In [ ]:

```
#install firefox, geckodriver, and selenium
!apt-get update
!pip install selenium
!apt install firefox-geckodriver

import time
from google.colab import files
```

```
Hit:1 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRelea
se
Hit:2 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:7 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Hit:8 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:9 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Hit:10 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Reading package lists... Done
Requirement already satisfied: selenium in /usr/local/lib/python3.10/dist-packages (4.17.
2)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in /usr/local/lib/python3.10/dist-
packages (from selenium) (2.0.7)
Requirement already satisfied: trio~=0.17 in /usr/local/lib/python3.10/dist-packages (fro
m selenium) (0.24.0)
Requirement already satisfied: trio-websocket~=0.9 in /usr/local/lib/python3.10/dist-pack
ages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in /usr/local/lib/python3.10/dist-packa
ges (from selenium) (2023.11.17)
Requirement already satisfied: typing_extensions>=4.9.0 in /usr/local/lib/python3.10/dist
-packages (from selenium) (4.9.0)
Requirement already satisfied: attrs>=20.1.0 in /usr/local/lib/python3.10/dist-packages (
from trio~=0.17->selenium) (23.2.0)
Requirement already satisfied: sortedcontainers in /usr/local/lib/python3.10/dist-package
s (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from trio
~=0.17->selenium) (3.6)
Requirement already satisfied: outcome in /usr/local/lib/python3.10/dist-packages (from t
rio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sniffio>=1.3.0 in /usr/local/lib/python3.10/dist-packages
(from trio~=0.17->selenium) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages
(from trio~=0.17->selenium) (1.2.0)
Requirement already satisfied: wsproto>=0.14 in /usr/local/lib/python3.10/dist-packages (
from trio-websocket~=0.9->selenium) (1.2.0)
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in /usr/local/lib/python3.10/d
ist-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: h11<1,>=0.9.0 in /usr/local/lib/python3.10/dist-packages (
from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package firefox-geckodriver is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  firefox

E: Package 'firefox-geckodriver' has no installation candidate
```

In [ ]:

```
from selenium import webdriver
import time
import sys
import json
```

```
import collections
import csv
import sys
import time
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import Select
from selenium.webdriver import ActionChains
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
import requests
from bs4 import BeautifulSoup
import pandas as pd
from google.colab import files




binary = '/usr/bin/firefox'
options = webdriver.FirefoxOptions()
options.binary = binary
options.add_argument('start-maximized')
options.add_argument('--headless')
driver = webdriver.Firefox(options=options)


page = 1
temp1 = []
temp2 = []
limite = 40
product_name_list = []
product_prices = []


while limite <=640 :

    url = f"https://fr.boohoo.com/hommes/nouveautes?start={limite}&sz=40"
    #here we try to avoid ip request limit
    try:

      driver.get(url)
      print(url)
      null = None

      try:
          element = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.C
SS_SELECTOR, "a.b-product_tile-link")))
      finally:
          # Extraire les noms des produits à partir de la page
          product_name_elements = driver.find_elements(By.CSS_SELECTOR, "a.b-product_til
e-link")
          for elem in product_name_elements:
            product_names_json = json.loads(elem.get_attribute('data-analytics'))
            product_name_list.append(product_names_json['name'])
      print('ça a marché')

      try:
        element = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CSS
_SELECTOR, "span.b-price-item")))
      finally:
        # Extraire les prix des produits à partir de la page
        product_price_elements = driver.find_elements(By.CSS_SELECTOR, "span.b-price-ite
m")
        product_prices_temp = [float(elem.get_attribute('content')) for elem in product_
price_elements]
        product_prices.extend(product_prices_temp)
```

```
      limite+=40

    except:
        print("Connection refused by the server..")
        print("Let me sleep for 5 seconds")
        print("ZZzzzz...")
        time.sleep(5)
        print("Was a nice sleep, now let me continue...")
        continue

df = pd.DataFrame(list(zip(product_name_list, product_prices)), columns=['Nom','Prix'])

df.to_csv('/content/bohoo_full.csv', index=False,encoding='utf-8-sig')
```

```
<ipython-input-3-bc0abc69080b>:31: DeprecationWarning: use binary_location instead
  options.binary = binary
```

```
https://fr.boohoo.com/hommes/nouveautes?start=40&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=80&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=120&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=160&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=200&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=240&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=280&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=320&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=360&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=400&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=440&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=480&sz=40
Connection refused by the server..
Let me sleep for 5 seconds
ZZzzzz...

---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-3-bc0abc69080b> in <cell line: 45>()
     60           for elem in product_name_elements:
---> 61             product_names_json = json.loads(elem.get_attribute('data-analytics'))
     62             product_name_list.append(product_names_json['name'])

/usr/local/lib/python3.10/dist-packages/selenium/webdriver/remote/webelement.py in get_at
tribute(self, name)
    177             _load_js()
--> 178         attribute_value = self.parent.execute_script(
    179             f"/* getAttribute */return ({getAttribute_js}).apply(null, arguments)
;", self, name

/usr/local/lib/python3.10/dist-packages/selenium/webdriver/remote/webdriver.py in execute
_script(self, script, *args)
    406
--> 407         return self.execute(command, {"script": script, "args": converted_args})
["value"]
    408

/usr/local/lib/python3.10/dist-packages/selenium/webdriver/remote/webdriver.py in execute
(self, driver_command, params)
    344
--> 345         response = self.command_executor.execute(driver_command, params)
    346         if response:

/usr/local/lib/python3.10/dist-packages/selenium/webdriver/remote/remote_connection.py in
```

```
/usr/local/lib/python3.10/dist-packages/selenium/webdriver/remote/remote_connection.py in
execute(self, command, params)
    301         LOGGER.debug("%s %s %s", command_info[0], url, str(trimmed))
--> 302         return self._request(command_info[0], url, body=data)
    303

/usr/local/lib/python3.10/dist-packages/selenium/webdriver/remote/remote_connection.py in
_request(self, method, url, body)
    321         if self.keep_alive:
--> 322             response = self._conn.request(method, url, body=body, headers=header
s)
    323             statuscode = response.status

/usr/local/lib/python3.10/dist-packages/urllib3/_request_methods.py in request(self, meth
od, url, body, fields, headers, json, **urlopen_kw)
    117         else:
--> 118             return self.request_encode_body(
    119                 method, url, fields=fields, headers=headers, **urlopen_kw

/usr/local/lib/python3.10/dist-packages/urllib3/_request_methods.py in request_encode_bod
y(self, method, url, fields, headers, encode_multipart, multipart_boundary, **urlopen_kw)
    216
--> 217         return self.urlopen(method, url, **extra_kw)

/usr/local/lib/python3.10/dist-packages/urllib3/poolmanager.py in urlopen(self, method, u
rl, redirect, **kw)
    442         else:
--> 443             response = conn.urlopen(method, u.request_uri, **kw)
    444

/usr/local/lib/python3.10/dist-packages/urllib3/connectionpool.py in urlopen(self, method
, url, body, headers, retries, redirect, assert_same_host, timeout, pool_timeout, release
_conn, chunked, body_pos, preload_content, decode_content, **response_kw)
    790             # Make the request on the HTTPConnection object
--> 791             response = self._make_request(
    792                 conn,

/usr/local/lib/python3.10/dist-packages/urllib3/connectionpool.py in _make_request(self,
conn, method, url, body, headers, retries, timeout, chunked, response_conn, preload_conte
nt, decode_content, enforce_content_length)
    536         try:
--> 537             response = conn.getresponse()
    538         except (BaseSSLError, OSError) as e:

/usr/local/lib/python3.10/dist-packages/urllib3/connection.py in getresponse(self)
    460         # Get the response from http.client.HTTPConnection
--> 461         httplib_response = super().getresponse()
    462

/usr/lib/python3.10/http/client.py in getresponse(self)
   1374             try:
-> 1375                 response.begin()
   1376             except ConnectionError:

/usr/lib/python3.10/http/client.py in begin(self)
    317         while True:
--> 318             version, status, reason = self._read_status()
    319             if status != CONTINUE:

/usr/lib/python3.10/http/client.py in _read_status(self)
    278     def _read_status(self):
--> 279         line = str(self.fp.readline(_MAXLINE + 1), "iso-8859-1")
    280         if len(line) > _MAXLINE:

/usr/lib/python3.10/socket.py in readinto(self, b)
    704             try:
--> 705                 return self._sock.recv_into(b)
    706             except timeout:

KeyboardInterrupt:

During handling of the above exception, another exception occurred:
```

```
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-3-bc0abc69080b> in <cell line: 45>()
     77         print("Let me sleep for 5 seconds")
     78         print("ZZzzzz...")
---> 79         time.sleep(5)
     80         print("Was a nice sleep, now let me continue...")
     81         continue

KeyboardInterrupt:
```

In [ ]:

```
from selenium import webdriver
import time
import sys
import json
import collections
import csv
import sys
import time
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import Select
from selenium.webdriver import ActionChains
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
import requests
from bs4 import BeautifulSoup
import pandas as pd
from google.colab import files




binary = '/usr/bin/firefox'
options = webdriver.FirefoxOptions()
options.binary = binary
options.add_argument('start-maximized')
options.add_argument('--headless')
driver = webdriver.Firefox(options=options)


page = 1
temp1 = []
temp2 = []
limite = 640
product_name_list = []
product_prices = []


while limite <=1280 :

    url = f"https://fr.boohoo.com/hommes/nouveautes?start={limite}&sz=40"
    #here we try to avoid ip request limit
    try:

      driver.get(url)
      print(url)
      null = None

      try:
          element = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.C
SS_SELECTOR, "a.b-product_tile-link")))
      finally:
          # Extraire les noms des produits à partir de la page
```

```
            product_name_elements = driver.find_elements(By.CSS_SELECTOR, "a.b-product_til
e-link")
            for elem in product_name_elements:
              product_names_json = json.loads(elem.get_attribute('data-analytics'))
              product_name_list.append(product_names_json['name'])
        print('ça a marché')

        try:
          element = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.CSS
_SELECTOR, "span.b-price-item")))
        finally:
          # Extraire les prix des produits à partir de la page
          product_price_elements = driver.find_elements(By.CSS_SELECTOR, "span.b-price-ite
m")
          product_prices_temp = [float(elem.get_attribute('content')) for elem in product_
price_elements]
          product_prices.extend(product_prices_temp)

        limite+=40

      except:
          print("Connection refused by the server..")
          print("Let me sleep for 5 seconds")
          print("ZZzzzz...")
          time.sleep(5)
          print("Was a nice sleep, now let me continue...")
          continue

df = pd.DataFrame(list(zip(product_name_list, product_prices)), columns=['Nom','Prix'])

df.to_csv('/content/bohoo_full2.csv', index=False,encoding='utf-8-sig')
```

```
<ipython-input-5-6a989f3e51a2>:31: DeprecationWarning: use binary_location instead
  options.binary = binary
```

```
https://fr.boohoo.com/hommes/nouveautes?start=640&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=680&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=720&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=760&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=800&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=840&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=880&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=920&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=960&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=1000&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=1040&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=1080&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=1120&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=1160&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=1200&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=1240&sz=40
ça a marché
https://fr.boohoo.com/hommes/nouveautes?start=1280&sz=40
ça a marché
```

In [ ]:

```
import pandas as pd
booho1 = pd.read_csv('/content/bohoo_full2.csv')
```

In [ ]:

```
booho1
```

Out[ ]:

	Nom	Prix
0	Grande taille - Jean large délavé	50.0
1	Survêtement de sport zippé avec jogging - MAN ...	50.0
2	Pantalon cargo délavé à boutons pression	28.0
3	T-shirt oversize à bords bruts	20.0
4	Chemise unie en viscose à manches courtes	15.0
...	...	...
835	T-shirt épais à col contrastant	20.0
836	Jogging large côtelé fendu	20.0
837	Short oversize en jersey	32.0
838	Sweat à capuche oversize à imprimé chien	10.0
839	Bonnet à imprimé espace	32.0

840 rows × 2 columns

In [ ]:

```
def get_type(nom):
  split_name = nom.split()

  res = split_name[0]

  if(res=='Tall'):
    res='Survêtement'

  return res

booho1['Type'] = booho1.Nom.apply(get_type)

booho1
```

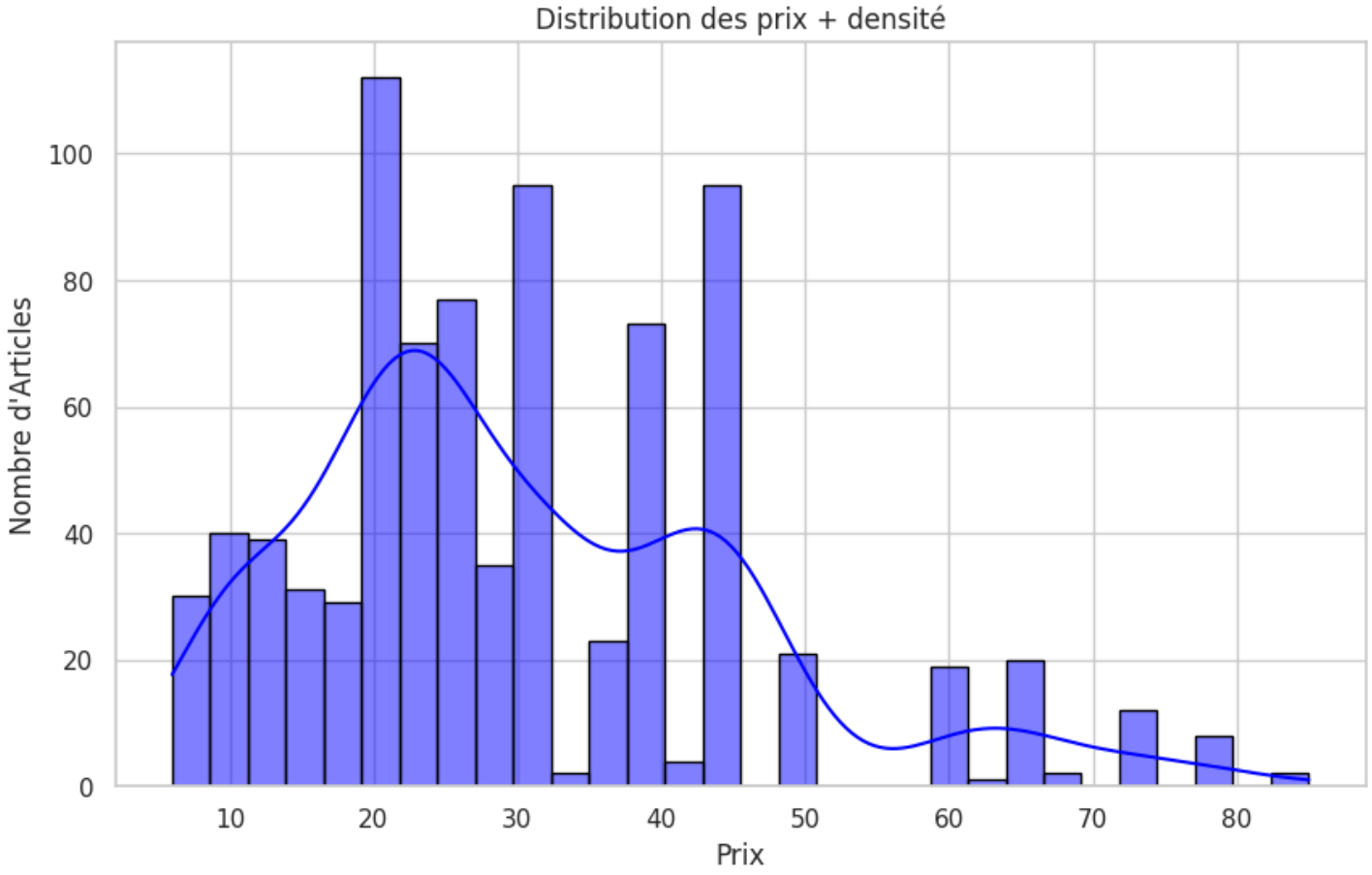
Out[ ]:

	Nom	Prix	Type
0	Grande taille - Jean large délavé	50.0	Grande
1	Survêtement de sport zippé avec jogging - MAN ...	50.0	Survêtement
2	Pantalon cargo délavé à boutons pression	28.0	Pantalon
3	T-shirt oversize à bords bruts	20.0	T-shirt
4	Chemise unie en viscose à manches courtes	15.0	Chemise
...	...	...	...
835	T-shirt épais à col contrastant	20.0	T-shirt
836	Jogging large côtelé fendu	20.0	Jogging
837	Short oversize en jersey	32.0	Short
838	Sweat à capuche oversize à imprimé chien	10.0	Sweat
839	Bonnet à imprimé espace	32.0	Bonnet

840 rows × 3 columns

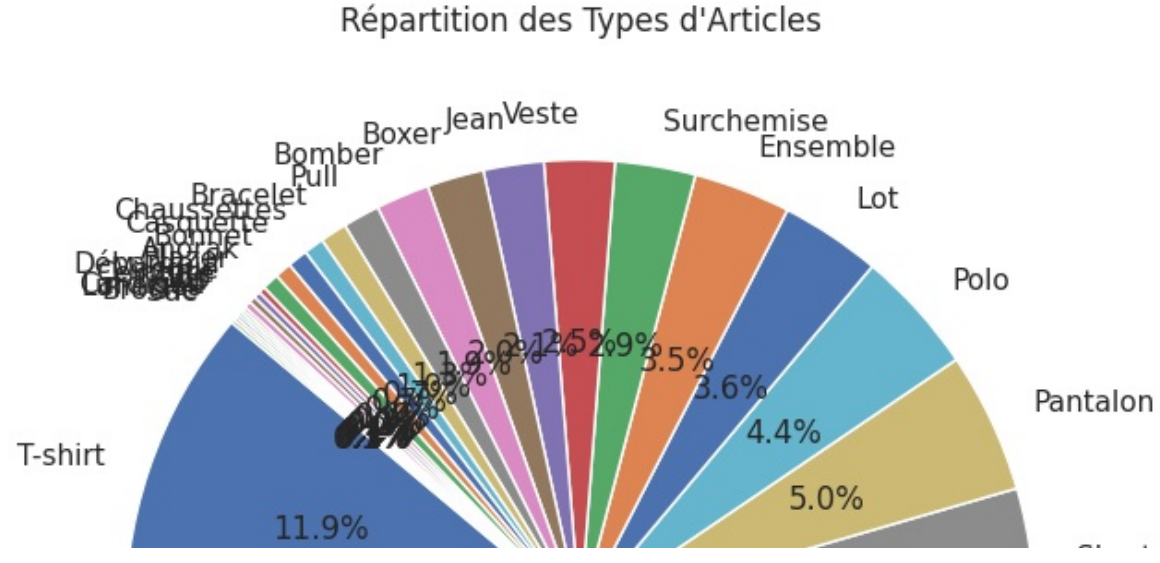
In [ ]:

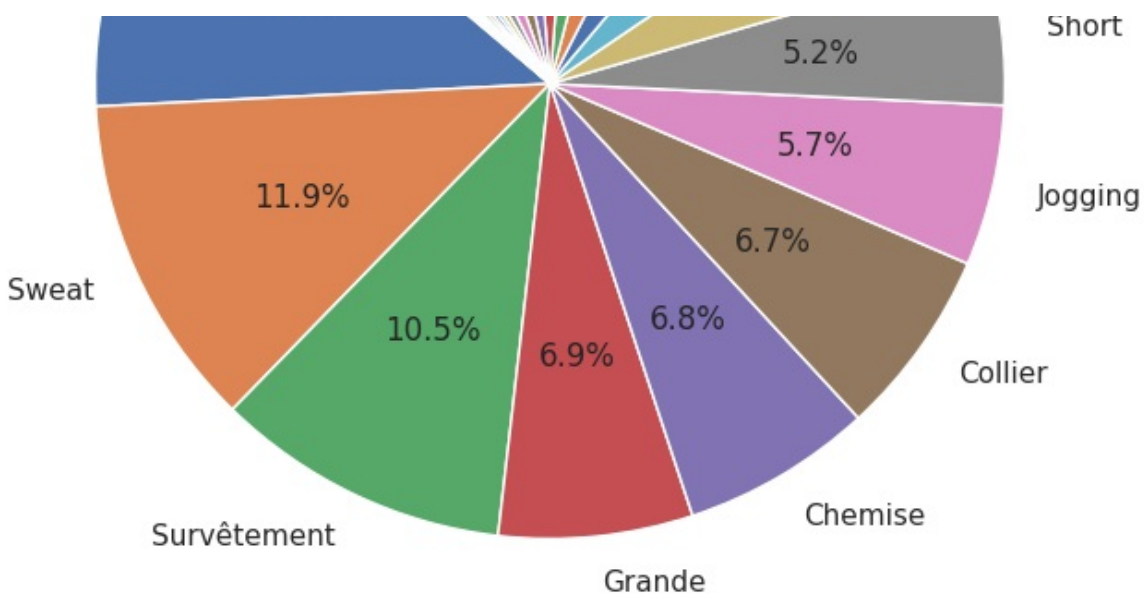
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.histplot(booho1['Prix'], kde=True, color="blue", bins=30, edgecolor='black')
plt.title('Distribution des prix + densité')
plt.xlabel('Prix')
plt.ylabel('Nombre d\'Articles')
plt.show()
```



In [ ]:

```
type_counts = booho1['Type'].value_counts()
plt.figure(figsize=(8, 8))
plt.pie(type_counts, labels=type_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Répartition des Types d\'Articles')
plt.show()
```

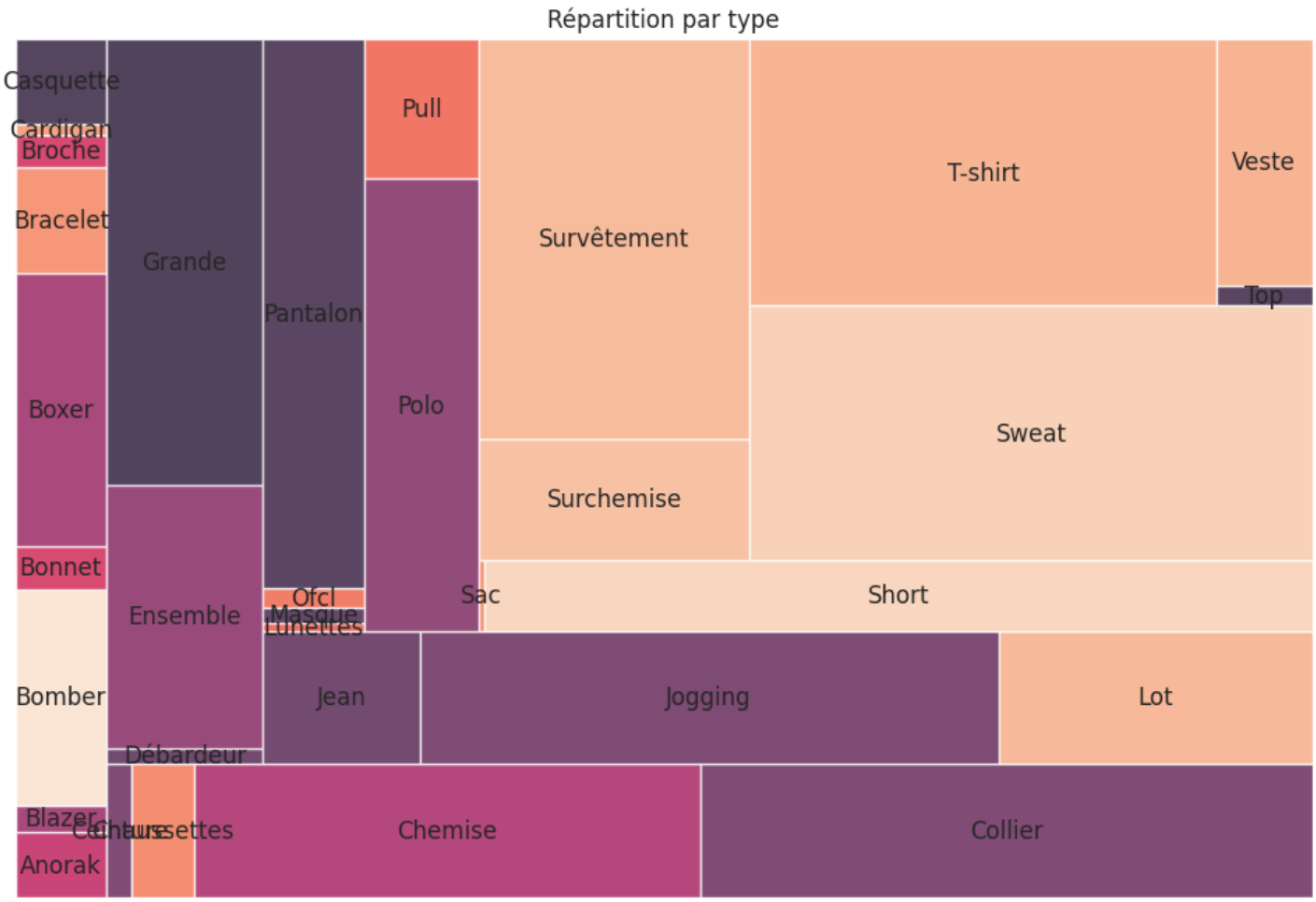




In [ ]:

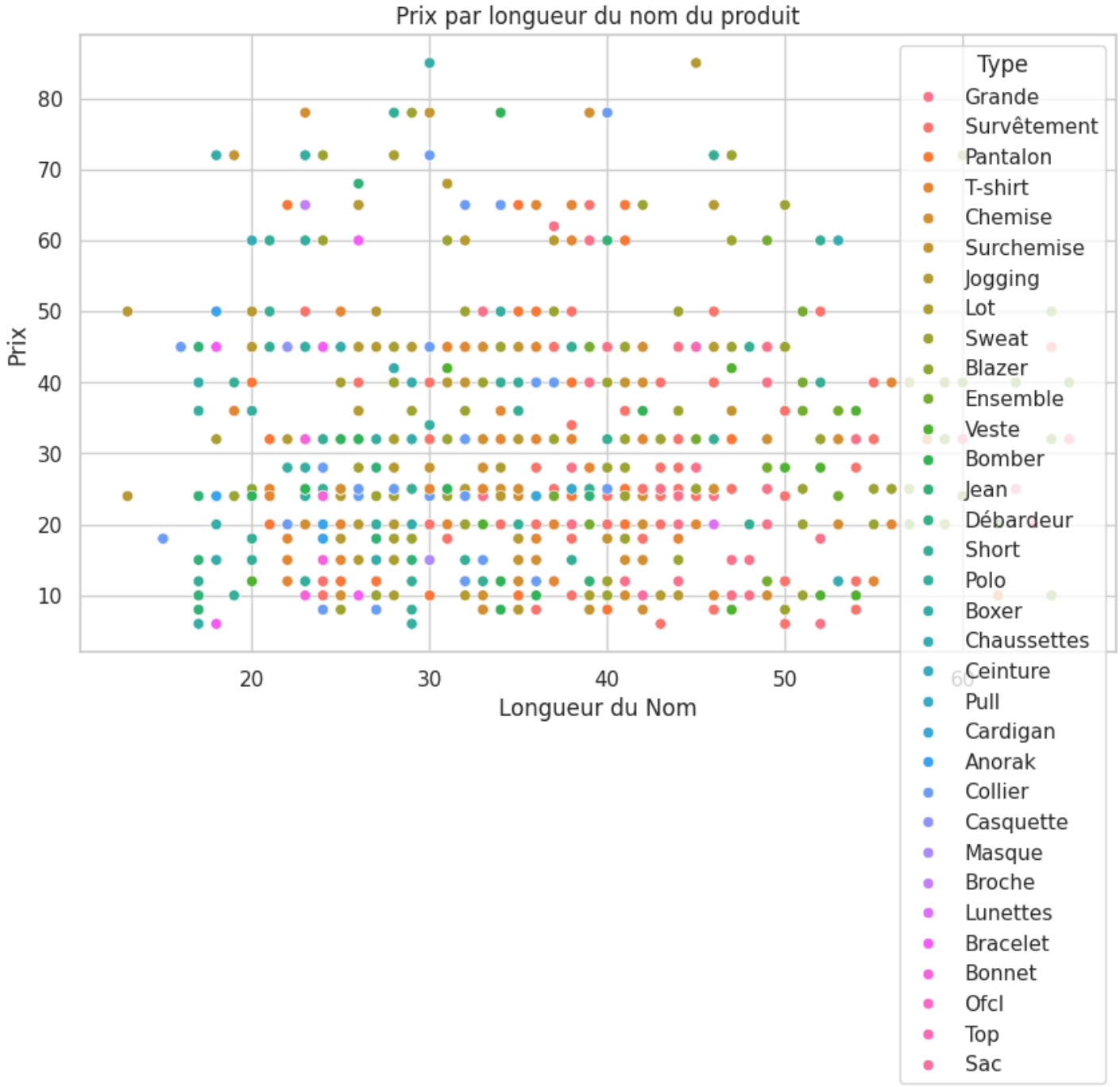
```
#!pip install squarify

import squarify

type_totals = booho1.groupby('Type')['Prix'].sum().reset_index()
plt.figure(figsize=(12, 8))
squarify.plot(sizes=type_totals['Prix'], label=type_totals['Type'], alpha=0.8)
plt.axis('off')
plt.title('Répartition par type')
plt.show()
```

```
Requirement already satisfied: squarify in /usr/local/lib/python3.10/dist-packages (0.4.3
)
```



In [ ]:

```
booho1['Nom_Longueur'] = booho1['Nom'].apply(len)

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Nom_Longueur', y='Prix', data=booho1, hue='Type')
plt.title('Prix par longueur du nom du produit')
plt.xlabel('Longueur du Nom')
plt.ylabel('Prix')
plt.legend(title='Type')
plt.show()
```

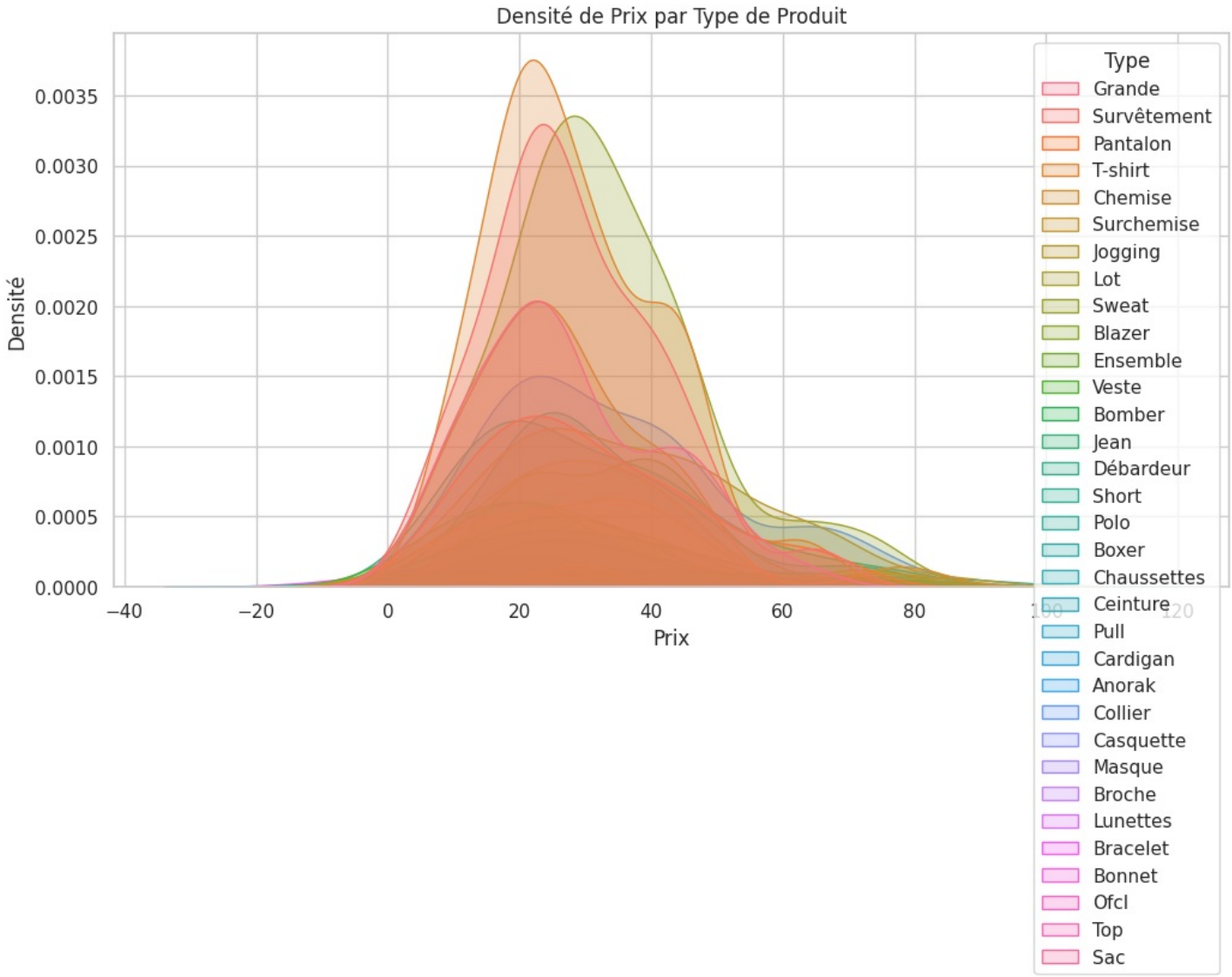


In [ ]:

```
plt.figure(figsize=(12, 6))
sns.kdeplot(data=booho1, x='Prix', hue='Type', fill=True)
plt.title('Densité de Prix par Type de Produit')
plt.xlabel('Prix')
plt.ylabel('Densité')
plt.show()
```

```
<ipython-input-20-00942c2be454>:2: UserWarning: Dataset has 0 variance; skipping density
estimate. Pass `warn_singular=False` to disable this warning.
  sns.kdeplot(data=booho1, x='Prix', hue='Type', fill=True)
<ipython-input-20-00942c2be454>:2: UserWarning: Dataset has 0 variance; skipping density
estimate. Pass `warn_singular=False` to disable this warning.
  sns.kdeplot(data=booho1, x='Prix', hue='Type', fill=True)
```

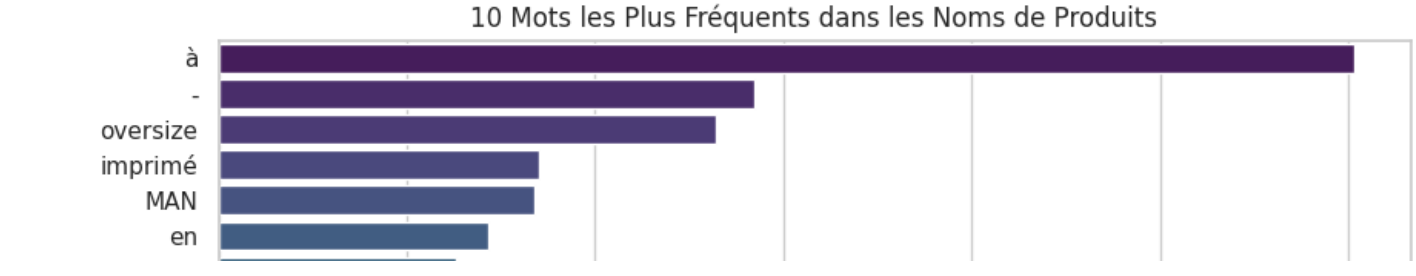
```
<ipython-input-20-00942c2be454>:2: UserWarning: Dataset has 0 variance; skipping density
estimate. Pass `warn_singular=False` to disable this warning.
  sns.kdeplot(data=booho1, x='Prix', hue='Type', fill=True)
```

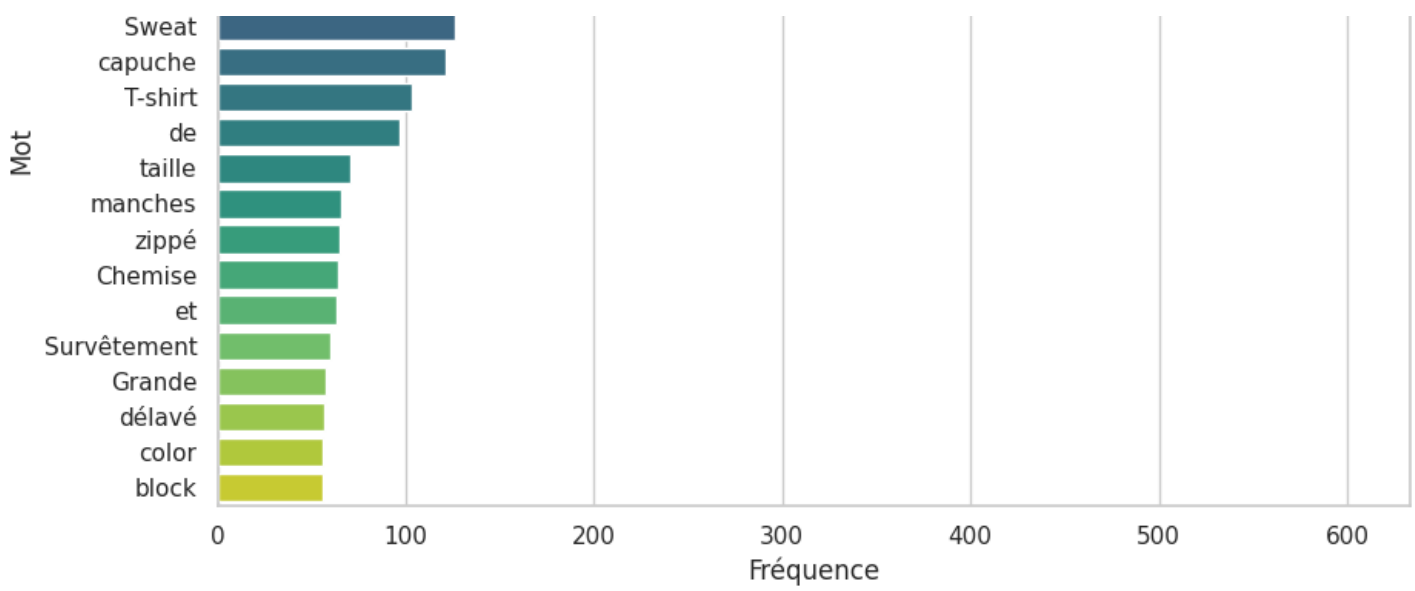


In [ ]:

```
from collections import Counter

words = Counter(" ".join(booho1['Nom']).split())
most_common_words = words.most_common(20)

words_df = pd.DataFrame(most_common_words, columns=['Mot', 'Fréquence'])
plt.figure(figsize=(10, 6))
sns.barplot(x='Fréquence', y='Mot', data=words_df, palette='viridis')
plt.title('10 Mots les Plus Fréquents dans les Noms de Produits')
plt.xlabel('Fréquence')
plt.ylabel('Mot')
plt.show()
```

```
<ipython-input-27-7b85d0986fa5>:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. A
ssign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Fréquence', y='Mot', data=words_df, palette='viridis')
```





Un peu de NLP:

In [ ]:

```
from sklearn.feature_extraction.text import CountVectorizer

booho_NLP = pd.read_csv('/content/bohoo_full2.csv')
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(booho_NLP['Nom'])
feature_df = pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names_out())

print(feature_df.head())
```

```
   aaliyah  active  aigle  air  ajusté  ample  angeles  anorak  apparentes  \
0        0       0      0    0       0      0        0       0           0
1        0       1      0    0       0      0        0       0           0
2        0       0      0    0       0      0        0       0           0
3        0       0      0    0       0      0        0       0           0
4        0       0      0    0       0      0        0       0           0

   argentée  ...  zippé  zippée  écusson  écussons  élastiquée  épais  \
0         0  ...      0       0        0         0           0      0
1         0  ...      1       0        0         0           0      0
2         0  ...      0       0        0         0           0      0
3         0  ...      0       0        0         0           0      0
4         0  ...      0       0        0         0           0      0

   épaisse  épaisses  étagés  œuf
0        0         0       0    0
1        0         0       0    0
2        0         0       0    0
3        0         0       0    0
4        0         0       0    0

[5 rows x 372 columns]
```

In [ ]:

```
word_sum = feature_df.sum(axis=0)
sorted_words = word_sum.sort_values(ascending=False)
print(sorted_words.head(20))
```

```
oversize       230
imprimé        153
man            152
sweat          130
en             112
capuche        111
shirt          108
de              78
```

```
taille          65
chemise         63
manches         62
zippé           60
survêtement     60
jogging         54
et              53
grande          53
block           50
délavé          50
color           50
pantalon        49
dtype: int64
```

In [ ]:

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

wordcloud = WordCloud(width=800, height=400, background_color ='white').generate_from_fr
equencies(word_sum)
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

In [ ]:

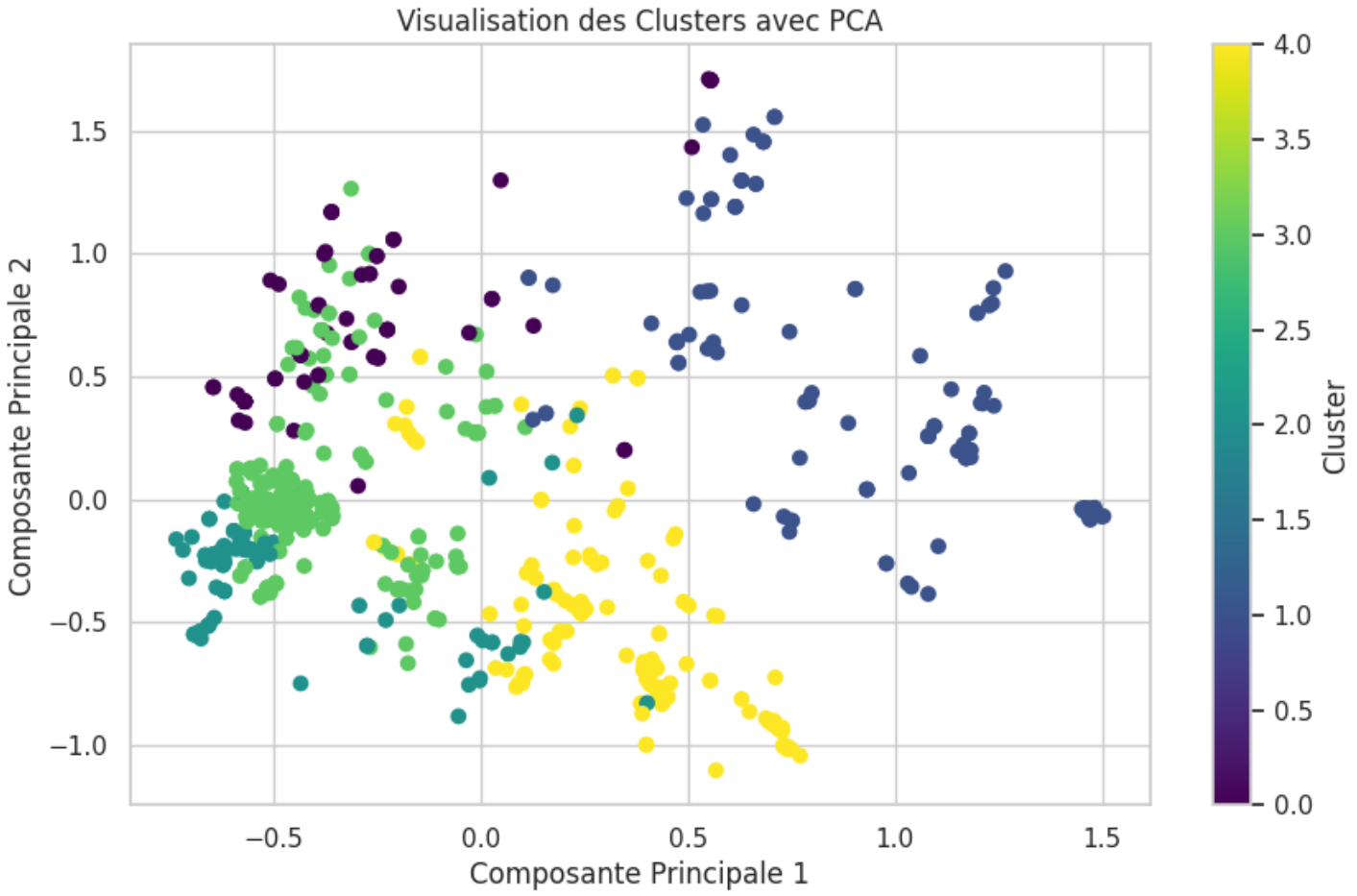
```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=5, random_state=0).fit(feature_df)

booho_NLP['Cluster'] = kmeans.labels_
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: Th
e default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_ini
t` explicitly to suppress the warning
  warnings.warn(
```

In [ ]:

```
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
reduced_features = pca.fit_transform(feature_df)

plt.figure(figsize=(10, 6))
```

```
plt.scatter(reduced_features[:, 0], reduced_features[:, 1], c=booho_NLP['Cluster'], cmap
='viridis', marker='o')
plt.title('Visualisation des Clusters avec PCA')
plt.xlabel('Composante Principale 1')
plt.ylabel('Composante Principale 2')
plt.colorbar(label='Cluster')
plt.show()
```



ML pour prédire la colonne le prix de l'article à partir de son nom.

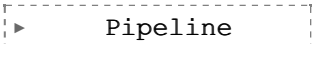
In [ ]:

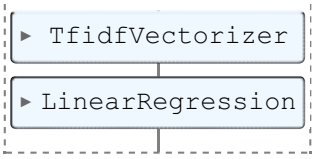
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

pipeline = Pipeline([
    ('tfidf', TfidfVectorizer(stop_words='english')),
    ('regressor', LinearRegression())
])

X = booho1['Nom']
y = booho1['Prix']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
)
```

In [ ]:

```
pipeline.fit(X_train, y_train)
```

Out[ ]:





In [ ]:

```
y_pred = pipeline.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

print(f"RMSE: {rmse}")
print(f"R²: {r2}")
```

```
RMSE: 19.77195067377321
R²: -0.7233982581070548
```

In [ ]:

```
examples = ["Grande taille - Jean large délavé", "Pantalon cargo délavé à boutons pressio
n", "Jogging large côtelé fendu"]
# A REMPLIR POUR TESTER

predicted_prices = pipeline.predict(examples)

for name, price in zip(examples, predicted_prices):
    print(f"Nom: {name}, Prix prédit: {price:.2f}")
```

```
Nom: Grande taille - Jean large délavé, Prix prédit: 33.71
Nom: Pantalon cargo délavé à boutons pression, Prix prédit: 37.66
Nom: Jogging large côtelé fendu, Prix prédit: 41.82
```

In [ ]:

```

```

Problématique => Proposer une liste d'article Wedressfair similaire à celui que le client vient de choisir sur Boohoo.

In [3]:

```
import pandas as pd
Wedressfair =  pd.read_csv('/content/Wedressfair.csv')
Wedressfair #j'ai importe un csv de Wedressfair pour l'exemple car j'ai uniquement scrapp
é  Boohoo, je n'ai pas eu le temps de faire Wedressfair
```

Out[3]:

	Nom	Prix	Type
0	Homme T-shirt à liséré contrastant à bouton	10.49	T-shirt
1	SHEIN Homme T-shirt graphique de slogan	8.99	T-shirt
2	Homme 4 pièces T-shirt unicolore	33.49	Undifined
3	Homme Polo à blocs de couleur	13.99	Polo
4	Homme T-shirt dessin animé	9.99	T-shirt
...	...	...	...
4663	Extended Sizes Homme T-shirt à lettres à blocs...	8.50	T-shirt
4664	DAZY Homme T-shirt unicolore	10.49	T-shirt
4665	DAZY Homme Chemise à carreaux	17.49	Chemise
4666	DAZY Homme T-shirt unicolore col rond	14.49	T-shirt
4667	SHEIN Homme T-shirt à imprimé expression et sl...	12.49	T-shirt

4668 rows × 3 columns

In [ ]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

article_num = int(input("Entrez le numéro de l'article que vous voulez : "))
selected_article = booho1.iloc[article_num]
print(f"Vous avez sélectionné : {selected_article['Nom']} au prix de {selected_article['P
rix']}€. Est-ce correct ? (oui/non)")

confirmation = input()
if confirmation.lower() != 'oui':
    print("Veuillez recommencer.")
else:
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(Wedressfair['Nom'].append(pd.Series(selected
_article['Nom'])))
    cosine_sim = cosine_similarity(tfidf_matrix[-1], tfidf_matrix[:-1])

    #les 5 articles les plus similaires
    similar_articles_indices = cosine_sim.argsort()[0][-6:-1]
    similar_articles = Wedressfair.iloc[similar_articles_indices]

    print("⚠⚠ ATTENTION, avant de finaliser votre achat sur Boohoo, veuillez considérer l
'impact environnemental et social de leurs pratiques ⚠⚠.\n"
          "Boohoo a été critiqué pour ses violations des droits du travail et son non res
pect des normes environnementales,\n \n"
          "IL N'EST PAS TROP TARD: Nous avons trouvé des articles similaires qui pourraie
nt vous intéresser, provenant de sources plus éthiques et durables :")

    for index, article in similar_articles.iterrows():
        print(f"\nNom de l'article : {article['Nom']}\n"
              f"Prix : {article['Prix']}€\n Lien: https://www.wedressfair.fr/XXXXXXXXXXX
XXXXXXXXX\n"
              f"Type : {article['Type']}")

    print("\nNous vous encourageons vivement à considérer ces alternatives plus respectue
uses de l'environnement et des droits des travailleurs.")
```

```
Entrez le numéro de l'article que vous voulez : 4
Vous avez sélectionné : Chemise unie en viscose à manches courtes au prix de 15.0€. Est-c
e correct ? (oui/non)
oui
⚠⚠ ATTENTION, avant de finaliser votre achat sur Boohoo, veuillez considérer l'impact env
ironnemental et social de leurs pratiques ⚠⚠.
Boohoo a été critiqué pour ses violations des droits du travail et son non respect des no
rmes environnementales,

IL N'EST PAS TROP TARD: Nous avons trouvé des articles similaires qui pourraient vous int
éresser, provenant de sources plus éthiques et durables :

Nom de l'article : SHEIN T-shirt dégradé à manches courtes
Prix : 10.49€
 Lien: https://www.wedressfair.fr/XXXXXXXXXXXXXXXXXXX
Type : T-shirt

Nom de l'article : T-shirt à dessin animé à manches courtes
Prix : 10.99€
 Lien: https://www.wedressfair.fr/XXXXXXXXXXXXXXXXXXX
Type : Undifined

Nom de l'article : Homme T-shirt unicolore manches courtes
Prix : 8.99€
 Lien: https://www.wedressfair.fr/XXXXXXXXXXXXXXXXXXX
Type : T-shirt

Nom de l'article : Homme T-shirt à lettres à manches courtes
Prix : 11.49€
 Lien: https://www.wedressfair.fr/XXXXXXXXXXXXXXXXXXX
```

```
Type : T-shirt

Nom de l'article : SHEIN Homme T-shirt unicolore col en V manches courtes
Prix : 7.49€
 Lien: https://www.wedressfair.fr/XXXXXXXXXXXXXXXXXXXX
Type : T-shirt

Nous vous encourageons vivement à considérer ces alternatives plus respectueuses de l'env
ironnement et des droits des travailleurs.
```

```
<ipython-input-48-c1d6a166d350>:13: FutureWarning: The series.append method is deprecated
and will be removed from pandas in a future version. Use pandas.concat instead.
  tfidf_matrix = vectorizer.fit_transform(Wedressfair['Nom'].append(pd.Series(selected_ar
ticle['Nom'])))
```

In [ ]:

```
```

LANCEMENT DE L'API pour rendre les données publics

In [ ]:

```
!wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
!unzip -o ngrok-stable-linux-amd64.zip
```

```
--2024-01-14 18:26:36--  https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zi
p
Resolving bin.equinox.io (bin.equinox.io)... 52.202.168.65, 54.161.241.46, 54.237.133.81,
...
Connecting to bin.equinox.io (bin.equinox.io)|52.202.168.65|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13921656 (13M) [application/octet-stream]
Saving to: 'ngrok-stable-linux-amd64.zip'

ngrok-stable-linux- 100%[===================>]  13.28M  54.0MB/s    in 0.2s

2024-01-14 18:26:36 (54.0 MB/s) - 'ngrok-stable-linux-amd64.zip' saved [13921656/13921656
]

Archive:  ngrok-stable-linux-amd64.zip
  inflating: ngrok
```

In [ ]:

```
!pip install fastapi nest-asyncio pyngrok uvicorn
```

```
Requirement already satisfied: fastapi in /usr/local/lib/python3.10/dist-packages (0.109.
0)
Requirement already satisfied: nest-asyncio in /usr/local/lib/python3.10/dist-packages (1
.5.8)
Collecting pyngrok
  Downloading pyngrok-7.0.5-py3-none-any.whl (21 kB)
Requirement already satisfied: uvicorn in /usr/local/lib/python3.10/dist-packages (0.25.0
)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7
.4 in /usr/local/lib/python3.10/dist-packages (from fastapi) (1.10.13)
Requirement already satisfied: starlette<0.36.0,>=0.35.0 in /usr/local/lib/python3.10/dis
t-packages (from fastapi) (0.35.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist
-packages (from fastapi) (4.9.0)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packages (from py
ngrok) (6.0.1)
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.10/dist-packages (fro
m uvicorn) (8.1.7)
Requirement already satisfied: h11>=0.8 in /usr/local/lib/python3.10/dist-packages (from
uvicorn) (0.14.0)
Requirement already satisfied: anyio<5,>=3.4.0 in /usr/local/lib/python3.10/dist-packages
(from starlette<0.36.0,>=0.35.0->fastapi) (3.7.1)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from
anyio<5,>=3.4.0->starlette<0.36.0,>=0.35.0->fastapi) (3.6)
```

```
anyio<5,>=3.4.0->starlette<0.36.0,>=0.35.0->fastapi) (3.6)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (f
rom anyio<5,>=3.4.0->starlette<0.36.0,>=0.35.0->fastapi) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages
(from anyio<5,>=3.4.0->starlette<0.36.0,>=0.35.0->fastapi) (1.2.0)
Installing collected packages: pyngrok
Successfully installed pyngrok-7.0.5
```

In [ ]:

```python
import pandas as pd
from fastapi import FastAPI
import Response

#shein = pd.read_csv("files/shein.csv")
boohoo = pd.read_csv("/content/bohoo_full2.csv")
#df = (boohoo.append(shein)).reset_index().drop('index', axis = 1) - Si il y a d'autre fi
chier alors faire des appends
df = (boohoo).reset_index().drop('index', axis = 1)
app = FastAPI()

@app.get("/")
async def root():
    return {"message": "Everything is working!"}


@app.get("/all_clothes")
async def root():
    return {
        df.to_string()
    }

@app.get("/cheapest_t_shirt")
async def root():
    sub_df = (df.loc[df['Type'] == 'T-shirt']).reset_index()
    sub_df.sort_values(by=['Prix'],inplace=True)
    return {
        sub_df.head(1).to_string()
    }

@app.get("/cheapest_clothes")
async def root():
    sub_df = (df.sort_values(by=['Prix'])).reset_index()
    return {
        sub_df.head(5).to_string()
    }
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-42-c5526a1fb6b0> in <cell line: 2>()
      1 import pandas as pd
----> 2 from fastapi import FastAPI
      3 import Response
      4
      5 #shein = pd.read_csv("files/shein.csv")

/usr/local/lib/python3.10/dist-packages/fastapi/__init__.py in <module>
      5 from starlette import status as status
      6
----> 7 from .applications import FastAPI as FastAPI
      8 from .background import BackgroundTasks as BackgroundTasks
      9 from .datastructures import UploadFile as UploadFile

/usr/local/lib/python3.10/dist-packages/fastapi/applications.py in <module>
     14 )
     15
---> 16 from fastapi import routing
     17 from fastapi.datastructures import Default, DefaultPlaceholder
     18 from fastapi.exception_handlers import (

/usr/local/lib/python3.10/dist-packages/fastapi/routing.py in <module>
     20 )
     21
```

```
---> 22 from fastapi import params
     23 from fastapi._compat import (
     24     ModelField,

/usr/local/lib/python3.10/dist-packages/fastapi/params.py in <module>
      3 from typing import Any, Callable, Dict, List, Optional, Sequence, Union
      4
----> 5 from fastapi.openapi.models import Example
      6 from pydantic.fields import FieldInfo
      7 from typing_extensions import Annotated, deprecated

/usr/local/lib/python3.10/dist-packages/fastapi/openapi/models.py in <module>
      2 from typing import Any, Callable, Dict, Iterable, List, Optional, Set, Type, Unio
n
      3
----> 4 from fastapi._compat import (
      5     PYDANTIC_V2,
      6     CoreSchema,

/usr/local/lib/python3.10/dist-packages/fastapi/_compat.py in <module>
     18 )
     19
---> 20 from fastapi.exceptions import RequestErrorModel
     21 from fastapi.types import IncEx, ModelNameMap, UnionType
     22 from pydantic import BaseModel, create_model

/usr/local/lib/python3.10/dist-packages/fastapi/exceptions.py in <module>
      4 from starlette.exceptions import HTTPException as StarletteHTTPException
      5 from starlette.exceptions import WebSocketException as StarletteWebSocketExceptio
n
----> 6 from typing_extensions import Annotated, Doc  # type: ignore [attr-defined]
      7
      8

ImportError: cannot import name 'Doc' from 'typing_extensions' (/usr/local/lib/python3.10
/dist-packages/typing_extensions.py)

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.
---------------------------------------------------------------------------
```