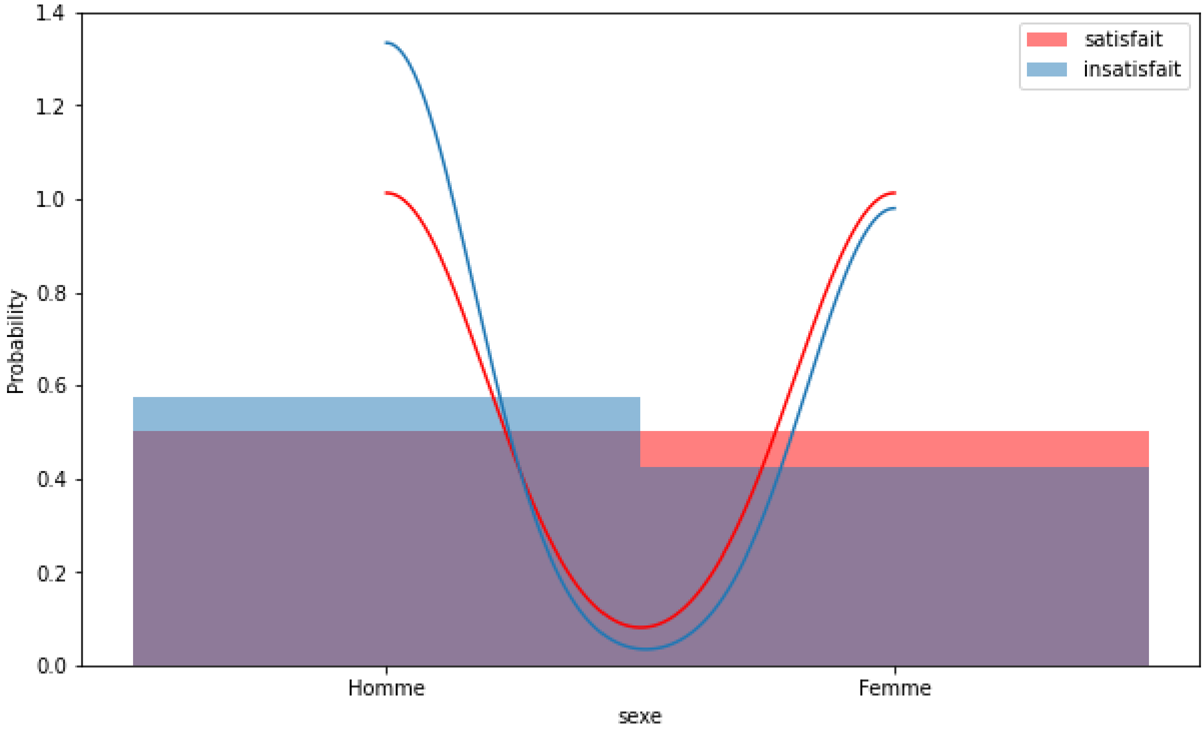
```
data=test.groupby('visite_score').sum()['la foule','Manque des employé','mauvaise ma
```

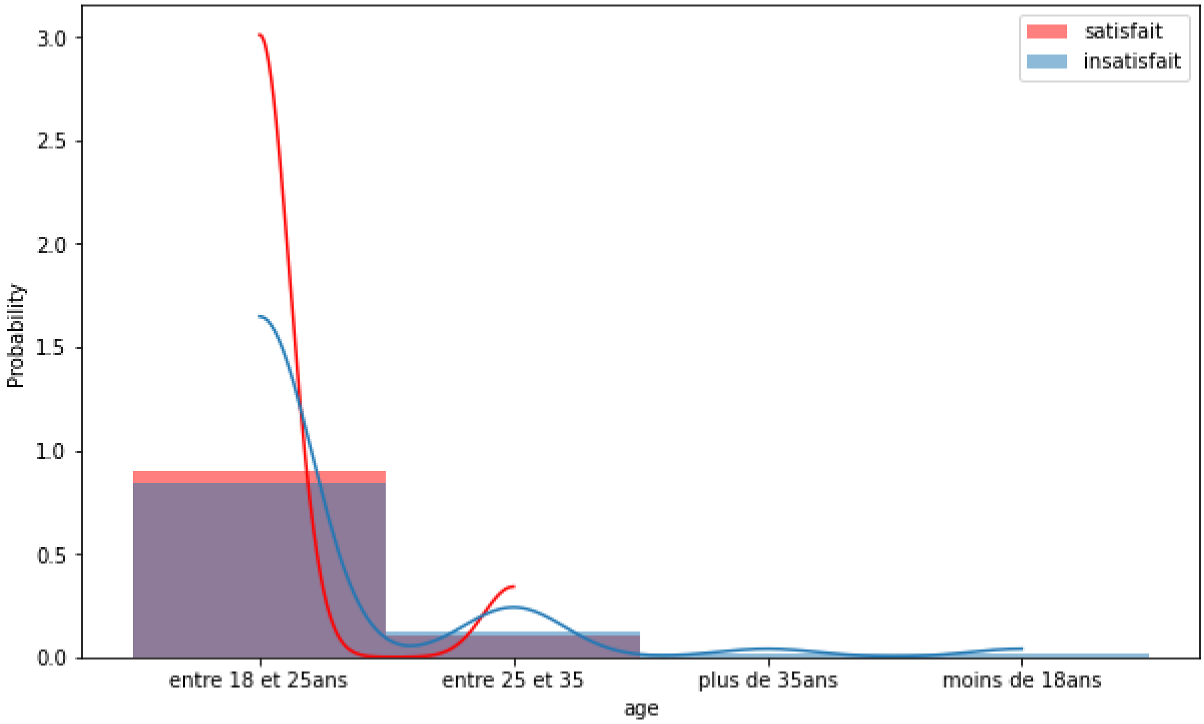
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, meth
od, tolerance)
   3360             try:
-> 3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in pandas._libs.index.IndexEngi
ne.get_loc()

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in pandas._libs.index.IndexEngi
ne.get_loc()
```

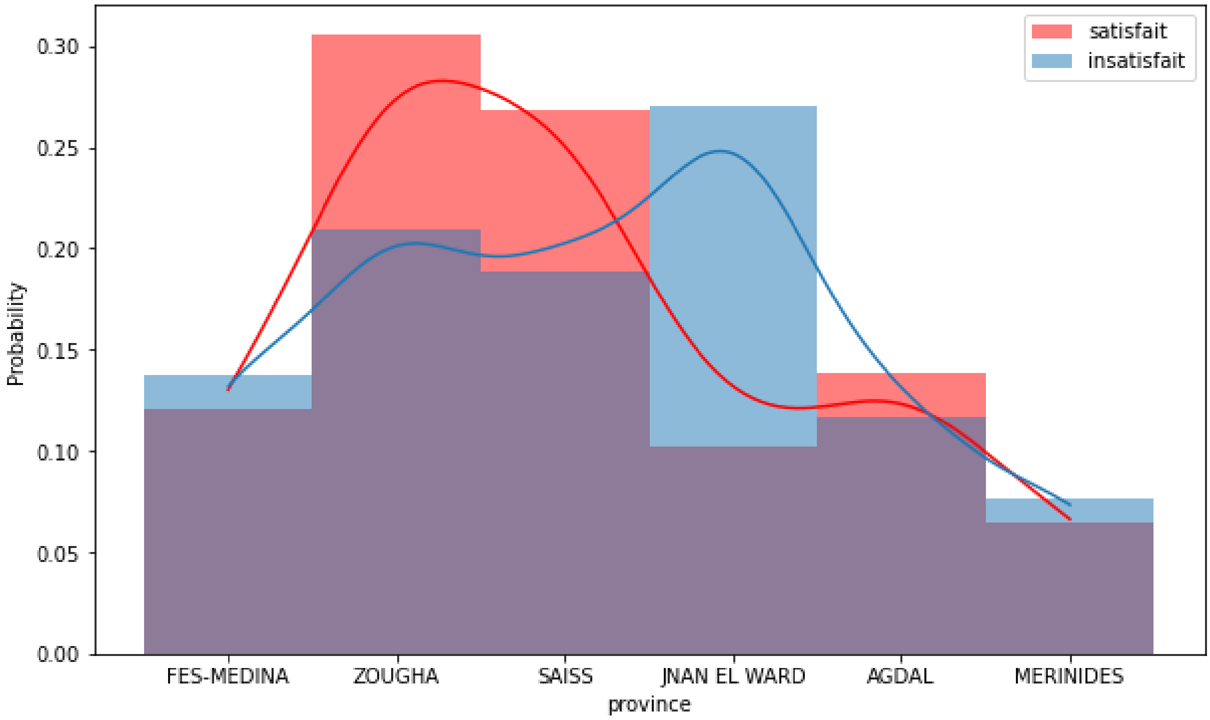
```
pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.
get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.
get_item()

KeyError: ('la foule', 'Manque des employé', 'mauvaise maniere Traitement')

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_22852/4223410551.py in <module>
----> 1 data=test.groupby('visite_score').sum()['la foule','Manque des employé','mau
vaise maniere Traitement'].plot.bar()

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
   3456             if self.columns.nlevels > 1:
   3457                 return self._getitem_multilevel(key)
-> 3458             indexer = self.columns.get_loc(key)
   3459             if is_integer(indexer):
   3460                 indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, meth
od, tolerance)
   3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:
-> 3363                 raise KeyError(key) from err
   3364 
   3365         if is_scalar(key) and isna(key) and not self.hasnans:

KeyError: ('la foule', 'Manque des employé', 'mauvaise maniere Traitement')
```

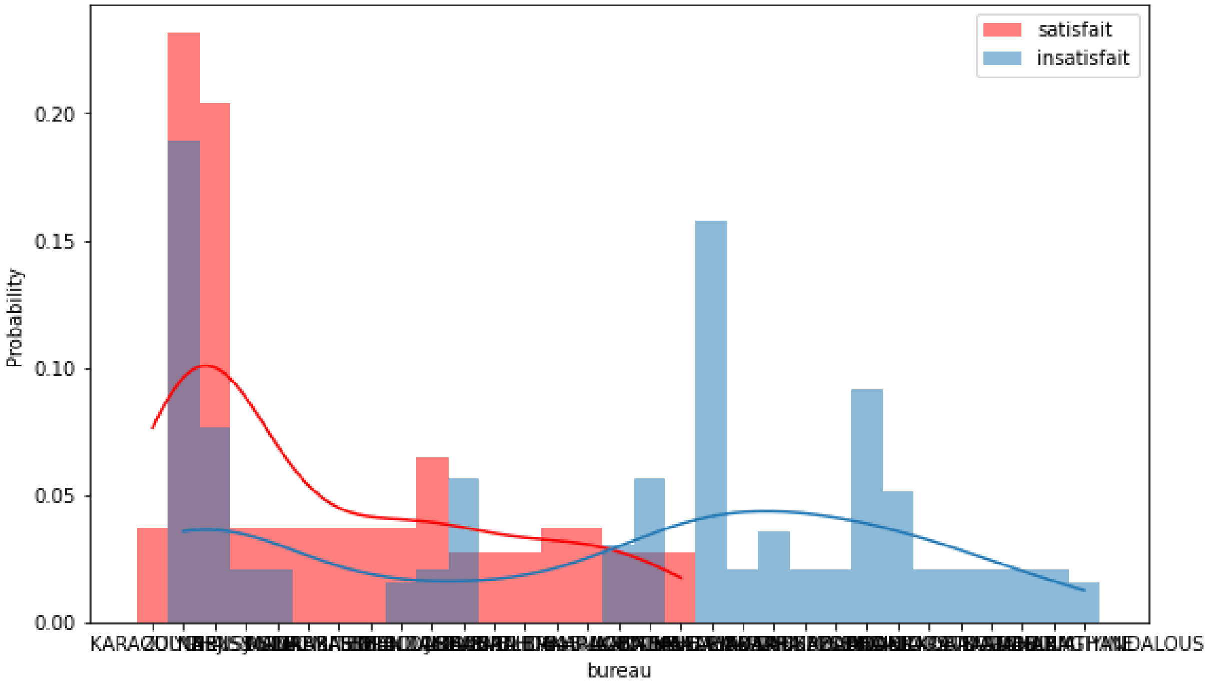
In [263…

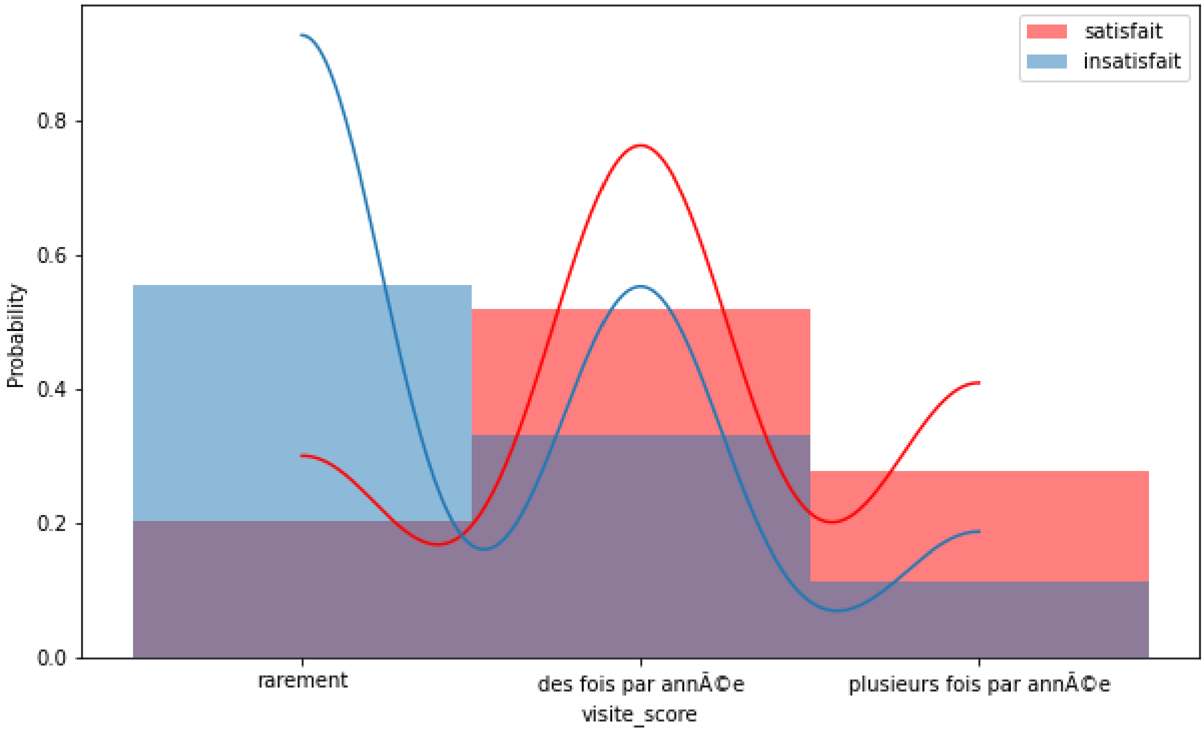
```
for col in test:
    fig = plt.figure(figsize=(10,6))
    colors = sns.color_palette('bright')[0:5]

    sns.histplot(satisfait[col], kde=True, stat="probability",label='satisfait',line
    sns.histplot(insatisfait[col], kde=True, stat="probability",label='insatisfait',

    plt.legend()
    plt.colors=colors
```

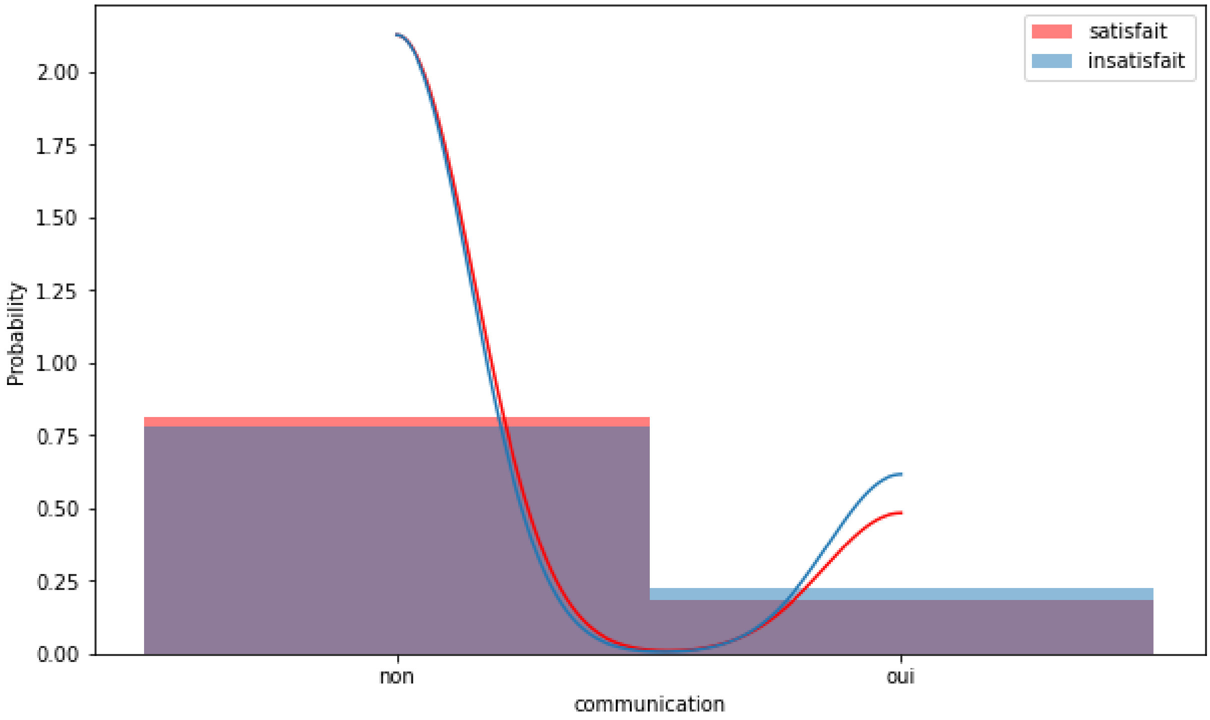

1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Probability
satisfait
insatisfait
Homme
Femme
sexe

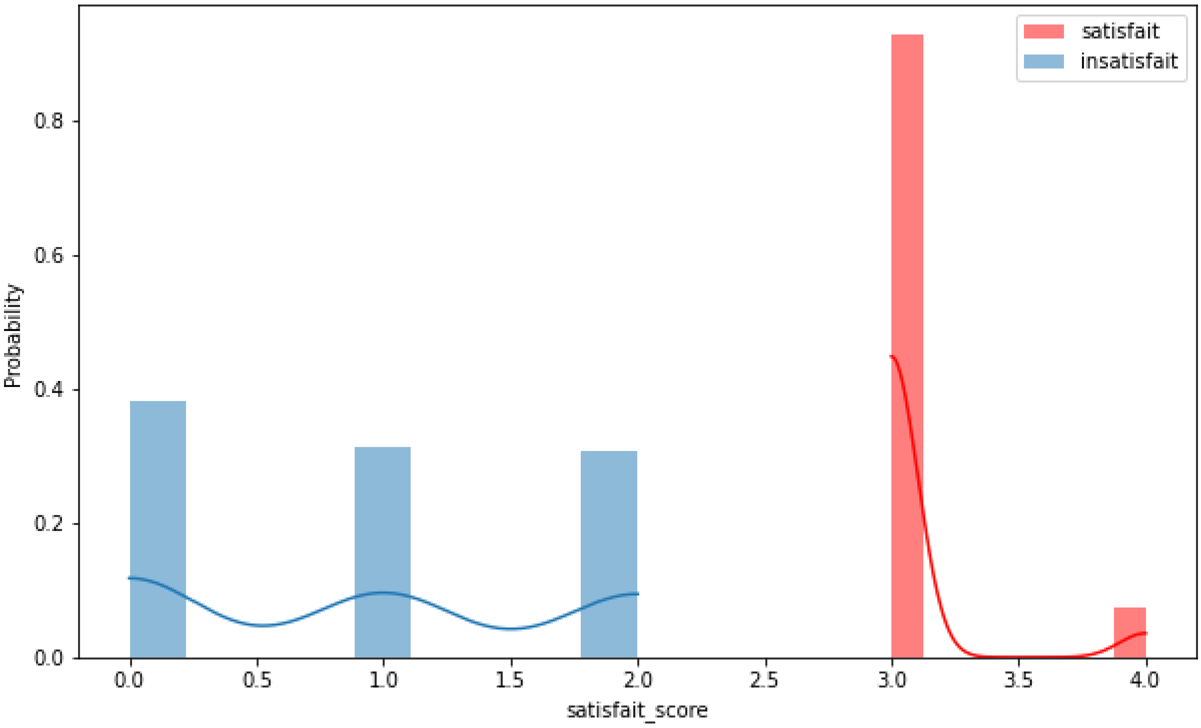

3.0
2.5
2.0
1.5
1.0
0.5
0.0
Probability
satisfait
insatisfait
entre 18 et 25ans
entre 25 et 35
plus de 35ans
moins de 18ans
age

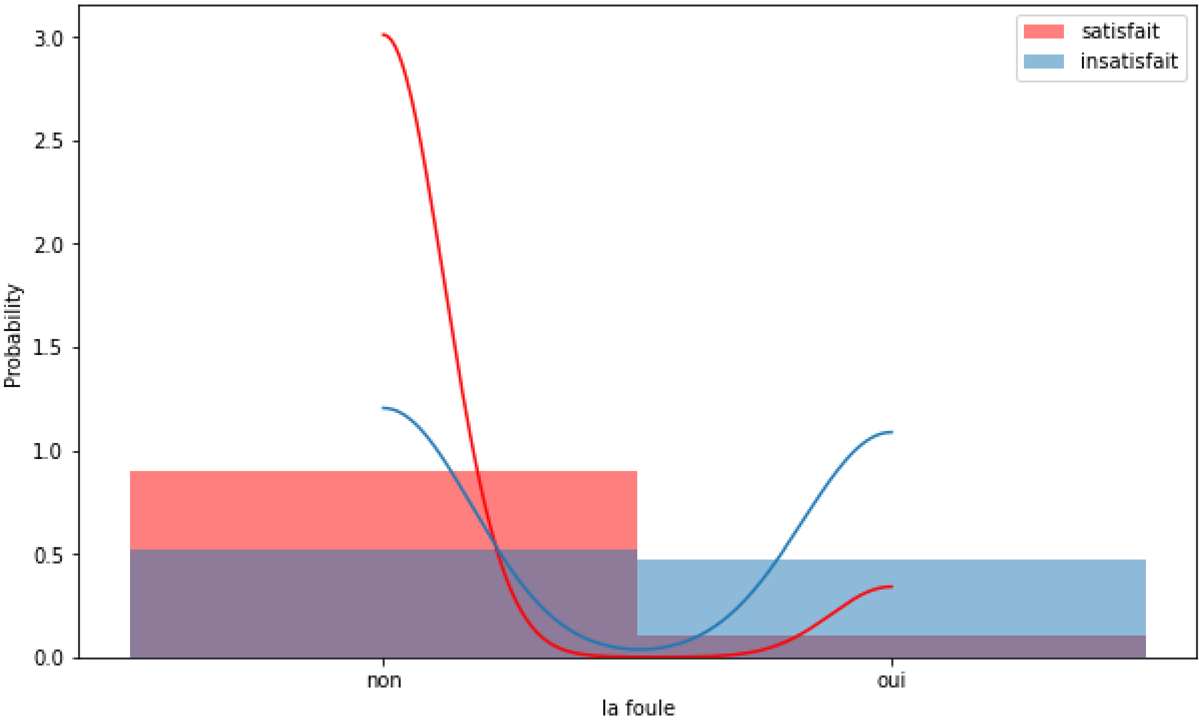

satisfait
insatisfait
Probability
0.00
0.05
0.10
0.15
0.20
0.25
0.30
FES-MEDINA
ZOUGHA
SAISS
JNAN EL WARD
AGDAL
MERINIDES
province

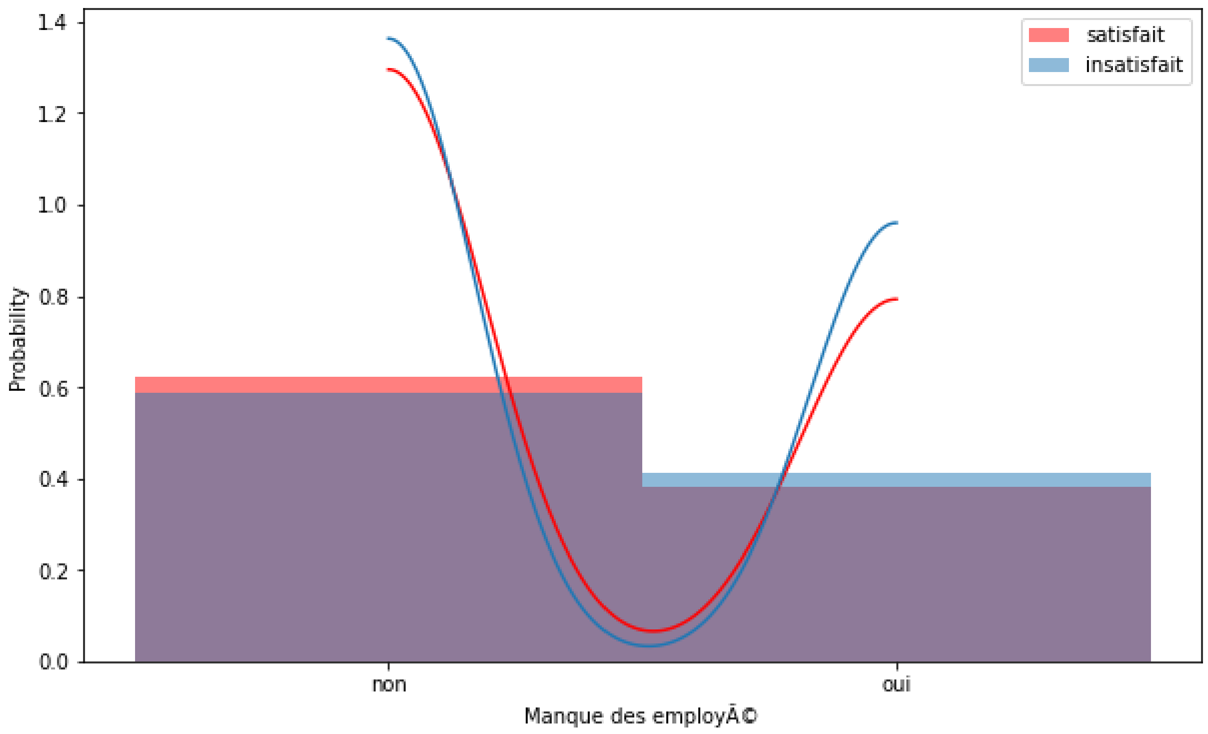

satisfait
insatisfait
Probability
0.00
0.05
0.10
0.15
0.20
bureau


satisfait
insatisfait
Probability
0.0
0.2
0.4
0.6
0.8
rarement
des fois par annÃ©e
plusieurs fois par annÃ©e
visite_score

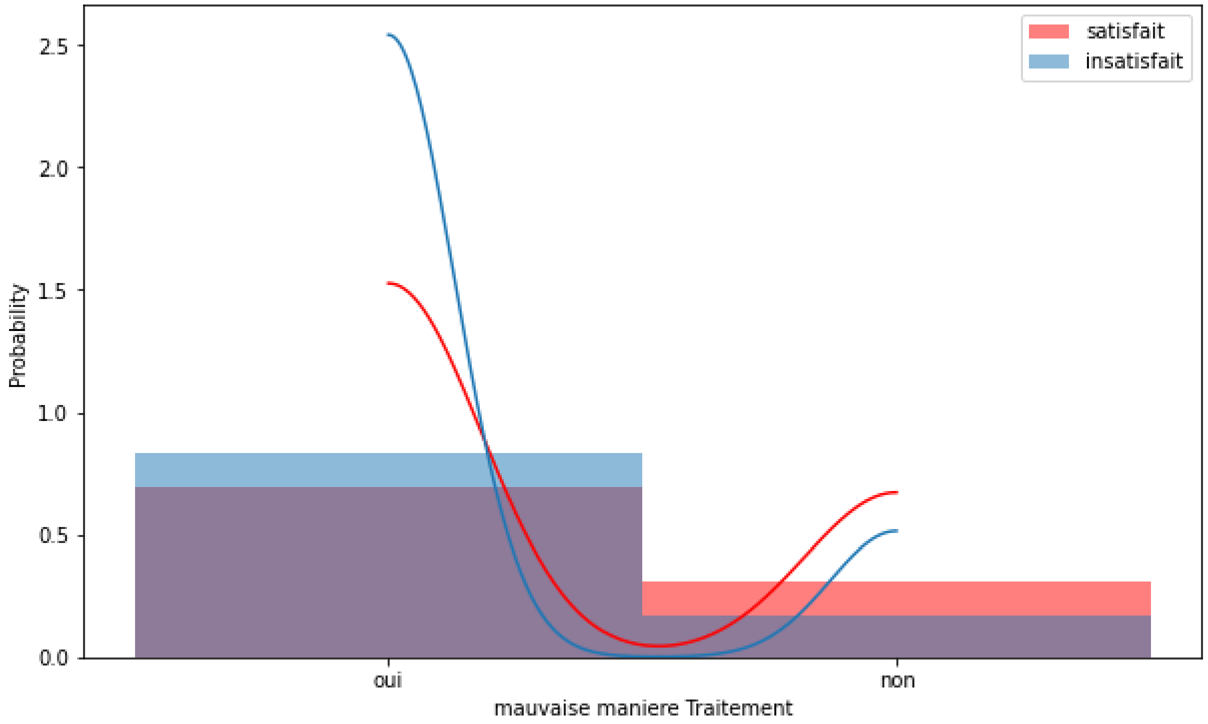

satisfait
insatisfait
Probability
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
non
oui
communication


satisfait
insatisfait
Probability
0.0
0.2
0.4
0.6
0.8
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
satisfait_score

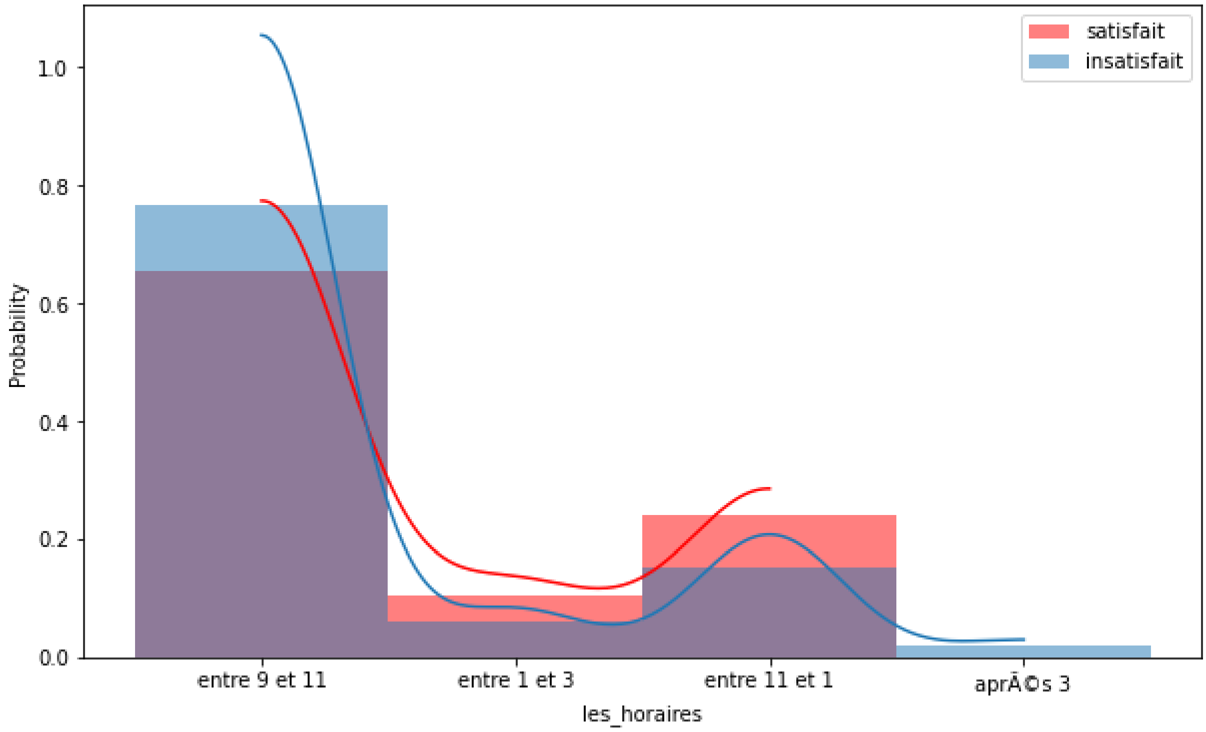

satisfait
insatisfait
Probability
0.0
0.5
1.0
1.5
2.0
2.5
3.0
non
oui
la foule

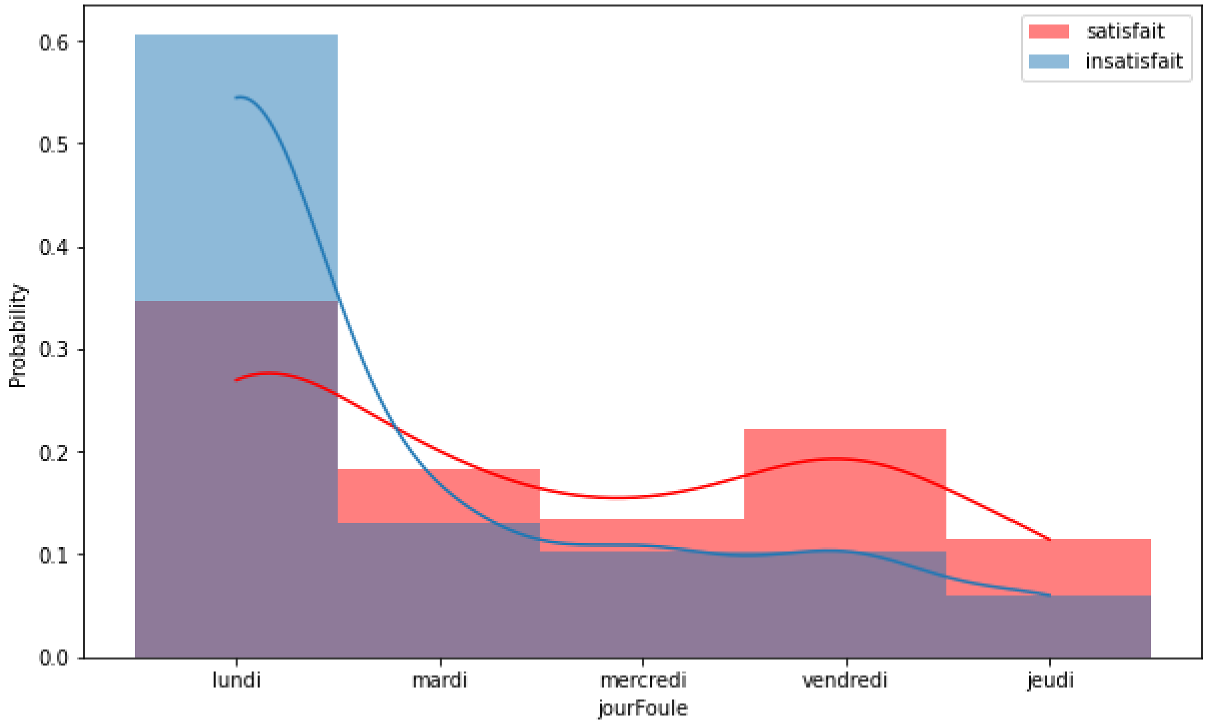


satisfait
insatisfait
Probability
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
non
oui
Manque des employÃ©



satisfait
insatisfait
Probability
2.5
2.0
1.5
1.0
0.5
0.0
oui
non
mauvaise maniere Traitement



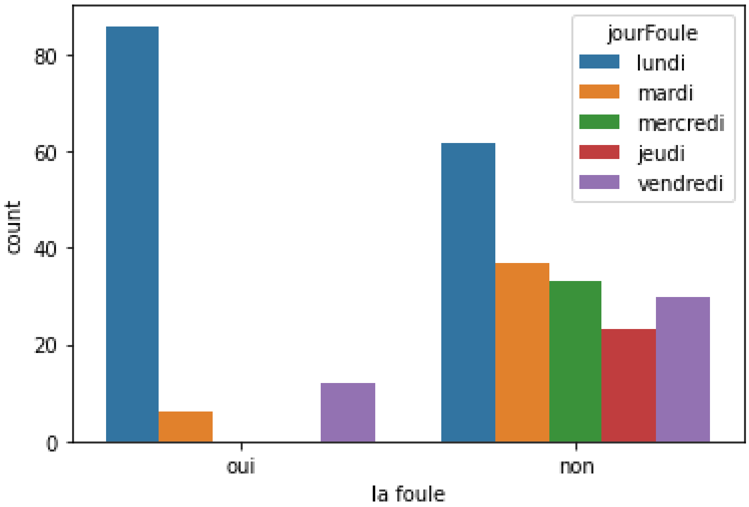


In [264…

```
sns.countplot(x='la foule',hue='jourFoule',data=test,linewidth=1)
```

Out[264…

```
<AxesSubplot:xlabel='la foule', ylabel='count'>
```

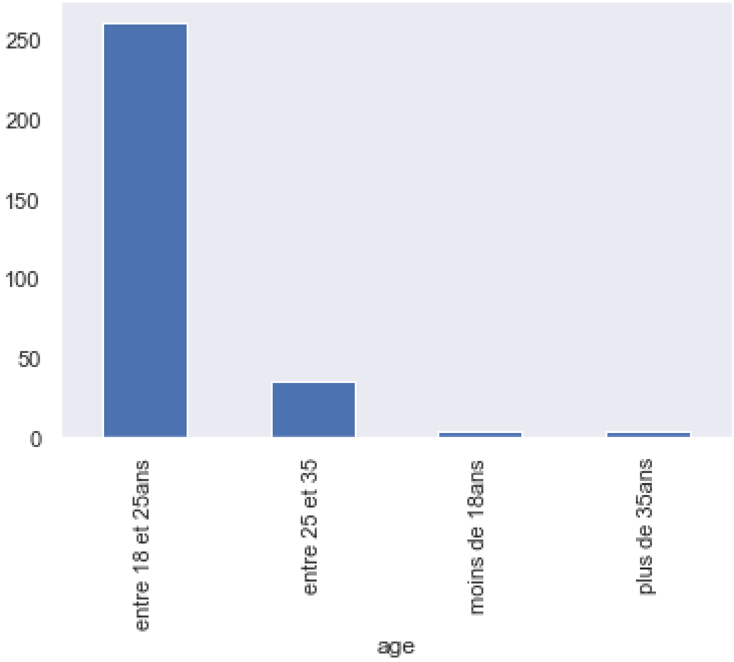


In [399…

```
test.groupby(['age']).count()['communication'].plot.bar()
```

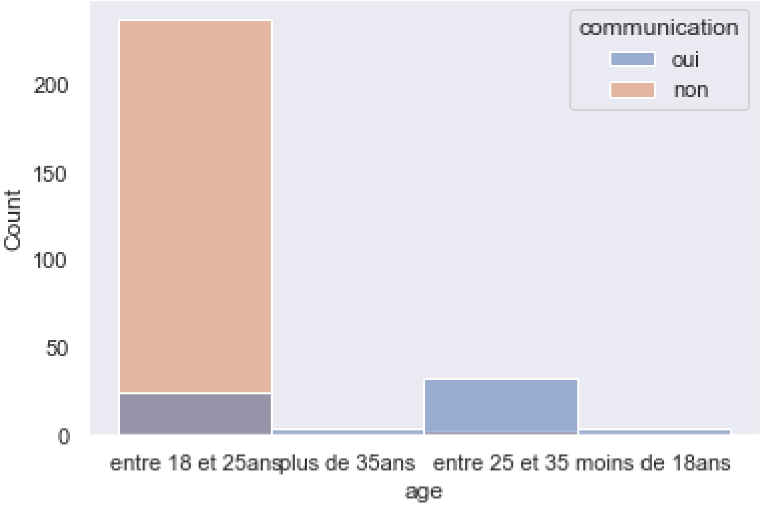
Out[399…

```
<AxesSubplot:xlabel='age'>
```



In [361…

```
sns.histplot(x='age',hue='communication',data=test,linewidth=1)
sns.set_style("dark")
```



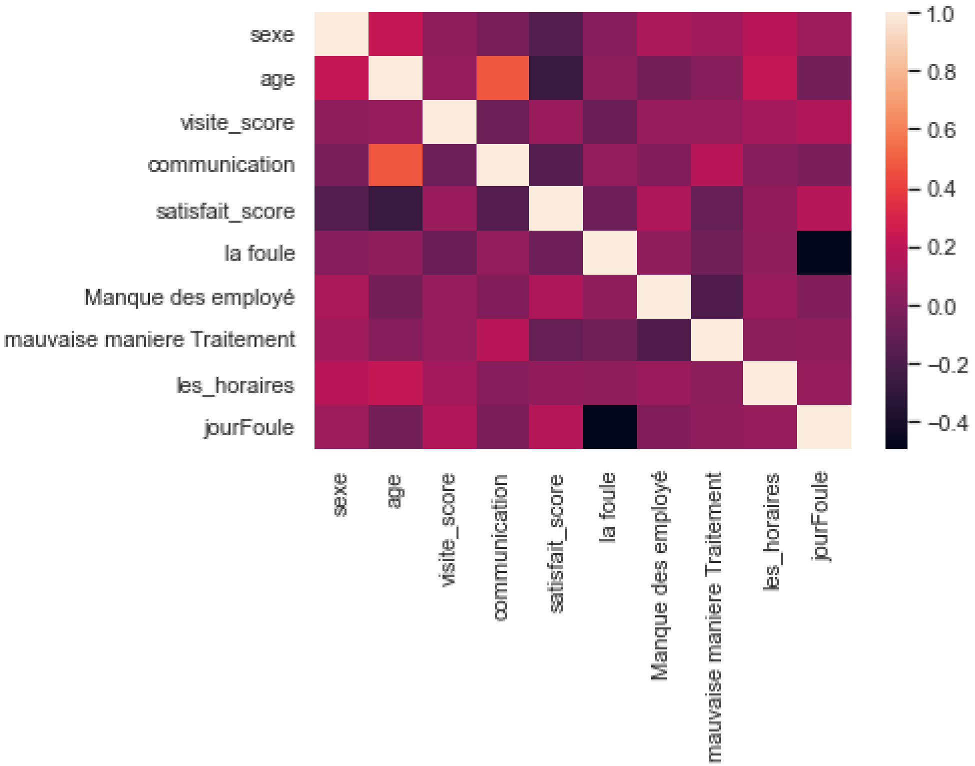
correlation entre les variables

In [483…

```
sns.heatmap(test.corr())
```

Out[483…

```
<AxesSubplot:>
```

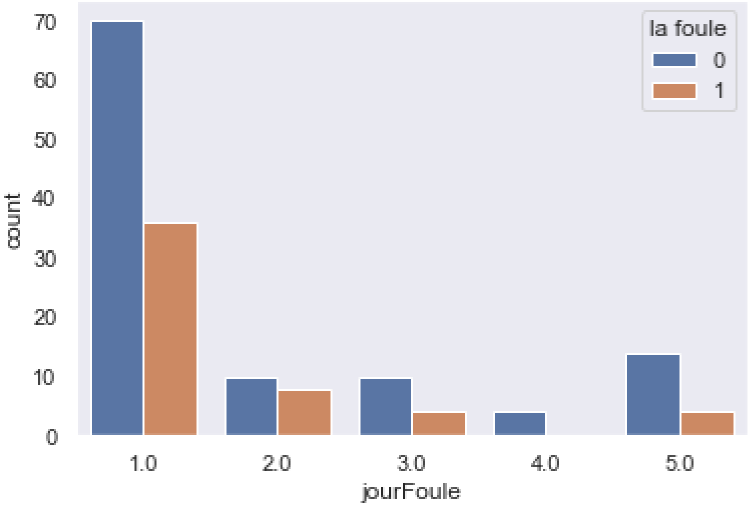


In [487…

```
sns.countplot(x='jourFoule',hue='la foule',data=test,linewidth=1)
```

Out[487…

```
<AxesSubplot:xlabel='jourFoule', ylabel='count'>
```



In [486…

```
data=test.groupby(['age'])['la foule','Manque des employÃ©','mauvaise maniere Trait
```

```
C:\Users\pc\AppData\Local\Temp/ipykernel_22852/2733498621.py:1: FutureWarning: Index
ing with multiple keys (implicitly converted to a tuple of keys) will be deprecated,
use a list instead.
  data=test.groupby(['age'])['la foule','Manque des employÃ©','mauvaise maniere Trai
tement'].sum().plot.bar()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_22852/2733498621.py in <module>
----> 1 data=test.groupby(['age'])['la foule','Manque des employÃ©','mauvaise manier
e Traitement'].sum().plot.bar()

~\anaconda3\lib\site-packages\pandas\core\groupby\generic.py in __getitem__(self, ke
y)
   1536                 stacklevel=2,
   1537             )
-> 1538         return super().__getitem__(key)
   1539 
   1540     def _gotitem(self, key, ndim: int, subset=None):

~\anaconda3\lib\site-packages\pandas\core\base.py in __getitem__(self, key)
    220             if len(self.obj.columns.intersection(key)) != len(key):
    221                 bad_keys = list(set(key).difference(self.obj.columns))
--> 222                 raise KeyError(f"Columns not found: {str(bad_keys)[1:-1]}")
    223             return self._gotitem(list(key), ndim=2)
    224 

KeyError: "Columns not found: 'Manque des employÃ©'"
```

In [269…

```
test.corr()['satisfit_score'].sort_values()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, meth
od, tolerance)
   3360             try:
-> 3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in pandas._libs.index.IndexEngi
ne.get_loc()

~\anaconda3\lib\site-packages\pandas\_libs\index.pyx in pandas._libs.index.IndexEngi
ne.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.
get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.
get_item()

KeyError: 'satisfit_score'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_22852/2339596374.py in <module>
----> 1 test.corr()['satisfit_score'].sort_values()

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
   3456             if self.columns.nlevels > 1:
   3457                 return self._getitem_multilevel(key)
-> 3458             indexer = self.columns.get_loc(key)
   3459             if is_integer(indexer):
   3460                 indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, meth
od, tolerance)
   3361                 return self._engine.get_loc(casted_key)
   3362             except KeyError as err:
-> 3363                 raise KeyError(key) from err
   3364 
   3365         if is_scalar(key) and isna(key) and not self.hasnans:

KeyError: 'satisfit_score'
```

In [404…

```
for col in test :
      if col=='satisfait_score':
          print("teeem")
      else:
                print("no")
```

```
no
no
no
no
no
no
teeem
no
no
no
no
no
```

test et des hépothes

In [ ]:

```
satisfait=test[test['satisfait_score']>2]
insatisfait=test[test['satisfait_score']<2]
```

In [1]:

```
test.describe()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_4880/286099727.py in <module>
----> 1 test.describe()

NameError: name 'test' is not defined
```

In [437…

```
insatisfait_simple=insatisfait.sample(100)
satisfait_simple=satisfait.sample(100)
```

In [497…

```
insatisfait_simple.to_csv("insatisfait_simple.csv",index=False)
```

In [ ]:

```

```

In [438…

```
from scipy.stats import ttest_ind
balanced_nid=insatisfait_simple.sample(satisfait_simple.shape[0])
```

In [493…

```
def t_test(col):
    alpha=0.05
    stat , p=ttest_ind(balanced_nid[col].dropna(),satisfait_simple[col].dropna())
    if p<alpha :
            return 'H0 reject '
        else :
            return 0
```

In [495…

```
teste=test.drop(['province','bureau'],axis=1)

for col in teste:
    print(f'{col :-<50} {t_test(col)}')
```

```
sexe---------------------------------------------- 0
age----------------------------------------------- H0 reject
visite_score-------------------------------------- H0 reject
satisfait_score----------------------------------- H0 reject
la foule------------------------------------------ H0 reject
Manque des employé-------------------------------- 0
mauvaise maniere Traitement----------------------- 0
les_horaires-------------------------------------- 0
jourFoule----------------------------------------- H0 reject
```

In [442…

```
teste=test.drop(['province','bureau'],axis=1)

for col in teste:
    print(f'{col :-<50} {t_test(col)}')
```

```
sexe---------------------------------------------- H0 reject
age----------------------------------------------- H0 reject
visite_score-------------------------------------- H0 reject
communication------------------------------------- H0 reject
satisfait_score----------------------------------- H0 reject
la foule------------------------------------------ H0 reject
Manque des employé-------------------------------- H0 reject
mauvaise maniere Traitement----------------------- 0
les_horaires-------------------------------------- H0 reject
jourFoule----------------------------------------- H0 reject
```

In [9]:

```
from sklearn.feature_selection import SelectKBest, chi2
```

In [20]:

```
X=X.dropna(axis=0)
X.head()
```

Out[20]:

	sexe	age	visite_score	la foule	Manque des employé	mauvaise maniere Traitement	les_horaires	jourFoule
0	0	1	0	1	0	0	1.0	1.0
4	0	1	0	1	0	0	1.0	2.0
6	0	1	0	1	0	1	1.0	1.0
7	1	1	1	1	1	1	1.0	5.0
8	1	1	1	0	1	1	3.0	1.0

In [2]:

```
test=test.dropna(axis=0)
Y=test['satisfait_score']
X=test.drop(['province','bureau'],axis=1)

rrr=chi2(X,Y)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_4880/2806542395.py in <module>
----> 1 test=test.dropna(axis=0)
      2 Y=test['satisfait_score']
      3 X=test.drop(['province','bureau'],axis=1)
      4 
      5 rrr=chi2(X,Y)

NameError: name 'test' is not defined
```

In [49]:

```
print(rrr)
```

```
(array([ 5.07479352, 36.56840444,  4.27882915, 75.83982684,  3.46366642,
        0.37930912,  2.39259744,  0.21474613,  8.06909466]), array([7.90719755e-02,
1.14623026e-08, 1.17723741e-01, 3.40087639e-17,
       1.76959708e-01, 8.27244847e-01, 3.02311082e-01, 8.98190528e-01,
       1.76936877e-02]))
```

In [46]:

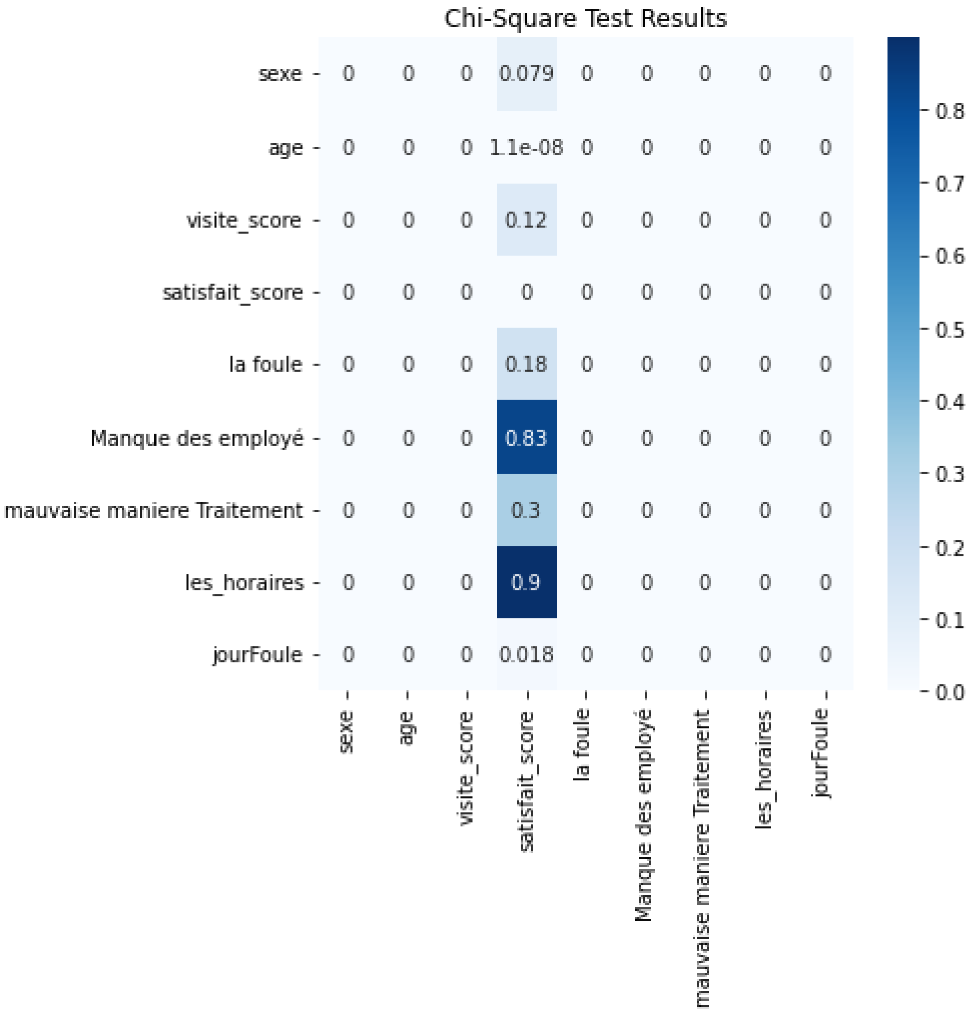
```
resultant = pd.DataFrame(data=[(0 for i in range(len(X.columns))) for i in range(len
                         columns=list(X.columns))
# Finding p_value for all columns and putting them in the resultant matrix
resultant.set_index(pd.Index(list(X.columns)), inplace = True)

for i in list(X.columns):
        j="satisfait_score"
        if(j=="satisfait_score"):
            print(j)
            if i != j:
                chi2_val, p_val = chi2(np.array(X[i]).reshape(-1, 1), np.array(X[j])
                resultant.loc[i,j] = p_val
print(resultant)
```

```
satisfait_score
satisfait_score
satisfait_score
satisfait_score
satisfait_score
satisfait_score
satisfait_score
satisfait_score
satisfait_score
                             sexe  age  visite_score  satisfait_score  \
sexe                            0    0             0     7.907198e-02
age                             0    0             0     1.146230e-08
visite_score                    0    0             0     1.177237e-01
satisfait_score                 0    0             0     0.000000e+00
la foule                        0    0             0     1.769597e-01
Manque des employé              0    0             0     8.272448e-01
mauvaise maniere Traitement     0    0             0     3.023111e-01
les_horaires                    0    0             0     8.981905e-01
jourFoule                       0    0             0     1.769369e-02

                             la foule  Manque des employé  \
sexe                                0                   0
age                                 0                   0
visite_score                        0                   0
satisfait_score                     0                   0
la foule                            0                   0
Manque des employé                  0                   0
mauvaise maniere Traitement         0                   0
les_horaires                        0                   0
```

```
jourFoule                            0                   0

                             mauvaise maniere Traitement  les_horaires   \
sexe                                                   0             0
age                                                    0             0
visite_score                                           0             0
satisfait_score                                        0             0
la foule                                               0             0
Manque des employé                                     0             0
mauvaise maniere Traitement                            0             0
les_horaires                                           0             0
jourFoule                                              0             0

                             jourFoule
sexe                                 0
age                                  0
visite_score                         0
satisfait_score                      0
la foule                             0
Manque des employé                   0
mauvaise maniere Traitement          0
les_horaires                         0
jourFoule                            0
```

In [47]:

```python
# Plotting a heatmap
fig = plt.figure(figsize=(6,6))
sns.heatmap(resultant, annot=True, cmap='Blues')
plt.title('Chi-Square Test Results')
plt.show()
```



In [53]:

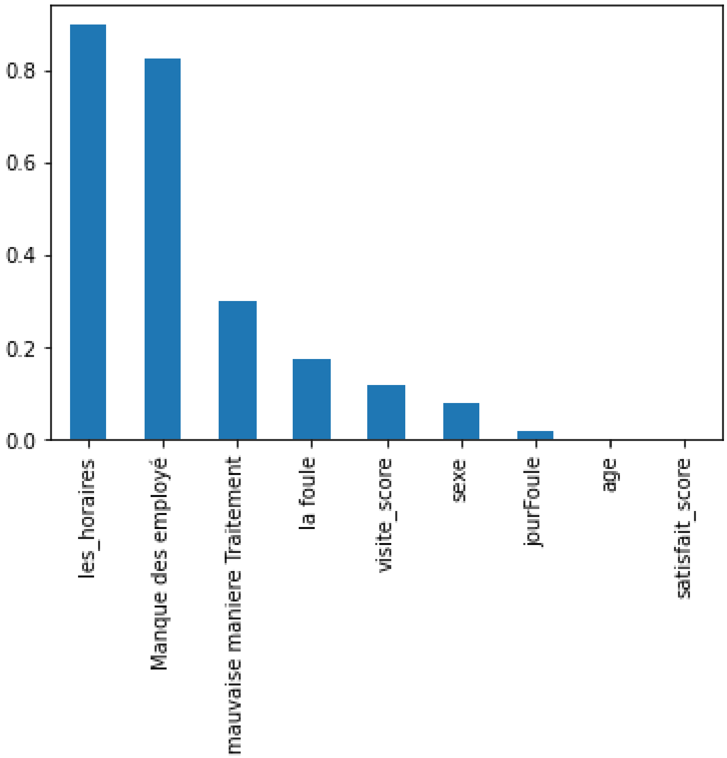
```
p_values = pd.Series(rrr[1],index = X.columns)
chi2score = pd.Series(rrr[0],index = X.columns)
chi2score.sort_values(ascending = False , inplace = True)

p_values.sort_values(ascending = False , inplace = True)
```

In [59]:

```
p_values.plot.bar()
```

Out[59]:

```
<AxesSubplot:>
```



In [52]:

```
pd.crosstab(rrr[1], df.body_style)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9856/3848959528.py in <module>
----> 1 pd.crosstab(rrr[1], X)

~\anaconda3\lib\site-packages\pandas\core\reshape\pivot.py in crosstab(index, columns, values, rownames, colnames, aggfunc, margins, margins_name, dropna, normalize)
    652         **dict(zip(unique_colnames, columns)),
    653     }
--> 654     df = DataFrame(data, index=common_idx)
    655 
    656     if values is None:

~\anaconda3\lib\site-packages\pandas\core\frame.py in __init__(self, data, index, columns, dtype, copy)
    612         elif isinstance(data, dict):
    613             # GH#38939 de facto copy defaults to False only in non-dict cases
--> 614             mgr = dict_to_mgr(data, index, columns, dtype=dtype, copy=copy, typ=manager)
    615         elif isinstance(data, ma.MaskedArray):
    616             import numpy.ma.mrecords as mrecords

~\anaconda3\lib\site-packages\pandas\core\internals\construction.py in dict_to_mgr(data, index, columns, dtype, typ, copy)
```

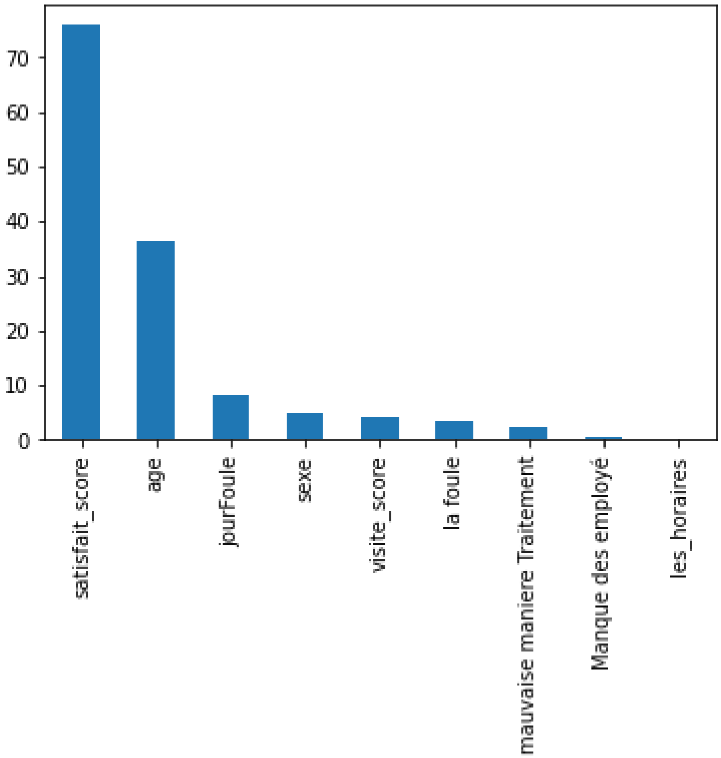
```
    462         # TODO: can we get rid of the dt64tz special case above?
    463 
--> 464     return arrays_to_mgr(
    465         arrays, data_names, index, columns, dtype=dtype, typ=typ, consolidate=copy
    466     )

~\anaconda3\lib\site-packages\pandas\core\internals\construction.py in arrays_to_mgr(arrays, arr_names, index, columns, dtype, verify_integrity, typ, consolidate)
    122 
    123         # don't force copy because getting jammed in an ndarray anyway
--> 124         arrays = _homogenize(arrays, index, dtype)
    125 
    126     else:

~\anaconda3\lib\site-packages\pandas\core\internals\construction.py in _homogenize(data, index, dtype)
    587                 val = lib.fast_multiget(val, oindex._values, default=np.nan)
    588 
--> 589             val = sanitize_array(
    590                 val, index, dtype=dtype, copy=False, raise_cast_failure=False
    591             )

~\anaconda3\lib\site-packages\pandas\core\construction.py in sanitize_array(data, index, dtype, copy, raise_cast_failure, allow_2d)
    574                 subarr = maybe_infer_to_datetimelike(subarr)
    575 
--> 576     subarr = _sanitize_ndim(subarr, data, dtype, index, allow_2d=allow_2d)
    577 
    578     if isinstance(subarr, np.ndarray):

~\anaconda3\lib\site-packages\pandas\core\construction.py in _sanitize_ndim(result, data, dtype, index, allow_2d)
    625             if allow_2d:
    626                 return result
--> 627             raise ValueError("Data must be 1-dimensional")
    628         if is_object_dtype(dtype) and isinstance(dtype, ExtensionDtype):
    629             # i.e. PandasDtype("O")

ValueError: Data must be 1-dimensional
```

In [60]:

```
chi2score.plot.bar()
```

Out[60]:

```
<AxesSubplot:>
```

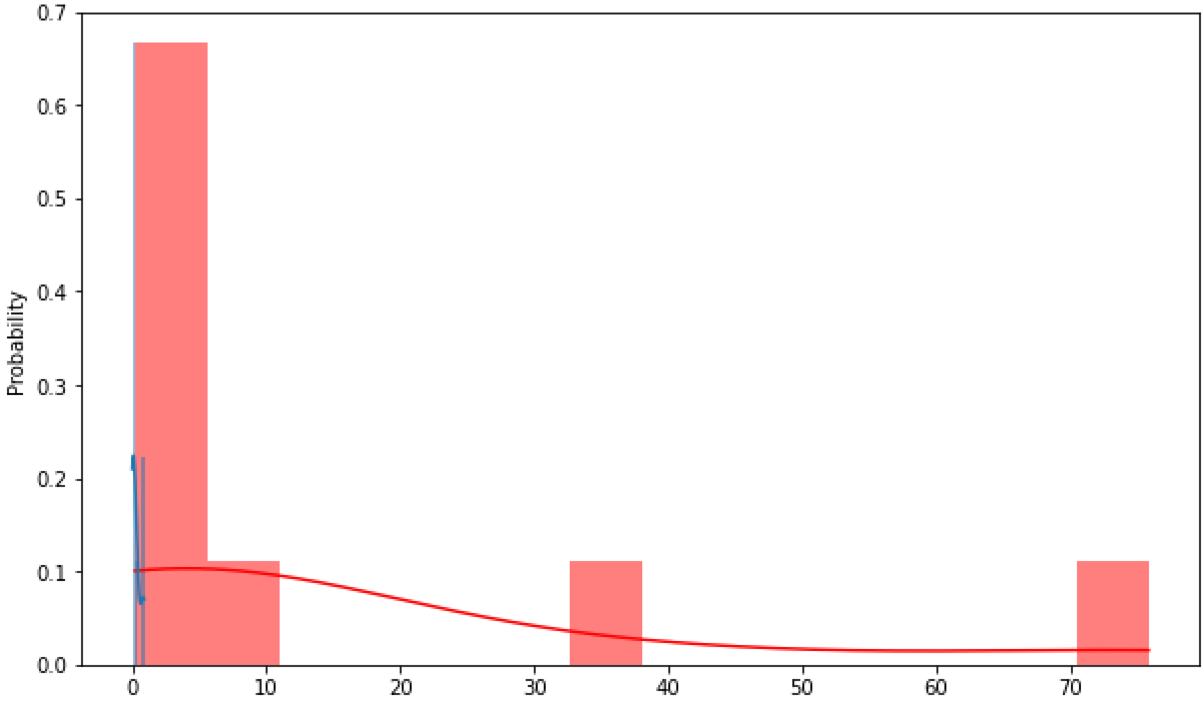


In [58]:

```
fig = plt.figure(figsize=(10,6))
colors = sns.color_palette('bright')[0:5]

sns.histplot(chi2score, kde=True, stat="probability",label='satisfait',linewidth
sns.histplot(p_values, kde=True, stat="probability",label='insatisfait', linewid
```

Out[58]:

```
<AxesSubplot:ylabel='Probability'>
```



In [ ]: