# Материалы

Презентация

# E-commerce - Выявление профилей потребления

Интернет-магазин товаров для дома «Пока все ещё тут». Девиз интернет-магазина «Пока все ещё тут» - мы создаем уют! Требуется расчет метрик и создание гипотез на основе полученных данных.
Ранее расчет метрик не производился. Данные могут содержать дубликаты. Ориентир интернет-магазина больше на сегмент покупателей B2C.

**Цель проекта:**
Сегментация покупателей по профилю потребления, чтобы осуществлять более персонализированные рекламные рассылки.

**Задачи проекта:** Необходимо сегментировать покупателей по профилю потребления.

Для этого требуется:

- Провести исследовательский анализ данных
- Сегментировать покупателей по профилю потребления на основе истории их покупок
- Сформулировать и проверить статистические гипотезы

# План выполнения проекта

**Загрузка данных и изучение общей информации**

**Импорт библиотек**

**Загрузка данных**

**Общая информация о датасете**

Изучение общей информации о датасете:

- Сколько строк и столбцов в датафрейме?
- Значения какого типа хранят столбцы?
- За какой период предоставлены данные? Найти максимальную и минимальные даты.

**Сделать выводы на основе изучения общей информации о датасете**

**Предобработка данных.**

Цель шага - подготовить данные к дальнейшему анализу.

**Типы данных**

Проверить корректность типов данных колонок. Привести типы к более подходящим содержимому столбцов. Проверить названия колонок датасета на соответствие стилю написания и содержимому. Переименовать при необходимости.

**Пропуски и дубликаты**

Проверить датасет на наличие пропусков и дубликатов (явных, неявных). Удалить дубликаты, заполнить пропуски если это необходимо и возможно. Обратить внимание, что в данных могут быть не полные дубликаты (возможны ошибки выгрузки/повторные загрузки данных).

**Подвести итоги предобработки**

**Исследовательский анализ данных**

**Сколько покупателей? Топ-10 заказов**

**Анализ данных**

**Данными за какой период мы располагаем?**

Найдем максимальную и минимальную дату.

**Сформулировать цель сегментации:**

- кто будет использовать результаты сегментации;
- для чего они будут использоваться.

**Выбор метода сегментации**

Выбрана методология проведения сегментации - RFM-анализ. Материалы по теме и еще.
Разобъем покупателей по трем показателям, которые получим из данных:

- по давности покупок (Recency). Необходимо определить критерии давности.
- частоте покупок (Frequency). Необходимо определить диапазоны частоты (количества) покупок.
- сумме чека (Monetary). Необходимо определить диапазоны сумм покупок.

Укрупнить получившиеся сегменты покупателей.

**Проверка сегментации покупателей**

Гипотезами проверяем корректность разбивки покупателей на категории. Необходимое минимальное количество гипотез к проверке - две.

- H0 - Нет различий между категориями покупателей в среднем чеке.

- H1 - Есть различия между категориями покупателей в среднем чеке.
- H0 - Нет различий между категориями покупателей по частоте покупок.
- H1 - Есть различия между категориями покупателей по частоте покупок.

Если хоть одна проверка покажет статистически значимую разницу между кластерами, значит кластеризация мы проведена корректно. Если две проверки не обнаружат статистически значимой разницы между двумя одинаковыми кластерами - стоит пересмотреть разбивку на кластеры.

**Разделение товаров на категории**

Для проведения анализа необходимо выделить товарные категории. Так как присутствует большое количество имен собственных, то разбивку на категории лучше произвести вручную. 5-7 категорий для текущих задач - достаточно.

**Сезонность категорий товаров для каждого сегмента покупателей**

**Добавить столбцы**

Добавление дополнительных столбцов для дальнейшего анализа: выделить из даты день недели и месяц заказа.
Сезонность выбрать исходя из имеющегося временного периода данных (месяц или сезон). Обосновать выбор.

- Есть ли сезонность в продаже товаров?

**День недели покупок для каждого сегмента покупателей**

Распределить категории покупателей по дням недели.

- Есть ли взаимосвязь между днем недели совершения покупок и количеством покупок?

**Популярные категории товаров для каждого сегмента покупателей**

Определить категории товаров - лидеров продаж по числу покупок для каждого сегмента покупателей.

**Средний чек для каждого сегмента покупателей**

# ВЫВОД

Рекомендации и презентация

# 1 Загрузка данных и изучение общей информации

## 1.1 Импорт библиотек

In [1]:

```
import pandas as pd
import numpy as np
from datetime import datetime
from datetime import timedelta
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import rcParams, rcParamsDefault
import seaborn as sns

import plotly.express as px
import plotly.graph_objects as go
#снимаем ограничение на ширину столбцов
pd.set_option('display.max_colwidth', None)
# игнорируем предупреждения
pd.set_option('chained_assignment', None)

import calendar as cl
# комбинации
import itertools

# Обязательно для приемлемого отображения графиков plt
rcParams['figure.figsize'] = 10, 6
%config InlineBackend.figure_format = 'svg'
# доп. для декорирования графиков
factor = .8
default_dpi = rcParamsDefault['figure.dpi']
```

```
rcParams['figure.dpi'] = default_dpi * factor

import scipy.stats as stats
from scipy import stats
from scipy import spatial
from scipy.stats import norm
```

## 1.2 Загрузка данных

Датасет содержит информацию о транзакциях интернет-магазина товаров для дома и быта "Пока все ещё тут".

Загрузим датасет `ecom_dataset_upd.csv` и сохраним его в переменную `df`.

In [2]:

```
df = pd.read_csv('https://code.s3.yandex.net/datasets/ecom_dataset_upd.csv', sep=',')
```

## 1.3 Общая информация о датасете

Изучим общую информацию о датасете. Для начала убедимся, что данные загружены, а также посмотрим что именно находится в датасете и на несколько случайных строк в `df`.

In [3]:

```
display(df.sample(n=5, random_state=4))
```

| | date | customer_id | order_id | product | quantity | price |
|---|---|---|---|---|---|---|
| **682** | 2018110707 | 90c27736-32fe-4afd-ba52-80a1b9f24b5a | 14505 | Пеларгония зональная диам. 12 см сиреневый полумахровый | 1 | 188.0 |
| **677** | 2018110620 | 1ac82730-7e1b-4581-9b99-59a9f513b0c9 | 68860 | Муляж Зеленое яблоко полиуретан d-6 см | 3 | 22.0 |
| **6548** | 2019100914 | ed0ff3ae-e963-4eef-a969-013bfe62d711 | 72904 | Сушилка для белья потолочная Лиана 2,0 м 1703009 | 1 | 599.0 |
| **7428** | 2020012409 | 70b30da1-c604-4726-849b-223b3775d382 | 110302 | Нолина в цветной керамике d-7 см | 1 | 239.0 |
| **2025** | 2019022520 | 3b2e7ead-3582-43bc-807e-4486f6511c47 | 70456 | Штора для ванной комнаты Quadretto 240x200 см белая, Bacchetta, 4062 | 1 | 1199.0 |

Данные загружены успешно.

Изучим детальнее информацию:

- Сколько строк и столбцов в датафрейме?
- Значения какого типа хранят столбцы?

In [4]:

```
# посчитаем строки и колонки
rows = len(df.axes[0])
cols = len(df.axes[1])
print(f"Датасет содержит {str(rows)} строк и {str(cols)} колонок\n\
о {len(df['order_id'].unique())} уникальных заказах.\
\nОбщее количество ненулевых значений в столбцах и\n\
типы данных каждого столбца выведем методом df.info():\n")
print(df.info())
```

Датасет содержит 7474 строк и 6 колонок
о 3521 уникальных заказах.
Общее количество ненулевых значений в столбцах и
типы данных каждого столбца выведем методом df.info():

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7474 entries, 0 to 7473
Data columns (total 6 columns):
 # Column Non-Null Count Dtype

```
---  ------       --------------  -----
 0   date         7474 non-null   int64
 1   customer_id  7474 non-null   object
 2   order_id     7474 non-null   int64
 3   product      7474 non-null   object
 4   quantity     7474 non-null   int64
 5   price        7474 non-null   float64
dtypes: float64(1), int64(3), object(2)
memory usage: 350.5+ KB
None
```

#### 1.3.1 Описание колонок

Датасет `df` содержит 6 столбцов:

- `date` — дата заказа;
- `customer_id` — идентификатор покупателя;
- `order_id` — идентификатор заказа;
- `product` — наименование товара;
- `quantity` — количество товара в заказе;
- `price` — цена товара.

В датасете 7474 строк, пропусков нет ни в одном столбце, названия столбцов даны в соответствии со стилем написания и отражают содержимое.

Столбец `date` с данными о дате заказа нужно перевести в тип данных datetime. В остальных столбцах типы данных определены корректно.

### 1.4 Итоги раздела

Данные датасета загружены: 7474 строк, 6 колонок. В столбце `date` тип данных не соответствует содержимому - временному, требуется изменить тип данных.
Проведем предобработку данных.

# 2 Предобработка данных.

## 2.1 Типы данных

При обзоре данных выяснили, что в столбце `date` тип данных не соответствует содержимому - временному.
Заменим тип данных в столбце `date` и убедимся, что тип данных после замены определяется как `datetime`.

In [5]:

```
# замена типа данных
df['date'] = pd.to_datetime(df['date'], format='%Y%m%d%H')
# для вывода на экран
df_print = df['date']
#вывод на экран
print(f"После преобразования тип данных:\nв колонке 'date' - {df_print.dtypes}.")
```

```
После преобразования тип данных:
в колонке 'date' - datetime64[ns].
```

## 2.2 Пропуски и дубликаты

Проверим датасет на наличие пропусков и дубликатов (явных, неявных). Удалим дубликаты, заполним пропуски если это необходимо и возможно. Обратим внимание, что в данных могут быть не полные дубликаты (возможны ошибки выгрузки/повторные загрузки данных).

#### 2.2.1 Обработка пропусков

При обзоре данных было выявлено, что в столбцах пропуски не содержатся. Проверим еще раз.

In [6]:

```
print(f"Всего датасет содержит {df.isnull().values.sum()} пропусков.")
```

Всего датасет содержит 0 пропусков.

## 2.3 Поиск дубликатов в данных

In [7]:

```
print(f'Количество явных дубликатов в датафрейме "df": {df.duplicated().sum()} шт.')
```

Количество явных дубликатов в датафрейме "df": 0 шт.

Явных дубликатов нет. Проверим датафрейм на "неявные" дубликаты. Поскольку в данных часто встречаются разного рода ошибки, полученные, например, при сборе из разных БД, использовании внешних данных. Поэтому следует сделать более тщательную проверку.

Сравним данные,используя дополнительный параметр subset() по столбцам `order_id`, `customer_id`, `product`. Дату заказа не будем учитывать.

In [8]:

```
#посчитаем неявные дубликаты без учета даты
df[df.duplicated(subset=['order_id', 'customer_id', 'product'])].count()
```

Out[8]:

```
date           1886
customer_id    1886
order_id       1886
product        1886
quantity       1886
price          1886
dtype: int64
```

Очень большое количество неявных дубликатов. Значение order_id в таком датасете по идее должно быть уникальным. Предположим, что уникальность order_id распространяется только на один год. Проверим это предположение на срезе данных за 2019 год. Если это неявных дубликатов не станет меньше, то предположение не обосновано.
Сделаем срез данных и посмотрим количество неявных дубликатов заказов в 2019 году.

In [9]:

```
# срез данных за 2019 год
df_2019 = df.query('"2018.12.31" < date < "2020.01.01"')
df_2019.head(2)
```

Out[9]:

| | date | customer_id | order_id | product | quantity | price |
|---|---|---|---|---|---|---|
| **1422** | 2019-01-01 10:00:00 | e382d4c4-a4c9-44d3-97a0-a8868e122563 | 69531 | Сумка-тележка хозяйственная Rolser BAB010 rojo JOY-1800 красная | 1 | 4139.0 |
| **1423** | 2019-01-01 14:00:00 | 0bbff16a-75df-4947-a5ef-f577c031a19d | 69689 | Вешалка металлическая Valiant с четырьмя разъемными перекладинами противоскользящим покрытием 35*30 см 18В30 | 7 | 135.0 |

In [10]:

```
# неявные дубликаты в 2019 году
df_2019[df_2019.duplicated(subset=['order_id', 'customer_id', 'product'])].count()
```

Out[10]:

```
date           1687
customer_id    1687
order_id       1687
product        1687
quantity       1687
price          1687
```

dtype: int64

In [11]:

```
ratio = round((df_2019[df_2019.duplicated(subset=['order_id', 'customer_id',
'product'])].count()\
               / df[df.duplicated(subset=['order_id', 'customer_id',
'product'])].count() -1) *100, 1)
print(f"Количество неявных дубликатов в 2019 г снизилось в % по отношению к общему
числу на\n{ratio}")
```

Количество неявных дубликатов в 2019 г снизилось в % по отношению к общему числу на
date -10.6
customer_id -10.6
order_id -10.6
product -10.6
quantity -10.6
price -10.6
dtype: float64

Неявных дубликатов не стало сильно меньше - 11% обосновано тем, что из датасета исключили период в 3 месяца 2018 года и 1 месяц 2020 года. Предположение, что неявные дубликаты возникли из-за того, что уникальность order_id распространяется только на один год не обосновано.
Посмотрим на дубликаты, отсортированные по номеру заказа и дате, выведем 30 первых строк датасета с фильтром на дубликаты.

In [12]:

```
df[df.duplicated(subset=['order_id', 'customer_id', 'product'])]\
.sort_values(by=['order_id', 'date']).head(30)
```

Out[12]:

| | date | customer_id | order_id | product | quantity | price |
|---|---|---|---|---|---|---|
| **376** | 2018-10-23 13:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Лаванды в кассете по 6 шт | 1 | 315.0 |
| **377** | 2018-10-23 13:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Розмарина в кассете по 6 шт | 1 | 207.0 |
| **378** | 2018-10-23 13:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Тимьяна в кассете по 6 шт | 1 | 162.0 |
| **509** | 2018-10-28 19:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Лаванды в кассете по 6 шт | 1 | 315.0 |
| **510** | 2018-10-28 19:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Розмарина в кассете по 6 шт | 1 | 207.0 |
| **511** | 2018-10-28 19:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Тимьяна в кассете по 6 шт | 1 | 162.0 |
| **588** | 2018-11-02 14:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Лаванды в кассете по 6 шт | 1 | 315.0 |
| **589** | 2018-11-02 14:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Розмарина в кассете по 6 шт | 1 | 207.0 |
| **590** | 2018-11-02 14:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Тимьяна в кассете по 6 шт | 1 | 162.0 |
| **758** | 2018-11-10 17:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Лаванды в кассете по 6 шт | 1 | 315.0 |
| **759** | 2018-11-10 17:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Розмарина в кассете по 6 шт | 1 | 207.0 |
| **760** | 2018-11-10 17:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Тимьяна в кассете по 6 шт | 1 | 162.0 |
| **816** | 2018-11-15 15:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Лаванды в кассете по 6 шт | 1 | 315.0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **817** | 2018-11-15 15:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Розмарина в кассете по 6 шт | 1 | 207.0 |
| **818** | 2018-11-15 15:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Тимьяна в кассете по 6 шт | 1 | 162.0 |
| **827** | 2018-11-16 16:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Лаванды в кассете по 6 шт | 1 | 315.0 |
| **828** | 2018-11-16 16:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Розмарина в кассете по 6 шт | 1 | 207.0 |
| **829** | 2018-11-16 16:00:00 | b80e4826-7218-4bf9-ac08-eb2c81ab3f62 | 13547 | Рассада зелени для кухни Тимьяна в кассете по 6 шт | 1 | 162.0 |
| **491** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Многолетнее растение Душица-орегано розовый объем 0,5 л | 1 | 89.0 |
| **492** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Многолетнее растение Тимьян-чабрец розовый объем 0,5 л | 1 | 89.0 |
| **493** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Пеларгония зональная диам. 12 см белая полумахровая | 1 | 188.0 |
| **494** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Пеларгония зональная диам. 12 см розовая с малиновым полумахровая | 1 | 188.0 |
| **495** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Пеларгония зональная диам. 12 см сиреневый полумахровый | 1 | 188.0 |
| **496** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Пеларгония зональная диам. 12 см ярко красная махровая | 1 | 188.0 |
| **497** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Рассада зелени для кухни Базилик Арарат, кассета по 6шт | 1 | 169.0 |
| **498** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Рассада зелени для кухни Лаванды в горшке диам. 9 см | 1 | 101.0 |
| **499** | 2018-10-28 09:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Рассада зелени для кухни Розмарина в кассете по 6 шт | 1 | 210.0 |
| **543** | 2018-10-31 06:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Многолетнее растение Душица-орегано розовый объем 0,5 л | 1 | 89.0 |
| **544** | 2018-10-31 06:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Многолетнее растение Тимьян-чабрец розовый объем 0,5 л | 1 | 89.0 |
| **545** | 2018-10-31 06:00:00 | 3ee43256-af7d-4036-90d4-eeefa1afc767 | 14500 | Пеларгония зональная диам. 12 см белая полумахровая | 1 | 188.0 |

Вывели 30 первых строк дубликатов. По заказу 13547 видно, что полностью дублируется заказ, за исключением столбца `date`. Задублированы данные отличаются только часом, днем или месяцем, аналогичная картина по заказу 14500.

Можно предположить, что такие дубликаты могли возникнуть из-за сбоя при загрузке по колонке даты в заказе или данные могли быть загружены повторно много раз. Также такие дубли могли возникнуть из-за того, что в датасет попали данные не только по завершенным заказом, но и по промежуточным стадиям заказа. Например "собран", "в пути". Что также можно считать ошибкой выгрузки.

Посчитаем сколько попало уникальных заказов в датасете с дубликатами, используя дополнительный параметр subset() по столбцам `order_id`, `customer_id`, `product`.

In [13]:

```
df_dubl = df[df.duplicated(subset=['order_id', 'customer_id', 'product'])]
print(f"По трем столбцам задублировано {len(df_dubl['order_id'].unique())} заказов, это \
{round(len(df_dubl['order_id'].unique()) / len(df['order_id'].unique()) *100, 1)} % от всех заказов датасета.")
```

По трем столбцам задублировано 257 заказов, это 7.3 % от всех заказов датасета.

7,3% заказов в датасете это дубликаты по столбцам `order_id`, `customer_id`, `product`. Оставлять такие данные - значит исказить исследование. Для исследования останется достаточно данных 92,7% оставшихся заказов. Поэтому без сожаления избавимся от неявных дубликатов, образованным по трем столбцам и перезапишем наш очищенный датафрейм как `dfc`.

In [14]:

```
#удаляем неявные дубликаты по трем столбцам -перезаписываем в dfc
dfc = df.drop_duplicates(subset=['order_id', 'customer_id', 'product'])
```

Поищем еще дубликаты и нет ли нескольких покупателей на один заказ. Для этого сделаем группировку по заказу и посчитаем у каждого заказа количество уникальных пользователей, отсортируем по убыванию. Если у заказа больше одного уникального покупателя, то значит с заказом что-то не то. Посчитаем количество дублей.

In [15]:

```
df_dubl2 = dfc.groupby('order_id')['customer_id'].nunique().sort_values(ascending = False)


print(f"Найдено {df_dubl2[df_dubl2 > 1].count()} заказов с двумя и более покупателями.")
```

Найдено 29 заказов с двумя и более покупателями.

Мы нашли 29 заказов, у которых несколько покупателей. Так как мы не знаем какой из заказов мы можем сохранить, то удалим все неявные дубликаты.
Для этого сохраним список с заказами в переменную `orders_dubl_list`. Затем пересохраним датафрейм `dfc`, в котором методом query и оператором not in исключим дубликаты, которые будем искать в переменной `orders_dubl_list`.

In [16]:

```
# сохраним order_id дублированных заказов
orders_dubl_list = list(df_dubl2[df_dubl2 > 1].index)
print("Выведем список задублированных заказов:\n", orders_dubl_list)
```

Выведем список задублированных заказов:
[72845, 71480, 69485, 69410, 71226, 69283, 72950, 70631, 69531, 69833, 70946, 70808, 71054, 69345, 72778, 72188, 71542, 68785, 69310, 71571, 72790, 70114, 14872, 71663, 70726, 71461, 70903, 70542, 71648]

In [17]:

```
# сохраним датафрейм без дубликатов - сделаем срез с условием, что датасет не содержит заказы из переменной orders_dubl_list
dfc = dfc.query('order_id not in @orders_dubl_list')
```

Проверим, удалены ли дубликаты, сделаем снова срез по номеру заказа и количеству уникальных покупателей на один заказ. Сосчитаем заказы, в которых более одного уникального покупателя. Если их 0, то значит мы успешно удалили их на предыдущем шаге.

In [18]:

```
df_dubl3 = dfc.groupby('order_id')['customer_id'].nunique().sort_values(ascending = False)
print(f"Дубликатов с одним номером заказа на нескольких покупателей после удаления осталось: \
{df_dubl3[df_dubl3>1].count()} штук.")
```

Дубликатов с одним номером заказа на нескольких покупателей после удаления осталось: 0 штук.

In [19]:

```
print(f"В датасете было  {len(df)} строк, после удаления неявных дубликатов, осталось: {len(dfc)} строк.\n\
Удалено как дубликаты {round((100 -(len(dfc) / len(df) ) *100), 1)} % строк.")
```

В датасете было  7474 строк, после удаления неявных дубликатов, осталось: 5522 строк.
Удалено как дубликаты 26.1 % строк.

### 2.3.1 Итоги поиска дубликатов в данных

- Количество явных дубликатов нет.

- Неявных дубликатов найдено 26% строк по всем столбцам. Возможно при выгрузке данных произошел сбой, в результате которого 7,3 % заказов (257 уникальных заказов) - было задублировано по номеру, покупателю и по товару, а еще у 29 заказов было более чем по одному покупателю. Дубликаты были удалены, так как они исказят исследование. В датасете до удаления было 7474 строк, после удаления неявных дубликатов, осталось: 5522 строк.

## 2.4 Добавление столбцов

### 2.4.1 Выручка по заказу

Добавим столбец `total`, в которую запишем результат расчета выручки по каждой строке заказа. Выручку найдем построчным перемножением цены за товар на количество.

In [20]:

```
# посчитана общая выручка с каждого товара
dfc['total'] = dfc['price'] * dfc['quantity']
dfc.sample(n=3, random_state=1)
```

Out[20]:

| | date | customer_id | order_id | product | quantity | price | total |
|---|---|---|---|---|---|---|---|
| **7320** | 2020-01-13 23:00:00 | 37d55ae9-9cf6-4c37-8ad0-01a4df15bc9b | 103901 | Пеларгония розебудная Apple Blossom укорененный черенок | 2 | 149.0 | 298.0 |
| **1086** | 2018-12-03 11:00:00 | 2c8b9386-2e8c-4ee9-8aa4-2b4875b68578 | 14514 | Эшшольция Карминный король 0,5 г 4660010770193 | 2 | 10.0 | 20.0 |
| **6564** | 2019-10-10 11:00:00 | d0a10ee2-fbdb-42df-8f23-46fa724fe7a3 | 72914 | Сумка-тележка хозяйственная GIMI Ideal синяя | 1 | 1649.0 | 1649.0 |

### 2.4.2 Месяц_год заказа

Выделим месяц и год покупки м добавим столбец `year_month` в датафрейм `dfc`.

In [21]:

```
# создание колонки с датой покупки в формате год-месяц
dfc['year_month'] = dfc['date'].dt.strftime('%Y-%m')
```

## 2.5 Итоги раздела предобработки данных

Столбцы поименованы с соблюдением стиля, значит корректировка не требуется.
В колонке `date` тип данных не соответствовал содержимому. Поэтому тип данных с целочисленного был заменен на формат даты.
Добавлен столбец `total`, в которую записан выручка по каждой строке заказа.
Добавлен столбец `year_month`, в которую записан месяци год заказа.
Предварительно можно утверждать, что предоставленного объема данных достаточно для исследования.
Предобработка завершена. Можно приступать к анализу данных.

# 3 Исследовательский анализ данных

## 3.1 Среднее количество заказов

Посмотрим количество уникальных значений по пред обработанному датасету `dfc`.

In [22]:

```
dfc.nunique()
```

Out[22]:

```
date          2700
customer_id   2413
order_id      3492
product       2334
quantity        49
price          407
total          713
year_month      16
dtype: int64
```

In [23]:

```
print(f"На одного покупателя в среднем приходится \
{round(len(dfc['order_id'].unique()) / len(dfc['customer_id'].unique()), 2)}
заказов.")
```

На одного покупателя в среднем приходится 1.45 заказов.

Всего насчитывается:

- 2413 уникальных покупателя, которые совершили 3492 заказов из ассортимента 2334 уникальных названий товара.

Заказов больше, чем покупателей, на одного покупателя в среднем приходится 1.45 заказов.

## 3.2 Количество и стоимость товаров

Посмотрим сколько товаров попадает в заказ. На диаграмме рассеяния сразу будет видна общая тенденция

In [24]:

```
# построена диаграмма рассеяния по количеству и стоимости товара в заказе
fig = px.scatter(dfc, x="total",
                 y="quantity",
                 opacity=0.9, # прозрачность
                 template='plotly_white', #цвет подложки
                 color_discrete_sequence=['indianred'],# ['goldenrod'], #['indianred'],
['green'] # color of histogram bars
                 width=650, height=350  #размер графика
                )
# Update the layout
fig.update_layout(xaxis_title='На какую сумму куплен товар в заказе',
                  yaxis_title='Количество товара в заказе',
                  showlegend=False)
fig.update_layout(title='Количество и стоимость товара в одном заказе', title_x=0.5) #
название
fig.show()
```


В заказ попал товар в количестве 1000 шт на общую сумму 675тыс. Остальные товары далеки от таких рекордов, есть товары в количестве 334, 300, 200 штук на не очень большую общую сумму не выше 49тыс и более дорогие товары в количестве 27 шт на 47 тыс.

Посмотрим на товар, который заказали в количестве более 100 штук в заказе.

In [25]:

```
dfc.query('customer_id == "aa42dc38-780f-4b50-9a65-83b6fa64e766"')
```

Out[25]:

| | date | customer_id | order_id | product | quantity | price | total | year_month |
|---|---|---|---|---|---|---|---|---|
| **568** | 2018-11-01 08:00:00 | aa42dc38-780f-4b50-9a65-83b6fa64e766 | 68815 | Муляж ЯБЛОКО 9 см красное | 170 | 51.0 | 8670.0 | 2018-11 |

## 3.3 Покупателей по количеству товаров в заказе

In [26]:

```
# кто заказал товар более 100 шт в заказе
display(dfc.query('quantity >= 100').sort_values(by=['quantity'], ascending = False).head(30))
```

| | date | customer_id | order_id | product | quantity | price | total | year_month |
|---|---|---|---|---|---|---|---|---|
| **5456** | 2019-06-18 15:00:00 | 312e9a3e-5fca-43ff-a6a1-892d2b2d5ba6 | 71743 | Вантуз с деревянной ручкой d14 см красный, Burstenmann, 0522/0000 | 1000 | 675.0 | 675000.0 | 2019-06 |
| **5071** | 2019-06-11 07:00:00 | 146cd9bf-a95c-4afb-915b-5f6684b17444 | 71668 | Вешалки мягкие для деликатных вещей 3 шт шоколад | 334 | 148.0 | 49432.0 | 2019-06 |
| **3961** | 2019-05-20 21:00:00 | 5d189e88-d4d6-4eac-ab43-fa65a3c4d106 | 71478 | Муляж ЯБЛОКО 9 см красное | 300 | 51.0 | 15300.0 | 2019-05 |
| **1158** | 2018-12-10 14:00:00 | a984c5b7-ff7e-4647-b84e-ef0b85a2762d | 69289 | Ручка-скоба РС-100 белая *Трибатрон*, 1108035 | 200 | 29.0 | 5800.0 | 2018-12 |
| **568** | 2018-11-01 08:00:00 | aa42dc38-780f-4b50-9a65-83b6fa64e766 | 68815 | Муляж ЯБЛОКО 9 см красное | 170 | 51.0 | 8670.0 | 2018-11 |
| **211** | 2018-10-11 14:00:00 | cd09ea73-d9ce-48c3-b4c5-018113735e80 | 68611 | Крепеж для пружины дверной, 1107055 | 150 | 19.0 | 2850.0 | 2018-10 |
| **212** | 2018-10-11 14:00:00 | cd09ea73-d9ce-48c3-b4c5-018113735e80 | 68611 | Пружина дверная 240 мм оцинкованная (Д-19 мм) без крепления, 1107014 | 150 | 38.0 | 5700.0 | 2018-10 |
| **2431** | 2019-03-23 10:00:00 | 685d3d84-aebb-485b-8e59-344b3df8b3d3 | 70841 | Плечики пластмассовые Размер 52 - 54 Тула 1205158 | 150 | 20.0 | 3000.0 | 2019-03 |
| **586** | 2018-11-02 11:00:00 | 0c5aaa88-e346-4f87-8f7a-ad8cbc04e965 | 68831 | Муляж ЯБЛОКО 9 см красное | 140 | 59.0 | 8260.0 | 2018-11 |
| **1103** | 2018-12-04 17:00:00 | 7d255526-fcc2-4f79-b28a-217d7d2373a8 | 69206 | Щетка для посуды *ОЛЯ*, Мультипласт 1807010 | 100 | 26.0 | 2600.0 | 2018-12 |
| **1555** | 2019-01-21 09:00:00 | 8eabcaca-e8c8-4eee-9079-4ff5f612273a | 69893 | Щетка для мытья посуды КОЛИБРИ М5202 большая | 100 | 34.0 | 3400.0 | 2019-01 |
| **6535** | 2019-10-07 11:00:00 | d933280e-5372-448f-be44-b269c8bafc2a | 72885 | Крепеж для пружины дверной оцинкованный, 1107054 | 100 | 19.0 | 1900.0 | 2019-10 |
| **6707** | 2019-10-28 10:00:00 | 018fb729-3525-4314-8e4d-1982b1062f9f | 73110 | Шпингалет 80 мм белый с пружиной, 1102188 | 100 | 44.0 | 4400.0 | 2019-10 |

In [27]:

```
# список покупателей заказавших товар более 100 штук
top_customer_per_quantity = list(dfc.query('quantity > 150')['customer_id'].unique())

print(f"{dfc.query('quantity >= 100')['customer_id'].nunique()} покупателей заказавших товар \
в кол-ве более 100 шт в заказе. Заказов \
{dfc.query('quantity >= 100')['order_id'].nunique()} штук.")
```

12 покупателей заказавших товар в кол-ве более 100 шт в заказе. Заказов 12 штук.

### 3.3.1 Выбросы по количеству товара

По количеству штук есть рекордсмены среди заказов это:

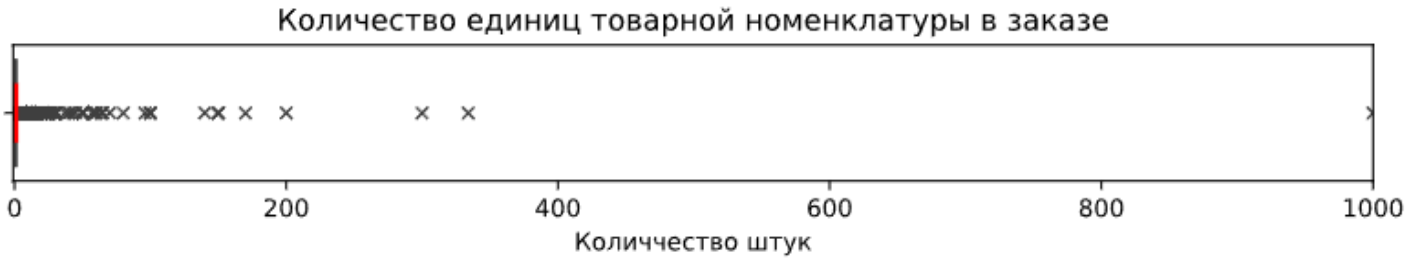
- `Вантуз с деревянной ручкой d14 см красный, Burstenmann, 0522/0000` - 1000 шт
- `Вешалки мягкие для деликатных вещей 3 шт шоколад` - 334 шт
- `Муляж ЯБЛОКО 9 см красное` - 300 шт
- `Ручка-скоба РС-100 белая *Трибатрон*, 1108035` - 200 шт

- `Муляж ЯБЛОКО 9 см красное` - 170 шт

Построим боксплот по количеству товара в заказе `quantity` и посмотрим, как распределяется количество товаров в заказах.

In [28]:

```
plt.figure(figsize=(10,1))

sns.boxplot(x=dfc['quantity'], notch=True, showcaps=False,
   flierprops={"marker": "x"},
   #boxprops={"facecolor": (.3, .5, .7, .5)},
   medianprops={"color": "r", "linewidth": 2})


plt.xlabel('Количчество штук')
plt.ylabel('')
plt.title('Количество единиц товарной номенклатуры в заказе')
plt.xlim(-1, 1000);
```




В заказах в основной массе товарная номенклатура в малом количестве заказывается. Боксплот состоит из одних выбросов. Но Некоторые товары вполне могут быть заказаны в таком количестве для бизнеса, даже для небольшого хостела потребуются несколько сотен вешалок.

На данном о очистке от выбросов заказы в которых больше 400 единиц товаров, а это заказ `71743` с товаром `Вантуз с деревянной ручкой d14 см красный, Burstenmann, 0522/0000` - 1000 шт.

## 3.4 Покупатели по сумме заказов

Теперь посмотрим на позиции в заказе на общую сумму более 10тыс. у.е. на один заказ.

In [29]:

```
# кто заказал товар на сумму более 10000 у.е. в заказе

display(dfc.query('total > 10000').sort_values(by=['total'], ascending =
False).head(6))
print(f"{dfc.query('total > 10000')['customer_id'].nunique()} покупателей кто заказал
уникальный \
товар более чем на 10000 у.е. в заказе. Заказов \
{dfc.query('total > 10000')['order_id'].nunique()} штук.")
```

| | date | customer_id | order_id | product | quantity | price | total | year_month |
|---|---|---|---|---|---|---|---|---|
| **5456** | 2019-06-18 15:00:00 | 312e9a3e-5fca-43ff-a6a1-892d2b2d5ba6 | 71743 | Вантуз с деревянной ручкой d14 см красный, Burstenmann, 0522/0000 | 1000 | 675.0 | 675000.0 | 2019-06 |
| **5071** | 2019-06-11 07:00:00 | 146cd9bf-a95c-4afb-915b-5f6684b17444 | 71668 | Вешалки мягкие для деликатных вещей 3 шт шоколад | 334 | 148.0 | 49432.0 | 2019-06 |
| **1518** | 2019-01-15 13:00:00 | 58a4c3cc-504f-43ea-a74a-bae19e665552 | 69830 | Простынь вафельная 200x180 см WELLNESS RW180-01 100% хлопок | 27 | 1755.0 | 47385.0 | 2019-01 |
| **1299** | 2018-12-21 16:00:00 | 6987e6d6-a63a-4ce2-a2d0-f424092d235e | 69504 | Тележка багажная DELTA ТБР-22 синий грузоподъемность 20 кг сумка и 50 кг каркас РОССИЯ | 57 | 574.0 | 32718.0 | 2018-12 |
| **1023** | 2018-11-29 17:00:00 | 8fba3604-ef57-4b9f-b2fe-3402fa8825c8 | 69156 | Набор ножей Attribute CHEF 5 предметов AKF522 | 64 | 457.0 | 29248.0 | 2018-11 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **661** | 2018-11-06 12:00:00 | 1b2764ad-9151-4051-a46a-9b01b62e6335 | 68878 | Урна уличная "Гео", Hobbyka/Хоббика, 59*37,5см, сталь | 5 | 4874.0 | 24370.0 | 2018-11 |

14 покупателей кто заказал уникальный товар более чем на 10000 у.е. в заказе. Заказов 16 штук.

## 3.5 Покупатели по общей сумме всех заказов

Посчитаем общую сумму заказа по каждому покупателю за весь период.

In [30]:

```
# посчитана общая выручка с каждого покупателя
customer_id_top_total =
dfc.groupby('customer_id')['total'].sum().sort_values(ascending=False)
customer_id_top_total = customer_id_top_total[customer_id_top_total > 10000]
print(f"На общую сумму более 10000 у.е. за все время  заказали
{len(customer_id_top_total)} покупателей,\n\
это {round(len(customer_id_top_total) / len(dfc.groupby('customer_id')['total'].sum())
* 100, 2)} % от всех покупателей.")
customer_id_top_total.head(6)
```

На общую сумму более 10000 у.е. за все время  заказали 25 покупателей,
это 1.04 % от всех покупателей.

Out[30]:

customer_id
312e9a3e-5fca-43ff-a6a1-892d2b2d5ba6  675000.0
c971fb21-d54c-4134-938f-16b62ee86d3b  159508.0
4d93d3f6-8b24-403b-a74b-f5173e40d7db  57278.0
58a4c3cc-504f-43ea-a74a-bae19e665552  53232.0
146cd9bf-a95c-4afb-915b-5f6684b17444  49432.0
498f12a4-6a62-4725-8516-cf5dc9ab8a3a  41900.0
Name: total, dtype: float64

## 3.6 Покупатели по среднему чеку

Посчитаем средний чек с каждого покупателя, который превышает 10000 у.е.

In [31]:

```
# средний чек с каждого покупателя свыше 10000
customer_id_top_mean =
dfc.groupby('customer_id')['total'].mean().sort_values(ascending=False)
customer_id_top_mean = customer_id_top_total[customer_id_top_mean >= 10000]
print(f"Средний чек больше 10000 у.е. за все время у {len(customer_id_top_mean)}
покупателей,\n\
это {round(len(customer_id_top_mean) / len(dfc.groupby('customer_id')['total'].mean())
* 100, 2)} % от всех покупателей.")
customer_id_top_mean
```

Средний чек больше 10000 у.е. за все время у 12 покупателей,
это 0.5 % от всех покупателей.

Out[31]:

customer_id
312e9a3e-5fca-43ff-a6a1-892d2b2d5ba6  675000.0
58a4c3cc-504f-43ea-a74a-bae19e665552  53232.0
146cd9bf-a95c-4afb-915b-5f6684b17444  49432.0
498f12a4-6a62-4725-8516-cf5dc9ab8a3a  41900.0

6987e6d6-a63a-4ce2-a2d0-f424092d235e 32718.0
1b2764ad-9151-4051-a46a-9b01b62e6335 24370.0
940c175f-ea87-44e0-9e16-0a3d0a9abecd 20232.0
909564b8-3a5c-4d3e-8310-5ba1c837bbd7 16536.0
5d189e88-d4d6-4eac-ab43-fa65a3c4d106 15300.0
0d87f4ae-465a-4fac-81e6-5d629761783e 14917.0
6be74251-7159-4cc0-99fb-d034a17c61b0 11250.0
af4d270b-c7ae-4af5-9582-4e61ff08eff0 11000.0
Name: total, dtype: float64

Посчитаем у скольки покупателей средний чек в диапазоне от 4000 до 10000 у.е. и у скольки меньше 4000 у.е.

In [32]:

```
# посчитана средний чек с каждого покупателя  от 4000 до 10000
customer_id_mean =
dfc.groupby('customer_id')['total'].mean().sort_values(ascending=False)
customer_id_mean = customer_id_mean[customer_id_mean < 10000]
customer_id_mean = customer_id_mean[customer_id_mean >= 4000]
print(f"Средний чек меньше 10000 у.е. и не выше 4000 за все время у
{len(customer_id_mean)} покупателей,\n\
это {round(len(customer_id_mean) / len(dfc.groupby('customer_id')['total'].mean()) *
100, 2)} % от всех покупателей.\n\
Остальные \
{(100 - round(len(customer_id_mean) / len(dfc.groupby('customer_id')['total'].mean())
* 100, 2)-round(len(customer_id_top_mean) /
len(dfc.groupby('customer_id')['total'].mean()) * 100, 2))}\
% покупателей имеют средний чек ниже 4000 у.е.")
customer_id_mean
```

Средний чек меньше 10000 у.е. и не выше 4000 за все время у 74 покупателей,
это 3.07 % от всех покупателей.
Остальные 96.43 % покупателей имеют средний чек ниже 4000 у.е.

Out[32]:

customer_id
794e66f5-4d30-4860-b44c-903c9f58127f 8810.0
c0c60544-3a99-49d0-8a8e-cf7f293c22cb 8737.0
aa42dc38-780f-4b50-9a65-83b6fa64e766 8670.0
f279d50f-a508-40b4-bde5-5cb4a1be3ad0 8278.5
0c5aaa88-e346-4f87-8f7a-ad8cbc04e965 8260.0
...
04416514-5346-4f90-93e3-fb7365e2ee8c 4053.5
a1034d4c-91e4-42f5-87d5-75c7313e06ef 4049.0
f08dabe2-ce2a-46c2-8369-3f40955ad50c 4049.0
c64f3a03-5fa8-4c41-96ad-ae1ecfa486e0 4049.0
6cc2b353-7824-4f48-b0a5-c44f6e2a4fb7 4040.0
Name: total, Length: 74, dtype: float64

## 3.7 Заказы с наибольшим количеством позиций(строк)

Посмотрим на заказы, в которых много позиций разных товаров. Для этого посчитаем строки с `order_id`

In [33]:

```
print("Топ-10 заказов с наибольшим количеством позиций(строк):")
df['order_id'].value_counts().head(10)
```

Топ-10 заказов с наибольшим количеством позиций(строк):

Out[33]:

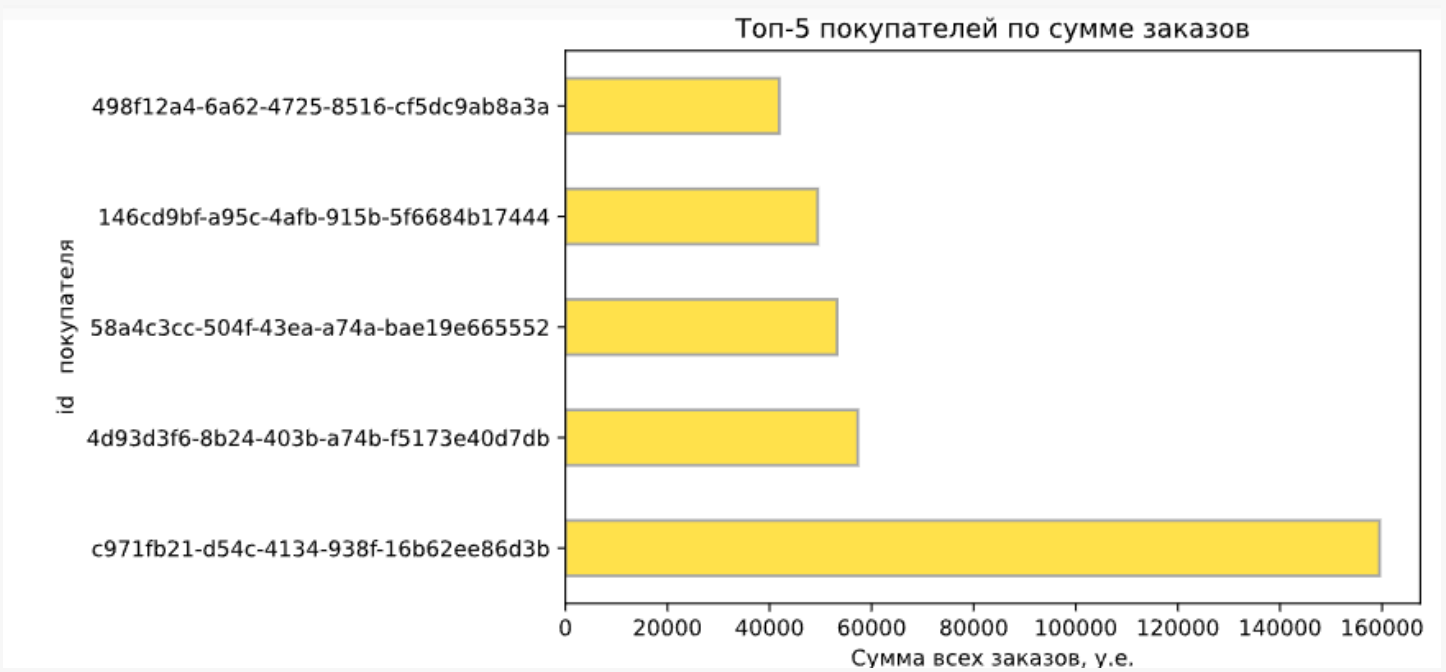
```
14833   888
14835   203
14753    90
14897    63
70960    60
14698    51
68760    50
14715    36
14500    34
14688    31
Name: order_id, dtype: int64
```

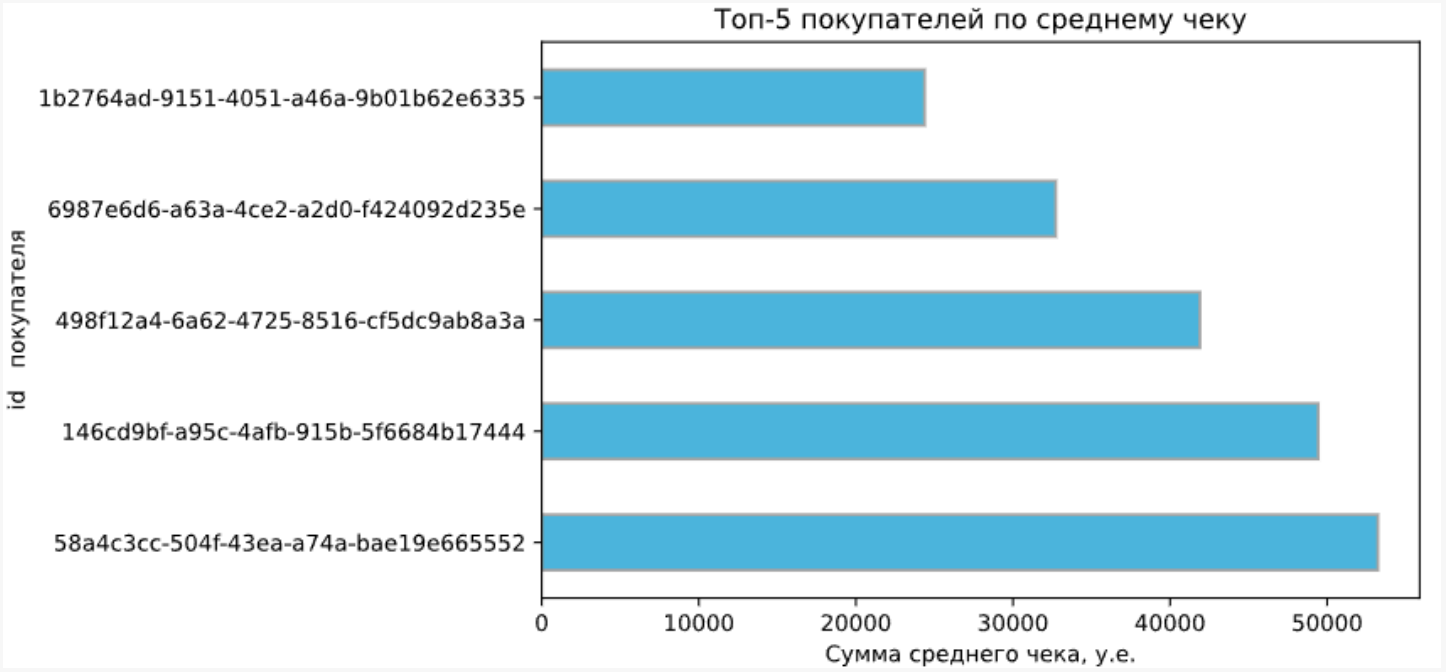
## 3.8 Графики покупателей топ-5 по сумме заказов и топ-5 по среднему чеку

Визуализируем результаты Топ-5 по среднему чеку и Топ-5 по общей сумме покупок. Покупателя `312e9a3e-5fca-43ff-a6a1-892d2b2d5ba6` вантузов на 675000 у.е. исключим из топ-5, так как он вне конкуренции. Этот супер оптовик специфического товара, заказавший один раз в июля 2019 года затруднит Визуализацию, так как показатель 675тыс сильно растянет весь график. Выведем два графика рядом.

In [34]:

```
#  используем subplot function построим первый график
plt.subplot(2, 1, 1)  # row 1, column 2, count 1
customer_id_top_total[customer_id_top_total < 675000].head(5).plot(
   kind='barh', figsize=(7, 10), alpha=0.7, color='#FFD700', ec='#808080',
linewidth=1.3) #, grid=True)
plt.title('Топ-5 покупателей по сумме заказов', )
plt.ylabel('id   покупателя')
plt.xlabel('Сумма всех заказов, у.е.');
# используем subplot function построим второй график
# row 1, column 2, count 2
plt.subplot(2, 1, 2)

customer_id_top_mean[customer_id_top_mean < 675000].head(5).plot(
   kind='barh', figsize=(7, 10), alpha=0.7, color='#009ACD', ec='#808080',
linewidth=1.3) #, grid=True)
plt.title('Топ-5 покупателей по среднему чеку', )
plt.ylabel('id   покупателя')
plt.xlabel('Сумма среднего чека, у.е.');
plt.show()
```




Покупатели из Топ-5 с единственным заказом на общую сумму и из Топ-5 со средним чеком:

- Покупатель `58a4c3cc-504f-43ea-a74a-bae19e665552` средний чек и общая сумма равна 53232.0. Заказ был всего один.
- Покупатель `146cd9bf-a95c-4afb-915b-5f6684b17444` средний чек 49432.0 у.е., а общая сумма 49432.0. Заказ был всего один.
- Покупатель `498f12a4-6a62-4725-8516-cf5dc9ab8a3a` средний чек 41900.0 у.е., а общая сумма 41900.0. Заказ был всего один.

Оставшиеся 2 покупателей из Топ-5 по среднему чеку не входят в топ-5 покупателей. принесших наибольшую прибыль. Значит это также покупатели, совершившие скорее всего по одной покупке, иначе бы они с большей долей вероятности попали в топ-5 по общей сумме всех заказов.

## 3.9 Товарная номенклатура

Посмотрим какую номенклатуру товара чаще всего покупали. Подсчитаем номенклатуру товара и выведем топ -20 товаров, которые заказали больше всего.

In [35]:

```
# просмотр кол. уникальных значений столбца product
print(f"Топ-20 номенклатурных товаров заказали
{dfc['product'].value_counts().head(20).sum()} раз")
top_20_product = dfc['product'].value_counts().head(20)
```

```
top_20_product
```

Топ-20 номенклатурных товаров заказали 574 раз

Out[35]:

Пеларгония розебудная Red Pandora укорененный черенок 65
Пеларгония розебудная Prins Nikolai укорененный черенок 54
Пеларгония зональная диам. 12 см сиреневый полумахровый 53
Сумка-тележка 2-х колесная Gimi Argo синяя 47
Пеларгония розебудная Mary укорененный черенок 40
Пеларгония розебудная Margaretha укорененный черенок 31
Пеларгония розебудная Queen Ingrid укорененный черенок 31
Пеларгония зональная Ринго Вайт d-7 см h-10 см укорененный черенок 25
Пеларгония зональная махровая лососевая 25
Пеларгония зональная диам. 12 см коралловая полумахровая 24
Пеларгония зональная диам. 12 см темнорозовая полумахровая 22
Пеларгония зональная диам. 12 см красная махровая 21
Однолетнее растение Петуния махровая в кассете 4 шт, Россия 20
Пеларгония розебудная Rosebud Red d-7 см 18
Тележка багажная DELTA ТБР-20 синий грузоподъемность 25 кг сумка и 50 кг каркас РОССИЯ 17
Сушилка для белья настенная Zalger Prima 510-720 веревочная 7 линий 25 м 17
Петуния махровая рассада однолетних цветов в кассете по 10 шт 16
Тележка багажная DELTA ТБР-20 коричневый с оранжевым грузоподъемность 25 кг сумка и 50 кг каркас РОССИЯ 16
Пеларгония Toscana Angeleyes Amarillo Burgundy укорененный черенок 16
Пеларгония зональная диам. 12 см сиреневый простая 16
Name: product, dtype: int64

In [36]:

```
print(f"Сумка-тележка 2-х колесная Gimi... заказали 47 раз -
{round(47/top_20_product.sum() * 100, 2)} % от топ-20 заказа.\n\
Сушилка для белья настенная Zalger... заказали 17 раз - {round(17/top_20_product.sum()
* 100, 2)} %от топ-20 заказа.\n\
Тележка багажная DELTA ТБР-20... заказали {(17+16)} раз
-{round((17+16)/top_20_product.sum() * 100, 2)} %от топ-20 заказа.\n\
Пеларгония...заказали {(574-47-17-17-16-20-16)} раз -\
{round((574-47-17-17-16-20-16)/top_20_product.sum() * 100, 2)} %от топ-20 заказа.\n\
Петуния...заказали  20 раз - {round((20)/top_20_product.sum() * 100, 2)} %от топ-20
заказа.")
```

Сумка-тележка 2-х колесная Gimi... заказали 47 раз - 8.19 % от топ-20 заказа.
Сушилка для белья настенная Zalger... заказали 17 раз - 2.96 %от топ-20 заказа.
Тележка багажная DELTA ТБР-20... заказали 33 раз -5.75 %от топ-20 заказа.
Пеларгония...заказали 441 раз -76.83 %от топ-20 заказа.
Петуния...заказали  20 раз - 3.48 %от топ-20 заказа.

Создадим датафрейм из топ-5 товарных номенклатур. Пеларгонию в ассортименте и тележки объединила. Визуализируем топ-5 товаров на диаграмме.

In [37]:

```
# Creates pandas DataFrame.
df_top_product = pd.DataFrame(data = {'count':[441, 47, 33, 20, 17]},
                              index = ['Пеларгония...',
                                       'Сумка-тележка 2-х колесная Gimi...',
```

```
                                    'Тележка багажная DELTA ТБР-20...',
'Петуния...',
                                    'Сушилка для белья настенная Zalger...'])
df_top_product
```

Out[37]:

| | count |
|---|---|
| **Пеларгония...** | 441 |
| **Сумка-тележка 2-х колесная Gimi...** | 47 |
| **Тележка багажная DELTA ТБР-20...** | 33 |
| **Петуния...** | 20 |
| **Сушилка для белья настенная Zalger...** | 17 |

In [38]:

```
df_top_product.plot(
   kind='barh', figsize=(5, 3), alpha=0.7, color='#228B22', ec='#808080',
linewidth=1.3, legend=False, grid=True)
plt.title('Топ-5 товаров по количеству заказов', )
plt.xlabel('Количество уникальных заказов')
plt.ylabel('')

plt.show()
```


Посмотрим какой товар приносит больше всего выручки. Сгруппируем и выведем топ-10 товарных единиц, которые принесли самую большую выручку. Вантуз, который заказали на 675000 в топ-20 входить не будет, он вне конкуренции.

In [39]:

```
# общая выручка с каждого товара
dfc.query('product != "Вантуз с деревянной ручкой d14 см красный, Burstenmann,
0522/0000"')\
.groupby('product')['total'].sum().sort_values(ascending=False).head(10)
```

Out[39]:

product

Простынь вафельная 200х180 см WELLNESS RW180-01 100% хлопок 53232.0

Сумка-тележка 2-х колесная Gimi Argo синяя 50405.0

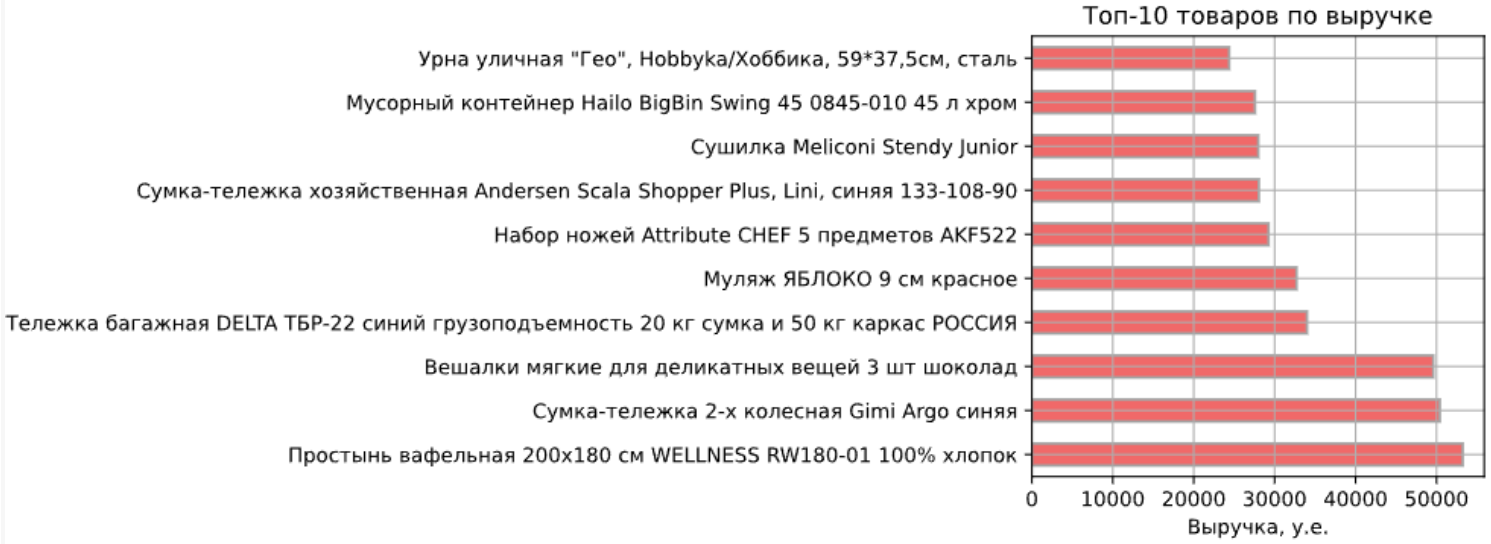
Вешалки мягкие для деликатных вещей 3 шт шоколад 49596.0

Тележка багажная DELTA ТБР-22 синий грузоподъемность 20 кг сумка и 50 кг каркас РОССИЯ 33992.0

Муляж ЯБЛОКО 9 см красное 32702.0
Набор ножей Attribute CHEF 5 предметов AKF522 29248.0
Сумка-тележка хозяйственная Andersen Scala Shopper Plus, Lini, синяя 133-108-90 28045.0
Сушилка Meliconi Stendy Junior 27970.0
Мусорный контейнер Hailo BigBin Swing 45 0845-010 45 л хром 27560.0
Урна уличная "Гео", Hobbyka/Хоббика, 59*37,5см, сталь 24370.0
Name: total, dtype: float64

In [40]:

```
dfc.query('product != "Вантуз с деревянной ручкой d14 см красный, Burstenmann, 0522/0000"')\
.groupby('product')['total'].sum().sort_values(ascending=False).head(10)\
.plot(kind='barh',
   figsize=(4, 4),
   alpha=0.7, color='#EE2C2C', ec='#808080',
   linewidth=1.3, legend=False, grid=True)
plt.title('Топ-10 товаров по выручке')
plt.xlabel('Выручка, у.е.')
plt.ylabel('')
plt.show();
```




## 3.10 Цена товара

Посмотрим на диапазон цен товаров интернет-магазина.

In [41]:

```
df['price'].sort_values(ascending=False).head(10)
```

Out[41]:

5992 14917.0
2697 8737.0
1981 8437.0
2997 8077.0
7436 8077.0
6629 7724.0
5994 7679.0
7190 7679.0
2339 7679.0
654 7597.0
Name: price, dtype: float64

In [42]:

```
df['price'].hist(color='teal', figsize=(5, 3))
plt.ylabel('Количество товаров')
plt.xlabel('Цена у.е.')
plt.title('Диапазон цен на товары');
```


Видим, что разброс на ценах очень большой, причем незаметный на гистограмме хвост тянется вправо до 14000 у.е. Большинство товаров имеют ценник не выше 2000, точнее по гистограмме не рассмотреть. Рассмотрим

In [43]:

```
fig = px.histogram(dfc.query('price <= 2000'), # датасет
                   x="price", # выбор столбца
                   #histnorm='percent', # в процентах ось y
                   nbins=20, #регулируем количество bins
                   #title='', # название
                   opacity=0.8, # прозрачность
                   template='plotly_white', #цвет подложки
                   width=550, height=350,  #размер графика
                   color_discrete_sequence=['#228B22'],# #228B22['goldenrod'],
#['indianred'], ['green'] # color of histogram bars
                  )
fig.update_layout(title_text = 'Товар с ценой до 2000у.е', title_x=0.5)  # название
fig.update_layout(
   xaxis={'title':'Цена, у.е.'},
   yaxis={'title':'кол-во'})  # название осей
fig.update_traces(marker_line_color= '#030303', #'rgb(18,98,107)',
                  marker_line_width=1.5, opacity=0.8) #добавление обводки
fig.show()

fig = px.histogram(dfc.query('price > 2000 & price <= 8000'), # датасет
                   x="price", # выбор столбца
                   #histnorm='percent', # в процентах ось y
                   nbins=20, #регулируем количество bins
                   #title='', # название
                   opacity=0.8, # прозрачность
                   template='plotly_white', #цвет подложки
                   width=550, height=250,  #размер графика
                   color_discrete_sequence=['#CDAD00'],# ['goldenrod'], #['indianred'],
['green'] # color of histogram bars
```

```
                  )
fig.update_layout(title_text = 'Товар с ценой от 2000 до 8000у.е', title_x=0.5)  #
название
fig.update_layout(
   xaxis={'title':'Цена, у.е.'},
   yaxis={'title':'кол-во'})  # название осей
fig.update_traces(marker_line_color= '#030303', #'rgb(18,98,107)',
                  marker_line_width=1.5, opacity=0.8) #добавление обводки
fig.show()


fig = px.histogram(dfc.query('price > 8000'), # датасет
                   x="price", # выбор столбца
                   #histnorm='percent', # в процентах ось y
                   nbins=20, #регулируем количество bins
                   #title='', # название
                   opacity=0.8, # прозрачность
                   template='plotly_white', #цвет подложки
                   width=550, height=220,  #размер графика
                   color_discrete_sequence=['#CD2626'],# ['goldenrod'], #['indianred'],
['green'] # color of histogram bars
                  )
fig.update_layout(title_text = 'Товар с ценой свыше 8000у.е', title_x=0.5)  # название
fig.update_layout(
   xaxis={'title':'Цена, у.е.'},
   yaxis={'title':'кол-во'})  # название осей
fig.update_traces(marker_line_color= '#030303', #'rgb(18,98,107)',
                  marker_line_width=1.5, opacity=0.8) #добавление обводки
fig.show()
```



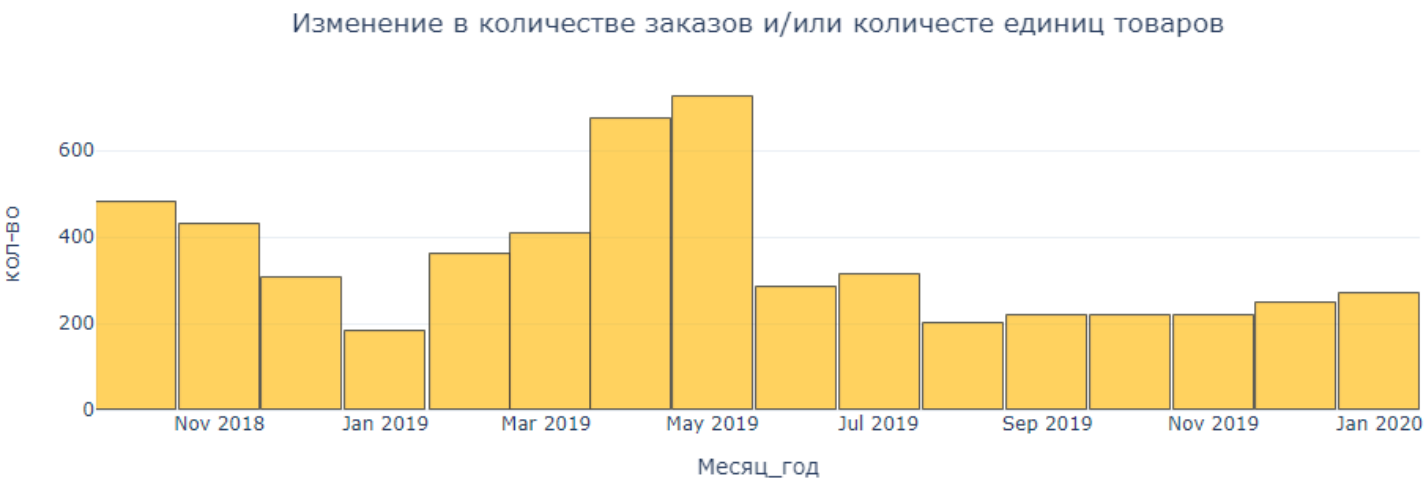
Почти 1,5тыс товаров с ценой до 100 у.е, свыше 1,8тыс товаров с ценой 100-200 у.е. Ассортимент товаров с ценой свыше 200 у.е. резко снижается.Товаров с ценой 7500-8000 у.е. всего 5 шт, 3500-4000 у.е. - 46 шт. Дорогой товар свыше 14.5тыс у.е. один, 8.25-9тыс. у.е. также один и три товара за 8тыс-8.5 тыс у.е.

## 3.11 Активность покупателей по месяцам

Посмотрим, в каком месяце продается больше всего наименований товара (строк в заказе). Эта гистограмма даст понимание в каком месяце больше всего заказов или в заказах покупатели добавляют разный товар. На гистограмме не показано количество товара. В целом это просто даст представление, есть ли период изменения активности в количестве заказов или в ассортимента заказываемых товаров.

In [44]:

```
fig = px.histogram(dfc, # датасет
                   x="year_month", # выбор столбца
                   #histnorm='percent', # в процентах ось y
                   nbins=20, #регулируем количество bins
                   #title='', # название
                   opacity=0.9, # прозрачность
                   template='plotly_white', #цвет подложки
                   width=1000, height=350,  #размер графика
                   color_discrete_sequence=['#FFC125'],# #228B22['goldenrod'],
#['indianred'], ['green'] # color of histogram bars
                  )
fig.update_layout(title_text = 'Изменение в количестве заказов и/или количестве единиц товаров', title_x=0.5)  # название
fig.update_layout(
   xaxis={'title':'Месяц_год'},
   yaxis={'title':'кол-во '})  # название осей
fig.update_traces(marker_line_color= '#030303', #'rgb(18,98,107)',
                  marker_line_width=0.9, opacity=0.8) #добавление обводки
fig.show()
```




В апреле и мае есть рост в количестве заказов и/или в увеличении ассортимента заказываемых товаров в заказе. С июня по январь количество заказов и/или количество наименований товаров стабильное.

## 3.12 Выводы исследовательского анализа данных

Исходя из проведенного анализа обобщенных данных, можно сделать вывод, что среди покупателей нашего интернет магазина высокий средний чек появляется у разовых клиентов. Топ-5 по среднему чеку и топ-5 по общей сумме заказов нам это показало.

При сопоставлении из таблицы заказа и наименования товара можно сделать вывод, что крупные заказы с большим средним чеком - разовые для бизнеса, например встречались простыни, одеяла в одном заказе с количеством более 10 шт. Или `Тележка багажная DELTA ТБР-22 синий грузоподъемность 20 кг сумка и 50 кг каркас РОССИЯ` в количестве 57 штук и т.д.

Обобщим полученные данные по покупателям, заказам, товару.

**Средний чек**.

- Имеется один аномальный заказ со средним чеком в 675000 у.е.
- Средний чек больше 10000 у.е. всего у 12 покупателей (0.5 % от всех покупателей), среди которых конечно же и наш чемпион со средним чеком 675тыс у.е.
- Мы выяснили, что средний чек в пределах 4000-10000 у.е. у 74 покупателей (3.07 % от всех покупателей).
- Остальные 96.43 % покупателей имеют средний чек ниже 4000 у.е.

**Общая сумма заказа**

- Имеется один аномальный заказ на общую сумму 675000 у.е. его не включали в топ-5, он вне конкуренции.
- На общую сумму более 10000 у.е. за все время заказали 25 покупателей (1.04 % от всех покупателей) в топ топ-5 лидирует покупатель `c971fb21-d54c-4134-938f-16b62ee86d3b` с общей суммой в `159508 у.е.`Остальные 4 покупателя набрали на суммы в пределах от 41900 до 57278 у.е.

**Количество наименований товаров в заказе**

- два заказа с большим количеством наименований товаров (строк) в 888 и 203 позиций в заказах `14833` и `14835`
- пять заказов от 90 до 50 строк, остальные
- остальные заказы от 1 до 36 позиций(строк) в одном заказе.

**Количество товаров в заказе, штук**

- имеется один аномальный заказ с количеством вантузов 1000 штук. Это наш чемпион со средним чеком и общей суммо в 675000 у.е.
- мы выяснили, что у нас 11 заказов у 11 разных покупателей, которые приобрели товарную номенклатуру в кол-ве 100 штук и более за один заказ.

**Топ популярной номенклатуры товаров**

Товар, который заказали много раз, т.е. не общее количество проданных штук, а количество уникальных заказов, в которые вошли эти товары:

- Пеларгония... ее в различных вариантах заказали аж 448 раз, это 78.05 % от топ-20 заказа.
- Сумка-тележка 2-х колесная Gimi... заказали 40 раз 6.97 % от топ-20 заказа.

- Сушилка для белья настенная Zalger... заказали 17 раз 2.96 % от топ-20 заказа.
- Тележка багажная DELTA ТБР-20... заказали 33 раз 5.75 % от топ-20 заказа.
- Петуния...заказали 36 раз 6.27 %от топ-20 заказа.

**Товары принесшие больше всего выручки**

- Простынь вафельная 200х180 см WELLNESS RW180-01 100% хлопок 53232.0
- Сумка-тележка 2-х колесная Gimi Argo синяя 50405.0
- Вешалки мягкие для деликатных вещей 3 шт шоколад 49596.0
- Тележка багажная DELTA ТБР-22 синий грузоподъемность 20 кг сумка и 50 кг каркас РОССИЯ 33992.0
- Муляж ЯБЛОКО 9 см красное 32702.0
- Набор ножей Attribute CHEF 5 предметов AKF522 29248.0
- Сумка-тележка хозяйственная Andersen Scala Shopper Plus, Lini, синяя 133-108-90 28045.0
- Сушилка Meliconi Stendy Junior 27970.0
- Мусорный контейнер Hailo BigBin Swing 45 0845-010 45 л хром 27560.0
- Урна уличная "Гео", Hobbyka/Хоббика, 59 37,5см, сталь 24370.0

**Цена на товар**

Основной ассортимент товаров в интернет магазине составляет товар со стоимостью до 200 у.е. Самый дорогой товар стоит около 14.5 тыс у.е. При этом товарные позиции стоимостью от 2000 до 9000 у.е. присутствуют равномерным хвостом. Также ectm самый дорогой товар стоиомстью 14917 у.е.

**Сезонность** В апреле и мае наблюдается рост количестве заказов и/или в увеличении ассортимента заказываемых товаров в заказе. С июня по январь количество заказов и/или количество наименований товаров стабильное.

## 3.13 Очистка датасета от выбросов

Перед удаление от выбросов посмотрим описательную статистику датафрейма `dfc`

In [45]:

```
dfc.describe()
```

Out[45]:

| | order_id | quantity | price | total |
|---|---|---|---|---|
| **count** | 5522.000000 | 5522.000000 | 5522.000000 | 5522.000000 |
| **mean** | 55927.056501 | 2.577508 | 531.584224 | 831.172839 |
| **std** | 32502.054146 | 16.506591 | 975.262577 | 9224.346313 |
| **min** | 12624.000000 | 1.000000 | 9.000000 | 9.000000 |
| **25%** | 14808.000000 | 1.000000 | 90.000000 | 120.000000 |
| **50%** | 69188.500000 | 1.000000 | 150.000000 | 190.000000 |
| **75%** | 71940.500000 | 1.000000 | 524.000000 | 734.000000 |
| **max** | 112789.000000 | 1000.000000 | 14917.000000 | 675000.000000 |

In [46]:

```
dfc['order_id'].nunique()
```

Out[46]:

3492

У нас 5522 заказов, в заказе в среднем 2,57 единицы товарной позиции, стандартное отклонение 16,5, с минимальным количеством 1 и максимальным 1000 штук, с ценой от 9 до 14917 у.е. и общей суммой за товарную номенклатуру от 9 до 675000 у.е. со стандартным отклонением 9224,34 у.е.

В предобработке мы выявили заказ по количеству штук товарной номенклатуры в заказе, который требуется удалить. Это заказ `71743` с товаром `Вантуз с деревянной ручкой d14 см красный, Burstenmann, 0522/0000` - 1000 шт. Перезапишем датасет без этого и далее посмотрим как изменится статистика.

In [47]:

```
dfc = dfc.query('order_id != 71743')
dfc.describe()
```

Out[47]:

| | order_id | quantity | price | total |
|---|---|---|---|---|
| **count** | 5521.000000 | 5521.000000 | 5521.000000 | 5521.000000 |
| **mean** | 55924.191813 | 2.396848 | 531.558247 | 709.062926 |
| **std** | 32504.300844 | 9.605078 | 975.349002 | 1658.751313 |
| **min** | 12624.000000 | 1.000000 | 9.000000 | 9.000000 |
| **25%** | 14808.000000 | 1.000000 | 90.000000 | 120.000000 |
| **50%** | 69188.000000 | 1.000000 | 150.000000 | 190.000000 |
| **75%** | 71941.000000 | 1.000000 | 524.000000 | 734.000000 |
| **max** | 112789.000000 | 334.000000 | 14917.000000 | 49432.000000 |

У нас 5521 строк заказов, что логично, удалили один. Теперь в заказе в среднем 2,39 единицы товарной позиции и стандартным отклонением 9.61 - максимум штук товарной позиции в строке заказа теперь 334. Показатели цены не изменились, а вот статистика по общей сумме изменилась. Средняя цена снизилась с 831 до 709, стандартное отклонение с 9224 снизилось до 1658 и максимальная общая сумма снизилась более чем в 13 раз от 675000 до 49432 у.е.

#### 3.13.1 Итоги очистки от выбросов.

Удаление одного заказа сильно повлияло на описательную статистику. Приступим к анализу данных.

# 4 Анализ данных. Сегментация покупателей

## 4.1 Цель сегментации

Для анализа и оптимизации маркетинговых стратегий на основе потребительского поведения необходимо провести сегментацию покупателей.
Необходимо оценить ценность клиентов и классифицировать их с учетом трех ключевых параметров: активности , частоты покупок и суммы потраченных денег.
Анализируя эти аспекты, можно более эффективно настраивать маркетинговые кампании, улучшать обслуживание клиентов и оптимизировать процессы удержания клиентов.

## 4.2 Выбор метода сегментации покупателей

Выбрана методология проведения сегментации - RFM-анализ.
RFM-анализ (от англ. Recency, Frequency, Monetary ) позволяет оценить ценность клиентов и классифицировать их с учетом трех ключевых параметров.

## 4.3 Подготовка к RFM-сегментации

Построим RFM-сегментацию покупателей, чтобы качественно оценить аудиторию. В кластеризации выберем следующие метрики:

- r - recency - время от последней покупки пользователя до текущей даты,
- f - frequency- суммарное количество покупок у пользователя за всё время,
- m - monetary - сумма покупок за всё время.

Для каждого RFM- сегмента построим границы метрик recency, frequency и monetary для интерпретации этих кластеров.

Присвоим каждому покупателю степень 1, 2, 3, 4, 5 в зависимости от общей суммы заказов за весь период, частоты заказов и насколько недавно была совершена последняя покупка.

Для начала определимся с периодом для анализа.

#### 4.3.1 Период для RFM-анализа

Для начала посмотрим за какой период предоставлены данные. Найдем максимальную и минимальные даты.

In [48]:

```
print(f"Мы располагаем данными за период с {df['date'].min()} по {df['date'].max()}.\
\nМаксимальная дата: {df['date'].min()}.\nМинимальная дата:  {df['date'].max()}")
```

Мы располагаем данными за период с 2018-10-01 00:00:00 по 2020-01-31 15:00:00.

Максимальная дата: 2018-10-01 00:00:00.

Минимальная дата:  2020-01-31 15:00:00

Чтобы определить период для RFM-анализа - посмотрим на кол-во пользователей и заказов по месяцам. Сохраним в `by_month` сгруппированный датафрейм по месяцу и году и визуализируем на графике получившийся результат.

In [49]:

```
by_month = dfc.groupby('year_month', as_index=False).agg({'customer_id':'count',
'order_id':'nunique'})
by_month.style.background_gradient(low=0.75, high=1.0)
```

Out[49]:

| | year_month | customer_id | order_id |
|---|---|---|---|
| 0 | 2018-10 | 482 | 238 |
| 1 | 2018-11 | 428 | 204 |
| 2 | 2018-12 | 306 | 278 |
| 3 | 2019-01 | 182 | 165 |
| 4 | 2019-02 | 359 | 286 |
| 5 | 2019-03 | 406 | 242 |
| 6 | 2019-04 | 673 | 264 |
| 7 | 2019-05 | 726 | 178 |
| 8 | 2019-06 | 282 | 156 |
| 9 | 2019-07 | 312 | 198 |
| 10 | 2019-08 | 199 | 172 |
| 11 | 2019-09 | 217 | 179 |
| 12 | 2019-10 | 217 | 201 |
| 13 | 2019-11 | 217 | 217 |
| 14 | 2019-12 | 246 | 246 |
| 15 | 2020-01 | 269 | 269 |

In [50]:

```
# визуализируем количество покупателей и заказов по месяцам
plt.figure(figsize=(10,6))
plt.xticks(np.arange(len(by_month)), by_month['year_month'], rotation=90) #,
alpha=0.4)
sns.lineplot(x='year_month', y ='customer_id', data=by_month, color='green',
label="Покупатели")
sns.lineplot(x='year_month', y ='order_id', data=by_month, color='red',
label="Заказы")
plt.title("Заказы и покупатели по месяцам", fontsize = 14, fontweight ='bold')
plt.xlabel('')
plt.ylabel('Количество, шт.', fontsize = 12);
```


В наших данных кол-во покупателей и заказов неравномерно. Для анализа RFM обычно используется год. Так как данных не много, не будем ограничивать интервал исследования и используем весь период из датасета с октября 2018 по январь 2020 гг.

Заметно, что с ноября 2019 идет рост покупателей и заказов и их количество одинаковое. С одной стороны рост, а с другой стороны нет повторных заказов от покупателей.

#### 4.3.2 Дата для RFM-анализа

Для расчета времени с последней покупки(r) нам нужна назначить дату анализа. Смоделируем проведение RFM-анализа на дату сразу после окончания выбранного нами периода периода и запишем ее в переменной `rfm_date`.

In [51]:

```
rfm_date = dfc['date'].max() + timedelta(days=1)
print('Дату анализа:', rfm_date)
```

Дату анализа: 2020-02-01 15:00:00

#### 4.3.3 Расчет показателей recency, frequency, monetary

Сгруппируем данные из таблицы `dfc` по покупателю `customer_id` в новый датафрейм `rfm` и посчитаем значения r, f, m. Для этого произведём агрегацию по столбцам `'date', 'order_id' и 'total'`.

Расчет `recency`:

- Чтобы найти срок в днях с последней покупки мы применим агрегатную функцию и найдем максимальное значение в столбце `date` для каждого покупателя - это дата которая ближе всего к дате анализа. Переименуем столбец в `recency` и уже после, в получившемся датафрейме `rfm`, перезапишем данные в этот же столбец с информацией о разнице в днях. Разницу в днях получим между минимальной датой -последней покупкой и датой анализа из переменной `rfm_date`. Это и будет значением `recency`.

Расчет `frequency`:

- Посчитаем количество уникальных заказов по колонке `order_id`, применив агрегатную функцию `nunique`.

Расчет `monetary`:

- применим агрегатную функцию к столбцу `total` sum и получим общую сумму по всем заказам покупателя.
- Сгруппируем данные в новый датафрейм `rfm` и посчитаем значения r, f, m.

Переименуем колонки в получившемся датафрейме. И выведем несколько случайных строк, чтобы убедиться, что все получилось.

В датафрейме `rfm` столбцы:

- `customer_id` уникальный номер id покупателя,
- `recency` указаны дни с даты последнего заказа по текущую дату проведения анализа,
- `frequency` - общее число уникальных заказов покупателя за весь период.
- `monetary` - общая сумма по покупателю за все заказы.

In [52]:

```
rfm = dfc.groupby('customer_id', as_index=False)  \
            .agg({'date':'max',  \
                  'order_id': 'nunique',  \
                 'total':'sum'})
rfm.rename(columns={'date': 'recency', 'order_id': 'frequency', 'total':'monetary'}, inplace=True)
rfm['recency'] = rfm['recency'].apply(lambda x: (rfm_date - x).days)
rfm.sample(n=3, random_state=6)
```

Out[52]:

| | customer_id | recency | frequency | monetary |
|---|---|---|---|---|
| **1058** | 71a39144-2a79-43f3-aebf-0a2ffb840fee | 38 | 2 | 2811.0 |
| **1154** | 7c07ced3-b809-4a37-8650-e79f9f1a4ea5 | 197 | 1 | 598.0 |
| **1459** | 9cf0b385-7af0-49ce-971e-48711a9742e9 | 235 | 1 | 824.0 |

Проверим, правильно ли агрегированы данные. Для этого по трем случайным покупателям, выведенных с помощью sample(), посмотрим данные в датафрейме `dfc` и сравним с полученными агрегированными значениями

In [53]:

```
#проверим данные и найдем в датафрейме `dfc` покупателей и сравним.
display(dfc.query('customer_id == "71a39144-2a79-43f3-aebf-0a2ffb840fee"'))
display(dfc.query('customer_id == "7c07ced3-b809-4a37-8650-e79f9f1a4ea5"'))
dfc.query('customer_id == "9cf0b385-7af0-49ce-971e-48711a9742e9"')
```

| | date | customer_id | order_id | product | quantity | price | total | year_month |
|---|---|---|---|---|---|---|---|---|
| **2478** | 2019-03-27 10:00:00 | 71a39144-2a79-43f3-aebf-0a2ffb840fee | 70895 | Швабра для мытья окон Leifheit Comfort с телескоп. Ручкой, 51010 | 1 | 2624.0 | 2624.0 | 2019-03 |
| **7133** | 2019-12-25 13:00:00 | 71a39144-2a79-43f3-aebf-0a2ffb840fee | 108334 | Цветущее комнатное растение Бегония Элатиор, цвет в ассортименте | 1 | 187.0 | 187.0 | 2019-12 |

| | date | customer_id | order_id | product | quantity | price | total | year_month |
|---|---|---|---|---|---|---|---|---|
| **5891** | 2019-07-18 18:00:00 | 7c07ced3-b809-4a37-8650-e79f9f1a4ea5 | 72032 | Прищепки для белья York Spring Prestige, 9603/Z027 | 2 | 299.0 | 598.0 | 2019-07 |

Out[53]:

| | date | customer_id | order_id | product | quantity | price | total | year_month |
|---|---|---|---|---|---|---|---|---|
| **5088** | 2019-06-11 10:00:00 | 9cf0b385-7af0-49ce-971e-48711a9742e9 | 71673 | Вешалка гардеробная Радуга 1 ЗМИ белое серебро ВНП 298 бс | 1 | 824.0 | 824.0 | 2019-06 |

По трем случайным покупателям показатели `recency, frequency и monetary` соответствуют данным, на основании которых и произведен расчет.

## 4.4 Подготовка к RFM-сегментации: recency

### 4.4.1 Описательная статистика recency

Прежде чем визуализировать recency, посмотрим как распределены покупатели по давности совершения последней покупки. Для этого посмотрим описание. Сгруппируем датасет `rfm` по уникальному id покупателя и посчитаем дни `recency`.

In [54]:

```
rfm.groupby('customer_id')['recency'].sum().describe()
```

Out[54]:

```
count    2412.000000
mean      216.177032
std       149.398106
min         1.000000
25%        73.000000
50%       208.500000
75%       344.000000
max       488.000000
Name: recency, dtype: float64
```

По нашим данным последняя покупка совершена 1 день назад, а самая давняя покупка 488 дней назад. Медиана по всем покупкам это 208 дней.

### 4.4.2 Гистограмма recency

Визуализируем recency на гистограмме "Давность покупок в днях", то есть, сколько дней прошло с момента последней покупки у покупателя.

In [55]:

```
# визуализация доли покупателей по давности
fig = px.histogram(rfm, # датасет
                   x="recency", # выбор столбца
                   #histnorm='percent', # в процентах ось y
                   nbins=16, #регулируем количество bins
                   #title='Давность покупок в днях', # название
                   opacity=0.9, # прозрачность
                   template='plotly_white', #цвет подложки
                   color_discrete_sequence= ['#79CDCD'],# ['goldenrod'],
#['indianred'], ['green'] # color of histogram bars
                   width=750, height=330  #размер графика
                  )
fig.update_layout(title_text = 'Распределение покупателей по тому<br>\
сколько прошло дней с последней покупки,', title_x=0.5)  # название
fig.update_layout(
   xaxis={'title':'Прошло дней с покупки'},
   yaxis={'title':'Покупатели, количество'})  # название осей
fig.update_traces(marker_line_color= '#F0F8FF', #'rgb(18,98,107)',
                  marker_line_width=2, opacity=0.8) #добавление обводки
fig.show()

fig = px.histogram(rfm, # датасет
                   x="recency", # выбор столбца
                   histnorm='percent', # в процентах ось y
                   nbins=16, #регулируем количество bins
```

```
                    #title='Давность покупок в днях', # название
                    opacity=0.9, # прозрачность
                    template='plotly_white', #цвет подложки
                    color_discrete_sequence= ['#79CDCD'],# ['goldenrod'],
#['indianred'], ['green'] # color of histogram bars
                    width=750, height=330  #размер графика
                   )
fig.update_layout(title_text = 'Распределение покупателей по тому<br>\
сколько прошло дней с последней покупки, %', title_x=0.5)  # название
fig.update_layout(
   xaxis={'title':'Прошло дней с покупки'},
   yaxis={'title':'Покупатели, %'})  # название осей
fig.update_traces(marker_line_color= '#F0F8FF', #'rgb(18,98,107)',
                  marker_line_width=2, opacity=0.8) #добавление обводки
fig.show()
```



Покупателей, которые совершали покупку в течении 50 последних дней почти 17,5 %, в течении 50-100 дней 14,5 %, 100-488 дней назад колеблется от 6,8 % до 10.5%.

Можно сделать вывод, что за период 100 дней произошел рост покупателей, совершивших покупку.

## 4.5 Подготовка к RFM-сегментации: frequency

### 4.5.1 Описательная статистика frequency

Прежде чем визуализировать frequency, посмотрим как распределено количество покупок на одного покупателя Для этого посмотрим описание. Сгруппируем датасет `rfm` по уникальному id покупателя и посчитаем покупки `frequency`.

In [56]:

```
rfm.groupby('customer_id')['frequency'].sum().describe()
```

Out[56]:

```
count   2412.000000
mean       1.447347
std        2.698627
min        1.000000
25%        1.000000
50%        1.000000
75%        2.000000
max      126.000000
Name: frequency, dtype: float64
```

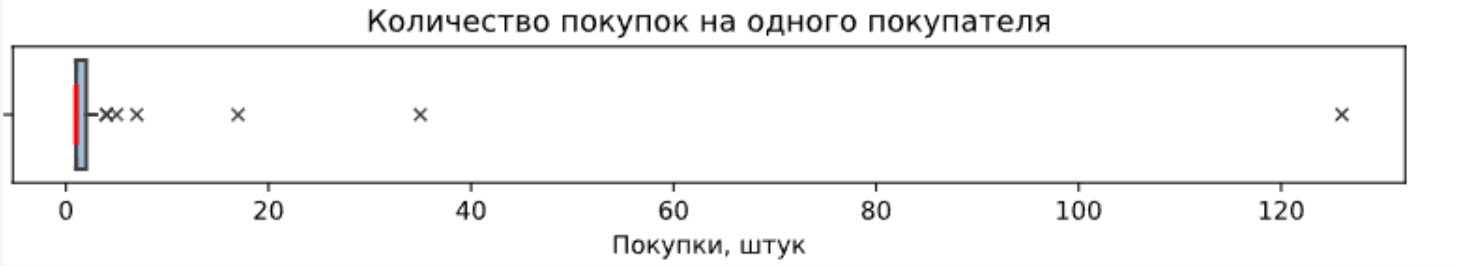
С помощью метода describe() мы получили данные описательной статистики. Видно очень большой разброс по количеству покупок на одного покупателя.

Построим боксплот и посмотрим как выглядит количество покупок на одного покупателя на боксплоте.

### 4.5.2 Боксплот frequency

In [57]:

```
plt.figure(figsize=(10,1))

sns.boxplot(x=rfm['frequency'], notch=True, showcaps=False,
   flierprops={"marker": "x"},
   boxprops={"facecolor": (.3, .5, .7, .5)},
   medianprops={"color": "r", "linewidth": 2})


plt.xlabel('Покупки, штук')
plt.ylabel('')
plt.title('Количество покупок на одного покупателя');
#plt.xlim(-1, 126);
```

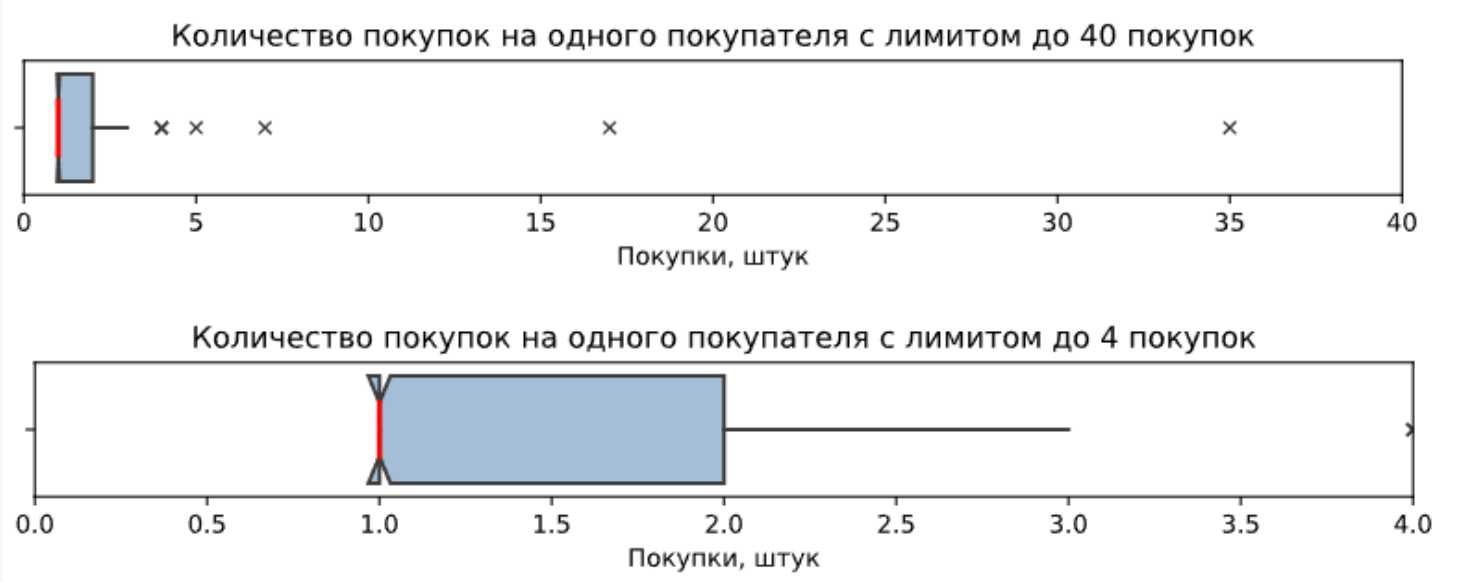



Большой разброс в значениях выбросов мешают рассмотреть остальные значения. Поэтому слегка ограничим боксплот по количеству совершенных покупок до 40 и затем построим еще один с ограничение в 4 покупки, чтобы рассмотреть все поближе значения.

In [58]:

```
plt.figure(figsize=(10,1))

sns.boxplot(x=rfm['frequency'], notch=True, showcaps=False,
   flierprops={"marker": "x"},
   boxprops={"facecolor": (.3, .5, .7, .5)},
   medianprops={"color": "r", "linewidth": 2})
```

```
plt.xlabel('Покупки, штук')
plt.ylabel('')
plt.title('Количество покупок на одного покупателя с лимитом до 40 покупок');
plt.xlim(0, 40);

plt.figure(figsize=(10,1))

sns.boxplot(x=rfm['frequency'], notch=True, showcaps=False,
   flierprops={"marker": "x"},
   boxprops={"facecolor": (.3, .5, .7, .5)},
   medianprops={"color": "r", "linewidth": 2})


plt.xlabel('Покупки, штук')
plt.ylabel('')
plt.title('Количество покупок на одного покупателя с лимитом до 4 покупок');
plt.xlim(0, 4);
```




Медиана по количеству покупок лежит на позиции 1, третий квартиль на двух покупках. Максимальное значение до выбросов - три. Все покупки больше трех нетипичны для нашего интернет-магазина. Из описательной статистики мы получили данные, что максимальное количество покупок 126.

### 4.5.3 Гистограмма frequency

Визуализируем frequency на гистограмме "Частота покупок", то есть, сколько покупок приходится на одного покупателя за все время с выбросами.

In [59]:

```
# визуализация сколько покупок приходится на одного покупателя за все время.
fig = px.histogram(rfm, # датасет
                   x="frequency", # выбор столбца
                   histnorm='percent', # в процентах ось y
                   nbins=16, #регулируем количество bins
                   #title='Давность покупок в днях', # название
                   opacity=0.8, # прозрачность
                   template='plotly_white', #цвет подложки
                   width=450, height=350,  #размер графика
                   color_discrete_sequence= ['indianred'],# ['goldenrod'],
#['indianred'], ['green'] # color of histogram bars
                  )
```

```
fig.update_layout(title_text = 'Частота заказов', title_x=0.5)  # название
fig.update_layout(
   xaxis={'title':'Количество покупок на одного покупателя по всем данным'},
   yaxis={'title':' %'})  # название осей
fig.update_traces(marker_line_color= '#636e72', #'rgb(18,98,107)',
                  marker_line_width=2, opacity=0.8) #добавление обводки
fig.show()
```

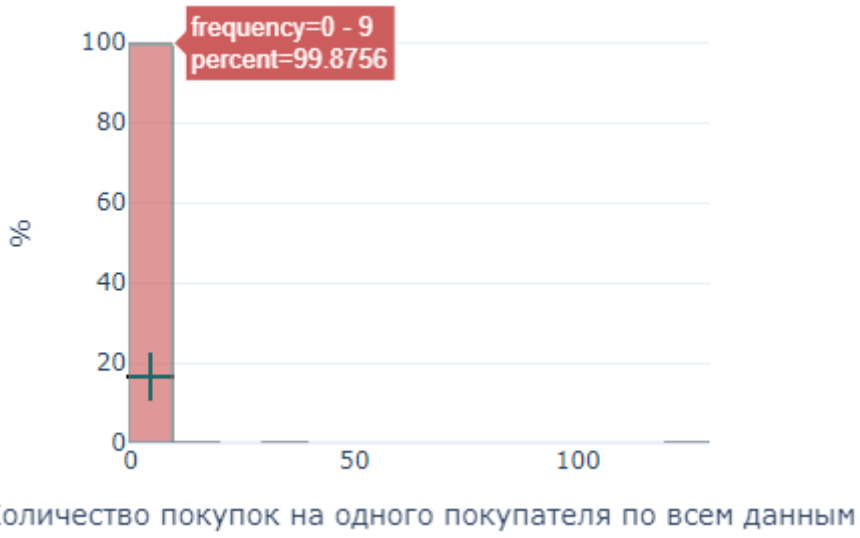



График получился неудобным. Видим, что большинство покупок почти 100% до 9 штук. Посмотрим как распределены покупки с учетом очистки от покупок оптовиков. Сделаем срез и посмотрим на гистограмму еще раз по тем покупателям, которые совершили не более трех покупок.

In [60]:

```
# визуализация сколько покупок приходится на одного покупателя за все время не более 9 покупок
fig = px.histogram(rfm.query('frequency < 4'), # датасет
                   x="frequency", # выбор столбца
                   histnorm='percent', # в процентах ось y
                   nbins=16, #регулируем количество bins
                   #title='Давность покупок в днях', # название
                   opacity=0.8, # прозрачность
                   template='plotly_white', #цвет подложки
                   width=450, height=350,  #размер графика
                   color_discrete_sequence=['indianred'],# ['goldenrod'],
#['indianred'], ['green'] # color of histogram bars
                  )
fig.update_layout(title_text = 'Частота заказов<br>без оптовиков', title_x=0.5)  # название
fig.update_layout(
   xaxis={'title':'Количество покупок на одного покупателя'},
   yaxis={'title':' %'})  # название осей
fig.update_traces(marker_line_color= '#636e72', #'rgb(18,98,107)',
                  marker_line_width=2, opacity=0.8) #добавление обводки
fig.show()
```


По отфильтрованной гистограмме Частота покупок не более 3 покупок, видим, что 63,6 % покупателей совершили не более одной покупки и 35,5% не более двух покупок, 3 покупки 0,87%.

#### 4.5.4 Вывод по frequency

По нетипичному поведению некоторых покупателей по частоте покупок можно предположить, что скорее всего среди обычных типичных покупателей есть оптовики. При сегментировании покупателей следует учесть, что покупателей, купивших более трех раз можно считать оптовиками.

## 4.6 Подготовка к RFM-сегментации: monetary

#### 4.6.1 Описательная статистика monetary

Прежде чем визуализировать, посмотрим как распределена общая сумма покупок по каждому покупателю (LTV). Для этого посмотрим описание. Сгруппируем датасет `rfm` по уникальному id покупателя и суммируем общую сумму monetary.

In [61]:

```
rfm.groupby('customer_id')['monetary'].sum().describe()
```

Out[61]:

```
count     2412.000000
mean      1623.025048
std       4341.333501
min         15.000000
25%        389.000000
50%        837.000000
75%       1798.250000
max     159508.000000
Name: monetary, dtype: float64
```

С помощью метода describe() мы получили данные описательной статистики.Видно очень большой разброс сумм по покупателям - минимальное и максимальное значения от 15 у.е. до 159508 у.е.
Построим боксплот и посмотрим как выглядит общая сумма покупок по каждому покупателю (LTV покупателя).

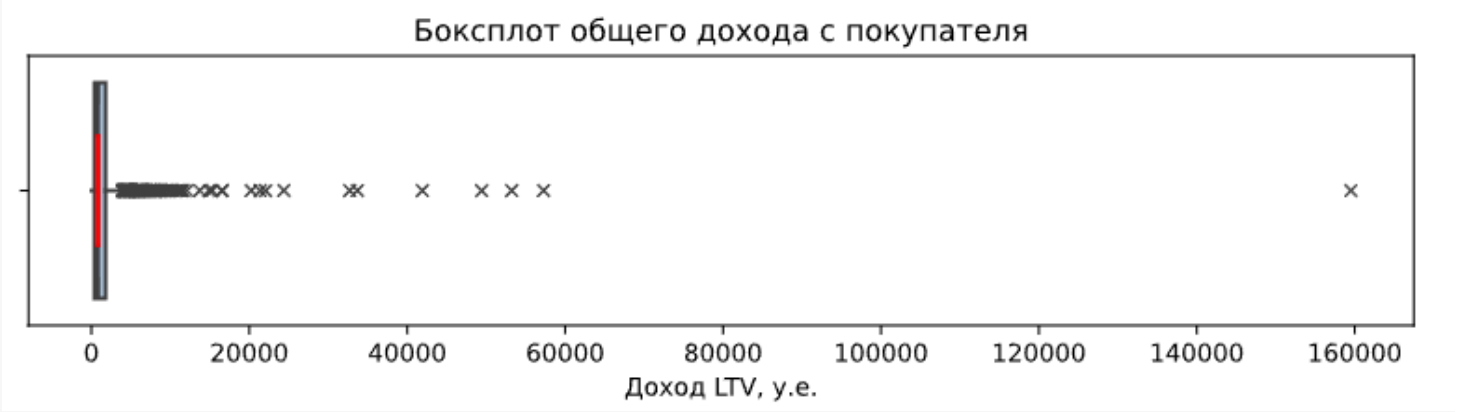
#### 4.6.2 Боксплот monetary

Построим боксплот и посмотрим как выглядит общая сумма покупок по каждому покупателю (LTV покупателя).

In [62]:

```
plt.figure(figsize=(10, 2))

sns.boxplot(x=rfm['monetary'], notch=True, showcaps=False,
   flierprops={"marker": "x"},
   boxprops={"facecolor": (.3, .5, .7, .5)},
   medianprops={"color": "r", "linewidth": 2})
plt.xlabel('Доход LTV, у.е.')
```

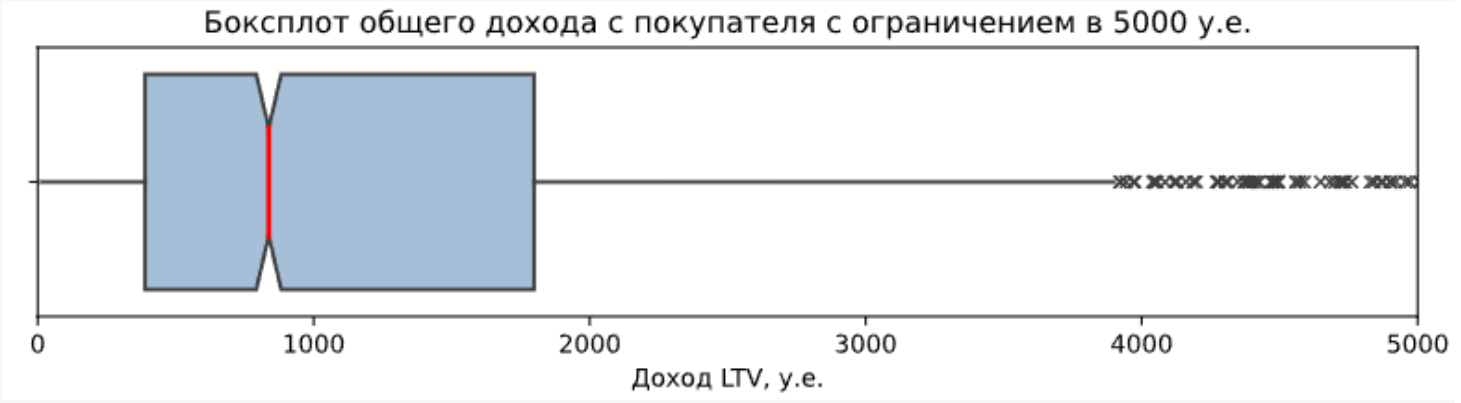
```
plt.ylabel('')
plt.title('Боксплот общего дохода с покупателя');
```




Очень трудно что-то рассмотреть, кроме нетипичных значений. разброс которых от нескольких тысяч до почти 160000 у.е. Как мы выяснили ранее у нас есть нетипичные покупатели, совершающие покупки больше трех раз. Можно предположить, что также есть нетипичные покупатели, покупающие дорогой товар по несколько штук или покупающие много раз и дающие нетипично большую общую сумму заказов в итоге. Ограничим график лимитом в 5000 у.е. И посмотрим, что получилось.

In [63]:

```
plt.figure(figsize=(10,2))

sns.boxplot(x=rfm['monetary'], notch=True, showcaps=False,
   flierprops={"marker": "x"},
   boxprops={"facecolor": (.3, .5, .7, .5)},
   medianprops={"color": "r", "linewidth": 2})


plt.xlabel('Доход LTV, у.е.')
plt.ylabel('')
plt.title('Боксплот общего дохода с покупателя с ограничением в 5000 у.е.')
plt.xlim(0, 5000);
```




Теперь видно, что LTV покупателя свыше 4000 - это нетипичные покупки. Будем считать таких покупателей оптовиками. Мы не можем отбрасывать из исследования тех, кто приносит много денег.

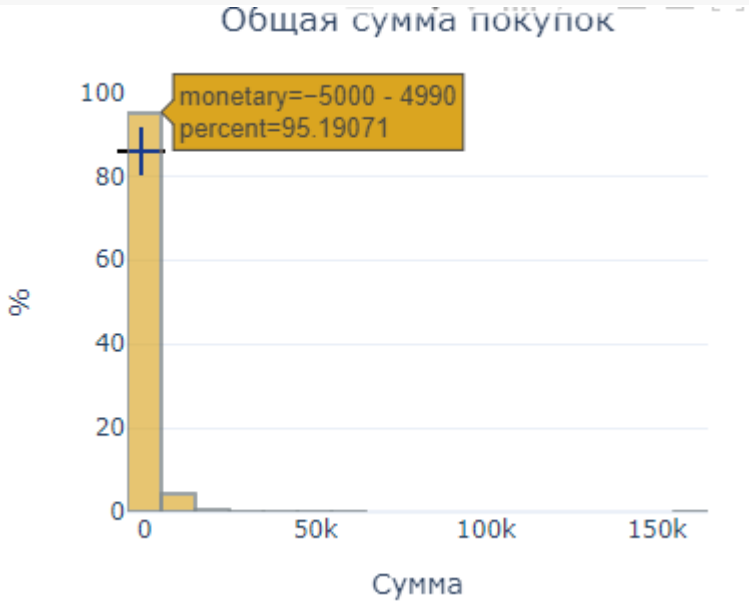
### 4.6.3 Гистограмма monetary

Визуализируем monetary на гистограмме "Общая сумма покупок" и посмотрим как распределена общая сумма покупок по каждому покупателю (LTV) с выбросами.

In [64]:

```
# общую сумму покупок по каждому покупателю (LTV покупателя)
fig = px.histogram(rfm, # датасет
                   x="monetary", # выбор столбца
                   histnorm='percent', # в процентах ось y
```

```
                    nbins=16, #регулируем количество bins
                    #title='Давность покупок в днях', # название
                    opacity=0.8, # прозрачность
                    template='plotly_white', #цвет подложки
                    color_discrete_sequence=['goldenrod'],# ['goldenrod'],
#['indianred'], ['green'] # color of histogram bars
                    width=450, height=350  #размер графика
                   )
fig.update_layout(title_text = 'Общая сумма покупок', title_x=0.5)  # название
fig.update_layout(
   xaxis={'title':'Сумма'},
   yaxis={'title':' %'})  # название осей
fig.update_traces(marker_line_color= '#636e72', #'rgb(18,98,107)',
                  marker_line_width=2, opacity=0.8) #добавление обводки
fig.show()
```

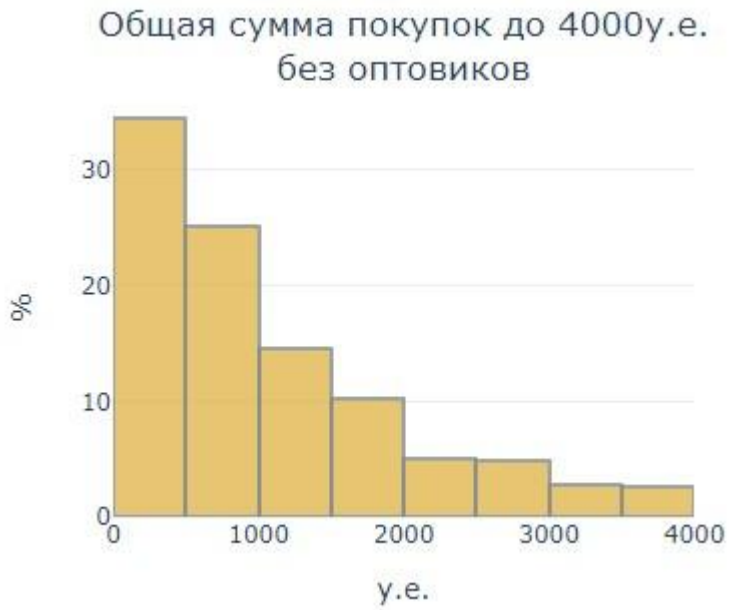



И тут мы убедились, что с выбросами мало что можно увидеть и понять. Видим, что разброс в общей сумме покупок на каждого покупателя очень большой. от нуля и около 160тыс.у.е. Точнее на графике трудно рассмотреть. Ранее по боксплоту мы точнее увидели сумму, а именно - превышение 4000 у.е. точно можно считать выбросами. Поэтому отсечем то, что превышает 4тыс. у.е. и посмотрим на гистограмму еще раз без выбросов.

In [65]:

```
# вобщую сумму покупок по каждому покупателю (LTV по общему доходу) без выбросов
fig = px.histogram(rfm.query('monetary < 4000'), # датасет
                    x="monetary", # выбор столбца
                    histnorm='percent', # в процентах ось у
                    nbins=16, #регулируем количество bins
                    #title='Давность покупок в днях', # название
                    opacity=0.8, # прозрачность
                    template='plotly_white', #цвет подложки
                    color_discrete_sequence=['goldenrod'],# ['goldenrod'],
#['indianred'], ['green'] # color of histogram bars
                    width=450, height=350  #размер графика
                   )
fig.update_layout(title_text = 'Общая сумма покупок до 4000у.е.<br>без оптовиков',
title_x=0.5)  # название
fig.update_layout(
```

```
   xaxis={'title':'y.e.'},
   yaxis={'title':' %'})  # название осей
fig.update_traces(marker_line_color= '#636e72', #'rgb(18,98,107)',
                  marker_line_width=2, opacity=0.8) #добавление обводки
fig.show()
```




Теперь видим плавное распределение LTV в сторону уменьшения количества покупателей при увеличении общей суммы покупки. Покупателей, каждый из которых потратил:

- до 500 у.е. 34,6% от общего числа покупателей,
- от 500 до 1000 у.е. 25.2% от общего числа покупателей,
- от 1000 до 1500 у.е. 14.6% от общего числа покупателей,
- от 1500 до 2000 у.е. 10.2% от общего числа покупателей,
- от 2000 до 2500 у.е. 5% от общего числа покупателей,
- от 2500 до 3000 у.е. 4.8% от общего числа покупателей,
- от 3000 до 3500 у.е. 2.85% от общего числа покупателей,
- от 3500 до 4000 у.е. 2.75% от общего числа покупателей,

## 4.7 Описательная статистика monetary, frequency, recency.

Мы выяснили, что у нас есть около 1% покупателей, которые сильно выбиваются из показателей всех остальных 99% покупателей по частоте совершаемых покупок и общей сумме этих покупок.

По боксплоту и с помощью описательной статистики выявлено, что покупатели, совершившие более трех покупок на общую сумму свыше 4000 у.е. нетипичны. Решено считать таких покупателей оптовиками. Ранее мы смотрели на цены товара. У нас основной товар стоимостью до 200 у.е. и далее с повышением цены на товар ассортимент очень сильно снижается. Но все же есть товар стоимостью и в 15тыс. у.е. С таким разбросом цен вполне возможен разброс по общей стоимости товара, если покупатель купил много дорогого (в рамках нашего интернет-магазина) товара. Посмотрим описательную статистику по всем данным, т.е. с этими выбросами:

In [66]:

```
# описательная статистика с выбросами
rfm.describe()
```

Out[66]:

| | recency | frequency | monetary |
|---|---|---|---|
| **count** | 2412.000000 | 2412.000000 | 2412.000000 |
| **mean** | 216.177032 | 1.447347 | 1623.025048 |
| **std** | 149.398106 | 2.698627 | 4341.333501 |
| **min** | 1.000000 | 1.000000 | 15.000000 |
| **25%** | 73.000000 | 1.000000 | 389.000000 |

| | | | |
|---|---|---|---|
| **50%** | 208.500000 | 1.000000 | 837.000000 |
| **75%** | 344.000000 | 2.000000 | 1798.250000 |
| **max** | 488.000000 | 126.000000 | 159508.000000 |

monetary:

- Средний чек по покупателю 1623 у.е по неочищенным от выбросов данным. Показатели monetary первый и третий квартиль 389 и 1798 у.е. соответственно, второй квартиль- медиана 837 у.е. Стандартное отклонение 4341.

frequency:

- Показатель frequency в среднем 1,45. Минимум, первый и второй квартиль равны 1, третий квартиль равен двум, а максимальная величина 126. Она то и повлияла на размер стандартного отклонения, который равен 2,7. Учитывая, что третий квартиль это 2, то становится понятно как сильно искажаются данные таким выбросом в 126 покупок. Следует учесть наличие оптовиков при делении покупателей на категории

recency:

- Показатель recency и на гистограмме и в описательной статистике без особых сюрпризов. Минимальные и максимальные значения отражают даты совершения покупок. Было принято решение не ограничивать период наблюдений. Очистка от выбросов не требуется.

# 5 Покупатели - разбивка на сегменты

Разбивку покупателей на сегменты произведем в датасете под названием `rfm_score`. Посмотрим как quantile разобьет данные по покупателям с шагом в 20%

In [67]:

```
rfm_score = rfm

# деление на 5 категорий - по 20%
print(rfm_score.quantile(q=[0.2,0.4,0.6,0.8]))
```

```
   recency  frequency  monetary
0.2    59.0        1.0     300.0
0.4   141.0        1.0     638.0
0.6   268.0        1.0    1087.0
0.8   372.0        2.0    2107.0
```

Смотрим как произошла разбивка с помощью quantile на пять категорий по значениям это поможет нам оценить покупателей по 5 бальной шкале по каждому показателю.

## 5.1 Критерии оценки

Присвоим критерии для оценки

- recency_scores = 5, 4, 3, 2, 1 -recency. Чем ниже значение, тем выше оценка.
- frequency_scores = 1, 2, 3, 4, 5 - Чем ниже значение, тем ниже оценка и наоборот frequency
- monetary_scores = 1, 2, 3, 4, 5 - Чем ниже значение, тем ниже оценка и наоборот monetary

Интерпретация У показателя **recency** (дней с последней покупки) - оценка присваивается по такой шкале,

- 1 - те кто купил 372 дня назад и позже,
- 2 - те кто купил от в диапазоне 268-372 дней назад,
- 3 - те кто купил в диапазоне 141-268 дней назад,
- 4 - те кто купил в диапазоне 59-141 дней и
- 5 - те кто купил в период 1-59 дней назад. В этом случае нужно не забывать, что чем выше цифра дней, тем ниже балл нужно присвоить для покупателя.

У показателя **frequency** (количество покупок) значение 1 у трех из четырех диапазонов, Это значит, что большинство покупателей купили по одному разу. Это учтено это при присвоении оценок - помним про оптовых покупателей

- 1- те кто купил 1 раз
- 2- те кто купил 2 раза

- 3- те кто купил 3 раза
- 4- те кто купил 4 раза
- 5- кто купил более 4-х раз

У показателя **monetary** (общая сумма которую принес каждый покупатель) покупателей можно поделить на 5 категории

- 1- те кто купил до 300 у.е,
- 2- те кто купил от 300 у.е. до 638 у.е.,
- 3- те кто купил от 638 до 1087 у.е.,
- 4- те кто купил в диапазоне от 1087 до 2107 у.е.
- 5- те, кто купил свыше 2107 у.е. В этом случае нужно не забывать, что чем выше цифра показателя monetary, тем выше балл нужно присвоить для покупателя.

Значения баллов оценки запишем в соответствующие столбцы `r_score` `f_score` `m_score`.

In [68]:

```
# критерии оценки для каждого значения RFM
recency_scores = [5, 4, 3, 2, 1]  # recency. Чем ниже значение, тем выше оценка.
frequency_scores = [1, 2, 3, 4, 5]  # Чем ниже значение, тем ниже оценка и наоборот frequency
monetary_scores = [1, 2, 3, 4, 5]  # Чем ниже значение, тем ниже оценка и наоборот monetary
```

## 5.2 Присваиваем оценку recency покупателю

Используем функцию pandas.cut() модуля pandas. Значение bins зададим как 5. Ранее мы смотрели как будут распределены диапазоны отсекающие по 20% пользователей, используем эти диапазоны.
Чем меньшее значение недавности покупки, т.е. количество дней с даты последней покупки и датой анализа, тем лучше.

А значит, чем больше значение recency, тем меньше ранг, чем меньше значение, тем выше ранг.

- 1 - те кто купил 372 дня назад и позже,
- 2 - те кто купил от в диапазоне 268-372 дней назад,
- 3 - те кто купил в диапазоне 141-268 дней назад,
- 4 - те кто купил в диапазоне 59-141 дней и
- 5 - те кто купил в период 1-59 дней назад. В этом случае нужно не забывать, что чем выше цифра дней, тем ниже балл нужно присвоить для покупателя.

In [69]:

```
# recency
intervals_r=[0, 59, 141, 268, 372, 489]
rfm_score['r_score']=pd.cut(rfm_score.recency, intervals_r, labels=recency_scores)
rfm_score.sample(n=5, random_state=745)
```

Out[69]:

| | customer_id | recency | frequency | monetary | r_score |
|---|---|---|---|---|---|
| **1387** | 95919727-3866-421f-8297-4f45348c3f33 | 375 | 1 | 4272.0 | 1 |
| **2196** | eb6521ae-56e3-4a72-9ea2-e9c69701ff3f | 298 | 3 | 2903.0 | 2 |
| **2154** | e6892eec-7b21-4a49-9e56-29268f9eb98a | 270 | 1 | 134.0 | 2 |
| **1602** | add6d8bd-36ca-4c46-8a0b-7f13bbd05555 | 60 | 2 | 658.0 | 4 |
| **1123** | 78b5ab9e-999b-4825-bc44-19958d4854c1 | 89 | 2 | 2410.0 | 4 |

## 5.3 Присваиваем оценку frequency покупателю

Используем функцию pandas.cut() модуля pandas. Значение bins зададим как 5. Ранее мы смотрели как будут распределены диапазоны отсекающие по 20% пользователей и выяснили, что такое распределение в нашем случае не подходит.

Используем список frequency_scores - где сохранены критерии оценки для каждого значения RFM в label. Зададим интервал, который поделит данные - intervals_f - данный интервал мы определили явно задав границы, исходя их того, что заказов от 1 до 126 и большинство это 1.

In [70]:

```
# frequency
intervals_f=[0, 1, 2, 3, 4, 127]
rfm_score['f_score']=pd.cut(rfm_score.frequency, intervals_f, labels=frequency_scores)
rfm_score.sample(n=5, random_state=745)
```

Out[70]:

| | customer_id | recency | frequency | monetary | r_score | f_score |
|---|---|---|---|---|---|---|
| **1387** | 95919727-3866-421f-8297-4f45348c3f33 | 375 | 1 | 4272.0 | 1 | 1 |
| **2196** | eb6521ae-56e3-4a72-9ea2-e9c69701ff3f | 298 | 3 | 2903.0 | 2 | 3 |
| **2154** | e6892eec-7b21-4a49-9e56-29268f9eb98a | 270 | 1 | 134.0 | 2 | 1 |
| **1602** | add6d8bd-36ca-4c46-8a0b-7f13bbd05555 | 60 | 2 | 658.0 | 4 | 2 |
| **1123** | 78b5ab9e-999b-4825-bc44-19958d4854c1 | 89 | 2 | 2410.0 | 4 | 2 |

## 5.4 Присваиваем оценку monetary покупателю

Присвоим оценку для покупателя - monetary. Чем больше общая сумма сумма покупок, тем лучше и выше ранг. И тут зададим явные границы. Так как у нас есть условные оптовики, которые сильно занизят нашим текущим покупателям оценку, скорректируем это, задав границы явно из данных описательной статистики, записав их в intervals_m. Последним ограничением поставим сумму максимальную по всем заказам 159508.

In [71]:

```
# monetary
intervals_m=[0, 300, 638, 1087, 2107, 159509]
rfm_score['m_score']=pd.cut(rfm_score.monetary, intervals_m, labels=frequency_scores)
rfm_score.sample(n=5, random_state=795)
```

Out[71]:

| | customer_id | recency | frequency | monetary | r_score | f_score | m_score |
|---|---|---|---|---|---|---|---|
| **727** | 4d5f6a00-ae1a-4bf4-988f-d05b6349a676 | 474 | 1 | 2990.0 | 1 | 1 | 5 |
| **613** | 409480c9-0085-4edb-9388-e499be4c6475 | 293 | 1 | 674.0 | 2 | 1 | 3 |
| **440** | 2f3400cc-7093-4c67-b244-2f5e44fad3bf | 201 | 1 | 254.0 | 3 | 1 | 1 |
| **648** | 44bb4a43-82ac-4c3c-b753-28c793ea9079 | 30 | 2 | 456.0 | 5 | 2 | 2 |
| **1291** | 8af95f8f-4dd4-46ef-8572-d52557f274cd | 409 | 1 | 568.0 | 1 | 1 | 2 |

Получили три столбца, в которых отдельно каждому показателю присвоена оценка от 1 до 5.

## 5.5 RFM-score

Теперь суммируем оценки из столбцов `r_score`, `f_score` и `m_score` и результат запишем в столбец `rfm_score`. Для суммирования столбцов тип данных из категориальных (после применения функции cut именно такой тип присваивается) переведем в целочисленный, применив к нужным столбцам функцию `to_numeric`.

In [72]:

```
# меняем тип данных
rfm_score[['r_score', 'f_score', 'm_score']] = rfm_score[['r_score', 'f_score',
'm_score']].apply(pd.to_numeric)
rfm_score.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2412 entries, 0 to 2411
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   customer_id  2412 non-null   object
 1   recency      2412 non-null   int64
 2   frequency    2412 non-null   int64
 3   monetary     2412 non-null   float64
```

```
 4   r_score      2412 non-null   int64
 5   f_score      2412 non-null   int64
 6   m_score      2412 non-null   int64
dtypes: float64(1), int64(5), object(1)
memory usage: 150.8+ KB
```

In [73]:

```
# суммируем оценки
rfm_score['rfm_score'] = rfm_score['r_score'] + rfm_score['f_score'] +
rfm_score['m_score']
rfm_score.sample(n=5, random_state=371)
```

Out[73]:

| | customer_id | recency | frequency | monetary | r_score | f_score | m_score | rfm_score |
|---|---|---|---|---|---|---|---|---|
| **899** | 6104e7b9-e091-4312-887a-78994dd0062e | 19 | 2 | 4971.0 | 5 | 2 | 5 | 12 |
| **746** | 4f7a63fc-8438-4c3a-a895-765bf982715d | 332 | 1 | 134.0 | 2 | 1 | 1 | 4 |
| **1520** | a48b8615-7a8c-4a3b-8237-4b4c74a4a522 | 445 | 1 | 899.0 | 1 | 1 | 3 | 5 |
| **844** | 5b6ebe47-d6fc-4063-bf61-c28751bafca2 | 6 | 2 | 1085.0 | 5 | 2 | 3 | 10 |
| **548** | 3a38e7e2-69cd-4b92-848d-25feec331af9 | 30 | 2 | 194.0 | 5 | 2 | 1 | 8 |

In [74]:

```
print(f"варианты получившихся оценок rfm_score покупателей
{list(rfm_score['rfm_score'].unique())}")
```

варианты получившихся оценок rfm_score покупателей [7, 5, 8, 10, 11, 9, 6, 13, 12, 4, 3, 14]

## 5.6 Сегменты покупателей

Произведем разделение покупателей на сегменты по их ценности (на основе RFM). Сегмент ценности RFM представляет собой категоризацию клиентов на основе их оценок RFM.

Выделим такие группы:

- 'Lost',
- 'Low-value' ,
- 'Medium-value' ,
- 'Top'
- 'Super Top.

Список с названием сегментов сохраним в `segment_labels`.

- 'Lost' - те кто покупал 372 дня назад и позднее, менее чем на 300 у.е., один раз.
- 'Low-value' - те кто покупал очень-очень давно или очень давно, и/или на маленькую сумму один раз.
- 'Medium-value' - кто покупал давно или недавно, на маленькую или высокую сумму, мало или много - в разных вариациях. Но где-то есть несколько низких оценок, которые и дают средний результат
- 'Top' - кто покупал скорее всего недавно, на среднюю или высокую сумму, один иди несколько раз
- 'Super Top'- кто купил недавно, на большую сумму и несколько раз.

Сегменты определяются путем разделения показателей RFM на отдельные диапазоны или группы, что позволяет проводить более детальный анализ общих характеристик RFM клиентов.

### 5.6.1 Сегментация с помощью qcut

Создадим столбец `value_segment` в который на основе оценки RFM из столбца `rfm_score` занесем результаты получившихся категорий.

Для того, чтобы разбить покупателей на категории по получившейся оценке RFM применим функцию pandas.qcut(). Она разбивает массив значений в сегменты одинакового размера на основе ранга или квантилей (аргумент q). Мы будем разбивать на 5 сегментов.

In [75]:

```
segment_labels = ['Lost', 'Low-value' , 'Medium-value' , 'Top', 'Super Top']
```

```
rfm_score['value_segment'] = pd.qcut(rfm_score['rfm_score'], q=5,
labels=segment_labels)
```

Посмотрим что получилось и выведем несколько случайных строк датафрейма `rfm_score`.

In [76]:

```
rfm_score.sample(n=5, random_state=2113)
```

Out[76]:

| | customer_id | recency | frequency | monetary | r_score | f_score | m_score | rfm_score | value_segment |
|---|---|---|---|---|---|---|---|---|---|
| **235** | 19385fc3-fdc2-41dd-8114-aefa73229132 | 70 | 2 | 418.0 | 4 | 2 | 2 | 8 | Medium-value |
| **186** | 13cf5f36-c807-4573-981d-72681fb835f8 | 337 | 1 | 450.0 | 2 | 1 | 2 | 5 | Lost |
| **232** | 190e21db-9be8-4c41-b14b-8e1510075e54 | 305 | 1 | 928.0 | 2 | 1 | 3 | 6 | Low-value |
| **2247** | f0054c60-0290-4608-b04f-84778d144bcd | 15 | 2 | 628.0 | 5 | 2 | 2 | 9 | Top |
| **530** | 37af149d-7e2a-4fe8-8fdb-a3a303e24875 | 296 | 1 | 405.0 | 2 | 1 | 2 | 5 | Lost |

In [77]:

```
print(f"Сегменты покупателей в толбце value_segment:\n\
{list(rfm_score['value_segment'].unique())}")
```

Сегменты покупателей в толбце value_segment:
['Medium-value', 'Lost', 'Top', 'Super Top', 'Low-value']

In [78]:

```
rfm_score.value_segment.value_counts()
```

Out[78]:

Medium-value 641
Lost 609
Top 480
Low-value 367
Super Top 315
Name: value_segment, dtype: int64

Описательная статистика получившихся сегментов покупателей.

In [79]:

```
rfm_score.groupby('value_segment').mean().T.round(2)
```

Out[79]:

| value_segment | Lost | Low-value | Medium-value | Top | Super Top |
|---|---|---|---|---|---|
| **recency** | 350.83 | 273.79 | 224.40 | 100.76 | 47.86 |
| **frequency** | 1.03 | 1.06 | 1.27 | 1.76 | 2.60 |
| **monetary** | 383.40 | 757.05 | 1838.01 | 1959.45 | 4078.44 |
| **r_score** | 1.71 | 2.44 | 2.92 | 4.08 | 4.70 |
| **f_score** | 1.03 | 1.06 | 1.27 | 1.76 | 2.07 |
| **m_score** | 1.65 | 2.51 | 3.27 | 3.61 | 4.65 |
| **rfm_score** | 4.38 | 6.00 | 7.47 | 9.44 | 11.43 |

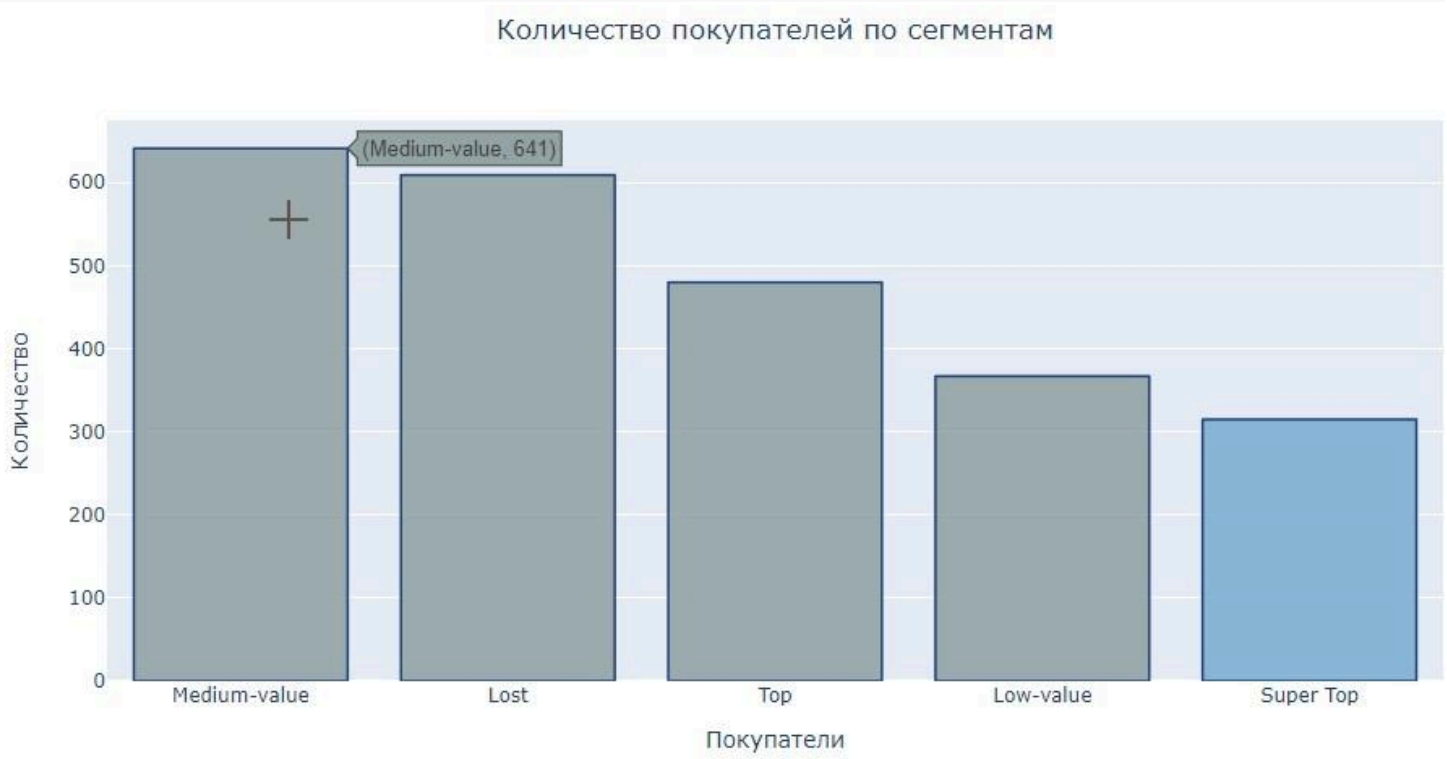
### 5.6.2 Диаграмма Сравнение RFM сегментов покупателей

Визуализируем получившиеся сегменты столбчатой диаграммой.

In [80]:

```
import plotly.colors
pastel_colors = plotly.colors.qualitative.Set3
segment_counts = rfm_score['value_segment'].value_counts()
```

```
fig = go.Figure(data=[go.Bar(x=segment_counts.index,
                             y=segment_counts.values,
                             marker=dict(color=pastel_colors))])

grey_color = '#95a5a6'
#зададим для всех кроме чемпионов цвет серый
fig.update_traces(marker_color=[grey_color if segment != 'Super Top' else
pastel_colors[i]
                                for i, segment in enumerate(segment_counts.index)],
                  marker_line_color='rgb(8, 48, 107)',
                  marker_line_width=1.5, opacity=0.9)

# Update the layout
fig.update_layout(xaxis_title='Покупатели',
                  yaxis_title='Количество',
                  showlegend=False)

fig.update_layout(title='Количество покупателей по сегментам', title_x=0.5) # название
fig.show()
```




На графике видно, что больше всего покупателей в сегменте Medium-value - 641 и Lost -609. Top занимает позицию посередине - 408 покупателей, у Low-value 367 покупателей, а у Super Top 315 покупателей.

## 5.7 Древовидная карта сегментов покупателей

Визуализируем сегменты покупателей и посмотрим, какие оценки из столбца `rfm_score` попали в каждый сегмент. Сгруппируем покупателей по сегментам и оценке и запишем в `segment_product_counts`. Затем построим древовидную карту.

In [81]:

```
segment_product_counts = rfm_score.groupby(['value_segment',
'rfm_score']).size().reset_index(name='count')

segment_product_counts = segment_product_counts.sort_values('count', ascending=False)
```

In [82]:

```
pastel_colors = plotly.colors.qualitative.Set2 #Set1 #Vivid #Dark24
fig_treemap_segment_product = px.treemap(segment_product_counts,
                                         path=['value_segment', 'rfm_score'],
                                         values='count',
                                         color='value_segment',
                                         color_discrete_sequence=pastel_colors) #
палитра

fig_treemap_segment_product.update_layout(title_text = 'Древо сегментов покупателей и
RFM оценки ', title_x=0.5)  # название
fig_treemap_segment_product.show()
```


Карта разбита по 5 цветам, что соответствует количеству получившихся сегментов покупателей. Два цвета занимают примерно 1/2 карты каждый цвет, и оставшиеся 3 цвета делят между собой оставшиеся 1/2 в неравных пропорциях. Очень наглядно.

Смотрим, что по оценкам, которые набрали покупатели в ходе RFM анализа:

- 'Medium-value' - Занимает примерно 1/5 карты, сюда попали покупатели, которые набрали 8 или 7 баллов.
- 'Lost' - Занимает примерно 1/5 карты, сюда попали покупатели, которые набрали 3, 4 или 5 баллов.
- 'Top' - Занимает примерно 1/5 карты, сюда попали покупатели, которые набрали 9 или 10 баллов, причем набравших 9 баллов и 10 примерно одинаково. Это хорошо.
- 'Low-value' - Занимает примерно 1/6 карты, сюда попали покупатели, которые набрали 6 баллов
- 'Super Top'- покупатели, сумевшие набрать 11 или 12 баллов из 15. 15 баллов не набрал никто, ведь по графику мы видели, что в последние 100 дней количество покупателей равно количеству заказов, а значит это все новые покупатели. совершившие по одной покупке, а значит они и не смогут получить повышенный балл за повторные покупки.

Судя по карте можно предположить, что разбивка покупателей на сегменты исходя из оценок RFM прошла удачно.

## 5.8 Итоги сегментации покупателей

С помощью rfm анализа мы смогли дать оценку и распределить покупателей на сегменты в зависимости от показателей давности, частоты и денег.

Мы смогли выделить сегменты покупателей которые покупают много, часто и приносят деньги. Этот сегмент самых лояльных пользователей.

Определить сегмент пользователей, которые делают заказы часто и много - работа с таким сегментом будет заключаться в попытках поднять средний чек - возможно, не обо всех продуктах и услугах компании известно этим пользователям.

Покупатели разбиты на сегменты. Теперь обработаем товары, чтобы разделить их на товарные категории. А в дальнейшем проведем сравнение сегментов покупателей по товарам и сезонности. Но сначала опишем профили получившихся сегментов покупателей.

In [83]:

```
#rfm_score.query('value_segment == "Lost"').describe()
```

# 6 Профили покупателей по сегментам:

### Super Top

- оценка по RFM диапазон: 11-14
- давность покупок медиана дней: 38
- средний чек у.е.: 1568
- медианный чек у.е.: 1405
- средняя общая сумма покупок у.е: 4078
- частота покупок медиана кол-во: 2

### Top

- оценка по RFM диапазон: 9-10
- давность покупок медиана дней: 76
- средний чек у.е.: 1113
- медианный чек у.е.: 552
- средняя общая сумма покупок у.е: 1959
- частота покупок медиана кол-во: 2

### Medium-value

- оценка по RFM диапазон: 7-8
- давность покупок медиана дней: 222
- средний чек у.е.: 1447
- медианный чек у.е.: 1087
- средняя общая сумма покупок у.е: 1838
- частота покупок медиана кол-во: 1

### Low-value

- оценка по RFM диапазон: 6
- давность покупок медиана дней: 281
- средний чек у.е.: 721
- медианный чек у.е.: 600
- средняя общая сумма покупок у.е: 757
- частота покупок медиана кол-во: 1

### Lost

- оценка по RFM диапазон: 3-5
- давность покупок медиана дней: 360
- средний чек у.е.: 372
- медианный чек у.е.: 299
- средняя общая сумма покупок у.е: 383
- частота покупок медиана кол-во: 1

# 7 Проверка корректности сегментов покупателей

Гипотезами проверяем корректность разбивки покупателей на категории. Выдвинем две гипотезы, если хотя бы одна из двух гипотез покажет статистически значимую разницу между сегментами покупателей, то

распределение покупателей на сегментов корректно. Если статистически значимой разницы нет, то необходимо пересмотреть подход в сегментации покупателей, разбить покупателей на сегменты снова и повторить проверку.

Если хоть одна проверка покажет статистически значимую разницу между кластерами, значит кластеризация мы проведена корректно. Если две проверки не обнаружат статистически значимой разницы между двумя одинаковыми кластерами - стоит пересмотреть разбивку на кластеры.

Прежде чем мы сформулируем и проверим гипотезы на основании исследования выберем предположения из которых будем формулировать гипотезу

- Есть различия между категориями покупателей в среднем чеке.
- Есть различия между категориями покупателей по частоте покупок.

Соответственно сформулируем нулевую и альтернативную гипотезы:

Гипотеза 1

- H0 - Нет различий между категориями покупателей в среднем чеке.
- H1 - Есть различия между категориями покупателей в среднем чеке.

Гипотеза 2

- H0 - Нет различий между категориями покупателей по частоте покупок.
- H1 - Есть различия между категориями покупателей по частоте покупок.

## 7.1 Присвоим id для Сегментов покупателей

Для удобства комбинаций, закодируем сегменты покупателей буквами и добавим столбец с кодировкой id_value_segment

- A - 'Super Top'
- B - 'Top',
- C - 'Medium-value',
- D - 'Low-value',
- E - 'Lost'.

In [84]:

```
rfm_score['id_value_segment'] = rfm_score['value_segment']\
.replace(['Medium-value', 'Lost', 'Top', 'Low-value', 'Super Top'], ['C', 'E', 'B',
'D', 'A'])
rfm_score.sample(n=2, random_state=1)
```

Out[84]:

| | customer_id | recency | frequency | monetary | r_score | f_score | m_score | rfm_score | value_segment | id_value_segment |
|---|---|---|---|---|---|---|---|---|---|---|
| **1662** | b3091c0c-a36e-460f-9123-1a642d1ade5c | 85 | 2 | 1394.0 | 4 | 2 | 4 | 10 | Top | B |
| **2383** | fc9641ba-8f74-45de-a60a-342d2e0dd7af | 258 | 1 | 152.0 | 3 | 1 | 1 | 5 | Lost | E |

Создадим столбец в датафрейме `rfm_score` со средним чеком по покупателю `avg_check`.

In [85]:

```
rfm_score['avg_check'] = rfm_score['monetary'] / rfm_score['frequency']
rfm_score.sample(n=3, random_state=1)
```

Out[85]:

| | customer_id | recency | frequency | monetary | r_score | f_score | m_score | rfm_score | value_segment | id_value_segment | avg_check |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1662** | b3091c0c-a36e-460f-9123-1a642d1ade5c | 85 | 2 | 1394.0 | 4 | 2 | 4 | 10 | Top | B | 697.0 |
| **2383** | fc9641ba-8f74-45de-a60a-342d2e0dd7af | 258 | 1 | 152.0 | 3 | 1 | 1 | 5 | Lost | E | 152.0 |
| **1398** | 96969c54-d375-48b3-b3bd-df07cb488a49 | 254 | 2 | 1508.0 | 3 | 2 | 4 | 9 | Top | B | 754.0 |

## 7.2 Двойные комбинации сегментов покупателей

Получим все комбинации сегментов, которые нужно сравнить между собой с помощью модуля itertools.

In [86]:

```
# запишем отсортированный по алфавиту список элементов для комбинации
segment_list = sorted(list(rfm_score['id_value_segment'].unique()))

# Сгенерируем все возможные двойные комбинации в список
combinations = list(itertools.combinations(segment_list, 2))

print(f"Получилось {len(combinations)} уникальных пар сегментов покупателей:\n\
{combinations}")
```

Получилось 10 уникальных пар сегментов покупателей:
[('A', 'B'), ('A', 'C'), ('A', 'D'), ('A', 'E'), ('B', 'C'), ('B', 'D'), ('B', 'E'), ('C', 'D'), ('C', 'E'), ('D', 'E')]

## 7.3 Таблицы для проверки гипотезы 1

Создадим таблицы для проверки гипотезы

In [87]:

```
segment_a = rfm_score.query('id_value_segment == "A"')
segment_b = rfm_score.query('id_value_segment == "B"')
segment_c = rfm_score.query('id_value_segment == "C"')
segment_d = rfm_score.query('id_value_segment == "D"')
segment_e = rfm_score.query('id_value_segment == "E"')
```

## 7.4 Гипотеза №1 о равенстве средних чеков

Мы предположили, что между средним чеком у покупателей разных сегментов есть разница. Формулируем нулевую и альтернативную гипотезы:

- H0 - Нет различий между категориями покупателей в среднем чеке.
- H1 - Есть различия между категориями покупателей в среднем чеке.

У нас 5 сегментов. Необходимо сравнить их между собой. Это дает 10 вариантов комбинаций гипотез между собой.

Для удобства, напишем функцию.

### 7.4.1 Функция проверки гипотезы 1

In [88]:

```
# функция для проверки  гипотезы по среднему чеку
def test(row_1, row_2, alpha=0.05):
    '''
Функция принимает на вход выборки сегментов покупателей. Попарно проверяет
есть ли статистически значимая разница между средним чеком в выборке

Входные параметры:
    - row_1, row_2, - сформированные выборки по сегментам
    - alpfa - выбранный уровень статистической значимости
    '''
    # статистическая значимость различия средних чеков между группами
    results = stats.mannwhitneyu(row_1, row_2)
    print('p-значение: ', (round(results.pvalue, 4)))

    if (results.pvalue < alpha):
        print('Внимание!\nОтвергаем нулевую гипотезу: между средними чеками есть
статистически значимая разница.\n\
Полученное значение p-value меньше заданного уровня значимости.')
    else:
```

```
        print('Не получилось отвергнуть нулевую гипотезу, нет оснований считать средние чеки разными')
    print()
```

### 7.4.2 Средний чек по сегментам покупателей

Выделим средние чеки по покупателям и запишем.

In [89]:

```
# выделим средний чек
segment_a_avg_check = segment_a['avg_check']
segment_b_avg_check = segment_b['avg_check']
segment_c_avg_check = segment_c['avg_check']
segment_d_avg_check = segment_d['avg_check']
segment_e_avg_check = segment_e['avg_check']
```

### 7.4.3 Статистический тест гипотезы 1

Запустим тест с помощью функции test(rpw_1, row2, alpha=0.05)

In [90]:

```
print(f"1. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'B'):\n")
row_1 = segment_a_avg_check
row_2 = segment_b_avg_check
test(row_1, row_2)

print(f"2. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'C'):\n")
row_1 = segment_a_avg_check
row_2 = segment_c_avg_check
test(row_1, row_2)

print(f"3. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'D'):\n")
row_1 = segment_a_avg_check
row_2 = segment_d_avg_check
test(row_1, row_2)

print(f"4. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'E'):\n")
row_1 = segment_a_avg_check
row_2 = segment_e_avg_check
test(row_1, row_2)
```

1. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'B'):

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

2. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'C'):

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.

Полученное значение p-value меньше заданного уровня значимости.

3. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'D'):

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

4. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'E'):

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

У покупателей Сегмента A есть статистически значимые отличия между средними чеками с другими сегментами покупателей. Посмотрим на другие сегменты. Продолжим статистический тест дальше

In [91]:

```
print(f"5. Рассмотрим различие в средних чеках между покупателей из сегментов ('B',
'C'):\n")
row_1 = segment_b_avg_check
row_2 = segment_c_avg_check
test(row_1, row_2)

print(f"6. Рассмотрим различие в средних чеках между покупателей из сегментов ('B',
'D') :\n")
row_1 = segment_b_avg_check
row_2 = segment_d_avg_check
test(row_1, row_2)

print(f"7. Рассмотрим различие в средних чеках между покупателей из сегментов ('B',
'E') :\n")
row_1 = segment_b_avg_check
row_2 = segment_e_avg_check
test(row_1, row_2)
```

5. Рассмотрим различие в средних чеках между покупателей из сегментов ('B', 'C'):

p-значение:  0.0055
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

6. Рассмотрим различие в средних чеках между покупателей из сегментов ('B', 'D') :

p-значение:  0.0021
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.

Полученное значение p-value меньше заданного уровня значимости.

7. Рассмотрим различие в средних чеках между покупателей из сегментов ('B', 'E') :

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

Комбинации с группой B дали статистически значимые отличия между средними чеками с остальными сегментами покупателей. Продолжим статистический тест дальше.

In [92]:

```
print(f"8. Рассмотрим различие в средних чеках между покупателей из сегментов ('C', 'D')  :\n")
row_1 = segment_c_avg_check
row_2 = segment_d_avg_check
test(row_1, row_2)

print(f"9. Рассмотрим различие в средних чеках между покупателей из сегментов ('C', 'E') :\n")
row_1 = segment_c_avg_check
row_2 = segment_e_avg_check
test(row_1, row_2)

print(f"10.Рассмотрим различие в средних чеках между покупателей из сегментов ('D', 'E') :\n")
row_1 = segment_d_avg_check
row_2 = segment_e_avg_check
test(row_1, row_2)
```

8. Рассмотрим различие в средних чеках между покупателей из сегментов ('C', 'D')  :

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

9. Рассмотрим различие в средних чеках между покупателей из сегментов ('C', 'E') :

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

10.Рассмотрим различие в средних чеках между покупателей из сегментов ('D', 'E') :

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

У категории покупателей С есть статистически значимая разница между средними чеками с остальными сегментами.

### 7.4.4 Итоги статистического теста гипотезы 1

Проверили гипотезе и выяснили, что при пороговом значении в 5%:

- нет оснований отвергать гипотезу о том, что есть различия между категориями покупателей в среднем чеке.

Во всех комбинациях сегментов покупателей статистически значимая разница между средними чеками сегментов покупателей выявлена.

## 7.5 Гипотеза № 2 о равенстве количества покупок

Мы предположили, что между частотой покупок у покупателей разных сегментов есть разница. У нас 5 сегментов. Необходимо сравнить их между собой. Это дает 10 вариантов комбинаций гипотез между собой.

Гипотеза 2

- H0 - Нет различий между категориями покупателей по частоте покупок.
- H1 - Есть различия между категориями покупателей по частоте покупок.

У нас 5 сегментов. Необходимо сравнить их между собой. Это дает 10 вариантов комбинаций гипотез между собой.

Скорректируем нашу написанную функцию и используем для статистического теста этой гипотезы.

### 7.5.1 Частота покупок по сегментам покупателей

Выделим частоту покупок по сегментам покупателей и запишем.

In [93]:

```
# выделим количество покупок
segment_a_frequency = segment_a['frequency']
segment_b_frequency = segment_b['frequency']
segment_c_frequency = segment_c['frequency']
segment_d_frequency = segment_d['frequency']
segment_e_frequency = segment_e['frequency']
```

### 7.5.2 Функция проверки гипотезы 2

In [94]:

```
# функция для проверки  гипотезы по среднему чеку
def test2(row_1, row_2, alpha=0.05):
    '''
Функция принимает на вход выборки сегментов покупателей. Попарно проверяет
есть ли статистически значимая разница между средним чеком в выборке

Входные параметры:
    - row_1, row_2, - сформированные выборки по сегментам
    - alpfa - выбранный уровень статистической значимости
    '''
    # статистическая значимость различия средних чеков между группами
    results = stats.mannwhitneyu(row_1, row_2)
    print('p-значение: ', (round(results.pvalue, 4)))

    if (results.pvalue < alpha):
        print('Внимание!\nОтвергаем нулевую гипотезу: между количеством покупок есть
статистически значимая разница.\n\
Полученное значение p-value меньше заданного уровня значимости.')
    else:
        print('Не получилось отвергнуть нулевую гипотезу, нет оснований считать
количеством покупок между сегментами разными.')
```

```
    print()
```

### 7.5.3 Статистический тест гипотезы 2

Запустим тест с помощью функции test(rpw_1, row2, alpha)

In [95]:

```
print(f"1. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'B'):\n")
row_1 = segment_a_frequency
row_2 = segment_b_frequency
test(row_1, row_2)

print(f"2. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'C'):\n")
row_1 = segment_a_frequency
row_2 = segment_c_frequency
test(row_1, row_2)

print(f"3. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'D'):\n")
row_1 = segment_a_frequency
row_2 = segment_d_frequency
test(row_1, row_2)

print(f"4. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'E'):\n")
row_1 = segment_a_frequency
row_2 = segment_e_frequency
test(row_1, row_2)
```

1. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'B'):

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

2. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'C'):

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

3. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'D'):

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

4. Рассмотрим различие в средних чеках между покупателей из сегментов ('A', 'E'):

p-значение:  0.0

Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

Между сегментом покупателей A и всеми остальными сегментами есть разница в количестве покупок.
Продолжим статистический тест дальше.

In [96]:

```
print(f"5. Рассмотрим различие в средних чеках между покупателей из сегментов ('B',
'C'):\n")
row_1 = segment_b_frequency
row_2 = segment_c_frequency
test(row_1, row_2)

print(f"6. Рассмотрим различие в средних чеках между покупателей из сегментов ('B',
'D') :\n")
row_1 = segment_b_frequency
row_2 = segment_d_frequency
test(row_1, row_2)

print(f"7. Рассмотрим различие в средних чеках между покупателей из сегментов ('B',
'E') :\n")
row_1 = segment_b_frequency
row_2 = segment_e_frequency
test(row_1, row_2)
```

5. Рассмотрим различие в средних чеках между покупателей из сегментов ('B', 'C'):

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

6. Рассмотрим различие в средних чеках между покупателей из сегментов ('B', 'D') :

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

7. Рассмотрим различие в средних чеках между покупателей из сегментов ('B', 'E') :

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

В комбинациях с сегментом покупателей B и всеми остальными сегментами есть разница в количестве покупок. Продолжим статистический тест дальше.

In [97]:

```
print(f"8. Рассмотрим различие в средних чеках между покупателей из сегментов ('C', 'D')  :\n")
row_1 = segment_c_frequency
row_2 = segment_d_frequency
test(row_1, row_2)

print(f"9. Рассмотрим различие в средних чеках между покупателей из сегментов ('C', 'E') :\n")
row_1 = segment_c_frequency
row_2 = segment_e_frequency
test(row_1, row_2)

print(f"10.Рассмотрим различие в средних чеках между покупателей из сегментов ('D', 'E') :\n")
row_1 = segment_d_frequency
row_2 = segment_e_frequency
test(row_1, row_2)
```

8. Рассмотрим различие в средних чеках между покупателей из сегментов ('C', 'D')  :

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

9. Рассмотрим различие в средних чеках между покупателей из сегментов ('C', 'E') :

p-значение:  0.0
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

10.Рассмотрим различие в средних чеках между покупателей из сегментов ('D', 'E') :

p-значение:  0.0143
Внимание!
Отвергаем нулевую гипотезу: между средними чеками есть статистически значимая разница.
Полученное значение p-value меньше заданного уровня значимости.

У категории покупателей C есть статистически значимая разница между количеством покупок с остальными категориями покупателей.

### 7.5.4 Итоги статистического теста гипотезы 2

Проверили гипотезу и выяснили, что при пороговом значении в 5%:

- нет оснований отвергать гипотезу о различии между категориями покупателей по частоте покупок.

Во всех случаях статистически значимая разница между количеством покупок среди разных сегментов покупателей выявлена.

### 7.5.5 Выводы по статистическим тестам гипотез

Можно сделать вывод, что покупатели разделены на категории корректно. Можно продолжить исследование уже по категориям покупателей.

# 8 Риск ошибок в статистическом тесте.

При проверке статистических гипотез был выбран уровень значимости 0.05. Это означает, что допустимая вероятность ложноположительного результата для одного теста равна 5%.

Всего было сделано 20 проверок статистических гипотез, так как проверено 2 гипотезы, в каждой проверено по 10 комбинаций выборок.

Найдем допустимую вероятность ложноположительного результата хотя бы в одном из тестов.

### 8.1 Риск ошибок при уровне значимости 0.05

In [98]:

```
print(f'Вероятность того, что не будет ни одного ложноположительного\
результата при 20 тестах, равна: {round(((1 - 0.05)**20) * 100, 1)}%')
print(f'Вероятность того, что хотя бы один тест даст ложноположительный\
результат, равна: {round((1 - (1 - 0.05)**20) * 100, 1)}%')
```

Вероятность того, что не будет ни одного ложноположительного результата при 20 тестах, равна: 35.8%

Вероятность того, что хотя бы один тест даст ложноположительный результат, равна: 64.2%

### 8.2 Риск ошибок при уровне значимости 0.025

Что если мы снизим уровень значимости до 0.025 и затем до 0.01

In [99]:

```
print(f'Вероятность того, что не будет ни одного ложноположительного\
результата при 20 тестах, равна: {round(((1 - 0.025)**20) * 100, 1)}%')
print(f'Вероятность того, что хотя бы один тест даст ложноположительный\
результат, равна: {round((1 - (1 - 0.025)**20) * 100, 1)}%')
```

Вероятность того, что не будет ни одного ложноположительного результата при 20 тестах, равна: 60.3%

Вероятность того, что хотя бы один тест даст ложноположительный результат, равна: 39.7%

### 8.3 Риск ошибок при уровне значимости 0.01

In [100]:

```
print(f'Вероятность того, что не будет ни одного ложноположительного\
результата при 20 тестах, равна: {round(((1 - 0.01)**20) * 100, 1)}%')
print(f'Вероятность того, что хотя бы один тест даст ложноположительный\
результат, равна: {round((1 - (1 - 0.01)**10) * 100, 1)}%')
```

Вероятность того, что не будет ни одного ложноположительного результата при 20 тестах, равна: 81.8%

Вероятность того, что хотя бы один тест даст ложноположительный результат, равна: 9.6%

### 8.4 Резюмируем по уровню значимости.

Оставим уровень значимости в пределах 0.05. Понижая уровень ошибок первого рода, мы увеличиваем риск ошибок второго рода, а именно ложнонегативный результат.

## 9 Анализ данных. Разделение товаров на категории

### 9.1 Соединение таблиц

Соединим таблицы для дальнейшей работы.

Сперва выведем по паре строк, чтобы визуально подсветить нужные столбцы, по которым будем соединять таблицы. Затем методом merge добавим к таблице `dfc` таблицу `rfm_score` с оценкой покупателей .

Получившуюся таблицу запишем как `dfc_rfm`.

In [101]:

```
display(dfc.head(2))
rfm_score.head(2)
```

| | date | customer_id | order_id | product | quantity | price | total | year_month |
|---|---|---|---|---|---|---|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **0** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Алоэ Вера, d12, h30 | 1 | 142.0 | 142.0 | 2018-10 |
| **1** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Кофе Арабика, d12, h25 | 1 | 194.0 | 194.0 | 2018-10 |

Out[101]:

| | customer_id | recency | frequency | monetary | r_score | f_score | m_score | rfm_score | value_segment | id_value_segment | avg_check |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 000d6849-084e-4d9f-ac03-37174eaf60c4 | 108 | 1 | 555.0 | 4 | 1 | 2 | 7 | Medium-value | C | 555.0 |
| **1** | 001cee7f-0b29-4716-b202-042213ab038 | 350 | 1 | 442.0 | 2 | 1 | 2 | 5 | Lost | E | 442.0 |

In [102]:

```
dfc_rfm = dfc.merge(rfm_score, on=["customer_id"])
# поменяем столбцы местами и избавимся от некоторых.
dfc_rfm = (dfc_rfm[['date', 'customer_id', 'order_id',
                    'product', 'quantity', 'price',
                    'total', 'value_segment', 'year_month']])
dfc_rfm.head(2)
```

Out[102]:

| | date | customer_id | order_id | product | quantity | price | total | value_segment | year_month |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Алоэ Вера, d12, h30 | 1 | 142.0 | 142.0 | Lost | 2018-10 |
| **1** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Кофе Арабика, d12, h25 | 1 | 194.0 | 194.0 | Lost | 2018-10 |

Проверим категории покупателей и посмотрим описательную статистику на наличие пропусков.

In [103]:

```
dfc_rfm.value_segment.unique()
```

Out[103]:

['Lost', 'Medium-value', 'Super Top', 'Top', 'Low-value']
Categories (5, object): ['Lost' < 'Low-value' < 'Medium-value' < 'Top' < 'Super Top']

In [104]:

```
dfc_rfm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5521 entries, 0 to 5520
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype         
---  ------         --------------  -----         
 0   date           5521 non-null   datetime64[ns]
 1   customer_id    5521 non-null   object        
 2   order_id       5521 non-null   int64         
 3   product        5521 non-null   object        
 4   quantity       5521 non-null   int64         
 5   price          5521 non-null   float64       
 6   total          5521 non-null   float64       
 7   value_segment  5521 non-null   category      
 8   year_month     5521 non-null   object        
dtypes: category(1), datetime64[ns](1), float64(2), int64(2), object(3)
memory usage: 393.8+ KB
```

### 9.1.1 Итоги объединения таблиц

Таблицы объединены, нужные столбцы оставлены. Категории товаров перенесены без ошибок. Можно приступать к категоризации товаров.

## 9.2 Категоризация товаров

### 9.2.1 Список всех товаров

Для проведения анализа необходимо выделить товарные категории. Так как присутствует большое количество имен собственных, то разбивку на категории лучше произвести вручную. Выведем список товаров для выбора ключевых слов в нижнем регистре. Используем его для первой итерации выбора ключевых слов вручную.

Внимание, подбор производился на неочищенных данных, после очистки возможно несколько значений в списках категорий товаров будут лишними. Не станем их искать и перебирать снова, на результат несколько лишних значений в списках товаров не повлияют, зато при необходимости списки можно использовать на неочищенных от выбросов данных.

In [105]:

```
#получим список товаров для анализа в нижнем регистре
dfc_product = dfc_rfm['product'].str.lower()
list(dfc_product.unique())
```

Out[105]:

['комнатное растение в горшке алое вера, d12, h30',
 'комнатное растение в горшке кофе арабика, d12, h25',
 'радермахера d-12 см h-20 см',
 ВЫРЕЗАНО
 ...]

### 9.2.2 Списки товаров по категориям

Вручную за 2-3 итерации выберем ключевые слова и распределим их по спискам товаров по категориям. Сохраним списки с ключевыми словами в переменные:

- цветы/сад - `flower`
- хозяйственная утварь - `household_utensils`
- для ремонта - `for_repairs`
- кухонная утварь - `kitchen_utensils`
- текстиль - `textile`
- интерьер/декор - `decor`
- хозяйственные сумки - `shopping_bags`

In [106]:

```
# цветы/сад
flower = ['растение', 'рассада',
          ' d-', 'герань', 'пеларгония', 'кампанула',
          'кориандр', 'афеляндра', 'аспарагус',
          'хризантема','бархатцы', 'космея', 'морковь', 'настурция', 'огурец', 'петуния',
          'алиссум', 'гвоздика', 'годеция', 'календула', ' капуста', 'кореопсис', 'лапчатка',
          'львиный зев', 'цинерария', 'эшшольция', 'подсолнечник',
          'флокс', 'гиностемма', 'горшок', 'ред лейс', 'черенок', 'фуксия', 'горшке',
'петрушка', 'салат',
          'гортензия', 'клен', 'помидор', 'вигна', 'шалфей', 'табак', 'сельдерей', 'любисток',
'капуста', 'в кассете',
          'анемона', 'лаватера', 'кашпо', 'небесная лазурь', 'колокольчик', 'камнеломка',
'бадан', 'физостегия', 'осина',
          'солидаго', 'мими эден', 'бузульник', 'астра', 'аквилегия', 'калибрахоа', 'вербена',
'барвинок', 'бакопа',
          'седум', 'рудбекия', 'нивянник', 'монарда', 'гайлардия', 'одноголовая', 'хоста',
'арбуз', 'маттиола', 'клубника',
          'примула', 'вероника', 'калибрахоа', 'цикламен', 'примула', 'декабрист', 'калла',
'пиретрум', 'лобелия',
          'виноград', 'базилик', 'цинния', 'дыня', 'гортензия', 'энотера', 'платикодон',
'папоротник', 'лилейник', 'виола',
          'калибрахоа', 'ясколка', ' эхинацея', 'котовник', 'колокольчик', 'вербейник',
'лавр', 'ель', 'антуриум', 'гипсофила',
```

```
        'дендробиум', 'горох', 'земляника', 'незабудка', 'тимьян', 'балконное чудо',
'укроп', ' дендробиум', 'тюльпан',
        'ранункулус', 'овсянница', 'георгина', 'смолевка', 'эхинацея'

       ]
# хозяйственная утварь
household_utensils = ['таз ', 'чехол ', 'вешалка ', 'сушилка', 'гладильная', 'щетка-сметка',
                      'прищепок', 'коврик', 'крючок', 'ерш', 'дозатор', 'ёрш',
                      'мусорный', 'перчатки', 'набор вешалок', 'плечики', 'стиральный
биопорошок',
                      'чистящий крем', 'щетка', 'щетка-утюжок', 'жидкое мыло', 'ковёр',
                      'пена для ванн', 'подрукавник', 'зубная', 'веник', 'швабра',
'подголовник',
                      'ведро', 'швабры', 'сиденье в ванну', 'утюг', 'корыто', ' урна',
                      'сиденье для унитаза', 'сметка', 'ложка обувная', 'к швабре',
'сантехнических',
                      'паста для полировки', 'петля-стрела', 'держатель', 'пробка', 'губка ',
                      'фен', 'антижир', 'чистки сантехники', 'ролик', 'для ролика', 'шило',
                      'для мытья', 'известкового налета', 'ополаскиватель', 'сетка для
глажения', 'вешалка-сушилка',
                      'мыльница', 'мыло', 'бегония', 'кольца', 'вантуз', 'кондиционер',
'посудомоечных',
                      'для стирки', 'котел', 'для ванной', 'FLIP BIN CURVER', 'для чехла', '
урна', 'для окон', 'вешалки',
                      'тряпка', 'вакуумный пакет', 'совок', 'для унитаза', 'leifheit',
'вешалка-стойка', 'прищепки',
                      'webber', 'отбеливатель', 'насадка-моп', 'штанга', 'окномойка', 'для
гладильной', 'отжим',
                      'придверный', 'прищепки', 'пылесос', 'карниз', 'сиденье для ванны',
'вешалка-перекладина',
                      'контейнер для мусора', 'нетканые', 'венчик', 'для белья', 'засоров',
'увлажняющая маска'

                     ]
# для ремонта
for_repairs = ['стремянка', 'стяжка', 'крепеж', 'пружина', 'петля', 'стремянки',
               'набор сверел', 'сверло-фреза', 'карниз' 'лестница-стремянка',
               'холодная сварка', 'угольник', 'линейка', ' бензин', 'бензин',
               'термометр', 'ручка-скоба', 'крючок', 'инструмент', 'стремянка-табурет',
              'основание для пробки', 'шпингалет', 'уголок', 'решетка', 'мебельная',
'завертка', 'мешок',
              'шпагат', 'насадка на валик', 'фиксатор-шар', 'сверло', 'фал капроновый',
'штангенциркуль',
              'напильников'
             ]
#кухонная утварь
kitchen_utensils = ['сковорода', 'тарелка', 'вилка', 'ложка', 'нож', 'luminarc',
                    'кувшин', 'толкушка', 'чайник', 'ёрш для бутылки', 'банка',
                    'противень', 'стеклянная крышка', 'посуды', 'миксер',
                    'термокружка', 'масленка', 'рыбочистка', 'салатников', 'овощеварка ',
                    'терка', 'контейнер для приготовления', 'кружка ',
                    'attribute', 'фужеров', 'салфетница', 'стакан', 'лоток для холодца',
'мантоварка',
                    'набор стаканов', 'ковш', 'ножницы', 'для выпечки', 'скребок',
                   'картофелемялка', 'рассекатель', 'кексов', 'весы',
                   'миска', 'пресс для чеснока', 'термостакан', 'дуршлаг', 'столовых',
                    'термокружка', 'термос', 'alparaisa', 'электроштопор', 'контейнер для
свч', 'мантоварка-пароварка',
                    'для соуса', 'просеиватель', 'кастрюля', 'luminar', 'кипятильник',
'соковыжималка', 'пьезозажигалка',
                   'соковарка', 'бидон', 'сахарница', 'хлебница', 'крышка', 'шприц', 'для
пикника', 'лоток',
                    'сотейник', 'блюдо', 'контейнер герметичный', 'для продуктов', 'емкость
для свч', 'скалка',
                   'разделочная', 'тортница'
                  ]
#текстиль
textile = ['плед', 'скатерть', 'простыня', 'штора', 'подушка', 'полотенце',
                'салфетка', 'наматрасник', 'наматрицник-чехол', 'халат',
```

```
                'покрывало', 'одеяло', 'постельное белье', 'пододеяльник', 'наволочка',
'наматрацник', 'хлопок']
# интерьер/декор
decor = ['корзина', 'муляж', 'цветок искусственный', 'искусственный цветок',
            'комплект для ванной', 'декоративная композиция', 'кофр' , 'светильник',
            'ящик', 'короб', 'коробка', 'комод', 'подставка', 'фоторамка', 'этажерка',
            'полки', 'ключница', 'искусственная', 'стеллаж', 'полка для обуви', 'корзинка',
        'искусственный', 'детский пуф', 'обувница', 'урна-пепельница', 'урна', 'ванна 70 л'
       ]

# хозяйственные сумки
shopping_bags = ['сумка-тележка', 'тележка','шнур', 'сумка', 'сумка для тележки', 'сумка
хозяйственная']
```

### 9.2.3 Функция для категоризации товаров set_category

Напишем функцию, которая позволит нам присвоить категорию товаров, путем поиска товаров в словаре с категориями товаров, которые мы составили вручную.

In [107]:

```
def set_category(list_product, column_category):
    """
     Привести список list_product в нижний регистр. Сохранить в переменную  key_words
     Привести список key_words в строку с разделителем спец. знак | между слов.
     Сделать срез нужного датасета по нужной колонке, привести его в нижний регистр,
     и осуществить поиск - содержит ли строка заданную подстроку в key_words. Если
содержит - осуществи замену.
    """
    key_words = [i.lower() for i in list_product]
    key_words = '|'.join(key_words)
    dfc_rfm.loc[dfc_rfm['product'].str.lower().str.contains(key_words, regex=True),
'product_category'] = column_category
```

### 9.2.4 Применение функции set_category

Добавим в датасет `dfc_rfm` столбец `product_category` и присвоим значение по умолчанию всем товарам категорию "прочие товары". Вызовем функцию для категоризации товаров и подставим переменные с категориями товаров и названиями категорий.

In [108]:

```
dfc_rfm['product_category'] = 'прочие товары'
# функция для категоризации товаров
set_category(flower, 'цветы/сад')
set_category(household_utensils, 'хозяйственная утварь')
set_category(for_repairs, 'для ремонта')
set_category(kitchen_utensils, 'кухонная утварь')
set_category(textile, 'текстиль')
set_category(decor, 'интерьер/декор')
set_category(shopping_bags, 'хозяйственные сумки')

print('Количество не отнесенных к категориям наименований товаров',\
     dfc_rfm.query('product_category == "прочие товары"')['product'].count(), 'шт.')
```

Количество не отнесенных к категориям наименований товаров 0 шт.

Посмотрели сколько наименований товаров попало в категорию "прочие товары" и снова добавили ключевые слова в список. Перезапустили функцию. Хватило 3 итерации и в категориях прочие товары не осталось товаров.

In [109]:

```
print(f"Количество уникальных товаров в товарных группах
{dfc_rfm['product'].nunique()} шт.\n\
```

```
В том числе по категориям товаров:")
dfc_rfm.groupby('product_category').agg({'product': 'nunique'}).reset_index()
```

Количество уникальных товаров в товарных группах 2333 шт.

В том числе по категориям товаров:

Out[109]:

| | product_category | product |
|---|---|---|
| 0 | для ремонта | 84 |
| 1 | интерьер/декор | 218 |
| 2 | кухонная утварь | 320 |
| 3 | текстиль | 135 |
| 4 | хозяйственная утварь | 551 |
| 5 | хозяйственные сумки | 110 |
| 6 | цветы/сад | 915 |

## 9.3 Диаграмма Продажи по товарным категориям

Визуализируем сколько единиц товаров продано в каждой товарной категории. Для этого построим диаграмму и распределим проданные товары в зависимости от товарной группы.

In [110]:

```
pastel_colors = plotly.colors.qualitative.Set3
product_category = dfc_rfm['product_category'].value_counts()

fig = go.Figure(data=[go.Bar(x=product_category.index,
                             y=product_category.values,
                             marker=dict(color=pastel_colors))])

grey_color = '#95a5a6'
#зададим для всех кроме лидеров по количеству цвет серый
fig.update_traces(marker_color=[grey_color if segment != 'цветы/сад' else
pastel_colors[i]
                                for i, segment in enumerate(product_category.index)],
                  marker_line_color='rgb(8, 48, 107)',
                  marker_line_width=1.5, opacity=0.9)

# Update the layout
fig.update_layout(xaxis_title='Товарные категории',
                  yaxis_title='количество строк заказов',
                  showlegend=False)

fig.update_layout(title='Предпочтения покупателей по товарным категориям ',
title_x=0.5) # название
fig.show()
```


In [111]:

```
dfc_rfm['product'].value_counts().sum()
```

Out[111]:

5521

### 9.3.1 Итоги категоризации товаров

Таблица содержит информацию о 2153 единицах уникальных наименованиях товаров, при этом больше всего товаров в категории цветы/сад, хозяйственная утварь и меньше всего в категории для ремонта.

# 10 Анализ данных. Сегменты покупателей и сезонность

Сезонность категорий товаров для каждого сегмента покупателей Сезонность выбрать исходя из имеющегося временного периода данных (месяц или сезон). Обосновать выбор.

- Есть ли сезонность в продаже товаров?

## 10.1 Добавим столбцы с днем недели

Добавим дополнительный столбец для дальнейшего анализа: выделим из даты день недели заказа:

- `order_weekday` - день недели в который был осуществлен заказ. Категоризируем: 0 — понедельник, 1 — вторник, 2 - среда, 3 - четверг, 4 - пятница, 5 - суббота, 6 - воскресенье.;

У нас присутствует неполные три года в исследуемом периоде. Поэтому месяц заказа остнется выделен с годом, эти данные хранятся в столбце `'year_month'`

In [112]:

```
#выделяем день недели заказа
dfc_rfm['order_weekday'] = dfc_rfm['date'].dt.weekday
#выделяем месяц заказа
#dfc_rfm['order_month'] = dfc_rfm['date'].dt.month
#dfc_rfm['order_weekday'] = dfc_rfm['order_weekday']\
#.replace([0, 1, 2, 3, 4, 5, 6], ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Вс'])

print(f"Дни недели: {dfc_rfm['order_weekday'].unique()}\n\
и месяц с годом {sorted(dfc_rfm['year_month'].unique())}\n\
в которых осуществлялись заказы.")
```

Дни недели: [0 3 1 2 4 5 6]

и месяц с годом ['2018-10', '2018-11', '2018-12', '2019-01', '2019-02', '2019-03', '2019-04', '2019-05', '2019-06', '2019-07', '2019-08', '2019-09', '2019-10', '2019-11', '2019-12', '2020-01']
в которых осуществлялись заказы.

## 10.2 Распределение сегментов покупателей по дням недели

Распределить категории покупателей по дням недели.

- Есть ли взаимосвязь между днем недели совершения покупок и количеством покупок?

In [113]:

```
customer_product = dfc_rfm.pivot_table(index='order_weekday',
                                       columns='value_segment',
                                       values='customer_id', aggfunc='count')
customer_product.style.background_gradient(low=0.75, high=1.0)
```

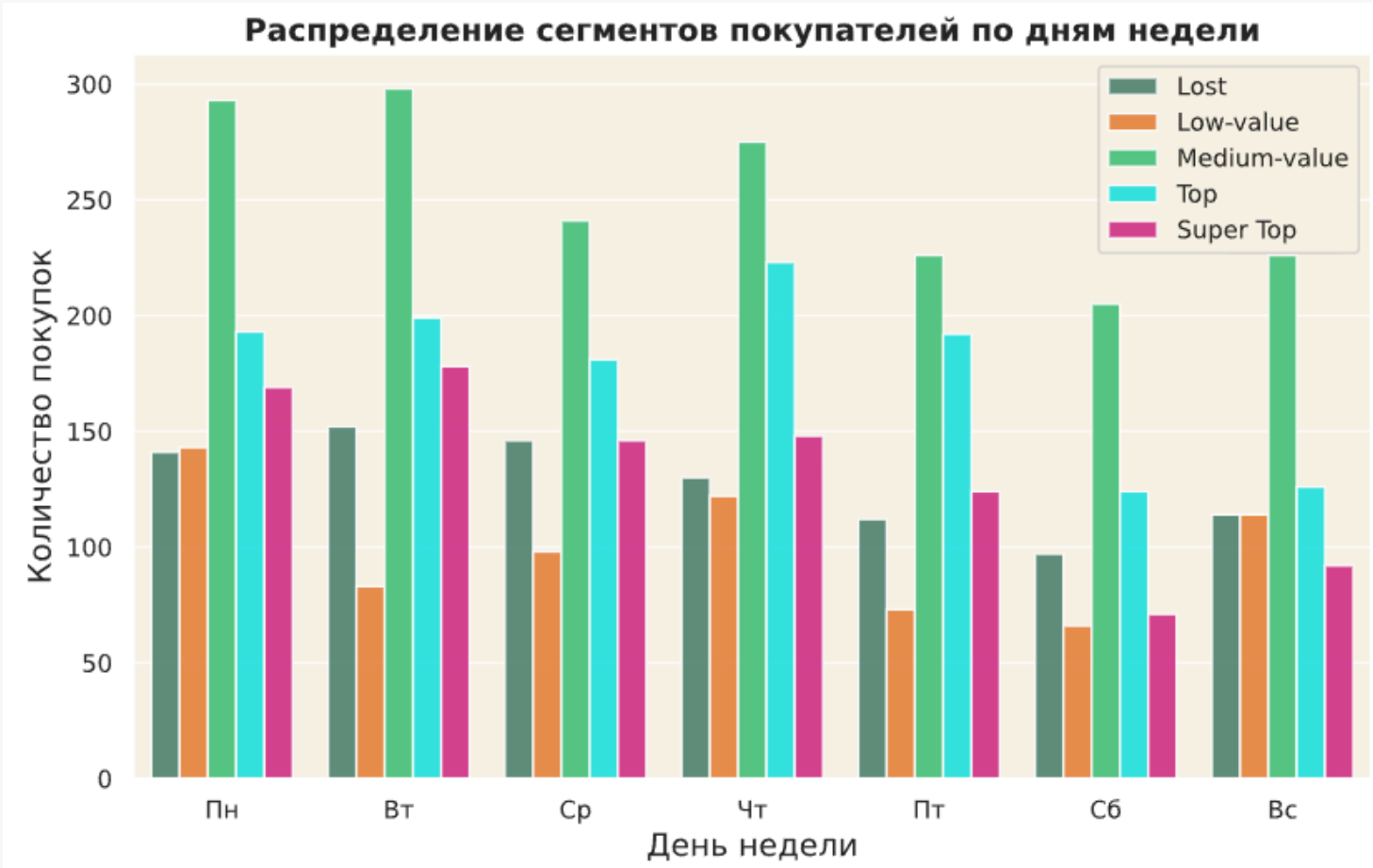
Out[113]:

| value_segment | Lost | Low-value | Medium-value | Top | Super Top |
|---|---|---|---|---|---|
| order_weekday | | | | | |
| 0 | 141 | 143 | 293 | 193 | 169 |
| 1 | 152 | 83 | 298 | 199 | 178 |
| 2 | 146 | 98 | 241 | 181 | 146 |
| 3 | 130 | 122 | 275 | 223 | 148 |
| 4 | 112 | 73 | 226 | 192 | 124 |
| 5 | 97 | 66 | 205 | 124 | 71 |
| 6 | 114 | 114 | 226 | 126 | 92 |

### 10.2.1 Диаграмма Распределение сегментов покупателей по дням недели

In [114]:

```
#Диаграмма День недели покупок для всех сегментов покупателей
plt.figure(figsize=(10,6))
sns.set (rc={'axes.facecolor':'#f7f1e3', 'figure.facecolor':
'#ffffff'})#'#f7f1e3'})#'#dfe4ea'
colors_customers = ['#458B74', '#FF7F24', '#2ed573', '#00FFFF', '#EE1289']
ax = sns.countplot(x=dfc_rfm['order_weekday'],
                   data=dfc_rfm,
                   hue='value_segment',
                   alpha=0.9,
                   palette=colors_customers)
                   #palette="pastel")
                   #dodge=False)
ax.set_title('Распределение сегментов покупателей по дням недели', fontsize = 14,
         fontweight ='bold')
#кастомизируем подписи по оси X - заменим на дни недели
ax.set_xticklabels(['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Вс']) #, rotation=45)

plt.legend(loc='upper right') #сместим легенду
plt.xlabel('День недели', fontsize = 14)
plt.ylabel('Количество покупок', fontsize = 14)
plt.show();
```




Как видим по диаграмме: **'Распределение сегментов покупателей по дням недели'** - у почти у всех категорий покупателей особых предпочтительных дней недели для покупок нет.

- Сегмент покупателей **'Lost'** чаще всего покупал по вторникам и средам, а реже - по субботам и пятницам.
- Сегмент покупателей **'Low-value'** чаще покупает по понедельникам, чем в остальные дни, а в субботу и пятницу покупает меньше всего.
- Сегмент покупателей **'Medium-value'** - чаще всего покупает по вторникам и понедельникам, а реже - по субботам.
- Сегмент покупателей **'Top'** - чаще всего покупает по четвергам, а реже - по субботам и воскресеньям.
- Сегмент покупателей **'Super Top'** чаще покупает во вторник и понедельник, реже в субботу и воскресенье.

### 10.2.2 Вывод по дням недели

Все сегменты покупателей покупают в любой день недели, какая-то взаимосвязь может быть между сегментом покупателей и днем недели покупок. Так как в выходные некоторые покупают меньше других, как и в разные будние дни. Но явной четкой взаимосвязи между днем недели совершения покупок и количеством покупок ни у какого сегмента покупателей по графику не видно.

## 10.3 Распределение сегментов покупателей по месяцам

Распределить категории покупателей по месяцам.

- Есть ли взаимосвязь между месяцем совершения покупок и количеством покупок?

Количество покупателей по сегментам купивших товары по месяцам, с учетом года покупки, так как у нас три неполных года наблюдений.

### 10.3.1 Диаграмма Распределение сегментов покупателей по месяцам

In [115]:

```
#Количество покупателей по сегментам купивших товары по месяцам
plt.figure(figsize=(12, 6))
sns.set (rc={'axes.facecolor':'#ffffff', 'figure.facecolor':'#dfe4ea'})
colors_customers = ['#458B74', '#FF7F24', '#2ed573', '#00FFFF', '#EE1289']
ax = sns.countplot(x=dfc_rfm['year_month'],
```

```
                    data=dfc_rfm,
                    hue='value_segment',
                    alpha=0.9,
                    palette=colors_customers)
                    #palette="pastel")
                    #dodge=False)
ax.set_title('Распределение сегментов покупателей по месяцам', fontsize = 14,
         fontweight ='bold')
ax.tick_params(axis='x', labelrotation=45) # градус наклона по оси x
plt.ylabel('количество покупателей', fontsize = 14)
plt.xlabel('', fontsize = 14)
plt.show();
```


Месяца: 1-Январь,2 - Февраль, 3 - Март, 4 - Апрель, 5 - Май, 6 - Июнь, 7 - Июль, 8 - Август, 9 - Сентябрь, 10 - Октябрь, 11 - Ноябрь, 12 - Декабрь. Года: 2018, 2019, 2020

In [116]:

```
customer_product = dfc_rfm.pivot_table(index='year_month',
                                       columns='value_segment',
                                       values='customer_id', aggfunc='count')
customer_product.style.background_gradient(low=0.75, high=1.0)
```

Out[116]:

| value_segment | Lost | Low-value | Medium-value | Top | Super Top |
|---|---|---|---|---|---|
| year_month | | | | | |
| 2018-10 | 148 | 99 | 118 | 34 | 83 |
| 2018-11 | 167 | 74 | 110 | 33 | 44 |
| 2018-12 | 104 | 38 | 65 | 38 | 61 |
| 2019-01 | 69 | 18 | 42 | 36 | 17 |
| 2019-02 | 80 | 55 | 87 | 66 | 71 |
| 2019-03 | 90 | 66 | 153 | 31 | 66 |
| 2019-04 | 103 | 88 | 304 | 106 | 72 |
| 2019-05 | 30 | 53 | 346 | 251 | 46 |

| | | | | | |
|---|---|---|---|---|---|
| **2019-06** | 35 | 40 | 91 | 78 | 38 |
| **2019-07** | 29 | 38 | 119 | 95 | 31 |
| **2019-08** | 29 | 40 | 43 | 46 | 41 |
| **2019-09** | 8 | 39 | 70 | 62 | 38 |
| **2019-10** | 0 | 49 | 82 | 62 | 24 |
| **2019-11** | 0 | 2 | 53 | 110 | 52 |
| **2019-12** | 0 | 0 | 41 | 84 | 121 |
| **2020-01** | 0 | 0 | 40 | 106 | 123 |

### 10.3.2 Вывод по месяцам

Можно предположить наличие взаимосвязи между месяцем совершения покупок и количеством покупок у всех сегментов покупателей, причем у разных сегментов покупателей сезонность отличается.

Предположим, что есть различия между категориями покупателей по сезонному спросу на товар.

Месяца, в которых было совершено больше всего покупок у разных категорий товаров.

- 'Super Top' - 2019-12 и 2020-01 - более чем в два раза больше покупок, чем в другие периоды. Как одно из предположений - органический спрос на товар привлек больше покупателей, либо расширение ассортимента позволило привлечь больше покупателей. Необходимо проанализировать какие покупки совершали покупатели из этой категории.
- 'Top' -2019-05 самый большой пик продаж пришелся на этот месяц. 2019-04 и период 2019-11,2019-12, 2020-01 - также есть неплохие продажи.
- 'Medium-value' - 2019-03, 2019-04 и 2019-05 эти три месяца пики по количеству покупок.
- 'Low-value' - эти покупатели почти 3 месяца уже ничего не приобретают, при этом в аналогичные месяца прошлых лет продажи были. Пик продаж приходился на 2019-02, 2019-03, 2019-04 и 2018-10. Необходимо посмотреть какие товары пользовались спросом в эти месяца, чтобы можно было оживить этих покупателей рассылкой.
- 'Lost' - больше 5 месяцев ничего не покупают. Пики продаж были 2018-10, 2018-11, 2018-12, 2019-03 и 2019-04.

# 11 Анализ данных. Сегменты покупателей и категории товаров

## 11.1 Количество покупателей по сегментам купивших товары в товарных группах

In [117]:

```
customer_product = dfc_rfm.pivot_table(index='product_category',
                                       columns='value_segment',
                                       values='customer_id', aggfunc='count')
customer_product.style.background_gradient(low=0.75, high=1.0)
```

Out[117]:

| value_segment | Lost | Low-value | Medium-value | Top | Super Top |
|---|---|---|---|---|---|
| **product_category** | | | | | |
| **для ремонта** | 21 | 8 | 30 | 28 | 43 |
| **интерьер/декор** | 158 | 69 | 90 | 89 | 52 |
| **кухонная утварь** | 108 | 57 | 134 | 111 | 84 |
| **текстиль** | 24 | 27 | 45 | 44 | 64 |
| **хозяйственная утварь** | 161 | 131 | 268 | 263 | 283 |
| **хозяйственные сумки** | 33 | 40 | 83 | 78 | 151 |
| **цветы/сад** | 387 | 367 | 1114 | 625 | 251 |

## 11.2 Диаграмма сегменты покупателей с разбивкой по товарным группам

In [118]:

```
plt.figure(figsize=(10, 6))
sns.set (rc={'axes.facecolor':'#ffffff', 'figure.facecolor':'#dfe4ea'})
colors = ['#aaa69d', '#227093', '#ffda79', '#cd6133', '#706fd3', '#34ace0', '#ff5252']
ax = sns.countplot(x=dfc_rfm['value_segment'],
```

```
                    data=dfc_rfm,
                    hue='product_category',
                    alpha=0.9,
                    palette=colors)
                    #palette="Set1",
                    #palette='rocket')

ax.set_title('Сегменты покупателей с разбивкой по категориям товаров',
             fontsize = 14,
             fontweight ='bold')
plt.xlabel('Сегменты покупателей/категории товаров', fontsize = 14)
plt.ylabel('Количество покупок', fontsize = 14)
#ax.tick_params(axis='x', labelrotation=45) # градус наклона по оси x
plt.show()
```


In [119]:

```
dfc_rfm.head(2)
```

Out[119]:

| | date | customer_id | order_id | product | quantity | price | total | value_segment | year_month | product_category | order_weekday |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Алоэ Вера, d12, h30 | 1 | 142.0 | 142.0 | Lost | 2018-10 | цветы/сад | 0 |
| **1** | 2018-10-01 | ee47d746-6d2f-4d3c- | 68477 | Комнатное растение в горшке | 1 | 194.0 | 194.0 | Lost | 2018-10 | цветы/сад | 0 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 9622-c314 12542920 | | Кофе Арабика, d12, h25 | | | | | | | |

Категории товаров: `'цветы/сад'`, `'хозяйственная утварь'`, `'кухонная утварь'`, `'для ремонта'`, `'интерьер/декор'`, `'хозяйственные сумки'`, `'текстиль'`.

### 11.2.1 Вывод по товарам

Как видим по диаграмме: **'Сегменты покупателей с разбивкой по категориям товаров'** - почти все категорий покупателей покупают все виды товаров, однако соотношение этих товаров у них на первый взгляд разное.

Категория товаров `'цветы/сад'` и `'хозяйственная утварь'` в предпочтениях у всех сегментов покупателей. Категория товаров `'для ремонта'` непопулярна у всех сегментов покупателей.

Категория покупателей `Super Top` больше всего предпочитает категории `'хозяйственная утварь'`, а только вторая по популярности `'цветы/сад'`, у остальных же сегментов покупателей категория `'цветы/сад'` с большим отрывом находится в фаворитах, нежели остальные категории. Возможно это связано с тем, что в сегменте покупателей `Super Top` больше тех, кто совершает дорогостоящие покупки, которые как раз и находятся в товарной категории `'хозяйственная утварь'`, ведь у них третья по популярности товарная категория это `'хозяйственные сумки'`, которые тоже по цене являются достаточно высокой товарной группой для нашего интернет-магазина.

Сегмент покупателей - `Lost` больше всех закупался товарами из категории `'интерьер/декор'`. Возможно это был единоразовый спрос тот период времени, достаточно вспомнить какое количество муляжей яблок было закуплено. Возможно из ассортимента пропали эти товары. Если это так и у магазина есть возможность продолжить продажи ушедших из ассортимента товаров, то можно подумать над рассылкой рекламных предложений по типу "снова в продаже".

# 12 Анализ данных. Сегменты покупателей и выручка

## 12.1 Средний чек

В датасет `rfm_score` добавим столбец со средним чеком `avg_check`.

In [120]:

```
rfm_score['avg_check'] = rfm_score['monetary'] / rfm_score['frequency']
rfm_score.sample(n=3, random_state=1)
```

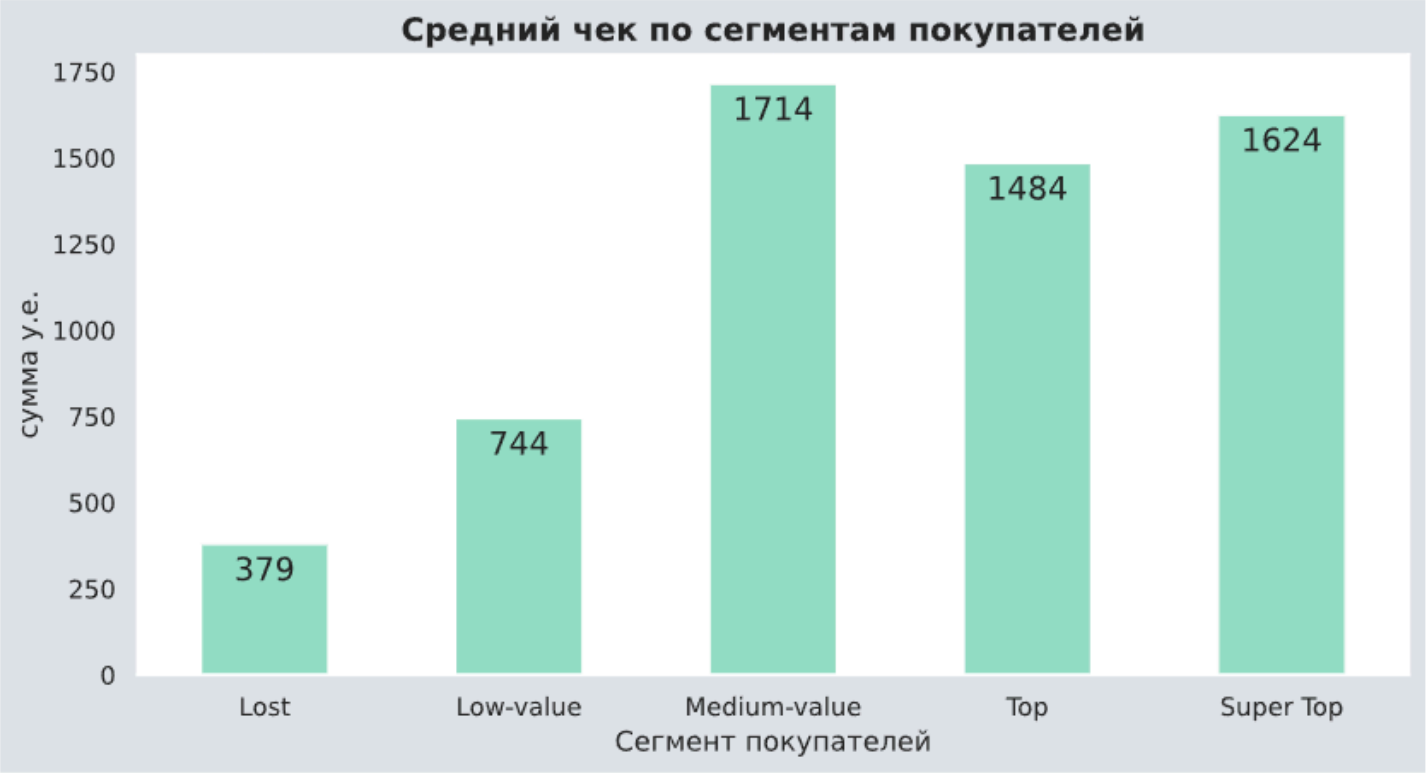
Out[120]:

| | customer_id | recency | frequency | monetary | r_score | f_score | m_score | rfm_score | value_segment | id_value_segment | avg_check |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1662** | b3091c0c-a36e-460f-9123-1a642d1ade5c | 85 | 2 | 1394.0 | 4 | 2 | 4 | 10 | Top | B | 697.0 |
| **2383** | fc9641ba-8f74-45de-a60a-342d2e0dd7af | 258 | 1 | 152.0 | 3 | 1 | 1 | 5 | Lost | E | 152.0 |
| **1398** | 96969c54-d375-48b3-b3bd-df07cb488a49 | 254 | 2 | 1508.0 | 3 | 2 | 4 | 9 | Top | B | 754.0 |

In [121]:

```
avg_check_df = rfm_score.groupby('value_segment',
).agg('avg_check').mean().plot(kind='bar', figsize=(10, 5),
                                                    color='#66CDAA',
alpha=0.7)

#добавляем подписи к столбцам со значением
for p in avg_check_df.patches:
   avg_check_df.annotate(format(round(p.get_height(), 3), '.0f'),
```

```
                    (p.get_x() + p.get_width() / 2., p.get_height()),
                    ha='center', va='center',
                    size=14,
                    xytext=(0, -12),
                    textcoords='offset points')

plt.title('Средний чек по сегментам покупателей', fontsize = 14, fontweight ='bold')
plt.xlabel('Сегмент покупателей')
plt.ylabel('сумма у.е.')
plt.xticks(rotation=360)
#plt.grid()
plt.show()
```




Средний чек по сегментам покупателей различается.

- 'Lost' - у этих покупателей средний чек 379 у.е.
- 'Low-value', - 744 у.е.
- 'Medium-value' - 1714 у.е.
- 'Top' - средний чек 1484 у.е.
- 'Super Top' - средний чек 1624 у.е.

### 12.1.1 Вывод по среднему чеку

Видим разницу в среднем чеке у разных сегментов покупателей. У сегмента Medium-value самый всоки средний чек.

In [122]:

```
dfc_rfm['value_segment'].unique()
```

Out[122]:

['Lost', 'Medium-value', 'Super Top', 'Top', 'Low-value']
Categories (5, object): ['Lost' < 'Low-value' < 'Medium-value' < 'Top' < 'Super Top']

# 13 Сегменты покупателей: Сезонность и товарная категория

Для удобства сформируем датасеты, куда отберем данные по категориям покупателей.

In [123]:

```
a_df = dfc_rfm.query('value_segment == "Super Top"')
```

```
b_df = dfc_rfm.query('value_segment == "Top"')
c_df = dfc_rfm.query('value_segment == "Medium-value"')
d_df = dfc_rfm.query('value_segment == "Low-value"')
e_df = dfc_rfm.query('value_segment == "Lost"')
```

## 13.1 Покупатели Super Top

Рассмотрим покупки этой категории клиентов интернет-магазина внимательнее. Какие товары они предпочитают покупать. Мы уже выяснили, что покупатели 'Super Top' в сезонах 2019-12 и 2020-01 - более чем в два раза больше покупок, чем в другие периоды. Посмотрим на категории товаров, которые были приобретены в эти месяцы, а также отдельно рассмотрим какие категории товаров предпочитают покупать во всех сезонах.

### 13.1.1 Топ - товаров сегмента покупателей Super Top.

Посмотрим топ-товаров во всех сезонах и в сезонах когда была повышенная активность.

In [124]:

```
#сгруппируем топ позиций товаров в месяцах спроса по цене
a_df_top_product = a_df.query('year_month == "2019-12" | year_month ==
"2020-01"').groupby('product', as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
a_df_top_product =
a_df_top_product.rename(columns={'order_id':'count'}).sort_values('total', ascending =
False)
a_df_top_product.head(5).style.background_gradient()

#a_df_top_product.style.background_gradient()
```

Out[124]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **146** | Сумка-тележка хозяйственная Rolser IMX006 bassi Logic Tour бордовая | хозяйственные сумки | 1 | 2 | 15358.000000 |
| **153** | Сушилка Meliconi Stendy Junior | хозяйственная утварь | 2 | 2 | 11188.000000 |
| **69** | Мусорный контейнер Hailo BigBin Swing 45 0845-010 45 л хром | хозяйственная утварь | 2 | 2 | 11024.000000 |
| **151** | Сумка-тележка хозяйственная Rolser Paris, бордовая, PEP001 bassi JOY | хозяйственные сумки | 2 | 2 | 8234.000000 |
| **150** | Сумка-тележка хозяйственная Rolser MNB019 rojo LOGIC DOS+2 красная | хозяйственные сумки | 1 | 1 | 8077.000000 |

In [125]:

```
#сгруппируем топ позиций товаров все сезоны по цене
a_df_top_product_all = a_df.groupby('product', as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
a_df_top_product_all =
a_df_top_product_all.rename(columns={'order_id':'count'}).sort_values('total',
ascending = False)

a_df_top_product_all.head(5).style.background_gradient()
```

Out[125]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **436** | Сушилка Meliconi Stendy Junior | хозяйственная утварь | 5 | 5 | 27970.000000 |
| **419** | Сумка-тележка хозяйственная Rolser IMX006 bassi Logic Tour бордовая | хозяйственные сумки | 2 | 3 | 23037.000000 |
| **430** | Сумка-тележка хозяйственная Rolser Pack Gloria Logic RG серая, PAC036 marengo LOGIC RG | хозяйственные сумки | 3 | 3 | 19674.000000 |
| **220** | Одеяло Wellness T142 белое темостеганое 140x205 см чехол 100% полиэстер 200 г/м 4690659000306 | текстиль | 2 | 11 | 17248.000000 |
| **428** | Сумка-тележка хозяйственная Rolser MNB019 rojo LOGIC DOS+2 красная | хозяйственные сумки | 2 | 2 | 16154.000000 |

### 13.1.2 Super Top: Распределение товарных категорий по сезонам

In [126]:

```
#Количество покупателей по сегменту А купивших товары по всем месяцам
plt.figure(figsize=(12, 6))
colors = ['#aaa69d', '#227093', '#ffda79', '#cd6133', '#706fd3', '#34ace0', '#ff5252']
sns.set (rc={'axes.facecolor':'#ffffff', 'figure.facecolor':'#dfe4ea'})
colors_customers = ['#458B74', '#FF7F24', '#2ed573', '#00FFFF', '#EE1289']
ax = sns.countplot(x=a_df['year_month'].sort_values(ascending=False),
                   data=a_df,
                   hue='product_category',
                   alpha=1,
                   palette=colors)
ax.set_title('"Super Top": Распределение товарных категорий по сезонам', fontsize =
12,
             fontweight ='bold')
plt.ylabel('количество покупателей', fontsize = 10)
plt.xlabel('', fontsize = 10)
ax.tick_params(axis='x', labelrotation=45) # градус наклона по оси x
plt.show();
```

Больше всего заказов в категориях цветы/сад и хозяйственная утварь, а также хозяйственные сумки. Остальные товарные категории также присутствуют.

# Выводы и рекомендации по Super Top

Рассмотрели покупки категории клиентов `Super Top` в сезоны с наибольшим ростом покупок и в остальные сезоны.

В сезонах 2019-12 и 2020-01 у этой категории покупателей выросло количество покупок - более чем в два раза, чем в другие периоды. Рост покупателей именно в этом сегменте дало не количество покупок, а стоимость приобретаемого товара и недавность приобретения. Это определило этих покупателей в эту категорию.

В сезон 2019-12 и 2020-01 рост покупок произошел за счет категорий товаров: `хозяйственная утварь` и `цветы/сад`.

В сезон 2019-12 и 2020-01 Топ-5 товаров, которые принесли наибольшую общую сумму:

- Сумка-тележка хозяйственная Rolser IMX006 bassi Logic Tour бордовая хозяйственные сумки - 15358у.е.
- Сушилка Meliconi Stendy Junior хозяйственная утварь - 11188.000000
- Мусорный контейнер Hailo BigBin Swing 45 0845-010 45 л хром хозяйственная утварь - 11024у.е.
- Сумка-тележка хозяйственная Rolser Paris, бордовая, PEP001 bassi JOY хозяйственные сумки - 8234у.е.
- Сумка-тележка хозяйственная Rolser MNB019 rojo LOGIC DOS+2 красная хозяйственные сумки - 8077у.е.

Во всех сезонах в топ-5 товаров, которые принесли или наибольшую общую сумму следующие товары:

- Сушилка Meliconi Stendy Junior хозяйственная утварь - 27970у.е.
- Сумка-тележка хозяйственная Rolser IMX006 bassi Logic Tour бордовая хозяйственные сумки - 23037у.е.
- Сумка-тележка хозяйственная Rolser Pack Gloria Logic RG серая, PAC036 marengo LOGIC RG хозяйственные сумки- 19674у.е.
- Одеяло Wellness T142 белое темостеганое 140х205 см чехол 100% полиэстер 200 г/м 4690659000306 текстиль- 17248у.е.
- Сумка-тележка хозяйственная Rolser MNB019 rojo LOGIC DOS+2 красная хозяйственные сумки - 16154у.е.

Рекомендации: покупатели сегмента `Super Top` можно предлагать персонализированно товары из категории `хозяйственная утварь` и `цветы/сад`. При этом на товары из категории `цветы/сад` в сезонах 2019-05 и 2019-04 был высокий спрос, можно можно напомнить о нашем магазине с предложением сезонных товаров из этой категории. Также можно включать в основную рассылку как дополнение дорогие товары из раздела `хозяйственные сумки` и `текстиль`. Учитывая, что покупатели в этой товарной категории чаще всего покупают дорогостоящий товар, то включение в рассылку дорогих товаров (по ассортиментному ряду нашего магазина) также целесообразно. Выручка больше от продажи трех хозяйственных сумок за 20000 у.е., чем 300 пакетов семян на 3000 у.е. Если конечно не брать в расчет себестоимость и прочие экономические показатели, которых у нас нет.

## 13.2 Покупатели Top

Рассмотрим покупки этой категории клиентов интернет-магазина внимательнее. Какие товары они предпочитают покупать. Мы уже выяснили, что покупатели 'Top' в сезоне 2019-05 случился самый большой пик продаж. в 2019-04 и период 2019-11,2019-12, 2020-01 - также есть неплохие продажи.

Посмотрим на категории товаров, которые были приобретены в эти месяцы, а также отдельно рассмотрим какие категории товаров предпочитают покупать во всех сезонах.

In [127]:

```
#сгруппируем топ позиций товаров в месяцах спроса по цене
b_df_top_product = b_df.query('year_month == "2019-05"').groupby('product',
as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
```

```
b_df_top_product =
b_df_top_product.rename(columns={'order_id':'count'}).sort_values('total', ascending =
False)
b_df_top_product.head(5).style.background_gradient()
```

Out[127]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **46** | Муляж ЯБЛОКО 9 см красное | интерьер/декор | 1 | 300 | 15300.000000 |
| **169** | Сумка-тележка хозяйственная Andersen Treppensteiger Scala Shopper, Hera, черная 119-004-80 | хозяйственные сумки | 1 | 1 | 6449.000000 |
| **170** | Сумка-тележка хозяйственная Rolser PAR015 mandarina DOS+2 оранжевая | хозяйственные сумки | 1 | 1 | 6097.000000 |
| **21** | Гладильная доска Hausmann HM-3170 Home Art 122x42 см | хозяйственная утварь | 1 | 1 | 4874.000000 |
| **95** | Полки QWERTY Мадрид белый/чёрный 50x30x12 см 2 штуки и 30x24x12 см 2 штуки 72019 | интерьер/декор | 1 | 1 | 4312.000000 |

In [128]:

```
#сгруппируем топ позиций товаров все сезоны
b_df_top_product_all = b_df.groupby('product', as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
b_df_top_product_all =
b_df_top_product_all.rename(columns={'order_id':'count'}).sort_values('total',
ascending = False)
b_df_top_product_all.head(5).style.background_gradient()
```

Out[128]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **69** | Вешалки мягкие для деликатных вещей 3 шт шоколад | хозяйственная утварь | 1 | 334 | 49432.000000 |
| **258** | Муляж ЯБЛОКО 9 см красное | интерьер/декор | 1 | 300 | 15300.000000 |
| **640** | Сушилка уличная Leifheit 85210 LINOMATIC V 400 40 м 175x237x237 см зеленая | хозяйственная утварь | 1 | 1 | 14917.000000 |
| **581** | Сумка-тележка хозяйственная Andersen Scala Shopper, Lini, синяя 112-108-90 | хозяйственные сумки | 3 | 3 | 13722.000000 |
| **582** | Сумка-тележка хозяйственная Andersen Treppensteiger Scala Shopper, Hera, черная 119-004-80 | хозяйственные сумки | 2 | 2 | 12898.000000 |

In [129]:

```
#Количество покупателей по сегменту B купивших товары по всем месяцам
plt.figure(figsize=(12, 6))
colors = ['#aaa69d', '#227093', '#ffda79', '#cd6133', '#706fd3', '#34ace0', '#ff5252']
sns.set (rc={'axes.facecolor':'#ffffff', 'figure.facecolor':'#dfe4ea'})
colors_customers = ['#458B74', '#FF7F24', '#2ed573', '#00FFFF', '#EE1289']
ax = sns.countplot(x=b_df['year_month'].sort_values(ascending=False),
```

```
                    data=b_df,
                    hue='product_category',
                    alpha=1,
                    palette=colors)
ax.set_title('Сегмент "Top": Распределение товарных категорий по сезонам', fontsize =
12,
             fontweight ='bold')
plt.ylabel('количество покупателей', fontsize = 10)
plt.xlabel('', fontsize = 10)
ax.tick_params(axis='x', labelrotation=45) # градус наклона по оси x
plt.show();
```


Больше всего заказов в категориях `цветы/сад`, более стабильный спрос на категорию `хозяйственная утварь` - скачки есть, но не такие интенсивные, как у категории `цветы/сад`. Остальные товарные категории присутствуют в покупках, но минимально.

## Выводы и рекомендации по Top

Рассмотрели покупки категории клиентов `Top` в сезоны с наибольшим ростом покупок и в остальные сезоны.

В сезоне 2019-05 случился самый большой пик продаж. Обеспечил его товар из категории `цветы/сад`. Товар в этой категории неплохо продавался и в сезоны 2019-04, 2019-06, 2019-07, 2019-11,2019-12, 2020-01.

Покупки товаров из категории `хозяйственная утварь` случаются стабильно, есть небольшие перепады из месяца в месяц. В сезоне 2020-01 и 2019-11 есть небольшой рост в категории `кухонная утварь`.

Можно сказать, что во всех сезонах рост покупок происходит в основном в большей части за счет категорий товаров: `цветы/сад`.

В сезон 2019-05 товары, которые принесли наибольшую общую сумму:

- Муляж ЯБЛОКО 9 см красное интерьер/декор - 300шт. 15300 у.е.
- Сумка-тележка хозяйственная Andersen Treppensteiger Scala Shopper, Hera, черная 119-004-80хозяйственные сумки- 6449у.е.
- Сумка-тележка хозяйственная Rolser PAR015 mandarina DOS+2 оранжевая хозяйственные сумки -6097у.е.
- Гладильная доска Hausmann HM-3170 Home Art 122x42 см хозяйственная утварь 4874у.е.

- Полки QWERTY Мадрид белый/чёрный 50x30x12 см 2 штуки и 30x24x12 см 2 штуки 72019 интерьер/декор - 4312у.е.

Во всех сезонах в топ-5 товаров, которые принесли или наибольшую общую сумму или заказаны несколько раз в разных заказах следующие товары:

- Вешалки мягкие для деликатных вещей 3 шт шоколад хозяйственная утварь - 49432у.е.
- Муляж ЯБЛОКО 9 см красное интерьер/декор шт. 300 15300у.е.
- Сушилка уличная Leifheit 85210 LINOMATIC V 400 40 м 175x237x237 см зеленая хозяйственная утварь- 14917у.е.
- Сумка-тележка хозяйственная Andersen Scala Shopper, Lini, синяя 112-108-90 хозяйственные сумки - 13722у.е.
- Сумка-тележка хозяйственная Andersen Treppensteiger Scala Shopper, Hera, черная 119-004-80 хозяйственные сумки -12898у.е.

Рекомендации: покупателям сегмента `Top` можно предлагать персонализированно товары из категории `цветы/сад`, учитывать, что с сезона 2019-04 по 2019-07 с пиком в мае был высокий спрос, можно можно напомнить о нашем магазине с предложением сезонных товаров из этой категории.

Также можно включать в основную рассылку как дополнение дорогие товары из раздела `хозяйственная утварь`, `кухонная утварь` и `интерьер/декор`. Учитывая, что покупатели в этой товарной категории чаще покупают дорогостоящий товар, включение в рассылку дорогих товаров (по ассортиментному ряду нашего магазина) целесообразно. Основной упор в сезонной рассылке сделать на товары из категории цветы.

## 13.3 Покупатели Medium-value

Рассмотрим покупки этой категории клиентов интернет-магазина внимательнее. Какие товары они предпочитают покупать. Мы уже выяснили, что покупатели `'Medium-value'` в сезонах - 2019-03, 2019-04 и 2019-05 имеют пики по количеству покупок.

Посмотрим на категории товаров, которые были приобретены в эти месяцы, а также отдельно рассмотрим какие категории товаров предпочитают покупать во всех сезонах в сегменте `'Medium-value'`.

In [130]:

```
#сгруппируем топ позиций товаров в месяцах спроса по цене
c_df_top_product = c_df.query('year_month == "2019-05" | year_month == "2019-04"\
| year_month == "2019-03"').groupby('product', as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
c_df_top_product =
c_df_top_product.rename(columns={'order_id':'count'}).sort_values('total', ascending =
False)

c_df_top_product.head(5).style.background_gradient()
```

Out[130]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **366** | Сумка-тележка хозяйственная Andersen Royal Shopper, Hera, синяя 166-004-90 | хозяйственные сумки | 1 | 1 | 8737.000000 |
| **148** | Пеларгония зональная диам. 12 см красная махровая | цветы/сад | 8 | 44 | 7094.000000 |
| **367** | Сумка-тележка хозяйственная Andersen Scala Shopper Plus, Lini, синяя 133-108-90 | хозяйственные сумки | 1 | 1 | 6149.000000 |
| **364** | Сумка-тележка 3-х колесная Gimi Tris красная | хозяйственные сумки | 1 | 2 | 5398.000000 |

| | | | | | |
|---|---|---|---|---|---|
| **386** | Урна-пепельница из нержавеющей стали, Hobbyka/Хоббика, 83*38см, ПА022 | интерьер/декор | 1 | 1 | 5287.000000 |

In [131]:

```
#сгруппируем топ позиций товаров все месяца
c_df_top_product_all = c_df.groupby('product', as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
c_df_top_product_all =
c_df_top_product_all.rename(columns={'order_id':'count'}).sort_values('total',
ascending = False)

c_df_top_product_all.head(5).style.background_gradient()
```

Out[131]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **557** | Простынь вафельная 200x180 см WELLNESS RW180-01 100% хлопок | текстиль | 2 | 30 | 53232.000000 |
| **856** | Тележка багажная DELTA ТБР-22 синий грузоподъемность 20 кг сумка и 50 кг каркас РОССИЯ | хозяйственные сумки | 1 | 57 | 32718.000000 |
| **359** | Набор ножей Attribute CHEF 5 предметов AKF522 | кухонная утварь | 1 | 64 | 29248.000000 |
| **875** | Урна уличная "Гео", Hobbyka/Хоббика, 59*37,5см, сталь | интерьер/декор | 1 | 5 | 24370.000000 |
| **337** | Муляж ЯБЛОКО 9 см красное | интерьер/декор | 2 | 310 | 16930.000000 |

In [132]:

```
#Количество покупателей по сегменту С купивших товары по всем месяцам
plt.figure(figsize=(12, 6))
colors = ['#aaa69d', '#227093', '#ffda79', '#cd6133', '#706fd3', '#34ace0', '#ff5252']
sns.set (rc={'axes.facecolor':'#ffffff', 'figure.facecolor':'#dfe4ea'})
colors_customers = ['#458B74', '#FF7F24', '#2ed573', '#00FFFF', '#EE1289']
ax = sns.countplot(x=c_df['year_month'].sort_values(ascending=False),
                   data=c_df,
                   hue='product_category',
                   alpha=1,
                   palette=colors)
ax.set_title('"Medium-value": Распределение товарных категорий по сезонам', fontsize =
12,
             fontweight ='bold')
plt.ylabel('количество покупателей', fontsize = 10)
ax.tick_params(axis='x', labelrotation=45) # градус наклона по оси x
plt.xlabel('', fontsize = 10)
plt.show();
```

Больше всего заказов в категориях `цветы/сад`, особенно в период с 2019-03 по 2019-07 с пиком 2019-05 и 2019-04. есть небольшой спрос на категорию `хозяйственная утварь` -Остальные товарные категории присутствуют в покупках минимально. При этом в сезонах 2018-10 спрос был более разнообразен по другим категориям товаров. и до сезона 2019-02 спрос на `цветы/сад` был низким.

## Выводы и рекомендации по `Medium-value`

Рассмотрели покупки категории клиентов `Medium-value` в сезоны с наибольшим ростом покупок и в остальные сезоны.

В сезоне 2019-05 случился самый большой пик продаж. Обеспечил его товар из категории `цветы/сад`. Товар в этой категории неплохо продавался и в сезоны 2019-03, 2019-04, 2019-06, 2019-07.

Можно сказать, что во всех сезонах рост покупок происходит в основном в большей части за счет категорий товаров: `цветы/сад`.

В сезон 2019-05, 2019-04 и 2019-03 - Топ-5 товаров, которые принесли наибольшую общую сумму:

- Сумка-тележка хозяйственная Andersen Royal Shopper, Hera, синяя 166-004-90 хозяйственные сумки - 8737у.е.
- Пеларгония зональная диам. 12 см красная махровая цветы/сад - 7094у.е.
- Сумка-тележка хозяйственная Andersen Scala Shopper Plus, Lini, синяя 133-108-90 хозяйственные сумки - 6149у.е.
- Сумка-тележка 3-х колесная Gimi Tris красная хозяйственные сумки - 5398у.е.
- Урна-пепельница из нержавеющей стали, Hobbyka/Хоббика, 83*38см, ПА022 интерьер/декор - 5287у.е.

Во всех сезонах в топ-5 товаров, которые принесли или наибольшую общую сумму или заказаны несколько раз в разных заказах следующие товары:

- Простынь вафельная 200х180 см WELLNESS RW180-01 100% хлопок текстиль - 53232 у.е.
- Тележка багажная DELTA ТБР-22 синий грузоподъемность 20 кг сумка и 50 кг каркас РОССИЯ хозяйственные сумки-32718 у.е.
- Набор ножей Attribute CHEF 5 предметов AKF522 кухонная утварь-29248 у.е.
- Урна уличная "Гео", Hobbyka/Хоббика, 59*37,5см, сталь интерьер/декор - 24370 у.е.
- Муляж ЯБЛОКО 9 см красное интерьер/декор - 16930 у.е.

Рекомендации: покупателям сегмента `Medium-value` можно предлагать персонализированно товары из категории `цветы/сад`, учитывать, что с сезона 2019-03 по 2019-07 с пиком в мае был высокий спрос, можно напомнить о нашем магазине с предложением сезонных товаров из этой категории.

Также можно включать как дополнение в основную рассылку дорогие товары из раздела `хозяйственная утварь`. Основной упор в сезонной рассылке сделать на товары из категории цветы. Можно изучить сезонные особенности высадки семян/черенков и предлагать этот товар к этим сезонам.

### 13.4 Покупатели Low-value

Рассмотрим покупки этой категории клиентов интернет-магазина внимательнее. Какие товары они предпочитают покупать. Мы уже выяснили, что покупатели `'Low-value'` почти 3 месяца январь2020, декабрь 2019 ии ноябрь 2019 уже ничего не приобретают, при этом в аналогичные месяца прошлых лет продажи были. Пик продаж приходился на 2019-02, 2019-03, 2019-04 и 2018-10.

Необходимо посмотреть какие товары пользовались спросом в эти месяца, чтобы можно было оживить этих покупателей рассылкой.

Рассмотрим также какие категории товаров предпочитают покупать во всех сезонах в сегменте `Low-value`.

In [133]:

```
#сгруппируем топ позиций товаров в месяцах спроса по цене
d_df_top_product = d_df.query('year_month == "2019-02" | year_month == "2019-03"\
| year_month == "2019-04" | year_month == "2018-10"').groupby('product',
as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
d_df_top_product = d_df_top_product.rename(columns={'order_id':'count'})
d_df_top_product = d_df_top_product.sort_values('total', ascending = False)
d_df_top_product.head(5).style.background_gradient()
```

Out[133]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **178** | Сумка-тележка 2-х колесная Gimi Argo синяя | хозяйственные сумки | 9 | 9 | 9745.000000 |
| **198** | Тележка багажная DELTA ТБР-20 коричневый с оранжевым грузоподъемность 25 кг сумка и 50 кг каркас РОССИЯ | хозяйственные сумки | 5 | 5 | 3649.000000 |
| **199** | Тележка багажная DELTA ТБР-20 черный с серым грузоподъемность 25 кг сумка 50 кг каркас РОССИЯ | хозяйственные сумки | 5 | 5 | 3642.000000 |
| **180** | Сумка-тележка 2-х колесная Gimi Argo черная | хозяйственные сумки | 2 | 2 | 2174.000000 |
| **177** | Сумка-тележка 2-х колесная Gimi Argo красная | хозяйственные сумки | 2 | 2 | 2174.000000 |

In [134]:

```
#сгруппируем топ позиций товаров во всех месяцах
d_df_top_product_all = d_df.groupby('product', as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
d_df_top_product_all =
d_df_top_product_all.rename(columns={'order_id':'count'}).sort_values('total',
ascending = False)

d_df_top_product_all.head(5).style.background_gradient()
```

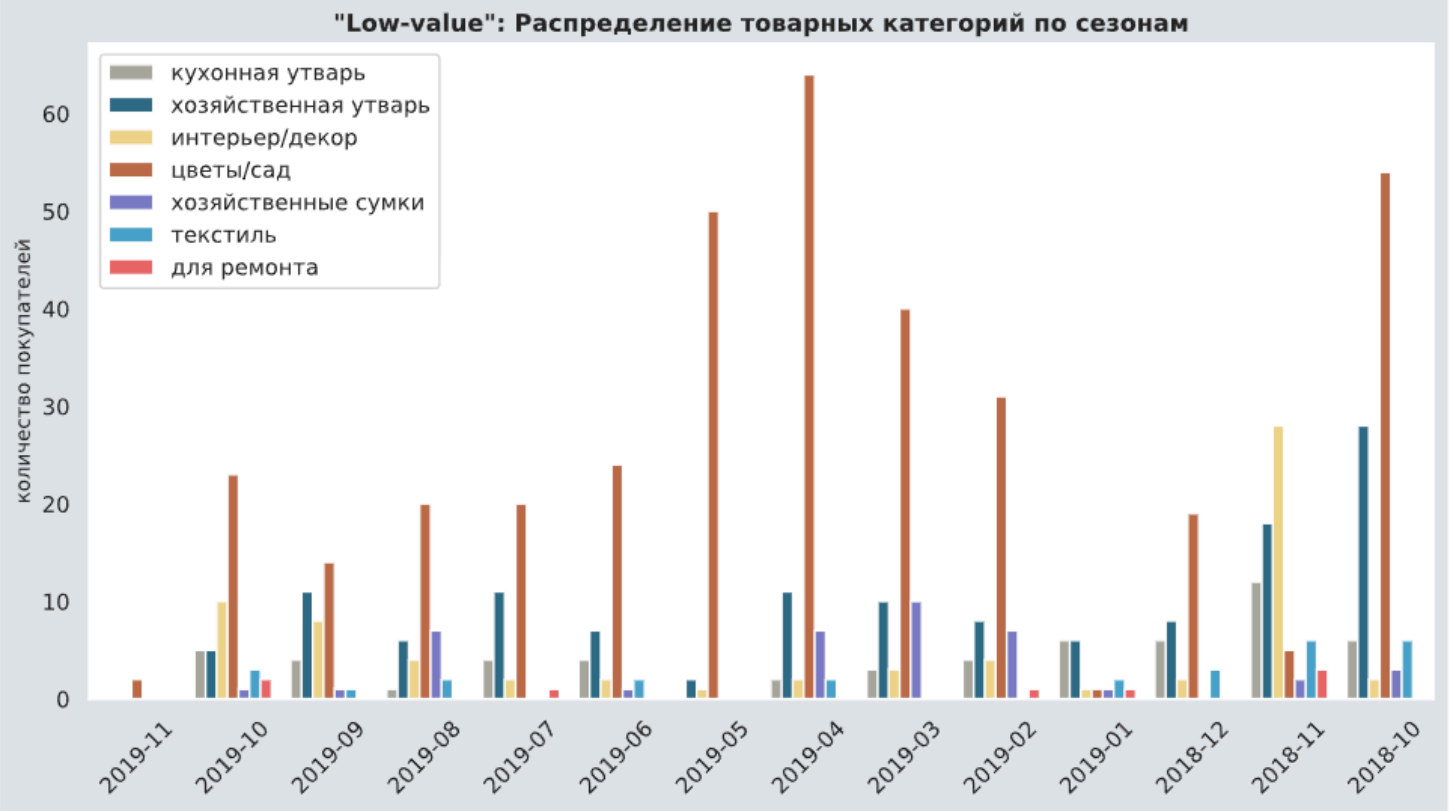
Out[134]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **387** | Сумка-тележка 2-х колесная Gimi Argo синяя | хозяйственные сумки | 9 | 9 | 9745.000000 |
| **438** | Тележка багажная DELTA ТБР-20 коричневый с оранжевым грузоподъемность 25 кг сумка и 50 кг каркас РОССИЯ | хозяйственные сумки | 5 | 5 | 3649.000000 |
| **439** | Тележка багажная DELTA ТБР-20 черный с серым грузоподъемность 25 кг сумка 50 кг каркас РОССИЯ | хозяйственные сумки | 5 | 5 | 3642.000000 |
| **396** | Сушилка для белья Meliconi Miss Stendy | хозяйственная утварь | 2 | 2 | 3598.000000 |
| **407** | Сушилка для белья напольная НИКА СБП1/С 18 м | хозяйственная утварь | 5 | 6 | 3594.000000 |

In [135]:

```
#Количество покупателей по сегменту D купивших товары по всем месяцам
plt.figure(figsize=(12, 6))
colors = ['#aaa69d', '#227093', '#ffda79', '#cd6133', '#706fd3', '#34ace0', '#ff5252']
sns.set (rc={'axes.facecolor':'#ffffff', 'figure.facecolor':'#dfe4ea'})
colors_customers = ['#458B74', '#FF7F24', '#2ed573', '#00FFFF', '#EE1289']
ax = sns.countplot(x=d_df['year_month'].sort_values(ascending=False),
                   data=d_df,
```

```
                  hue='product_category',
                  alpha=1,
                  palette=colors)
ax.set_title('"Low-value": Распределение товарных категорий по сезонам', fontsize =
12,
             fontweight ='bold')
plt.ylabel('количество покупателей', fontsize = 10)
plt.legend(loc='upper left') # сместить легенду вверх влево
plt.xlabel('', fontsize = 10)
ax.tick_params(axis='x', labelrotation=45) # градус наклона по оси x
plt.show();
```




## Выводы и рекомендации по `Low-value`

Рассмотрели покупки категории клиентов `Low-value` по сезонам и категориям товаров.

В период с 2019-03 по 2019-05и и и 2018-10 больше всего заказов в категориях `цветы/сад`. В период с 2019-02 по 2019-10 наблюдался стабильный спрос на товары из категории `цветы/сад`, а также в 2019-10, 2019-09 и особенно 2018-11 был спрос на товары категории `'интерьер/декор'`. Категория товаров `хозяйственная утварь` стабильно пользовалась спросом, особенно в октябре 2018.

Рекомендации: покупателей сегмента `Low-value` можно попробовать оживить рассылкой по категории товаров `цветы/сад`, учитывать, что с сезона 2019-03 и 2019-04, а также 2018-11 и 2018-10 был спрос на эти товары.

Однако, при ограниченном рекламном бюджете - на этой категории покупателей можно сэкономить, ограничив количество рассылки и т.д.

Также можно включать как дополнение в основную рассылку не дорогие товары из раздела `хозяйственная утварь`, `'интерьер/декор'`. Учитывая, что покупатели в этой товарной категории не отличились покупками дорогих товаров, поэтому включение в рассылку дорогих товаров (по ассортиментному ряду нашего магазина) нецелесообразно. Основной упор в сезонной рассылке сделать на товары из категории цветы. Можно изучить сезонные особенности высадки семян/черенков и предлагать этот товар к этим сезонам.

### 13.5 Покупатели Lost

Рассмотрим покупки этой категории клиентов интернет-магазина внимательнее. Какие товары они предпочитают покупать. Мы уже выяснили, что покупатели `'Lost'` больше 5 месяцев уже ничего не приобретают, при этом в аналогичные месяца прошлых лет продажи были. Пик продаж приходился на 2018-10, 2018-11, 2018-12, 2019-03 и 2019-04.

Необходимо посмотреть какие товары пользовались спросом у этой категории покупателей, чтобы можно было оживить этих покупателей рассылкой.

Рассмотрим также какие категории товаров предпочитают покупать во всех сезонах в сегменте `Lost`.

In [136]:

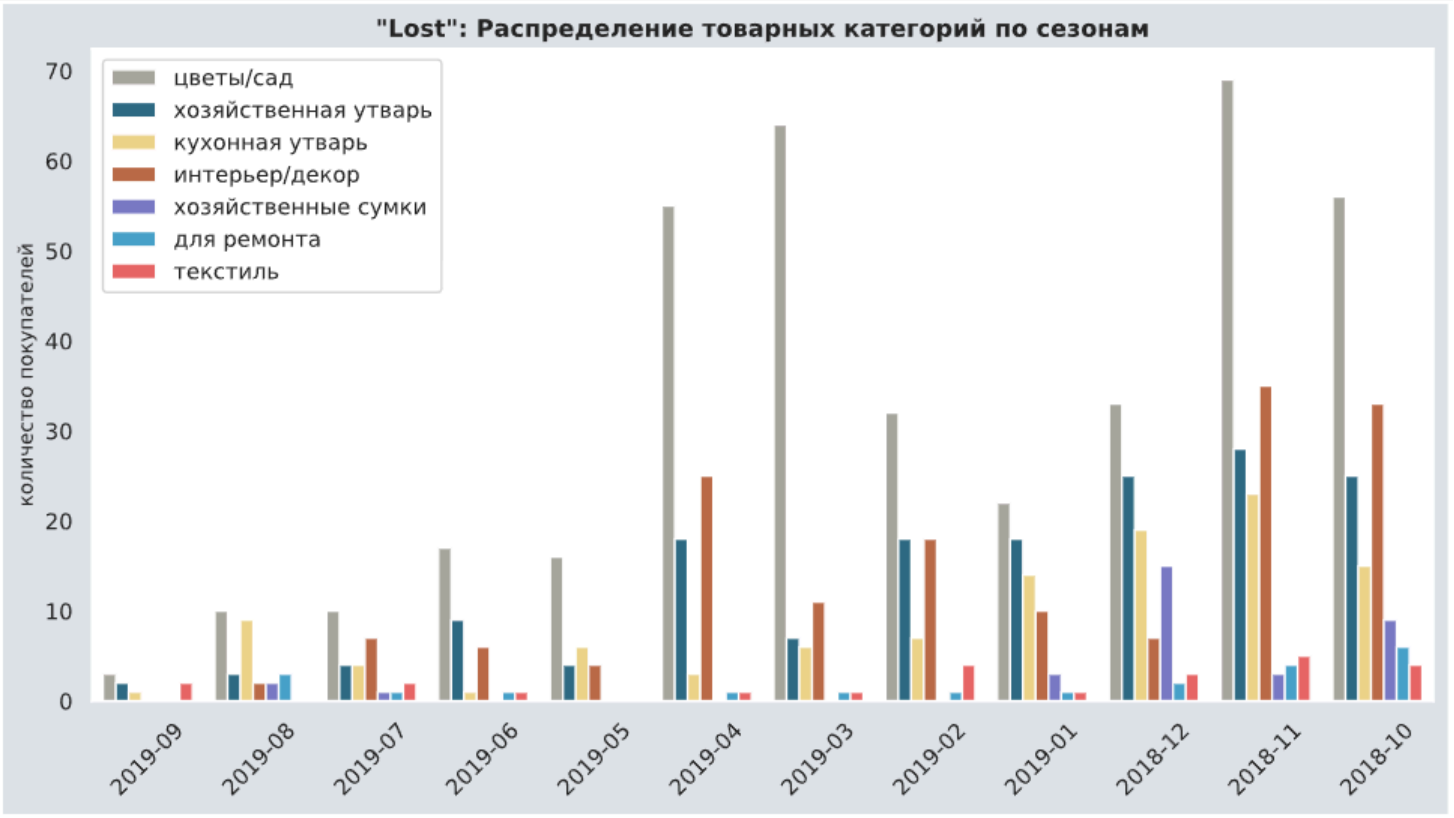
```
#сгруппируем топ позиций товаров во всех месяцах
e_df_top_product_all = e_df.groupby('product', as_index=False)\
.agg({'product_category' : pd.Series.mode, 'order_id': 'count', 'quantity': 'sum',
'total': 'sum'})
#переименуем колонки
e_df_top_product_all = e_df_top_product_all.rename(columns={'order_id':'count'})
e_df_top_product_all = e_df_top_product_all.sort_values('total', ascending = False)
e_df_top_product_all.head(5).style.background_gradient()
```

Out[136]:

| | product | product_category | count | quantity | total |
|---|---|---|---|---|---|
| **441** | Сумка-тележка 2-х колесная Gimi Argo синяя | хозяйственные сумки | 14 | 14 | 14686.000000 |
| **270** | Новогоднее дерево Ель сербская d-21 см h-60 см | цветы/сад | 6 | 6 | 6294.000000 |
| **452** | Сушилка для белья настенная Zalger Prima 510-720 веревочная 7 линий 25 м | хозяйственная утварь | 12 | 12 | 3588.000000 |
| **304** | Пеларгония зональная диам. 12 см сиреневый полумахровый | цветы/сад | 13 | 14 | 2404.000000 |
| **454** | Сушилка для белья потолочная Zalger Lift Comfort 520 180 см 9 м | хозяйственная утварь | 3 | 3 | 2247.000000 |

In [137]:

```
#Количество покупателей по сегменту E купивших товары по всем месяцам
plt.figure(figsize=(12, 6))
colors = ['#aaa69d', '#227093', '#ffda79', '#cd6133', '#706fd3', '#34ace0', '#ff5252']
sns.set (rc={'axes.facecolor':'#ffffff', 'figure.facecolor':'#dfe4ea'})
colors_customers = ['#458B74', '#FF7F24', '#2ed573', '#00FFFF', '#EE1289']
ax = sns.countplot(x=e_df['year_month'].sort_values(ascending=False),
                   data=e_df,
                   hue='product_category',
                   alpha=1,
                   palette=colors)
ax.set_title('"Lost": Распределение товарных категорий по сезонам', fontsize = 12,
             fontweight ='bold')
plt.ylabel('количество покупателей', fontsize = 10)
plt.legend(loc='upper left') # сместить легенду вверх влево
plt.xlabel('', fontsize = 10)
ax.tick_params(axis='x', labelrotation=45) # градус наклона по оси x
plt.show();
```




## Выводы и рекомендации по `Lost`

Рассмотрели покупки категории клиентов `Lost` по сезонам и категориям товаров.

В период с 2019-04, 2019-03 и 2018-11, 2018-10 больше всего заказов в категориях `цветы/сад`, а также в 2018-10, 2018-11 был спрос на товары категории `'интерьер/декор'`. Категория товаров `хозяйственная утварь`, 'кухонная утварь' стабильно пользовалась спросом.

Рекомендации: покупателей сегмента `Lost` можно попробовать оживить рассылкой по категории товаров `цветы/сад`, учитывать, что в сезоны 2019-03 и 2019-04, а также 2018-11 и 2018-10 был спрос на эти товары.

Однако, при ограниченном рекламном бюджете - на этой категории покупателей можно сэкономить, ограничив количество рассылки и т.д.

Также можно включать как дополнение в основную рассылку не дорогие товары - не дороже 1000 у.е. за шт из раздела `хозяйственная утварь`, `'интерьер/декор'`,'кухонная утварь'.

# 14 Общие выводы

## Результаты анализа данных

Исследуемый период: с 2018-10-01 00:00:00 по 2020-01-31 15:00:00.

- 5521 строчек заказов
- 3491 заказов
- 2412 покупателей
- 2333 единиц товарной номенклатуры
- 9 у.е. минимальная цена на товар
- 14917 у.е. максимальная цена на товар
- 531 у.е средняя цена на товар.
- 150 у.е. медианная цена на товар
- 524 у.е. цена у 75% товара
- 1 покупка у 60%
- 2 покупки у 39%
- 3 и более покупок у 1%

## Категоризация товаров

Товарную номенклатуру поделили на товарные категории. Список категорий товара с количеством товарной номенклатуры:

- цветы/сад - 915 ед.
- хозяйственная утварь - 551 ед
- кухонная утварь - 320 ед
- интерьер/декор - 218 ед
- текстиль - 135 ед
- хозяйственные сумки - 110
- для ремонта - 84 ед.

## Информация к сведению:

- С ноября 2019 идет параллельный рост покупателей и заказов - это положительная динамика, однако нет повторных заказов - покупки совершаются один раз.
- Самые популярные товары относятся к категории «цветы/сад».

## Сегментация покупателей

Провели оценку покупателей по давности, частоте и сумме покупок. Оценивали результат по баллам от 1 до 5 за каждый показатель. Баллы суммировали и получили RFM оценку по каждому покупателю. На основе этой оценки провели сегментацию и получили 5 кластеров покупателей.

- Super Top
- Top
- Medium-value
- Low-value
- Lost

Корректность разбивки на сегменты проверены гипотезами.

- Итоги статистического теста гипотезы № 1: - нет оснований отвергать гипотезу о том, что есть различия между категориями покупателей в среднем чеке. Во всех комбинациях сегментов покупателей статистически значимая разница между средними чеками сегментов покупателей выявлена.
- Итоги статистического теста гипотезы № 2: - нет оснований отвергать гипотезу о различии между категориями покупателей по частоте покупок.Во всех комбинациях сегментов покупателей статистически значимая разница между количеством покупок среди разных сегментов покупателей выявлена.

## Профили сегментов покупателей

### `Super Top`

- оценка по RFM диапазон: 11-14
- давность покупок медиана дней: 38
- средний чек у.е.: 1568
- медианный чек у.е.: 1405
- средняя общая сумма покупок у.е: 4078
- частота покупок медиана кол-во: 2

### `Top`

- оценка по RFM диапазон: 9-10
- давность покупок медиана дней: 76
- средний чек у.е.: 1113
- медианный чек у.е.: 552
- средняя общая сумма покупок у.е: 1959
- частота покупок медиана кол-во: 2

### `Medium-value`

- оценка по RFM диапазон: 7-8
- давность покупок медиана дней: 222
- средний чек у.е.: 1447

- медианный чек у.е.: 1087
- средняя общая сумма покупок у.е: 1838
- частота покупок медиана кол-во: 1

`Low-value`

- оценка по RFM диапазон: 6
- давность покупок медиана дней: 281
- средний чек у.е.: 721
- медианный чек у.е.: 600
- средняя общая сумма покупок у.е: 757
- частота покупок медиана кол-во: 1

`Lost`

- оценка по RFM диапазон: 3-5
- давность покупок медиана дней: 360
- средний чек у.е.: 372
- медианный чек у.е.: 299
- средняя общая сумма покупок у.е: 383
- частота покупок медиана кол-во: 1

# 15 Выводы и рекомендации по сегментам покупателей

**Выводы сегмент `Super Top`**

В сезонах 2019-12 и 2020-01 у этой категории покупателей выросло количество покупок - более чем в два раза, чем в другие периоды. Рост покупателей именно в этом сегменте дало не количество покупок, а стоимость приобретаемого товара и недавность приобретения.

В сезон 2019-12 и 2020-01 рост покупок произошел за счет категорий товаров: `хозяйственная утварь` и `цветы/сад`.

В сезон 2019-12 и 2020-01 Топ-5 товаров, которые принесли наибольшую общую сумму:

- Сумка-тележка хозяйственная Rolser IMX006 bassi Logic Tour бордовая хозяйственные сумки - 15358у.е.
- Сушилка Meliconi Stendy Junior хозяйственная утварь - 11188.000000
- Мусорный контейнер Hailo BigBin Swing 45 0845-010 45 л хром хозяйственная утварь - 11024у.е.
- Сумка-тележка хозяйственная Rolser Paris, бордовая, PEP001 bassi JOY хозяйственные сумки - 8234у.е.
- Сумка-тележка хозяйственная Rolser MNB019 rojo LOGIC DOS+2 красная хозяйственные сумки - 8077у.е.

Во всех сезонах в топ-5 товаров, которые принесли или наибольшую общую сумму следующие товары:

- Сушилка Meliconi Stendy Junior хозяйственная утварь - 27970у.е.
- Сумка-тележка хозяйственная Rolser IMX006 bassi Logic Tour бордовая хозяйственные сумки - 23037у.е.
- Сумка-тележка хозяйственная Rolser Pack Gloria Logic RG серая, PAC036 marengo LOGIC RG хозяйственные сумки- 19674у.е.
- Одеяло Wellness T142 белое темостеганое 140х205 см чехол 100% полиэстер 200 г/м 4690659000306 текстиль- 17248у.е.
- Сумка-тележка хозяйственная Rolser MNB019 rojo LOGIC DOS+2 красная хозяйственные сумки - 16154у.е.

**Рекомендации сегмент `Super Top`:**

Покупателям сегмента `Super Top` можно предлагать персонализированно товары из категории `хозяйственная утварь` и `цветы/сад`.

При этом на товары из категории `цветы/сад` в сезонах 2019-05 и 2019-04 был высокий спрос, поэтому можно напомнить о нашем магазине с предложением сезонных товаров из этой категории перед началом сезона. В

сезоне 2019-05 случился самый большой пик продаж. Обеспечил его товар из категории `цветы/сад`. Товар в этой категории неплохо продавался и в сезоны 2019-04, 2019-06, 2019-07, 2019-11,2019-12, 2020-01.

Можно сказать, что во всех сезонах рост покупок происходит в большей части за счет категорий товаров: `цветы/сад`.

Для стимуляции возвратности покупателя, можно предлагать товары из категории `цветы/сад`с апреля по июль с предложением сезонных товаров из этой категории.

Для повышения среднего чека можно включать в рассылку как дорогие, так и товары со средними ценами из раздела `хозяйственная утварь`, `кухонная утварь` и `интерьер/декор`. У этого сегмента покупателей средний чек 1113 у.е., а медианный чек 552 у.е. Это значит, что покупатели покупают и дорогие и недорогие товары.

Основной упор в сезонной рассылке сделать на товары из категории цветы/сад.

**Выводы сегмент `Top`**

В сегменте клиентов `Top` в сезоне 2019-05 случился самый большой пик продаж. Обеспечил его товар из категории `цветы/сад`. Товар в этой категории неплохо продавался и в сезоны 2019-04, 2019-06, 2019-07, 2019-11,2019-12, 2020-01.

Также категория `хозяйственная утварь` с небольшими перепадами, но стабильно имеет спрос у сегмента `Top` . В сезоне 2020-01 и 2019-11 есть небольшой рост в категории `кухонная утварь`.

Можно сказать, что во всех сезонах рост покупок происходит в большей части за счет категорий товаров: `цветы/сад`.

**Рекомендации сегмент`Top`:**

Для стимуляции возвратности покупателя, можно предлагать товары из категории `цветы/сад` с апреля по июль с предложением сезонных товаров из этой категории.

Для повышения среднего чека можно включать в рассылку дорогие товары из раздела `хозяйственная утварь`, `кухонная утварь` и `интерьер/декор`. Учитывая, что покупатели в этом сегмента покупают дорогостоящий товар, включение в рассылку дорогих товаров (по ассортиментному ряду нашего магазина) целесообразно. Основной упор в сезонной рассылке сделать на товары из категории цветы/сад.

**Выводы сегмент `Medium-value`** У сегмента `Medium-value` пик покупок в категориях `цветы/сад` пришелся на 2019-05 и 2019-04, с хорошими продажами и в сезоны 2019-03, 2019-06, 2019-07 - больше всего заказов в категориях `цветы/сад`.

Можно сказать, что во всех сезонах рост покупок происходит в основном в большей части за счет категорий товаров: `цветы/сад`. В сезоне 2019-05 был спрос на категорию `хозяйственная утварь`.

В топ-5 товаров, которые принесли наибольшие продажи вошли

- Простынь вафельная 200х180 см WELLNESS RW180-01 100% хлопок текстиль - 53232 у.е.
- Тележка багажная DELTA ТБР-22 синий грузоподъемность 20 кг сумка и 50 кг каркас РОССИЯ хозяйственные сумки-32718 у.е.
- Набор ножей Attribute CHEF 5 предметов AKF522 кухонная утварь-29248 у.е.
- Урна уличная "Гео", Hobbyka/Хоббика, 59*37,5см, сталь интерьер/декор - 24370 у.е.
- Муляж ЯБЛОКО 9 см красное интерьер/декор - 16930 у.е.

**Рекомендации сегмент `Medium-value`** Покупателям сегмента `Medium-value` можно предлагать персонализированно товары из категории `цветы/сад`, учитывать, что с сезона 2019-03 по 2019-07 с пиком в мае был высокий спрос, можно напомнить о нашем магазине с предложением сезонных товаров из этой категории.

Для повышения среднего чека можно включить в основную рассылку не только товары категории цветы/сад, но и дорогие товары из раздела `хозяйственная утварь`. Но основной упор в сезонной рассылке сделать на товары из категории цветы. Можно изучить сезонные особенности высадки семян/черенков и предлагать этот товар более прицельно по датам ожидаемого спроса.

Медианный чек у этой категории покупателей выше, чем у сегмента `Top`, это значит, что это более стабильная категория покупателей. При ограниченном рекламном бюджете рекомендовано не исключать этот сегмент из рассылки.

**Выводы сегмент `Low-value`:**

В периоды с 2019-02 по 2019-10 по этой товарной категории наблюдался стабильный спрос на товары из категории `цветы/сад`, а в сезонах 2019-03, 2019-04, 2019-05 и 2018-10 был рост заказов в большинстве за счет товарной категории `цветы/сад`.

В сезоне 2019-10, 2019-09 и особенно 2018-11 был спрос на товары категории `'интерьер/декор'`. Категория товаров `хозяйственная утварь` стабильно пользовалась спросом, особенно в октябре 2018. После 2019-11 этот сегмент неактивен.

**Рекомендации сегмент `Low-value`:**

Покупателей сегмента `Low-value` можно попробовать пробудить рассылкой по категории товаров `цветы/сад`, учитывать, что с сезона 2019-03 и 2019-04, а также 2018-11 и 2018-10 был спрос на эти товары.

При ограниченном рекламном бюджете - на этой категории покупателей можно сэкономить, ограничив количество рассылки и т.д.

Средняя сумма покупок у покупателей этого сегмента 757 у.е. при частоте покупок 1.

Для повышения среднего чека можно включать как дополнение в основную рассылку не дорогие товары из раздела `хозяйственная утварь`, `'интерьер/декор'`. Учитывая, что покупатели в этом сегменте не отличились покупками дорогих товаров, поэтому включение в рассылку дорогих товаров (по ассортиментному ряду нашего магазина) нецелесообразно. Основной упор в сезонной рассылке сделать на товары из категории цветы.

**Выводы сегмент `Lost`:**

Сегмент покупателей `Lost` 2019-04, 2019-03, 2018-11, 2018-10 больше всего заказал товаров из категории `цветы/сад`. Отмечен спрос 2018-10 и 2018-11 на товары категории `'интерьер/декор'`. Категория товаров `хозяйственная утварь`, 'кухонная утварь' стабильно пользовалась спросом в период активности покупателей.

С 2019-10 этот сегмент покупателей не проявлял активности.

**Рекомендации сегмент `Lost`:**

Покупателей сегмента `Lost` можно попробовать оживить рассылкой по категории товаров `цветы/сад`, учитывать, что в сезоны 2019-03 и 2019-04, а также 2018-11 и 2018-10 был особенно высокий спрос на эти товары.

Однако, при ограниченном рекламном бюджете - на этой категории покупателей можно сэкономить, ограничив количество рассылки и т.д.

Средний чек у этого сегмента 372 у.е.

Для повышения среднего чека можно включать как дополнение в основную рассылку недорогие товары - не дороже 1000 у.е. за шт из раздела `хозяйственная утварь`, `'интерьер/декор'`,'кухонная утварь'.

## Общие рекомендации:

Покупателям всех товарных категорий подойдет рассылка с товарами из категорий `цветы/сад`.

Более дорогие товары находятся в категории `хозяйственная утварь`.

Основной упор в сезонной рассылке сделать на товары из категории цветы. Если сопоставить ассортимент товара с посевным календарем и работами в саду, для которого необходим инструмент, то можно составлять сеты из ассортиментов товаров и предлагать комплексно товар.

Предлагать рекламу можно всем сегментам покупателей: 'Lost', 'Medium-value', 'Super Top', 'Top', 'Low-value'

Однако, при ограниченном рекламном бюджете - на категории покупателей 'Lost' и 'Low-value' можно сэкономить, ограничив количество рассылки и т.д.