ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГОРОДА МОСКВЫ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ ЦЕНТР
ПРОФЕССИОНАЛЬНЫХ КВАЛИФИКАЦИЙ И СОДЕЙСТВИЯ ТРУДОУСТРОЙСТВУ
«ПРОФЕССИОНАЛ»

АНАЛИТИЧЕСКИЙ ОТЧЕТ

на тему

«Анализ данных с использованием Python»
(на примере данных исследуемого продукта)

слушателя Базуткина Василия Валерьевича группы № 213
программы повышения квалификации
«Аналитик данных»

Москва, 2023

Цель исследования:

Необходимо выявить определяющие популярность марки вина закономерности и попытаться выяснить, что можно предложить покупателям вина при выборе вина. Это позволит сделать ставку на потенциально популярный продукт и спланировать например рекламную кампанию для интернет-магазинов, осуществляющих продажи вина.

Выполнение задачи предполагает:
1. Предобработку данных
2. Исследовательский анализ данных
3. Составление портрета пользователя.
4. Исследование статистических показателей.
5. Проверку гипотез.
6. Выводы

Цель этого проекта — выявить, какие признаки больше всего влияют на рейтинг вина. Для анализа используется набор данных из Kaggle, крупнейшего в мире сообщества специалистов по данным и машинному обучению. Набор данных состоит из 13 признаков (2 числовых признака и 11 категориальных признаков).

Столбцы данных

- Страна - страна происхождения вина.
- Описание — описание вкусового профиля вина.
- Обозначение - виноградник, откуда берется виноград для вина.
- Баллы - количество баллов на которое критик журнала Wine Enthusiast оценил вино по шкале от 1 до 100.
- Цена - стоимость одной бутылки вина.
- Провинция — провинция или штат, из которого произведено вино.
- Регион 1 — зона виноделия в провинции или штате (например, долина Напа в Калифорнии).
- Регион 2 — (не обязательно) более конкретный регион в винодельческой области (например, Резерфорд в долине Напа).
- Разновидность — сорт винограда, из которого делают вино (например, Пино Нуар).
- Винодельня — винодельня, производящая вино.

Шаг 1. Открытие файла с данными и изучение общей информации

Шаг 2. Подготовка данных

- Заменить названия столбцов (привести к нижнему регистру).
- Преобразовать данные в нужные типы. Описать, в каких столбцах заменили тип данных и почему.
- Обработать пропуски при необходимости.
- Объяснить, почему заполнили пропуски определённым образом или почему не стали это делать.
- Описать причины, которые могли привести к пропускам.
- Посчитать среднии цены для каждой страны.
- Внести новый столбец "Континенты"

```
country_to_continent = {
'Italy':'Europe',
```

'Portugal':'Europe',
'US':'North America',
'Spain':'Europe',
'France':'Europe',
'Germany':'Europe',
'Argentina':'Latin America',
'Chile':'Latin America',
'Australia': 'Oceania',
'Austria': 'Europe',
'South Africa': 'Africa',
'New Zealand': 'Oceania',
'Israel': 'Asia',
'Hungary':'Europe',
'Greece':'Europe',
'Romania':'Europe',
'Mexico':'Latin America',
'Canada':'North America',
'Turkey': 'Asia',
'Czech Republic': 'Europe',
'Slovenia': 'Europe',
'Luxembourg': 'Europe',
'Croatia': 'Europe',
'Georgia':'Europe',
'Uruguay': 'Latin America',
'England': 'Europe',
'Lebanon': 'Asia',
'Serbia': 'Europe',
'Brazil': 'Latin America',
'Moldova': 'Europe',
'Morocco':'Africa',
'Peru':'Latin America',
'India':'Asia',
'Bulgaria':'Europe',
'Cyprus': 'Europe',
'Armenia':'Asia',
'Switzerland':'Europe',
'Bosnia and Herzegovina':'Europe',
'Ukraine':'Europe',
'Slovakia':'Europe',
'Macedonia':'Europe',
'China':'Asia',
'Egypt':'Africa'
}

Шаг 3. Провести исследовательский анализ данных

- Определить, какие сорта лидируют по рейтингам. Найти популярные сорта по региону.
- Выбрать сорта с наибольшими ценами. Для каждого региона найдите среднюю цену вина.
- Определить, популярные сорта вина в бюджетном сегменте.
- Определить, какие сорта вина лидируют по рейтингам.
- Построить график «ящик с усами» по рейтингам в разбивке по странам, по сортам вина.

- Выявить закономерность влияния на цену цвета и рейтинга. Построить диаграмму рассеяния и посчитать корреляцию.

Шаг 4. Составить портрет потребителя каждого региона

Определить для пользователя каждого континента :

- Самые популярные сорта (топ-5).
- Влияет ли рейтинг на цены по регионам?

Шаг 5. Провести исследование статистических показателей

- Выполнить подсчитать среднего количества, дисперсии и стандартного отклонения для цен на продукт различных регионов. Построить гистограммы. Описать распределения.
- Построить линейную регрессию зависимости между ценой продукта и его рейтингом.

Шаг 6. Проверка гипотез

- H0: Средние пользовательские рейтинги красного и белого вина одинаковые.
- H1:Средние пользовательские рейтинги красного и белого вина разные.
- H0:Средние цены двух популярных сортов вина одинаковые.
- H1:Средние цены двух популярных сортов вина разные.

Задать самостоятельно пороговое значение alpha.

Вывод

1.Предобработка данных

Импортируем необходимые библиотеки

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.cm as cm
import missingno as msno
import scipy.stats as st
# импорт библиотеки warnings
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
#загрузка словаря континенты
country_to_continent = {
'Italy':'Europe',
'Portugal':'Europe',
'US':'North America',
'Spain':'Europe',
'France':'Europe',
'Germany':'Europe',
'Argentina':'Latin America',
'Chile':'Latin America',
'Australia': 'Oceania',
'Austria': 'Europe',
'South Africa': 'Africa',
'New Zealand': 'Oceania',
```

```
'Israel': 'Asia',
'Hungary':'Europe',
'Greece':'Europe',
'Romania':'Europe',
'Mexico':'Latin America',
'Canada':'North America',
'Turkey': 'Asia',
'Czech Republic': 'Europe',
'Slovenia': 'Europe',
'Luxembourg': 'Europe',
'Croatia': 'Europe',
'Georgia':'Europe',
'Uruguay': 'Latin America',
'England': 'Europe',
'Lebanon': 'Asia',
'Serbia': 'Europe',
'Brazil': 'Latin America',
'Moldova': 'Europe',
'Morocco':'Africa',
'Peru':'Latin America',
'India':'Asia',
'Bulgaria':'Europe',
'Cyprus': 'Europe',
'Armenia':'Asia',
'Switzerland':'Europe',
'Bosnia and Herzegovina':'Europe',
'Ukraine':'Europe',
'Slovakia':'Europe',
'Macedonia':'Europe',
'China':'Asia',
'Egypt':'Africa'
}
# загрузка цвета вина по  разновидностивинограда
color = {
    "Chardonnay": "white",
    "Pinot Noir": "red",
    "Cabernet Sauvignon": "red",
    "Red Blend": "red",
    "Bordeaux-style Red Blend": "red",
    "Sauvignon Blanc": "white",
    "Syrah": "red",
    "Riesling": "red",
    "Merlot": "red",
    "Zinfandel": "red",
    "Sangiovese": "red",
    "Malbec": "red",
    "White Blend": "white",
    "Rosé": "other",
    "Tempranillo": "red",
    "Nebbiolo": "red",
    "Portuguese Red": "red",
    "Sparkling Blend": "other",
    "Shiraz": "red",
    "Corvina, Rondinella, Molinara": "red",
    "Rhône-style Red Blend": "red",
    "Barbera": "red",
    "Pinot Gris": "white",
    "Viognier": "white",
    "Bordeaux-style White Blend": "white",
    "Champagne Blend": "other",
    "Port": "red",
    "Grüner Veltliner": "white",
    "Gewürztraminer": "white",
    "Portuguese White": "white",
    "Petite Sirah": "red",
```

```
    "Carmenère": "red"
}
```

Загрузка данных

In [2]:
```
df = pd.read_csv('wine_reviews.csv')
df
```

Out[2]:

	country	description	designation	points	price	province	region_1	region_2	variety
0	US	With a delicate, silky mouthfeel and bright ac...	NaN	86	23.0	California	Central Coast	Central Coast	Pinot Noir
1	Italy	D'Alceo is a drop dead gorgeous wine that ooze...	D'Alceo	96	275.0	Tuscany	Toscana	NaN	Red Blend
2	France	The great dominance of Cabernet Sauvignon in t...	NaN	91	40.0	Bordeaux	Haut-Médoc	NaN	Bordeaux-style Red Blend
3	Italy	The modest cherry, dark berry and black tea no...	NaN	81	15.0	Tuscany	Chianti Classico	NaN	Sangiovese
4	US	Exceedingly light in color, scent and flavor, ...	NaN	83	25.0	Oregon	Rogue Valley	Southern Oregon	Pinot Noir
...	...	...	...	...	...	...	...	...	...
19995	France	Firm wine, with tannins to match the chunky st...	Mansois	88	12.0	Southwest France	Marcillac	NaN	Mansois
19996	US	The vineyard is on the Napa side of Carneros. ...	Estate Vineyard	89	50.0	California	Carneros	Napa-Sonoma	Pinot Noir
19997	Italy	Lighea is a terrific wine and an excellent pai...	Lighea	87	20.0	Sicily & Sardinia	Sicilia	NaN	Zibibbo
19998	Italy	Organically farmed Cannonau grapes deliver sma...	Le Sabbie	87	NaN	Sicily & Sardinia	Cannonau di Sardegna	NaN	Cannonau
19999	US	Grown on the Sonoma side of the appellation, i...	NaN	92	35.0	California	Carneros	Napa-Sonoma	Syrah

20000 rows × 10 columns

In [3]:
```
#На основе словаря `country_to_continent` создайте переменную,принадлежности страны к

df['country_to_continent']=df['country'].map(country_to_continent)
```

In [4]:
```
# Для значений `country_to_continent`не вошедших в словарь даем значение переменной "
df['country_to_continent'].fillna("Unknown",inplace=True)
```

In [5]:
```
#На основе словаря `color` создайте переменную,
#в которой закодирован цвет вина
df['color']=df['variety'].map(color)
```

In [6]:
```
# Для значений `color`не вошедших в словарь даем значение переменной "other1"
df['color'].fillna("other1",inplace=True)
```

In [7]:
```
#Определяем структуру данных
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   country               20000 non-null  object
 1   description           20000 non-null  object
 2   designation           13999 non-null  object
 3   points                20000 non-null  int64
 4   price                 18198 non-null  float64
 5   province              20000 non-null  object
 6   region_1              16543 non-null  object
 7   region_2              8058 non-null   object
 8   variety               20000 non-null  object
 9   winery                20000 non-null  object
 10  country_to_continent  20000 non-null  object
 11  color                 20000 non-null  object
dtypes: float64(1), int64(1), object(10)
memory usage: 1.8+ MB
```

Количество значений в столбцах различается. Это говорит о том, что в данных есть пустые значения. Признак points и price числовые. С помощью библиотеки missingno визуализируем пустые значения.С помощью библиотеки seaborn построим тепловую карту для визуализации данных и подтверждения наличия пустых значений.

In [8]:
```
sns.set()# выполняет настройку стиля графиков по умолчанию
```

In [9]:
```
msno.bar(df)
```

Out[9]: <Axes: >

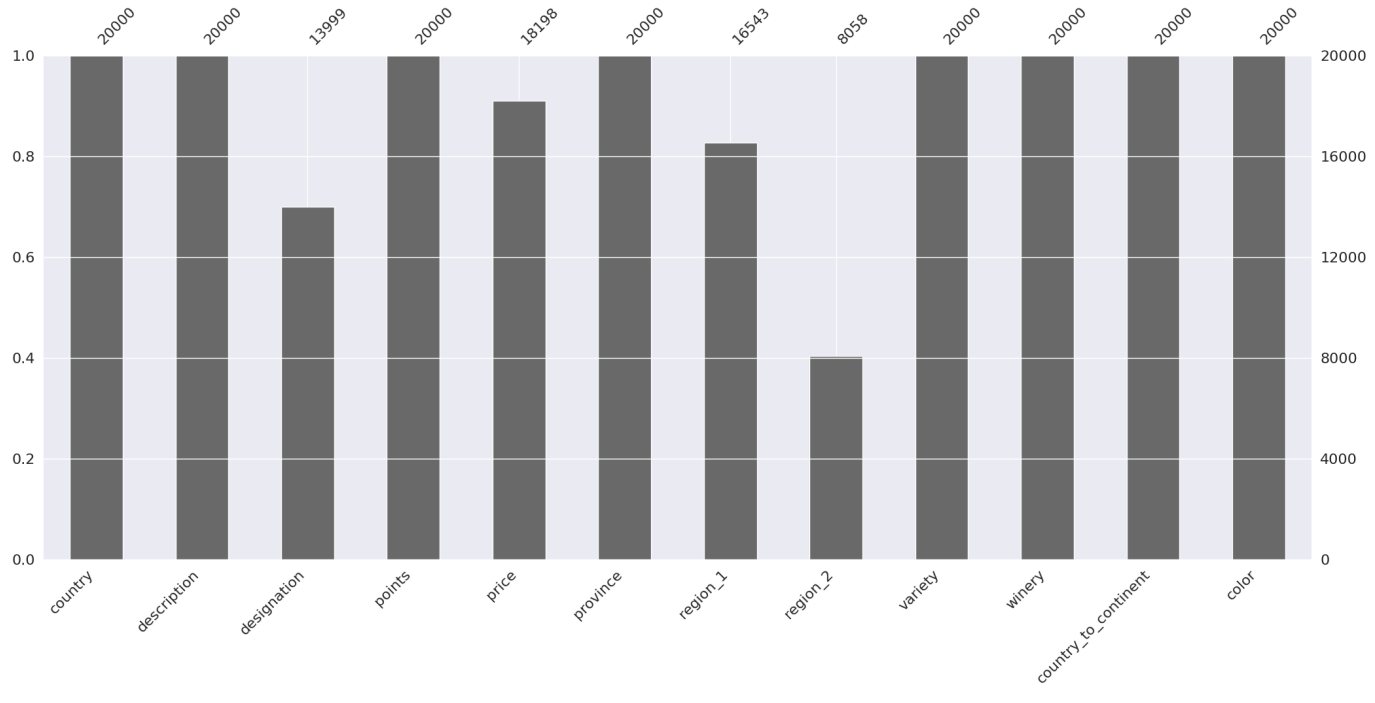


Рисунок1.Столбчетая диаграмма пропущенных значени набора данных.

In [10]:
```
#Тепловая карта
colours = ['#08e8de', '#FF0000']
sns.heatmap(df.isnull(), cmap=sns.color_palette(colours))
# Decorations
plt.title('Матрица пропущенных значений набора данных', fontsize=14)
plt.tick_params(axis='x', labelrotation=60)
plt.xticks(fontsize=7)
plt.yticks(fontsize=7)
plt.figtext(0.1, -0.2, "  Рисунок2.  Матрица пропущенных значений набора данных",font
plt.show()
```

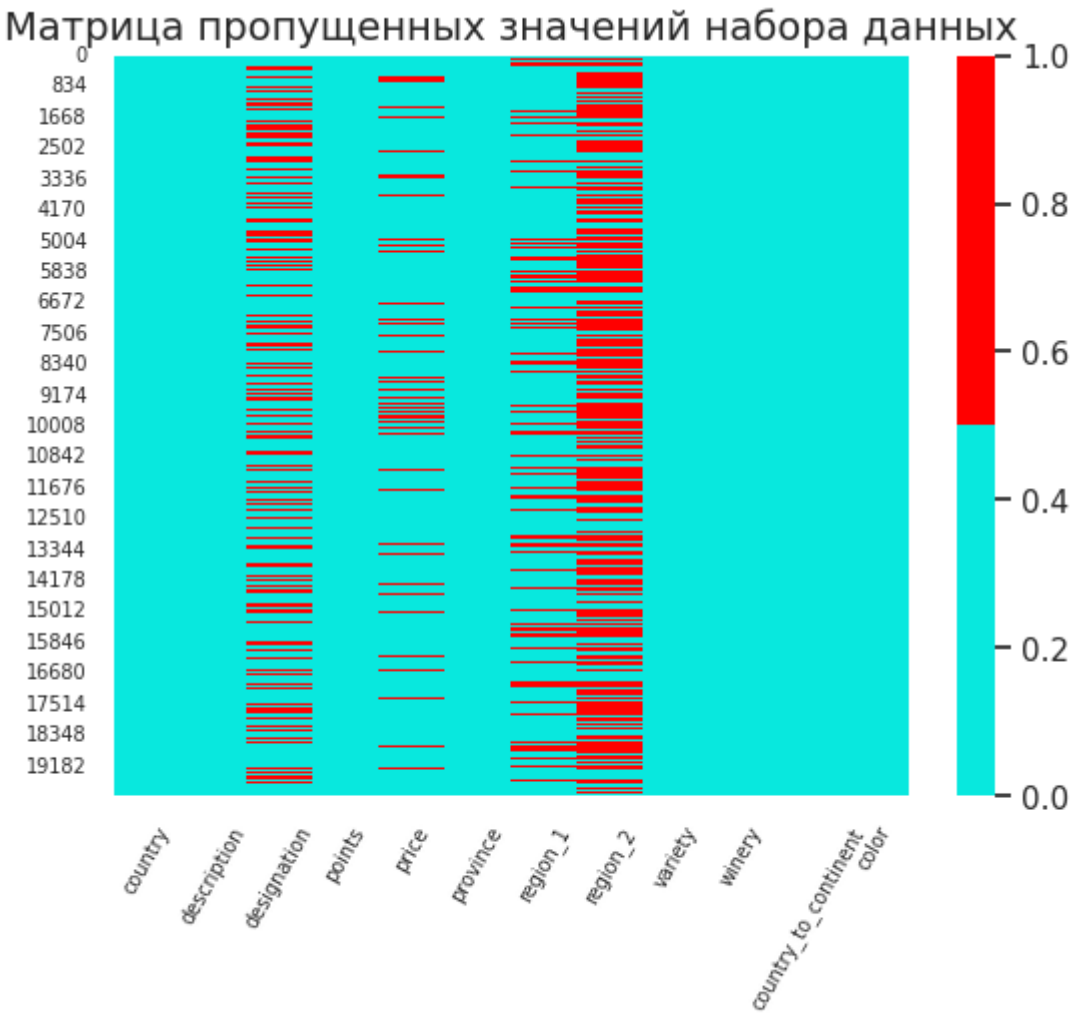


Рисунок2. Матрица пропущенных значений набора данных

In [11]:
```
# Процентное и числовое значение пропущенных данных
MissingValue = df.isnull().sum()
Percent = (df.isnull().sum()/df.isnull().count()*100)
MissingData = pd.concat([MissingValue, Percent], axis=1, keys=['Пропущенные значения'
MissingData
```

Out[11]:

	Пропущенные значения	Процент
country	0	0.000
description	0	0.000
designation	6001	30.005
points	0	0.000
price	1802	9.010
province	0	0.000
region_1	3457	17.285
region_2	11942	59.710
variety	0	0.000
winery	0	0.000
country_to_continent	0	0.000
color	0	0.000

In [12]:
```
df.dropna(axis='index',subset=['price'],inplace=True)# убираем пустые строки без цены
df.drop(columns=['region_2'],inplace=True)# убираем столббец регион2 -60%
#заполняем designation и region_1 значениями "other_design","other_region_1"
df['designation'].fillna("other_design",inplace=True)
df['region_1'].fillna("other_region_1",inplace=True)
# убираем дубликаты
df.drop_duplicates(inplace=True)
msno.bar(df)# Столбчетая диаграмма пропущенных значени набора данных.
```

Out[12]: <Axes: >

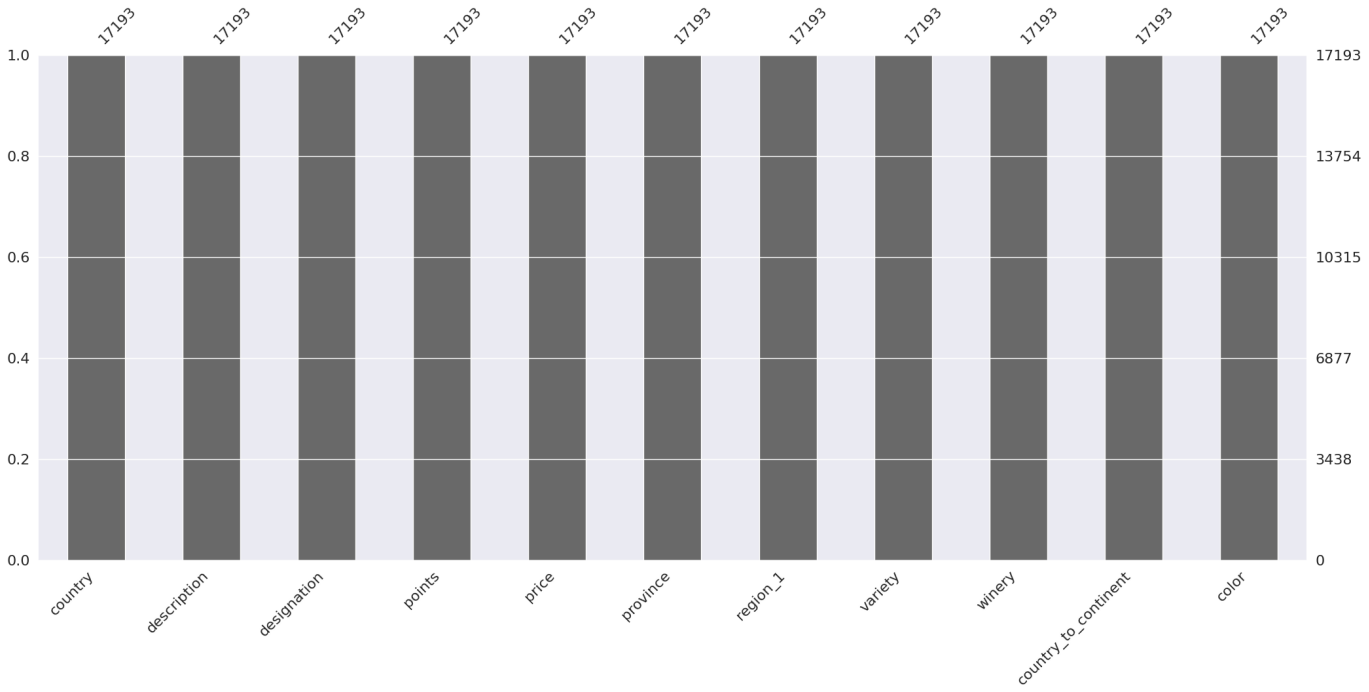


Рисунок3.Столбчетая диаграмма пропущенных значений набора данных.

In [13]:
```
#Делаем перевод названий в новой скопированной таблице
df_rus=df.copy()
df_rus=df_rus.rename(columns=
                     {'country': 'страна',
                      'description':'описание',
                      'designation':'обозначение',
                      'points':'баллы',
```

```
                      'price':'цена',
                      'province':'провинция',
                      'region_1':'регион_1',
                      'variety':'разновидность',
                      'winery':'винодельня',
                      'country_to_continent':'континент',
                      'color':'цвет',
                     })  # заменить имя столбцов
df_rus   #выводим новые названия столбцоы и таблицу
```

Out[13]:

	страна	описание	обозначение	баллы	цена	провинция	регион_1	разновидность
0	US	With a delicate, silky mouthfeel and bright ac...	other_design	86	23.0	California	Central Coast	Pinot Noir
1	Italy	D'Alceo is a drop dead gorgeous wine that ooze...	D'Alceo	96	275.0	Tuscany	Toscana	Red Blend
2	France	The great dominance of Cabernet Sauvignon in t...	other_design	91	40.0	Bordeaux	Haut-Médoc	Bordeaux-style Red Blend
3	Italy	The modest cherry, dark berry and black tea no...	other_design	81	15.0	Tuscany	Chianti Classico	Sangiovese
4	US	Exceedingly light in color, scent and flavor, ...	other_design	83	25.0	Oregon	Rogue Valley	Pinot Noir
...	...	...	...	...	...	...	...	...
19994	US	A little too strong in feline spray character,...	Grand Reserve	84	20.0	California	Mendocino County	Sauvignon Blanc
19995	France	Firm wine, with tannins to match the chunky st...	Mansois	88	12.0	Southwest France	Marcillac	Mansois
19996	US	The vineyard is on the Napa side of Carneros. ...	Estate Vineyard	89	50.0	California	Carneros	Pinot Noir
19997	Italy	Lighea is a terrific wine and an excellent pai...	Lighea	87	20.0	Sicily & Sardinia	Sicilia	Zibibbo
19999	US	Grown on the Sonoma side of the appellation, i...	other_design	92	35.0	California	Carneros	Syrah

17193 rows × 11 columns

In [14]:
```
# создаем два списка переменных меняя их на тип(float) для пр
sampleX = df['price'].astype(float).to_numpy()
sampleY = df['points'].astype(float).to_numpy()
```

In [15]:
```
 #посчитаем средние значения цены для каждой страны
PriceCountry=df.groupby(['country'])['price']
```

```
PriceCountry.describe()
```

Out[15]:

country	count	mean	std	min	25%	50%	75%	max
Argentina	680.0	22.847059	24.016872	6.0	11.00	15.0	22.00	215.0
Australia	592.0	30.925676	35.121561	5.0	15.00	20.0	35.00	550.0
Austria	324.0	32.055556	61.719254	8.0	17.00	24.0	36.00	1100.0
Bosnia and Herzegovina	1.0	12.000000	NaN	12.0	12.00	12.0	12.00	12.0
Brazil	3.0	24.666667	11.060440	13.0	19.50	26.0	30.50	35.0
Bulgaria	9.0	10.888889	4.935698	8.0	8.00	8.0	10.00	20.0
Canada	23.0	47.086957	34.889322	13.0	20.50	32.0	70.00	145.0
Chile	740.0	19.706757	19.119854	6.0	11.00	14.0	20.00	260.0
China	1.0	27.000000	NaN	27.0	27.00	27.0	27.00	27.0
Croatia	19.0	21.684211	11.518356	13.0	18.00	19.0	20.50	65.0
Cyprus	4.0	16.000000	3.366502	11.0	15.50	17.5	18.00	18.0
France	1829.0	46.511208	85.322285	6.0	17.00	26.0	50.00	2300.0
Georgia	6.0	16.000000	5.329165	9.0	14.00	15.0	17.50	25.0
Germany	331.0	36.075529	40.908587	8.0	18.00	25.0	40.00	395.0
Greece	107.0	21.037383	10.094763	7.0	14.00	18.0	26.50	65.0
Hungary	33.0	62.151515	129.513542	7.0	17.00	30.0	55.00	764.0
Israel	79.0	31.164557	17.077824	8.0	15.50	25.0	41.00	85.0
Italy	2335.0	37.159743	38.278069	6.0	17.00	25.0	45.00	495.0
Lebanon	4.0	32.500000	15.524175	12.0	25.50	35.0	42.00	48.0
Luxembourg	1.0	36.000000	NaN	36.0	36.00	36.0	36.00	36.0
Mexico	13.0	25.461538	7.933215	15.0	19.00	25.0	28.00	40.0
Moldova	14.0	14.714286	9.824840	8.0	9.00	11.0	13.00	42.0
Montenegro	1.0	10.000000	NaN	10.0	10.00	10.0	10.00	10.0
New Zealand	384.0	24.549479	14.763148	8.0	16.00	20.0	27.00	125.0
Portugal	558.0	27.921147	33.376868	5.0	11.00	17.0	32.00	426.0
Romania	18.0	12.500000	5.669838	7.0	9.00	10.5	14.75	30.0
Serbia	2.0	16.500000	2.121320	15.0	15.75	16.5	17.25	18.0
Slovenia	12.0	24.416667	9.876127	7.0	19.50	21.5	35.00	40.0
South Africa	282.0	21.411348	13.532389	6.0	12.00	17.0	26.00	96.0
South Korea	1.0	11.000000	NaN	11.0	11.00	11.0	11.00	11.0
Spain	1019.0	28.224730	40.395874	5.0	12.00	17.0	28.00	599.0
Switzerland	1.0	19.000000	NaN	19.0	19.00	19.0	19.00	19.0
Turkey	10.0	29.800000	31.884514	15.0	17.00	20.5	23.00	120.0
US	7748.0	33.575891	23.367191	6.0	19.00	28.0	42.00	300.0
US-France	1.0	50.000000	NaN	50.0	50.00	50.0	50.00	50.0
Ukraine	1.0	13.000000	NaN	13.0	13.00	13.0	13.00	13.0
Uruguay	7.0	18.428571	6.852181	10.0	14.00	17.0	22.00	30.0

2. Исследовательский анализ данных

In [16]:
```
#Сорта лидирующее по рейтенгу(топ 10)
df.sort_values(by='points',ascending=False).head(10)
```

Out[16]:

	country	description	designation	points	price	province	region_1	variety
323	France	A wine that has created its own universe. It h...	Clos du Mesnil	100	1400.0	Champagne	Champagne	Chardonnay
17967	US	Impossibly aromatic. Hard to imagine greater c...	Red Wine	100	245.0	California	Rutherford	Cabernet Blend
5955	Italy	A perfect wine from a classic vintage, the 200...	Masseto	100	460.0	Tuscany	Toscana	Merlot
13188	France	A big, powerful wine that sums up the richness...	other_design	99	2300.0	Bordeaux	Pauillac	Bordeaux-style Red Blend
9203	Italy	Elegant and complex, this gorgeous wine is all...	Monprivato	99	175.0	Piedmont	Barolo	Nebbiolo
9990	Portugal	This is the latest release of Portugal's most ...	Barca Velha	99	426.0	Douro	other_region_1	Portuguese Red
7306	US	The only one of the Cayuse Syrahs that is co-f...	Cailloux Vineyard	99	65.0	Oregon	Walla Walla Valley (OR)	Syrah
19147	France	From arguably the finest white wine vineyard i...	other_design	98	757.0	Burgundy	Montrachet	Chardonnay
6363	Italy	Immensely inviting, this opens with fragrant p...	other_design	98	95.0	Tuscany	Brunello di Montalcino	Sangiovese
5447	US	This is the best of the winery's new releases,...	Litton Estate Vineyard	98	100.0	California	Russian River Valley	Pinot Noir

In [17]:
```
#Сорта лидирующее по региону
popular_variety = df.groupby('country_to_continent')['variety'].apply(lambda x: x.val
print("Самые популярные сорта вин по регионам:")
print(popular_variety)
```

```
Самые популярные сорта вин по регионам:
country_to_continent
Africa                Sauvignon Blanc
Asia               Cabernet Sauvignon
Europe                      Red Blend
Latin America                  Malbec
North America              Pinot Noir
Oceania                        Shiraz
Unknown                        Vranec
Name: variety, dtype: object
```

In [18]:
```
#Сорта лидирующее по ценам(топ 5)
df.sort_values(by='price',ascending=False).head()
```

Out[18]:

	country	description	designation	points	price	province	region_1	variety
13188	France	A big, powerful wine that sums up the richness...	other_design	99	2300.0	Bordeaux	Pauillac	Bordeaux-style Red Blend
323	France	A wine that has created its own universe. It h...	Clos du Mesnil	100	1400.0	Champagne	Champagne	Chardonnay
4324	Austria	Wet earth, rain-wet stones, damp moss, wild sa...	Ried Loibenberg Smaragd	94	1100.0	Wachau	other_region_1	Grüner Veltliner
19501	France	While there is certainly plenty of wood here, ...	other_design	95	850.0	Bordeaux	Saint-Émilion	Bordeaux-style Red Blend
8493	Hungary	Surprisingly subtle, yet maddeningly complex, ...	Essencia	94	764.0	Tokaji	other_region_1	Furmint

In [19]:
```
#Средние цены по региону
df.groupby('country_to_continent')['price'].mean()
```

Out[19]:
```
country_to_continent
Africa           21.411348
Asia             31.031915
Europe           36.883152
Latin America    21.242550
North America    33.615880
Oceania          28.417008
Unknown          23.666667
Name: price, dtype: float64
```

In [20]:
```
#Популярные вина в бюджетном сегменте до 20$
filtered_df = df.loc[df['price'] <= 20]  # Фильтрация строк по условию "price"<20
popular_varieties = filtered_df['variety'].value_counts().sort_values(ascending=False
popular_varieties.head(10)
```

Out[20]:
```
variety
Chardonnay            743
Sauvignon Blanc       547
Cabernet Sauvignon    492
Red Blend             405
Riesling              332
Merlot                301
Pinot Noir            297
Rosé                  282
Malbec                223
White Blend           210
Name: count, dtype: int64
```

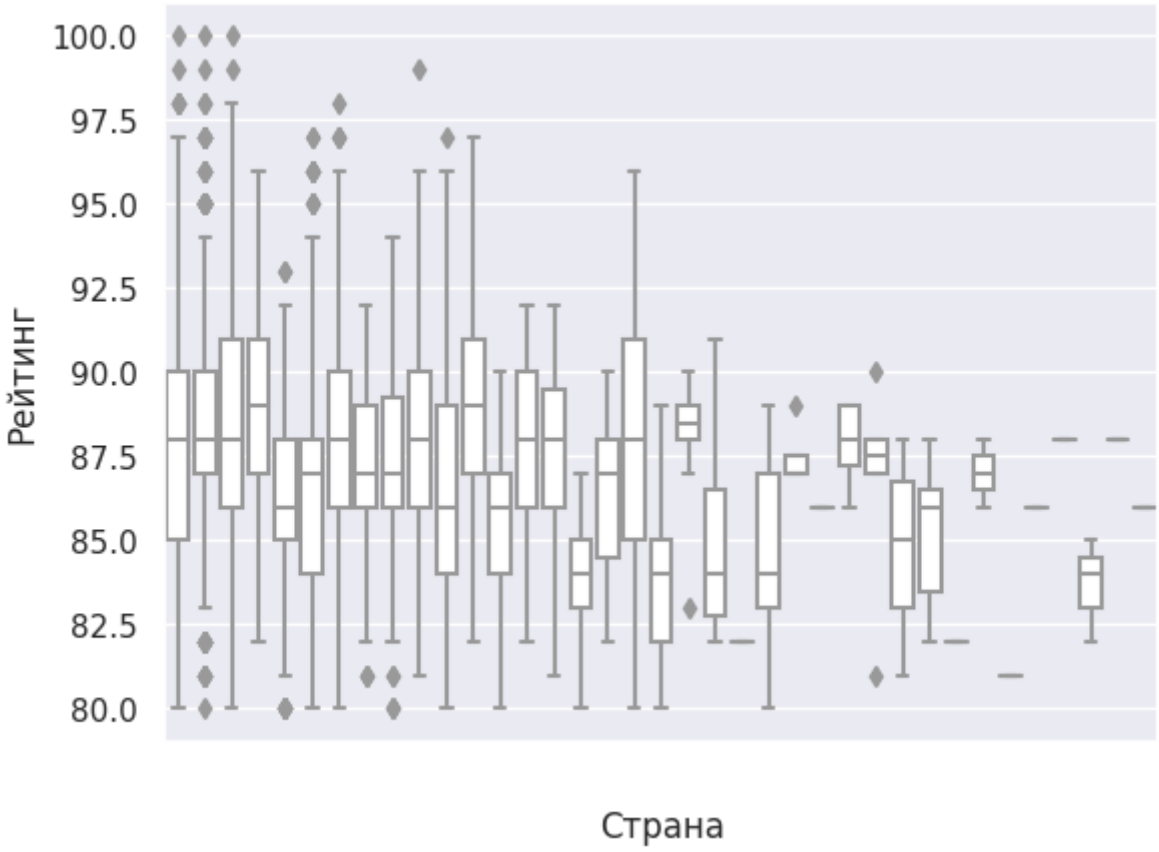
In [21]:
```
#Сорта лидирующее по рейтенгу
df.sort_values(by='points',ascending=False).head(3)
```

Out[21]:

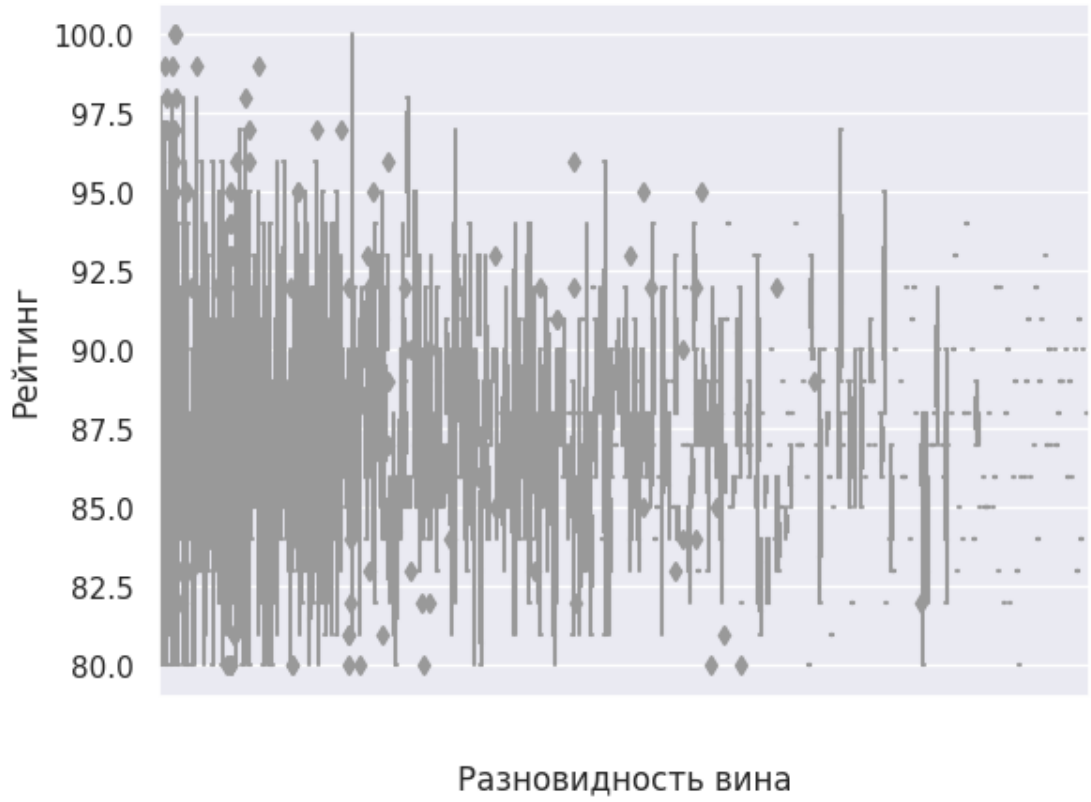
	country	description	designation	points	price	province	region_1	variety	
323	France	A wine that has created its own universe. It h...	Clos du Mesnil	100	1400.0	Champagne	Champagne	Chardonnay	
17967	US	Impossibly aromatic. Hard to imagine greater c...	Red Wine	100	245.0	California	Rutherford	Cabernet Blend	
5955	Italy	A perfect wine from a classic vintage, the 200...	Masseto	100	460.0	Tuscany	Toscana	Merlot	dell'C

In [22]:
```
# Построение ящика с усами в разбивке по странам
ax=sns.boxplot(x='country',y='points',data=df,color='white')
ax.set_xticklabels(ax.get_xticklabels(), color='white')
plt.xlabel("Страна")
plt.ylabel("Рейтинг")
plt.show
```

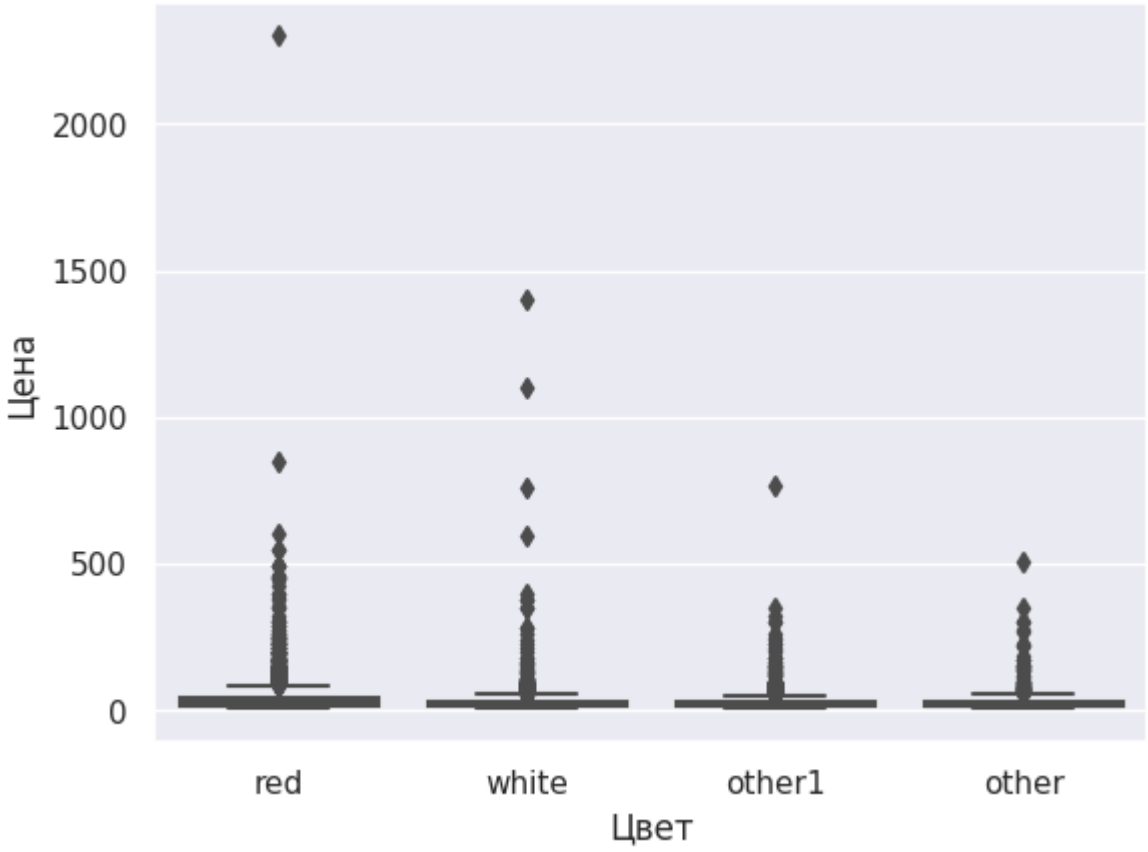
Out[22]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [23]:
```
# Построение ящика с усами в разбивке по сортам вина
bx=sns.boxplot(x='variety',y='points',data=df,color='white')
bx.set_xticklabels(bx.get_xticklabels(), color='white')
plt.xlabel("Разновидность вина")
plt.ylabel("Рейтинг")
plt.show()
```

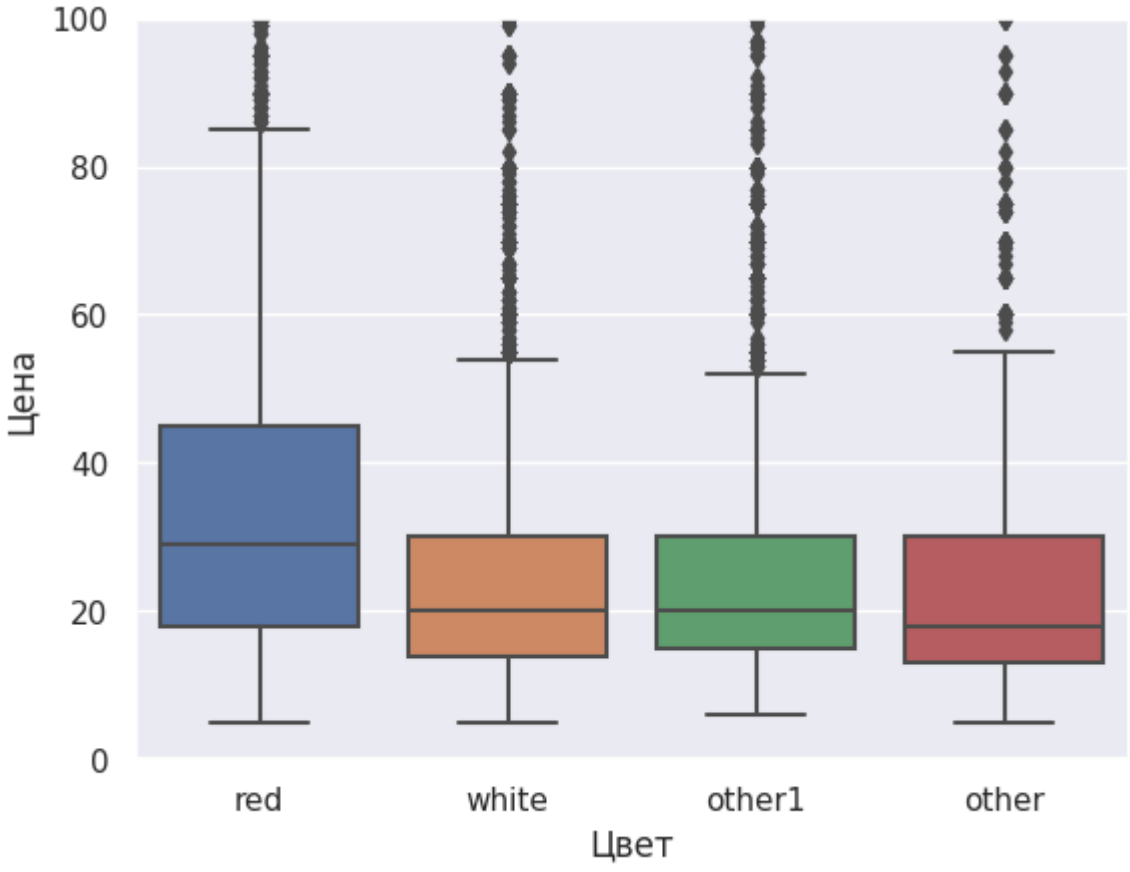


In [24]:
```
# Влияние на цену(с выбросами) цвета
sns.boxplot(x='color',y='price',data=df,)
plt.xlabel("Цвет")
plt.ylabel("Цена")
plt.show()
```



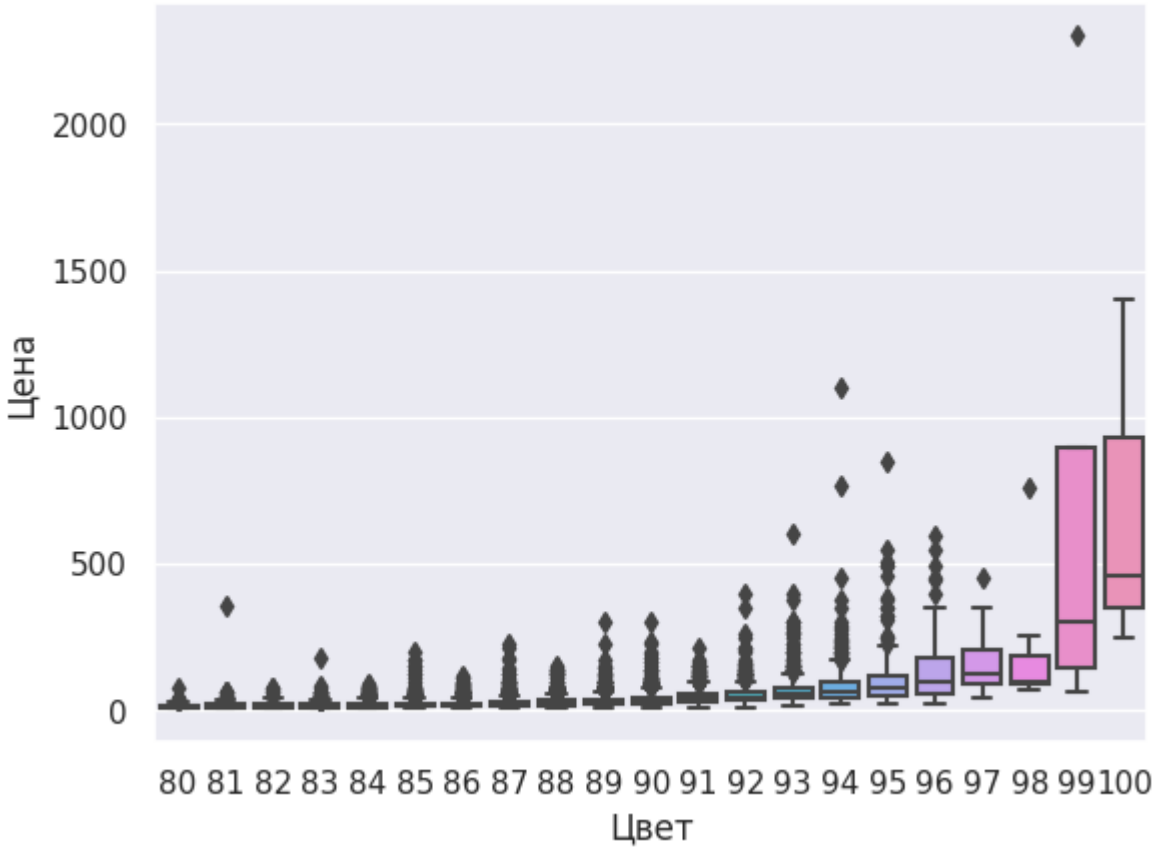
In [25]:
```
# Влияние на цену(диапозон до 100$) цвета
sns.boxplot(x='color',y='price',data=df,)
plt.xlabel("Цвет")
plt.ylabel("Цена")
plt.ylim(0,100)
plt.show
```

Out[25]: <function matplotlib.pyplot.show(close=None, block=None)>

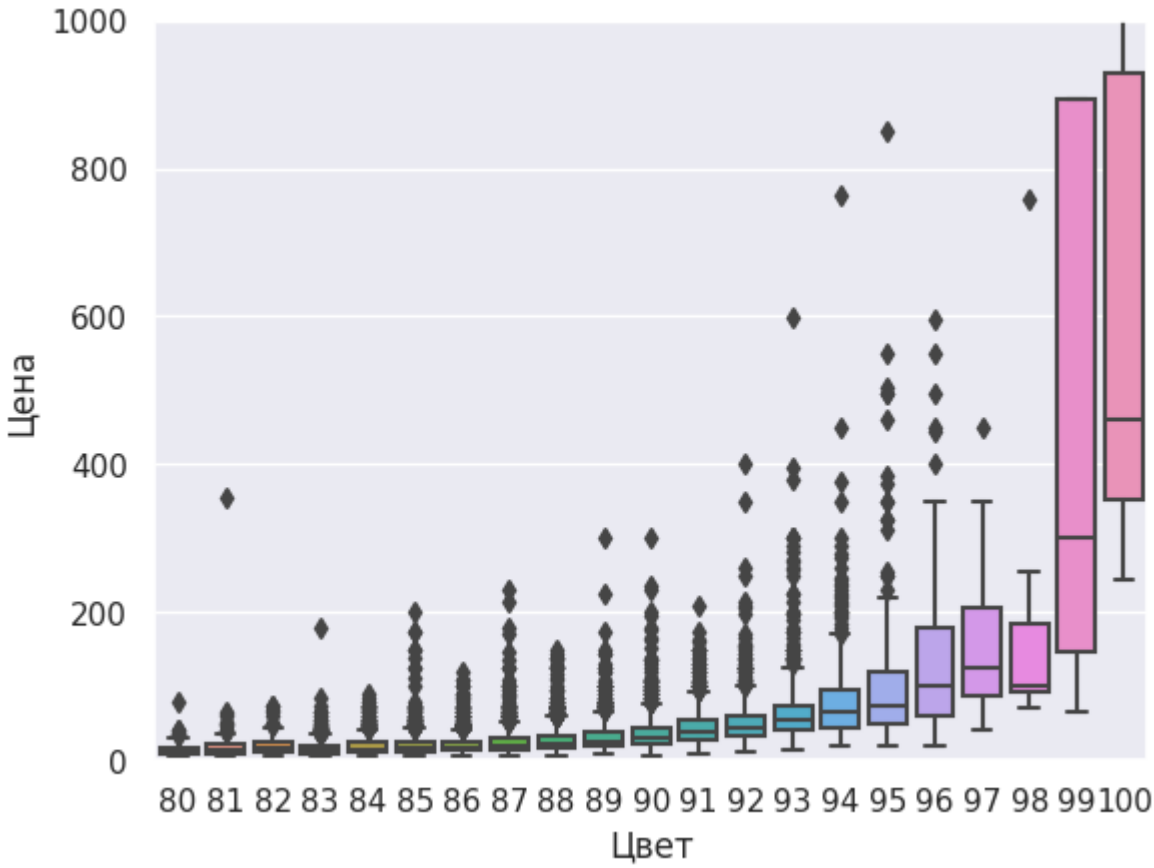


In [26]:
```
# Влияние на цену(с выбросами) рейтинга
sns.boxplot(x='points',y='price',data=df,)
plt.xlabel("Цвет")
```

```
plt.ylabel("Цена")
plt.show()
```



In [27]:
```
#Влияние на цену(диапозон до 1000$) рейтинга
sns.boxplot(x='points',y='price',data=df,)
plt.ylim(0,1000)
plt.xlabel("Цвет")
plt.ylabel("Цена")
plt.show()
plt.show
```



Out[27]: <function matplotlib.pyplot.show(close=None, block=None)>

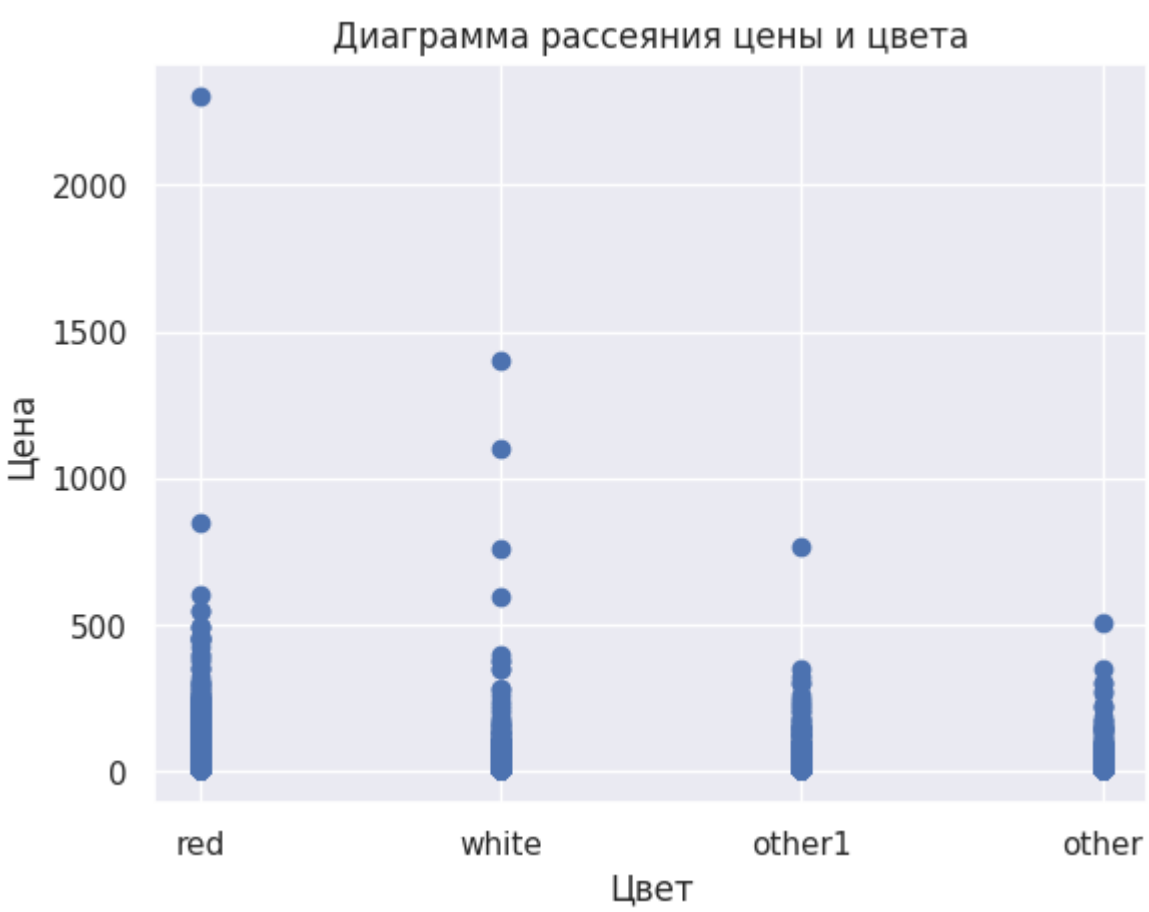
In [28]:
```
plt.scatter(x = df['points'], y = df['price']) #  рейтинг и цена
plt.ylabel("Цена")
plt.xlabel("Рейтинг")
plt.title('Диаграмма рассеяния рейтинга и цены')
plt.show()
```

In [29]:
```
df[['price','points']].corr()# корреляция рейтинга и цены показывает их зависимость
```

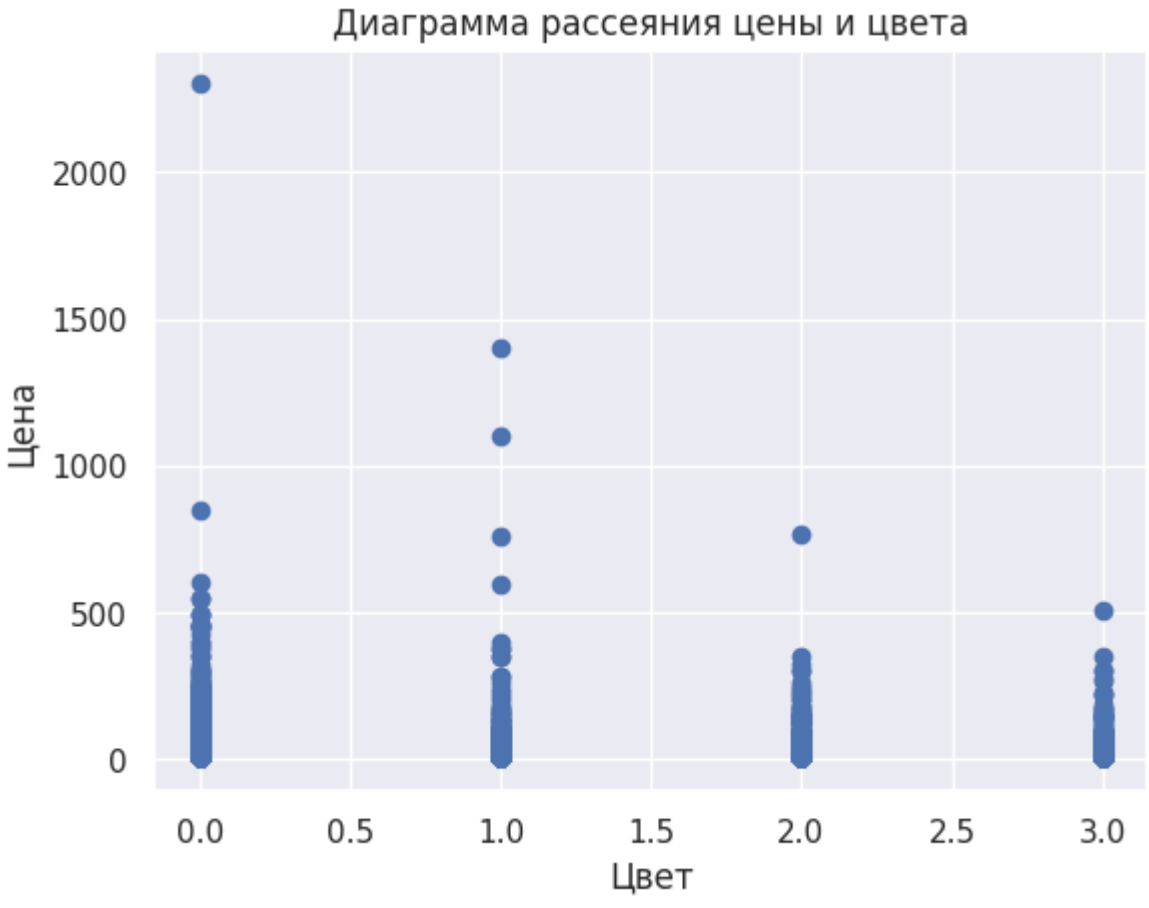
Out[29]:

	price	points
price	1.000000	0.426109
points	0.426109	1.000000

In [30]:
```
#Диаграмма рассеяния цены и цвета показывает независимость цены и цвета
plt.scatter(x = df['color'], y = df['price']) #  цена  и цвет
plt.ylabel("Цена")
plt.xlabel("Цвет")
plt.title('Диаграмма рассеяния цены и цвета')
plt.show()
```



In [31]:
```
color_num={'red':0,'white':1,'other1':2,'other':3}
df_rus['цвет_число']=df_rus['цвет'].map(color_num)
plt.scatter(x = df_rus['цвет_число'], y = df_rus['цена']) #  цена  и цвет
plt.ylabel("Цена")
plt.xlabel("Цвет")
plt.title('Диаграмма рассеяния цены и цвета')
plt.show()
```



In [32]:
```
df_rus[['цена','цвет_число']].corr()
```

```
# корреляция цвета и цены показывает их независимость (-0,105172)
```

Out[32]:

	цена	цвет_число
цена	1.000000	-0.105172
цвет_число	-0.105172	1.000000

3. Портрет пользователя

In [33]:
```
#Сорта лидирующее по региону (топ-5)
result = df.groupby(['country_to_continent', 'variety']).size().reset_index(name='cou
sorted_result = result.sort_values(by=['country_to_continent', 'count'], ascending=[T
print(sorted_result.head())
print(sorted_result[28:33])
print(sorted_result[56:61])
print(sorted_result[373:378])
print(sorted_result[433:438])
print(sorted_result[569:574])
print(sorted_result.tail(3))
```

```
   country_to_continent            variety  count
7                Africa          Chardonnay     41
18               Africa     Sauvignon Blanc     41
19               Africa              Shiraz     39
3                Africa  Cabernet Sauvignon     26
14               Africa            Pinotage     25
   country_to_continent                   variety  count
33                 Asia        Cabernet Sauvignon     22
36                 Asia                Chardonnay     14
46                 Asia                 Red Blend     10
28                 Asia  Bordeaux-style Red Blend      7
41                 Asia                    Merlot      5
    country_to_continent                   variety  count
273               Europe                 Red Blend    584
278               Europe                  Riesling    406
109               Europe                Chardonnay    365
290               Europe                Sangiovese    290
85                Europe  Bordeaux-style Red Blend    267
    country_to_continent             variety  count
397        Latin America              Malbec    282
379        Latin America  Cabernet Sauvignon    263
391        Latin America          Chardonnay    133
419        Latin America     Sauvignon Blanc    111
415        Latin America           Red Blend    110
    country_to_continent            variety  count
521        North America          Pinot Noir   1318
452        North America  Cabernet Sauvignon   1111
465        North America          Chardonnay   1001
546        North America               Syrah    509
568        North America           Zinfandel    487
    country_to_continent            variety  count
605              Oceania              Shiraz    168
601              Oceania     Sauvignon Blanc    147
577              Oceania          Chardonnay    139
593              Oceania          Pinot Noir    127
573              Oceania  Cabernet Sauvignon     70
    country_to_continent   variety  count
619              Unknown     Meoru      1
620              Unknown  Viognier      1
621              Unknown    Vranec      1
```

In [34]:
```
PriceMeanRegion=df.groupby('country_to_continent')['price'].mean()
RatingMeanRegion=df.groupby('country_to_continent')['points'].mean()
```

```
plt.scatter(x = RatingMeanRegion, y = PriceMeanRegion) #  рейтинг и цена
plt.ylabel("Цена")
plt.xlabel("Рейтинг")
plt.title('Диаграмма рассеяния рейтинга и цены по регионам')
plt.show()
```

In [35]:
```
PriceMeanRegionValues=df.groupby('country_to_continent')['price'].mean().values
RatingMeanRegionValues=df.groupby('country_to_continent')['points'].mean().values
data = {'x': RatingMeanRegionValues,
        'y': PriceMeanRegionValues}
dfRegionRatingPrice = pd.DataFrame(data)

correlation = dfRegionRatingPrice ['x'].corr(dfRegionRatingPrice ['y'])
print("Корреляция между двумя наборами данных:", correlation,
      "показывает зависимсть цены от рейтинга по регионам.Значит ответ,да, влияет.")
```

```
Корреляция между двумя наборами данных: 0.615426868190533 показывает зависимсть цены о
т рейтинга по регионам.Значит ответ,да, влияет.
```

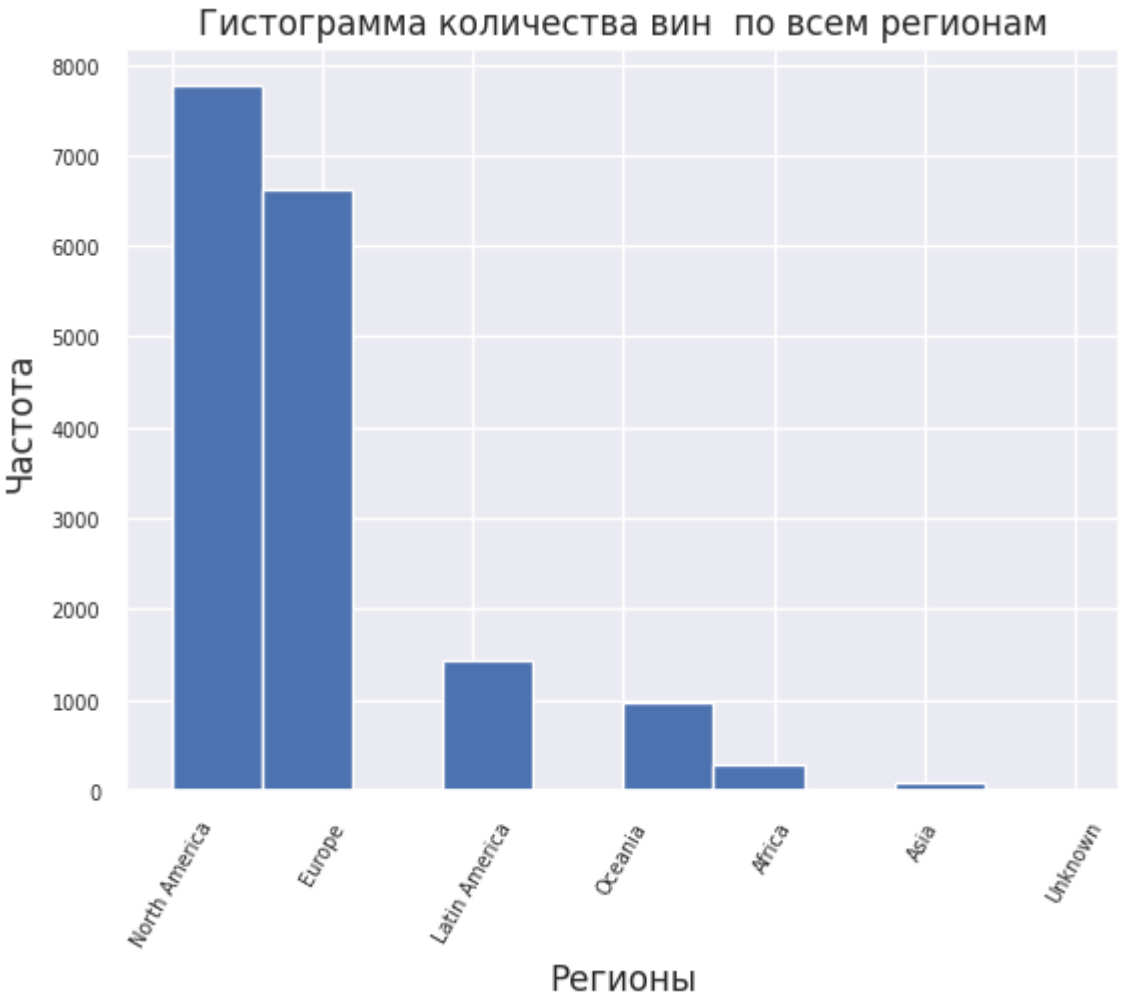
4. Исследование статистических показателей.

In [36]:
```
#подсчет среднего количества,дисперсии и стандартного отклонения
#выполним с помощью функции describe()
df.groupby('country_to_continent')['price'].describe()
```

Out[36]:

country_to_continent	count	mean	std	min	25%	50%	75%	max
Africa	282.0	21.411348	13.532389	6.0	12.0	17.0	26.0	96.0
Asia	94.0	31.031915	18.740705	8.0	16.0	25.0	40.0	120.0
Europe	6624.0	36.883152	57.277640	5.0	15.0	23.0	41.0	2300.0
Latin America	1443.0	21.242550	21.505557	6.0	11.0	15.0	22.0	260.0
North America	7771.0	33.615880	23.417828	6.0	19.0	28.0	42.0	300.0
Oceania	976.0	28.417008	29.035032	5.0	15.0	20.0	30.0	550.0
Unknown	3.0	23.666667	22.810816	10.0	10.5	11.0	30.5	50.0

In [37]:
```
#Гистограмма количества вин  по всем регионам
plt.hist(df['country_to_continent'])
plt.xlabel('Регионы')
plt.ylabel('Частота')
plt.title('Гистограмма количества вин  по всем регионам')
plt.tick_params(axis='x', labelrotation=60)
plt.xticks(fontsize=7)
plt.yticks(fontsize=7)
plt.show()
```

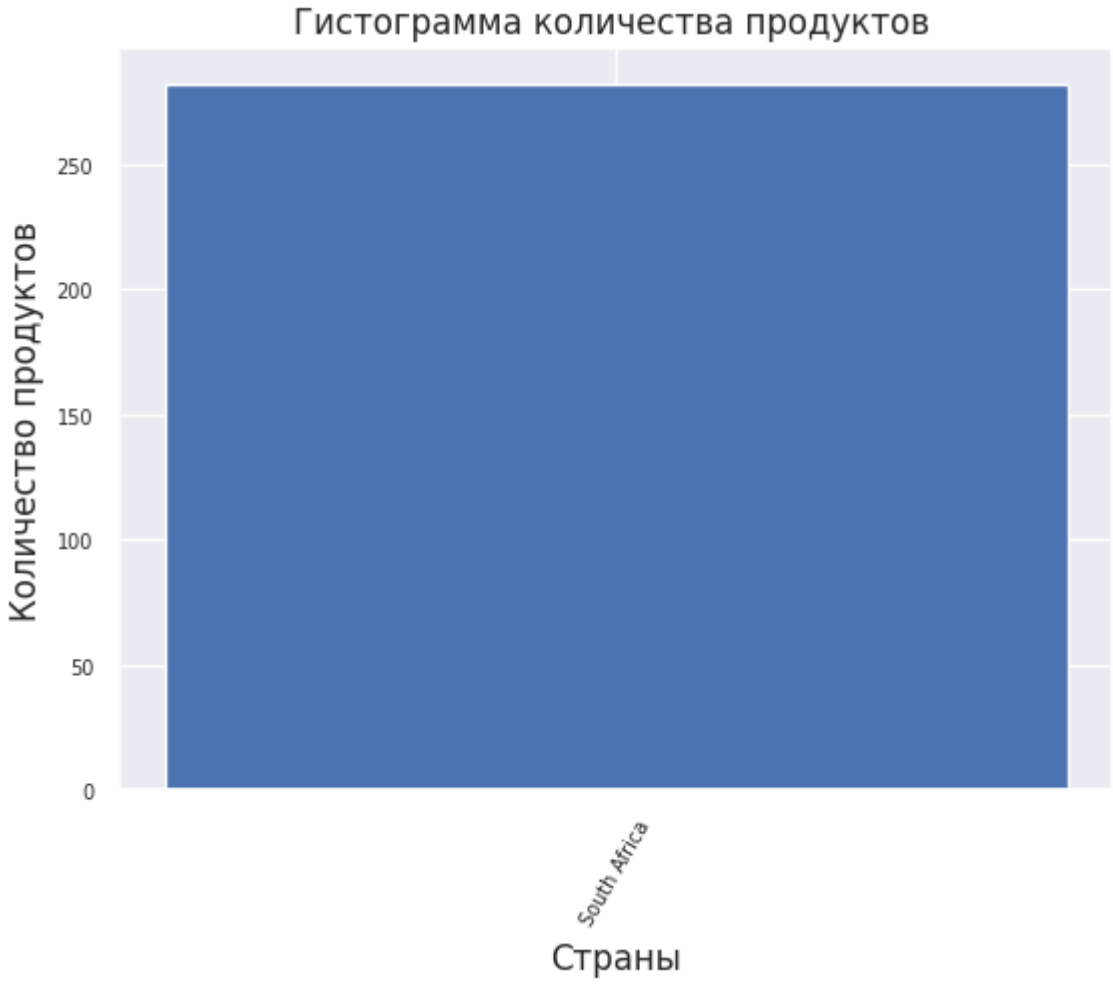


Данное распределение можно расматривать как пуассоновское или геометрическое

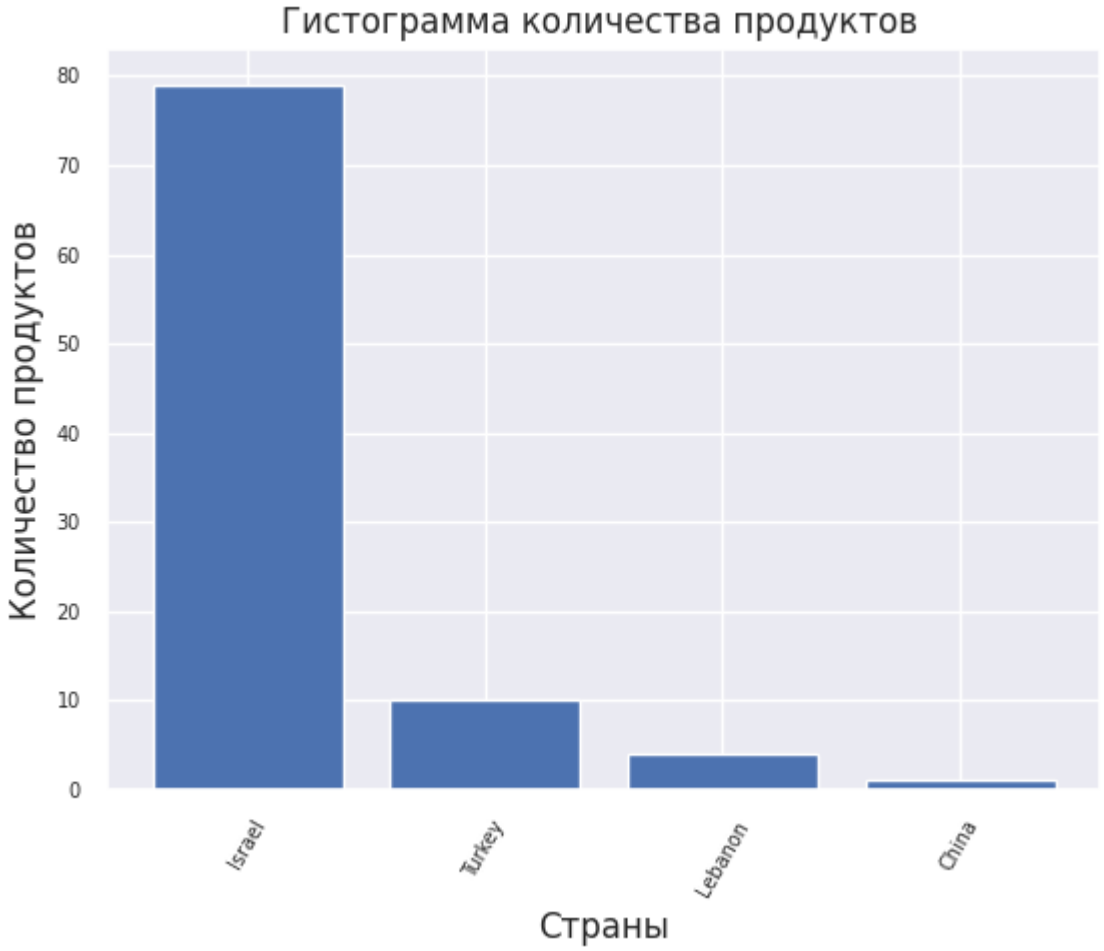
In [38]:
```
countryTMP = df.groupby('country_to_continent')['country'].unique()
for countries in countryTMP.values:
    print(*countries)

#     Фильтрация данных по заданным странам
    filtered_df = df[df['country'].isin(countries)]
```

```
#     Получение количества продуктов в каждой стране

    counts = filtered_df['country'].value_counts()

    # Построение гистограммы
    plt.bar(counts.index, counts.values)
    plt.xlabel('Страны')
    plt.ylabel('Количество продуктов')
    plt.title('Гистограмма количества продуктов ')
    plt.tick_params(axis='x', labelrotation=60)
    plt.xticks(fontsize=7)
    plt.yticks(fontsize=7)
    plt.show()
```

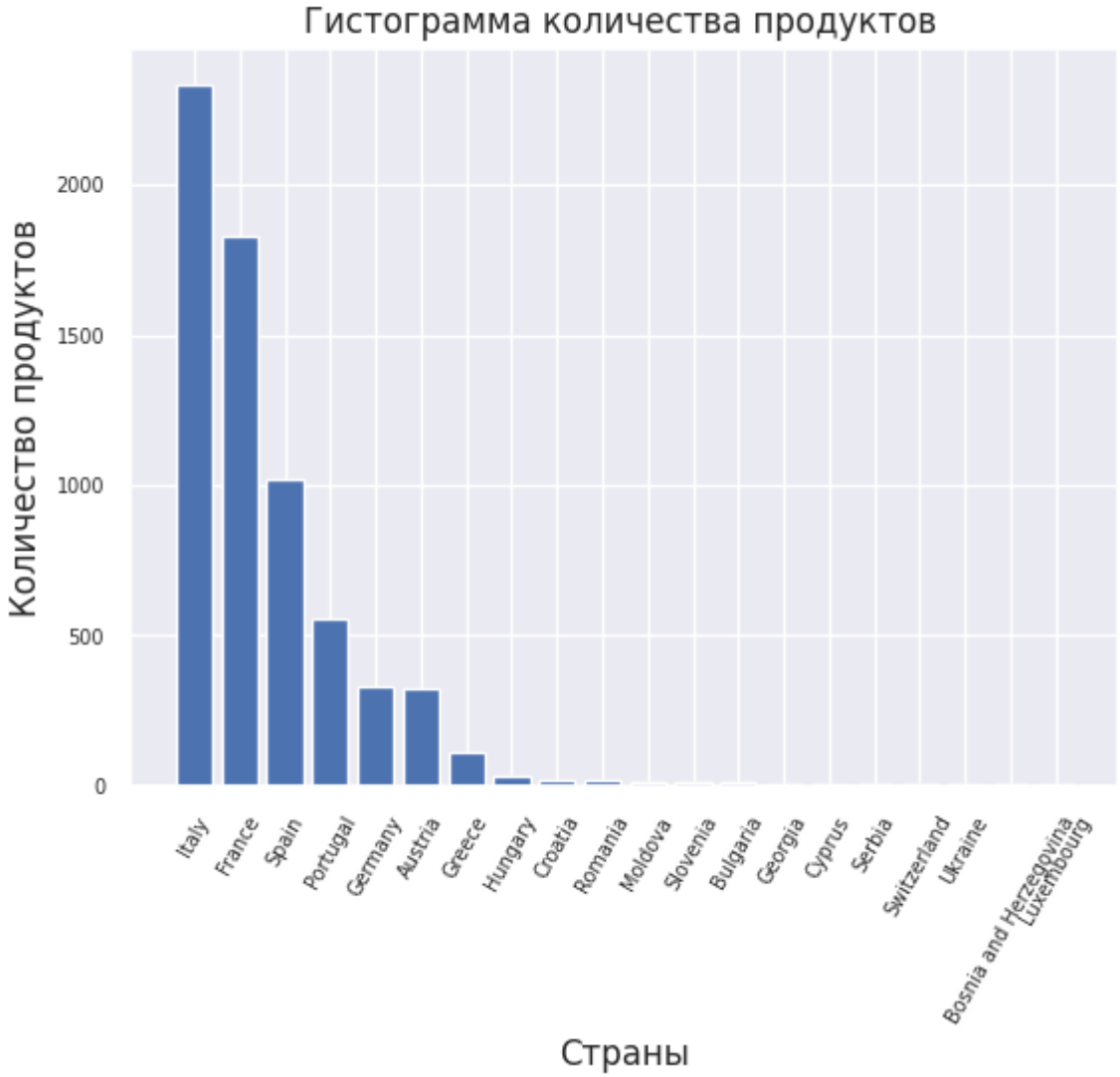
South Africa



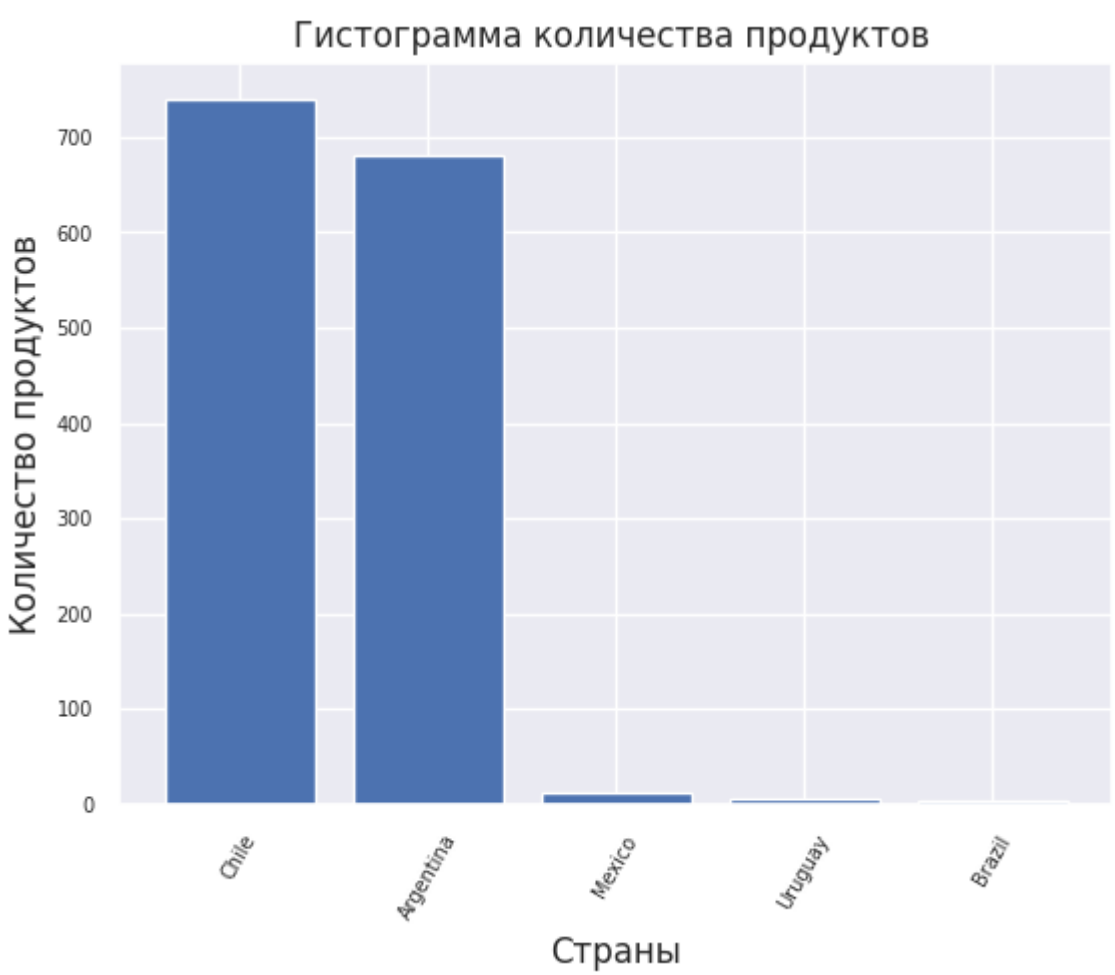
Israel Lebanon China Turkey



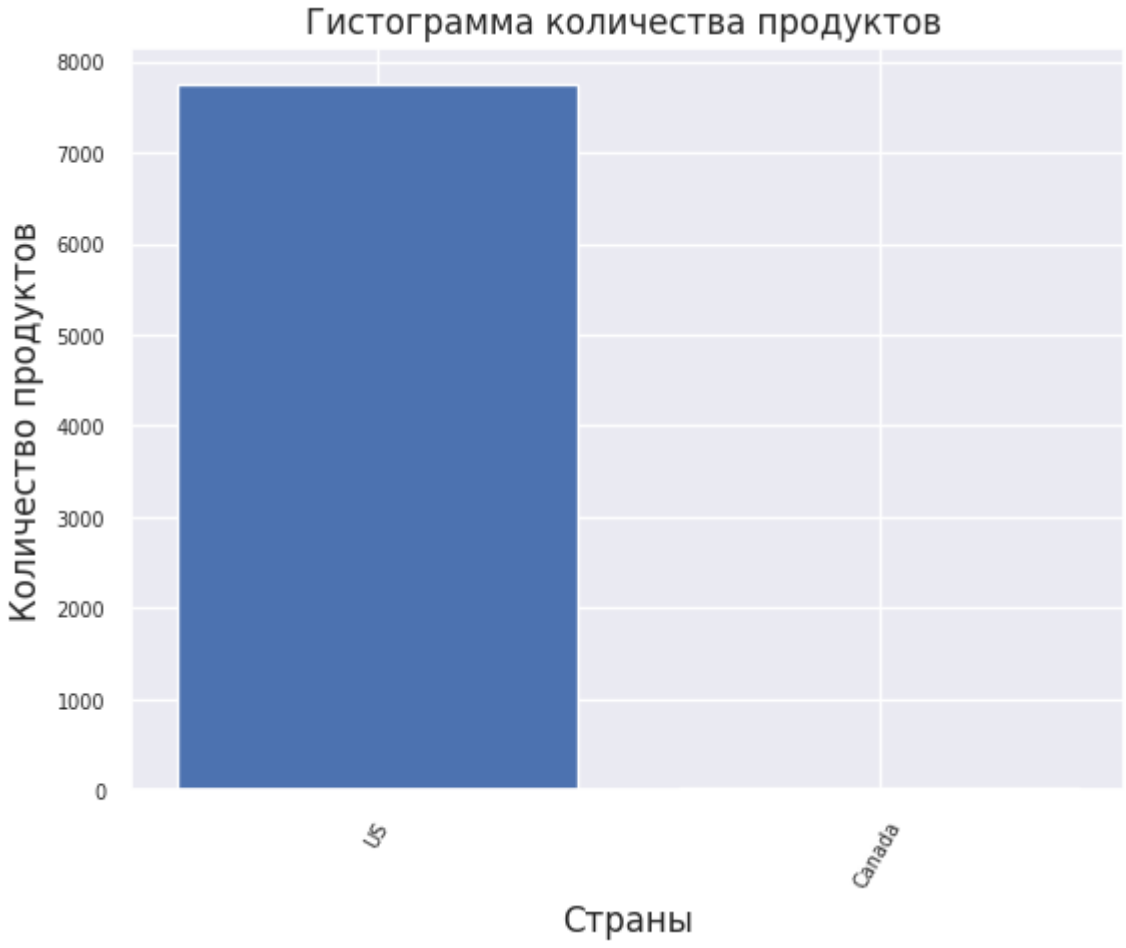
Italy France Austria Spain Portugal Germany Greece Romania Croatia Hungary Slovenia Bu
lgaria Cyprus Switzerland Georgia Moldova Serbia Ukraine Bosnia and Herzegovina Luxemb
ourg



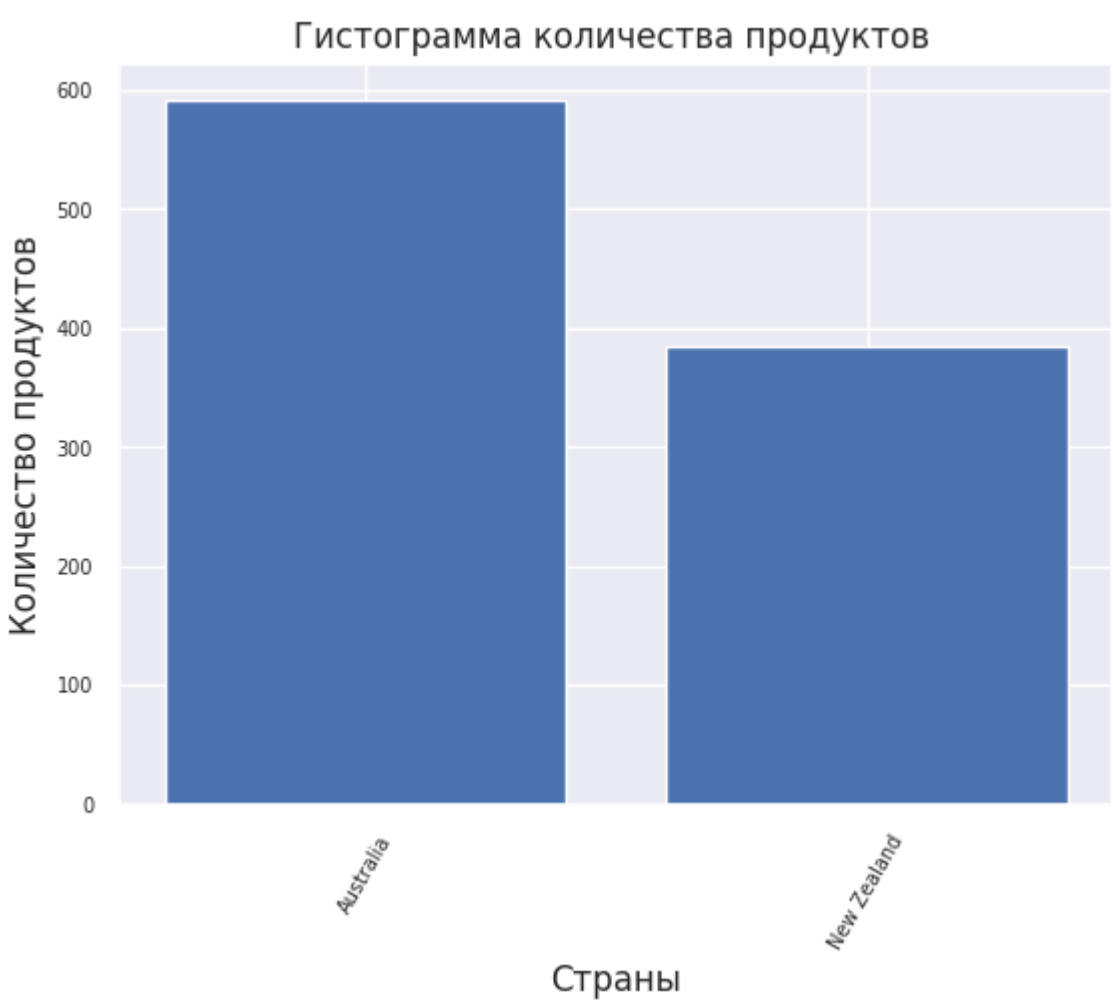
Chile Argentina Mexico Uruguay Brazil



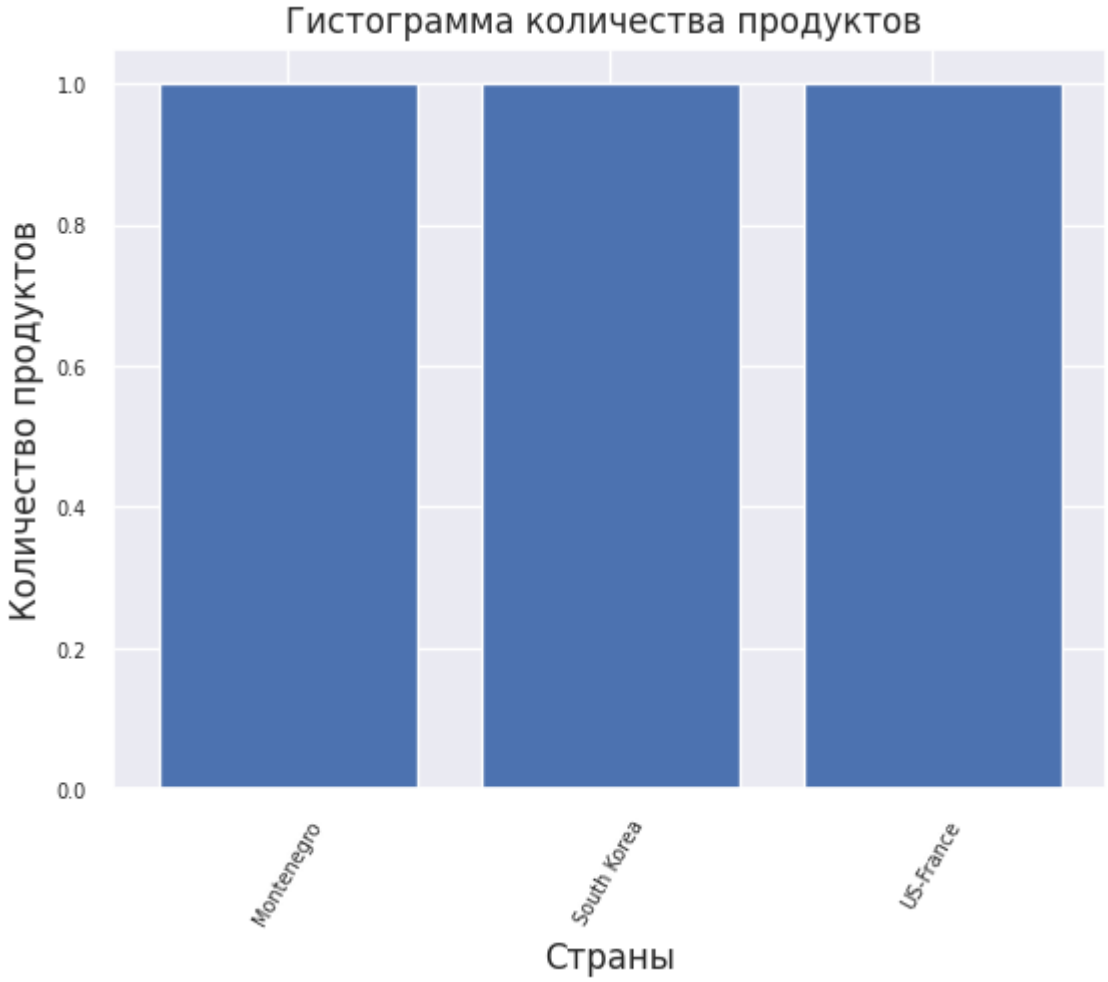
US Canada



Australia New Zealand




```
Montenegro South Korea US-France
```

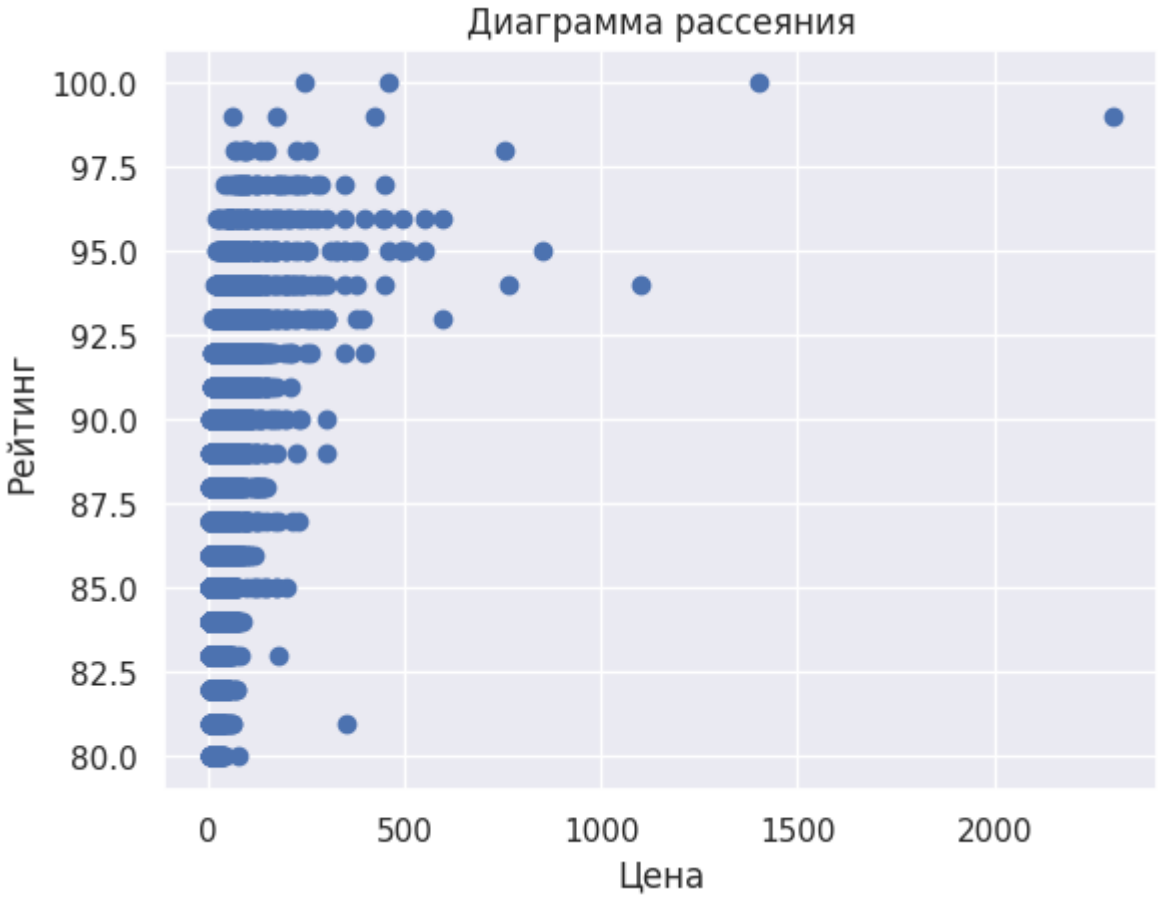


По континентам распредиления:Africa-равномерное, Asia-геометрическое,Europe-пуасоновское,Latin America-равномерное,North America-равномерное,Oceania-биноминальное,Unknown - равномерное.

In [39]:
```
countryTMP = df.groupby('country_to_continent')['country'].unique()
for countries in countryTMP.values:
    print(*countries)
```

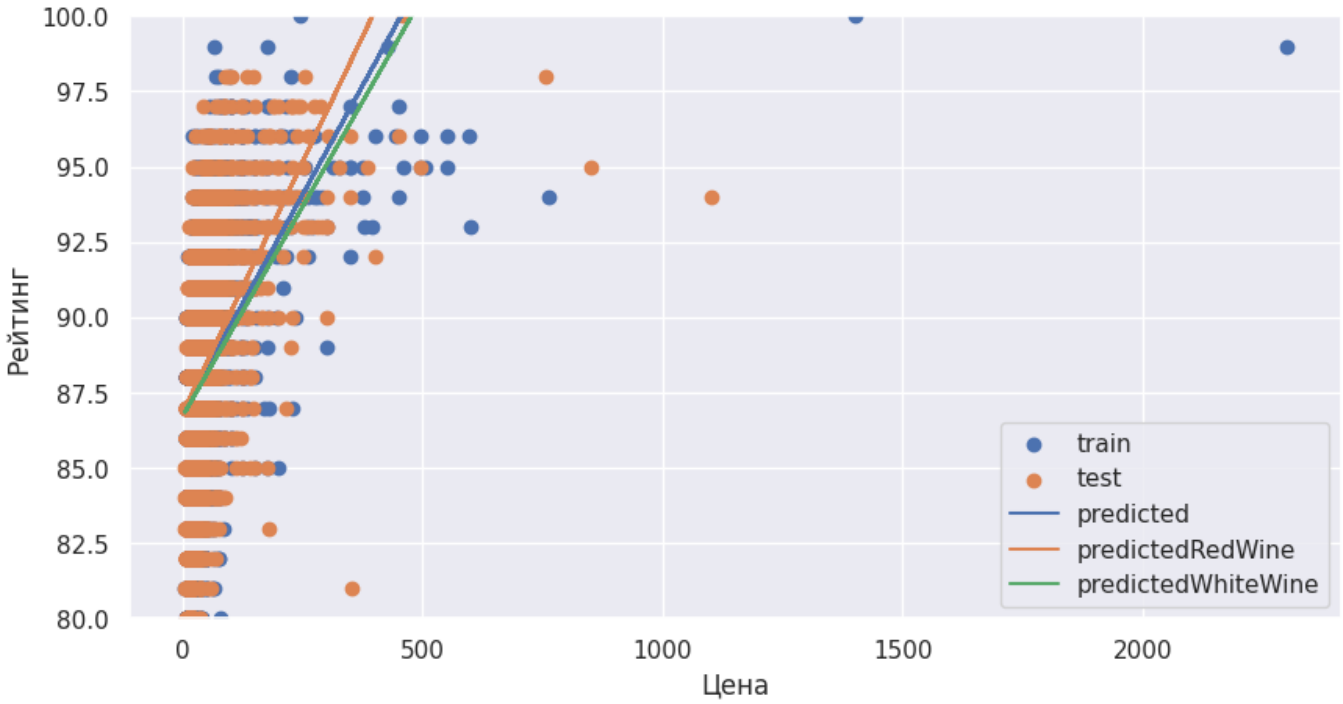
```
South Africa
Israel Lebanon China Turkey
Italy France Austria Spain Portugal Germany Greece Romania Croatia Hungary Slovenia Bu
lgaria Cyprus Switzerland Georgia Moldova Serbia Ukraine Bosnia and Herzegovina Luxemb
ourg
Chile Argentina Mexico Uruguay Brazil
US Canada
Australia New Zealand
Montenegro South Korea US-France
```

In [40]:
```
#Строим диаграмму рассеивания
plt.scatter(x =df['price'], y =df['points'] ) # рейтинг и цена
plt.xlabel("Цена")
plt.ylabel("Рейтинг")
plt.title('Диаграмма рассеяния')
plt.show()
```



In [41]:
```
# Линейная регрессия зависимости между рейтингом и ценой продукта
# Загружаем библотеку класса LinearRegression
from sklearn.linear_model import LinearRegression
#Подготавливаем данные
sampleR = df.loc[df['color'] == 'red', 'price'].astype(float).to_numpy()
sampleW = df.loc[df['color'] == 'white', 'price'].astype(float).to_numpy()
sampleX = df['price'].astype(float).to_numpy()
sampleY = df['points'].astype(float).to_numpy()
rating_red_wine = df.loc[df['color'] == 'red', 'points'].astype(float).to_numpy()
rating_white_wine = df.loc[df['color'] == 'white', 'points'].astype(float).to_numpy()
# разделяем данные на обучающую и тестовую часть
X_test = sampleX[0::2].reshape(-1,1)
y_test = sampleY[0::2]
X_train = sampleX[1::2].reshape(-1,1)
y_train = sampleY[1::2]
# Делаем ряды данных для красного и белого вина
```

```
X_testRW = sampleR.reshape(-1,1)
y_testRW =rating_red_wine
X_testWW = sampleW.reshape(-1,1)
y_testWW =rating_white_wine


#регриссионная модель
model =LinearRegression()
model.fit(X_train,y_train)
w = model.coef_

b = model.intercept_

#регриссионная модель по красному вину
modelRedWine =LinearRegression()
modelRedWine.fit(X_testRW,y_testRW)
wRW = modelRedWine.coef_

bRW = modelRedWine.intercept_

#регриссионная модель по белому вину
modelWhiteWine =LinearRegression()
modelWhiteWine.fit(X_testWW,y_testWW)
wWW = modelWhiteWine.coef_

bWW = modelWhiteWine.intercept_



plt.figure(figsize=(10,5))
#plt.plot(X_train,y_train,label='real')

plt.scatter(X_train,y_train,label='train')
plt.scatter(X_test,y_test,label='test')
# Чертим линию тренда
plt.plot(sampleX[1::2],sampleX[1::2].reshape(-1,1).dot(w)+b,label='predicted')
# Чертим линию тренда красного вина
plt.plot(sampleR,sampleR.reshape(-1,1).dot(wRW)+bRW,label='predictedRedWine')
# Чертим линию тренда белого вина
plt.plot(sampleW,sampleW.reshape(-1,1).dot(wWW)+bWW,label='predictedWhiteWine')
plt.legend(loc='best')
plt.xlabel('Цена')
plt.ylabel('Рейтинг')
plt.ylim(80, 100)  # Установка пределов для оси y
# Метод plt.axis()
plt.axis(ymin=80, ymax=100)  # Установка пределов для оси y
plt.show()
# Выводим на печать линию тренда
print('Линия тренда:Рейтинг=',w,"Цена +",b)



from sklearn.metrics import mean_squared_error
y_train_predicted=model.predict(X_train)
y_test_predicted =model.predict(X_test)
print('TrainMSE:',mean_squared_error(y_train,y_train_predicted))
print('TrainMSE:',mean_squared_error(y_test,y_test_predicted))
```




```
Линия тренда:Рейтинг= [0.02895724] Цена + 86.82489811835337
TrainMSE: 8.970129769139975
TrainMSE: 8.26195517474155
```

5. Проверка гипотез

- H0: Средние пользовательские рейтинги красного и белого вина одинаковые.
- H1:Средние пользовательские рейтинги красного и белого вина разные.

Задаем порговое значение alpha=0.05

In [42]:
```
# Подготовка данных
sampleR = df.loc[df['color'] == 'red', 'points'].astype(float).to_numpy()
sampleW = df.loc[df['color'] == 'white', 'points'].astype(float).to_numpy()
# С помощью библиотеки stats и функции ttest_ind проводим проверку гипотез H0 и H1
from scipy import stats
print(stats.ttest_ind(sampleR, sampleW))
```

```
Ttest_indResult(statistic=9.649402481972755, pvalue=5.822246350896306e-22)
```

pvalue=5.822246350896306e-22 поэтому гипотезу H0 отвергаем

- H0:Средние цены двух популярных сортов вина одинаковые.
- H1:Средние цены двух популярных сортов вина разные.

In [43]:
```
# Создадим список популярности
popular_type = df['variety'].value_counts().head()
print("Победители по по пулярности:", popular_type)
popular_type
```

```
Победители по по пулярности: variety
Pinot Noir            1755
Chardonnay            1693
Cabernet Sauvignon    1542
Red Blend             1163
Sauvignon Blanc        770
Name: count, dtype: int64
```

Out[43]:
```
variety
Pinot Noir            1755
Chardonnay            1693
Cabernet Sauvignon    1542
Red Blend             1163
Sauvignon Blanc        770
Name: count, dtype: int64
```

Перепишим гипотезы НО и Н1:

- H0:Средние цены двух популярных сортов Pinot Noir и Chardonnay одинаковые.
- H1:Средние цены двух популярных сортов Pinot Noir и Chardonnay одинаковые.

In [44]:
```
# Найдем ряды данных  цен на Pinot Noir и Chardonnay
samplePinotNoir = df.loc[df['variety'] == 'Pinot Noir', 'price'].astype(float).to_num
sampleChardonnay = df.loc[df['variety'] == 'Chardonnay', 'price'].astype(float).to_nu
# С помощью библиотеки stats и функции ttest_ind проводим проверку гипотез H0 и H1
print(stats.ttest_ind(samplePinotNoir, sampleChardonnay))
```

```
Ttest_indResult(statistic=8.153145631000864, pvalue=4.912364067080219e-16)
```

pvalue=4.912364067080219e-16 поэтому гипотезу H0 отвергаем

6. Выводы

- 1.Данные/Подготовка данных.

1. Набор данных представляет собой таблицу из 20000 строк и 10 столбцов.Выявленный характер заполнения показал наличие дубликатов строк. В порядка 9% строк датасета отсутвуют значения цены продукции.Стобец Регион2 на 60% содержит пустые значения.
2. После удаления дубликатов и пустых значений остался датасет 17193 строки и 9столбцов. Наличие дополнительных словарей по континентам и по цветам вина дали датасет 17193x11.
3. По данным датасета был проведен анализ стран производителей вина. Он показал, отсутвие Российской Федерации в списке основных поставщиков вина на мировой рынок. Это может говорить о том, что компании составлявшие данные списки могут быть аффилированы производителями данной продукции.
4. Рейтинги представлены в диапозоне от 80 до 100 баллов.Возникает законный вопрос: почему нет значений рейтенга от 0 до 80 баллов.
5. Цены колеблются от 5$ до 2300$.Цены указанные в датасете не могут дать ответ на вопрос: это цена производителя продукции или цена реализации в продовольственных сетях. Цены на вина в РФ (например магазин "Окей" розница) начинаются от 2,2$(199руб), что тоже дает сомнения в реалистичности приводимых данных. Приведенный анализ опирается только на данные представленные в наборе данных.Предполагается представленные цены - конечные значения реализации продукции.

* 2.Исследвательский анализ данных.

- Список лидирующих по рейтенгу сортов вин:

In [45]:
```
df.sort_values(by='points',ascending=False).head(3)
```

Out[45]:

	country	description	designation	points	price	province	region_1	variety	
323	France	A wine that has created its own universe. It h...	Clos du Mesnil	100	1400.0	Champagne	Champagne	Chardonnay	
17967	US	Impossibly aromatic. Hard to imagine greater c...	Red Wine	100	245.0	California	Rutherford	Cabernet Blend	
5955	Italy	A perfect wine from a classic vintage, the 200...	Masseto	100	460.0	Tuscany	Toscana	Merlot	dell'(

-

- Список самых дорогих вин:

In [46]:
```
df.sort_values(by='price',ascending=False).head(3)
```

Out[46]:

	country	description	designation	points	price	province	region_1	variety
13188	France	A big, powerful wine that sums up the richness...	other_design	99	2300.0	Bordeaux	Pauillac	Bordeaux-style Red Blend
323	France	A wine that has created its own universe. It h...	Clos du Mesnil	100	1400.0	Champagne	Champagne	Chardonnay
4324	Austria	Wet earth, rain-wet stones, damp moss, wild sa...	Ried Loibenberg Smaragd	94	1100.0	Wachau	other_region_1	Grüner Veltliner	E

-

- Из приведенных значений видно, что максимальная цена на определенный вид вина может не получить максимальный рейтинговый балл.

- Средняя цена в регинах лежит от 21$ до 37$.

- Бюджетные вина до 20$.

In [47]:
```
filtered_df = df.loc[df['price'] <= 20]  # Фильтрация строк по условию "price"<20
popular_varieties = filtered_df['variety'].value_counts().sort_values(ascending=False
popular_varieties.head(10)
```

Out[47]:
```
variety
Chardonnay            743
Sauvignon Blanc       547
Cabernet Sauvignon    492
Red Blend             405
Riesling              332
Merlot                301
Pinot Noir            297
Rosé                  282
Malbec                223
White Blend           210
Name: count, dtype: int64
```

-

- Корреляция цвета и цены вина показала их независимость.

* 3.Портрет потребителя региона:

```
<ul>
    <li>Марка "Chardonnay" представлена во всех регионах в достаточном количестве. Группу вин "Sauvignon Blanc", "Pinot Noir",  "Red Blend" - можно отнести ко второй группе привлекательности в регионах.
</ul>
<ul>
    <li>Рейтинг вина показывает тенденцию к росту цен на виды вина.
</li>
</ul>
```

- 4.Уравнения линейной регрессии рейтинга от цены:
  - Получено уравнение связывающее рейтинг и цену:
    Рейтинг= 0.02896*Цена + 86.8

Список литературы

1. Андерсон, К, Аналитическая культура: от сбора данных до бизнес-результатов / Карл Андерсон. - Москва : Манн, Иванов и Фербер, 2017. - 324 с.
2. Бенгфорт Бенджамин, Билбро Ребекка, Охеда Тони, Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. — СПб.: Питер, 2019.
3. Мэтиз Э., Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017.
4. Плас Дж. Вандер, Python для сложных задач: наука о данных и машинное обучение. — СПб.: Питер, 2018.
5. Рашка С., Рашка С. Р28 Python и машинное обучение / пер. с англ. А. В. Логунова. - М.: ДМК Пресс, 2017.
6. Шарден Б., Массарон Л., Боскетти А., Крупномасштабное машинное обучение вместе с Python. Пер. с анг. А. В. Логунова. – М.: ДМК Пресс, 2018.