Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

«Наглядный вероятностно-статистический анализ данных»

Практическое занятие 2

«Визуализация эмпирических распределений средствами Python»

Пройдакова Екатерина Вадимовна,

доцент кафедры ТВиАД ИИТММ

Содержание

- Пример: данные из онлайн-магазина
- Визуализация эмпирических распределений: matplotlib
- Визуализация эмпирических распределений: seaborn
- Практическое задание

1. ПРИМЕР: ДАННЫЕ ИЗ ОНЛАЙН-МАГАЗИНА

1. Пример: данные из онлайн-магазина
1.1 Семантика данных

	A	B	C	D	E	F	G	H	I
1	event_time	event_type	product_id	category_id	brand	price	user_id	user_session	
2	2019-11-01 00:00:02 UTC	view	5802432	1487580009286598681		0.32	562076640	09fafd6c-6c99-46b1-834f-33527f4de241	
3	2019-11-01 00:00:09 UTC	cart	5844397	1487580006317032337		2.38	553329724	2067216c-31b5-455d-a1cc-af0575a34ffb	
4	2019-11-01 00:00:10 UTC	view	5837166	1783999064103190764	pnb	22.22	556138645	57ed222e-a54a-4907-9944-5a875c2d7f4f	
5	2019-11-01 00:00:11 UTC	cart	5876812	1487580010100293687	jessnail	3.16	564506666	186c1951-8052-4b37-adce-dd9644b1d5f7	
6	2019-11-01 00:00:24 UTC	remove_from_c	5826182	1487580007483048900		3.33	553329724	2067216c-31b5-455d-a1cc-af0575a34ffb	
7	2019-11-01 00:00:24 UTC	remove_from_c	5826182	1487580007483048900		3.33	553329724	2067216c-31b5-455d-a1cc-af0575a34ffb	
8	2019-11-01 00:00:25 UTC	view	5856189	1487580009026551821	runail	15.71	562076640	09fafd6c-6c99-46b1-834f-33527f4de241	
9	2019-11-01 00:00:32 UTC	view	5837835	1933472286753424063		3.49	514649199	432a4e95-375c-4b40-bd36-0fc039e77580	

- Определим тип данных по каждому столбцу:
  - **price** – непрерывные количественные данные.
  - **event_type, brand** – номинальные качественные данные.
  - **product_id, category_id, user_id** – дискретные количественные данные.

1. Пример: данные из онлайн-магазина
1.2 Случайные величины

	A	B	C	D	E	F	G	H	I
1	event_time	event_type	product_id	category_id	brand	price	user_id	user_session	
2	2019-11-01 00:00:02 UTC	view	5802432	1487580009286598681		0.32	562076640	09fafd6c-6c99-46b1-834f-33527f4de241	
3	2019-11-01 00:00:09 UTC	cart	5844397	1487580006317032337		2.38	553329724	2067216c-31b5-455d-a1cc-af0575a34ffb	
4	2019-11-01 00:00:10 UTC	view	5837166	1783999064103190764	pnb	22.22	556138645	57ed222e-a54a-4907-9944-5a875c2d7f4f	
5	2019-11-01 00:00:11 UTC	cart	5876812	1487580010100293687	jessnail	3.16	564506666	186c1951-8052-4b37-adce-dd9644b1d5f7	
6	2019-11-01 00:00:24 UTC	remove_from_c	5826182	1487580007483048900		3.33	553329724	2067216c-31b5-455d-a1cc-af0575a34ffb	
7	2019-11-01 00:00:24 UTC	remove_from_c	5826182	1487580007483048900		3.33	553329724	2067216c-31b5-455d-a1cc-af0575a34ffb	
8	2019-11-01 00:00:25 UTC	view	5856189	1487580009026551821	runail	15.71	562076640	09fafd6c-6c99-46b1-834f-33527f4de241	
9	2019-11-01 00:00:32 UTC	view	5837835	1933472286753424063		3.49	514649199	432a4e95-375c-4b40-bd36-0fc039e77580	

- $\xi_1, \xi_2, \xi_3, \xi_4$ – стоимость произвольного продукта, просматриваемого на сайте (событие view), добавленного в корзину (событие cart), удаленного из корзины (событие remove_from_cart), купленного продукта (событие purchase);
- $\xi \in \{\xi_1, \xi_2, \xi_3, \xi_4\}$ – стоимость произвольного продукта, с которым любым образом взаимодействует пользователь (событие любого типа).
- Объем выборки для ξ - 500 наблюдений

1. Пример: данные из онлайн-магазина
1.3 Объем выборки

```
import xlrd
#указывается полный путь до файла
book = xlrd.open_workbook('.../02_Онлайн_продажи.xlsx')
#извлекаем лист с данными по индексу
sheet = book.sheet_by_index(0)

event_types = [] # список всех наблюдаемых событий
prices = [] # список стоимостей всех продуктов
prices_on_event = {} # словарь стоимостей продуктов по типу события

sample_size = sheet.nrows-1 # объем выборки

for i in range(sample_size):
    event_types.append(sheet.cell_value(i+1, 1))
    prices.append(float(sheet.cell_value(i+1, 5)))
    if prices_on_event.get(sheet.cell_value(i+1, 1)) == None: # проверка, присутствует ли элемент с таким именем в словаре
        prices_on_event[sheet.cell_value(i+1, 1)] = [] # если нет - создаем новую запись, в качестве значения - пустой список
    prices_on_event[sheet.cell_value(i+1, 1)].append(float(sheet.cell_value(i+1, 5))) # добавляем стоимость в соответствующий спис


from collections import Counter
event_types_counter = Counter(event_types) #словарь частотности типов событий

# то же самое, другим способом
event_types_dict = {}
for type in prices_on_event.keys():
    event_types_dict[type] = len(prices_on_event[type])

```

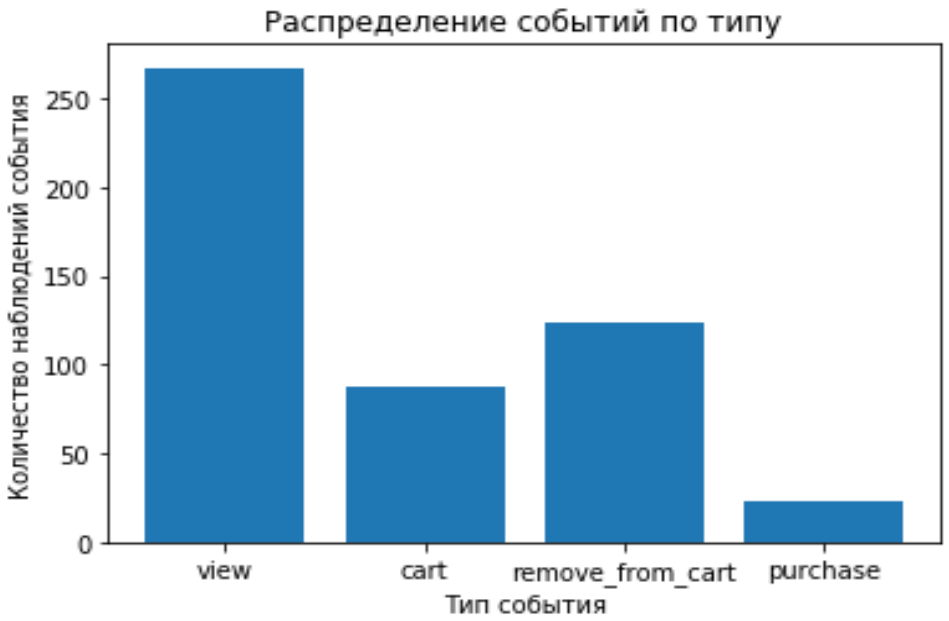
Имеем:

```
In [53]: event_types_counter
Out[53]: Counter({'view': 267, 'cart': 87, 'remove_from_cart': 123, 'purchase': 23})

In [54]: event_types_dict
Out[54]: {'view': 267, 'cart': 87, 'remove_from_cart': 123, 'purchase': 23}
```

2. ВИЗУАЛИЗАЦИЯ ЭМПИРИЧЕСКИХ РАСПРЕДЕЛЕНИЙ: MATPLOTLIB

2. Визуализация эмпирических распределений: matplotlib
2.1. Столбчатая диаграмма

```
28
29 import matplotlib.pyplot as plt
30 # по горизонтальной оси откладываем тип события, по вертикальной - количество наблюдений
31 plt.bar(event_types_counter.keys(), event_types_counter.values())
32 plt.xlabel("Тип события") # название горизонтальной оси
33 plt.ylabel("Количество наблюдений события") # название вертикальной оси
34 plt.title("Распределение событий по типу") # название графика
35 plt.show()
```



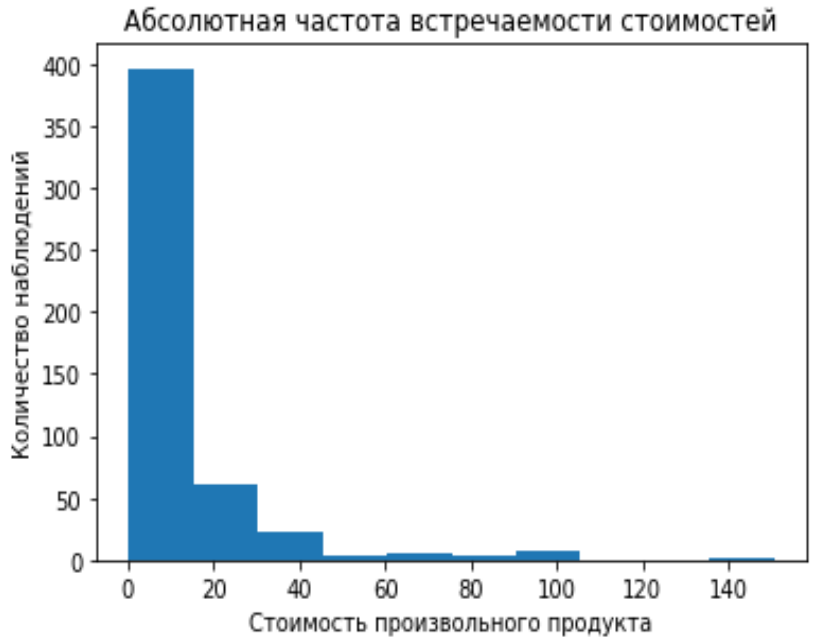
2. Визуализация эмпирических распределений: matplotlib
2.1. Столбчатая диаграмма

```
37 prices.sort() # вариационный ряд
38 min_price = prices[0]
39 max_price = prices[-1]
40 w = max_price - min_price # размах ряда
41
42 prices_stat_series = Counter(prices) # статистический ряд
43
44 x = plt.hist(prices) # вычисление и отрисовка гистограммы
45 plt.xlabel("Стоимость произвольного продукта") # название горизонтальной оси
46 plt.ylabel("Количество наблюдений") # название вертикальной оси
47 plt.title("Абсолютная частота встречаемости стоимостей") # название графика
48 plt.show() # отображение рисунка на экране
49
```

2. Визуализация эмпирических распределений: matplotlib
2.2. Гистограмма для ξ

Функция `hist()`

- Не вероятностная гистограмма;
- Возвращает группированный статистический ряд (информационную совокупность);
- По умолчанию – 10 промежутков равной длины.



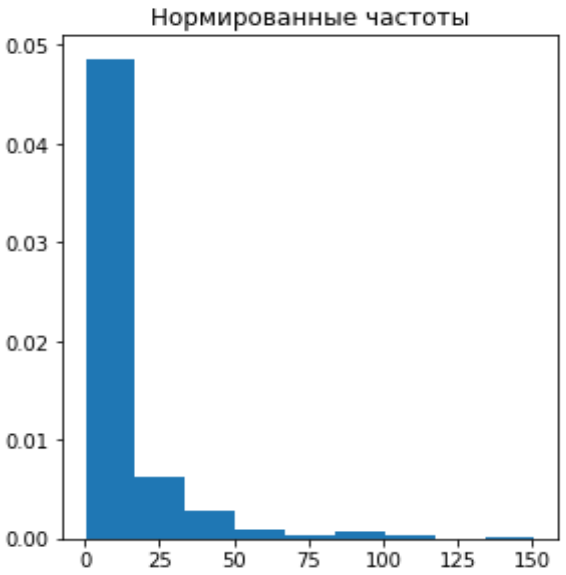

```
In [73]: x
Out[73]:
(array([396.,  60.,  23.,   4.,   5.,   4.,   7.,   0.,   0.,   1.]),
 array([  0.22 ,  15.277,  30.334,  45.391,  60.448,  75.505,  90.562,
        105.619, 120.676, 135.733, 150.79 ]),
 <a list of 10 Patch objects>)
```

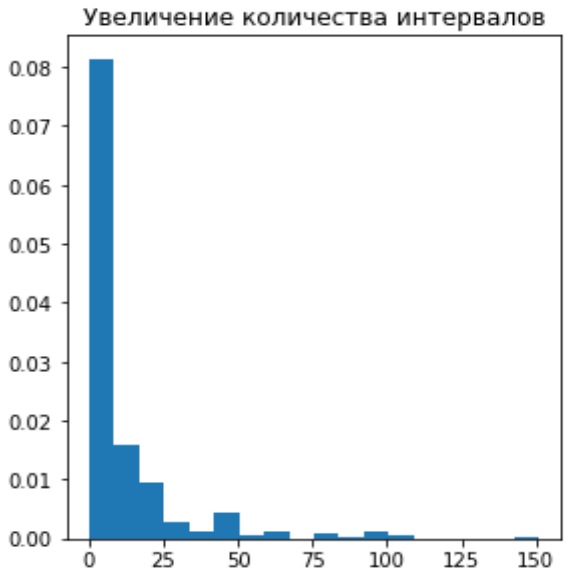
2. Визуализация эмпирических распределений: matplotlib
2.2. Гистограмма для ξ

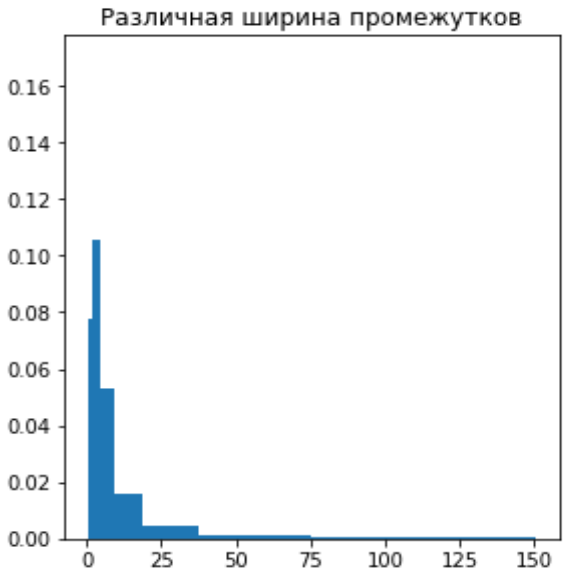
```
49
50 fig, axes = plt.subplots(nrows=2, ncols=3, figsize = (14, 9))
51
52 import math
53 k = int(math.log(sample_size, 2)) + 1 # количество промежутков по форуле Стёрджеса
54 delta = (max_price-min_price) / k # ширина промежутков
55
56 axes[0][0].hist(prices, bins = k, density = True) # количество промежутков по формуле Стёрджеса
57 axes[0][0].set_title("Нормированные частоты") # название графика
58
59 axes[0][1].hist(prices, bins = 2*k, density = True) # удвоение количества промежутков
60 axes[0][1].set_title("Увеличение количества интервалов") # название графика
61
62 # функция расчета границ промежутков увеличивающейся ширины
63 def exp_borders(min, max, n):
64     bord = []
65     bord.append(min)
66     delta = (max - min) / (2 ** n - 1)
67     for i in range(n):
68         bord.append(bord[i] + delta)
69         delta  *= 2
70     return bord
71
72 axes[0][2].hist(prices, bins = exp_borders(min_price, max_price, k-1), density = True) # пользовательские границы промежутков
73 axes[0][2].set_title("Различная ширина промежутков") # название графика
74
```

2. Визуализация эмпирических распределений: `matplotlib`
2.2. Гистограмма для ξ

- Вероятностные гистограммы
- Изменяемые границы разбиения





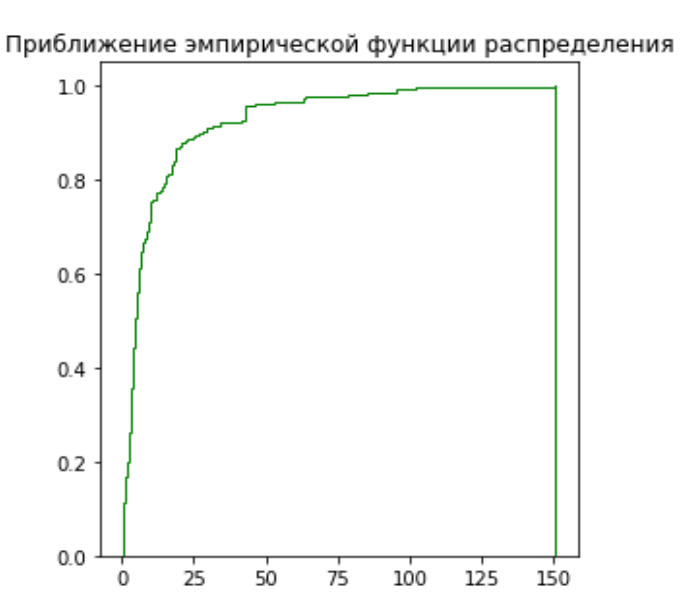


2. Визуализация эмпирических распределений: matplotlib
2.3. Эмпирическая функция распределения для ξ

```
75 axes[1][0].hist(prices, bins = k, color = 'green', cumulative = True) # график с накопленными абсолютными частотами
76 axes[1][0].set_title("Накопленные частоты") # название графика
77
78 axes[1][1].hist(prices, bins = 2*k, density = True, color = 'green', cumulative = True) # нормированный график
79 # но еще не эмпирическая функция распределения
80 axes[1][1].set_title("Накопленные нормированные частоты,\n увеличение количества интервалов") # название графика
81
82 axes[1][2].hist(prices, bins = sample_size, density = True, cumulative = True, color = 'green', histtype='step', fill = False)
83 # почти эмпирическая функция распределения
84 axes[1][2].set_title("Приближение эмпирической функции распределения") # название графика
```

2. Визуализация эмпирических распределений: matplotlib
2.3. Эмпирическая функция распределения для ξ

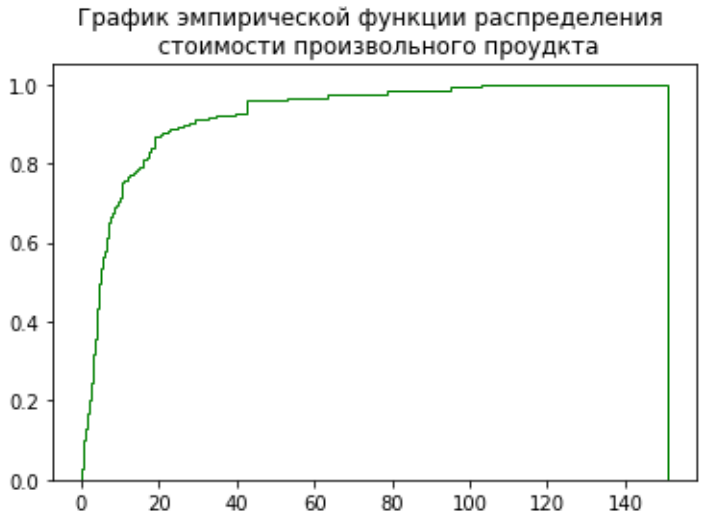
- Накопленные частоты
- Оценка эмпирической функции распределения



2. Визуализация эмпирических распределений: matplotlib
2.3. Эмпирическая функция распределения для ξ

```
88
89 # в точности эмпирическая функция распределения
90 plt.hist(prices, bins = list(prices_stat_series.keys()), density = True, cumulative = True,
91          color = 'green', histtype='step', fill = False)
92 plt.title("График эмпирической функции распределения \n стоимости произвольного проудкта") # название графика
93 plt.show()
94
```

- Пользовательские границы разбиения: различные выборочные значения
- Эмпирическая функция распределения



2. Визуализация эмпирических распределений: matplotlib

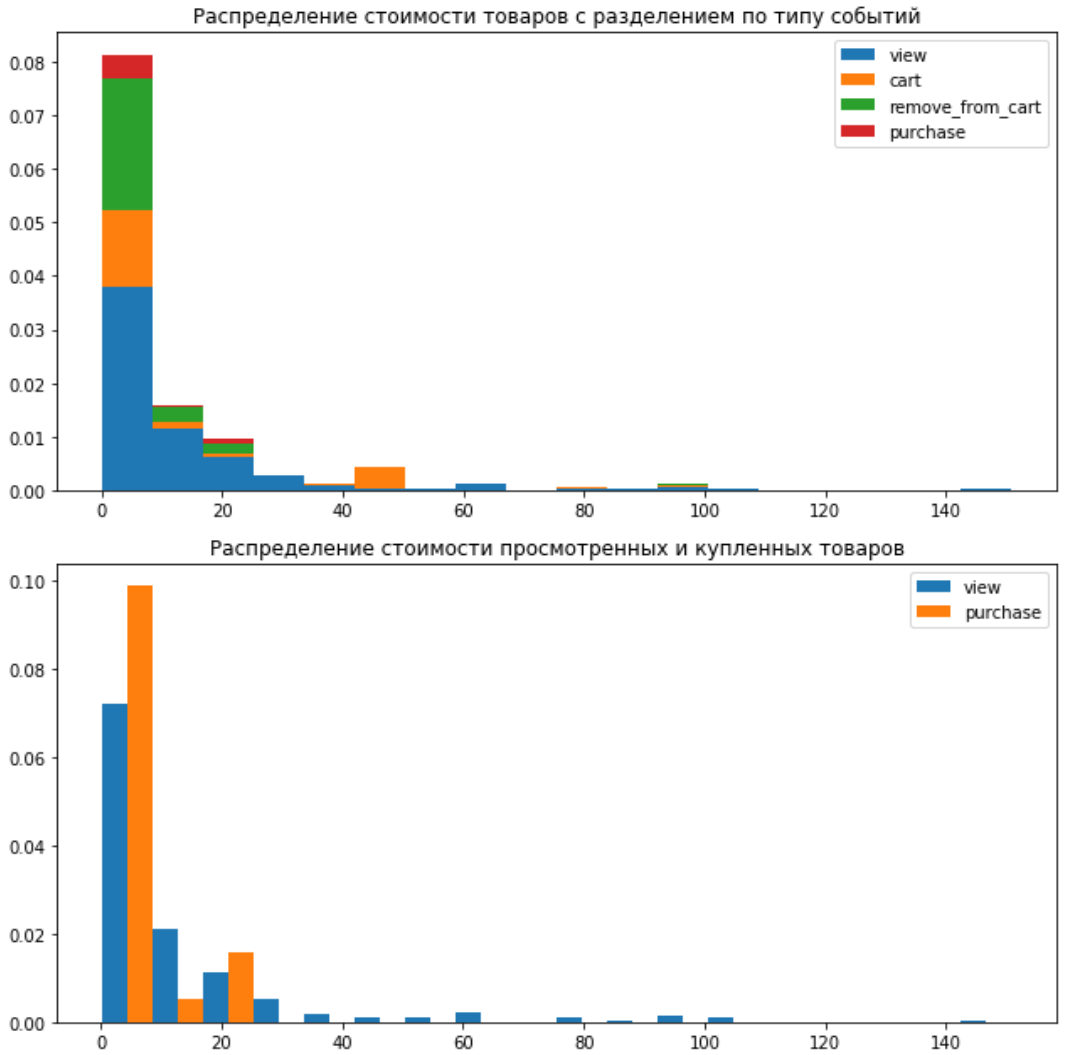
2.4. Распределения для ξ_1, ξ_2, ξ_3, ξ_4

```
fig, axes = plt.subplots(2, figsize = (9, 9))

labels = ['view', 'cart', 'remove_from_cart', 'purchase']

axes[0].hist([prices_on_event['view'], prices_on_event['cart'], prices_on_event['remove_from_cart'], prices_on_event['purchase']],
         bins = 2*k, label = labels, density = True, histtype='bar', stacked=True)
axes[0].legend()
axes[0].set_title("Распределение стоимости товаров с разделением по типу событий")

axes[1].hist([prices_on_event['view'], prices_on_event['purchase']],
         bins = 2*k, label = ['view', 'purchase'], density = True, histtype='bar', rwidth = 1)
axes[1].legend()
axes[1].set_title("Распределение стоимости просмотренных и купленных товаров")

fig.tight_layout()
plt.show()
```

2. Визуализация эмпирических распределений: `matplotlib`

2.4. Распределения для ξ_1, ξ_2, ξ_3, ξ_4

- stacked = True: вклад каждой из величин в общее распределение
- Распределение каждой из величин в отдельности



2. Визуализация эмпирических распределений: matplotlib

2.4. Распределения для ξ_1, ξ_2, ξ_3, ξ_4

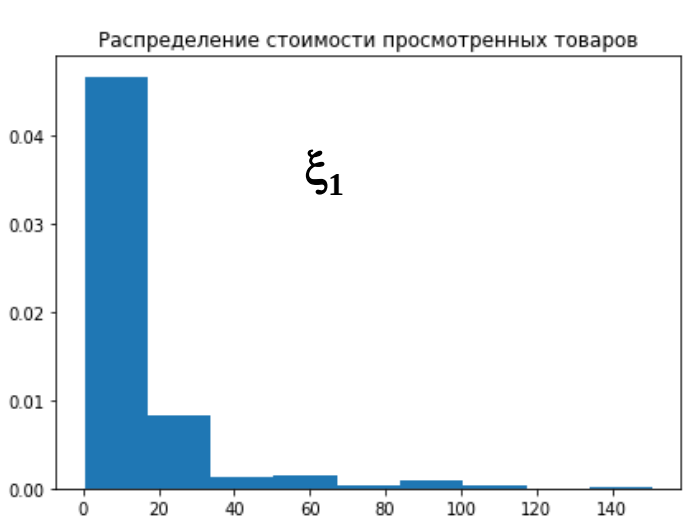
```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize = (12, 9))

axes[0][0].hist(prices_on_event['view'], bins = int(math.log(event_types_counter['view'], 2)) + 1, density = True)
axes[0][0].set_title("Распределение стоимости просмотренных товаров")

axes[0][1].hist(prices_on_event['cart'], bins = int(math.log(event_types_counter['cart'], 2)) + 1, density = True)
axes[0][1].set_title("Распределение стоимости товаров, \n отложенных в корзину")

axes[1][0].hist(prices_on_event['remove_from_cart'], bins = int(math.log(event_types_counter['remove_from_cart'], 2)) + 1,
    density = True)
axes[1][0].set_title("Распределение стоимости товаров, \n удаленных из корзины")

axes[1][1].hist(prices_on_event['purchase'], bins = int(math.log(event_types_counter['purchase'], 2)) + 1, density = True)
axes[1][1].set_title("Распределение стоимости купленных товаров")

fig.tight_layout()
plt.show()
```

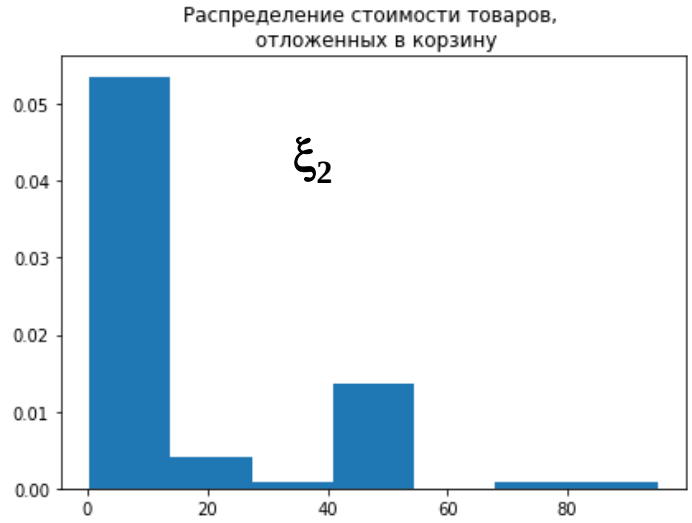
2. Визуализация эмпирических распределений: matplotlib

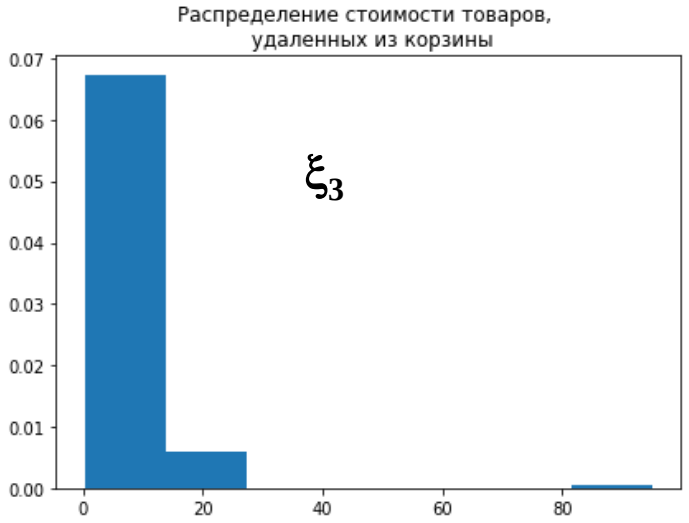
2.4. Распределения для ξ_1, ξ_2, ξ_3, ξ_4

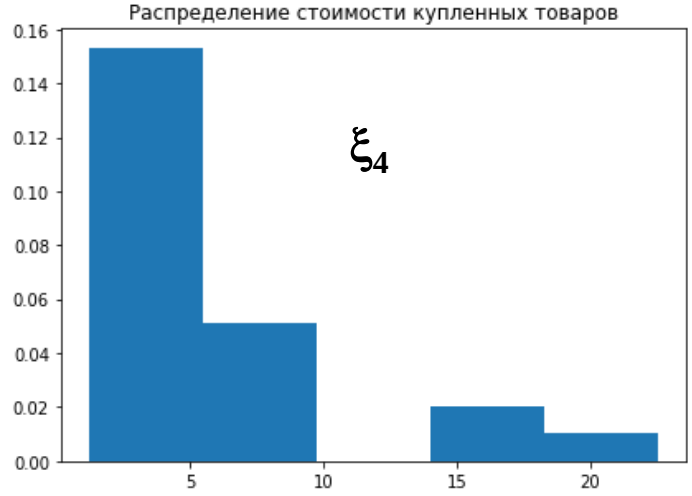
Выводы:

- Покупаются товары невысокой стоимости (≈ до 20у.е.)
- «Пики» интереса на товары стоимостью 40 и 60 у.е.





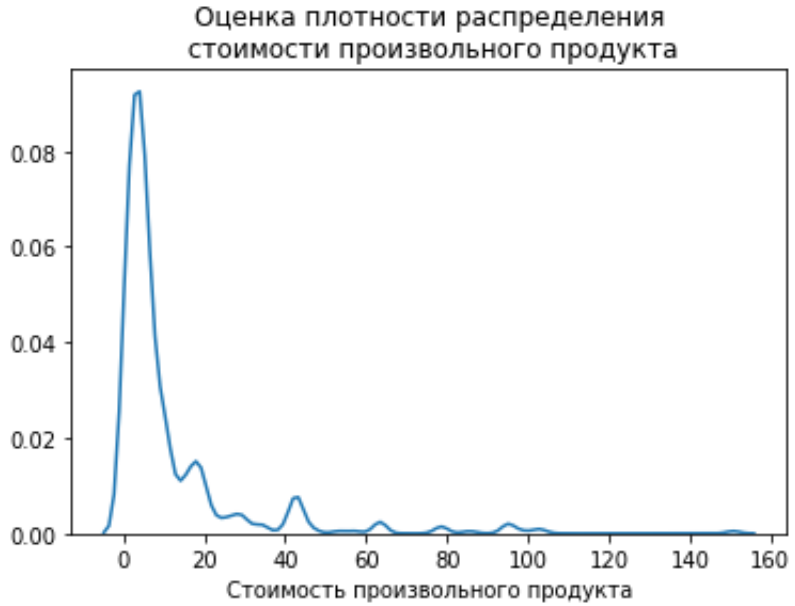




3. ВИЗУАЛИЗАЦИЯ ЭМПИРИЧЕСКИХ РАСПРЕДЕЛЕНИЙ: SEABORN

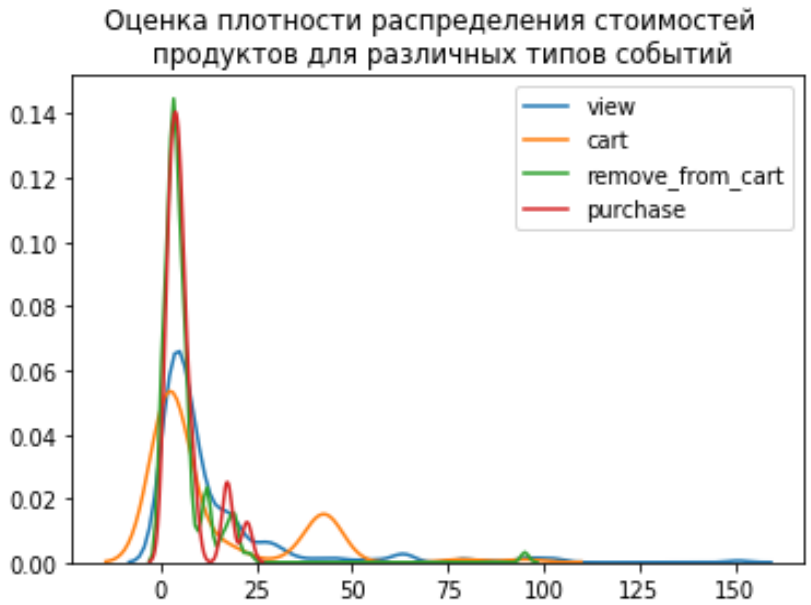
3. Визуализация эмпирических распределений: seaborn
3.1. Оценка плотности распределения

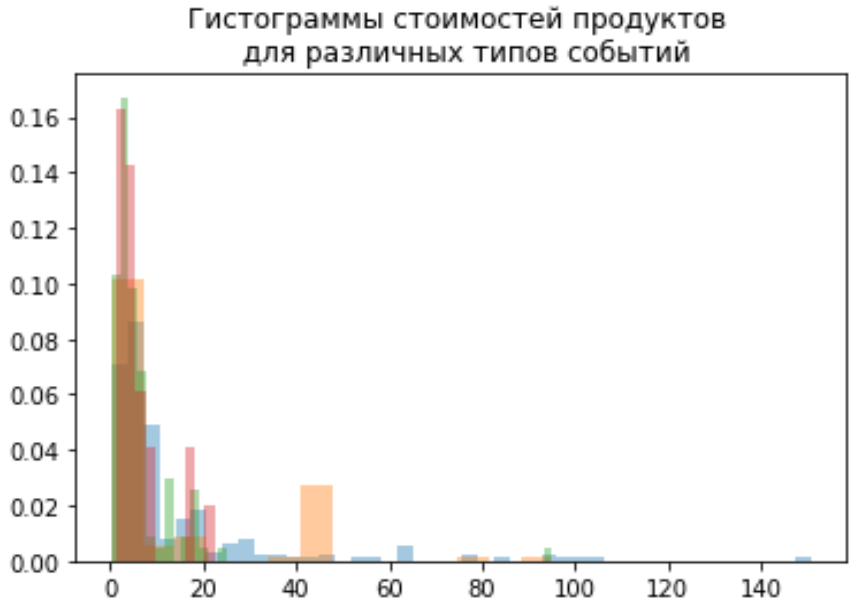
```
132 import seaborn as sns
133 sns.distplot(prices) # построение гистограммы, параметры по умолчанию
134 plt.xlabel("Стоимость произвольного продукта") # название горизонтальной оси
135 plt.title("Распределение стоимости произвольного продукта") # название графика
136 plt.show() # отображение на экране текущего рисунка
137
138 sns.distplot(prices, hist = False, kde = True) # отображение только оценки плотности
139 plt.xlabel("Стоимость произвольного продукта") # название горизонтальной оси
140 plt.title("Оценка плотности распределения\n стоимости произвольного продукта") # название графика
141 plt.show() # отображение рисунка на экране
142
```



3. Визуализация эмпирических распределений: seaborn
3.1. Оценка плотности распределения

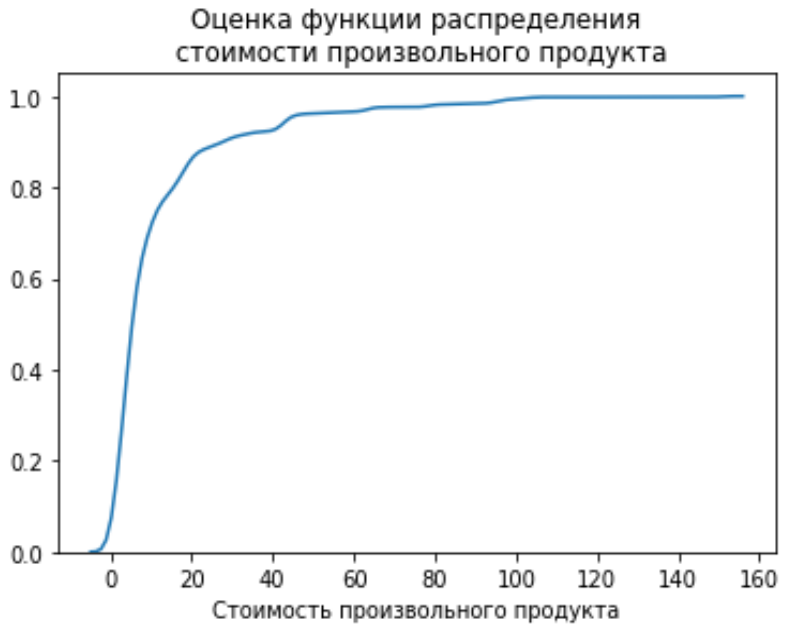
```
143 for type in prices_on_event.keys(): # оценки плотности стоимости продукта по типам событий
144     sns.distplot(prices_on_event[type], hist = False, kde = True, label = type)
145 plt.title("Оценка плотности распределения стоимостей  \n продуктов для различных типов событий") # название графика
146 plt.show()
147
148 for type in prices_on_event.keys(): # гистограммы стоимости продукта по типам событий
149     sns.distplot(prices_on_event[type], hist = True, norm_hist = True, kde = False)
150 plt.title("Гистограммы стоимостей продуктов \n для различных типов событий") # название графика
151
```





3. Визуализация эмпирических распределений: seaborn
3.1. Оценка функции распределения

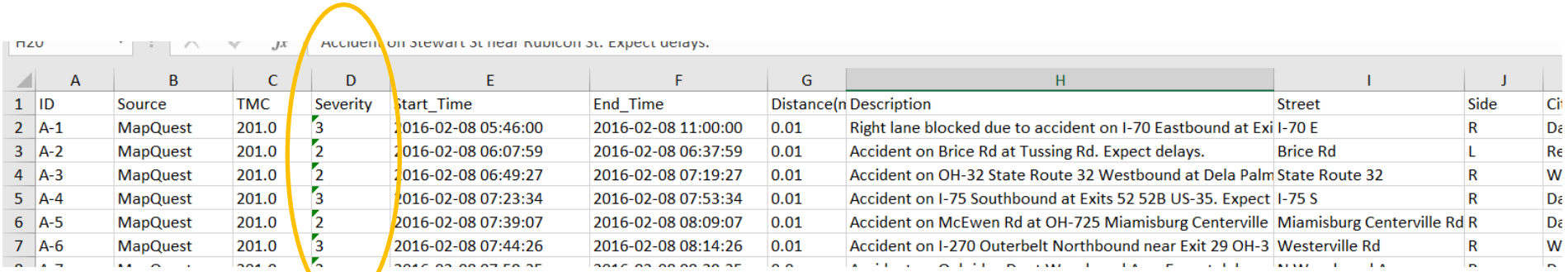
```
154 x = sns.kdeplot(prices, cumulative = True) # отображение только оценки плотности
155 plt.xlabel("Стоимость произвольного продукта") # название горизонтальной оси
156 plt.title("Оценка функции распределения\n стоимости произвольного продукта")
157 plt.show() # отображение рисунка на экране
```



4. ПРАКТИЧЕСКОЕ ЗАДАНИЕ

4. Практическое задание

- К заданию прилагается документ *02_Автоаварии.xls* с данными об автомобильных авариях в США.
  - https://www.kaggle.com/sobhanmoosavi/us-accidents#US_Accidents_Dec19.csv



	A	B	C	D	E	F	G	H	I	J
1	ID	Source	TMC	Severity	Start_Time	End_Time	Distance(n	Description	Street	Side
2	A-1	MapQuest	201.0	3	2016-02-08 05:46:00	2016-02-08 11:00:00	0.01	Right lane blocked due to accident on I-70 Eastbound at Exi	I-70 E	R
3	A-2	MapQuest	201.0	2	2016-02-08 06:07:59	2016-02-08 06:37:59	0.01	Accident on Brice Rd at Tussing Rd. Expect delays.	Brice Rd	L
4	A-3	MapQuest	201.0	2	2016-02-08 06:49:27	2016-02-08 07:19:27	0.01	Accident on OH-32 State Route 32 Westbound at Dela Palm	State Route 32	R
5	A-4	MapQuest	201.0	3	2016-02-08 07:23:34	2016-02-08 07:53:34	0.01	Accident on I-75 Southbound at Exits 52 52B US-35. Expect	I-75 S	R
6	A-5	MapQuest	201.0	2	2016-02-08 07:39:07	2016-02-08 08:09:07	0.01	Accident on McEwen Rd at OH-725 Miamisburg Centerville	Miamisburg Centerville Rd	R
7	A-6	MapQuest	201.0	3	2016-02-08 07:44:26	2016-02-08 08:14:26	0.01	Accident on I-270 Outerbelt Northbound near Exit 29 OH-3	Westerville Rd	R

	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	City	State	Zipcode	Weather_Timestamp	Temperat	Wind_Chi	Humidity(	Pressure(i	Visibility(	Wind_Dire	Wind_Spe	Precipitat	Weather
2	Dayton	OH	45424	2016-02-08 05:58:00	36.9		91.0	29.68	10.0	Calm		0.02	Light Rai
3	Reynoldsb	OH	43068-340	2016-02-08 05:51:00	37.9		100.0	29.65	10.0	Calm		0.0	Light Rai
4	Williamsb	OH	45176	2016-02-08 06:56:00	36.0	33.3	100.0	29.67	10.0	SW	3.5		Overcast
5	Dayton	OH	45417	2016-02-08 07:38:00	35.1	31.0	96.0	29.64	9.0	SW	4.6		Mostly C
6	Dayton	OH	45459	2016-02-08 07:53:00	36.0	33.3	89.0	29.65	6.0	SW	3.5		Mostly C
7	Westervil	OH	43081	2016-02-08 07:51:00	37.9	35.5	97.0	29.63	7.0	SSW	3.5	0.03	Light Rai

4. Практическое задание

1. Определить тип данных для каждого из столбцов файла *02_Автоаварии.xls*.
2. Изучить распределение случайных величин ξ – видимость дороги в момент совершения аварии (**Visibility**) и ξ_1, ξ_2, ξ_3, ξ_4 – видимость дороги в момент совершения аварии степени серьезности 1, 2, 3, 4 (**Severity**). Выбрать инструмент Python для решения задачи (matplotlib.pyplot.hist(), seaborn.distplot() или seaborn.kdeplot()), построить гистограммы и функции распределения величин.
3. Построить вариационный ряд и статистический ряд для величины η – температура воздуха в момент совершения аварии (**Temperature**), построить группированный статистический ряд.
4. Определить 5 городов с наибольшим количеством автомобильных аварий за наблюдаемый период. Построить функции распределения случайных величин ζ_i - протяженность участка дороги, задействованного при аварии (**Distance**), в i-ом городе из данного списка, $i = 1, 2, \ldots, 5$.

4. Практическое задание

Факультативное задание. Для города с максимальным количеством аварий (из пункта 4) составить частотный рейтинг среди следующих факторов, зафиксированных в момент аварии:

1) видимость дороги менее 2 миль (Visibility);

2) скорость ветра более 10 миль в час (Wind_Speed);

3) наличие осадков в виде снега (Light Snow, Snow, etc.);

4) наличие вблизи места аварии перекрестка (Crossing) или транспортной развязки (Junction);

5) наличие вблизи места аварии светофора (Traffic_Signal).

Отразить данный рейтинг графически (например, столбцовой диаграммой с относительной частотой наличия факторов по вертикальной оси).

Литература

1. Источник данных: https://www.kaggle.com/mkechinov/ecommerce-events-history-in-cosmetics-shop, https://rees46.com/ru.

2. Источник данных: https://www.kaggle.com/sobhanmoosavi/us-accidents#US_Accidents_Dec19.csv

3. Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. “A Countrywide Traffic Accident Dataset.”, 2019.

4. Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. "Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights." In proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2019.