

E-commerce Sales Analysis

Purpose of the project:

Оптимизация процессов принятия решений и стратегического планирования на основе детального анализа данных о продажах, выявление ключевых трендов, моделей покупок и оценка эффективности маркетинговых кампаний.

Main tasks of the project:

- Сбор и интеграция данных из различных источников.(csv-file in my case)
- Анализ данных с использованием SQL для создания отчетов и выявления инсайтов.
- Визуализация данных и создание интерактивных дашбордов в Power BI.
- Разработка рекомендаций по улучшению продаж и маркетинговых стратегий.

1. Data Preparation

Ввод [7]:

```
import pandas as pd

data = pd.read_csv('Sales Data Analysis (1).csv')

data.head()
```

Out[7]:

	Column1	Order ID	Product Category	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sal
0	0	295665	Laptops and Computers	Macbook Pro Laptop	1	1700.00	30-12-2019 00:01	136 Church St, New York City, NY 10001	12	1700.
1	1	295666	Home Appliances	LG Washing Machine	1	600.00	29-12-2019 07:03	562 2nd St, New York City, NY 10001	12	600.
2	2	295667	Charging Cables	USB-C Charging Cable	1	11.95	12-12-2019 18:21	277 Main St, New York City, NY 10001	12	11.
3	3	295668	Monitors	27in FHD Monitor	1	149.99	22-12-2019 15:13	410 6th St, San Francisco, CA 94016	12	149.
4	4	295669	Charging Cables	USB-C Charging Cable	1	11.95	18-12-2019 12:38	43 Hill St, Atlanta, GA 30301	12	11.

Исследование структуры данных и проверка на наличие аномалий

1. Структура данных

```
Ввод [27]: total_records = data['Column1'].count()
print(f'1. Total amount of records(Number of orders): {total_records}')
col_num = data.shape[0]
print(f'2. Number of columns: {col_num}')
```

```
1. Total amount of records: 185950
2. Number of columns: 185950
```

2. Проверка на наличие аномалий

- **Пропущенные значения:** Отсутствуют пропущенные значения во всех столбцах.
- **Дата и время заказа:** Преобразовано в формат datetime для дальнейшего анализа.(дальнейшем в PostgreSQL)

```
Ввод [13]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 185950 entries, 0 to 185949
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   Column1                185950 non-null int64  
1   Order ID               185950 non-null int64  
2   Product Category      185950 non-null object 
3   Product                185950 non-null object 
4   Quantity Ordered      185950 non-null int64  
5   Price Each             185950 non-null float64 
6   Order Date             185950 non-null object 
7   Purchase Address       185950 non-null object 
8   Month                  185950 non-null int64  
9   Sales                  185950 non-null float64 
10  City                   185950 non-null object 
11  Hour                   185950 non-null int64  
12  Time of Day            185950 non-null object 
dtypes: float64(2), int64(5), object(6)
memory usage: 18.4+ MB
```

```
Ввод [51]: df = {
    'Column_name': ['Column1', 'Order ID', 'Product Category', 'Product',
                    'Quantity Ordered', 'Price Each', 'Order Date', 'Purchase Address',
                    'Month', 'Sales', 'City', 'Hour', 'Time of Day'],
    'Description': ['Идентификатор строки', 'Уникальный идентификатор заказа',
                    'Категория товара', 'Название товара', 'Количество заказа',
                    'Цена за единицу товара', 'Дата и время заказа', 'Адрес п',
                    'Месяц заказа', 'Общая сумма продажи', 'Город покупки', 'Время суток']
}

descr = pd.DataFrame(df)

descr
```

Out[51]:

	Column_name	Description
0	Column1	Идентификатор строки
1	Order ID	Уникальный идентификатор заказа
2	Product Category	Категория товара
3	Product	Название товара
4	Quantity Ordered	Количество заказанных единиц
5	Price Each	Цена за единицу товара
6	Order Date	Дата и время заказа
7	Purchase Address	Адрес покупки
8	Month	Месяц заказа
9	Sales	Общая сумма продажи
10	City	Город покупки
11	Hour	Час заказа
12	Time of Day	Время суток

3. Описание основных метрик и показателей

```
Ввод [49]: min_quan = data['Quantity Ordered'].min()
max_quan = data['Quantity Ordered'].max()
avg_quan = data['Quantity Ordered'].mean()
min_price = data['Price Each'].min()
max_price = data['Price Each'].max()
avg_price = data['Price Each'].mean()
min_sales = data['Sales'].min()
max_sales = data['Sales'].max()
avg_sales = data['Sales'].mean()

print(f'Quantity Ordered: Минимальное значение - {min_quan}, максимальное - {max_quan}, среднее - {avg_quan}')
print(f'Price Each: Минимальное значение - ${min_price}, максимальное - ${max_price}, среднее - ${avg_price}')
print(f'Sales: Минимальное значение - ${min_sales}, максимальное - ${max_sales}, среднее - ${avg_sales}')
```

Quantity Ordered: Минимальное значение - 1, максимальное - 9, среднее - 1.1243828986286637.

Price Each: Минимальное значение - \$2.99, максимальное - \$1700.0, среднее - \$184.3997347674106.

Sales: Минимальное значение - \$2.99, максимальное - \$3400.0, среднее - \$185.490916751815.

Основные метрики для анализа

- **Общие продажи:** Анализ суммарных продаж по различным параметрам.
- **Средний чек:** Анализ среднего значения заказа по городам и категориям товаров.
- **Количество заказов:** Анализ количества заказов по различным временным интервалам и категориям товаров.
- **Топ продукты:** Определение самых продаваемых товаров.

2. Data Analysis

1. Анализ суммарных продаж и количества проданных единиц по различным параметрам

```
SELECT product, SUM("Quantity Ordered") AS total_sold, SUM(sales) AS total_profit
FROM public.orders
GROUP BY product
ORDER BY total_sold DESC
```

	product character varying	total_sold bigint	total_profit numeric
1	AAA Batteries (4-pack)	31017	92740.83
2	AA Batteries (4-pack)	27635	106118.40
3	USB-C Charging Cable	23975	286501.25
4	Lightning Charging Cable	23217	347094.15
5	Wired Headphones	20557	246478.43
6	Apple Airpods Headphones	15661	2349150
7	Bose SoundSport Headphones	13457	1345565.43
8	27in FHD Monitor	7550	1132424.50
9	iPhone	6849	4794300
10	27in 4K Gaming Monitor	6244	2435097.56
11	34in Ultrawide Monitor	6199	2355558.01
12	Google Phone	5532	3319200
13	Flatcreen TV	4819	1445700

По категориям товара

```
SELECT "Product Category", SUM("Quantity Ordered") AS total_sold, SUM(sales) AS total_profit
FROM public.orders
GROUP BY "Product Category"
ORDER BY total_sold DESC
```

	Product Category character varying	total_sold bigint	total_profit numeric
1	Batterie	58652	198859.23
2	Audio Devices	49675	3941193.86
3	Charging Cables	47192	633595.40
4	Monitors	24122	6377228.78
5	Phones and Accessories	14449	8940700
6	Laptops and Computers	8858	12167558.70
7	Entertainment Devices	4819	1445700
8	Home Appliances	1312	787200

По городам

```
SELECT city, SUM("Quantity Ordered") AS total_sold, SUM(sales) AS total_profit
FROM public.orders
GROUP BY city
ORDER BY total_sold DESC
```

	city character varying	total_sold bigint	total_profit numeric
1	San Francisco	50239	8262203.91
2	Los Angeles	33289	5452570.80
3	New York City	27932	4664317.43
4	Boston	22528	3661642.01
5	Dallas	16730	2767975.40
6	Atlanta	16602	2795498.58
7	Seattle	16553	2747755.48
8	Portland	14053	2320490.61
9	Austin	11153	1819581.75

По времени суток

```
SELECT "Time of Day", SUM("Quantity Ordered") AS total_sold
FROM public.orders
GROUP BY "Time of Day"
ORDER BY total_sold DESC
```

	Time of Day character varying	total_sold bigint
1	Afternoon	75531
2	Evening	71248
3	Morning	50497
4	Night	11803


По годам

```
SELECT EXTRACT(YEAR FROM("Order Date")), SUM("Quantity Ordered") AS total_sold
FROM public.orders
GROUP BY EXTRACT(YEAR FROM("Order Date"))
ORDER BY total_sold DESC
```



	extract numeric	total_sold bigint
1	2019	209038
2	2020	41

2. Анализ среднего значения заказа по городам и категориям товаров.

```
SELECT city, AVG(Sales) AS average_order_value
FROM public.orders
GROUP BY city
ORDER BY average_order_value DESC
```

	city character varying 	average_order_value numeric 
1	Atlanta	187.8569034339090115
2	New York City	187.5027106447981991
3	Dallas	186.7729689608636977
4	Seattle	186.5161200108607114
5	Portland	186.1604981949458484
6	San Francisco	184.7045495394795672
7	Los Angeles	184.1773619321060632
8	Austin	183.7033568904593640
9	Boston	183.6882717969298686



```
SELECT "Product Category", AVG(Sales) AS average_order_value
FROM public.orders
GROUP BY "Product Category"
ORDER BY average_order_value DESC
```

	Product Category character varying 	average_order_value numeric 
1	Laptops and Computers	1374.5547559873474921
2	Phones and Accessories	619.5052660753880266
3	Home Appliances	600.0000000000000000
4	Entertainment Devices	301.1875000000000000
5	Monitors	265.5076722594612598
6	Audio Devices	82.5277213334450121
7	Charging Cables	14.5450150363857579
8	Batterie	4.8245725168615653

3.Топ продукты: Определение самых продаваемых

товаров

```
SELECT Product, SUM(Sales) AS TotalSales
FROM public.orders
GROUP BY Product
ORDER BY TotalSales DESC
LIMIT 10;
```

	product character varying 	totalsales numeric 
1	Macbook Pro Laptop	8037600
2	iPhone	4794300
3	ThinkPad Laptop	4129958.70
4	Google Phone	3319200
5	27in 4K Gaming Monitor	2435097.56
6	34in Ultrawide Monitor	2355558.01
7	Apple Airpods Headphones	2349150
8	Flatscreen TV	1445700
9	Bose SoundSport Headphones	1345565.43
10	27in FHD Monitor	1132424.50

Топ 10 продуктов по количеству проданных единиц:

```
SELECT Product, COUNT("Order ID") AS OrderCount
FROM public.orders
GROUP BY Product
ORDER BY OrderCount DESC
LIMIT 10;
```


	<div>product</div> <div>character varying</div>	<div>ordercount</div> <div>bigint</div>
1	USB-C Charging Cable	21903

4.Анализ продаж по датам(2019/01 - 2020/01)

```
SELECT TO_CHAR("Order Date", 'YYYY-MM') AS month_, SUM("Quantity Ordered") AS total_sold
FROM public.orders
GROUP BY TO_CHAR("Order Date", 'YYYY-MM')
ORDER BY TO_CHAR("Order Date", 'YYYY-MM');
```

	<div>year_month</div> <div>text</div>	<div>total_sold</div> <div>bigint</div>
1	2019-01	10862
2	2019-02	13449
3	2019-03	17005
4	2019-04	20558
5	2019-05	18667
6	2019-06	15253
7	2019-07	16072
8	2019-08	13448
9	2019-09	13109
10	2019-10	22703
11	2019-11	19798
12	2019-12	28114
13	2020-01	41

Анализ продаж по городам

```
SELECT DISTINCT city, COUNT("Order ID") AS purchase_count
FROM public.orders
GROUP BY city
ORDER BY COUNT("Order ID") DESC
```

	city character varying	purchase_count bigint
1	San Francisco	44732
2	Los Angeles	29605
3	New York City	24876
4	Boston	19934
5	Atlanta	14881

5.Анализ поведения клиентов

(первые 10)

```
SELECT "Purchase Address", COUNT("Order ID") AS Order_Count, SUM("Price Each" * "Quantity Ordered")
FROM public.orders
GROUP BY "Purchase Address"
ORDER BY Order_Count DESC
LIMIT 10;
```

	Purchase Address character varying	order_count bigint	total_price numeric
1	193 Forest St, San Francisco, CA 94016	9	1657.67
2	279 Sunset St, San Francisco, CA 94016	8	1827.82
3	223 Elm St, Los Angeles, CA 90001	8	1402.91
4	640 West St, San Francisco, CA 94016	7	1138.88
5	284 Walnut St, San Francisco, CA 94016	7	629.45
6	176 North St, San Francisco, CA 94016	7	1619.63
7	662 Chestnut St, San Francisco, CA 94016	7	385.86
8	550 Cherry St, San Francisco, CA 94016	7	1224.60
9	727 9th St, San Francisco, CA 94016	7	1042.72
10	292 Washington St, San Francisco, CA 94016	7	2186.91

6.Средний чек от общего количества продаж и общего количества покупателей

```
SELECT SUM("Price Each" * "Quantity Ordered") / COUNT(*) AS overall_average_ticket
FROM public.orders;
```

	overall_average_ticket numeric
1	185.4909167518150040

Ввод []:

