

DEA_SEMINAIRE DE VISUALISATION

Présenté par Assistant MBANGU_NDUNGA_Elie Apprenant en Data Science et Intelligence Artificielle 

Dirigé par le Professeur Félicien Jordan MASAKUNA, Université de Kinshasa, Faculté des Sciences, Mention Mathématique, Statistique et Informatique

Entrée [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
```

1. Chargement de notre Dataset et Visualisation de données

Entrée [2]:

```
dt = pd.read_csv('elie_dataset.csv')
```

Entrée [3]:

```
dt.head()
```

Out[3]:

	Phone	Account.Length	Day.Mins	Day.Calls	Day.Charge	Eve.Mins	Eve.Calls	Eve.Charge
0	382-4657	128	265.1	110	45.07	197.4	99	16.78
1	371-7191	107	161.6	123	27.47	195.5	103	16.62
2	358-1921	137	243.4	114	41.38	121.2	110	10.30
3	375-9999	84	299.4	71	50.90	61.9	88	5.26
4	330-6626	75	166.7	113	28.34	148.3	122	12.61

Entrée [4]: `dt.describe()`

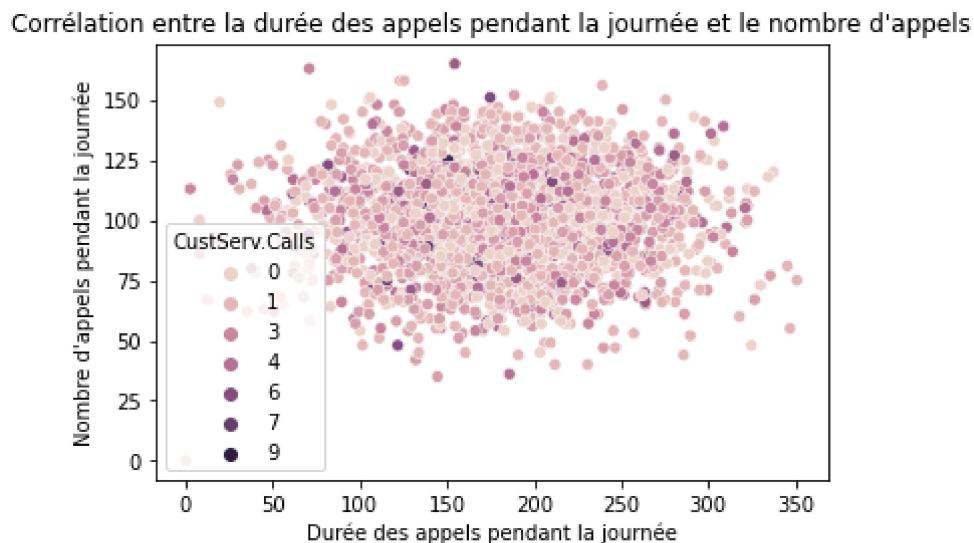
	Account.Length	Day.Mins	Day.Calls	Day.Charge	Eve.Mins	Eve.Calls	Eve.Charge
count	2667.000000	2667.000000	2667.000000	2667.000000	2667.000000	2667.000000	2667.000000
mean	101.246344	179.752943	100.198350	30.558538	201.467942	100.176978	10.000000
std	39.515839	54.607640	19.859277	9.283274	50.233057	20.059761	1.000000
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	74.000000	143.750000	87.000000	24.440000	167.250000	87.000000	7.000000
50%	101.000000	179.300000	101.000000	30.480000	202.300000	100.000000	10.000000
75%	127.000000	216.850000	113.000000	36.865000	235.100000	114.000000	13.000000
max	243.000000	350.800000	165.000000	59.640000	363.700000	170.000000	2667.000000

2. Visualisation des données avec Nuage

Un nuage de points est une excellente visualisation pour explorer visuellement la relation entre deux variables. Il offre des informations sur la corrélation, la dispersion et les tendances dans les données.

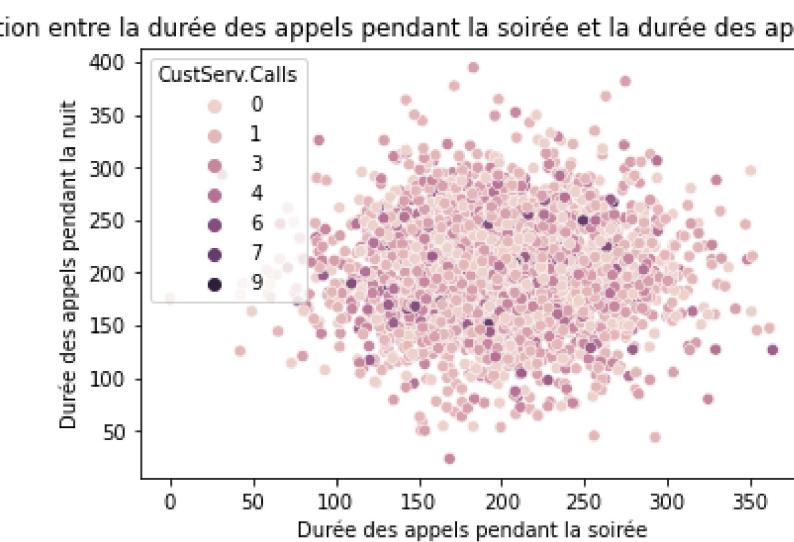
a) Nuage de points montrant la corrélation entre la durée des appels téléphoniques pendant la journée (Day.Mins) et le nombre d'appels pendant la journée (Day.Calls)

Entrée [5]: `#plt.plot(x= dt['Day.Mins'], y= dt['Day.Calls'], hue='c')`
`sns.scatterplot(x=dt['Day.Mins'], y= dt['Day.Calls'], hue = dt['CustServ.Ca]`
`plt.xlabel('Durée des appels pendant la journée')`
`plt.ylabel('Nombre d\'appels pendant la journée')`
`plt.title('Corrélation entre la durée des appels pendant la journée et le no`
`plt.show()`



b) Nuage de points montrant la corrélation des variables entre la durée des appels téléphoniques pendant la soirée (Eve.Mins) et la durée des appels téléphoniques pendant la nuit (Night.Mins):

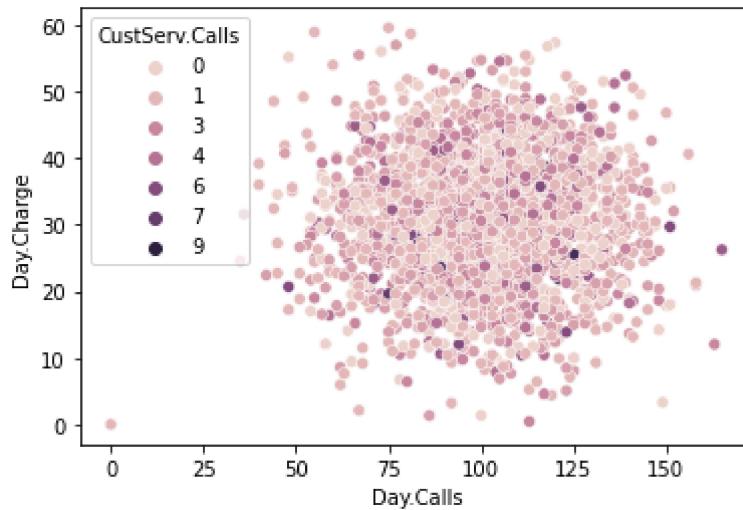
```
Entrée [6]: #plt.scatter(dt['Eve.Mins'], dt['Night.Mins'], c = 'r')
sns.scatterplot(x=dt['Eve.Mins'], y= dt['Night.Mins'], hue = dt['CustServ.Calls'])
plt.xlabel('Durée des appels pendant la soirée')
plt.ylabel('Durée des appels pendant la nuit')
plt.title('Corrélation entre la durée des appels pendant la soirée et la durée des appels pendant la nuit')
plt.show()
```



c) Nuage de points pour connaître le nombre d'appels journalier (Day.Calls) et les frais des appels journaliers (Day.Charge)

Entrée [7]: `sns.scatterplot(x=dt['Day.Calls'], y= dt['Day.Charge'], hue = dt['CustServ.Calls'])`

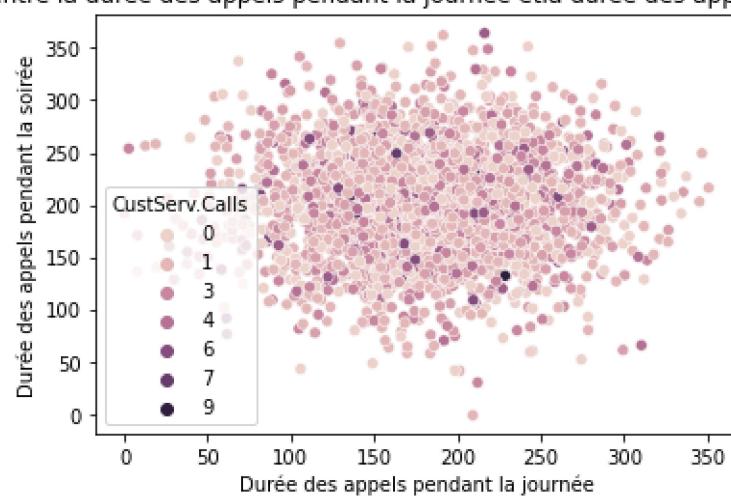
Out[7]: <AxesSubplot:xlabel='Day.Calls', ylabel='Day.Charge'>



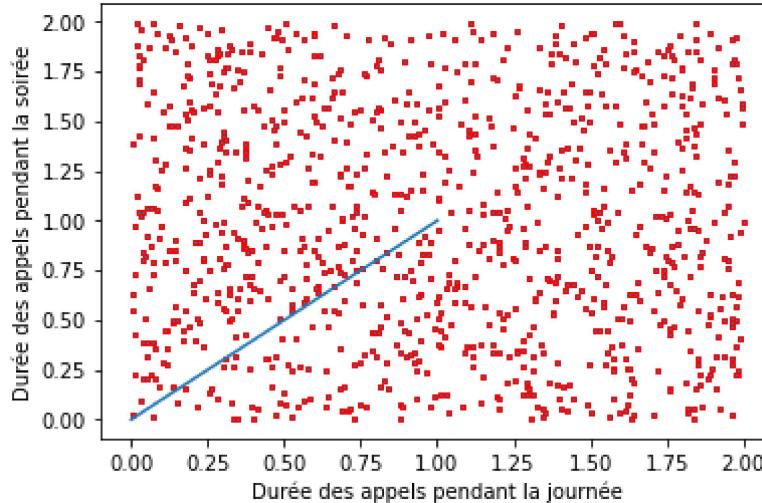
d) Nuage de points montrant la corrélation entre la durée des appels pendant la journée (Day.Mins) et la durée des appels pendant la soirée (Eve.Mins)

Entrée [8]: `#plt.scatter(x= dt['Day.Mins'], y=dt['Eve.Mins'])`
`sns.scatterplot(x=dt['Day.Mins'], y= dt['Eve.Mins'], hue = dt['CustServ.Calls'])`
`plt.xlabel('Durée des appels pendant la journée')`
`plt.ylabel('Durée des appels pendant la soirée')`
`plt.title('Corrélation entre la durée des appels pendant la journée et la durée des appels pendant la soirée')`
`plt.show()`

Corrélation entre la durée des appels pendant la journée et la durée des appels pendant la soirée



```
Entrée [9]: points=np.random.uniform(0, 2, (1000,2))
plt.xlabel('Durée des appels pendant la journée')
plt.ylabel('Durée des appels pendant la soirée')
plt.scatter(points[:,0], points[:,1],
            marker=',', edgecolor='r', s=6)
plt.plot([i for i in np.arange(0, 2)],
          [i for i in np.arange(0, 2)])
plt.show()
```

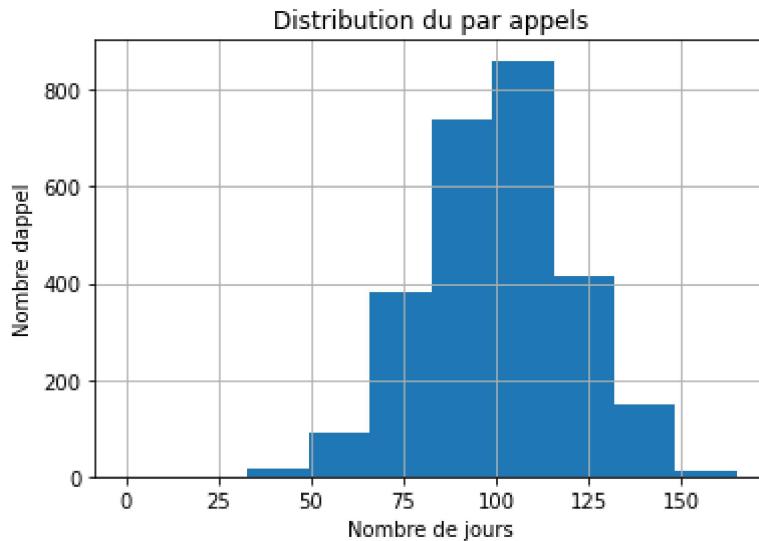


3. Visualisation avec des Histogrammes de la variable minute par jours

Un histogramme est une représentation visuelle puissante de la distribution des données. En l'interprétant correctement, vous pouvez obtenir des informations sur la centralité, la dispersion et la forme de la distribution.

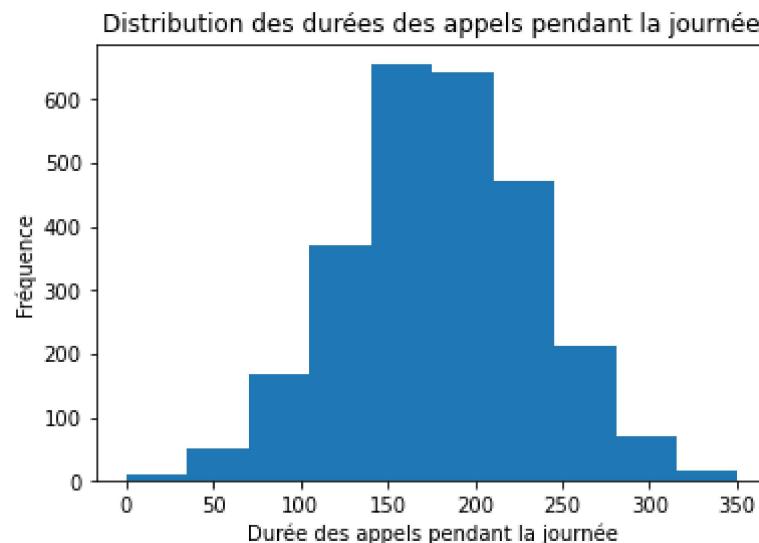
a) Histogrammes de la variable minute par jours

```
Entrée [10]: dt['Day.Calls'].hist()
plt.title('Distribution du par appels')
plt.xlabel('Nombre de jours')
plt.ylabel('Nombre d'appel')
plt.show()
```

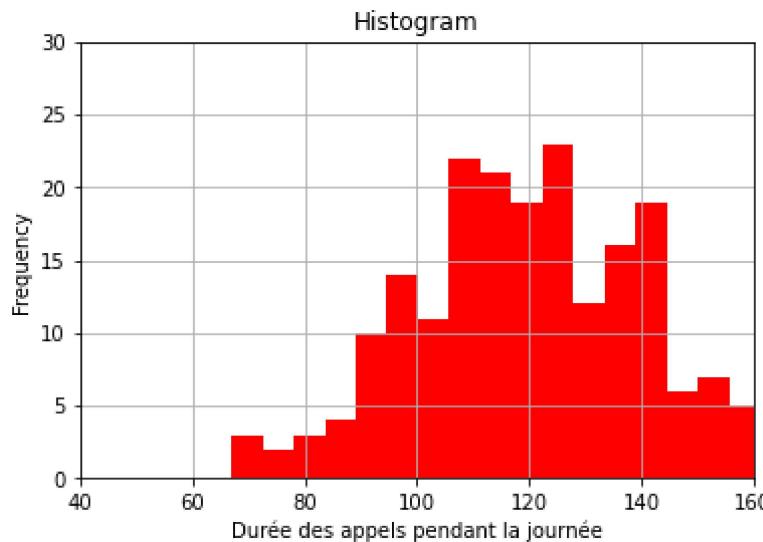


b) Histogramme de la distribution des durées des appels téléphoniques pendant la journée (Day.Mins):

```
Entrée [11]: plt.hist(dt['Day.Mins'], bins=10)
plt.xlabel('Durée des appels pendant la journée')
plt.ylabel('Fréquence')
plt.title('Distribution des durées des appels pendant la journée')
plt.show()
```

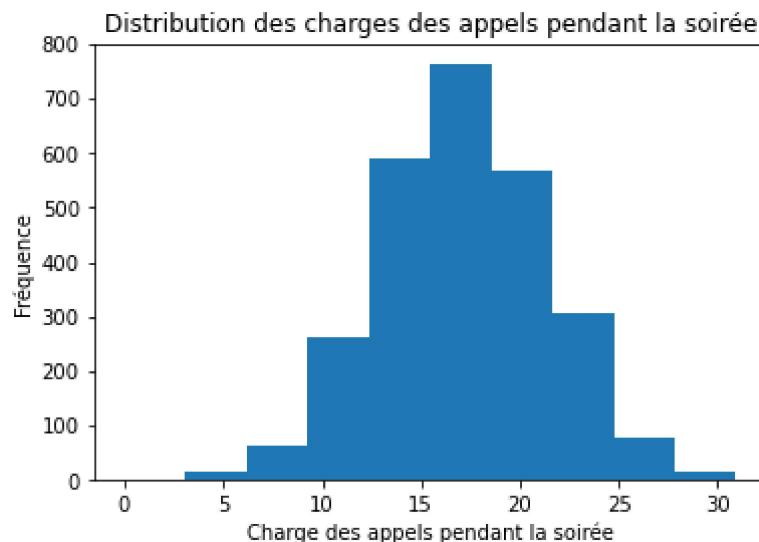


```
Entrée [12]: x = 120 + 22 * np.random.randn(200)
n, bins, patches = plt.hist(x, bins=20, facecolor='r')
plt.xlabel('Durée des appels pendant la journée')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.axis([40, 160, 0, 30])
plt.grid(True)
plt.show()
```



c) Histogramme de la distribution des charges des appels téléphoniques pendant la soirée (Eve.Charge):

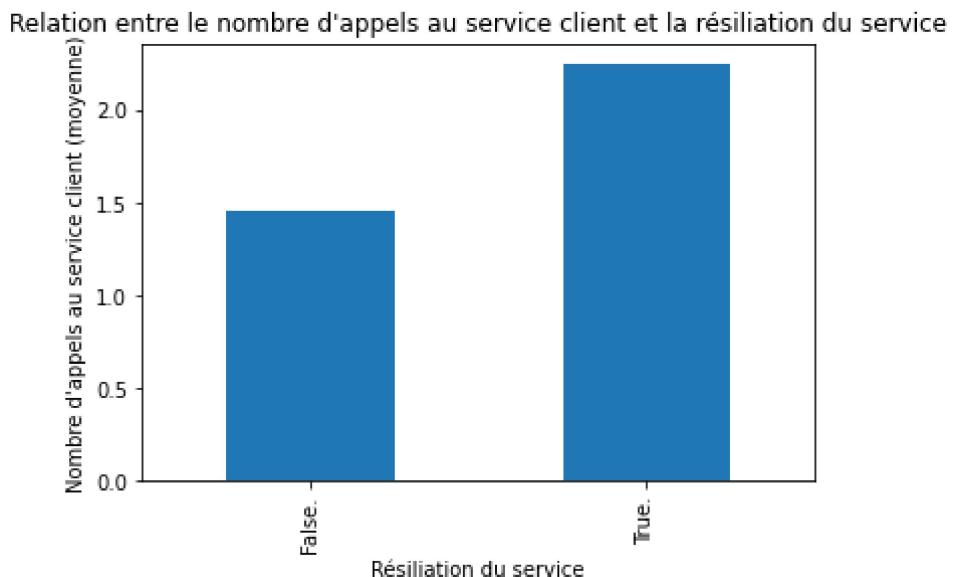
```
Entrée [13]: plt.hist(dt['Eve.Charge'], bins=10)
plt.xlabel('Charge des appels pendant la soirée')
plt.ylabel('Fréquence')
plt.title('Distribution des charges des appels pendant la soirée')
plt.show()
```



d) Diagramme en barres montrant la relation entre le nombre d'appels au service client (CustServ.Calls) et le fait de résilier le service (Churn)

Entrée [14]:

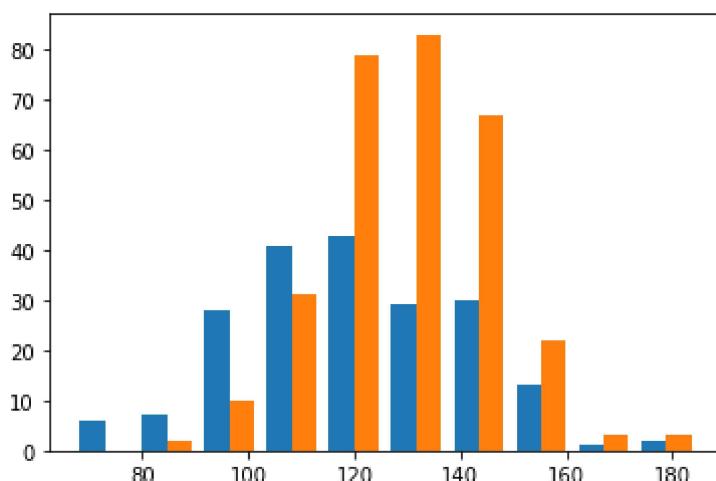
```
churn_counts = dt.groupby('Churn.')[['CustServ.Calls']].mean()
churn_counts.plot(kind='bar')
plt.xlabel('Résiliation du service')
plt.ylabel('Nombre d\'appels au service client (moyenne)')
plt.title('Relation entre le nombre d\'appels au service client et la résili:')
plt.show()
```



e) Histogramme avec plusieurs ensembles de données

Entrée [15]:

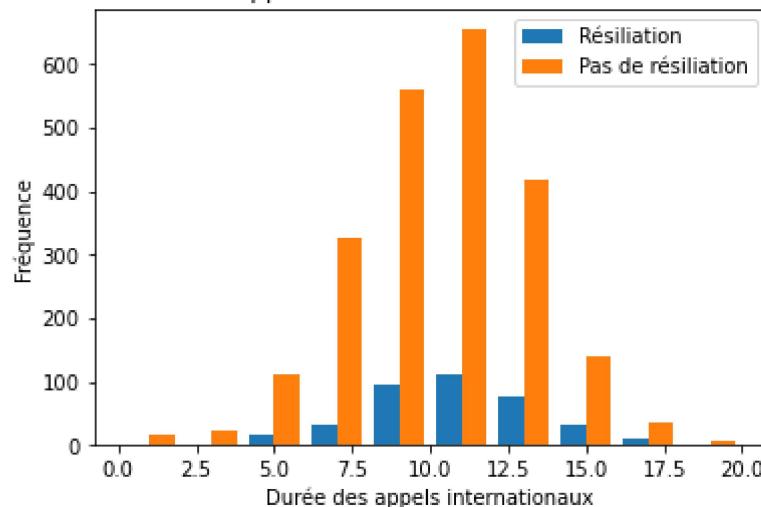
```
y = 130 + 15*np.random.randn(300)
n, bins, patches = plt.hist([x, y])
plt.show()
```



```
Entrée [16]: churn_intlmins = dt[dt['Churn.'] == 'True.']['Intl.Mins']
nochurn_intlmins = dt[dt['Churn.'] == 'False.']['Intl.Mins']

plt.hist([churn_intlmins, nochurn_intlmins], bins=10, label=['Résiliation',
plt.xlabel('Durée des appels internationaux')
plt.ylabel('Fréquence')
plt.title('Distribution de la durée des appels internationaux en fonction de la résiliation')
plt.legend()
plt.show()
```

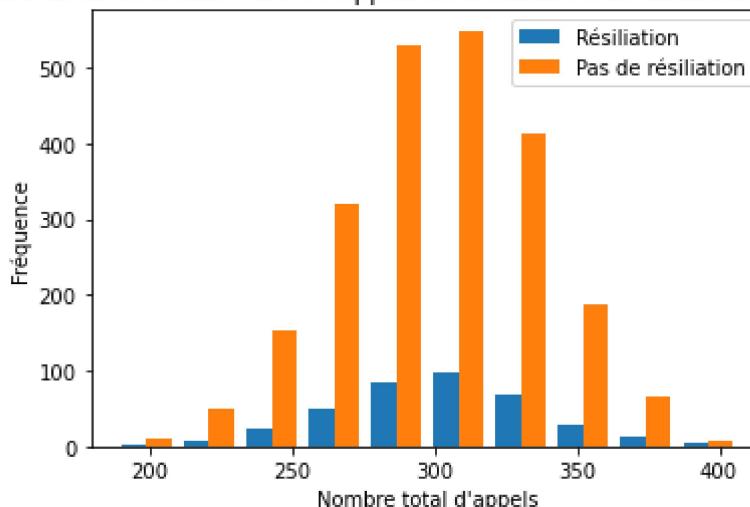
Distribution de la durée des appels internationaux en fonction de la résiliation du service



```
Entrée [17]: churn_totalcalls = dt[dt['Churn.'] == 'True.']['Day.Calls'] + dt[dt['Churn.']
nochurn_totalcalls = dt[dt['Churn.'] == 'False.']['Day.Calls'] + dt[dt['Churn.']

plt.hist([churn_totalcalls, nochurn_totalcalls], bins=10, label=['Résiliation',
plt.xlabel('Nombre total d\'appels')
plt.ylabel('Fréquence')
plt.title('Distribution du nombre total d\'appels en fonction de la résiliation')
plt.legend()
plt.show()
```

Distribution du nombre total d'appels en fonction de la résiliation du service

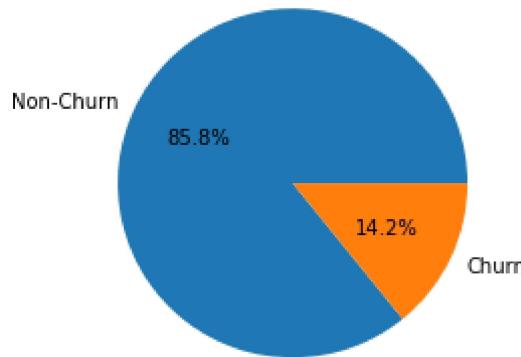


4. Diagrammes

a) Diagramme en secteurs du taux de désabonnement (Churn)

```
Entrée [18]: plt.pie(dt['Churn.'].value_counts(), labels=['Non-Churn', 'Churn'], autopct='%.2f')
plt.title('Répartition du taux de désabonnement')
plt.show()
```

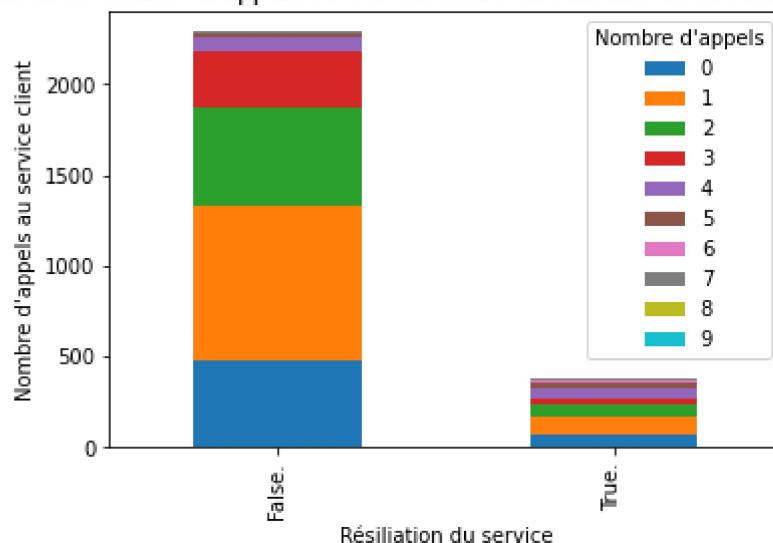
Répartition du taux de désabonnement



b) Diagramme en barres montrant la distribution du nombre d'appels au service client (CustServ.Calls) en fonction de la résiliation du service (Churn)

```
Entrée [19]: churn_counts = dt.groupby('Churn.')['CustServ.Calls'].value_counts().unstack()
churn_counts.plot(kind='bar', stacked=True)
plt.xlabel('Résiliation du service')
plt.ylabel('Nombre d\'appels au service client')
plt.title('Distribution du nombre d\'appels au service client en fonction de la résiliation du service')
plt.legend(title='Nombre d\'appels')
plt.show()
```

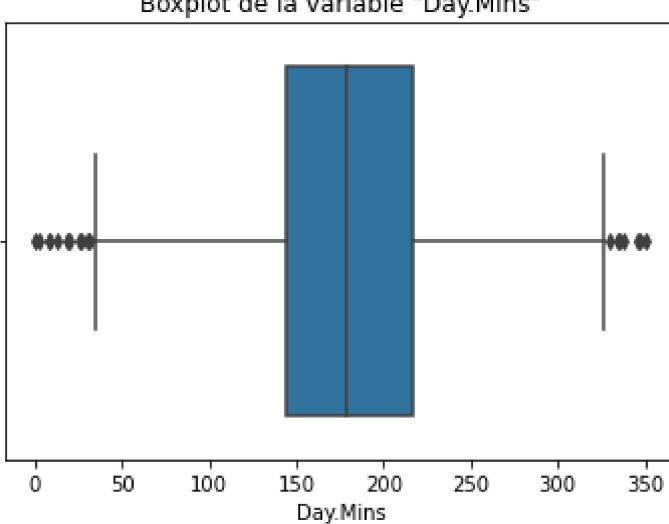
Distribution du nombre d'appels au service client en fonction de la résiliation du service



c) Génération des boxplots pour chaque variable numérique sur des graphiques séparés

Entrée [20]:

```
for col in ['Day.Mins', 'Day.Calls', 'Day.Charge', 'Eve.Mins', 'Eve.Calls']:
    sns.boxplot(x=dt[col])
    plt.title(f'Boxplot de la variable "{col}"')
    plt.show()
```



Boxplot de la variable "Day.Calls"

5. chargement des données et Visualisation des données réduites de Iri

Entrée [21]:

```
digits = datasets.load_digits()
iris = datasets.load_iris()
```

Entrée [22]:

```
X_digits, y_digits = digits.data, digits.target
X_iris, y_iris = iris.data, iris.target
```

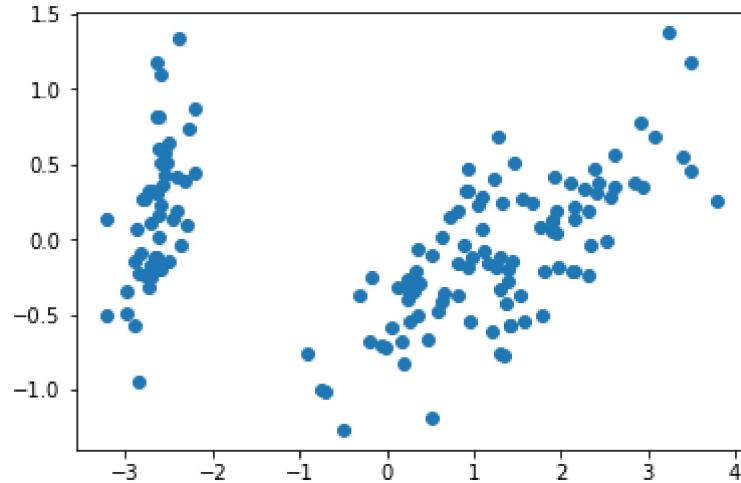
Entrée [23]:

```
acp_digits = PCA(n_components=2)
X_digits_acp = acp_digits.fit_transform(X_digits)
```

Entrée [24]:

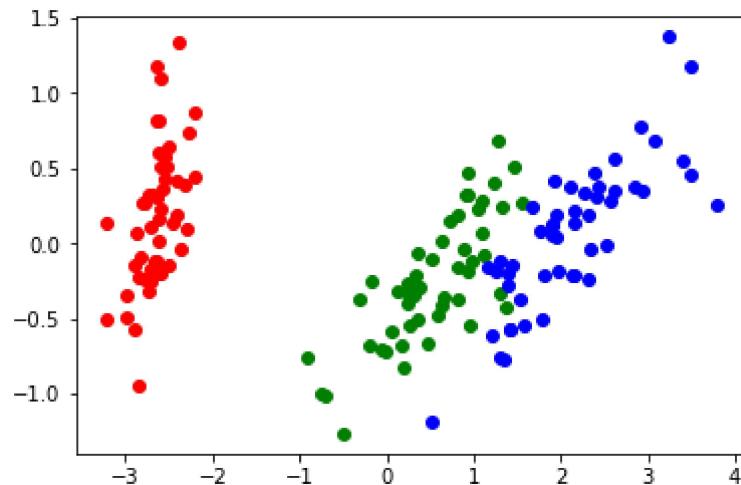
```
acp_iris = PCA(n_components=2)
X_iris_acp = acp_iris.fit_transform(X_iris)
```

```
Entrée [25]: plt.scatter(X_iris_acp[:, 0], X_iris_acp[:, 1])
plt.show()
```

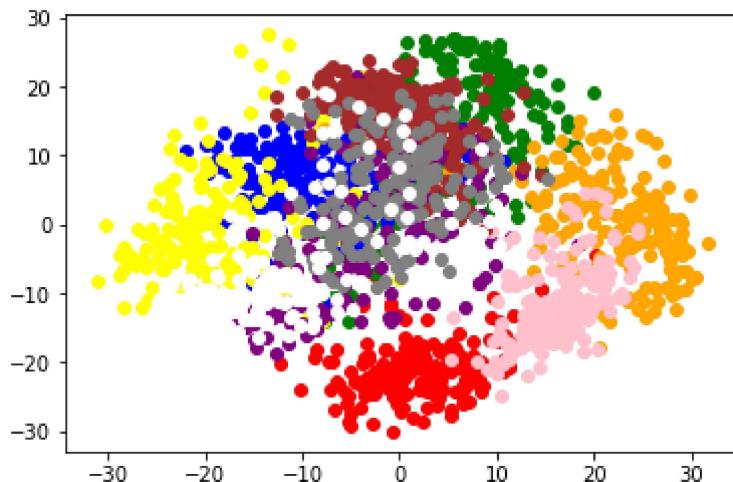


Indication des clusters connus de Iris

```
Entrée [26]: colors = ['red', 'green', 'blue', 'yellow', 'orange', 'purple', 'pink', 'brown']
for i in range(3):
    plt.scatter(X_iris_acp[y_iris==i, 0], X_iris_acp[y_iris==i, 1], color=colors[i])
plt.show()
```



Entrée [27]: `for i in range(10):
 plt.scatter(X_digits_acp[y_digits==i, 0], X_digits_acp[y_digits==i, 1],
 plt.show()`



conclusion

L'interprétation d'un graphique linéaire ou non linéaire dépend du type de relation entre les variables représentées et cela implique l'analyse de la forme de la relation entre les variables, la force de la corrélation, et la possibilité de prévoir ou d'extrapoler les données. En ce qui concerne le nuage de points est une excellente visualisation pour explorer visuellement la relation entre deux variables. Il offre des informations sur la corrélation, la dispersion et les tendances dans les données. Et Un histogramme est une représentation visuelle puissante de la distribution des données. En l'interprétant correctement, vous pouvez obtenir des informations sur la centralité, la dispersion et la forme de la distribution. L'interprétation d'un diagramme à barres dépend du contexte spécifique de vos données. Il est important de comprendre les axes, les catégories représentées et l'objectif général du graphique pour en tirer des conclusions significatives.

Génération de fichier txt

Entrée [28]: `import nbformat`

Entrée [29]: `with open('TP_MBANGU_NDUNGA_Elie.ipynb', 'r', encoding='utf-8') as f:
 notebook = nbformat.read(f, as_version=4)`

Entrée [30]: `text_content = ""
for cell in notebook.cells:
 if cell.cell_type == 'code':
 text_content += f"### CODE CELL ###\n{cell.source}\n\n"
 elif cell.cell_type == 'markdown':
 text_content += f"### MARKDOWN CELL ###\n{cell.source}\n\n"`

Entrée [31]: `with open('TP_MBANGU_NDUNGA_Elie.txt', 'w', encoding='utf-8') as f:
 f.write(text_content)`

Entrée [32]: `print("Le contenu du notebook a été extrait avec succès dans le fichier 'TP_`

Le contenu du notebook a été extrait avec succès dans le fichier 'TP_MBANG
U_NDUNGA_Elie.txt'.

Entrée []: