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

ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА

на тему

«Анализ данных с использованием Python»

(на примере анализа данных исследуемого продукта)

слушателя Полуниной Марии Михайловны
группы № 143
по программе профессиональной переподготовки

«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',
  '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.

Вывод

1.Предобработка данных

Импортируем необходимые библиотеки:

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.cm as cm
import math
import statistics

import scipy.stats as st
import scipy
import statsmodels.api as sm

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_absolute_error, r2_score

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
warnings.simplefilter(action='ignore', category=UserWarning)
```

Загрузка данных:

In [3]:
```
df = pd.read_csv('wine_reviews.csv')
df.head()
```

Out[3]:

	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	Ma
1	Italy	D'Alceo is a drop dead gorgeous wine that ooze...	D'Alceo	96	275.0	Tuscany	Toscana	NaN	Red Blend	R
2	France	The great dominance of Cabernet Sauvignon in t...	NaN	91	40.0	Bordeaux	Haut-Médoc	NaN	Bordeaux-style Red Blend	Ber
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	De

Общая информация о данных:

In [4]:
```
print(df.info())
print(df.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 10 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 
dtypes: float64(1), int64(1), object(8)
memory usage: 1.5+ MB
None
(20000, 10)
```

Просмотр пустых значений:

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
country            0
description        0
designation     6001
points             0
price           1802
province           0
region_1        3457
region_2       11942
variety            0
winery             0
dtype: int64
```

In [6]:
```
MissingValue = df.isnull().sum().sort_values(ascending = False)
Percent = (df.isnull().sum()/df.isnull().count()*100).sort_values(ascending = Fals
MissingData = pd.concat([MissingValue, Percent], axis=1, keys=['Пропущенные значени
MissingData
```

Out[6]:

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

In [7]:
```
colours = ['#993366', '#FFFF00']
sns.heatmap(df.isnull(), cmap=sns.color_palette(colours))
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.figtext(0.2, -0.2, "Рисунок 1. - Матрица пропущенных значений набора данных")
plt.show()
```

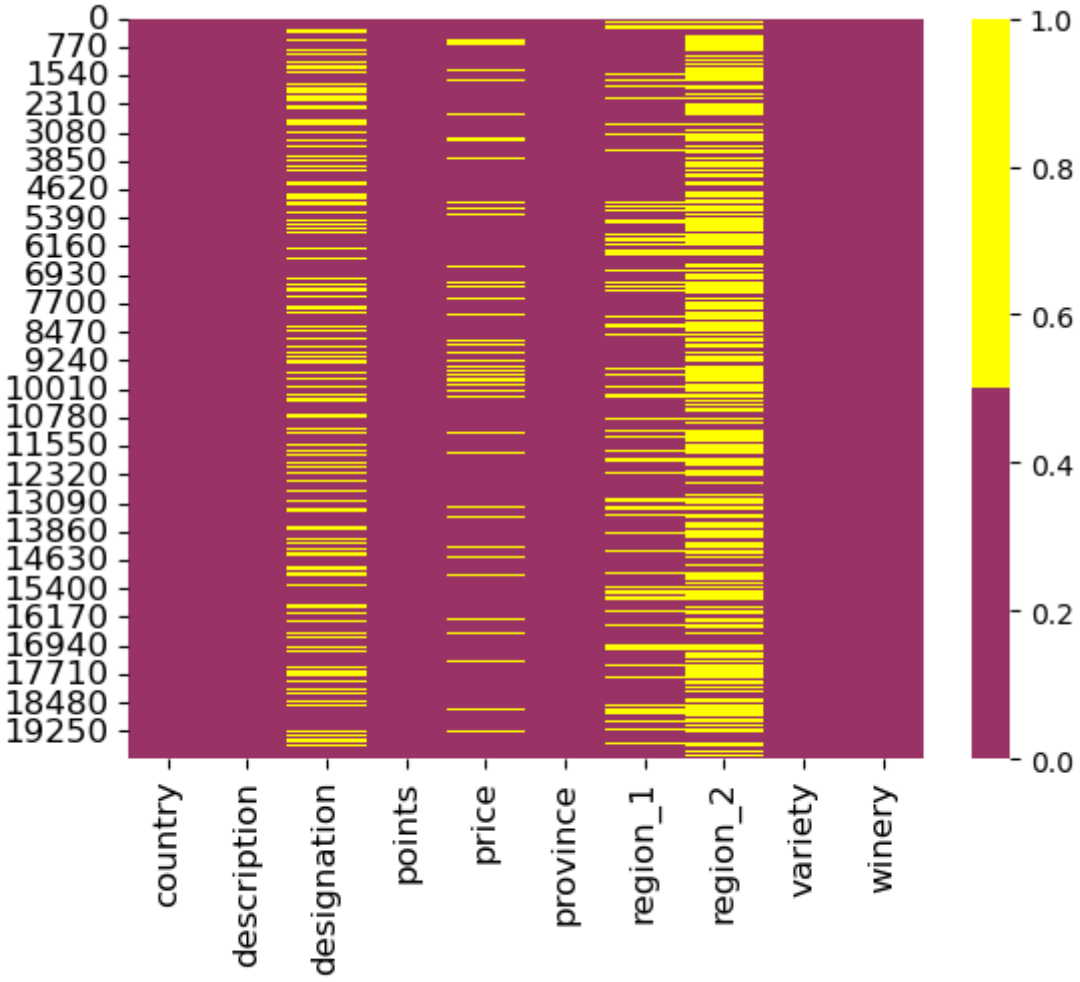


Рисунок 1. - Матрица пропущенных значений набора данных

Описательная статистика по количественным показателям:

In [8]:
```
df.describe().T
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
points	20000.0	87.898700	3.243049	80.0	86.0	88.0	90.0	100.0
price	18198.0	33.206891	39.716685	5.0	16.0	24.0	40.0	2300.0

Как мы видим, в датасете лишь два количественных признака: Баллы (points) и Цена (price). Последний имеет около 9% пропусков, которые вполне можно заменить синтетическими данными, например средними значениями.

In [9]:
```
df['price'].fillna(df['price'].mean(), inplace=True)
```

Проверяем пустые значения

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
country            0
description        0
designation     6001
points             0
price              0
province           0
region_1        3457
region_2       11942
variety            0
winery             0
dtype: int64
```

Посмотрим, как изменилась статистика после замены пустых значений с ценой на общее среднее:

In [11]:
```
df.describe().T
```

Out[11]:

	count	mean	std	min	25%	50%	75%	max
points	20000.0	87.898700	3.243049	80.0	86.0	88.0	90.0	100.0
price	20000.0	33.206891	37.885127	5.0	16.0	25.0	38.0	2300.0

Изменения показателей незначительны, а значит с датасетом можно работать дальше, без ущерба достоверности.

Для удобства работы с количесвенными показателями изменим тип данных в столбце Цена (price) с float на int64:

In [12]:
```
df['price'] = df['price'].astype('int64')
```

In [13]:
```
df.dtypes
```

Out[13]:
```
country        object
description    object
designation    object
points          int64
price           int64
province       object
region_1       object
region_2       object
variety        object
winery         object
dtype: object
```

Заполним оставшиеся пропуски по колонкам Регион 1 (region_1), Регион 2 (region_2) и Обозначение (designation) на значение "Неизвестно" (Unknown)

In [14]:
```
for col in ('designation', 'region_2', 'region_1'):
    df[col]=df[col].fillna('Unknown')
df.isnull().sum()
```

Out[14]:
```
country        0
description    0
designation    0
points         0
price          0
province       0
region_1       0
region_2       0
variety        0
winery         0
dtype: int64
```

Подсчитаем средние цены для каждой страны:

In [15]:
```
print(df['country'].unique())
print('Количество стран:', len(pd.unique(df['country'])))
```

```
['US' 'Italy' 'France' 'Austria' 'Chile' 'Spain' 'Australia'
 'South Africa' 'New Zealand' 'Portugal' 'Argentina' 'Germany' 'Greece'
 'Canada' 'Israel' 'Romania' 'Croatia' 'Hungary' 'Mexico' 'Slovenia'
 'Lebanon' 'China' 'Bulgaria' 'Cyprus' 'Uruguay' 'Switzerland' 'Turkey'
 'Georgia' 'Moldova' 'Montenegro' 'Serbia' 'South Korea' 'Ukraine'
 'Bosnia and Herzegovina' 'Brazil' 'US-France' 'Egypt' 'Luxembourg']
Количество стран: 38
```

In [16]:
```
print(df['variety'].unique())
print('Сортов вина:', len(pd.unique(df['variety'])))
```

```
['Pinot Noir' 'Red Blend' 'Bordeaux-style Red Blend' 'Sangiovese'
 'Riesling' 'Syrah' 'Merlot' 'Chardonnay' 'Sauvignon Blanc' 'Albariño'
 'Cabernet Sauvignon' 'Shiraz' 'Rosé' 'Vermentino' 'Pinot Grigio'
 'Pinot Gris' 'Nebbiolo' 'Gamay' 'Tinto del Pais' 'Brachetto' 'Grenache'
 'Portuguese White' 'Alicante Bouschet' 'Tempranillo'
 'Corvina, Rondinella, Molinara' 'Pinot Noir-Gamay' 'Moscato'
 'Chenin Blanc' 'Cabernet Franc' 'Monastrell-Syrah'
 'Rhône-style Red Blend' 'Austrian white blend' 'White Blend' 'Barbera'
 'Tempranillo Blend' "Nero d'Avola" 'Champagne Blend' 'Zinfandel' 'Port'
 'Sparkling Blend' 'Grüner Veltliner' 'Malbec' 'Vidal Blanc'
 'Touriga Nacional' 'Bastardo' 'Portuguese Red' 'Verdejo' 'Viognier'
 'Rhône-style White Blend' 'Sémillon' 'Petite Sirah' 'Verdejo-Viura'
 'Bobal' 'Nasco' 'Fumé Blanc' 'Mission' 'Assyrtico' 'Falanghina'
 'Garnacha' 'Viura' 'Sangiovese Grosso' 'Pinotage' 'Meritage' 'Marsanne'
 'Tannat-Cabernet' 'Loureiro' 'Merlot-Cabernet' 'Malbec-Merlot'
 'Pinot Blanc' 'Bordeaux-style White Blend' 'Mourvèdre'
 'Malbec-Cabernet Sauvignon' 'Gewürztraminer' 'Petit Verdot' 'Vernaccia'
 'Silvaner' 'Primitivo' 'Merlot-Cabernet Sauvignon' 'Tannat'
 'Gros and Petit Manseng' 'Muscat' 'Gelber Muskateller' 'Negroamaro'
 'Dolcetto' 'Cabernet Sauvignon-Shiraz' 'Melon' 'Cabernet Blend'
 'Shiraz-Cabernet Sauvignon' 'Spätburgunder' 'Carmenère' 'Baga'
 'Encruzado' 'Zweigelt' 'Pinot Nero' 'Malvasia Bianca' 'Malvasia'
 'Montepulciano' 'Sauvignon' 'Carricante' 'Mencía' 'G-S-M'
 'Shiraz-Viognier' 'Tokaji' 'Chardonnay-Viognier' 'Kekfrankos'
 'Agiorgitiko' 'Blaufränkisch' 'Prieto Picudo' 'Pigato' 'Castelão'
 'Friulano' 'Tokay' 'Müller-Thurgau' 'Viognier-Chardonnay' 'Zibibbo'
 'Sauvignon Blanc-Semillon' 'Cabernet Sauvignon-Merlot' 'Prosecco'
 'Syrah-Grenache' 'St. Laurent' 'Glera' 'Aglianico' 'Malvasia Nera'
 'Grillo' 'Roussanne' 'Prugnolo Gentile' 'Mavrodaphne' 'Black Muscat'
 'Garganega' 'Insolia' 'Antão Vaz' 'Frappato'
 'Touriga Nacional-Cabernet Sauvignon' 'Syrah-Cabernet'
 'Syrah-Petite Sirah' 'Tinta de Toro' 'Provence red blend'
 'Tocai Friulano' 'Nuragus' 'Grenache Blanc' 'Claret' 'Petit Manseng'
 'Muscatel' 'Passerina' 'Torrontés' 'Teran' 'Malbec-Tannat'
 "Cesanese d'Affile" 'Vignoles' 'Verdicchio' 'Sherry' 'Malbec-Syrah'
 'Turbiana' 'Trebbiano' 'Austrian Red Blend' 'Muscat Canelli'
 'Hondarrabi Zuri' 'Verdelho' 'Grenache-Syrah' 'Verduzzo Friulano '
 'Rkatsiteli' 'Pelaverga Piccolo' 'Cabernet Sauvignon-Malbec'
 'Portuguese Sparkling' 'Alvarinho' 'Weissburgunder' 'White Riesling'
 'Cayuga' 'Fiano' 'Semillon-Chardonnay' 'Rosado' 'Bonarda' 'Apple'
 'Godello' 'Cabernet Sauvignon-Carmenère' 'Touriga Franca'
 'Welschriesling' 'Roditis' 'Monastrell' 'Carignane' 'Pecorino' 'Arinto'
 'Lacrima' 'Moschofilero' 'Angevine' 'Cabernet Sauvignon-Syrah'
 'Pallagrello' 'Moscatel' 'Okuzgozu' 'Semillon-Sauvignon Blanc'
 'Cabernet Merlot' 'Duras' 'Shiraz-Tempranillo' 'Furmint' 'Catarratto'
 'Tempranillo-Shiraz' 'Susumaniello' 'Nerello Mascalese' 'Neuburger'
 'Picolit' 'Viura-Sauvignon Blanc' 'Carignan' 'Tempranillo-Garnacha'
 'Debit' 'Karalahna' 'Provence white blend' 'Muscat Blanc' 'Lemberger'
 'Corvina' 'Merlot-Cabernet Franc' 'Lambrusco' 'Sylvaner'
 'Tempranillo-Cabernet Sauvignon' 'Muscat Blanc à Petit Grain'
 'Alfrocheiro' 'Dornfelder' 'Arneis' 'Pinot Bianco' 'Zelen' 'Cabernet'
 'Cabernet Sauvignon-Tempranillo' 'Ribolla Gialla' 'Sagrantino'
 'Gros Manseng' 'Viognier-Valdiguié' 'Ugni Blanc-Colombard' 'Moscadello'
 'Graciano' 'Carignano' 'Chardonnay-Semillon' 'Xinomavro' 'Muscadet'
 'Carmenère-Cabernet Sauvignon' 'Pedro Ximénez' 'Cabernet Franc-Carmenère'
 'Cannonau' 'Grenache-Shiraz' 'Chinuri' 'Raboso' 'Viognier-Marsanne'
 'Teroldego' 'Greco' 'Airen' 'Alsace white blend'
 'Cabernet Sauvignon-Cabernet Franc' 'Nosiola' 'Fernão Pires' 'Cinsault'
 'Verduzzo' 'Xarel-lo' 'Aligoté' 'Merlot-Malbec' 'Petite Verdot' 'Rosato'
 'Veltliner' 'Sauvignon Gris' 'Baco Noir' 'Orange Muscat' 'Muscadel'
 'Syrah-Cabernet Sauvignon' 'Lagrein' 'Vranec' "Pineau d'Aunis"
 'Tinta Fina' 'Trepat' 'Tinta Roriz' 'Fer Servadou' 'Madeira Blend'
 'Cortese' 'Marsanne-Roussanne' 'Traminette' 'Siegerrebe' 'Monica'
 'Norton' 'Chardonnay-Sauvignon' 'Cabernet Franc-Merlot' 'Tai' 'Traminer'
```

```
 'Macabeo' 'Syrah-Tempranillo' 'St. George' 'Colombard-Sauvignon Blanc'
 'Tamjanika' 'Colombard' 'Vermentino Nero' 'Muskat Ottonel' 'Inzolia'
 'Azal' 'Shiraz-Pinotage' 'Palomino' 'Bical' 'Tempranillo-Merlot'
 'Rieslaner' 'Auxerrois' 'Garnacha-Cabernet' 'Posip' 'Pinot Meunier'
 'Schiava' 'Grauburgunder' 'Saperavi' 'Merlot-Shiraz' 'White Port' 'Meoru'
 'Gelber Traminer' 'Verdeca' 'Cariñena-Garnacha' 'Tinto Fino' 'Piedirosso'
 'Blatina' 'Chenin Blanc-Viognier' 'Carignan-Grenache' 'Plavac Mali'
 'Charbono' 'Moscato Giallo' 'Roter Traminer' 'Mataro' 'Maréchal Foch'
 'Gaglioppo' 'Tintilia ' 'Shiraz-Grenache' 'Mantonico' 'Savagnin'
 'Gamay Noir' 'Uva di Troia' 'Scheurebe' 'Aidani' 'Trincadeira' 'Ansonica'
 "Muscat d'Alexandrie" 'Grechetto' 'Malvasia Istriana'
 'Chardonnay Weissburgunder' 'Listán Negro' 'Rotgipfler' 'Cabernet-Syrah'
 'Mauzac' 'Chenin Blanc-Chardonnay' 'Vespaiolo' 'Chardonnay-Pinot Grigio'
 'Carmenère-Syrah' 'Malbec-Cabernet Franc' 'Boğazkere' 'Elbling'
 'Garnacha Blanca' 'Feteasca Neagra' 'Zlahtina' 'Viura-Chardonnay'
 'Counoise' 'Pied de Perdrix' 'Huxelrebe' 'Viognier-Roussanne'
 'Syrah-Viognier' 'Cabernet Sauvignon-Sangiovese' 'Picpoul' 'Savatiano'
 'Johannisberg Riesling' 'Ruché' 'Jaen' 'Nero di Troia'
 'Roussanne-Viognier' 'Tempranillo Blanco' 'Dafni' 'Greco Bianco' 'Robola'
 'Siria' 'Mtsvane' 'Doña Blanca' 'Aragonês' 'Negrette' 'Macabeo-Moscatel'
 'Portuguiser' 'Coda di Volpe' 'Sauvignon Blanc-Verdejo' 'Portuguese Rosé'
 'Cabernet Sauvignon and Tinta Roriz' 'Black Monukka' 'Rabigato'
 'Feteasca' 'Albana' 'Malbec-Bonarda' 'Magliocco' 'Trebbiano-Malvasia'
 'Pignoletto' 'Moscatel Roxo' 'Durif' 'Garnacha-Syrah' 'Parraleta'
 'Mondeuse' 'Malagousia' 'Roter Veltliner' 'Cococciola' 'Moscatel Graúdo'
 'Aleatico' 'Chambourcin' 'Kadarka' 'Gouveio' 'Terret Blanc'
 'Roussanne-Grenache Blanc' 'Marsanne-Viognier' 'Pinot Auxerrois'
 'Cabernet Franc-Cabernet Sauvignon' 'Zierfandler' 'Tannat-Cabernet Franc'
 'Chasselas' 'Mavroudi' 'Enantio' 'Karasakiz'
 'Sauvignon Blanc-Sauvignon Gris' 'Incrocio Manzoni' 'Malvazija'
 'Pansa Blanca' 'Bukettraube' 'Malagouzia' 'Morio Muskat' 'Premsal'
 'Syrah-Mourvèdre' 'Gewürztraminer-Riesling' 'Schwartzriesling'
 'Kalecik Karasi' 'Mansois']
Сортов вина: 420
```

In [17]:

```
a = df.groupby(['country'])['price'].agg(['mean']).sort_values(by='mean',ascending
a
```

Out[17]:

country	mean
Hungary	60.911765
US-France	50.000000
Canada	44.769231
France	42.324246
Italy	36.339586
Luxembourg	36.000000
Germany	35.565460
US	33.534740
Egypt	33.000000
Lebanon	32.500000
Austria	31.947494
Israel	31.735632
Australia	30.627010
Turkey	29.250000
Portugal	28.800266
Spain	28.013761
China	27.000000
Slovenia	25.642857
Mexico	25.571429
New Zealand	25.103139
Brazil	24.666667
Argentina	22.588556
Croatia	21.772727
Uruguay	21.666667
South Africa	21.506579
Greece	21.133929
Chile	19.715385
Switzerland	19.000000
Cyprus	16.666667
Serbia	16.500000
Georgia	16.000000
Moldova	15.062500
Ukraine	13.000000
Romania	12.500000
Bosnia and Herzegovina	12.000000

	mean
country	
South Korea	11.000000
Bulgaria	10.600000
Montenegro	10.000000

Добавим новый столбец "Континенты":

In [18]:
```
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'
    }
```

In [19]:
```
df['continent']=df['country'].map(country_to_continent)
df.head()
```

Out[19]:

	country	description	designation	points	price	province	region_1	region_2	variety	
0	US	With a delicate, silky mouthfeel and bright ac...	Unknown	86	23	California	Central Coast	Central Coast	Pinot Noir	Ma
1	Italy	D'Alceo is a drop dead gorgeous wine that ooze...	D'Alceo	96	275	Tuscany	Toscana	Unknown	Red Blend	R
2	France	The great dominance of Cabernet Sauvignon in t...	Unknown	91	40	Bordeaux	Haut-Médoc	Unknown	Bordeaux-style Red Blend	Ber
3	Italy	The modest cherry, dark berry and black tea no...	Unknown	81	15	Tuscany	Chianti Classico	Unknown	Sangiovese	
4	US	Exceedingly light in color, scent and flavor, ...	Unknown	83	25	Oregon	Rogue Valley	Southern Oregon	Pinot Noir	De

Заодно добавим небольшой словарь по цвету вина:

In [20]:

```
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"
}
```

In [21]:
```
df['color_wine']=df['variety'].map(color)
df.head()
```

Out[21]:

	country	description	designation	points	price	province	region_1	region_2	variety	
0	US	With a delicate, silky mouthfeel and bright ac...	Unknown	86	23	California	Central Coast	Central Coast	Pinot Noir	Ma
1	Italy	D'Alceo is a drop dead gorgeous wine that ooze...	D'Alceo	96	275	Tuscany	Toscana	Unknown	Red Blend	R
2	France	The great dominance of Cabernet Sauvignon in t...	Unknown	91	40	Bordeaux	Haut-Médoc	Unknown	Bordeaux-style Red Blend	Be
3	Italy	The modest cherry, dark berry and black tea no...	Unknown	81	15	Tuscany	Chianti Classico	Unknown	Sangiovese	
4	US	Exceedingly light in color, scent and flavor, ...	Unknown	83	25	Oregon	Rogue Valley	Southern Oregon	Pinot Noir	De

Проверим новые колонки на наличие пустых значений:

In [22]:
```
df.isnull().sum()
```

Out[22]:
```
country           0
description       0
designation       0
points            0
price             0
province          0
region_1          0
region_2          0
variety           0
winery            0
continent         3
color_wine     3501
dtype: int64
```

Пропущенные значения в новых колонке Цвета так же заменим на Unknown, чтобы не портить общую картину. В свою очередь, значения с тремя пустыми континентами можно удалить без опасения нарушения общего баланса.

In [23]:
```
df['color_wine'].fillna('other', inplace=True)
df.dropna(subset=['continent'], inplace=True)
df.isnull().sum()
```

Out[23]:
```
country        0
description    0
designation    0
points         0
price          0
province       0
region_1       0
region_2       0
variety        0
winery         0
continent      0
color_wine     0
dtype: int64
```

После проведенных работ по предобработке данных можно приступать к полноценному исследовательскому анализу.

Выводы по Шагу 1 - Предобработка данных:

- В датасете представлены данные о 38 странах-производителях, 420 сортах вина;
- Количественных показателей в датасете всего два: Цена и Баллы (рейтинг). В стобце Цена пропущего около 9% данных, которые были заменены на среднее значение по данному столбцу;
- Много пропущенных данных по Регионам выращивания винограда, названий виноградников (30-60%). Пропуски были заменены на значение 'Unknown' для дальнейшего удобства в работе;
- Тип данных столбца Цена был изменен с 'float64' на 'int64';
- Добавлены два небольших словаря "Континенты" и "Цвет вина" для расширенного анализа;

2. Исследовательский анализ данных

Общее распределение показателей рейтинга:

In [24]:
```
plt.figure(figsize = (10,5))
plt.hist(df.points.to_list(),color='navy')
plt.title('Распределение вин по рейтингу')
plt.xlabel('Рейтинг')
plt.ylabel('Доля')
plt.grid()
plt.figtext(0.6, -0.01, "Рисунок 2. - Распределение вин по рейтингу", fontsize =10
```

Out[24]:
```
Text(0.6, -0.01, 'Рисунок 2. - Распределение вин по рейтингу')
```

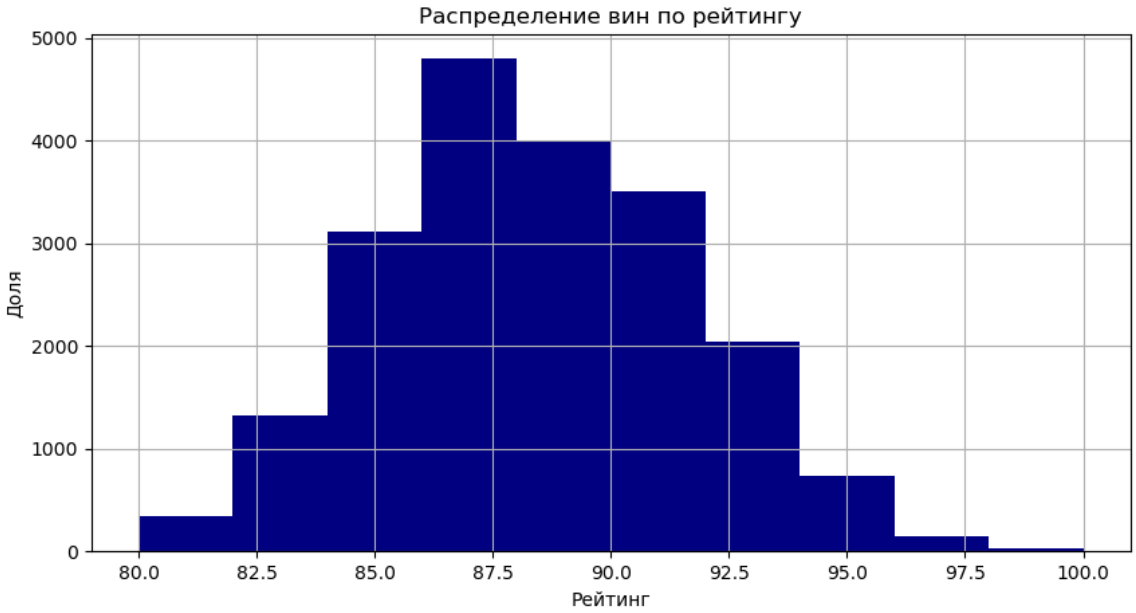


Рисунок 2. - Распределение вин по рейтингу

Другая визуализация Баллов:

In [25]:
```
df['points'].value_counts().sort_index().plot.bar()
plt.xlabel('Рейтинг', fontsize=7)
plt.ylabel('Доля', fontsize=7)
plt.figtext(0.45, -0.04, "Рисунок 3. - Распределение вин по рейтингу", fontsize =9
```

Out[25]:
```
Text(0.45, -0.04, 'Рисунок 3. - Распределение вин по рейтингу')
```

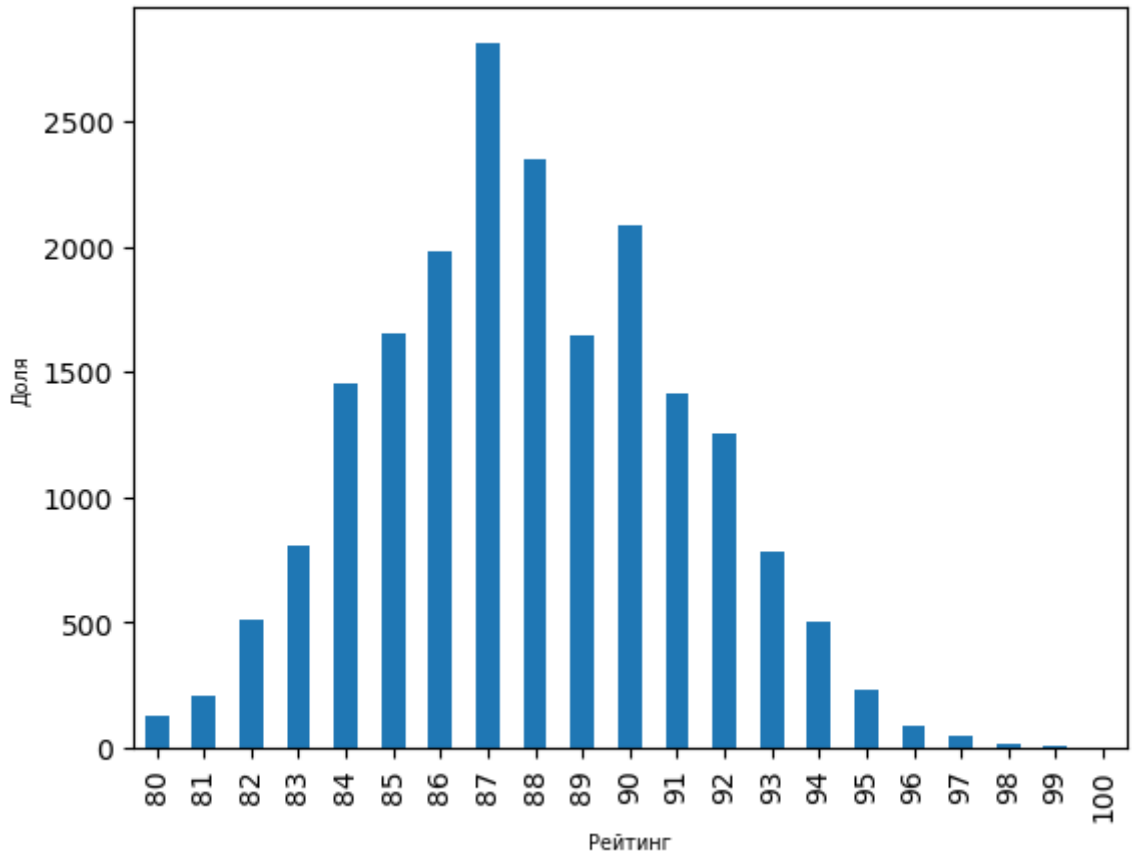


Рисунок 3. - Распределение вин по рейтингу

Как видно из гистограмм, на первый взгляд распределение баллов соответствует нормальному распределению показателей.

Определяем, какие сорта лидируют по рейтингам:

In [26]:
```
# по количеству оценок
df['variety'].value_counts().head(10)
```

Out[26]:
```
Pinot Noir                  1945
Chardonnay                  1893
Cabernet Sauvignon          1636
Red Blend                   1329
Bordeaux-style Red Blend     952
Sauvignon Blanc              848
Syrah                        779
Riesling                     747
Merlot                       654
Zinfandel                    517
Name: variety, dtype: int64
```

In [27]:
```
# по сумме баллов
plt.figure(figsize=(7,5))
w = df.groupby(df['variety'])['points'].agg(['sum']).sort_values(by='sum',ascendin
sns.barplot(x= w['sum'], y = w.index, data = w, orient='h')
plt.xlabel('Сумма баллов',fontsize=8)
plt.ylabel('Сорт',fontsize=8)
plt.figtext(0.5, -0.01, "Рисунок 4 - Самые популярные вина", fontsize =10)
```

Out[27]:
```
Text(0.5, -0.01, 'Рисунок 4 - Самые популярные вина')
```

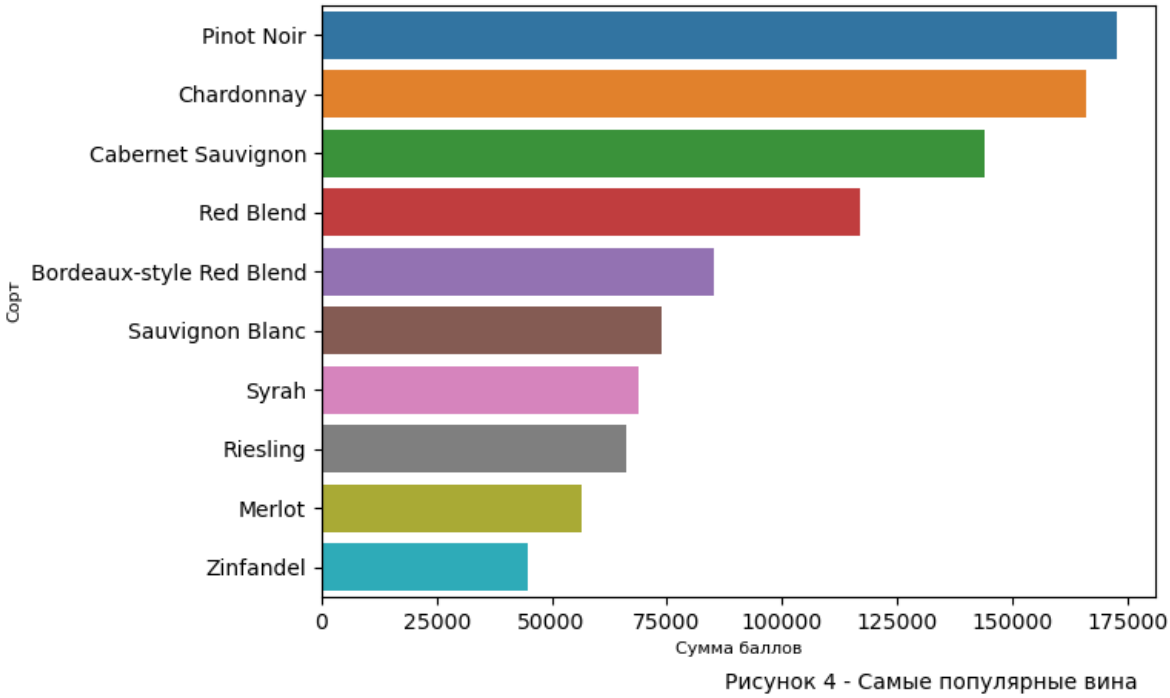


Рисунок 4 - Самые популярные вина

Список рейтинга вин получается одинаковым, что по количеству оценок, что по сумме баллов.

Самые дорогие сорта вин:

In [28]:
```
v = df.groupby(['variety'])['price'].agg(['sum']).sort_values(by='sum',ascending=Fa
v
```

Out[28]:

variety	sum
Pinot Noir	84349
Cabernet Sauvignon	69194
Chardonnay	61033
Red Blend	46825
Bordeaux-style Red Blend	41795
Syrah	28212
Riesling	21960
Merlot	17748
Nebbiolo	16974
Sauvignon Blanc	16560

In [29]:
```
plt.figure(figsize=(7,5))
w = df.groupby(df['variety'])['price'].agg(['sum']).sort_values(by='sum',ascending
sns.barplot(x= w['sum'], y = w.index, data = w, orient='h')
plt.xlabel('Сумма баллов',fontsize=8)
plt.ylabel('Сорт',fontsize=8)
plt.figtext(0.5, -0.01, "Рисунок 5 - Самые дорогие сорта вин", fontsize =10)
```

Out[29]:
```
Text(0.5, -0.01, 'Рисунок 5 - Самые дорогие сорта вин')
```

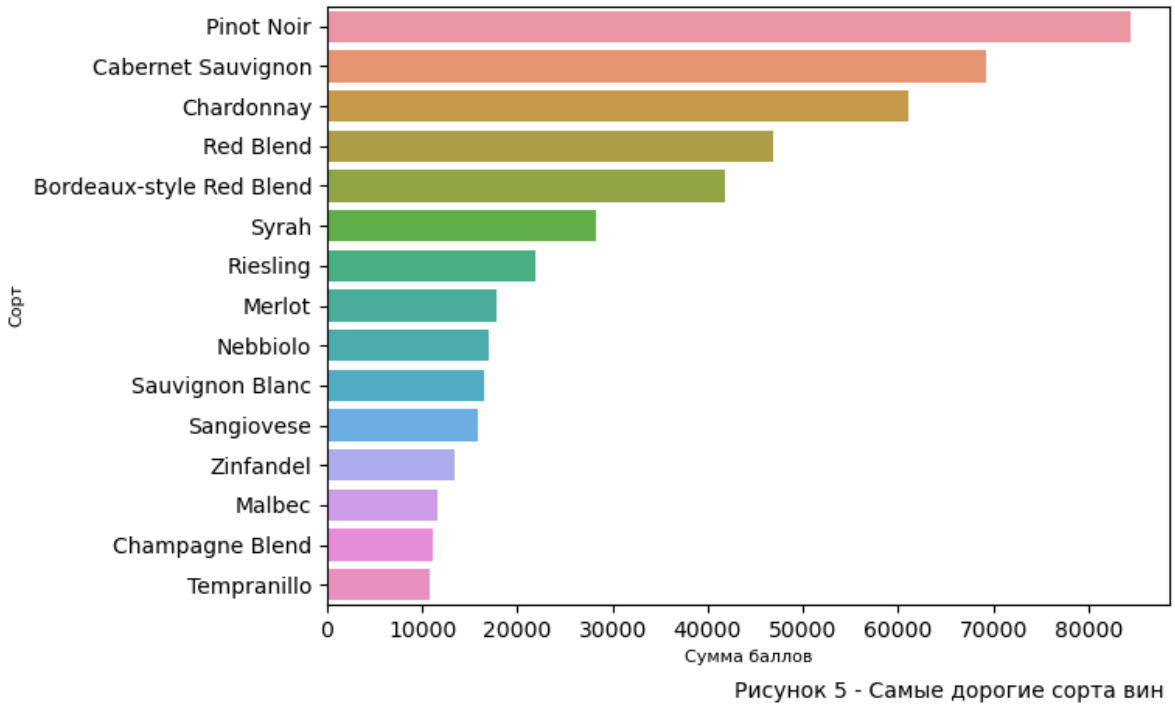


Рисунок 5 - Самые дорогие сорта вин

Как видно из рисунков, сорта Pinot Noir, Cabernet Sauvignon и Chardonnay являются как самыми популярными (судя по оценкам пользователей), так и самыми дорогими сортами вин.

Для того, чтобы посмотреть, какое количество выбросов присутствует в данных построим график «ящик с усами» по рейтингам в разбивке по сортам вин:

In [30]:
```
df1= df[df.variety.isin(df.variety.value_counts().head(15).index)]
sns.boxplot(x ='points',y ='variety',data = df1)
plt.xlabel('Рейтинг',fontsize=8)
plt.ylabel('Сорта вин',fontsize=8)
plt.figtext(0.48, -0.01, "Рисунок 6. - Выбросы по сортам вина", fontsize =10)
```

Out[30]:
```
Text(0.48, -0.01, 'Рисунок 6. - Выбросы по сортам вина')
```

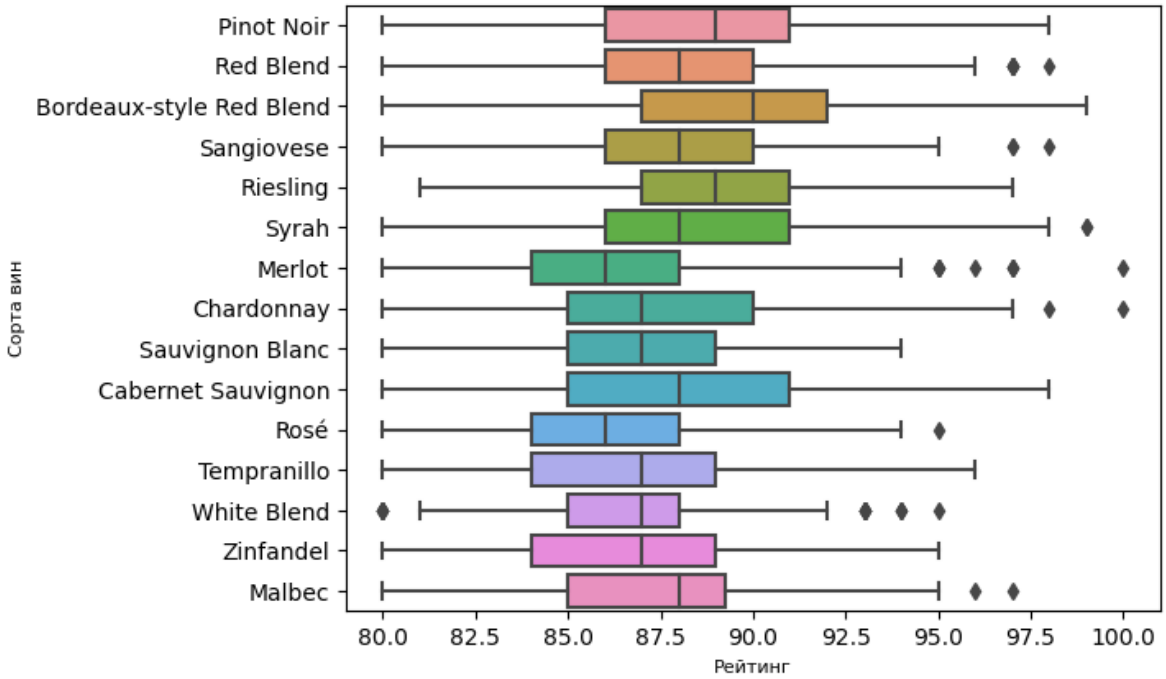


Рисунок 6. - Выбросы по сортам вина

Поскольку распределение данных по баллам является нормальным, то и выбросов присутствует достаточно умеренное количество. Лишь малый процент пользователей дал нестандартную оценку тому или иному сорту вина.

Построим график «ящик с усами» по рейтингам в разбивке по странам:

In [31]:
```
fig, ax = plt.subplots(figsize = (20,7))
chart = sns.boxplot(x='country',y='points', data=df, ax = ax)
plt.xticks(rotation = 90)
plt.xlabel('Рейтинг',fontsize=12)
plt.ylabel('Страны',fontsize=12)
plt.figtext(0.7, -0.2, "Рисунок 7. - Выбросы по странам", fontsize =17)
```

Out[31]:
```
Text(0.7, -0.2, 'Рисунок 7. - Выбросы по странам')
```

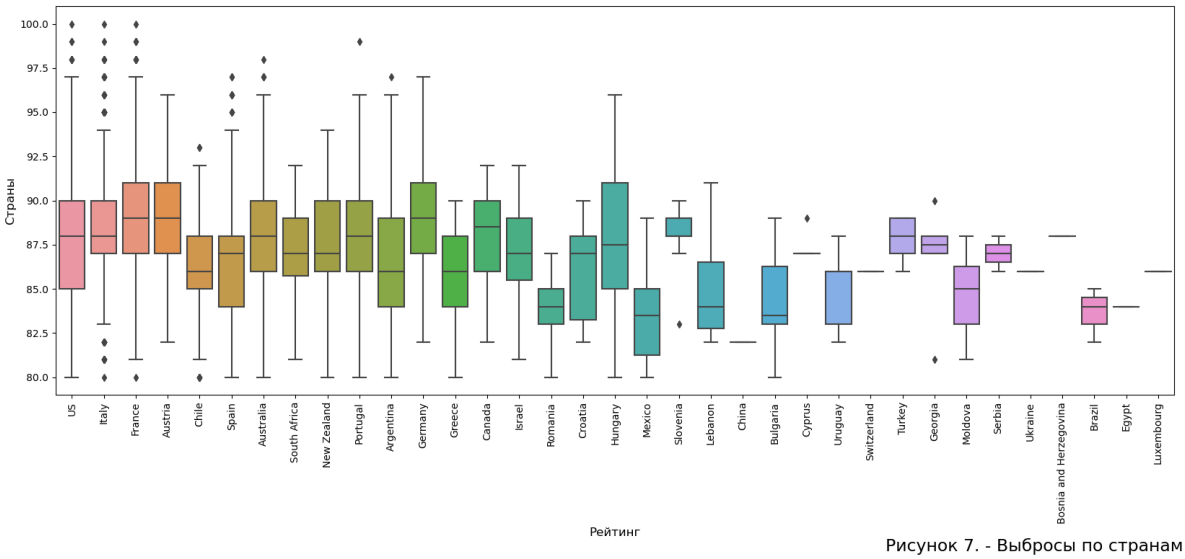


Рисунок 7. - Выбросы по странам

На Рисунке 7 можно заметить, что больше всего нестандартных оценок дают итальянскому вину, в то время как вина других европейских стран оценивают нормально.

Самые популярные страны по средней оценке баллов:

In [32]:
```
w = df.groupby(['country']).agg({'points':'sum'}).sort_values(by='points',ascending
w.reset_index(inplace=True)
w.style.background_gradient(cmap='coolwarm', high=0.5)
```

Out[32]:

	country	points
0	US	724284
1	Italy	273316
2	France	244781
3	Spain	94332
4	Chile	67331
5	Portugal	66177
6	Argentina	63282
7	Australia	54722
8	New Zealand	39076
9	Austria	37331

Нарисуем круговую диаграмму с распределением производства вин по странам:

In [33]:
```
df['country'].value_counts().head(10).plot(kind='pie',
                                           figsize=(7,7),
                                           ylabel=' ',
                                           autopct = '%.2f%%',
                                            textprops = {'size' : 'small','fontweig
                                                         'rotation' : 0})
plt.figtext(0.65, 0.11, "Рисунок 8. - Распределение вин по странам", fontsize=10)
```

Out[33]:
```
Text(0.65, 0.11, 'Рисунок 8. - Распределение вин по странам')
```

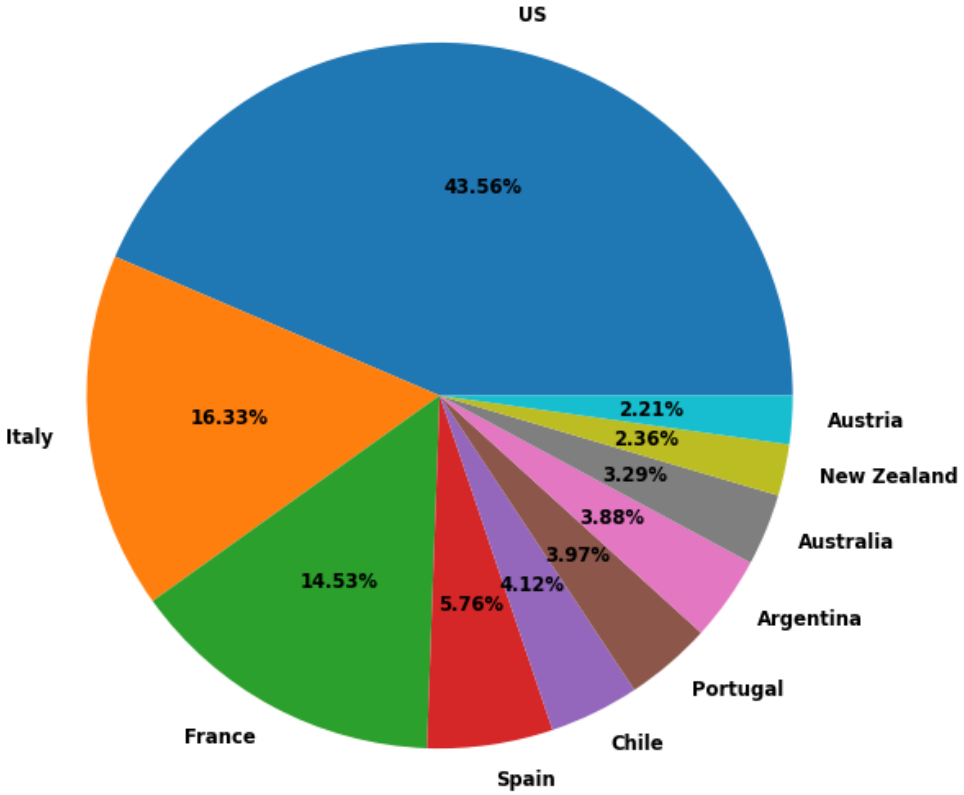


Рисунок 8. - Распределение вин по странам

В масштабе данного набора данных можно понять, что основным производителем вин являются США, на втором и третием местах с большим отрывом идут Италия и Франция соответственно.

Немного углубимся и посмотрим, какие провинции лидируют в выпуске вин по количеству:

In [34]:
```
a=df.groupby('province')['variety'].count()
a.sort_values(ascending=False).head(10).plot(kind = 'bar',figsize=(6, 5))
plt.xlabel('')
plt.ylabel('')
plt.figtext(0.3, -0.2, "Рисунок 9. - Провинции-лидеры по выпуску вина", fontsize=1
sns.despine()
```

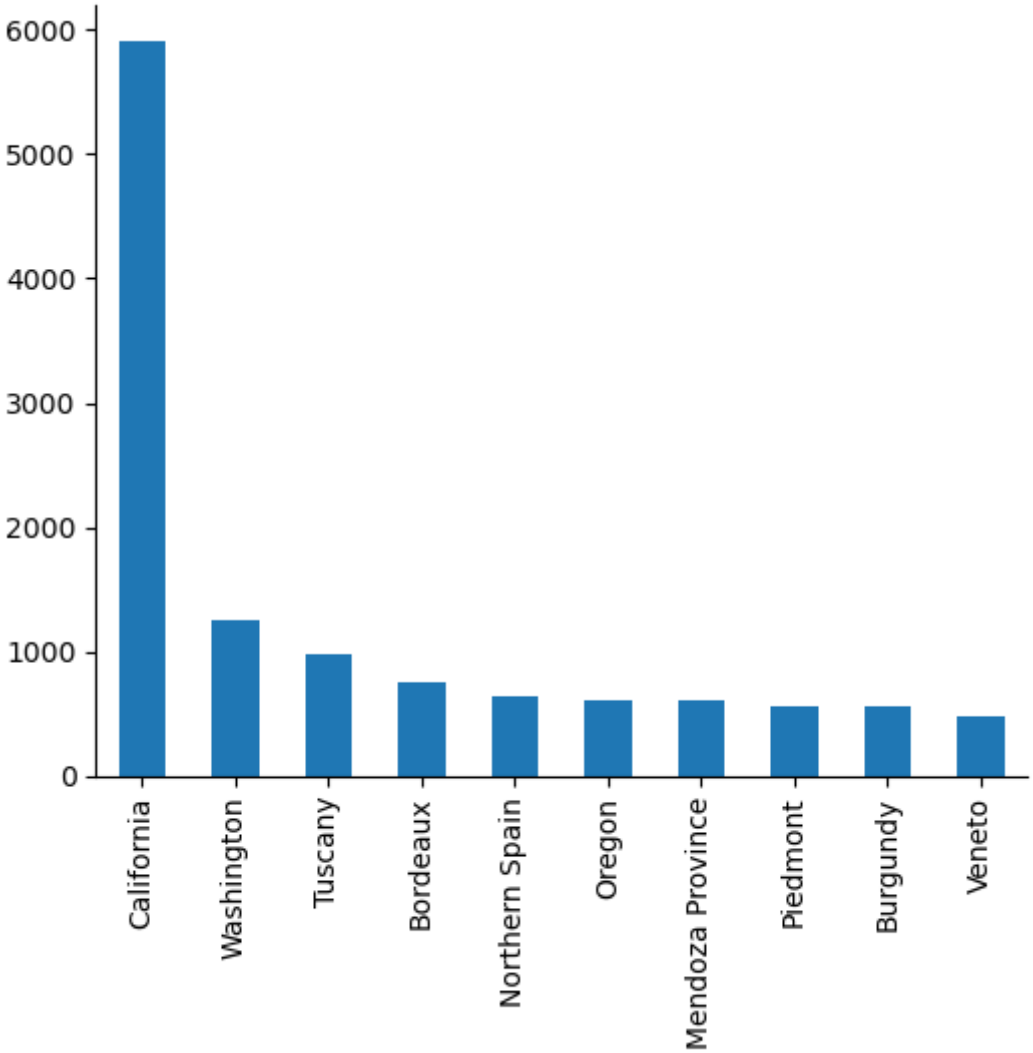


Рисунок 9. - Провинции-лидеры по выпуску вина

California лидирует с большим отрывом, что не удивительно, ведь б0льшую часть датасета представляют вина Америки.

Посмотрим, какие винодельни производят самое популярное вино:

In [35]:
```
w = df.groupby(['country','province','winery','variety','points'])['price'].agg(['
w.reset_index(inplace=True)
w.style.background_gradient(cmap='coolwarm', high=0.5)
```

Out[35]:

	country	province	winery	variety	points	count	min	max	mean
0	US	California	Sloan	Cabernet Blend	100	1	245	245	245.000000
1	Italy	Tuscany	Tenuta dell'Ornellaia	Merlot	100	1	460	460	460.000000
2	France	Champagne	Krug	Chardonnay	100	1	1400	1400	1400.000000
3	US	Oregon	Cayuse	Syrah	99	2	65	65	65.000000
4	France	Bordeaux	Château Latour	Bordeaux-style Red Blend	99	1	2300	2300	2300.000000
5	France	Bordeaux	Château d'Yquem	Bordeaux-style White Blend	99	1	33	33	33.000000
6	Portugal	Douro	Casa Ferreirinha	Portuguese Red	99	1	426	426	426.000000
7	Italy	Piedmont	Mascarello Giuseppe e Figlio	Nebbiolo	99	1	175	175	175.000000
8	France	Bordeaux	Château Palmer	Bordeaux-style Red Blend	99	1	33	33	33.000000
9	Australia	Victoria	Campbells	Tokay	98	1	97	97	97.000000

Проверим тройку лидеров:

In [36]:
```
df[(df['points']==100)]
```

Out[36]:

	country	description	designation	points	price	province	region_1	region_2	v
323	France	A wine that has created its own universe. It h...	Clos du Mesnil	100	1400	Champagne	Champagne	Unknown	Chard
5955	Italy	A perfect wine from a classic vintage, the 200...	Masseto	100	460	Tuscany	Toscana	Unknown	
17967	US	Impossibly aromatic. Hard to imagine greater c...	Red Wine	100	245	California	Rutherford	Napa	Cal

In [37]:
```
r = df[df.points == df.points.max()]
print(f"Наилучшие рейтинги получают вина областей:{r.province.tolist()}")
```

```
Наилучшие рейтинги получают вина областей:['Champagne', 'Tuscany', 'California']
```

Не смотря на то, что Американским вин достаточно много (почти половина датасета), в тройку лидеров вошли Италия и Франция. Стоит отметить, что стоимость вин с самым

высоким рейтингом сильно варьируется.

In [38]:
```
w = df.groupby(['country','province','winery','variety','points'])['price'].agg(['
w.reset_index(inplace=True)
w.style.background_gradient(cmap='coolwarm', high=0.5)
```

Out[38]:

	country	province	winery	variety	points	count	min	max	mean
0	New Zealand	Awatere Valley	The Crossings	Pinot Noir	80	1	19	19	19.000000
1	US	California	California's Jewel	Zinfandel	80	1	10	10	10.000000
2	US	California	Candor	Zinfandel	80	1	18	18	18.000000
3	Portugal	Alentejano	Cartuxa	Portuguese Red	80	1	33	33	33.000000
4	Argentina	Mendoza Province	Hat in the Ring	Pinot Grigio	80	2	10	10	10.000000
5	US	California	Terremoto Cellars	Cabernet Sauvignon	80	2	35	35	35.000000
6	New Zealand	Marlborough	Saint Clair	Pinot Noir	80	2	16	16	16.000000
7	US	California	Carivintas	Tempranillo	80	1	25	25	25.000000
8	US	California	TWODOG	Zinfandel	80	1	11	11	11.000000
9	US	California	CK Mondavi	Zinfandel	80	1	8	8	8.000000

Самый низкий рейтинг в основном получили винодельни, находящиеся в США, Аргентине и Северной Испании.

Самые дорогие вина нашего рейтинга были выпущены преимущественно во Франции:

In [39]:
```
w = df.groupby(['country','province','winery','variety','price'])['points'].agg(['
w.reset_index(inplace=True)
w.style.background_gradient(cmap='coolwarm', high=0.5)
```

Out[39]:

	country	province	winery	variety	price	count	min	max	mean
0	France	Bordeaux	Château Latour	Bordeaux-style Red Blend	2300	1	99	99	99.000000
1	France	Champagne	Krug	Chardonnay	1400	1	100	100	100.000000
2	Austria	Wachau	Emmerich Knoll	Grüner Veltliner	1100	1	94	94	94.000000
3	France	Bordeaux	Château Ausone	Bordeaux-style Red Blend	850	1	95	95	95.000000
4	Hungary	Tokaji	Royal Tokaji	Furmint	764	1	94	94	94.000000
5	France	Burgundy	Bouchard Père & Fils	Chardonnay	757	1	98	98	98.000000
6	Spain	Northern Spain	García Figuero	Tempranillo	599	1	93	93	93.000000
7	France	Burgundy	Joseph Drouhin	Chardonnay	596	1	96	96	96.000000
8	France	Bordeaux	Château Margaux	Bordeaux-style Red Blend	550	1	95	95	95.000000
9	Australia	South Australia	Henschke	Shiraz	550	1	96	96	96.000000

Посмотрим, что со средними ценами на вино по странам:

In [40]:
```
mean_wine_price = df.groupby(['country','province']).agg({'price':'mean'}).sort_va
#mean_wine_price.reset_index(inplace=True) - для построения графика индексы лучше н
mean_wine_price.style.background_gradient(cmap='coolwarm', high=0.5)
```

Out[40]:

		price
country	**province**	
Hungary	**Tokaji**	133.100000
Chile	**Santa Cruz**	95.000000
France	**Champagne**	86.513812
Croatia	**Middle and South Dalmatia**	65.000000
France	**Burgundy**	61.381206
Austria	**Wachau**	60.134615
New Zealand	**Martinborough Terrace**	60.000000

In [42]:
```
mean_wine_price.plot(kind = 'barh',figsize=(6, 5), color='green')
plt.xlabel('Средняя цена на вино',fontsize=8)
plt.ylabel('Страна, провинция',fontsize=8)
plt.figtext(0.25, -0.01, "Рисунок 10. - Средние цены на вино: верхний сегмент", fo
```

Out[42]:
```
Text(0.25, -0.01, 'Рисунок 10. - Средние цены на вино: верхний сегмент')
```

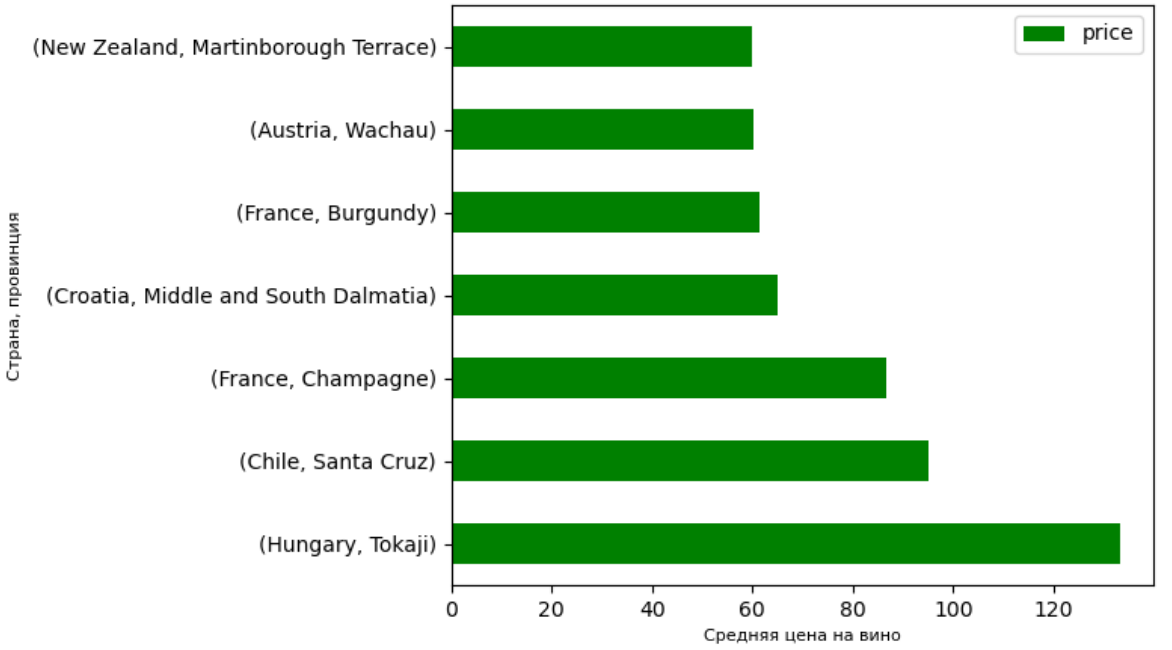


Рисунок 10. - Средние цены на вино: верхний сегмент

Совершенно неожиданно, но самые дорогие вина по средней оценке оказались в Венгрии.

Популярные сорта вина по странам:

In [43]:
```
wine_country = df.groupby(['country'])['variety','points'].agg('max').sort_values(
wine_country.style.background_gradient(cmap='coolwarm', high=0.5)
```

Out[43]:

country	variety	points
Argentina	White Blend	97
Australia	Zinfandel	98
Austria	Zweigelt	96
Bosnia and Herzegovina	Blatina	88
Brazil	Sparkling Blend	85
Bulgaria	Traminer	89
Canada	Viognier	92
Chile	White Blend	93
China	Chardonnay	82
Croatia	Zlahtina	90

Популярные сорта вина по провинциям:

In [44]:
```
wine_province = df.groupby(['province'])['variety','points'].agg('max').sort_value
wine_province
```

Out[44]:

province	variety	points
Champagne	Pinot Noir	100
Tuscany	White Blend	100
California	Zinfandel	100
Piedmont	White Blend	99
Bordeaux	Sémillon	99
...	...	...
Beotia	Roditis	81
Samson	Tempranillo	81
Table wine	Portuguese Red	81
San Antonio de las Minas Valley	Cinsault	80
Patras	Roditis	80

311 rows × 2 columns

Подробнее о рейтинге вин внутри каждой провинции:

In [45]:
```
df.pivot_table('variety', ['country','province','points'], aggfunc='max', sort=Tru
```

Out[45]:

country	province	points	variety
Argentina	**Mendoza Province**	**80**	Torrontés
		81	White Blend
		82	Torrontés
		83	Torrontés
		84	Viognier
...	**...**	**...**	...
Uruguay	**Juanico**	**82**	Red Blend
		86	Tannat
	Progreso	**88**	Tannat
	Uruguay	**82**	Pinot Noir
		83	Tannat

1583 rows × 1 columns

Проанализируем взаимосвязь количесвенных показателей: Цены и Баллов

In [46]:
```
df.plot(kind='scatter', x='price', y='points')
plt.xlabel('Цена',fontsize=8)
plt.ylabel('Баллы',fontsize=8)
plt.figtext(0.48, -0.01, "Рисунок 11. - Диаграмма рассеивания", fontsize=10)
```

Out[46]:
```
Text(0.48, -0.01, 'Рисунок 11. - Диаграмма рассеивания')
```

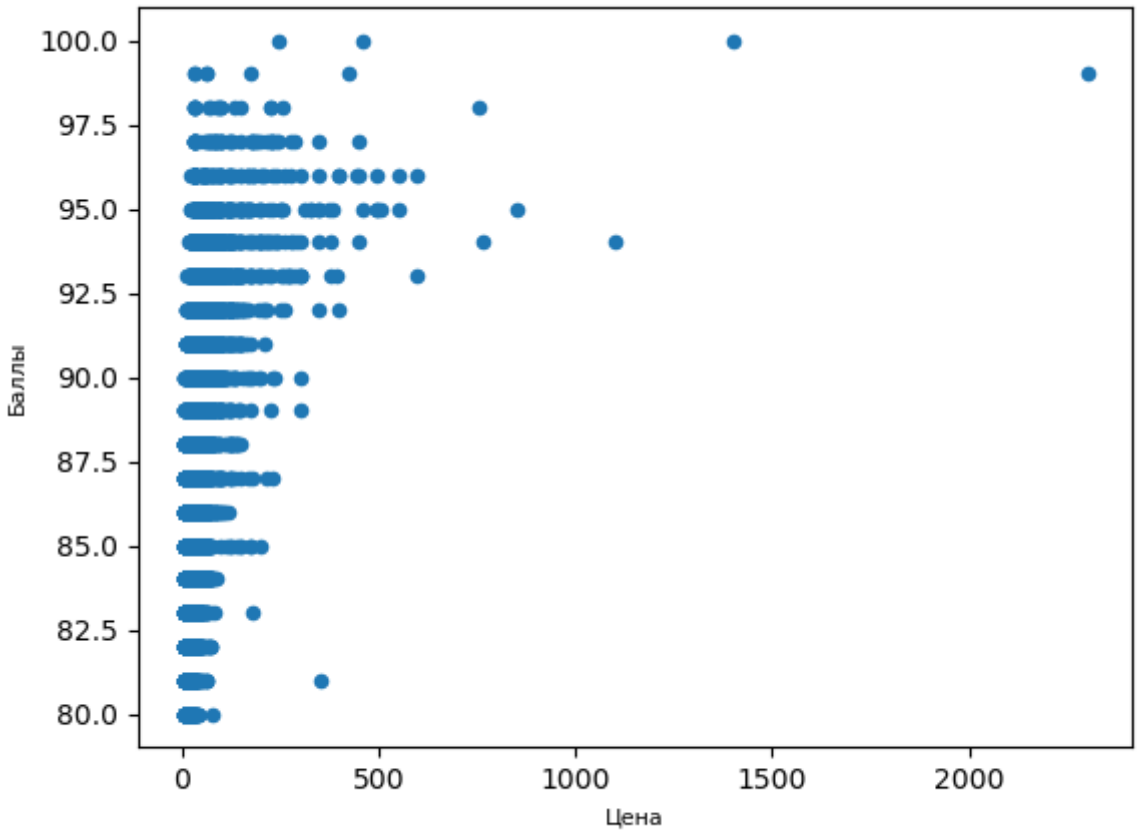


Рисунок 11. - Диаграмма рассеивания

Из диаграммы рассеяния видно, что большинство вин имеют неплохие оценки даже находясь в бюджетном сегменте. И не всегда дорогие вина - это хорошие вина.

Подробнее про корреляцию:

In [47]:
```
df.corr()
```

Out[47]:

	points	price
points	1.000000	0.408527
price	0.408527	1.000000

In [48]:
```
# Более наглядный вид - тепловая карта:
plt.figure(figsize=(5,2))
sns.heatmap(df.corr(), annot=True, fmt='.3f', cmap='turbo')
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.figtext(0.3, -0.07, "Рисунок 12. - Корреляционная матрица", fontsize=10)
```

Out[48]:
```
Text(0.3, -0.07, 'Рисунок 12. - Корреляционная матрица')
```

Рисунок 12. - Корреляционная матрица

Как видно из корреляционной матрицы влияние рейтинга (баллов) на стоимость вина прослеживается, но не сильно. Для более детальной работы необходимо кодировать нечисловые показатели датасета и пытаться проследить взаимосвязь более широкого ряда критериев.

При работе с этим датасетом я специально не удаляла выбросы, чтоб проследить общую тенденцию в реализации и оценках вина.

Разберемся с цветом вина:

In [49]:
```
plt.figure(figsize=(5,4))
df['color_wine'].value_counts().plot.bar()
plt.title('Количество вин по цветам', fontsize = 14)
plt.title('Цвет/тип вина')
plt.figtext(0.55, -0.07, "Рисунок 13. - Цвет вина", fontsize=10)
```

Out[49]:
```
Text(0.55, -0.07, 'Рисунок 13. - Цвет вина')
```

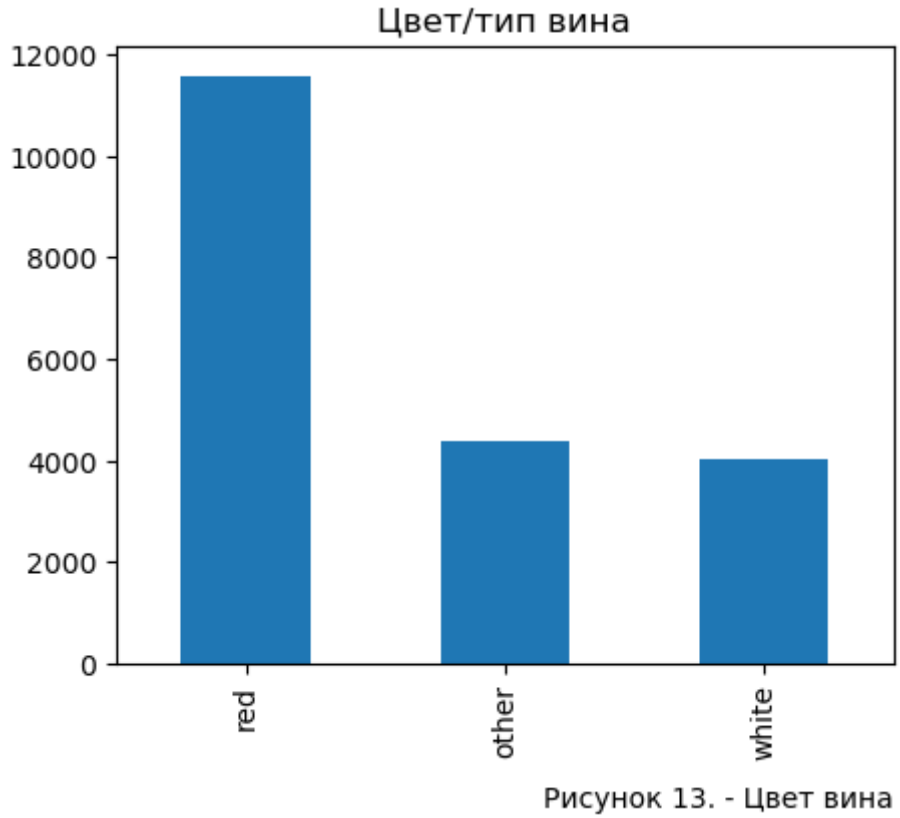
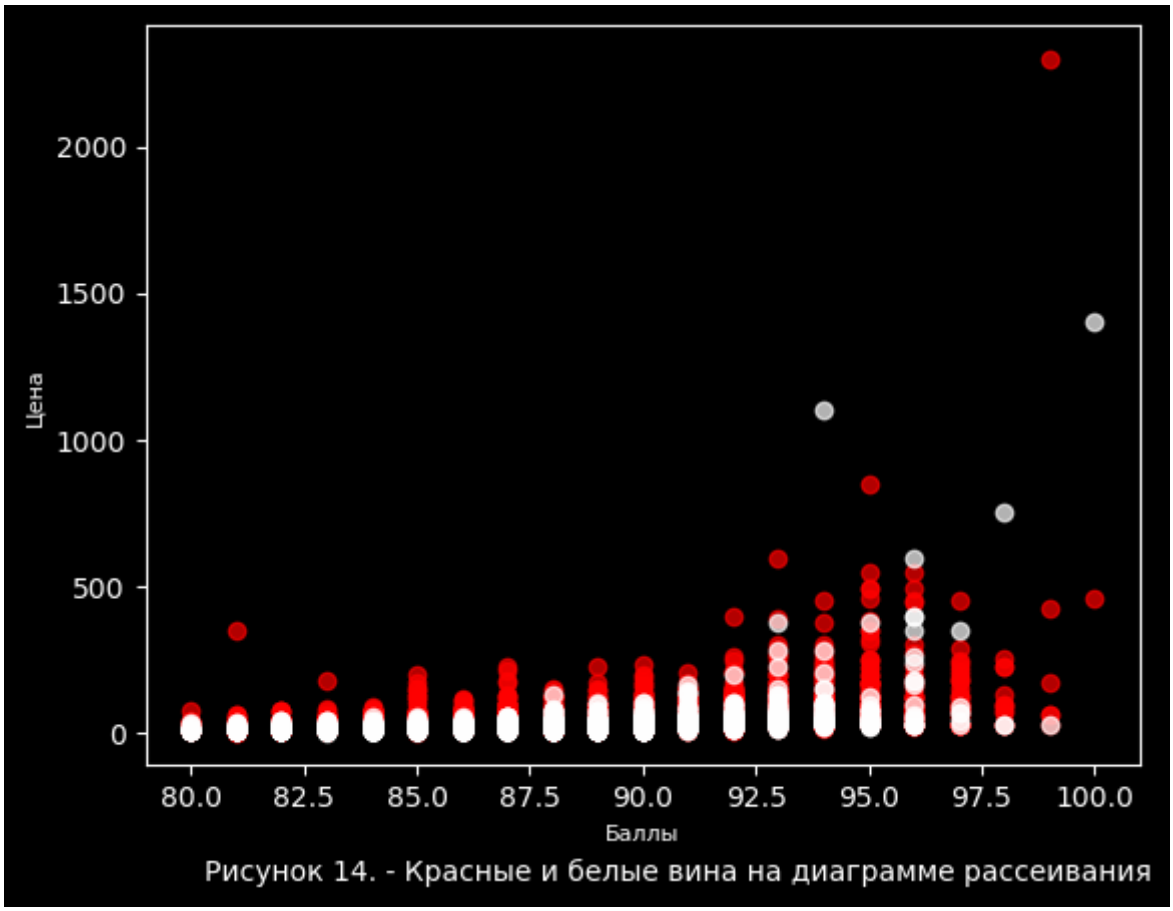


Рисунок 13. - Цвет вина

Сколько красного!

In [50]:
```
plt.style.use("dark_background")
plt.plot(df[df.color_wine=='red']['points'], df[df.color_wine=='red']['price'], 'o
plt.plot(df[df.color_wine=='white']['points'], df[df.color_wine=='white']['price']
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('Цена',fontsize=8)
plt.figtext(0.17, -0.01, "Рисунок 14. - Красные и белые вина на диаграмме рассеива
```

Out[50]:
```
Text(0.17, -0.01, 'Рисунок 14. - Красные и белые вина на диаграмме рассеивания')
```



Примерное понимание того, сколько вина и какого цвета есть в нашем датасете поможет для дальнейшей работы с гипотезами.

Выводы по Шагу 2 - Исследовательский анализ данных:

- По предварительной визуальной оценке данные в датасете распределены нормально;
- Сорта вин 'Pinot Noir', 'Chardonnay', 'Cabernet Sauvignon' являются как самыми популярными сортами, так и самыми дорогими. Напрашивается предварительный вывод о прямой зависимости Цена/Баллы;
- Выбросы по сортам вин и ко странам их производства присутствуют в достаточно умеренном количестве. Лишь малый количество пользователей дало нестандартную оценку тому или иному сорту вина, поэтому было принято решение от выбросов не избавляться;
- Больше всего нестандартных оценок дают итальянскому вину, в то время как вина других европейских стран оценивают нормально;
- Самыми крупными странами по производству вина являются США, Италия и Франция. Причем США занимает почти 44% рынка;

- Стоимость вин с высоким рейтингом сильно варьируется: присутствуют дешевые вина с отличным рейтингом;
- В перечне вин с низким рейтингом также лидирует США;
- Самые дорогие вина нашего рейтинга были выпущены преимущественно во Франции;
- Наиболее высокими в среднем являются цены на вино в Венгрии;
- Предварительный просмотр корреляции Цена/Баллы показывает достаточно слабую взаимосвязь. Это говорит о том, что не всегда большая цена на вино сопровождается высоким рейтингом и наоборот;
- В датасете преобладают красные вина;

3. Портрет пользователя

Рассмотрим собирательные образы потребителей вина по разным континентам.

In [51]:
```
plt.style.use('default')
(df
    .pivot_table(index='continent', values='points', aggfunc='sum')
    .plot(grid=True, kind='bar',figsize=(8, 4))
)
plt.xlabel('Континенты',fontsize=8)
plt.ylabel('Баллы',fontsize=8)
plt.figtext(0.43, -0.26, "Рисунок 15. - Популярность вин по континентам", fontsize
```

Out[51]:
```
Text(0.43, -0.26, 'Рисунок 15. - Популярность вин по континентам')
```

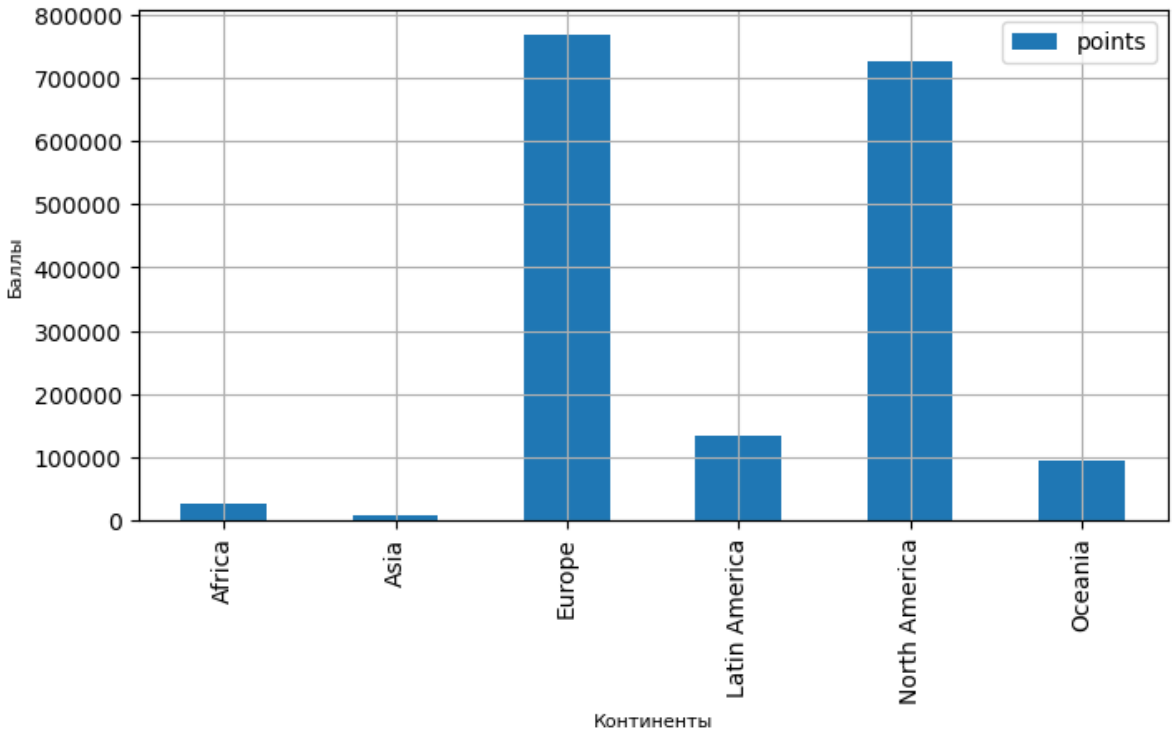


Рисунок 15. - Популярность вин по континентам

Северная Америка, в частности США судя по результатам исследования являются лидерами по производству вина, однако в потреблении (популярности) все же уступают жителям Европы.

Разберемся, на каком континенте какой сорт вина наиболее популярный:

In [52]:
```
(df
    .query('continent=="Africa"')
    .pivot_table(index='variety', values='points', aggfunc=sum, sort=False)
    .sort_values(by='points', ascending=True).head(20)
    .plot(grid=True, kind='barh',figsize=(7, 5), color='#e67300')
)
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('Сорт вина',fontsize=8)
plt.figtext(0.4, -0.01, "Рисунок 16. - Популярные сорта вин в Африке", fontsize=10

(df
    .query('continent=="Asia"')
    .pivot_table(index='variety', values='points', aggfunc=sum, sort=False)
    .sort_values(by='points', ascending=True).head(20)
    .plot(grid=True, kind='barh',figsize=(7, 5), color='#739900')
)
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('Сорт вина',fontsize=8)
plt.figtext(0.43, -0.01, "Рисунок 17. - Популярные сорта вин в Азии", fontsize=10)

(df
    .query('continent=="Europe"')
    .pivot_table(index='variety', values='points', aggfunc=sum, sort=False)
    .sort_values(by='points', ascending=True).head(20)
    .plot(grid=True, kind='barh',figsize=(7, 5), color='#0000b3')
)
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('Сорт вина',fontsize=8)
plt.figtext(0.4, -0.01, "Рисунок 18. - Популярные сорта вин в Европе", fontsize=10

(df
    .query('continent=="Latin America"')
    .pivot_table(index='variety', values='points', aggfunc=sum, sort=False)
    .sort_values(by='points', ascending=True).head(20)
    .plot(grid=True, kind='barh',figsize=(7, 5), color='#990033')
)
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('Сорт вина',fontsize=8)
plt.figtext(0.3, -0.01, "Рисунок 19. - Популярные сорта вин в Латинской Америке",

(df
    .query('continent=="North America"')
    .pivot_table(index='variety', values='points', aggfunc=sum, sort=False)
    .sort_values(by='points', ascending=True).head(20)
    .plot(grid=True, kind='barh',figsize=(7, 5), color='#52527a')
)
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('Сорт вина',fontsize=8)
plt.figtext(0.3, -0.01, "Рисунок 20. - Популярные сорта вин в Северной Америке", f

(df
    .query('continent=="Oceania"')
    .pivot_table(index='variety', values='points', aggfunc=sum, sort=False)
    .sort_values(by='points', ascending=True).head(20)
    .plot(grid=True, kind='barh',figsize=(7, 5), color='#00e6b8')
)
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('Сорт вина',fontsize=8)
plt.figtext(0.4, -0.01, "Рисунок 21. - Популярные сорта вин в Океании", fontsize=1
```

Out[52]:
```
Text(0.4, -0.01, 'Рисунок 21. - Популярные сорта вин в Океании')
```

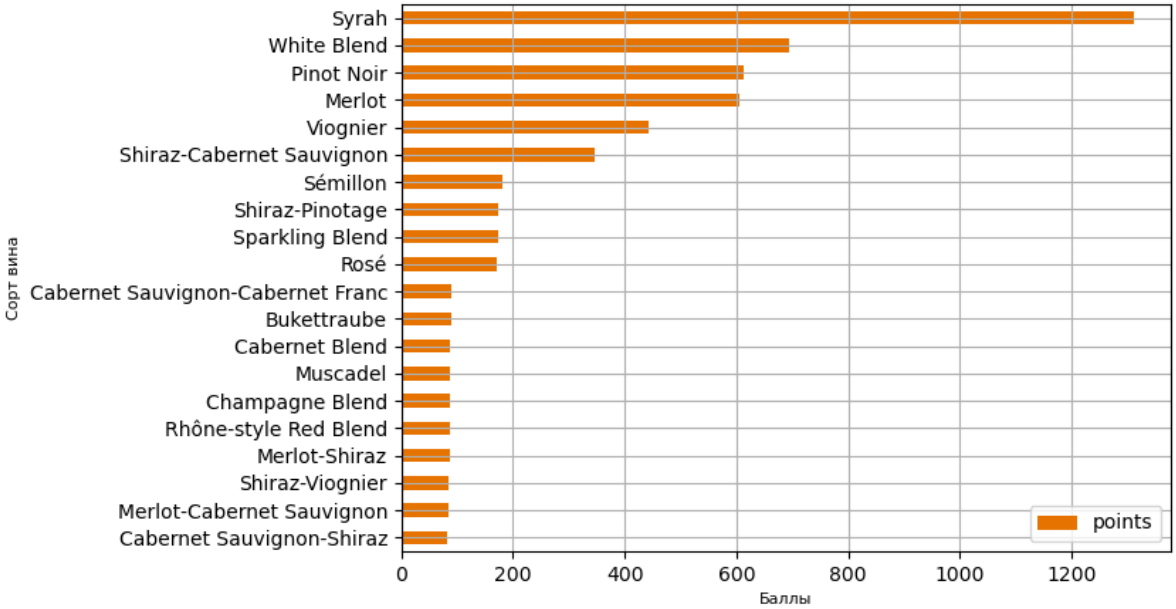


Рисунок 16. - Популярные сорта вин в Африке

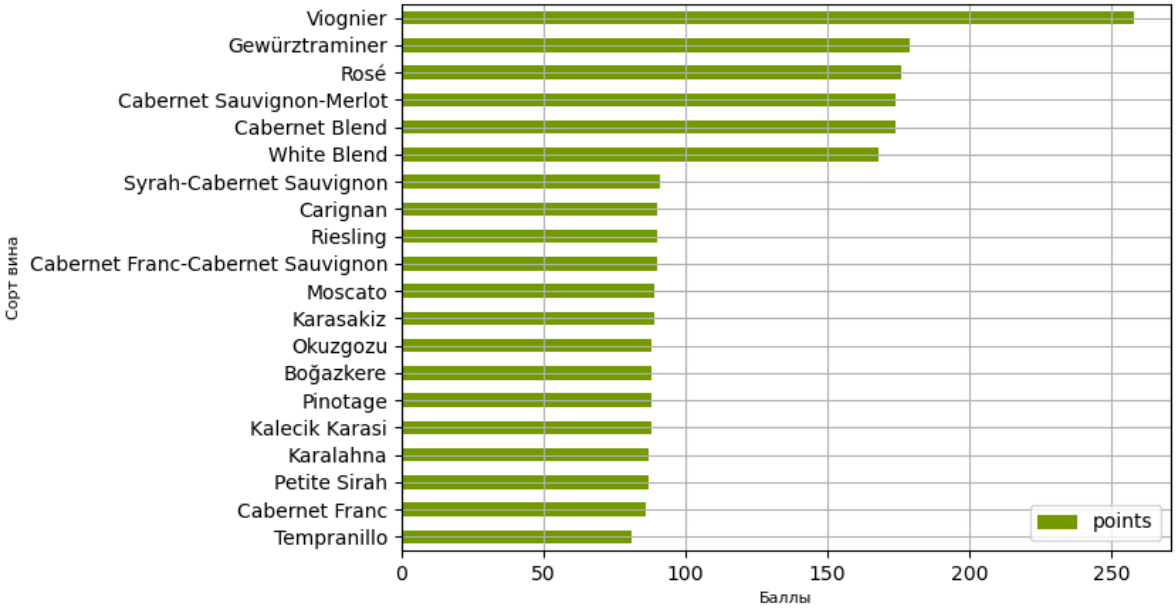


Рисунок 17. - Популярные сорта вин в Азии

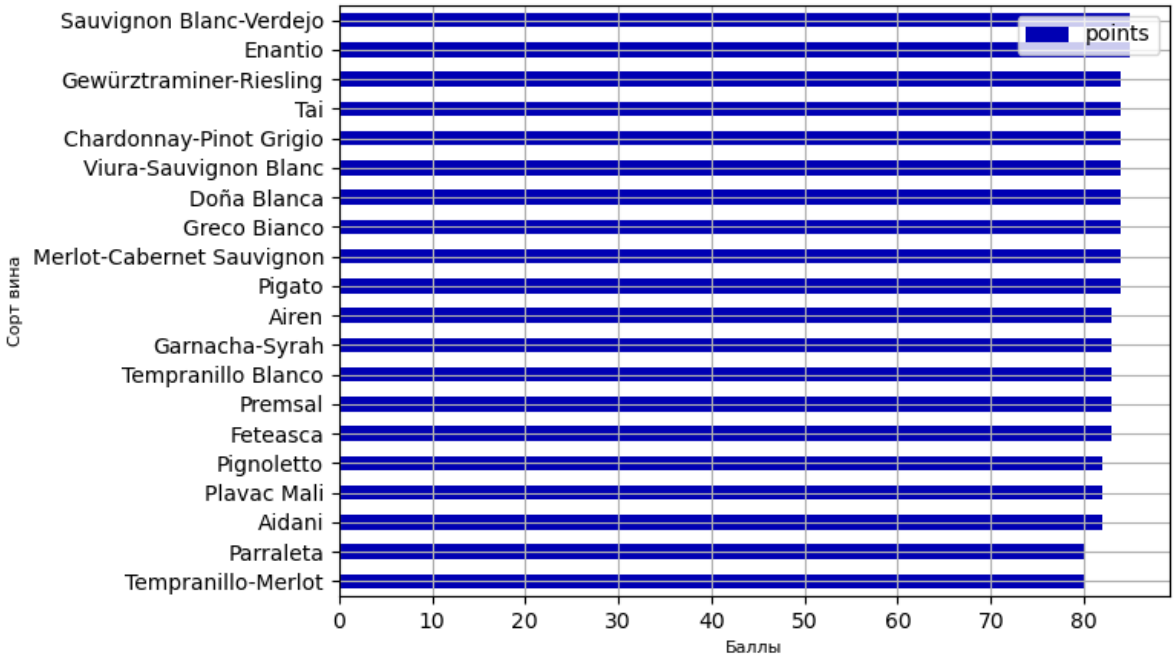


Рисунок 18. - Популярные сорта вин в Европе

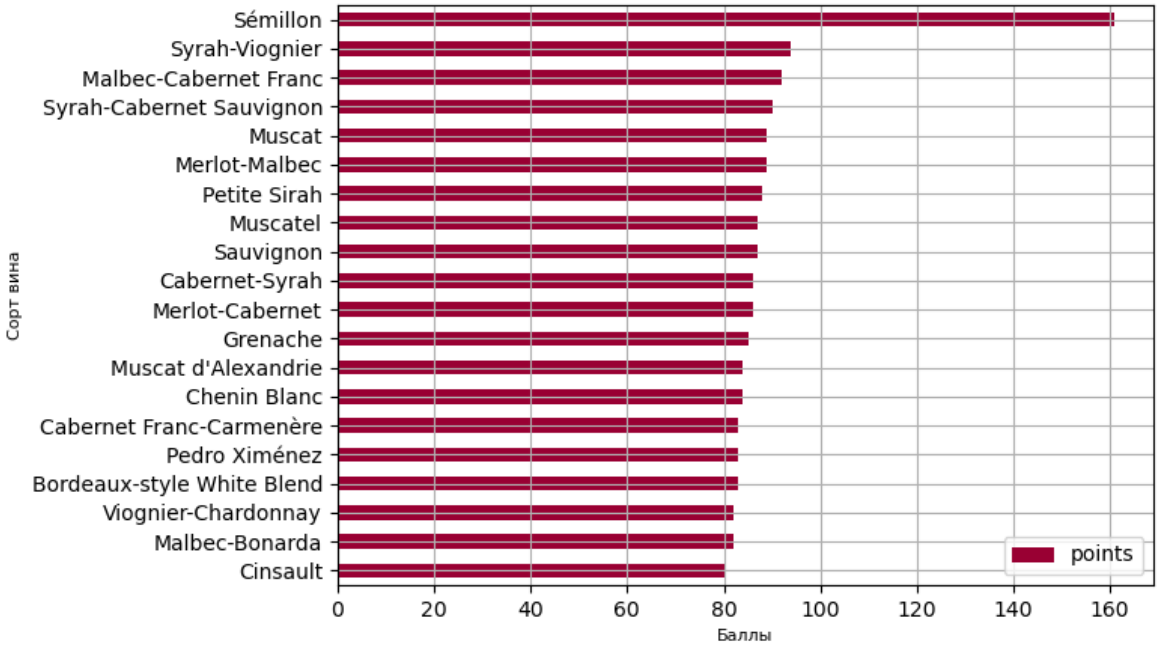


Рисунок 19. - Популярные сорта вин в Латинской Америке

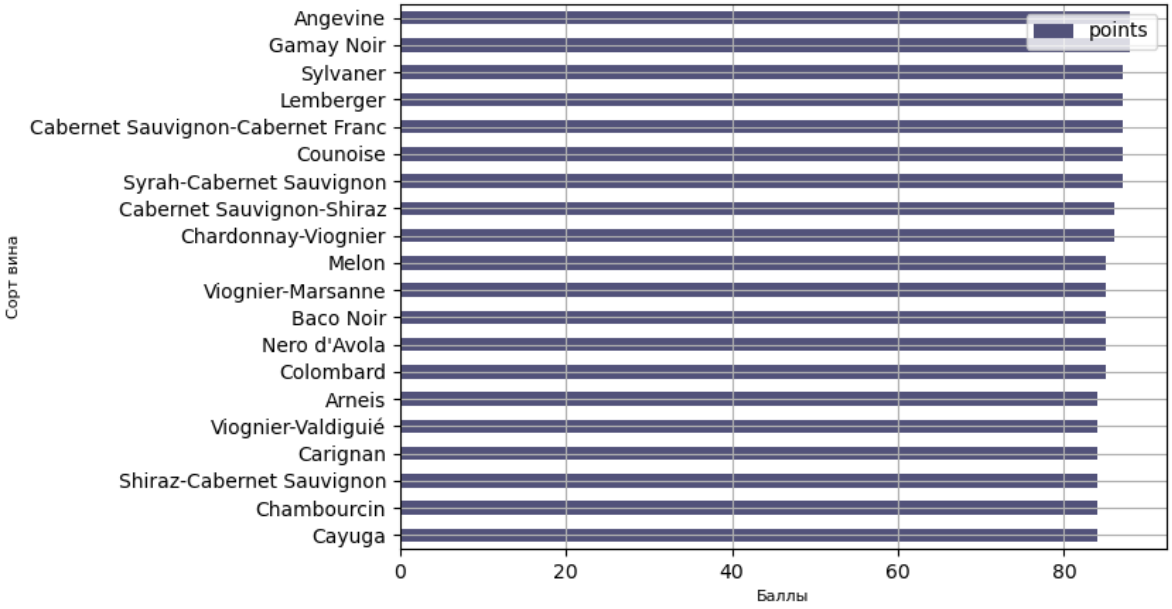


Рисунок 20. - Популярные сорта вин в Северной Америке

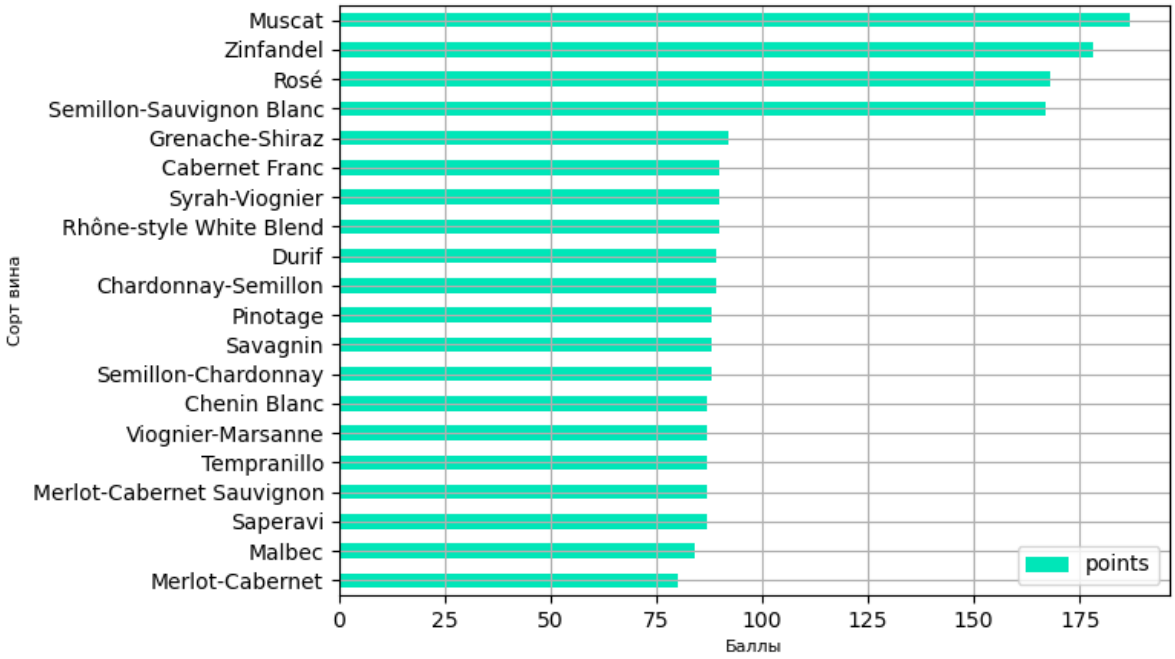


Рисунок 21. - Популярные сорта вин в Океании

Теперь рассмотрим влияние рейтинга цены по континентам:

In [53]:
```
africa=(df
    .query('continent=="Africa"')
    .pivot_table(values=['points','price'], index=['province','variety'], aggfunc=
    .groupby('province')['points','price'].max().sort_values(by='points',ascending
)
print(africa.head())
print(africa.corr())
```

```
                        points  price
province
Paarl                     91.5   58.0
Simonsberg-Stellenbosch   91.0   45.0
Franschhoek               91.0   39.0
Western Cape              90.0   42.0
Groenekloof               90.0   15.0
          points     price
points  1.000000  0.699503
price   0.699503  1.000000
```

In [54]:
```
africa.plot(kind='scatter', x='price', y='points')
plt.xlabel('Средняя цена',fontsize=8)
plt.ylabel('Баллы в среднем',fontsize=8)
plt.figtext(0.33, -0.01, "Рисунок 22. - Диаграмма рассеивания по Африке", fontsize
```

Out[54]:
```
Text(0.33, -0.01, 'Рисунок 22. - Диаграмма рассеивания по Африке')
```

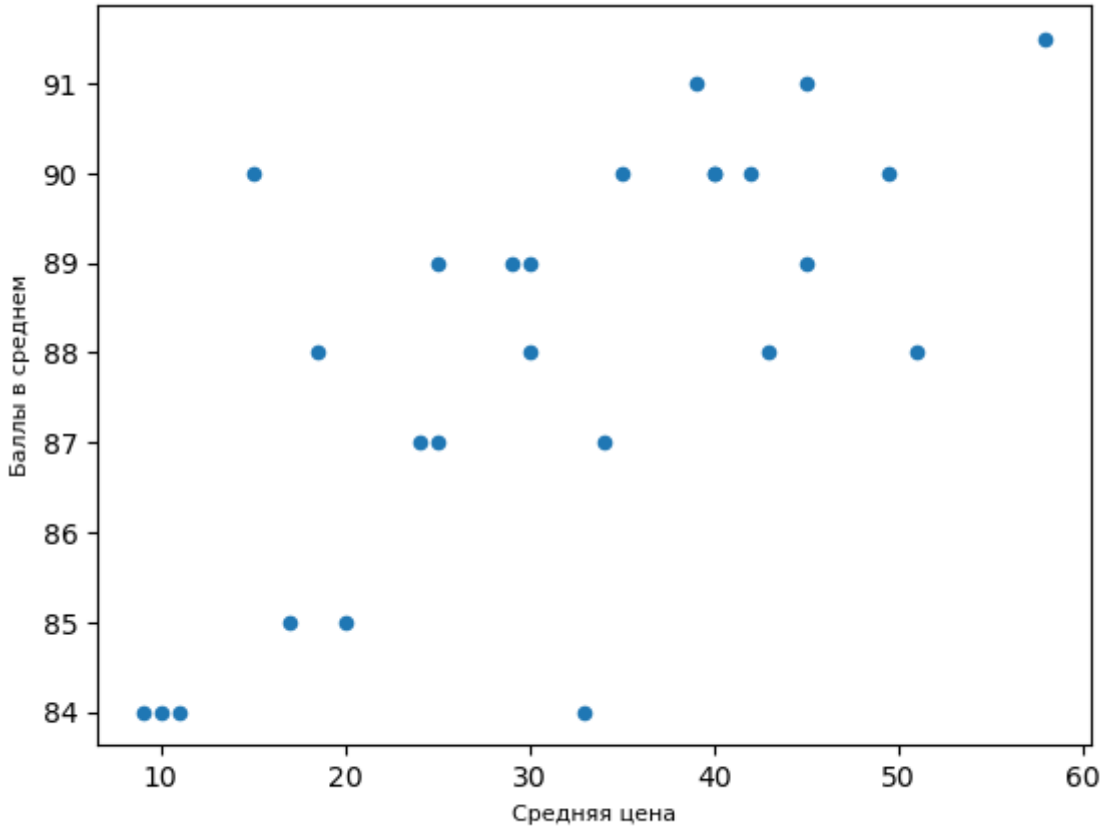


Рисунок 22. - Диаграмма рассеивания по Африке

In [55]:
```
asia=(df
    .query('continent=="Asia"')
    .pivot_table(values=['points','price'], index=['province','variety'], aggfunc=
    .groupby('province')['points','price'].max().sort_values(by='points',ascending=
)
print(asia.head())
print(asia.corr())
```

```
              points  price
province
Galilee         91.0   50.0
Lebanon         91.0   30.0
Judean Hills    90.5   70.0
Shomron         90.0   60.0
Aegean          89.0  120.0
          points     price
points  1.000000  0.233517
price   0.233517  1.000000
```

In [56]:
```
asia.plot(kind='scatter', x='price', y='points')
plt.xlabel('Средняя цена',fontsize=8)
plt.ylabel('Баллы в среднем',fontsize=8)
plt.figtext(0.37, -0.01, "Рисунок 23. - Диаграмма рассеивания по Азии", fontsize=1
```

Out[56]:
```
Text(0.37, -0.01, 'Рисунок 23. - Диаграмма рассеивания по Азии')
```

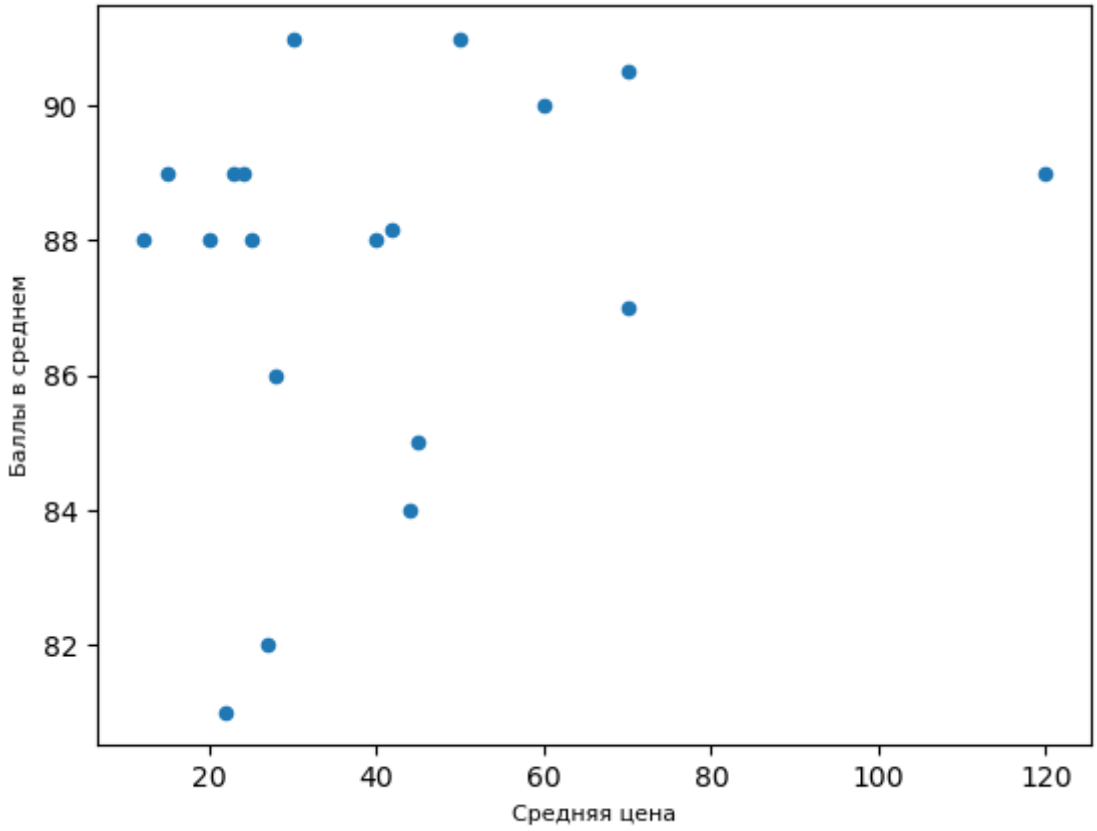


Рисунок 23. - Диаграмма рассеивания по Азии

In [57]:
```
europe=(df
    .query('continent=="Europe"')
    .pivot_table(values=['points','price'], index=['province','variety'], aggfunc=
    .groupby('province')['points','price'].max().sort_values(by='points',ascending=
)
print(europe.head())
print(europe.corr())
```

```
                          points       price
province
Bordeaux                    95.0   40.561345
Tokaji                      94.0  764.000000
Southwest France            94.0   55.000000
Neusiedlersee-Hügelland     94.0   48.000000
Piedmont                    94.0  224.000000
          points     price
points  1.000000  0.375799
price   0.375799  1.000000
```

In [63]:
```
europe.plot(kind='scatter', x='price', y='points')
plt.xlabel('Средняя цена',fontsize=8)
plt.ylabel('Баллы в среднем',fontsize=8)
plt.figtext(0.36, -0.01, "Рисунок 24. - Диаграмма рассеивания по Европе", fontsize=
```

Out[63]:
```
Text(0.36, -0.01, 'Рисунок 24. - Диаграмма рассеивания по Европе')
```

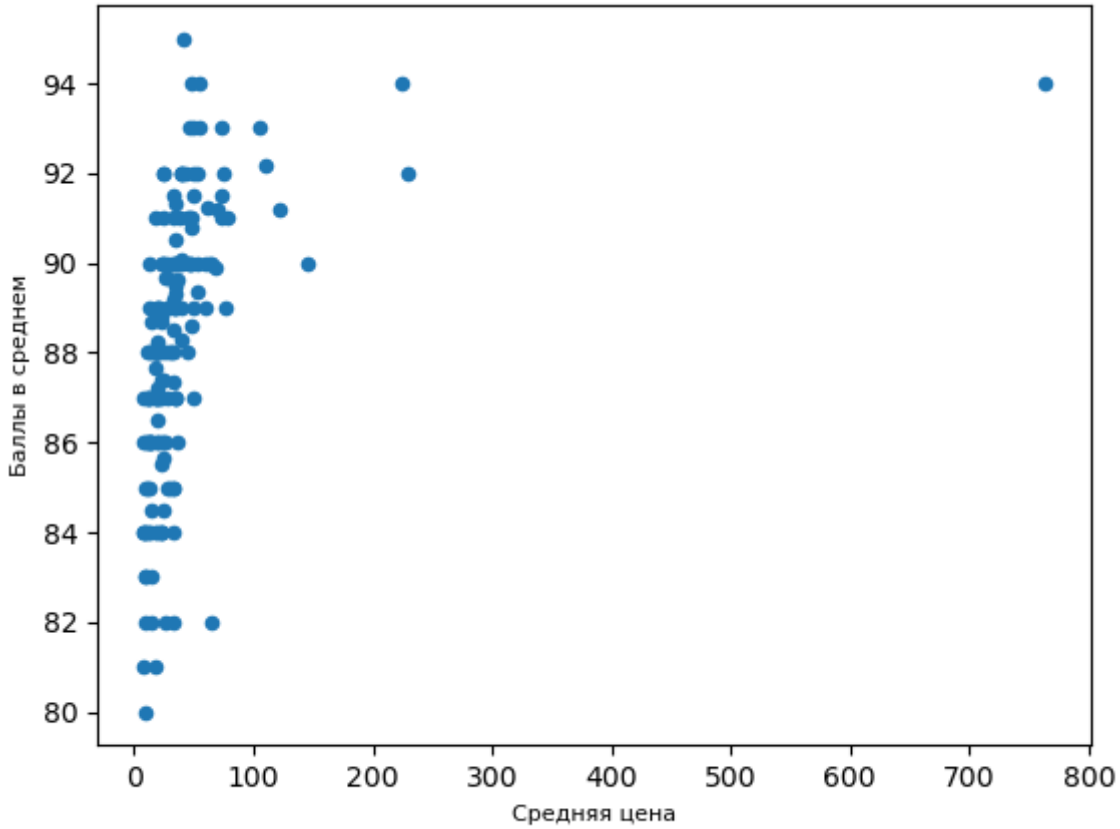


Рисунок 24. - Диаграмма рассеивания по Европе

In [64]:
```
latin_a=(df
    .query('continent=="Latin America"')
    .pivot_table(values=['points','price'], index=['province','variety'], aggfunc=
    .groupby('province')['points','price'].max().sort_values(by='points',ascending=
)
print(latin_a.head())
print(latin_a.corr())
```

```
                  points  price
province
Mendoza Province    94.0   77.0
San Antonio         92.5   64.5
Peumo               92.0   38.0
Loncomilla Valley   91.0   45.0
Pirque              91.0   37.0
          points     price
points  1.000000  0.714643
price   0.714643  1.000000
```

In [65]:
```
latin_a.plot(kind='scatter', x='price', y='points')
plt.xlabel('Средняя цена',fontsize=8)
plt.ylabel('Баллы в среднем',fontsize=8)
plt.figtext(0.2, -0.01, "Рисунок 25. - Диаграмма рассеивания по Латинской Америке"
```

Out[65]:
```
Text(0.2, -0.01, 'Рисунок 25. - Диаграмма рассеивания по Латинской Америке')
```

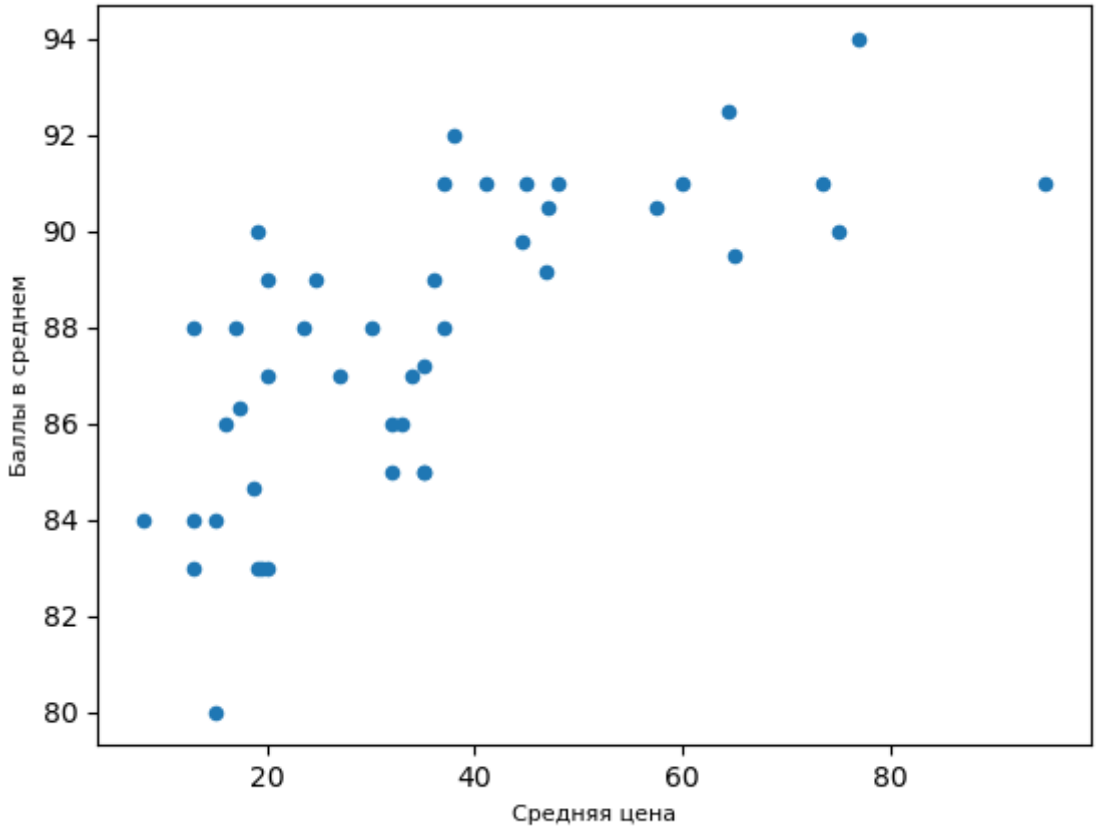


Рисунок 25. - Диаграмма рассеивания по Латинской Америке

In [66]:
```
north_a=(df
    .query('continent=="North America"')
    .pivot_table(values=['points','price'], index=['province','variety'], aggfunc=
    .groupby('province')['points','price'].max().sort_values(by='points',ascending=
)
print(north_a.head())
print(north_a.corr())
```

```
                  points       price
province
Washington          95.0   55.000000
California          95.0  130.000000
British Columbia    91.0  145.000000
Oregon              91.0   50.000000
Ontario             91.0   88.333333
          points     price
points  1.000000  0.636376
price   0.636376  1.000000
```

In [67]:
```
north_a.plot(kind='scatter', x='price', y='points')
plt.xlabel('Средняя цена',fontsize=8)
plt.ylabel('Баллы в среднем',fontsize=8)
plt.figtext(0.2, -0.01, "Рисунок 26. - Диаграмма рассеивания по Северной Америке",
```

Out[67]:
```
Text(0.2, -0.01, 'Рисунок 26. - Диаграмма рассеивания по Северной Америке')
```

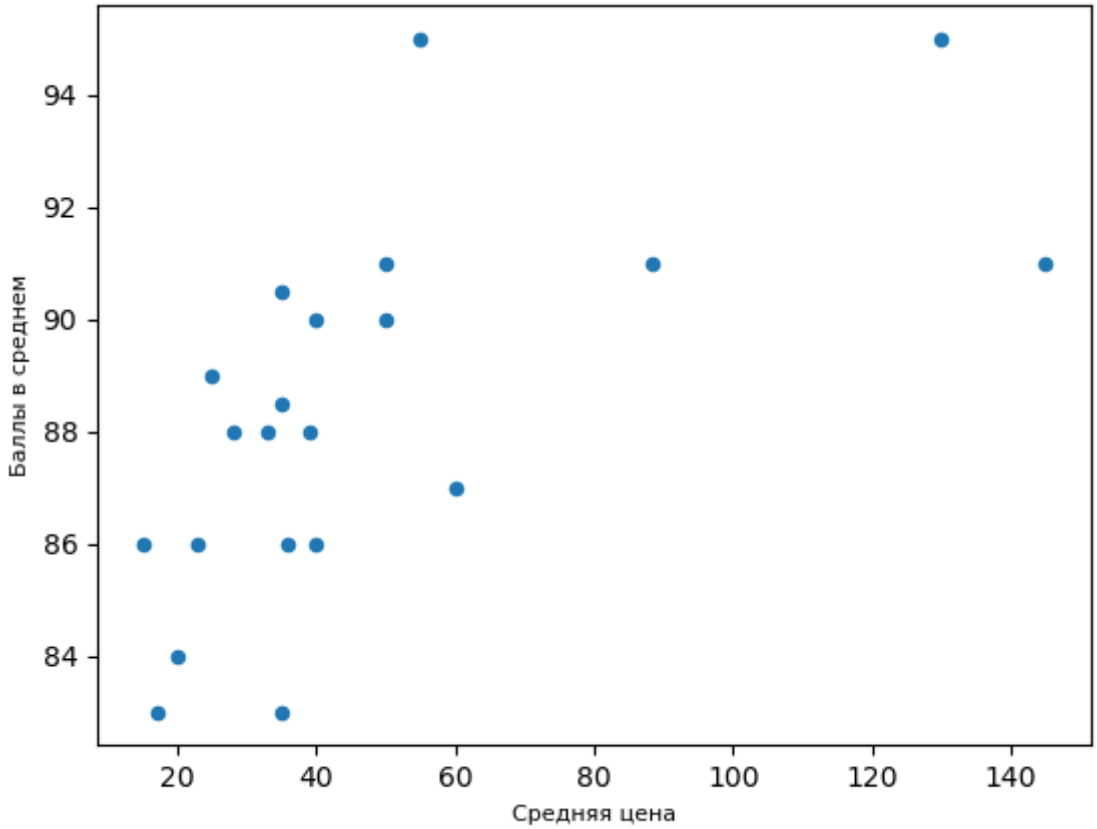


Рисунок 26. - Диаграмма рассеивания по Северной Америке

In [68]:
```
oceania=(df
    .query('continent=="Oceania"')
    .pivot_table(values=['points','price'], index=['province','variety'], aggfunc=
    .groupby('province')['points','price'].max().sort_values(by='points',ascending=
)
print(oceania.head())
print(oceania.corr())
```

```
                       points       price
province
Victoria                 96.0  160.666667
Martinborough Terrace    93.0   60.000000
Waiheke Island           92.0   45.000000
South Australia          92.0   70.000000
Kumeu                    90.5   40.166667
          points     price
points  1.000000  0.818386
price   0.818386  1.000000
```

In [69]:
```
oceania.plot(kind='scatter', x='price', y='points')
plt.xlabel('Средняя цена',fontsize=8)
plt.ylabel('Баллы в среднем',fontsize=8)
plt.figtext(0.32, -0.01, "Рисунок 27. - Диаграмма рассеивания по Океании", fontsiz
```

Out[69]:
```
Text(0.32, -0.01, 'Рисунок 27. - Диаграмма рассеивания по Океании')
```

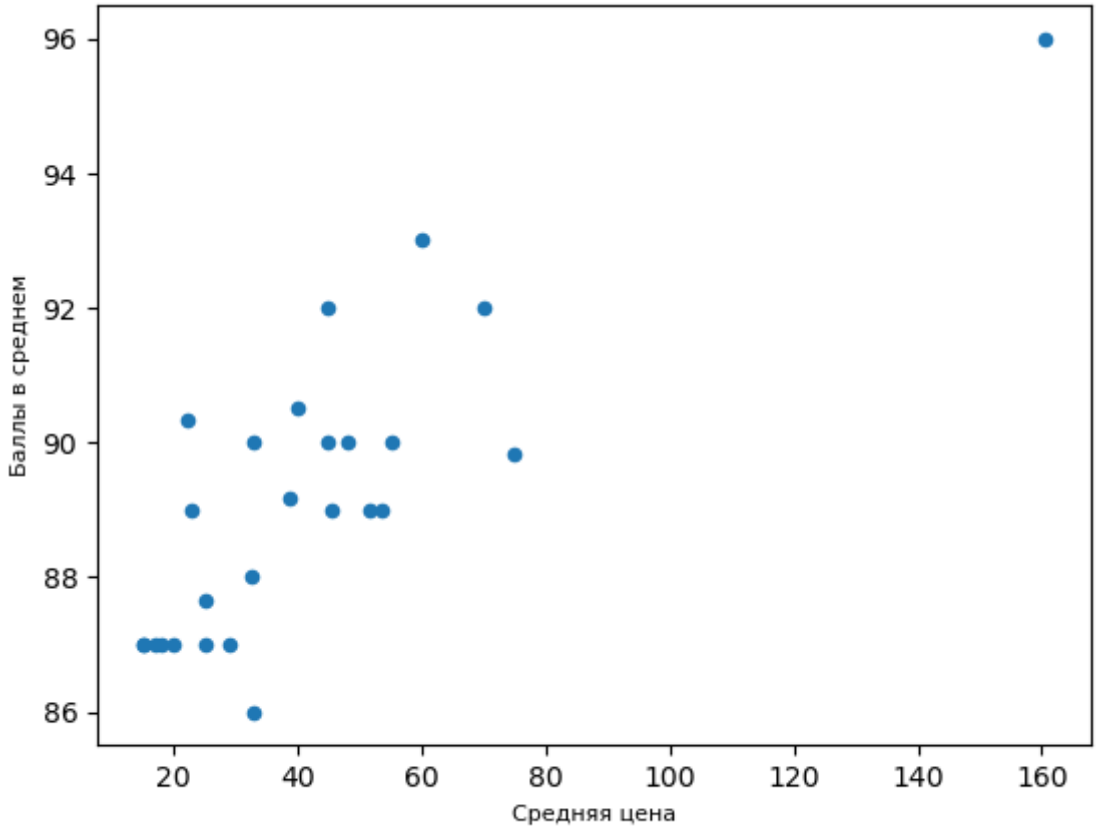


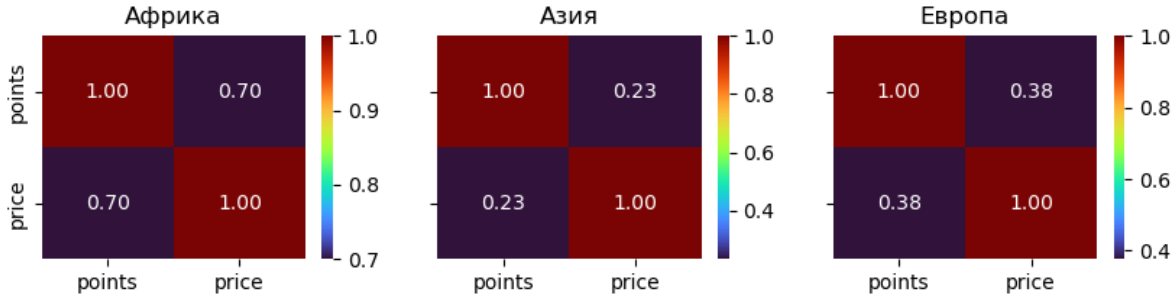
Рисунок 27. - Диаграмма рассеивания по Океании

Все корреляционные матрицы одновременно:

In [70]:
```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(10,2))
sns.heatmap(africa.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f', cmap='
sns.heatmap(asia.corr(method='pearson'), ax=ax[1], annot=True, fmt='.2f', cmap='tu
sns.heatmap(europe.corr(method='pearson'), ax=ax[2], annot=True, fmt='.2f', cmap='
ax[0].title.set_text('Африка')
ax[1].title.set_text('Азия')
ax[2].title.set_text('Европа')

fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(10,2))
sns.heatmap(latin_a.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f', cmap=
sns.heatmap(north_a.corr(method='pearson'), ax=ax[1], annot=True, fmt='.2f', cmap=
sns.heatmap(oceania.corr(method='pearson'), ax=ax[2], annot=True, fmt='.2f', cmap=
ax[0].title.set_text('Латинская Америка')
ax[1].title.set_text('Северная Америка')
ax[2].title.set_text('Океания')
plt.figtext(0.48, -0.25, "Рисунок 28. - Корреляционные матрицы по континентам", fo
```

Out[70]:
```
Text(0.48, -0.25, 'Рисунок 28. - Корреляционные матрицы по континентам')
```



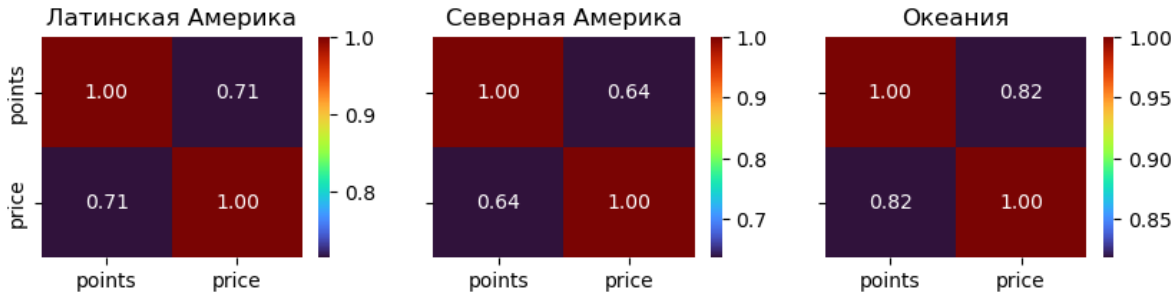


Рисунок 28. - Корреляционные матрицы по континентам

Судя по данным корреляционных матриц можно сказать, что самое понятное взаимодействие показателей Цены и Баллов(рейтинга) проявляется на континенте Океания - зависимость в 80% случаев абсолютно простая: дорогое вино = хорошее, дешевое = плохое. Чуть менее уверенно чувствуют себя потребители вина на Африканском и Латиноамериканском континентах. И самая слабая корреляция в Азии, где от цены на вино практически не зависит его рейтинг, а значит и качество.

После такого подробного анализа становится понятно, почему общая корреляция Цена\Баллы составляет всего 0.4

Выводы по Шагу 3 - Портрет пользователя:

- Наиболее популярными являются европейские вина, Северная Америка лишь на втором месте, и далее и большим отрывом производители остальных континентов;
- в Африке первое место держит сорт 'Syrah' (Шираз), древнее вино крестоносцев, теперь пользующее популярностью в ЮАР;
- В Азии - сорт 'Viognier' (Вионье), название которого произошло от латинского via Gehennae, что переводится как «дорога в ад»;
- В Европе с небольшим отрывом отличился сорт 'Sauvignon Blanc-Verdejo' (Вердехо), сухое белое вино;
- В Латинской Америке первое место держит 'Semillon' (Семильон) — светлокожий сорт французского винограда;
- В Северной Америке - 'Angevine' (Анжевин), из самого раннего по созреванию сорта винограда, названного так в честь Angevin Empire - Анжуйской империи, владении франко-английских монахов в 11-12 веках;
- В Океании - 'Muscat' (Мускат), десертное вино из черного винограда;
- Корреляционные матрицы по континентам показывают, что самая сильная корреляция показателей Цена/Баллы в Океании, самая слабая - в Азии.

4. Исследование статистических показателей.

Подсчитаем среднее количество, дисперсию и стандартное отклонение для цен на вина самых популярных регионов:

In [71]:
```
region1 = df[(df['province'] == 'Champagne') & (df['price'] > 0)]['price']
print('Показатели цен для региона Champagne:')
print('Среднее', statistics.mean(region1))
print('Дисперсия', statistics.pvariance(region1))
print('Стандартное отклонение', statistics.stdev(region1))
```

```
Показатели цен для региона Champagne:
Среднее 86.51381215469613
Дисперсия 15277.277433533776
Стандартное отклонение 123.94414547308561
```

In [72]:
```
region2 = df[(df['province'] == 'California') & (df['price'] > 0)]['price']
print('Показатели цен для региона California:')
print('Среднее', statistics.mean(region2))
print('Дисперсия', statistics.pvariance(region2))
print('Стандартное отклонение', statistics.stdev(region2))
```

```
Показатели цен для региона California:
Среднее 35.41264192509744
Дисперсия 639.3964102545989
Стандартное отклонение 25.288431792571153
```

In [73]:
```
region3 = df[(df['province'] == 'Tuscany') & (df['price'] > 0)]['price']
print('Показатели цен для региона Tuscany:')
print('Среднее', statistics.mean(region3))
print('Дисперсия', statistics.pvariance(region3))
print('Стандартное отклонение', statistics.stdev(region3))
```

```
Показатели цен для региона Tuscany:
Среднее 44.906666666666666
Дисперсия 1461.6989811965811
Стандартное отклонение 38.25179340679877
```

Проверим наши данные на нормальность при помощи теста Шапиро-Уилка.

Тест оценивает набор данных и дает количественную оценку вероятности того, что данные были получены из Гауссовского (нормального) распределения.

In [74]:
```
stat, p = st.shapiro(region1)
print('Statistics =', stat, 'p =', p)
alpha = 0.05
if p > alpha:
    print('Отклонить гипотезу о нормальности')
else:
    print('Принять гипотезу о нормальности')
```

```
Statistics = 0.40345996618270874 p = 3.497288999982973e-24
Принять гипотезу о нормальности
```

In [75]:
```
stat, p = st.shapiro(region2)
print('Statistics =', stat, 'p =', p)
alpha = 0.05
if p > alpha:
    print('Отклонить гипотезу о нормальности')
else:
    print('Принять гипотезу о нормальности')
```

```
Statistics = 0.7648299336433411 p = 0.0
Принять гипотезу о нормальности
```

In [76]:
```
stat, p = st.shapiro(region3)
print('Statistics =', stat, 'p =', p)
alpha = 0.05
if p > alpha:
```

```
    print('Отклонить гипотезу о нормальности')
else:
    print('Принять гипотезу о нормальности')
```

```
Statistics = 0.6694200038909912 p = 8.164427281649611e-40
Принять гипотезу о нормальности
```

Тест Д'Агостино вычисляет итоговую статистику на основе данных, чтобы определить, отклоняется ли распределение данных от нормального распределения:

In [77]:
```
value, p = st.normaltest(region1)
alpha = 0.05
if p < alpha:
    print('Данные распределены нормально')
else:
     print('Данные распределены НЕ нормально')
print('p =',p)
```

```
Данные распределены нормально
p = 5.872439822875567e-63
```

In [78]:
```
value,p = st.normaltest(region2)
alpha = 0.05
if p < 0.05:
    print('Данные распределены нормально')
else:
     print('Данные распределены НЕ нормально')
print('p =',p)
```

```
Данные распределены нормально
p = 0.0
```

In [79]:
```
value,p = st.normaltest(region3)
alpha = 0.05
if p < 0.05:
    print('Данные распределены нормально')
else:
     print('Данные распределены НЕ нормально')
print('p =',p)
```

```
Данные распределены нормально
p = 7.13166037102052e-201
```

Оба теста подтверждают, что данные наших выборок по самым популярным регионам распределены нормально.

Для более наглядно оценки распределения данных посмотрим графики:

In [141…
```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(12,3))
sns.distplot(region1, ax=ax[0], color='b')
sns.distplot(region2, ax=ax[1], color='y')
sns.distplot(region3, ax=ax[2], color='r')
ax[0].title.set_text('Регион Champagne')
ax[1].title.set_text('Регион California')
ax[2].title.set_text('Регион Tuscany')
plt.figtext(0.41, -0.19, "Рисунок 29. - Распределение данных в выборках по регионам
```

Out[141]:
```
Text(0.41, -0.19, 'Рисунок 29. - Распределение данных в выборках по регионам')
```

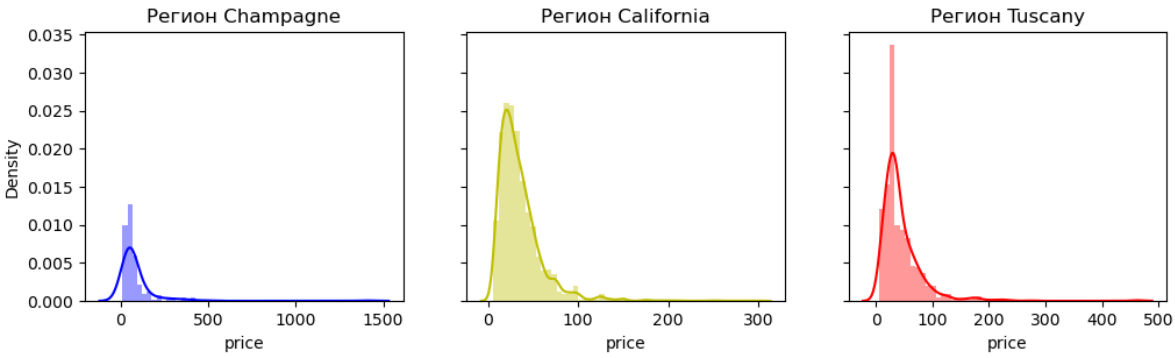


Рисунок 29. - Распределение данных в выборках по регионам

Поскольку данные по трем самым популярным регионам распределены нормально, можно сделать вывод, что датасет в целом имеет нормальное распределение.

Посмотрим на распределение показателей цены и баллов (рейтинга):

In [86]:
```
fig,(ax1)=plt.subplots(1)
fig.set_size_inches(8,4)
pp=df.loc[df['price'],'points']
sns.distplot(pp,ax=ax1)
plt.xlabel('')
plt.ylabel('')
plt.figtext(0.35, -0.04, "Рисунок 30. - Распределение показателей Цены и Баллов",
```

Out[86]:
```
Text(0.35, -0.04, 'Рисунок 30. - Распределение показателей Цены и Баллов')
```

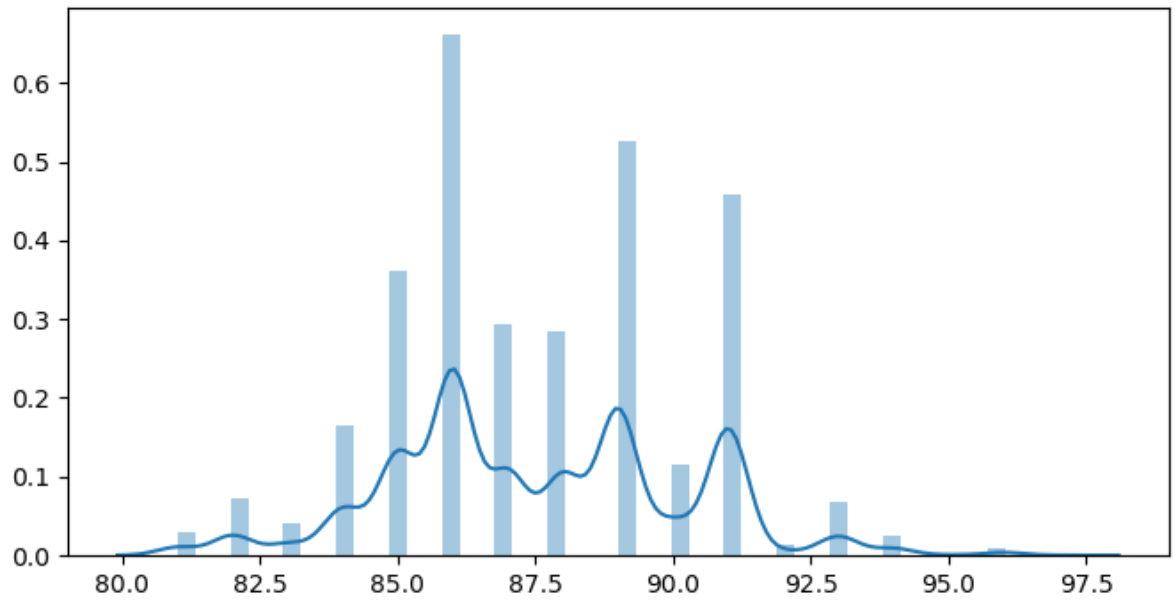


Рисунок 30. - Распределение показателей Цены и Баллов

Возникает закономерный вопрос, а сильно ли будет зависить цена на определенные сорта вина от рейтинга, оценок пользователей? Не смотря на то, что корреляция в целом по датасету достаточно слабая, есть соблазн построить модель линейной регрессии и посмотреть, как будут распределены показатели.

Для этого сделаем копию нашего датасета и зададим целевую переменную:

In [87]:
```
df_copy1 = df.copy()
df_copy1.shape
```

Out[87]:
```
(19997, 12)
```

In [88]:
```
df_copy1.head(3)
```

Out[88]:

	country	description	designation	points	price	province	region_1	region_2	variety	
0	US	With a delicate, silky mouthfeel and bright ac...	Unknown	86	23	California	Central Coast	Central Coast	Pinot Noir	Mac
1	Italy	D'Alceo is a drop dead gorgeous wine that ooze...	D'Alceo	96	275	Tuscany	Toscana	Unknown	Red Blend	Ra
2	France	The great dominance of Cabernet Sauvignon in t...	Unknown	91	40	Bordeaux	Haut-Médoc	Unknown	Bordeaux-style Red Blend	Berr

Удаленим колонки с уникальными значениями, которые не пригодятся для статистики:

In [89]:
```
df_copy1.drop(['country','description','designation','province','region_1','region_
df_copy1.head(3)
```

Out[89]:

	points	price
0	86	23
1	96	275
2	91	40

In [90]:
```
y = df_copy1['price']
X = df_copy1['points']
```

In [94]:
```
x=sm.add_constant(X)
results=sm.OLS(y,x).fit()
results.summary()
```

Out[94]:

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.167
Model:	OLS	**Adj. R-squared:**	0.167
Method:	Least Squares	**F-statistic:**	4006.
Date:	Mon, 19 Jun 2023	**Prob (F-statistic):**	0.00
Time:	13:32:44	**Log-Likelihood:**	-99230.
No. Observations:	19997	**AIC:**	1.985e+05
Df Residuals:	19995	**BIC:**	1.985e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-386.3738	6.634	-58.243	0.000	-399.377	-373.371
points	4.7732	0.075	63.290	0.000	4.625	4.921

Omnibus:	45637.996	**Durbin-Watson:**	1.995
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	899367092.273
Skew:	21.537	**Prob(JB):**	0.00
Kurtosis:	1041.050	**Cond. No.**	2.39e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.39e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Для того, чтобы повысить коэффициент детерминации закодируем переменные, которые также могут повлиять на цену вина:

Делаем еще одну копию нашего основного датасета и кодируем столбец с цветом вина номером индекса:

In [95]:
```
df_copy2 = df.copy()
df_copy2.shape
```

Out[95]:
```
(19997, 12)
```

In [100…
```
color_code = df_copy2.groupby('color_wine').size()
color_code = color_code/len(df_copy2)
df_copy2['color_code'] = df_copy2['color_wine'].map(color_code).round(3)

df_copy2['color_code']
```

Out[100]:
```
0        0.579
1        0.579
2        0.579
3        0.579
4        0.579
         ...
19995    0.219
19996    0.579
19997    0.219
19998    0.219
19999    0.579
Name: color_code, Length: 19997, dtype: float64
```

Кодируем столбец с сортом вина номером индекса:

In [101…
```
wine_code = df_copy2.groupby('variety').size()
wine_code = wine_code/len(df_copy2)
df_copy2['wine_code'] = df_copy2['variety'].map(wine_code).round(3)

df_copy2['wine_code']
```

Out[101]:
```
0        0.097
1        0.066
2        0.048
3        0.022
4        0.097
         ...
19995    0.000
19996    0.097
19997    0.000
19998    0.000
19999    0.039
Name: wine_code, Length: 19997, dtype: float64
```

Кодируем столбец страны производителя номером индекса:

In [102…
```
country_code = df_copy2.groupby('country').size()
country_code = country_code/len(df_copy2)
df_copy2['country_code'] = df_copy2['country'].map(country_code).round(3)

df_copy2['country_code']
```

Out[102]:
```
0        0.412
1        0.155
2        0.138
3        0.155
4        0.412
         ...
19995    0.138
19996    0.412
19997    0.155
19998    0.155
19999    0.412
Name: country_code, Length: 19997, dtype: float64
```

Смотрим, что получилось. Добавились новые колонки:

In [103…
```
df_copy2.head(3)
```

Out[103]:

	country	description	designation	points	price	province	region_1	region_2	variety	
0	US	With a delicate, silky mouthfeel and bright ac...	Unknown	86	23	California	Central Coast	Central Coast	Pinot Noir	Mac
1	Italy	D'Alceo is a drop dead gorgeous wine that ooze...	D'Alceo	96	275	Tuscany	Toscana	Unknown	Red Blend	Ra
2	France	The great dominance of Cabernet Sauvignon in t...	Unknown	91	40	Bordeaux	Haut-Médoc	Unknown	Bordeaux-style Red Blend	Berr

Удаленим колонок с уникальными значениями, которые не пригодятся для статистики:

In [104...
```
df_copy2.drop(['country','description','designation','province','region_1','region_
df_copy2.head(3)
```

Out[104]:

	points	price	color_code	wine_code	country_code
0	86	23	0.579	0.097	0.412
1	96	275	0.579	0.066	0.155
2	91	40	0.579	0.048	0.138

Перегруппируем колонки для удобства:

In [105...
```
neworder = ['price','points','color_code','wine_code','country_code']
df_copy2=df_copy2.reindex(columns=neworder)
```

Посмотрим общую статистику получившегося датасета:

In [106...
```
df_copy2.describe()
```

Out[106]:

	price	points	color_code	wine_code	country_code
count	19997.000000	19997.000000	19997.000000	19997.000000	19997.000000
mean	33.189678	87.899335	0.424184	0.042427	0.222677
std	37.887149	3.242656	0.181827	0.034929	0.164754
min	5.000000	80.000000	0.202000	0.000000	0.000000
25%	16.000000	86.000000	0.219000	0.010000	0.055000
50%	25.000000	88.000000	0.579000	0.037000	0.155000
75%	38.000000	90.000000	0.579000	0.082000	0.412000
max	2300.000000	100.000000	0.579000	0.097000	0.412000

Корреляция:

In [107…
```
df_copy2.corr()
```

Out[107]:

	price	points	color_code	wine_code	country_code
price	1.000000	0.408527	0.121349	0.102949	0.038706
points	0.408527	1.000000	0.106404	0.074409	0.021745
color_code	0.121349	0.106404	1.000000	0.282613	0.126390
wine_code	0.102949	0.074409	0.282613	1.000000	0.310762
country_code	0.038706	0.021745	0.126390	0.310762	1.000000

Корреляция на тепловой карте:

In [108…
```
plt.figure(figsize=(7,4))
sns.heatmap(df_copy2.corr(), annot=True, fmt='.3f', cmap='turbo')
plt.xticks(fontsize=8)
plt.yticks(fontsize=8)
plt.figtext(0.13, -0.07, "Рисунок 31. - Корреляционная матрица для регрессионной мо
```

Out[108]:
```
Text(0.13, -0.07, 'Рисунок 31. - Корреляционная матрица для регрессионной модели')
```

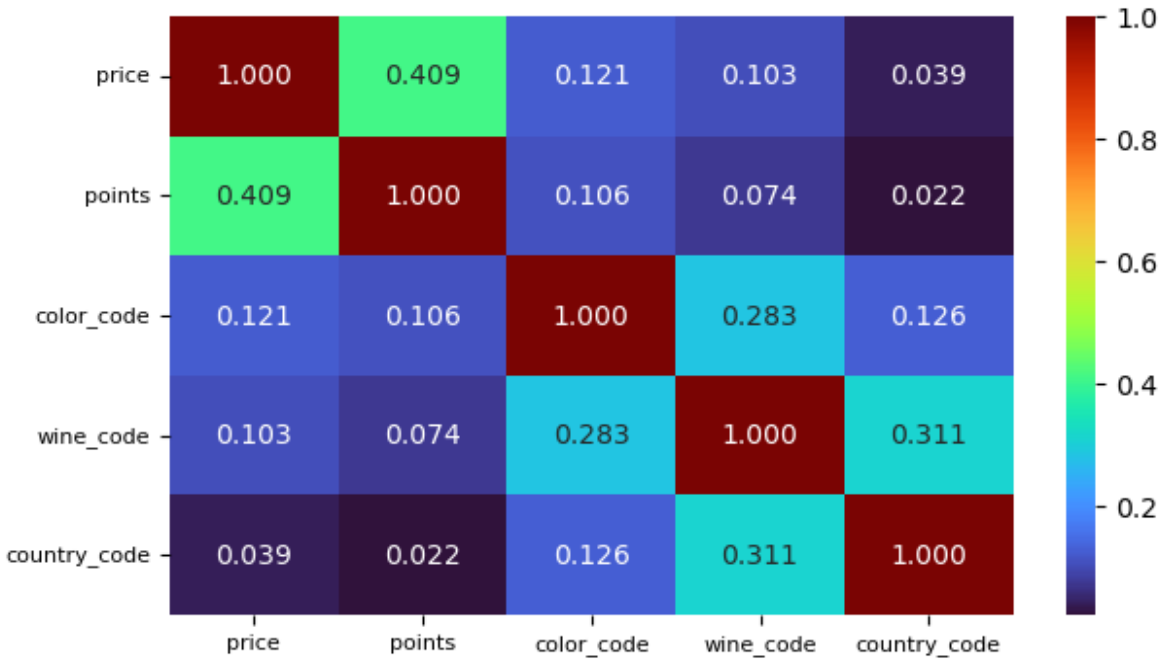


Рисунок 31. - Корреляционная матрица для регрессионной модели

Снова перемещестим целевые данные в отдельную переменную, остальные - во вторую:

In [109…
```
y = df_copy2['price']
X = df_copy2.iloc[:, 1:]
```

Посмотрим в целом, что покажет график с расширенным набором числовых показателей:

In [110…

```
fig.set_size_inches(4,2)
sns.jointplot(x='points', y='price', data=df_copy2, kind='reg', color='b')
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('Цена',fontsize=8)
plt.figtext(0.3, -0.01, "Рисунок 33. - Распределение данных Цены и Баллов", fontsi
```

Out[110]:

```
Text(0.3, -0.01, 'Рисунок 33. - Распределение данных Цены и Баллов')
```

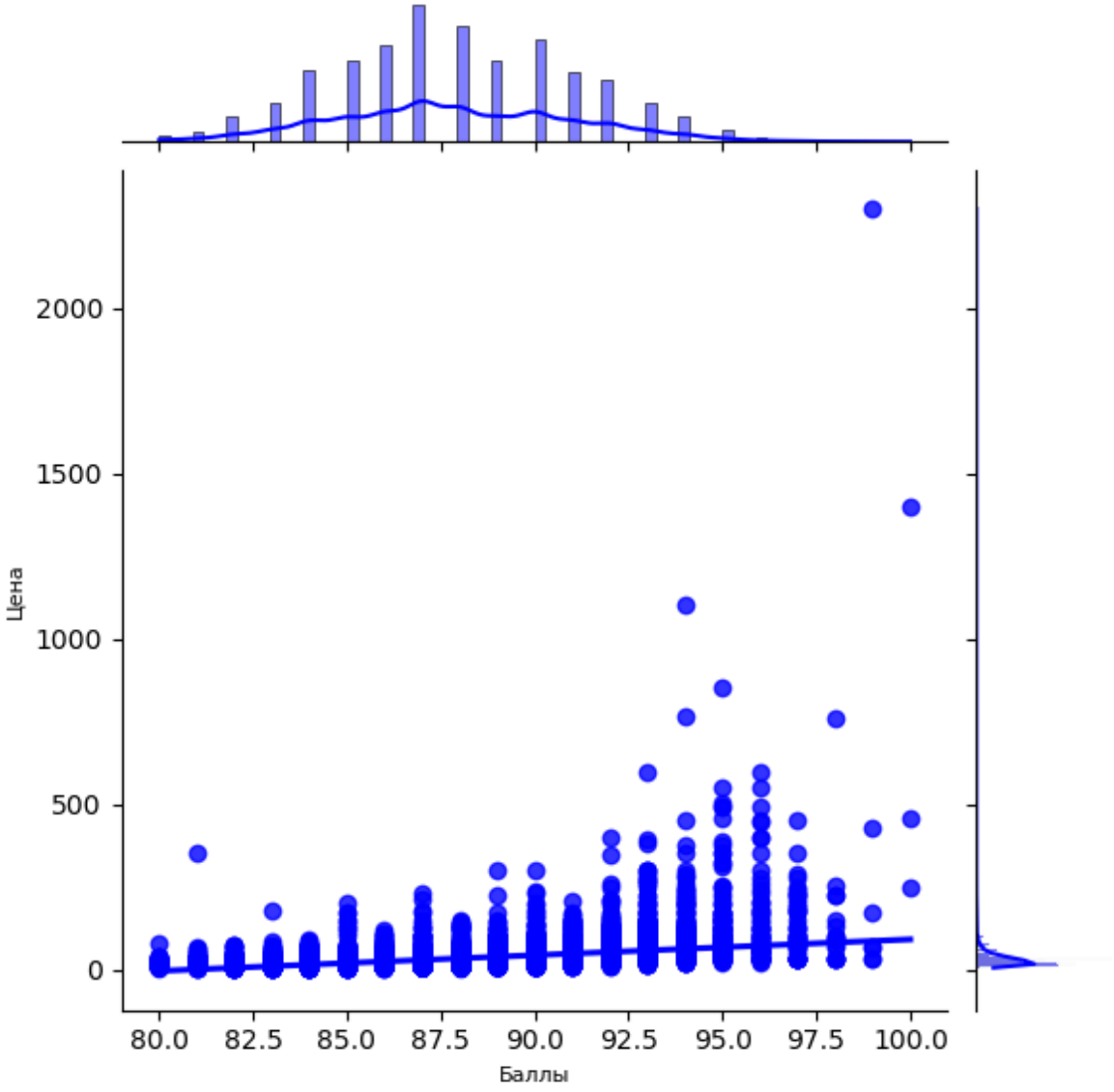


Рисунок 33. - Распределение данных Цены и Баллов

Посмотрим статистику с новыми переменными:

In [111…

```
x=sm.add_constant(X)
results=sm.OLS(y,x).fit()
results.summary()
```

Out[111]:

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.176
Model:	OLS	**Adj. R-squared:**	0.176
Method:	Least Squares	**F-statistic:**	1067.
Date:	Mon, 19 Jun 2023	**Prob (F-statistic):**	0.00
Time:	13:34:20	**Log-Likelihood:**	-99121.
No. Observations:	19997	**AIC:**	1.983e+05
Df Residuals:	19992	**BIC:**	1.983e+05
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-383.5719	6.607	-58.055	0.000	-396.522	-370.621
points	4.6466	0.076	61.521	0.000	4.499	4.795
color_code	13.1564	1.402	9.387	0.000	10.409	15.904
wine_code	58.4136	7.592	7.694	0.000	43.533	73.294
country_code	1.2286	1.555	0.790	0.429	-1.819	4.276

Omnibus:	45964.888	**Durbin-Watson:**	1.999
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	940166436.451
Skew:	21.915	**Prob(JB):**	0.00
Kurtosis:	1064.342	**Cond. No.**	2.76e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.76e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Результат показателя коэффициента детерминации улучшен, но не намного. Рассмотрим построение модели машинного обучения используя метод линейной регрессии.

Разделяем массивы для обучения модели, где train_size - размер тренировочной части (70% данных):

In [112…
```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_s
X_train.shape, X_test.shape
```

Out[112]:
```
((13997, 4), (6000, 4))
```

Создаем модель, настраиваем параметры, обучаем:

In [119…
```
model  = LinearRegression()
parameters = {'fit_intercept': [True, False],'normalize': [True, False]}
```

```
# fit_intercept - логический параметр, который решает, вычислять отрезок (True) или
# normalize - логический параметр, который решает, нормализовать входные переменные

grad_Linear = GridSearchCV(model, parameters, refit=True, cv = 10) # Кросс-валидаци
grad_Linear.fit(X_train, y_train)

print('Лучший результат: ', grad_Linear.best_score_, '\nЛучшие параметры: ', grad_
```

```
Лучший результат:  0.2014271988343491
Лучшие параметры:  {'fit_intercept': True, 'normalize': True}
```

Показатель "Лучший результат" в кросс-валидации будет примерно похож на коэффициент корреляции в результатах обучения.

Предсказание на модели линейной регрессии:

In [117…
```
pred_lr = grad_Linear.predict(X_test)
```

Результаты обучения:

In [118…
```
print('R2: ', r2_score(y_test, pred_lr))
print('MAE: ' +str(np.sqrt(mean_absolute_error(y_test, pred_lr))))
```

```
R2:  0.2017308857660015
MAE: 3.906424909582232
```

Сравнение актуальных цен и предсказанных моделью:

In [121…
```
pred = pd.DataFrame({'Обучающие данные': y_test.tolist(),
                     'Предсказание': pred_lr.tolist()}).head(5)
pred.head()
```

Out[121]:

	Обучающие данные	Предсказание
0	42	52.621857
1	15	29.203723
2	53	37.814169
3	30	43.234330
4	13	35.753689

Не смотря на то, что результаты работы прогнозной модели являются удовлетворительными, коэффициент детерминации удалось повысить.

Выводы по Шагу 4 - Исследование статистических показателей:

- Проверка на нормальность средних показателей цен самых популярных регонов по тестам Шариро-Уилка и Д'Агостино подтверждает предположение о том, что данные распределены нормально - это дает нам поработать с линейной регрессией;
- Коэффициент детерминации на модели OLS с двумя числовыми показателями (Цена и Баллы) дает результат 0.167, что говорит о слабой взаимосвязи данных;
- После кодировки показателей Цвета, Сорта вина и Страны коэффициент детерминации на модели OLS показывает, что линейная связь между

переменными выражена чуть лучше, 0.176;
- Модель линейной регрессии из sklearn показала результат коэффициента детерминации 0.201, что немного лучше предыдущих попыток, но все равно недостаточно хорошо.

5. Проверка гипотез

Самыми популярными сортами вин являются Pinot Noir и Cabernet Sauvignon. Рассмотрим на их примере проверку гипотез:

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

In [122...
```
wine1 = df[(df['variety'] == 'Pinot Noir') & (df['price'] > 0)]['price']
print('Среднее по цене для Pinot Noir', statistics.mean(wine1))
```

```
Среднее по цене для Pinot Noir 43.36709511568123
```

In [123...
```
wine2 = df[(df['variety'] == 'Cabernet Sauvignon') & (df['price'] > 0)]['price']
print('Среднее по цене для Cabernet Sauvignon', statistics.mean(wine2))
```

```
Среднее по цене для Cabernet Sauvignon 42.294621026894866
```

Для проверки наших гипотез используем различные тесты, в которых фигурирует значение alpha. Значение alpha связано с уровнем достоверности наших тестов.

Уровни достоверности с соответствующими значениями alpha:

- Для результатов с 90-процентным уровнем достоверности значение альфа равно 1 — 0,90 = 0,10.
- Для результатов с 95-процентным уровнем достоверности значение альфа равно 1 — 0,95 = 0,05.
- Для результатов с 99-процентным уровнем достоверности значение альфа равно 1 — 0,99 = 0,01.

Для оценки результатов наших тестов возьмем пороговое значение alpha равное 0.05, т.е. 95% успеха нам будет достаточно.

Одним из критериев повышения точности тестирования является равенство дисперсий двух показателей. Для этого проверим дисперсии при помощи теста Левена:

In [124...
```
l, pl = st.levene(wine1,wine2)
alpha = 0.05 # критический уровень статистической значимости
print('pl =',pl)
if (pl < alpha):
    print("Дисперсии одинаковы")
else:
    print("Дисперсии различны")
```

```
pl = 1.9567679746437315e-13
Дисперсии одинаковы
```

Далее проверяем гипотезы при помощи критерия t-Стьюдента:

- Если p > alpha: принять нулевую гипотезу.
- Если p <= alpha: отклонить нулевую гипотезу.

In [125...
```
alpha = 0.05
results = scipy.stats.ttest_ind(wine1,wine2)
print('p-значение:', results.pvalue)
if (results.pvalue > alpha):
    print("Принять нулевую гипотезу")
elif (results.pvalue <= alpha):
    print("Отклонить нулевую гипотезу")
stat, p = scipy.stats.ttest_ind(wine1,wine2)
print('Statistics=%.3f, p=%.3f' % (stat, p))
```

```
p-значение: 0.31800012067413436
Принять нулевую гипотезу
Statistics=0.999, p=0.318
```

Таким образом у нас получается, что нулевая гипотеза верна и средние цены двух популярных сортов вина одинаковые.

In [126...
```
plt.figure(figsize=(10,4))
sns.histplot(wine1, color='b')
sns.histplot(wine2, color='y')
plt.xlim(0,200)
plt.xlabel('Цена',fontsize=8)
plt.ylabel('',fontsize=8)
plt.figtext(0.42, -0.08, "Рисунок 34. - Средние цены для Pinot Noir и Cabernet Sauv
```

Out[126]:
```
Text(0.42, -0.08, 'Рисунок 34. - Средние цены для Pinot Noir и Cabernet Sauvigno
n')
```

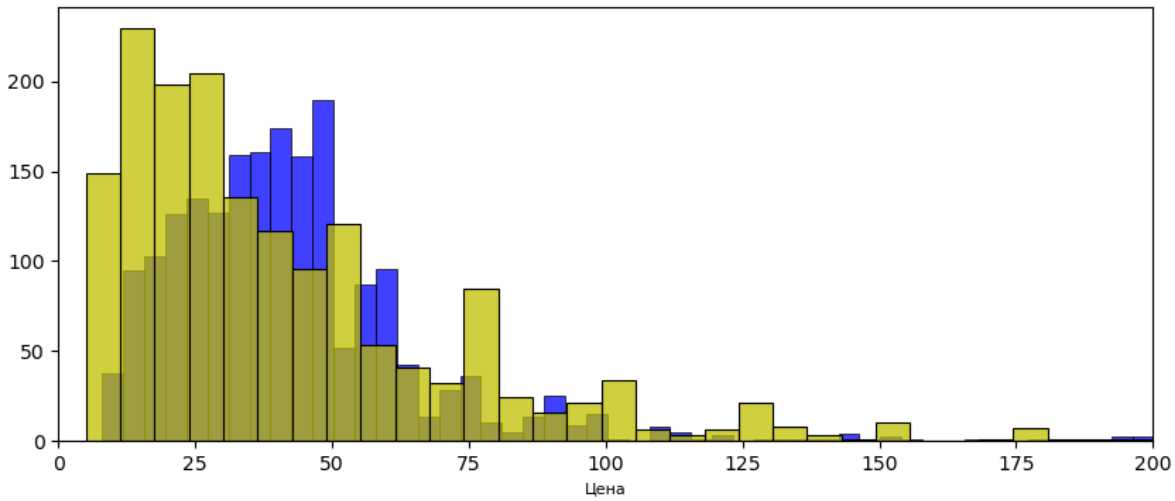


Рисунок 34. - Средние цены для Pinot Noir и Cabernet Sauvignon

Проверим следующую гипотезу:

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

In [127...
```
red_wine = df[(df['color_wine'] == 'red') & (df['points'] > 0)]['points']
print('Средний рейтинг для красного вина', statistics.mean(red_wine))
```

```
Средний рейтинг для красного вина 88.19451156368658
```

In [128...
```
white_wine = df[(df['color_wine'] == 'white') & (df['points'] > 0)]['points']
print('Средний рейтинг для белого вина', statistics.mean(white_wine))
```

```
Средний рейтинг для белого вина 87.56293359762141
```

Проверяем дисперсии:

In [129…
```
l, pl = st.levene(red_wine,white_wine)
alpha = 0.05 # критический уровень статистической значимости
print('pl =',pl)
if (pl < alpha):
    print("Дисперсии одинаковы")
else:
    print("Дисперсии различны")
```

```
pl = 7.28941927535056e-07
Дисперсии одинаковы
```

In [130…
```
alpha = 0.05
results = scipy.stats.ttest_ind(red_wine,white_wine)
print('p-значение:', results.pvalue)
if (results.pvalue > alpha):
    print("Принять нулевую гипотезу")
elif (results.pvalue <= alpha):
    print("Отклонить нулевую гипотезу")
    stat, p = scipy.stats.ttest_ind(red_wine,white_wine)
print('Statistics=%.3f, p=%.3f' % (stat, p))
```

```
p-значение: 9.48333289389097e-26
Отклонить нулевую гипотезу
Statistics=10.510, p=0.000
```

In [131…
```
print('Statistics=%.3f, p=%.3f' % (stat, p))
```

```
Statistics=10.510, p=0.000
```

При отклонении нулевой гипотезы получается, что средние рейтинги на красные и белые вина не равны.

In [132…
```
plt.figure(figsize=(10,4))
sns.histplot(red_wine, color='b')
sns.histplot(white_wine, color='y')
plt.xlim(70,100)
plt.xlabel('Баллы',fontsize=8)
plt.ylabel('',fontsize=8)
plt.figtext(0.5, -0.08, "Рисунок 35. - Средние баллы для красного и белых вин", fo
```

Out[132]:
```
Text(0.5, -0.08, 'Рисунок 35. - Средние баллы для красного и белых вин')
```

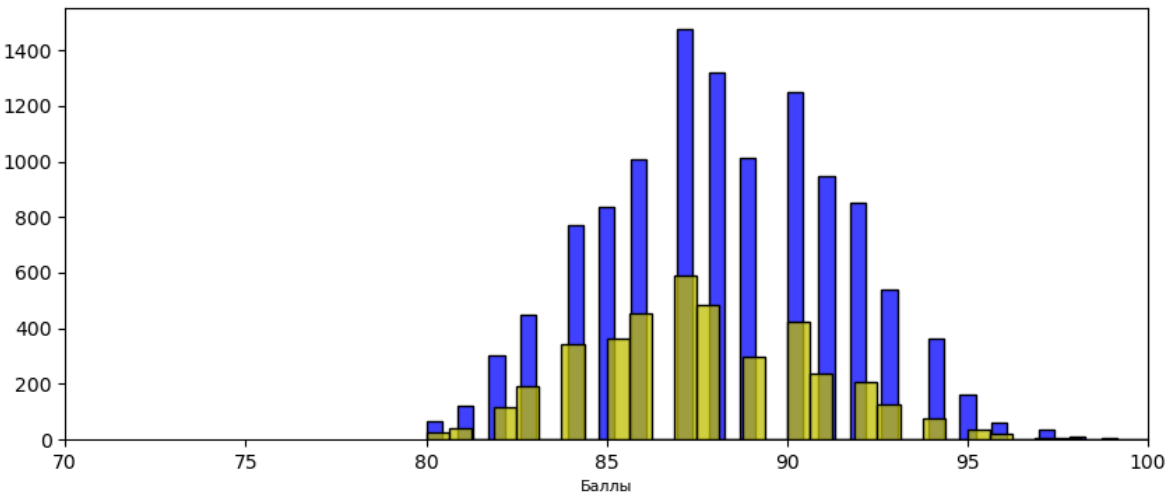


Рисунок 35. - Средние баллы для красного и белых вин

Выводы по Шагу 5 - Проверка гипотез:

- Средние цены двух популярных сортов вина Pinot Noir и Cabernet Sauvignon одинаковы, нулевая гипотеза принята;
- Средний рейтинг красного и белых вин различны, нулевая гипотеза отвергнута;
- Визуальное распределение показателей может быть обманчиво, необходимо проводить тесты.

6. Выводы

После проведенного исследования хочется выделить тезисно несколько основных выводов:

- Датасет был представлен в не полном объеме, большое пропусков количество в некоторых колонках могло повлиять на исследования;
- Количественных показателей в датасете всего два: Цена и Баллы (рейтинг);
- Данных с нестандартной оценкой было очень мало, поэтому выбросы из работы удалены не были;
- Предметно исходя из этих данных: лидер в производстве вин - США (44%), провинция - California; количество красного вина значительно преобладает над белым; цены на вино варьируются от 5 долл.США до 2300 долл.США, самые дорогие вина по средней оценке оказались в Венгрии; самый популярный сорт - Pinot Noir, самый непопулярный - Macabeo-Moscatel;
- Корелляция числовых показателей Цены и Баллов слабая, около 0.4 по всему датасету, варьируется в зависимости от местоположения винодельни. Ситуация по континентам выглядит интереснее: в Океании самая сильная корреляция данных 0.82, В Азии - самая слабая 0.23;
- Регрессионный анализ не выявил причинно-следственной взаимосвязи между Ценой и Баллами (рейтингом), что говорит о некорректном или не полном наполнении датасета, субъективности выставления данных оценок. Возможно вне этого датасета присутствуют факторы, непосредственно влияющие на изменение данных числовых показателей, но они не были представлены для анализа. (Рейтинг проставлен случайным образом, синтетические данные? Проплачена определенная реклама, накрутка отзывов?)

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. P28 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.