# E-commerce — Выявление профилей потребления

# Описание проекта.

Интернет-магазину «Пока все еще тут» требуется рассчет основных метрик и создание гипотез на основе полученных данных для того чтобы стать лучше для своих клиентов.

# Описание данных:

Датасет описывает транзакции интернет-магазина товаров для дома и быта «Пока все ещё тут».

Колонки:

* date — дата заказа;
* customer\_id — идентификатор покупателя;
* order\_id — идентификатор заказа;
* product — наименование товара;
* quantity — количество товара в заказе;
* price — цена товара.

# Задача:

Сегментировать покупателей по профилю потребления (на основе истории их покупок).

# План исследования:

1. Загрузка данных и подготовка их к анализу

* открытие файла с данными и изучение общей информации
* проверка на пропуски и типы данных, их корректировка при необходимости
* проверка на дубликаты
* промежуточные выводы

1. Исследовательский анализ данных

* установление общего количества пользователей в логе и количества дней анализа
* изучение количества товаров в заказе
* изучение распределения стоимости заказа
* промежуточные выводы

1. Разделение товаров на группы

* лемматизация
* промежуточные выводы

1. RFM-анализ для сегментирования клиентов

* RFM-анализ
* промежуточные выводы

1. Формирование и проверка статистических гипотез.
2. Выводы и рекомендации.

## 1  Загрузка данных и подготовка их к анализу.

Ввод [1]:

*#подключаем все необходимые библиотеки*

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**import** datetime **as** dt

**from** scipy **import** stats **as** st

**from** scipy.stats **import** mannwhitneyu

**from** pymystem3 **import** Mystem

**from** collections **import** Counter

**import** warnings

**from** pandas.core.common **import** SettingWithCopyWarning

warnings.simplefilter(action**=**"ignore", category**=**SettingWithCopyWarning)

warnings.filterwarnings("ignore", category**=**FutureWarning)

Ввод [2]:

*#выгрузка и ознакомление с данными датасета*

data **=** pd.read\_csv('/datasets/ecom\_dataset\_upd.csv')

data.head()

Out[2]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 2018100100 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Алое Вера, d12, h30 | 1 | 142.0 |
| **1** | 2018100100 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Кофе Арабика, d12,... | 1 | 194.0 |
| **2** | 2018100100 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Радермахера d-12 см h-20 см | 1 | 112.0 |
| **3** | 2018100100 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Хризолидокарпус Лутесценс d-9 см | 1 | 179.0 |
| **4** | 2018100100 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Циперус Зумула d-12 см h-25 см | 1 | 112.0 |

Ввод [3]:

*#получим общую информацию о данных*

data.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

Данные включают 7474 строки и 6 столбцов, вижу логическое несоответствие типа данных в столбце date.

Ввод [4]:

*#произвожу замену типа данных в столбце date*

data['date'] **=** pd.to\_datetime(data['date'], format**=**'%Y%m%d%H')

*#проверяю изменения*

data.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 datetime64[ns]

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: datetime64[ns](1), float64(1), int64(2), object(2)

memory usage: 350.5+ KB

Ввод [5]:

*#проверяю изменения*

data.head()

Out[5]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Алое Вера, d12, h30 | 1 | 142.0 |
| **1** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Комнатное растение в горшке Кофе Арабика, d12,... | 1 | 194.0 |
| **2** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Радермахера d-12 см h-20 см | 1 | 112.0 |
| **3** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Хризолидокарпус Лутесценс d-9 см | 1 | 179.0 |
| **4** | 2018-10-01 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | Циперус Зумула d-12 см h-25 см | 1 | 112.0 |

Ввод [6]:

*#проверяю наличие пропусков*

data.isna().sum()

Out[6]:

date 0

customer\_id 0

order\_id 0

product 0

quantity 0

price 0

dtype: int64

Ввод [7]:

*#проверяю наличие дубликатов*

print('Количество дубликатов:', data.duplicated().sum())

Количество дубликатов: 0

Ввод [8]:

*#проверяю наличие неявных дубликатов*

**if** len(set(data)) **==** len(data):

print("Нет неявных дубликатов.")

**else**:

print("Есть неявные дубликаты.")

Есть неявные дубликаты.

Ввод [9]:

*#выполним проверку неявных дубликатов*

*#приведем столбцы к нижнему регистру*

data['product'] **=** data['product'].str.lower()

data[data[['customer\_id', 'order\_id', 'product', 'quantity']].duplicated(keep**=False**)].head()

Out[9]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** |
| --- | --- | --- | --- | --- | --- | --- |
| **15** | 2018-10-01 18:00:00 | 17213b88-1514-47a4-b8aa-ce51378ab34e | 68476 | мини-сковорода marmiton "сердце" с антипригарн... | 1 | 239.0 |
| **16** | 2018-10-01 18:00:00 | 17213b88-1514-47a4-b8aa-ce51378ab34e | 68476 | сковорода алюминиевая с антипригарным покрытие... | 1 | 824.0 |
| **17** | 2018-10-01 18:00:00 | 17213b88-1514-47a4-b8aa-ce51378ab34e | 68476 | стеклянная крышка для сковороды alpenkok 26 см... | 1 | 262.0 |
| **18** | 2018-10-01 18:00:00 | 17213b88-1514-47a4-b8aa-ce51378ab34e | 68476 | сушилка для белья напольная colombo star 18, 3679 | 1 | 1049.0 |
| **19** | 2018-10-01 21:00:00 | b731df05-98fa-4610-8496-716ec530a02c | 68474 | доска гладильная eurogold professional 130х48 ... | 1 | 3299.0 |

Ввод [10]:

*#удалим неявные дубли*

data.drop\_duplicates(subset**=**['customer\_id', 'order\_id', 'product', 'quantity'], inplace**=True**)

*#еще раз проверим*

data[data[['customer\_id', 'order\_id', 'product', 'quantity']].duplicated(keep**=False**)]

​

Out[10]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** |
| --- | --- | --- | --- | --- | --- | --- |

Ввод [11]:

data.info()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 5608 entries, 0 to 7473

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 date 5608 non-null datetime64[ns]

1 customer\_id 5608 non-null object

2 order\_id 5608 non-null int64

3 product 5608 non-null object

4 quantity 5608 non-null int64

5 price 5608 non-null float64

dtypes: datetime64[ns](1), float64(1), int64(2), object(2)

memory usage: 306.7+ KB

Ввод [12]:

*# находим строки с заказов с несколькими customer\_id*

duplicated\_orders **=** data[data.duplicated(subset**=**['order\_id', 'customer\_id'], keep**=False**)]

​

*# удаляем*

data **=** data.drop\_duplicates(subset**=**['order\_id', 'customer\_id'])

​

​

Ввод [13]:

*#посмотрим есть ли заказы с более чем 1 customer\_id*

data.groupby('order\_id')['customer\_id'].nunique().sort\_values()

Out[13]:

order\_id

12624 1

72313 1

72315 1

72320 1

72321 1

..

70542 2

71226 2

71480 3

69485 3

72845 4

Name: customer\_id, Length: 3521, dtype: int64

Ввод [14]:

*#перезапишем data, исключив из него такие строки*

data **=** data.groupby('order\_id').filter(**lambda** x: x['customer\_id'].nunique() **<** 2)

Ввод [15]:

*#проверим еще раз*

pivot **=** pd.pivot\_table(data, index**=**'order\_id', values**=**'customer\_id', aggfunc**=**'nunique').sort\_values(by**=**'customer\_id', ascending**=False**)

pivot

Out[15]:

|  | **customer\_id** |
| --- | --- |
| **order\_id** |  |
| **12624** | 1 |
| **72339** | 1 |
| **72325** | 1 |
| **72327** | 1 |
| **72329** | 1 |
| **...** | ... |
| **70076** | 1 |
| **70077** | 1 |
| **70078** | 1 |
| **70079** | 1 |
| **112789** | 1 |

3492 rows × 1 columns

Ввод [16]:

data.info()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 3492 entries, 0 to 7473

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 date 3492 non-null datetime64[ns]

1 customer\_id 3492 non-null object

2 order\_id 3492 non-null int64

3 product 3492 non-null object

4 quantity 3492 non-null int64

5 price 3492 non-null float64

dtypes: datetime64[ns](1), float64(1), int64(2), object(2)

memory usage: 191.0+ KB

В данных отсутствуют пропуски и явные дубликаты, но есть неявные дубликаты - 3919 строк из 7474 - удалены, откорректитрован тип данных в столбце с датой, наименования товаров приведены к нижнему регистру: лог подготовлен для исследовательского анализа.

## 2  Исследовательский анализ данных.

Ввод [17]:

print('Всего в логе уникальных уникальных пользователей, совершавших покупки:', data['customer\_id'].nunique (),'.')

Всего в логе уникальных уникальных пользователей, совершавших покупки: 2413 .

Ввод [18]:

min\_date **=** data["date"].min()

max\_date **=** data["date"].max()

​

print(f"Период наличия данных: с {min\_date} по {max\_date}.")

Период наличия данных: с 2018-10-01 00:00:00 по 2020-01-31 15:00:00.

Ввод [19]:

*#добавим столбец monetary для RFM-анализа*

data['monetary'] **=** data['quantity']**\***data['price']

data.head()

Out[19]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** | **monetary** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2018-10-01 00:00:00 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | комнатное растение в горшке алое вера, d12, h30 | 1 | 142.0 | 142.0 |
| **7** | 2018-10-01 08:00:00 | 375e0724-f033-4c76-b579-84969cf38ee2 | 68479 | настенная сушилка для белья gimi brio super 100 | 1 | 824.0 | 824.0 |
| **8** | 2018-10-01 08:00:00 | 6644e5b4-9934-4863-9778-aaa125207701 | 68478 | таз пластмассовый 21,0 л круглый "водолей" с61... | 1 | 269.0 | 269.0 |
| **9** | 2018-10-01 09:00:00 | c971fb21-d54c-4134-938f-16b62ee86d3b | 68480 | чехол для гладильной доски colombo persia beig... | 1 | 674.0 | 674.0 |
| **10** | 2018-10-01 11:00:00 | 161e1b98-45ba-4b4e-8236-e6e3e70f6f7c | 68483 | вешалка для брюк металлическая с резиновым пок... | 10 | 82.0 | 820.0 |

Ввод [20]:

*#изучим значения столбца quantity*

data['quantity'].describe()

Out[20]:

count 3492.000000

mean 3.164948

std 20.501022

min 1.000000

25% 1.000000

50% 1.000000

75% 1.000000

max 1000.000000

Name: quantity, dtype: float64

Ввод [21]:

plt.figure(figsize**=**(16, 3))

plt.hist(data['quantity'], bins**=**100, color**=**"blue")

plt.xlabel("Количество товаров")

plt.title("Распределение количества товаров")

plt.show()

****

Мы можем видеть, что чаще всего пользователи преобретают 1 ед товара, но есть и выбросы - от 50 до 1000 ед - предположу что это уже не физические лица, а компании, которые делают закупы для бизнеса, что относится к B2B-сегменту, который в данном исследовании нас не интересует, и который может сильно исказить его результаты, поэтому я перезапишу датасет, исключив "оптовиков" из него.

Ввод [22]:

data\_2 **=** data.query('quantity <= 200')

Ввод [23]:

*#изучим еще раз значения столбца quantity*

data\_2['quantity'].describe()

Out[23]:

count 3489.000000

mean 2.699341

std 8.883155

min 1.000000

25% 1.000000

50% 1.000000

75% 1.000000

max 200.000000

Name: quantity, dtype: float64

Ввод [24]:

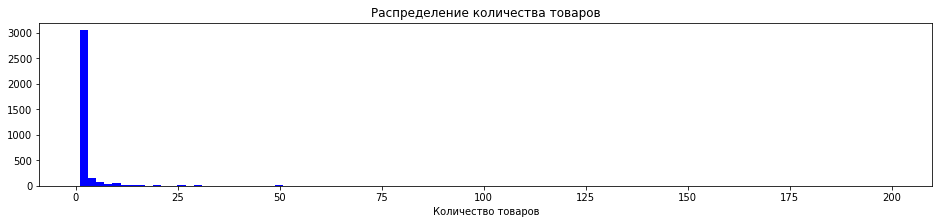
plt.figure(figsize**=**(16, 3))

plt.hist(data\_2['quantity'], bins**=**100, color**=**"blue")

plt.xlabel("Количество товаров")

plt.title("Распределение количества товаров")

plt.show()

****

Теперь мы видим, что физически лица чаще всего покупают 1-2 товара за раз, оставленные скачки от 10 до 50 ед можно объяснить покупками каких-то бытовых вещей для дома при переезде или при ремонте (например вешалки/розетки), так же стоит упомянуть сезонные закупы семенами (а это стабильное ежегодное мероприятие). Оставленые нами значения выглядят нормальными.

Ввод [25]:

*#изучим значения столбца monetary*

data\_2['monetary'].describe()

Out[25]:

count 3489.000000

mean 974.824424

std 1757.750505

min 9.000000

25% 149.000000

50% 449.000000

75% 1087.000000

max 47385.000000

Name: monetary, dtype: float64

Ввод [26]:

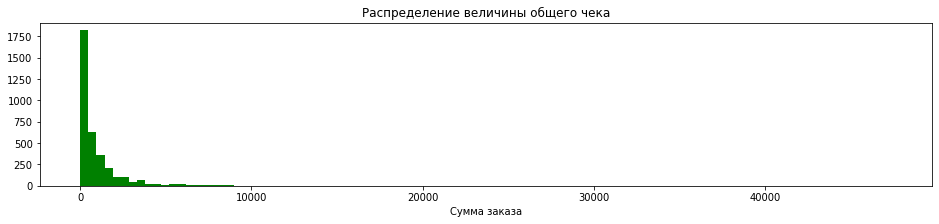
plt.figure(figsize**=**(16, 3))

plt.hist(data\_2['monetary'], bins**=**100, color**=**"green")

plt.xlabel("Сумма заказа")

plt.title("Распределение величины общего чека")

plt.show()

****

Исходя из описания и гистограммы исследуемой величины можно сделать вывод, что чаще всего покупают "дешевые товары", средняя стоимость заказа составляет около 600 ед, выбросы сохраняются, но на мой взгляд являются адекватными: стоимость заказа в 40 тыс уе может говорить о сезонности каких-то покупок, от данных по которым не имеет смысла отказываться в силу стабильности дохода от них (например подготовка ребенка к школе или покупка рассады, или обновление гардероба).

## 3  Разделение товаров на группы.

Ввод [27]:

m **=** Mystem()

​

*#создаем новый столбец lemmas и заполняем его леммами из столбца product*

data\_2['lemmas'] **=** data\_2['product'].apply(**lambda** x: ' '.join(m.lemmatize(x)))

​

data\_2.head()

Out[27]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** | **monetary** | **lemmas** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2018-10-01 00:00:00 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | комнатное растение в горшке алое вера, d12, h30 | 1 | 142.0 | 142.0 | комнатный растение в горшок алый вер... |
| **7** | 2018-10-01 08:00:00 | 375e0724-f033-4c76-b579-84969cf38ee2 | 68479 | настенная сушилка для белья gimi brio super 100 | 1 | 824.0 | 824.0 | настенный сушилка для белье gimi bri... |
| **8** | 2018-10-01 08:00:00 | 6644e5b4-9934-4863-9778-aaa125207701 | 68478 | таз пластмассовый 21,0 л круглый "водолей" с61... | 1 | 269.0 | 269.0 | таз пластмассовый 21 , 0 л круглый " ... |
| **9** | 2018-10-01 09:00:00 | c971fb21-d54c-4134-938f-16b62ee86d3b | 68480 | чехол для гладильной доски colombo persia beig... | 1 | 674.0 | 674.0 | чехол для гладильный доска colombo p... |
| **10** | 2018-10-01 11:00:00 | 161e1b98-45ba-4b4e-8236-e6e3e70f6f7c | 68483 | вешалка для брюк металлическая с резиновым пок... | 10 | 82.0 | 820.0 | вешалка для брюки металлический с ре... |

Ввод [28]:

*#cчитаем количество упоминаний каждой леммы в столбце lemmas*

lemmas\_count **=** Counter(data\_2['lemmas'].str.split(expand**=True**).stack())

​

*#выбираем наиболее популярные направления (леммы)*

top\_lemmas **=** lemmas\_count.most\_common()

top\_lemmas

​

Out[28]:

[(',', 1944),

('-', 1791),

('см', 1742),

('для', 877),

('d', 399),

('тележка', 373),

('с', 365),

('сумка', 344),

('пеларгония', 335),

('белье', 332),

('л', 304),

('сушилка', 295),

('в', 275),

('белый', 262),

('2', 261),

('доска', 258),

('12', 250),

('gimi', 240),

('гладильный', 214),

('5', 209),

('зональный', 206),

('/', 202),

('(', 198),

('.', 196),

('синий', 180),

('красный', 176),

('кг', 175),

('цветок', 172),

('черенок', 172),

('10', 170),

('укоренять', 170),

('и', 167),

('0', 166),

('ника', 164),

('м', 160),

('шт', 159),

('хозяйственный', 155),

(')', 152),

('20', 152),

('h', 151),

('1', 151),

('7', 149),

('25', 144),

('рассада', 142),

('диам', 137),

('махровый', 135),

('х', 132),

('3', 126),

('9', 126),

('\*', 125),

('россия', 124),

('из', 122),

('30', 122),

('ванный', 119),

('коврик', 115),

('чехол', 113),

('потолочный', 112),

('кассета', 112),

('колесный', 111),

('муляж', 111),

('50', 111),

('искусственный', 110),

('черный', 107),

('розебудный', 107),

('6', 104),

('4', 104),

('пластмассовый', 102),

('пластиковый', 102),

('лиана', 101),

('delta', 100),

('luminarc', 97),

('горшок', 94),

('грузоподъемность', 93),

('багажный', 90),

('ручка', 89),

('комната', 88),

('объем', 84),

('штора', 82),

('розовый', 79),

('каркас', 79),

('attribute', 78),

('тбр', 78),

('по', 77),

('сорт', 77),

('d12', 76),

('мл', 75),

('полумахровый', 72),

('"', 70),

('argo', 70),

('на', 70),

('серый', 69),

('настенный', 68),

('100', 68),

('полиуретан', 68),

('хлопок', 67),

('цвет', 67),

('zalger', 67),

('%', 66),

('герань', 64),

('домашний', 64),

('цвести', 64),

('h25', 64),

('таз', 63),

('зеленый', 63),

('вешалка', 62),

('однолетний', 62),

('крышка', 61),

('придверный', 61),

('напольный', 60),

('роза', 60),

('ассортимент', 59),

('wellness', 58),

('leifheit', 58),

('коричневый', 57),

('стремянка', 54),

('оранжевый', 54),

('г', 54),

('rolser', 53),

('8', 52),

('пвх', 51),

('желтый', 51),

('голубой', 51),

('набор', 50),

('люкс', 50),

('сталь', 49),

('алюминиевый', 49),

('joy', 49),

('мм', 48),

('кухня', 47),

('бежевый', 47),

('№', 47),

('со', 47),

('),', 46),

('сиреневый', 46),

('21', 45),

('тарелка', 45),

('60', 45),

('18', 45),

('виолета', 45),

('томат', 45),

('200', 44),

('red', 43),

('деревянный', 42),

('15', 42),

('45', 41),

('зелень', 41),

('помидор', 41),

('круглый', 40),

('мини', 40),

('штука', 40),

('покрытие', 39),

('щетка', 39),

('скатерть', 39),

('bacchetta', 39),

('colombo', 38),

('pandora', 38),

('lift', 37),

('корзина', 37),

('hausmann', 37),

('240х200', 36),

('нержавеющий', 35),

('микс', 34),

('пищевой', 34),

('яблоко', 34),

('складной', 34),

('среднеспелый', 34),

('прямоугольный', 32),

('овальный', 32),

('растение', 31),

('малиновый', 31),

('банка', 31),

('rozenbal', 30),

('полиэстер', 30),

('одежда', 30),

('180х200', 30),

('ампельный', 30),

('полка', 30),

('петуния', 30),

('простой', 29),

('упаковка', 29),

('металлический', 28),

('контейнер', 28),

('22', 28),

('qwerty', 28),

('17', 27),

('spirella', 27),

('вт', 27),

('webber', 27),

('индетерминантный', 27),

('иж', 26),

('стеклянный', 26),

('90', 26),

('вайт', 26),

('eurogold', 25),

('80', 25),

('хлопковый', 25),

('текстильный', 25),

('ступень', 25),

('нож', 25),

('базилик', 25),

('ринго', 25),

('мята', 25),

('andersen', 24),

('shopper', 24),

('лососевый', 24),

('dasch', 24),

('16', 24),

('13', 24),

('hm', 23),

('ванна', 23),

('valiant', 23),

('40', 23),

('кольцо', 23),

('curver', 23),

('ллп', 23),

('потолочно', 22),

('scala', 22),

('\_', 22),

('ступенчатый', 22),

('180', 22),

('салатник', 22),

('120', 22),

('basic', 21),

('большой', 21),

('коралловый', 21),

('чайник', 21),

('ведро', 21),

('гвоздик', 21),

('фиолетовый', 21),

('садовый', 21),

('dogrular', 20),

('оцинковывать', 20),

('композиция', 20),

('букет', 20),

('122х40', 20),

('раздвижной', 20),

('капуста', 20),

('крючок', 19),

('обеденный', 19),

('франция', 19),

('ерш', 19),

('линия', 19),

('поролон', 19),

('пластик', 19),

('салфетка', 19),

('полимербыт', 19),

('окно', 19),

('подставка', 18),

('510', 18),

('кружка', 18),

('23', 18),

('400', 18),

('дерево', 18),

('противоскользящий', 18),

('logic', 18),

('десертный', 18),

('6шт', 18),

('rg', 18),

('швабра', 18),

('лиловый', 18),

('margaretha', 18),

('prima', 17),

('720', 17),

('веревочный', 17),

('apache', 17),

('jet', 17),

('flexi', 17),

('угловой', 17),

('106', 17),

('подарочный', 17),

('малый', 17),

('мытье', 17),

('210', 17),

('бакоп', 17),

('тимьян', 16),

('под', 16),

('лестница', 16),

('насадка', 16),

('500', 16),

('плечики', 16),

('универсальный', 16),

('ве', 16),

('19', 16),

('комнатный', 15),

('без', 15),

('24', 15),

('hailo', 15),

('lux', 15),

('azul', 15),

('штанга', 15),

('+', 15),

('карниз', 15),

('бегония', 15),

('электрический', 15),

('5х29', 15),

('negro', 15),

('galaxy', 15),

('l', 15),

('стальной', 15),

('helex', 15),

('ель', 15),

('ротанг', 15),

('подвесной', 15),

('300', 14),

('tris', 14),

('112', 14),

('декоративный', 14),

('alpenkok', 14),

('14', 14),

('антуриум', 14),

('кухонный', 14),

('2000', 14),

('обувь', 14),

('мультиколор', 14),

('новогодний', 14),

('серебро', 14),

('mary', 14),

('примула', 14),

('prins', 14),

('nikolai', 14),

('раннеспелый', 14),

('арбуз', 14),

('кашпо', 14),

('dt', 14),

('clean', 13),

('120х38', 13),

('70', 13),

('эвкалипт', 13),

('textil', 13),

('sarayli', 13),

('телескопический', 13),

('хром', 13),

('44', 13),

('прозрачный', 13),

('be', 13),

('пуансеттия', 13),

('фиалка', 13),

('белокачать', 13),

('рисунок', 12),

('",', 12),

('унитазный', 12),

('plus', 12),

('лаванда', 12),

('перец', 12),

('ковер', 12),

('58', 12),

('костюм', 12),

('55', 12),

('c', 12),

('азалия', 12),

('45х75', 12),

('ak', 12),

('эмалированный', 12),

('m', 12),

('дсп', 12),

('полиэфир', 12),

('перекладина', 12),

('s', 12),

('php', 12),

('srl', 12),

('42', 12),

('крокус', 12),

('easy', 12),

('250', 12),

('фуксия', 12),

('160', 12),

('ромашка', 12),

('e6', 12),

('резиновый', 11),

('сковорода', 11),

('ажурный', 11),

('50х80', 11),

('хранение', 11),

('посуда', 11),

('бэлль', 11),

('1703009', 11),

('new', 11),

('122х34', 11),

('123х46', 11),

('сидение', 11),

('blue', 11),

('00', 11),

('тюльпан', 11),

('280', 11),

('термокружок', 11),

('queen', 11),

('ingrid', 11),

('мультиблум', 11),

('скарлет', 11),

('ай', 11),

('детерминантный', 11),

('перечный', 11),

('окномойка', 11),

('1520', 10),

('антипригарный', 10),

('сметка', 10),

('подкладка', 10),

('135х50', 10),

('мыло', 10),

('гунни', 10),

('ufuk', 10),

('комплект', 10),

('стиральный', 10),

('1404021', 10),

('1703011', 10),

('войлок', 10),

('андрианум', 10),

('спирелла', 10),

('резина', 10),

('мт', 10),

('баклажан', 10),

('бальзамин', 10),

('василиса', 10),

('зми', 10),

('антик', 10),

('classic', 10),

('go', 10),

('алюминий', 10),

('white', 10),

('эльфпласт', 10),

('=', 10),

('предмет', 10),

('золотой', 10),

('циперус', 10),

('зумул', 10),

('коробка', 10),

('pasabahce', 10),

('полотенце', 10),

('цикламен', 10),

('март', 10),

('блю', 10),

('р', 10),

('super', 9),

('пластика', 9),

('130х48', 9),

('комбинированный', 9),

('1209021', 9),

('хозлэнд', 9),

('11', 9),

('fancy', 9),

('fair', 9),

('ваир', 9),

('чили', 9),

('color', 9),

('оконный', 9),

('140', 9),

('scovo', 9),

('0000', 9),

('жаккард', 9),

('овощ', 9),

('лоток', 9),

('150', 9),

('стелла', 9),

('размер', 9),

('подрукавник', 9),

('lini', 9),

('108', 9),

('кокос', 9),

('цветной', 9),

('новогвинейский', 9),

('медный', 9),

('ручной', 9),

('виола', 9),

('средство', 9),

('750', 9),

('весы', 9),

('нидерланды', 9),

('сербский', 9),

('60х140', 9),

('розмарин', 9),

('meliconi', 9),

('свч', 9),

('75', 9),

('emma', 9),

('образный', 9),

('80х170', 9),

('луковица', 9),

('калибрахоа', 9),

('вербена', 9),

('керамика', 9),

('beige', 8),

('26', 8),

('\*,', 8),

('лимон', 8),

('вилка', 8),

('столовая', 8),

('floral', 8),

('1712012', 8),

('market', 8),

('кувшин', 8),

('comfort', 8),

('душа', 8),

('б', 8),

('dl', 8),

('керамический', 8),

('28', 8),

('ярко', 8),

('блэк', 8),

('премиум', 8),

('мусор', 8),

('бамбук', 8),

('минеральный', 8),

('вода', 8),

('48', 8),

('54', 8),

('одиночный', 8),

('классик', 8),

('marengo', 8),

('строительный', 8),

('уличный', 8),

('чайный', 8),

('york', 8),

('ideal', 8),

('м2', 8),

('format', 8),

('fashion', 8),

('blues', 8),

('ложка', 8),

('convert', 8),

('metal', 8),

('стенка', 8),

('суповой', 8),

('35', 8),

('60х90', 8),

('stendy', 8),

('06', 8),

('пол', 8),

('отжим', 8),

('плантация', 8),

('сбп1', 8),

('текстиль', 8),

('донна', 8),

('эмпилабль', 8),

('60х110', 8),

('r261016', 8),

('арарат', 8),

('крупноцветковый', 8),

('декабрист', 8),

('17х17х10', 8),

('чайногибридный', 8),

('закрытый', 8),

('ящик', 8),

('9см', 8),

('лобелия', 8),

('120х42', 7),

('мягкий', 7),

('рядная', 7),

('интерьерный', 7),

('стандарт', 7),

('133', 7),

('051', 7),

('ff', 7),

('152', 7),

('towa', 7),

('co', 7),

('простыня', 7),

('burstenmann', 7),

('series', 7),

('стяжка', 7),

('цинк', 7),

('т', 7),

('002', 7),

('искуственный', 7),

('три', 7),

('capri', 7),

('jet017', 7),

('батарея', 7),

('122х42', 7),

('protec', 7),

('30х43', 7),

('влаговпитывающий', 7),

('135х45', 7),

('1703006', 7),

('свисток', 7),

('leonardo', 7),

('eva', 7),

('пэмби', 7),

('130', 7),

('зубной', 7),

('паста', 7),

('elegant', 7),

('125x45', 7),

('лувр', 7),

('бутон', 7),

('120х40', 7),

('quadretto', 7),

('корыто', 7),

('бесшовный', 7),

('meiwa', 7),

('1800', 7),

('дизайн', 7),

('dos', 7),

('350х150', 7),

('ворсовый', 7),

('эконом', 7),

('эк1', 7),

('40х68', 7),

('бирюзовый', 7),

('lora', 7),

('hlf', 7),

('124х46', 7),

('5757', 7),

('170', 7),

('twin', 7),

('56', 7),

('decs', 7),

('rigone', 7),

('112х34', 7),

('разделочный', 7),

('эклипсы', 7),

('корея', 7),

('1840', 7),

('mop', 7),

('замок', 7),

('tex', 7),

('флауэр', 7),

('44х24х10', 7),

('40х20х10', 7),

('36х16х10', 7),

('180х180', 7),

('кастрюля', 7),

('бук', 7),

('хлебница', 7),

('ландыш', 7),

('тонус', 7),

('angeleyes', 7),

('открытка', 7),

('&', 7),

('la', 7),

('овощной', 7),

('смесь', 7),

('2810', 7),

('125х43', 7),

('дыня', 7),

('p9', 7),

('клубника', 7),

('вечноцветущий', 7),

('огурец', 7),

('blossom', 7),

('h30', 6),

('130х50', 6),

('сетка', 6),

('450', 6),

('прищепок', 6),

('wc', 6),

('мультипласт', 6),

('виноград', 6),

('elba', 6),

('40х60', 6),

('перчатка', 6),

('silver', 6),

('220', 6),

('jp', 6),

('болт', 6),

('ст', 6),

('1108354', 6),

('индик', 6),

('петля', 6),

('металлизированный', 6),

('встроенный', 6),

('полиамидный', 6),

('аквамарин', 6),

('hsc', 6),

('эльза', 6),

('салатовый', 6),

('520114', 6),

('топ', 6),

('110', 6),

('malva', 6),

('bella', 6),

('nature', 6),

('кокосовый', 6),

('кот', 6),

('гербера', 6),

('130х46', 6),

('микрофибры', 6),

('50х85', 6),

('520', 6),

('внп', 6),

('125х42', 6),

('браун', 6),

('мелисса', 6),

('продукт', 6),

('knit', 6),

('жестяной', 6),

('наволочка', 6),

('50х70', 6),

('ekspo', 6),

('форма', 6),

('электронный', 6),

('миска', 6),

('одеяло', 6),

('канадский', 6),

('коника', 6),

('столовый', 6),

('этажерка', 6),

('термос', 6),

('1303007', 6),

('mohawk', 6),

('iron', 6),

('thermo', 6),

('эхеверие', 6),

('деликатный', 6),

('89', 6),

('чистка', 6),

('2110', 6),

('проволочный', 6),

('010', 6),

('0401', 6),

('1860', 6),

('подарок', 6),

('factotum', 6),

('prisma', 6),

('junior', 6),

('классика', 6),

('wet', 6),

('ребристый', 6),

('рукав', 6),

('стакан', 6),

('газание', 6),

('диаметр', 6),

('афеляндра', 6),

('скуаррос', 6),

('дания', 6),

('hausrein', 6),

('10шт', 6),

('41', 6),

('telegant', 6),

('purple', 6),

('кабачок', 6),

('шоколадный', 6),

('chocolate', 6),

('apple', 6),

('surprise', 6),

('вера', 5),

('brio', 5),

('persia', 5),

('5379', 5),

(':', 5),

('c42', 5),

('ripple', 5),

('мусорный', 5),

('флисовый', 5),

('6194', 5),

('кумган', 5),

('1408003', 5),

('связка', 5),

('резинка', 5),

('металл', 5),

('welfull', 5),

('тыква', 5),

('сб5', 5),

('240', 5),

('5031', 5),

('tepмокружка', 5),

('avex', 5),

('140х50', 5),

('калатея', 5),

('metalloni', 5),

('секционный', 5),

('universal', 5),

('веник', 5),

('сорго', 5),

('шов', 5),

('унитаз', 5),

('подголовник', 5),

('25х34', 5),

('грин', 5),

('pink', 5),

('framar', 5),

('4062', 5),

('90х150', 5),

('оранж', 5),

('урна', 5),

('88', 5),

('05', 5),

('1703010', 5),

('60x40', 5),

('amc015', 5),

('овощеварка', 5),

('045', 5),

('1502079', 5),

('совок', 5),

('вафельный', 5),

('truper', 5),

('терка', 5),

('горох', 5),

('гортензия', 5),

('радуга', 5),

('шоколад', 5),

('дендробиум', 5),

('ствол', 5),

('mito', 5),

('п', 5),

('57', 5),

('03676', 5),

('1703004', 5),

('dinamik', 5),

('косметь', 5),

('выпечка', 5),

('гель', 5),

('алиссум', 5),

('122х38', 5),

('пламень', 5),

('покрывало', 5),

('soehnle', 5),

('fragmmento', 5),

('alpina', 5),

('st194', 5),

('60х180', 5),

('кофе', 5),

('004', 5),

('1л', 5),

('а', 5),

('юниор1', 5),

('140х60', 5),

('abm106', 5),

('36', 5),

('30mt', 5),

('колба', 5),

('320', 5),

('с04', 5),

('моп', 5),

('01', 5),

('персик', 5),

('сб1г', 5),

('хлорофитум', 5),

('сб3б', 5),

('спатифиллум', 5),

('стирка', 5),

('ткань', 5),

('узор', 5),

('темнорозовый', 5),

('xl', 5),

('плющ', 5),

('средний', 5),

('лина', 5),

('мимоза', 5),

('paris', 5),

('св5', 5),

('велад', 5),

('l7997', 5),

('toscana', 5),

('подснежник', 5),

('103', 5),

('яромир', 5),

('кремль', 5),

('яр', 5),

('2406м', 5),

('банан', 5),

('мирт', 5),

('калла', 5),

('бордовый', 5),

('bassi', 5),

('тач', 5),

('tachi', 5),

('мандариновый', 5),

('a', 5),

('лилия', 5),

('hard', 5),

('2350', 5),

('котон', 5),

('5974', 5),

('verga', 5),

('вишня', 5),

('мега', 5),

('крупный', 5),

('синтетический', 5),

('лагуна', 5),

('g300a02w', 5),

('звезда', 5),

('9х9', 5),

('астра', 5),

('double', 5),

('волгоградец', 5),

('stiramaniche', 5),

('53х13', 5),

('b138lgpw', 5),

('вещь', 5),

('зверобой', 5),

('продырявливать', 5),

('мп', 5),

('настурция', 5),

('полимерный', 4),

('acr013', 4),

('туалет', 4),

('welcome', 4),

('520140', 4),

('плед', 4),

('д', 4),

('modular', 4),

('маргарита', 4),

('кремовый', 4),

('пузырек', 4),

('190', 4),

('55х55', 4),

('лавр', 4),

('daisy', 4),

('85', 4),

('пружина', 4),

('110х30', 4),

('миксер', 4),

('em', 4),

('1404094', 4),

('haushalt', 4),

('135х53', 4),

('34', 4),

('налет', 4),

('32', 4),

('95х33', 4),

('848n', 4),

('тх0252', 4),

('cannetto', 4),

('4061', 4),

('polyline', 4),

('аметист', 4),

('6230', 4),

('al', 4),

('алюминевый', 4),

('jet001', 4),

('эссенция', 4),

('fwm', 4),

('душица', 4),

('52', 4),

('slimmy', 4),

('planet', 4),

('полипропиленовый', 4),

('ананас', 4),

('115х35', 4),

('дема', 4),

('45x76', 4),

('стоунмания', 4),

('шнур', 4),

('mc355', 4),

('31', 4),

('тканевый', 4),

('валиант', 4),

('полипропилен', 4),

('40х70', 4),

('kokarda', 4),

('виктория', 4),

('pegasus', 4),

('грань', 4),

('orthospa', 4),

('river', 4),

('rocks', 4),

('светлый', 4),

('степанов', 4),

('крыло', 4),

('бензин', 4),

('зажигалка', 4),

('1205214', 4),

('1100', 4),

('морковь', 4),

('измельчитель', 4),

('шк', 4),

('1516020', 4),

('холодец', 4),

('стэм', 4),

('чай', 4),

('стекло', 4),

('воларэ', 4),

('мок', 4),

('комод', 4),

('термометр', 4),

('сб4', 4),

('1703007', 4),

('s280', 4),

('olive', 4),

('oil', 4),

('sideboard', 4),

('56х86', 4),

('5062', 4),

('гардеробная', 4),

('кекс', 4),

('стеганый', 4),

('фаленопсис', 4),

('saquet', 4),

('saq002', 4),

('80x40', 4),

('amc080', 4),

('белль', 4),

('микрофибра', 4),

('hera', 4),

('аптение', 4),

('светло-бежевый', 4),

('листопад', 4),

('работа', 4),

('airboard', 4),

('701402ng', 4),

('многофункциональный', 4),

('natural', 4),

('объемный', 4),

('джулия', 4),

('капучино', 4),

('японский', 4),

('океан', 4),

('бест', 4),

('нб', 4),

('колесо', 4),

...]

Ввод [29]:

*#создаю списки категорий с ключевыми словами, для определения категорий товаров*

house\_plants **=** ['автополив', 'аквилегия', 'алиссум', 'анемон', 'антуриум', 'арбуз', 'астра', 'бадан', 'бакоп', 'барвинок', 'бегония',

'бензин', 'бузульник', 'веник', 'вербейник', 'вербена', 'веревка', 'виола', 'г', 'гайлардий', 'гвоздик', 'георгин',

'гипсофил', 'гортензия', 'девичий', 'дендробиум', 'дыня', 'ель', 'зола', 'календула', 'калибрахоа', 'камнеломка',

'кассета', 'кашпо', 'кипарисовик', 'клен', 'клубника', 'колокольчик', 'колосковый', 'кореопсис', 'корыто', 'косметь',

'котовник', 'лапчатка', 'лен', 'лилейник', 'лобелия', 'многолетний', 'монарда', 'мусор', 'мусорный', 'настурция',

'нивянник', 'осина', 'папоротник', 'пеларгония', 'петуния', 'пикник', 'пиретрум', 'платикодон', 'почтовый',

'пьезозажигалка', 'ранункулус', 'рассада', 'роза', 'рудбекия', 'садовый', 'седум', 'солидаго', 'сумка-тележка',

'тележка', 'томат', 'тюльпан', 'уличный', 'урна', 'урна-пепельница', 'фал', 'физостегия', 'флокс', 'фуксия', 'хоста',

'хозяйственный', 'хризантема', 'цинния', 'черенок', 'шнур', 'шпагат', 'щетка-сметка', 'энотера', 'эхинацея', 'ясколка']

​

home **=** ['d-', 'd-5', 'd-7', 'd-9', 'D10', 'd-10', 'd-11', 'D12', 'D12,', 'd12', 'd-12', 'd-13', 'd-14', 'd-15',

'd-17', 'd-18', 'd-19', 'd-20', 'd-21', 'D23', 'd-23', 'бак', 'ванна', 'ванная', 'ванный', 'вантуз', 'ваза',

'ведро', 'весы', 'вешалка', 'вешалка-перекладина', 'вешалка-плечики', 'вешалка-стойка', 'вешалка-сушилка',

'вещь', 'гладильный', 'глажение', 'засор', 'зубной', 'известковый', 'искусственный', 'искуственный', 'калла',

'ключница', 'ковер', 'коврик', 'ковш', 'комнатный', 'кондиционер', 'корзина', 'коробка', 'кофр', 'крючок',

'лавр', 'лоток', 'мыло', 'мыло-скраб', 'муляж', 'наволочка', 'наматрацник', 'наматрицник-чехол',

'наматрасник', 'насадка', 'насадка-моп', 'одеяло', 'окномойка', 'отбеливатель', 'перчатка', 'плед', 'плечики',

'подголовник', 'пододеяльник', 'подрукавник', 'подушка', 'покрывало', 'пол', 'полимербыт', 'полировка',

'полотенце', 'постельный', 'прищепок', 'простыня', 'пыль', 'пятно', 'радиаторный', 'ролик', 'салфетка',

'сантехника', 'светильник', 'скребок', 'совок', 'стиральный', 'стирка', 'сушилка', 'таз', 'ткань', 'тряпка',

'тряпкодержатель', 'туалет', 'туалетный', 'увлажнять', 'унитаз', 'унитазный', 'утюг', 'фен', 'фоторамка',

'халат', 'хранение', 'цикламен,', 'чехол', 'швабра', 'щетка', 'щетка-утюжок', 'шило', 'штора']

​

dishes **=** ['антижир', 'банка', 'бидон', 'блюдо', 'блюдце', 'бокал', 'бульонница', 'бутылка', 'бутылочный', 'венчик',

'вилка', 'выпечка', 'герметичный', 'губка', 'измельчитель', 'картофелемялка', 'кастрюля', 'кекс',

'кипятильник', 'кондитерский', 'кружка', 'крышка', 'кухонный', 'лопатка', 'ложка', 'мантоварка',

'мантоварка-пароварка', 'миксер', 'мини-сковорода', 'миска', 'мука', 'нож', 'ножеточка', 'овощеварка',

'овощечистка', 'овсянница', 'орехоколка', 'отделитель', 'плита', 'половник', 'посуда', 'посудомоечный',

'пресс', 'продукт', 'противень', 'разделочный', 'рыбочистка', 'салатник', 'салфетница', 'сахарница', 'свч',

'сито', 'скалка', 'скатерть', 'сковорода', 'соковарка', 'соковыжималка', 'сотейник', 'стакан', 'столовый',

'тарелка', 'терка', 'термокружок', 'термос', 'термостакан', 'толкушка', 'тортница', 'фужер', 'хлебница',

'чайник', 'чайный', 'электроштопор', 'яйцо']

​

furniture **=** ['болт', 'вентиляционный', 'инструмент', 'карниз', 'комод', 'крепеж', 'лестница-стремянка', 'линейка', 'линейка,',

'мебельный', 'напильник', 'обувница-3', 'петля', 'петля-стрела', 'подставка', 'полк', 'полка', 'пробка', 'пуф',

'сварка', 'сверло', 'сверло-фреза,', 'свереть', 'стеллаж', 'стремянка', 'стремянка-табурет', 'строительный',

'уголок', 'угольник', 'угольник,', 'фиксатор-шар', 'форточный', 'шпингалет', 'штангенциркуль', 'этажерка']

Ввод [30]:

*#распределю товары по категориям с помощью ключевых слов*

**def** product\_category(lemmas):

**for** article **in** dishes:

**if** article **in** lemmas:

**return** 'посуда и кухня'

**for** article **in** home:

**if** article **in** lemmas:

**return** 'товары для дома'

**for** article **in** house\_plants:

**if** article **in** lemmas:

**return** 'дом и растения'

**for** article **in** furniture:

**if** article **in** lemmas:

**return** 'строительство, ремонта, мебель'

**return** 'хобби'

Ввод [31]:

*#добавляю столбец с категориями в датасет*

data\_2['category'] **=** data\_2['lemmas'].apply(product\_category)

data\_2.head()

Out[31]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** | **monetary** | **lemmas** | **category** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2018-10-01 00:00:00 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | комнатное растение в горшке алое вера, d12, h30 | 1 | 142.0 | 142.0 | комнатный растение в горшок алый вер... | товары для дома |
| **7** | 2018-10-01 08:00:00 | 375e0724-f033-4c76-b579-84969cf38ee2 | 68479 | настенная сушилка для белья gimi brio super 100 | 1 | 824.0 | 824.0 | настенный сушилка для белье gimi bri... | товары для дома |
| **8** | 2018-10-01 08:00:00 | 6644e5b4-9934-4863-9778-aaa125207701 | 68478 | таз пластмассовый 21,0 л круглый "водолей" с61... | 1 | 269.0 | 269.0 | таз пластмассовый 21 , 0 л круглый " ... | товары для дома |
| **9** | 2018-10-01 09:00:00 | c971fb21-d54c-4134-938f-16b62ee86d3b | 68480 | чехол для гладильной доски colombo persia beig... | 1 | 674.0 | 674.0 | чехол для гладильный доска colombo p... | товары для дома |
| **10** | 2018-10-01 11:00:00 | 161e1b98-45ba-4b4e-8236-e6e3e70f6f7c | 68483 | вешалка для брюк металлическая с резиновым пок... | 10 | 82.0 | 820.0 | вешалка для брюки металлический с ре... | товары для дома |

Ввод [32]:

*#удаляю столбец с леммами*

data\_2 **=** data\_2.drop('lemmas', axis**=**1)

data\_2.head()

Out[32]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** | **monetary** | **category** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2018-10-01 00:00:00 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | комнатное растение в горшке алое вера, d12, h30 | 1 | 142.0 | 142.0 | товары для дома |
| **7** | 2018-10-01 08:00:00 | 375e0724-f033-4c76-b579-84969cf38ee2 | 68479 | настенная сушилка для белья gimi brio super 100 | 1 | 824.0 | 824.0 | товары для дома |
| **8** | 2018-10-01 08:00:00 | 6644e5b4-9934-4863-9778-aaa125207701 | 68478 | таз пластмассовый 21,0 л круглый "водолей" с61... | 1 | 269.0 | 269.0 | товары для дома |
| **9** | 2018-10-01 09:00:00 | c971fb21-d54c-4134-938f-16b62ee86d3b | 68480 | чехол для гладильной доски colombo persia beig... | 1 | 674.0 | 674.0 | товары для дома |
| **10** | 2018-10-01 11:00:00 | 161e1b98-45ba-4b4e-8236-e6e3e70f6f7c | 68483 | вешалка для брюк металлическая с резиновым пок... | 10 | 82.0 | 820.0 | товары для дома |

Ввод [33]:

*# Группируем по категориям и считаем количество товаров в каждой категории*

category\_counts **=** data\_2.groupby("category").size()

category\_counts **=** category\_counts.sort\_values(ascending**=False**)

​

*# Строим гистограмму*

plt.figure(figsize**=**(10, 6)) *# размер графика*

plt.bar(category\_counts.index, category\_counts)

plt.title("Распределение товаров по категориям")

plt.xlabel("Категории")

plt.ylabel("Количество товаров")

plt.xticks(rotation**=**50)

​

*# Добавляем долю товаров в процентах над каждым столбцом*

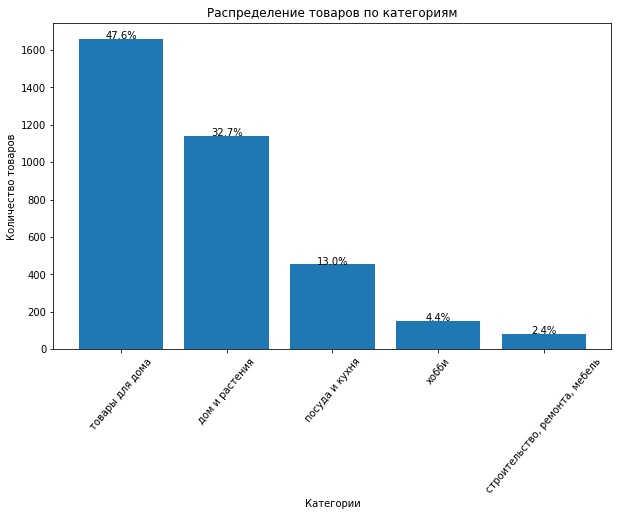
**for** i, category **in** enumerate(category\_counts.index):

plt.text(i, category\_counts[i], f"{category\_counts[i]**/**len(data\_2)**\***100:.1f}%", ha**=**"center")

​

​

plt.show()

****

С помощью библиотек pymystem3 и collections мне удалось разделить товары на такие категории как: дом и растения, товары для дома, посуда и кухня, строительство/ремонт/мебель и иное. Больше всего клиенты покупают товары в категории Дом и растения (47% от всех товаров) и меньше всего в категории Строительство/ремонт/мебель (2,3% от всех товаров).

## 4  RFM-анализ для сегментирования клиентов.

RFM — это метод, используемый для анализа потребительской ценности. Он группирует клиентов на основе истории их транзакций:

* Recency (Давность) — Как давно клиент совершил покупку?
* Frequency (Частота) — Как часто они совершают покупки?
* Monetary (Денежная ценность) — Сколько они тратят?

Ввод [34]:

*#установим дату исследования*

max\_day **=** data\_2['date'].max()

Ввод [35]:

*#создаю таблицу RFM*

RFM\_table**=**data\_2.groupby(['customer\_id']).agg({'date': **lambda** x: (max\_day **-** x.max()).days, *# Recency*

'order\_id': 'count', *# Frequency*

'monetary': 'sum'}) *# Monetary*

*#переименую столбцы*

RFM\_table.rename(columns**=**{'date': 'recency',

'order\_id': 'frequency'}, inplace**=True**)

RFM\_table.head()

Out[35]:

|  | **recency** | **frequency** | **monetary** |
| --- | --- | --- | --- |
| **customer\_id** |  |  |  |
| **000d6849-084e-4d9f-ac03-37174eaf60c4** | 107 | 1 | 150.0 |
| **001cee7f-0b29-4716-b202-0042213ab038** | 349 | 1 | 442.0 |
| **00299f34-5385-4d13-9aea-c80b81658e1b** | 108 | 1 | 914.0 |
| **002d4d3a-4a59-406b-86ec-c3314357e498** | 368 | 1 | 1649.0 |
| **003bbd39-0000-41ff-b7f9-2ddaec152037** | 123 | 1 | 2324.0 |

Ввод [36]:

*#рассмотрим распределение показателей внимательнее*

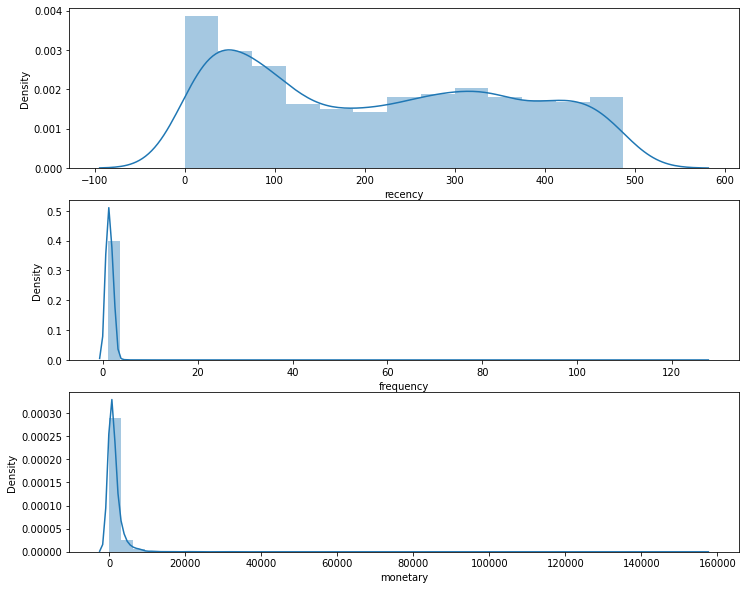
plt.figure(figsize**=**(12, 10))

plt.subplot(3, 1, 1); sns.distplot(RFM\_table['recency'])

plt.subplot(3, 1, 2); sns.distplot(RFM\_table['frequency'])

plt.subplot(3, 1, 3); sns.distplot(RFM\_table['monetary'])

plt.show()

****

Ближе всего к нормальному у нас распределение давности покупок (от последней покупки прямо сегодня до почти 500 дней бездействия), на остальных 2 параметрах мы не видим ничего нового: есть хвосты, которые мы осознанно оставили в логе, тк они являются адекватными, самое часто встречающееся количество покупок - 1-2 за раз, ну и средний чек - около 600уе.

Ввод [37]:

*#используем логорифмирование для нормализации распределений*

RFM\_table['recency\_log'] **=** np.where(RFM\_table['recency'] **==** 0, 0, np.log(RFM\_table['recency'].replace(0,1)))

RFM\_table['frequency\_log'] **=** np.where(RFM\_table['recency'] **==** 0, 0, np.log(RFM\_table['frequency'].replace(0,1)))

RFM\_table['monetary\_log'] **=** np.where(RFM\_table['recency'] **==** 0, 0, np.log(RFM\_table['monetary'].replace(0,1)))

Ввод [38]:

*#еще раз рассмотрим распределение показателей внимательнее*

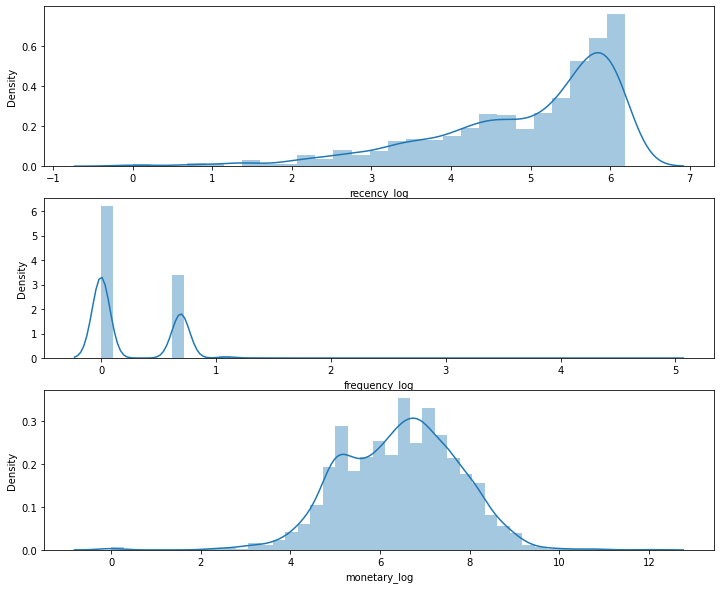
plt.figure(figsize**=**(12, 10))

plt.subplot(3, 1, 1); sns.distplot(RFM\_table['recency\_log'])

plt.subplot(3, 1, 2); sns.distplot(RFM\_table['frequency\_log'])

plt.subplot(3, 1, 3); sns.distplot(RFM\_table['monetary\_log'])

plt.show()

****

По логарифмированным данным можно отметить что самым частым периодом "молчания" между покупками является отрезок в 6 дней. По количеству покупок мы можем видеть новый пик около нуля: можно связать это долей клиентов, которые возможно еще не совершили покупку. Данные по прибыли выглядят нормальными.

Ввод [39]:

*#разбиваем показатели на равные группы-оценки для дальнейшего сегментирования покупателей*

r\_labels **=** range(4, 0, **-**1); f\_labels **=** range(1, 5); m\_labels **=** range(1, 5)

r\_groups **=** pd.cut(RFM\_table['recency\_log'], bins**=**4, labels**=**r\_labels)

f\_groups **=** pd.cut(RFM\_table['frequency\_log'], bins**=**4, labels**=**f\_labels)

m\_groups **=** pd.cut(RFM\_table['monetary\_log'], bins**=**4, labels**=**m\_labels)

RFM\_table **=** RFM\_table.assign(R **=** r\_groups.values, F **=** f\_groups.values, M **=** m\_groups.values)

RFM\_table.head()

Out[39]:

|  | **recency** | **frequency** | **monetary** | **recency\_log** | **frequency\_log** | **monetary\_log** | **R** | **F** | **M** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **customer\_id** |  |  |  |  |  |  |  |  |  |
| **000d6849-084e-4d9f-ac03-37174eaf60c4** | 107 | 1 | 150.0 | 4.672829 | 0.0 | 5.010635 | 1 | 1 | 2 |
| **001cee7f-0b29-4716-b202-0042213ab038** | 349 | 1 | 442.0 | 5.855072 | 0.0 | 6.091310 | 1 | 1 | 3 |
| **00299f34-5385-4d13-9aea-c80b81658e1b** | 108 | 1 | 914.0 | 4.682131 | 0.0 | 6.817831 | 1 | 1 | 3 |
| **002d4d3a-4a59-406b-86ec-c3314357e498** | 368 | 1 | 1649.0 | 5.908083 | 0.0 | 7.407924 | 1 | 1 | 3 |
| **003bbd39-0000-41ff-b7f9-2ddaec152037** | 123 | 1 | 2324.0 | 4.812184 | 0.0 | 7.751045 | 1 | 1 | 3 |

Я оценила показатели (напомню их значение: R-давность, F-частота, M-денежная ценность) по трехбальной шкале: чем выше оценка, тем лучше, например: клиент с RFM-индексом 444-покупал совсем недавно на большую сумму и делает это максимально часто - идеальный клиент, которого мы обязательно включим в рекламную рассылку, потому что она обязательно окупится!

Ввод [40]:

rfm\_heatmap **=** RFM\_table.pivot\_table(index**=**'F', columns**=**'R', values**=**'monetary', aggfunc**=**np.mean)

plt.figure(figsize**=**(10, 8))

sns.heatmap(rfm\_heatmap, annot**=True**, fmt **=** '.2f', linewidths**=**1)

plt.title('Выручка по сегментам')

plt.xlabel('Давность последней покупки')

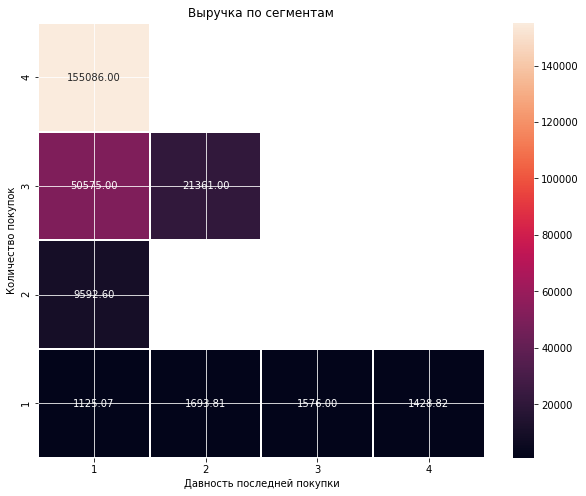
plt.ylabel('Количество покупок')

plt.grid(color**=**'white')

plt.gca().invert\_xaxis()

plt.gca().invert\_yaxis()

plt.show()

****

На тепловой карте по выручке по сегментам мы видим, что есть пользователи, котрые совершали покупки на достаточно крупные суммы, но с их последней покупки прошло уже значительное количество времени - эти пользователи важны для нас и их стоит стараться сохранить в роли своих покупателей.

Ввод [41]:

*#добавлю новый столбец rfm\_index, который хранит индекс клиента, полученный при RFM-сегментации*

RFM\_table['rfm\_index'] **=** RFM\_table['R'].astype(str) **+** RFM\_table['F'].astype(str) **+** RFM\_table['M'].astype(str)

Ввод [42]:

*#посмотрим на финальную таблицу, которая поможет ответить нам на основные вопросы бизнеса*

RFM\_table.head()

Out[42]:

|  | **recency** | **frequency** | **monetary** | **recency\_log** | **frequency\_log** | **monetary\_log** | **R** | **F** | **M** | **rfm\_index** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **customer\_id** |  |  |  |  |  |  |  |  |  |  |
| **000d6849-084e-4d9f-ac03-37174eaf60c4** | 107 | 1 | 150.0 | 4.672829 | 0.0 | 5.010635 | 1 | 1 | 2 | 112 |
| **001cee7f-0b29-4716-b202-0042213ab038** | 349 | 1 | 442.0 | 5.855072 | 0.0 | 6.091310 | 1 | 1 | 3 | 113 |
| **00299f34-5385-4d13-9aea-c80b81658e1b** | 108 | 1 | 914.0 | 4.682131 | 0.0 | 6.817831 | 1 | 1 | 3 | 113 |
| **002d4d3a-4a59-406b-86ec-c3314357e498** | 368 | 1 | 1649.0 | 5.908083 | 0.0 | 7.407924 | 1 | 1 | 3 | 113 |
| **003bbd39-0000-41ff-b7f9-2ddaec152037** | 123 | 1 | 2324.0 | 4.812184 | 0.0 | 7.751045 | 1 | 1 | 3 | 113 |

Ввод [43]:

*#создадим новый датафрейм с данными по клиентам и их сегментам*

RFM\_table\_1 **=** RFM\_table.drop(['recency', 'frequency', 'monetary', 'recency\_log', 'frequency\_log', 'monetary\_log'], axis**=**1)

merged\_df **=** pd.merge(data\_2, RFM\_table\_1, on**=**'customer\_id')

merged\_df.head()

Out[43]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** | **monetary** | **category** | **R** | **F** | **M** | **rfm\_index** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2018-10-01 00:00:00 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | комнатное растение в горшке алое вера, d12, h30 | 1 | 142.0 | 142.0 | товары для дома | 1 | 1 | 2 | 112 |
| **1** | 2018-10-01 08:00:00 | 375e0724-f033-4c76-b579-84969cf38ee2 | 68479 | настенная сушилка для белья gimi brio super 100 | 1 | 824.0 | 824.0 | товары для дома | 1 | 1 | 3 | 113 |
| **2** | 2018-10-01 08:00:00 | 6644e5b4-9934-4863-9778-aaa125207701 | 68478 | таз пластмассовый 21,0 л круглый "водолей" с61... | 1 | 269.0 | 269.0 | товары для дома | 2 | 1 | 2 | 212 |
| **3** | 2019-10-31 16:00:00 | 6644e5b4-9934-4863-9778-aaa125207701 | 111409 | рассада дыни сорт казачка, горшок 9\*9см | 1 | 38.0 | 38.0 | дом и растения | 2 | 1 | 2 | 212 |
| **4** | 2018-10-01 09:00:00 | c971fb21-d54c-4134-938f-16b62ee86d3b | 68480 | чехол для гладильной доски colombo persia beig... | 1 | 674.0 | 674.0 | товары для дома | 1 | 4 | 4 | 144 |

Ввод [44]:

*#добавим столбец seasons, для анализа данных в разрезе сезонов*

*# создаем список сезонов*

season **=** ['зима', 'весна', 'лето', 'осень']

*# создаем новый столбец с сезоном*

merged\_df['season'] **=** pd.cut(merged\_df['date'].dt.month, [0, 3, 6, 9, 12], labels**=**season)

merged\_df.head()

Out[44]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** | **monetary** | **category** | **R** | **F** | **M** | **rfm\_index** | **season** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 2018-10-01 00:00:00 | ee47d746-6d2f-4d3c-9622-c31412542920 | 68477 | комнатное растение в горшке алое вера, d12, h30 | 1 | 142.0 | 142.0 | товары для дома | 1 | 1 | 2 | 112 | осень |
| **1** | 2018-10-01 08:00:00 | 375e0724-f033-4c76-b579-84969cf38ee2 | 68479 | настенная сушилка для белья gimi brio super 100 | 1 | 824.0 | 824.0 | товары для дома | 1 | 1 | 3 | 113 | осень |
| **2** | 2018-10-01 08:00:00 | 6644e5b4-9934-4863-9778-aaa125207701 | 68478 | таз пластмассовый 21,0 л круглый "водолей" с61... | 1 | 269.0 | 269.0 | товары для дома | 2 | 1 | 2 | 212 | осень |
| **3** | 2019-10-31 16:00:00 | 6644e5b4-9934-4863-9778-aaa125207701 | 111409 | рассада дыни сорт казачка, горшок 9\*9см | 1 | 38.0 | 38.0 | дом и растения | 2 | 1 | 2 | 212 | осень |
| **4** | 2018-10-01 09:00:00 | c971fb21-d54c-4134-938f-16b62ee86d3b | 68480 | чехол для гладильной доски colombo persia beig... | 1 | 674.0 | 674.0 | товары для дома | 1 | 4 | 4 | 144 | осень |

Ввод [45]:

*#рассмотрим количество клиентов в сегменте, в котором покупают много/часто/дорогой товар*

merged\_df[merged\_df['rfm\_index']**==**'444'].sort\_values('monetary', ascending**=False**).head()

Out[45]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** | **monetary** | **category** | **R** | **F** | **M** | **rfm\_index** | **season** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Настолько идеальных клиентов в нашем датафрейме, к сожалению нет, это и не удевительно: как мы выяснили ранее, большинство предпочитает покупать недорогой товар.

Ввод [46]:

*#есть ли у нас потерянные клиенты?*

merged\_df[merged\_df['rfm\_index']**==**'111'].sort\_values('monetary', ascending**=False**).head(10)

Out[46]:

|  | **date** | **customer\_id** | **order\_id** | **product** | **quantity** | **price** | **monetary** | **category** | **R** | **F** | **M** | **rfm\_index** | **season** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **642** | 2018-12-03 11:00:00 | 2c8b9386-2e8c-4ee9-8aa4-2b4875b68578 | 14514 | алиссум (лобулярия) снежные одежды 0,2 г 46500... | 2 | 9.0 | 18.0 | дом и растения | 1 | 1 | 1 | 111 | осень |
| **515** | 2018-11-13 11:00:00 | 2330d859-e9cb-4c8f-abd0-55f9e27e6745 | 68985 | паста для полировки "гои" 25 гр | 1 | 15.0 | 15.0 | товары для дома | 1 | 1 | 1 | 111 | осень |
| **2617** | 2019-07-09 17:00:00 | 42d3cb27-6519-4627-9b4a-5aa8ee600203 | 14897 | базилик зеленый тонус d-7 см | 1 | 15.0 | 15.0 | дом и растения | 1 | 1 | 1 | 111 | лето |
| **594** | 2018-11-25 14:00:00 | 52aff75c-a972-4c25-bad5-dfe6a48cd981 | 14512 | бархатцы веселая полянка 0,3 г 4660010777505 | 1 | 11.0 | 11.0 | товары для дома | 1 | 1 | 1 | 111 | осень |
| **2022** | 2019-04-17 20:00:00 | f83eedd3-aaad-48ed-9977-55f6046477a3 | 14655 | базилик застольный 0,3 г 4660010775419 | 1 | 11.0 | 11.0 | дом и растения | 1 | 1 | 1 | 111 | весна |
| **3199** | 2019-09-21 21:00:00 | 6718a7fc-75c2-4b8a-820d-b2960be23ca7 | 14932 | базилик застольный 0,3 г 4660010775419 | 1 | 11.0 | 11.0 | дом и растения | 1 | 1 | 1 | 111 | лето |
| **2571** | 2019-07-03 22:00:00 | 1ec1ad59-a29b-4949-929d-f4fc4fe14758 | 14887 | алиссум (лобулярия) снежные одежды 0,2 г 46500... | 1 | 9.0 | 9.0 | дом и растения | 1 | 1 | 1 | 111 | лето |

Всего 7 клиент, который совершил всего одну покупку много мемсяцев назад на сумму до 18уе и больше не "воскресали" - это тоже очень хороший результат.

Ввод [47]:

*#сколько у нас лояльных клиентов, совершивших свыше среднего числа покупок?*

print(len(merged\_df[merged\_df['F'] **>=** 3]))

178

У магазина 178 лояльных покупателя, которые за весь период совершили от 3 и более покупок, их без сомнения нужно включить в рассылку.

Ввод [48]:

*#сколько у нас клиентов, которые совершаю покупки стоимостью выше среднего?*

print(len(merged\_df[merged\_df['M'] **==** 1]))

18

Всего 17 клиентов во всем датафрейме, которые совершали дорогие покупки за анализируемы период.

Ввод [49]:

*# функция для более наглядной и понятной категоризации клиентов*

**def** rfm\_category(row):

r\_score **=** row['R']

f\_score **=** row['F']

m\_score **=** row['M']

**if** r\_score **<=** 2 **and** f\_score **<=** 2 **and** m\_score **<=** 2:

**return** 'в зоне бездействия'

**elif** r\_score **<=** 2 **and** f\_score **<=** 2 **and** m\_score **<=** 3:

**return** 'в зоне риска'

**elif** r\_score **<=** 2 **and** f\_score **<=** 2 **and** m\_score **>=** 4:

**return** 'не можем потерять'

**elif** r\_score **<=** 2 **and** f\_score **<=** 3 **and** m\_score **<=** 2:

**return** 'засыпающий'

**elif** r\_score **<=** 2 **and** f\_score **<=** 3 **and** m\_score **>=** 3:

**return** 'нуждающихся во внимании'

**elif** r\_score **>=** 3 **and** f\_score **>=** 3 **and** m\_score **>=** 3:

**return** 'постоянный пользователь'

**elif** r\_score **>=** 3 **and** f\_score **>=** 3 **and** m\_score **>=** 4:

**return** 'перспективный пользователь'

**elif** r\_score **>=** 3 **and** f\_score **>=** 4 **and** m\_score **>=** 4:

**return** 'потенциальный лоялист'

​

*# применение функции к датасету*

merged\_df['category\_2'] **=** merged\_df.apply(rfm\_category, axis**=**1)

​

*# вывод таблицы с количеством клиентов в каждой категории*

print(merged\_df['category\_2'].value\_counts())

в зоне риска 1935

в зоне бездействия 932

не можем потерять 53

нуждающихся во внимании 52

Name: category\_2, dtype: int64

Ввод [50]:

*# создаем сводную таблицу*

pivot\_df **=** merged\_df.pivot\_table(index**=**'category', columns**=**'category\_2', values**=**'customer\_id', aggfunc**=**'count')

​

*# строим гистограмму*

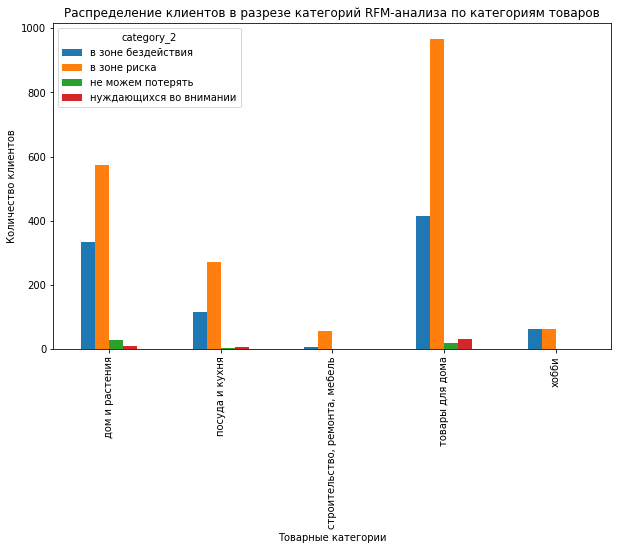
pivot\_df.plot.bar(figsize**=**(10, 6))

plt.xlabel('Товарные категории')

plt.ylabel('Количество клиентов')

plt.title('Распределение клиентов в разрезе категорий RFM-анализа по категориям товаров')

plt.show()

****

По значениям, которые у нас есть по нашим клиентам мы получаем 4 категории: "в зоне риска" - 1977 человек, "в зоне бездействия" - 949 человек, "нуждающихся во внимании" - 52 человек, "не можем потерять" - 53 человек. В зоне риска находятся те пользователи, которые совершали покупки давно и в среднем количестве. В зоне бездействия находятся малоактивные пользователи, заходили давно и покупали мало. Не можем потерять - это люди, которые делали много покупок, но много дней назад. В зоне бездействия-находятся покупатели на грани оттока, которых важно не потерять и нужно активировать. Как мы видим из графиков выше и ниже излюбленной категорией товаров для всех сегментов покупателей является "дом и растения" и "товары для дома", по ним разделены покупатели с самым распространенным чеком - со средним - это сегммент "в зоне риска"и "нуждающиеся во внимании". Меньше всего внимания сегменты покупают товаров для строительства и ремонта, а так же есть единичные покупки в категории "иное".

Ввод [51]:

*#рассмотрим как распределена активность сегментов покупателей в разрезе сезонов*

plt.figure(figsize**=**(8, 6))

sns.histplot(x**=**'season', hue**=**'category\_2', data**=**merged\_df)

​

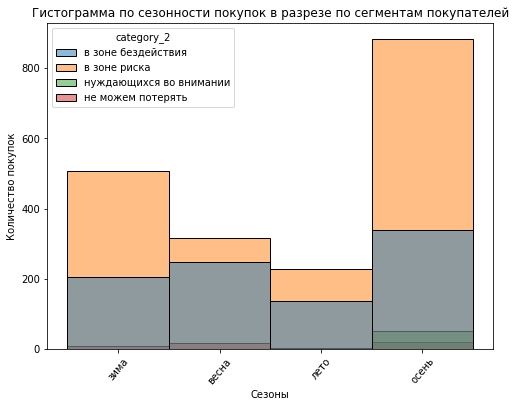
plt.ylabel('Количество покупок')

plt.xlabel('Сезоны')

plt.xticks(rotation**=**50)

plt.title('Гистограмма по сезонности покупок в разрезе по сегментам покупателей');

​

****

В покупках в разрезе по сезонам все так же лидируют "в зоне риска"и "нуждающиеся во внимании" в сезоны весна/осень, а летом продажи по каждому из сегментов покупателей значительно падают.

Ввод [52]:

*#рассмотрим как распределена активность клиентов в разрезе категорий товаров и сезонов*

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

sns.histplot(x**=**'season', hue**=**'category', data**=**merged\_df)

​

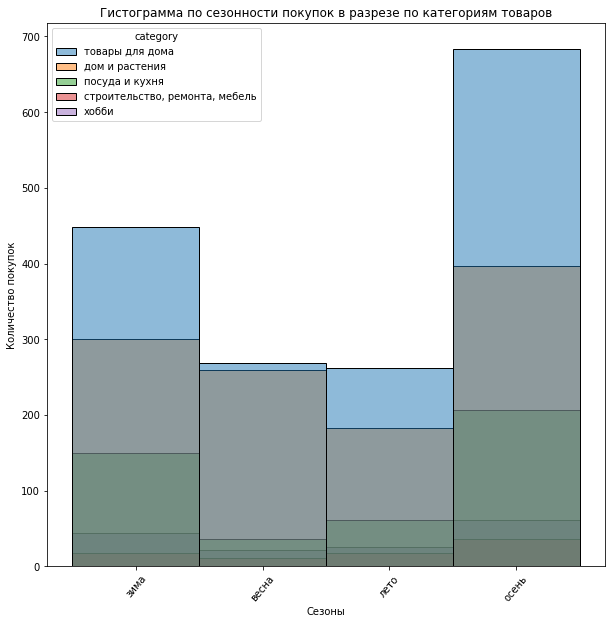
plt.ylabel('Количество покупок')

plt.xlabel('Сезоны')

plt.xticks(rotation**=**50)

plt.title('Гистограмма по сезонности покупок в разрезе по категориям товаров');

​

****

Ввод [53]:

*# группируем данные по категориям покупателей и вычисляем средний чек*

grouped\_df **=** merged\_df.groupby('category\_2')['monetary'].mean()

​

*# выводим результаты*

print(grouped\_df)

category\_2

в зоне бездействия 162.850680

в зоне риска 1182.981008

не можем потерять 8086.094340

нуждающихся во внимании 1383.384615

Name: monetary, dtype: float64

Товары для дома и сада закономерно больше всего покупают в конце зимы и в течении всей весны, осенью продажи в данной категории стремятся к нулю; на все остальные категории товаров сезонность не оказывает заметного влияния.

Ввод [54]:

*# Отфильтровываем merged\_df по категории "в зоне риска"*

risk\_df\_1 **=** merged\_df.loc[merged\_df['category\_2'] **==** 'в зоне риска']

​

*# Строим гистограмму*

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

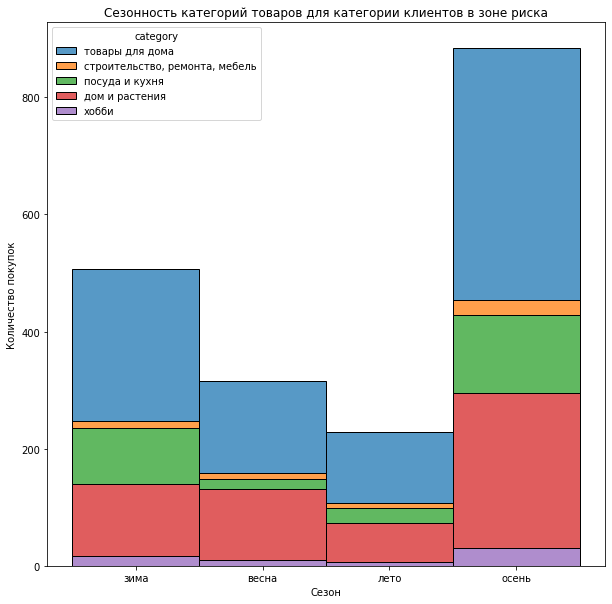
sns.histplot(data**=**risk\_df\_1, x**=**'season', hue**=**'category', multiple**=**'stack', binwidth**=**0.5)

plt.title('Сезонность категорий товаров для категории клиентов в зоне риска')

plt.xlabel('Сезон')

plt.ylabel('Количество покупок')

plt.show()

****

На графике по данным категории клиентов "в зоне риска" мы видим, в течение всего года они отдают предпочтение товарам в гатегории "товары для дома", товары в категории "дом и растения" чуть чаще покупают осенью, на остальные категории товаров для данной категории клиентов сезонность не оказывает значительного влияния.

Ввод [55]:

*# Отфильтровываем merged\_df по категории "в зоне риска"*

risk\_df\_2 **=** merged\_df.loc[merged\_df['category\_2'] **==** 'в зоне бездействия']

​

*# Строим гистограмму*

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

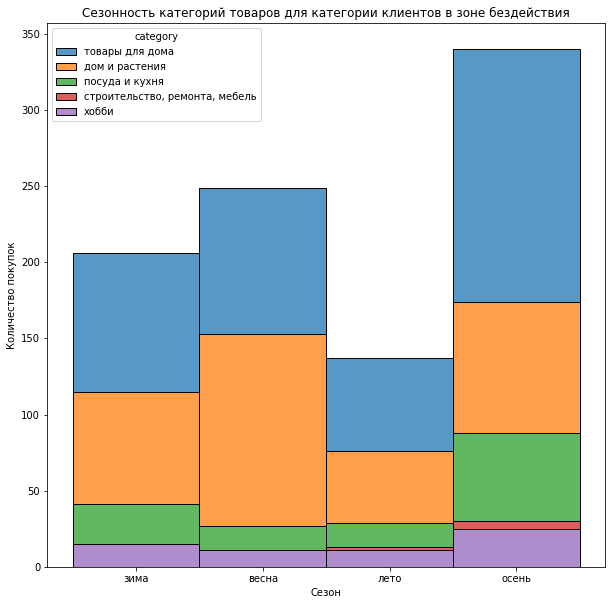
sns.histplot(data**=**risk\_df\_2, x**=**'season', hue**=**'category', multiple**=**'stack', binwidth**=**0.5)

plt.title('Сезонность категорий товаров для категории клиентов в зоне бездействия')

plt.xlabel('Сезон')

plt.ylabel('Количество покупок')

plt.show()

****

Клиенты же в категории "в зоне бездействия" осенью отдают предпочтение покупкам в категории "товары для дома, а весной товарам в категории "дом и растения".

Ввод [56]:

*# Отфильтровываем merged\_df по категории "в зоне риска"*

risk\_df\_3 **=** merged\_df.loc[merged\_df['category\_2'] **==** 'не можем потерять']

​

*# Строим гистограмму*

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

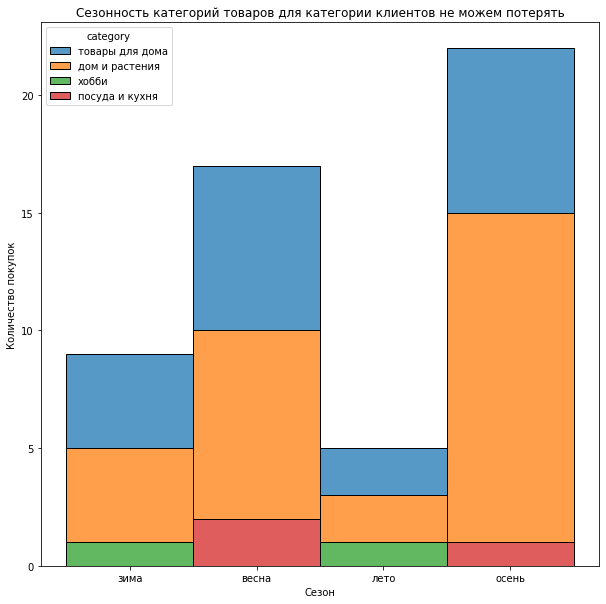
sns.histplot(data**=**risk\_df\_3, x**=**'season', hue**=**'category', multiple**=**'stack', binwidth**=**0.5)

plt.title('Сезонность категорий товаров для категории клиентов не можем потерять')

plt.xlabel('Сезон')

plt.ylabel('Количество покупок')

plt.show()

****

Клиенты из категирии "не можем потерять" предпочитают покупать больше всего товаров осенью и весной и в категории "дом и растения".

Ввод [57]:

*# Отфильтровываем merged\_df по категории "в зоне риска"*

risk\_df\_4 **=** merged\_df.loc[merged\_df['category\_2'] **==** 'нуждающихся во внимании']

​

*# Строим гистограмму*

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

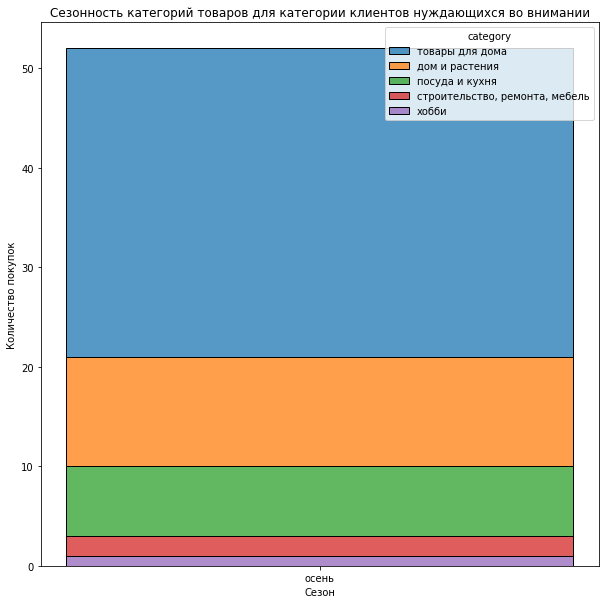
sns.histplot(data**=**risk\_df\_4, x**=**'season', hue**=**'category', multiple**=**'stack', binwidth**=**0.5)

plt.title('Сезонность категорий товаров для категории клиентов нуждающихся во внимании')

plt.xlabel('Сезон')

plt.ylabel('Количество покупок')

plt.show()

****

Катеория клиенов "нуждающихся во внимании" вообще совершает покупки только осенью, в приоритете у них так же категории "товары для дома" и "дом и растения".

## 5  Формирование и проверка статистических гипотез.

Чтобы рассчитать статистическую значимость различий в среднем чеке между категориями клиентов полученными в ходе RFM-анализа, применим критерий Манна-Уитни.

H0: Средний чек по категориям покупателей различается не существенно.

H1: Наблюдаются существенные различия средних чеков по категориям покупателей .

Ввод [58]:

*# группируем данные по категориям покупателей и вычисляем средний чек*

grouped\_df **=** merged\_df.groupby('category\_2')['price'].mean().reset\_index()

​

*# создаем таблицу для хранения результатов теста Манна-Уитни*

results\_df **=** pd.DataFrame(columns**=**['category\_1', 'category\_2', 'p\_value'])

​

*# проходимся по всем парам категорий и вычисляем статистическую значимость различий в среднем чеке*

n\_comparisons **=** len(grouped\_df) **\*** (len(grouped\_df)**-**1) **/** 2

**for** i **in** range(len(grouped\_df)):

**for** j **in** range(i**+**1, len(grouped\_df)):

category\_1 **=** grouped\_df.loc[i, 'category\_2']

category\_2 **=** grouped\_df.loc[j, 'category\_2']

prices\_1 **=** merged\_df.loc[merged\_df['category\_2']**==**category\_1, 'price']

prices\_2 **=** merged\_df.loc[merged\_df['category\_2']**==**category\_2, 'price']

\_, p\_value **=** mannwhitneyu(prices\_1, prices\_2)

adjusted\_p\_value **=** p\_value **\*** n\_comparisons

results\_df **=** results\_df.append({'category\_1': category\_1, 'category\_2': category\_2, 'p\_value': adjusted\_p\_value}, ignore\_index**=True**)

​

*# добавляем столбец с результатом теста (отвергаем нулевую гипотезу, если p\_value < 0.05)*

results\_df['result'] **=** results\_df['p\_value'] **<** adjusted\_p\_value

​

*# выводим результаты*

print(results\_df)

category\_1 category\_2 p\_value result

0 в зоне бездействия в зоне риска 1.616325e-173 True

1 в зоне бездействия не можем потерять 2.364737e-25 True

2 в зоне бездействия нуждающихся во внимании 3.815718e-16 True

3 в зоне риска не можем потерять 1.183775e-11 True

4 в зоне риска нуждающихся во внимании 4.977755e+00 False

5 не можем потерять нуждающихся во внимании 3.741648e-06 False

Столбец result показывает отвергается ли нулевая гипотеза (True - отвергается, False - не отвергается). Н0 по результатам теста отвергается: наблюдаются существенные различия средних чеков по сегментам покупателей, для всех сравниваемых пар, кроме пар в зоне риска/нуждающихся во внимании и не можем потерять/нуждающихся во внимании.

Чтобы рассчитать статистическую значимость различий в среднем количестве товаров в заказемежду категориями клиентов полученными в ходе RFM-анализа, применим критерий Манна-Уитни.

H0: Разница в среднем количестве товаров в заказе по категориям клиентов не существенна.

H1: Разница в среднем количестве товаров в заказе по категориям клиентов статистически значима.

Ввод [59]:

*# группируем данные по категориям покупателей и вычисляем среднее количество товаров в заказе*

grouped\_df **=** merged\_df.groupby('category\_2')['quantity'].mean().reset\_index()

​

*# создаем таблицу для хранения результатов теста Манна-Уитни*

results\_df **=** pd.DataFrame(columns**=**['category\_1', 'category\_2', 'p\_value'])

​

*# проходимся по всем парам категорий и вычисляем статистическую значимость различий в среднем количестве товаров в заказе*

n\_comparisons **=** len(grouped\_df) **\*** (len(grouped\_df)**-**1) **/** 2

**for** i **in** range(len(grouped\_df)):

**for** j **in** range(i**+**1, len(grouped\_df)):

category\_1 **=** grouped\_df.loc[i, 'category\_2']

category\_2 **=** grouped\_df.loc[j, 'category\_2']

quantities\_1 **=** merged\_df.loc[merged\_df['category\_2']**==**category\_1, 'quantity']

quantities\_2 **=** merged\_df.loc[merged\_df['category\_2']**==**category\_2, 'quantity']

\_, p\_value **=** mannwhitneyu(quantities\_1, quantities\_2)

adjusted\_p\_value **=** p\_value **\*** n\_comparisons

results\_df **=** results\_df.append({'category\_1': category\_1, 'category\_2': category\_2, 'p\_value': adjusted\_p\_value}, ignore\_index**=True**)

​

*# добавляем столбец с результатом теста (отвергаем нулевую гипотезу, если p\_value < 0.05)*

results\_df['result'] **=** results\_df['p\_value'] **<** adjusted\_p\_value

​

*# выводим результаты*

print(results\_df)

category\_1 category\_2 p\_value result

0 в зоне бездействия в зоне риска 3.281707 False

1 в зоне бездействия не можем потерять 0.002970 True

2 в зоне бездействия нуждающихся во внимании 2.183644 False

3 в зоне риска не можем потерять 0.006688 True

4 в зоне риска нуждающихся во внимании 2.826746 False

5 не можем потерять нуждающихся во внимании 0.807463 False

Столбец result показывает отвергается ли нулевая гипотеза (True - отвергается, False - не отвергается). Н0 по результатам теста отвергается для пар в зоне бездействия/не можем потерять, в зоне риска/не можем потерять.

## 6  Выводы и рекомендации.

В ходе работы мною были описаны полученные данные, изменен тип данных там, где это было необходимо, созданы новые столбцы для дальнейших рассчетов, удалены явные и неявные дубликаты, выявлены и удалены аномалии в данных. Проведен исследовательский анализ данных при помощи методов визуализации; товары разделены на категории при помощи лемматизации; проведен RFM-анализ для разделения на категории самих клиентов с учетом частоты совершаемых ими покупок, их стоимости и давности совершенных покупок; профили клиентов рассмотрены в разрезе категорий товаров и сезонов; проверены 2 заявленные гипотезы с помощью метода Манна-Уитни.

При помощи лемматизации товары были разделены мною на такие категории:

* товары для дома;
* дом и растения;
* посуда и кухня;
* строительство/ремонт/мебель;
* иное.

RFM-анализ выявил наличие клиентов в таких категориях как:

1) в зоне риска - 1935 ед - совершили покупки в течении послених 250 дней с частотой примерно 1 раз в 6 дней, предпочитают такие категори товаров как "товары для дома" и "дом и растения", сезон активности - весь год, средний чек - 1182 уе

* включить в круглогодичную рекламную рассылку на категории "товары для дома" и "дом и растения";

2) в зоне бездействия - 932 ед - совершили покупки в течении послених 250 дней с частотой примерно 1 раз в 6 дней, предпочитают такие категори товаров как "товары для дома" осенью, а весной товарам в категории "дом и растения", средний чек - 162 уе

* рассылка осень на категорию "товары для дома", а весной в категории "дом и растения";

3) нуждающиеся во внимании - 52 ед - совершили покупки в течении послених 250 дней с частотой примерно 1 раз в 6 дней, совершает покупки только осенью, в приоритете такие категории товаров "товары для дома" и "дом и растения", средний чек - 1383 уе

* рассылка осень на категории "товары для дома" и "дом и растения";

4) не можем потерять - 53 ед - совершили покупки в течении послених 250 дней с частотой примерно 1 раз в 6 дней, предпочитают покупать больше всего товаров осенью и весной и в категории "дом и растения", средний чек - 8086 уе

* рассылка осенью и весной и в категории "дом и растения".

Возможно увеличить количество продаваемых товаров в таких категориях как "товары для дома" и "дом и растения", это поможет повысить возвращаемость покупателей.

# Ссылка на презентацию: <https://drive.google.com/file/d/1RJOkJZL_0VX_A70itwC_4UD2yC5r92BO/view?usp=drivesdk>