

```
In [3]: # Имортируем нужные библиотеки
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
matplotlib inline

In [4]: df_11 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L1.XLSX")
df_12 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L2.XLSX")
df_13 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L3.XLSX")
df_14 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L4.XLSX")
df_15 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L5.XLSX")
df_16 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L6.XLSX")
df_17 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L7.XLSX")
df_18 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L8.XLSX")
df_19 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L9.XLSX")
df_110 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L10.XLSX")
df_111 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L11.XLSX")
df_112 = pd.read_excel("C:/Users/sumen/OneDrive/Рабочий стол/Analist/L12.XLSX")
```

```
In [5]: # Объединение данных в один Dataset
df = pd.concat([df_11, df_12, df_13, df_14, df_15, df_16, df_17, df_18, df_19, df_110, df_111, df_112])
df = pd.DataFrame(df)
df.to_excel('all_data.xlsx', index=False)
df
```

| | Код | Наименование | Кол. | Мин. цена | Макс. цена | Сред. цена | Заход | Дата посл. продажи | На складе | В офисе | Приб. | Сумма продаж |
|-----|------|---|------|-----------|------------|------------|-------|--------------------|-----------|---------|---------|--------------|
| 0 | 1166 | Apple AirPods Pro 2 | 2319 | 16610 | 21000 | 18560 | 0 | 28.08.23 18:35 | 0 | 2 | 1510290 | 43042060 |
| 1 | 1100 | Apple AirPods 3 | 1398 | 13280 | 15800 | 14199 | 0 | 28.08.23 18:35 | 0 | 0 | 710070 | 19851030 |
| 2 | 1136 | Apple iPhone 14 Pro Max 256Gb Purple | 1389 | 92000 | 115500 | 104471 | 0 | 26.08.23 12:58 | 0 | 1 | 3442180 | 145110650 |
| 3 | 1088 | Apple AirPods 2 | 1342 | 8800 | 11400 | 9587 | 0 | 28.08.23 18:56 | 0 | 0 | 231240 | 12866540 |
| 4 | 2062 | Apple iPhone 14 Pro 256Gb Purple | 1194 | 89000 | 105500 | 98177 | 0 | 26.08.23 12:58 | 0 | 0 | 4354480 | 117223700 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 4 | 6075 | Samsung Galaxy S21 FE 5G 8/128 Olive | 1 | 38000 | 38000 | 38000 | 0 | 11.08.23 12:57 | 0 | 0 | 1000 | 38000 |
| 5 | 6077 | Samsung Galaxy S21 FE 5G 128 White | 1 | 36000 | 36000 | 36000 | 0 | 08.03.23 15:39 | 0 | 0 | 700 | 36000 |
| 6 | 7397 | Pixel 7 Pro 12/512 Obsidian (Black) | 1 | 90000 | 90000 | 90000 | 0 | 20.06.23 15:15 | 0 | 0 | 4000 | 90000 |
| 7 | 8752 | Apple MacBook Pro 16" (M1 Max 10C CPU, 32C GPU... | 1 | 243000 | 243000 | 243000 | 0 | 07.03.23 13:53 | 0 | 0 | 6030 | 243000 |
| 8 | 9100 | Apple iPad Pro 12.9 M2 128GB WiFi Space Gray | 1 | 83000 | 83000 | 83000 | 0 | 18.03.23 14:04 | 0 | 0 | 550 | 83000 |

403 rows × 12 columns

Какой товар покупали больше всего ?

```
In [6]: #Какой товар покупали больше всего
most_common_product = df.groupby("Наименование")["Кол."].sum().idxmax()
most_common_product
```

Out[6]: 'Apple AirPods Pro 2'

Какая общая прибыль ?

```
In [7]: #Общая прибыль
total_profit = df["Приб."].sum()
total_profit

# Общая прибыль показана за весь период от 01.01.2023 до 28.08.2023
```

Out[7]: 64384267

Какая корреляция между ценами и количеством продаж?

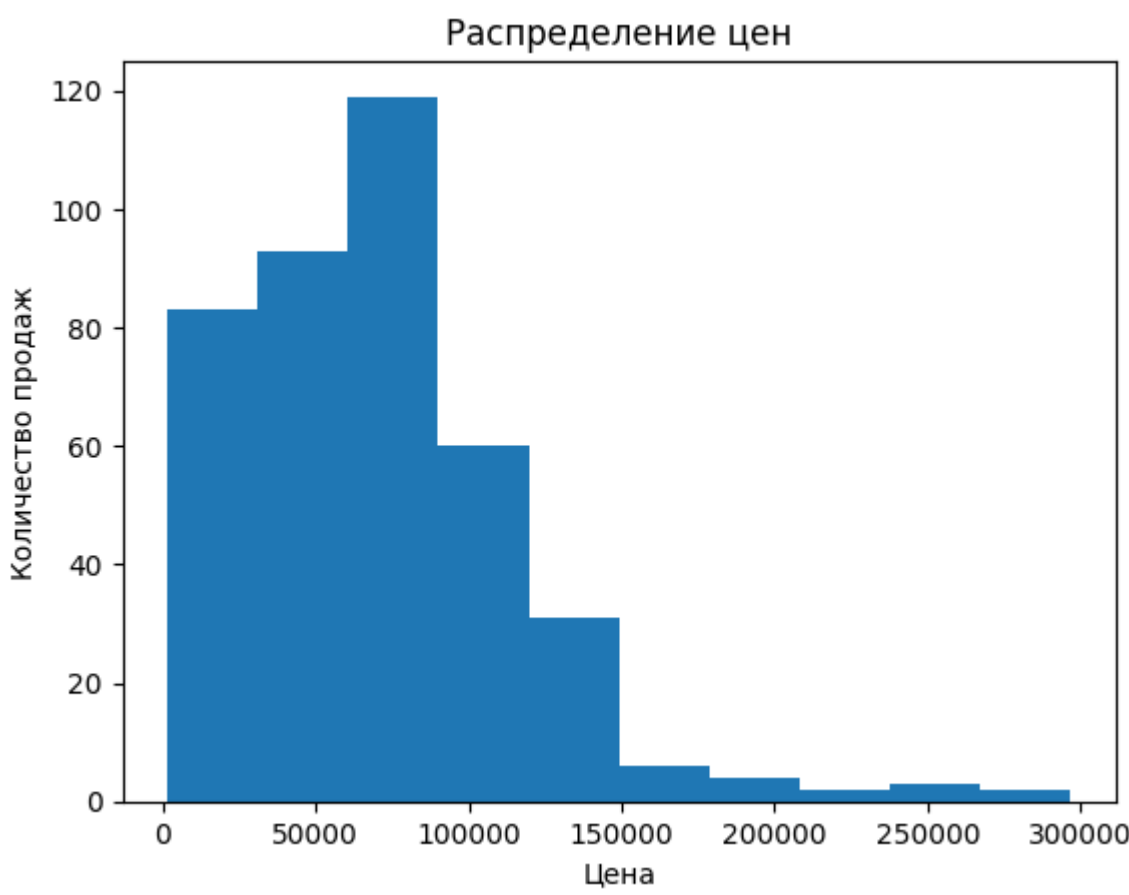
```
In [8]: #Поиск корреляций между ценами и количеством продаж
correlation = df["Кол."].corr(df["Сред. цена"])
correlation

# Значение -0.06726849926070573 указывает на очень слабую отрицательную корреляцию между этими двумя переменными.
# Это означает, что в данном случае цена не имеет значительного влияния на количество продаж.
```

Out[8]: -0.06726849926070573

```
In [12]: #Анализ распределения цен и количества продаж(какой товар продан по высокой цене)

# Гистограмма цен
most_sold_item = df["Наименование"].value_counts().idxmax()
plt.hist(df["Сред. цена"], bins=10)
plt.xlabel('Цена')
plt.ylabel('Количество продаж')
plt.title('Распределение цен')
plt.show()
print(most_sold_item)
```



Apple AirPods Pro 2

```
In [13]: #Исследование влияния наличия товара на складе и в офисе на продажи
grouped = df.groupby(['На складе', 'В офисе'])
mean_sales = grouped["Кол."].mean()
print(mean_sales)
```

| На складе | В офисе |
|-----------|------------|
| 0 | 73.205479 |
| 1 | 159.000000 |
| 2 | 596.750000 |
| 4 | 6.000000 |
| 5 | 116.000000 |
| 7 | 22.000000 |
| 9 | 46.000000 |
| 11 | 133.000000 |
| 17 | 69.000000 |
| 19 | 8.000000 |
| 21 | 10.000000 |
| 25 | 513.000000 |
| 35 | 34.000000 |
| 44 | 6.000000 |
| 52 | 36.000000 |
| 75 | 65.000000 |

Name: Кол., dtype: float64

Когда товара нет ни на складе, ни в офисе, среднее количество продаж составляет около 73 единиц. Однако, когда товар есть как на складе, так и в офисе, среднее количество продаж возрастает до 159 единиц. Это может указывать на то, что наличие товара в физической близости к покупателям способствует увеличению продаж.

```
In [2]: df["Дата посл. продажи"] = pd.to_datetime(df["Дата посл. продажи"])
df["Год"] = df["Дата посл. продажи"].dt.year
df["Месяц"] = df["Дата посл. продажи"].dt.month

monthly_sales = df.groupby(["Год", "Месяц"]).size().reset_index(name="Количество продаж")
plt.plot(monthly_sales["Месяц"], monthly_sales["Количество продаж"])
plt.xlabel("Месяц")
plt.ylabel("Количество продаж")
plt.title("Продажи по месяцам")
plt.show()
```

```
-----
NameError                                Traceback (most recent call last)
Cell In[2], line 1
----> 1 df["Дата посл. продажи"] = pd.to_datetime(df["Дата посл. продажи"])
      2 df["Год"] = df["Дата посл. продажи"].dt.year
      3 df["Месяц"] = df["Дата посл. продажи"].dt.month

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

```
In [16]: df["Дата посл. продажи"] = pd.to_datetime(df["Дата посл. продажи"])
df["Дата посл. продажи"] = df["Дата посл. продажи"].dt.date
df["Прибыль"] = df["Макс. цена"] - df["Мин. цена"]

profit_by_date = df.groupby("Дата посл. продажи")["Прибыль"].sum().reset_index()
plt.plot(profit_by_date["Дата посл. продажи"], profit_by_date["Прибыль"])
plt.xlabel("Дата посл. продажи")
plt.ylabel("Прибыль")
plt.title("Прибыль по дате после продажи")
plt.show()
```



```
In [17]: # Подсчет количества товаров на складе и в офисе
stock_count = df["На складе"].sum()
office_count = df["В офисе"].sum()

# Построение графика наличия товара
labels = ["На складе", "В офисе"]
counts = [stock_count, office_count]
plt.bar(labels, counts)
plt.xlabel('Место')
plt.ylabel('Количество')
plt.title('Наличие товара')
plt.show()

# Большинство товара хранилось в офисе
```



Продажи имеют сезонность или тренд в зависимости от месяца. График показывает количество продаж по месяцам, что позволяет наблюдать изменения в объеме продаж в течение года.

```
In [18]: top_products = df.nlargest(5, 'Кол.') # выбираем 5 самых продаваемых товаров
top_products_names = top_products["Наименование"].tolist() # получаем список названий товаров
top_products_sales = top_products["Кол."].tolist() # получаем список количества продаж
top_products_share = top_products_sales / df["Кол."].sum() * 100 # вычисляем долю от общих продаж
for name, sales, share in zip(top_products_names, top_products_sales, top_products_share):
    print(f"Товар: {name}, Количество продаж: {sales}, Доля от общих продаж: {share:.2f}%")
```

Товар: Apple AirPods Pro 2, Количество продаж: 2319, Доля от общих продаж: 6.85%
Товар: Apple AirPods 3, Количество продаж: 1398, Доля от общих продаж: 4.13%
Товар: Apple iPhone 14 Pro Max 256Gb Purple, Количество продаж: 1389, Доля от общих продаж: 4.10%
Товар: Apple AirPods 2, Количество продаж: 1342, Доля от общих продаж: 3.96%
Товар: Apple iPhone 14 Pro 256Gb Purple, Количество продаж: 1194, Доля от общих продаж: 3.53%

```
In [19]: mean_sales = df["Кол."].mean() # среднее количество продаж
min_sales = df["Кол."].min() # минимальное количество продаж
max_sales = df["Кол."].max() # максимальное количество продаж
print(f"Среднее количество продаж: {mean_sales}")
print(f"Минимальное количество продаж: {min_sales}")
print(f"Максимальное количество продаж: {max_sales}")
```

Среднее количество продаж: 84.02977667493796
Минимальное количество продаж: 1
Максимальное количество продаж: 2319

При разборе данного датасета, я получил ответы на интересующие меня информацию