Лабораторная работ 5.
Частотный анализ.

Цель: провести частотный анализ выбранных данных и визуализировать облако слов.
Задачи:

1. Загрузить пред обработанные данные;
2. Выбрать нужные данные и сформировать новый датасет;
3. Удалить стоп слова;
4. Выделить леммы;
5. Визуализировать облако слов.

Пример выполнения лабораторной работы.

Выведем первые пять строк из подготовленного датасета.

```
import pandas as pd
import numpy as np
import nltk
import pymorphy2
from matplotlib import pyplot as plt
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from mlxtend.preprocessing import TransactionEncoder
from wordcloud import WordCloud
df = pd.read_csv('lab5.csv',encoding='cp1251')
print(df.head(5))
```

Результат:

	id	gender	age	income	product
0	1	Male	19	15	cigarettes
1	2	Male	21	15	milk
2	3	Female	20	16	chocolate
3	4	Female	23	16	milk
4	5	Female	31	17	bread

В нашем случае нас интересуют столбей айди и столбец с продуктами. Выведем количество уникальных покупателей и количество уникальных продуктов.

```
df = pd.read_csv('lab5.csv',encoding='cp1251')
quantity_id = list(set(df['id']))
quantity_product = list(set(df['product']))
print(len(quantity_id), len(quantity_product))
```

Результат:

```
198 10
```

Сформируем наш список товаров в датасет, который необходим для частотного анализа. Сольем все товары каждого покупателя в один список.

```
df = pd.read_csv('lab5.csv',encoding='cp1251')
quantity_id = list(set(df['id']))
quantity_product = list(set(df['product']))
all_product = [[elem for elem in df[df['id'] == id]['product'] if elem in
quantity_product] for id in quantity_id]
print(all_product)
```

Результат:

```
[['cigarettes '], ['milk'], ['chocolate '], ['milk'], ['bread'], ['bread'], ['milk'], ['water '], ['water '], ['bread'], ['wa
ter '], ['water '], ['chocolate '], ['milk'], ['water '], ['oil '], ['chocolate '], ['oil '], ['chocolate '], ['chocolate '],
['cigarettes'], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerines '], ['cigarettes'], ['tangerines'], ['cigarettes'],
['water '], ['flakes '], ['tangerines '], ['cigarettes '], ['cigarettes '], ['milk'], ['chocolate '], ['milk'], ['bread'], ['
bread'], ['milk'], ['water '], ['water '], ['bread'], ['water '], ['water '], ['chocolate '], ['milk'], ['water '], ['oil '],
['chocolate '], ['oil '], ['chocolate '], ['chocolate '], ['cigarettes'], ['flakes '], ['cigarettes'], ['tangerines'], ['tang
erines '], ['cigarettes'], ['tangerines'], ['cigarettes'], ['water '], ['flakes '], ['tangerines '], ['cigarettes '], ['cigar
ettes '], ['milk'], ['chocolate '], ['milk'], ['bread'], ['bread'], ['milk'], ['water '], ['water '], ['bread'], ['water '],
['water '], ['chocolate '], ['milk'], ['water '], ['oil '], ['chocolate '], ['oil '], ['chocolate '], ['chocolate '], ['cigar
ettes'], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerines '], ['cigarettes'], ['tangerines'], ['cigarettes'], ['wate
r '], ['flakes '], ['tangerines '], ['cigarettes '], ['cigarettes '], ['milk'], ['chocolate '], ['milk'], ['bread'], ['bread'
], ['milk'], ['water '], ['water '], ['bread'], ['water '], ['water '], ['chocolate '], ['milk'], ['water '], ['oil '], ['cho
colate '], ['oil '], ['chocolate '], ['chocolate '], ['cigarettes'], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerine
s '], ['cigarettes'], ['tangerines'], ['cigarettes'], ['water '], ['flakes '], ['tangerines '], ['cigarettes '], ['cigarettes
'], ['milk'], ['chocolate '], ['milk'], ['bread'], ['bread'], ['milk'], ['water '], ['water '], ['bread'], ['water '], ['wate
r '], ['chocolate '], ['milk'], ['water '], ['oil '], ['chocolate '], ['oil '], ['chocolate '], ['chocolate '], ['cigarettes'
], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerines '], ['cigarettes'], ['tangerines'], ['cigarettes'], ['water '],
['flakes '], ['tangerines '], ['cigarettes '], ['cigarettes '], ['milk'], ['chocolate '], ['milk'], ['bread'], ['bread'], ['m
ilk'], ['water '], ['water '], ['bread'], ['water '], ['water '], ['chocolate '], ['milk'], ['water '], ['oil '], ['chocolate
'], ['oil '], ['chocolate '], ['chocolate '], ['cigarettes'], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerines '], [
'cigarettes'], ['tangerines'], ['cigarettes'], ['water '], ['flakes '], ['tangerines '], ['cigarettes '], ['cigarettes '], ['
milk'], ['chocolate '], ['milk'], ['bread'], ['bread']]
```

В нашем случае, каждому покупателю соответствует один товар, но, если бы их было несколько, полученный результат стал не пригоден для анализа. Представим наши товары в виде матрицы.

```
te = TransactionEncoder()
te_allp = te.fit(all_product).transform(all_product)
df = pd.DataFrame(te_allp, columns=te.columns_)
print(df)
```

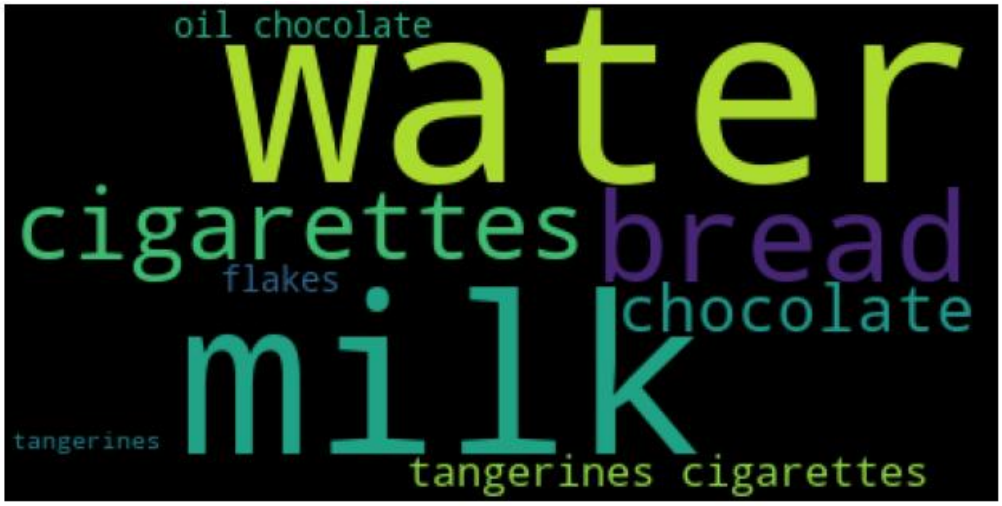
Результат:

```
    bread  chocolate   cigarettes  ...  tangerines  tangerines   water
0   False       False       False  ...       False       False   False
```

Теперь построим облако слов, на основе сырых данных. Объединив все из колонки продуктов данные в один текст.

```
df = pd.read_csv('lab5.csv',encoding='cp1251')
text = " ".join(df['product'])
cloud = WordCloud().generate(text)
plt.imshow(cloud)
plt.axis('off')
plt.show()
```

Результат:



Но если бы мы хотели избавиться от стоп слов, можно было бы воспользоваться библиотекой nltk. По умолчанию для объекта облака слова WordCloud() уже присутствует параметр stopwords, он автоматически убирает стоп слова на английском, но на русском языке необходимо сделать это вручную.

```
stop_words = stopwords.words('russian')
df = pd.read_csv('lab5.csv',encoding='cp1251')
text = " ".join(df['product'])
cloud = WordCloud().generate(text)
cloud = WordCloud(stopwords=stop_words).generate(text)
plt.imshow(cloud)
plt.axis('off')
plt.show()
```

А если бы у нас было много склонений одного и того же слова, было бы не плохо получить леммы, предварительно тонизировав текст.

```
stop_words = stopwords.words('russian')
df = pd.read_csv('lab5.csv',encoding='cp1251')
text = " ".join(df['product'])
text = word_tokenize(text)
lemmatizer = pymorphy2.MorphAnalyzer()
def lemmatize_text(tokens):
    text_new=''
    for word in tokens:
        word = lemmatizer.parse(word)
        text_new = text_new + ' ' + word[0].normal_form
    return text_new
text = lemmatize_text(text)
cloud = WordCloud(stopwords=stop_words).generate(text)
plt.imshow(cloud)
```

```
plt.axis('off')
plt.show()
```

В ходе лабораторной работы мы провели частотный анализ и визуализировали облако слов с использованием серых данных и после обработки, в результате ….

Задание.

1. Загрузить пред обработанные данные, вывести первые 5 строк из датасета;
2. Вывести уникальное количество выбранных колонок;
3. Сформировать новый датасет из нужных столбцов;
4. Построить облако слов на основе серых данных;
5. Почистить стоп слова и найти леммы, построить новое облако слов;
6. Описать полученный результат.

Формат отчета.

Протокол лабораторной работы в формате PDF, который должен содержать, поэтапное выполнение всех задач с текстовым описанием ваших действий и экранными формами, отображающими данные действия.