```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directo

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you cre
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/wine-reviews/winemag-data_first150k.csv
/kaggle/input/wine-reviews/winemag-data-130k-v2.json
/kaggle/input/wine-reviews/winemag-data-130k-v2.csv
```

```
winedata = pd.read_csv('../input/wine-reviews/winemag-data_first150k.csv')
winedata.describe().T
```

[3]:

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	150930.0	75464.500000	43569.882402	0.0	37732.25	75464.5	113196.75	150929.0
points	150930.0	87.888418	3.222392	80.0	86.00	88.0	90.00	100.0
price	137235.0	33.131482	36.322536	4.0	16.00	24.0	40.00	2300.0

葡萄酒的评论按国家地区分布

[3]:

```
Review_Country = pd.DataFrame(winedata["country"].value_counts())
Review_Country.describe().T
```

[3]:

	count	mean	std	min	25%	50%	75%	max
country	48.0	3144.270833	9930.379643	1.0	5.75	47.5	1227.5	62397.0

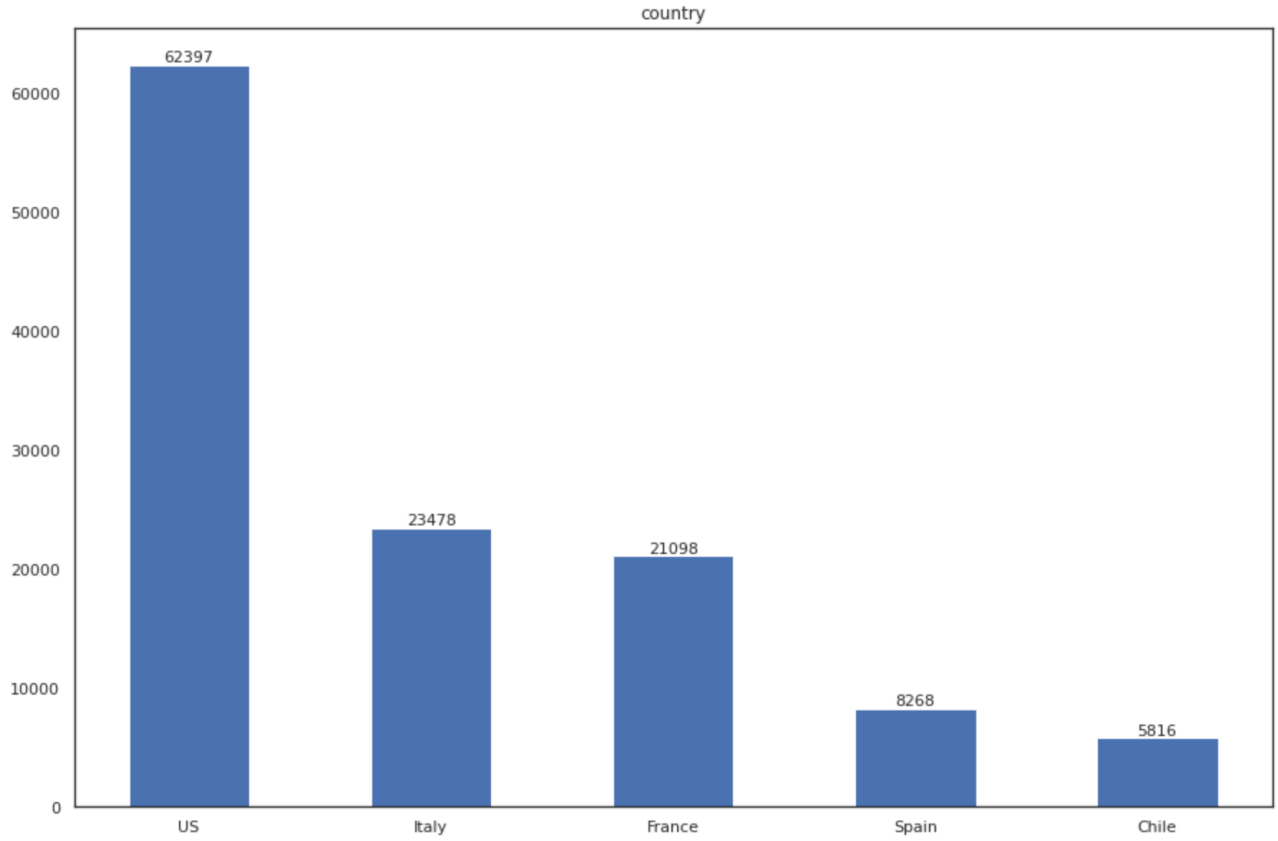
标称属性的频数

[40]:

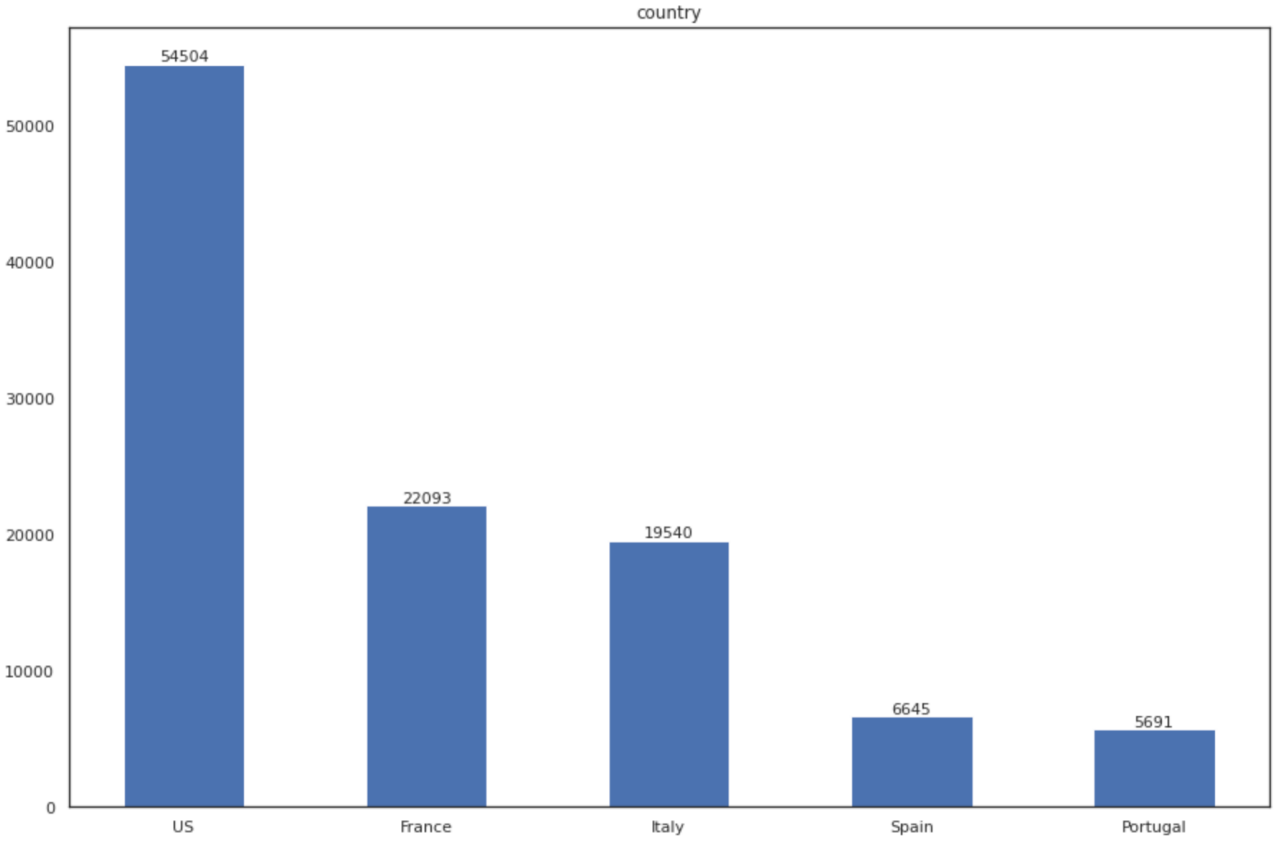
```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.spatial.distance import pdist
from math import ceil
import numpy as np
%matplotlib inline

data1_wine_review = pd.read_csv("../input/wine-reviews/winemag-data_first150k.csv", encoding="utf-8")[
    ["country", "province", "variety", "winery", "designation", "region_1","region_2" ]]
data2_wine_review = pd.read_csv("../input/wine-reviews/winemag-data-130k-v2.csv", encoding="utf-8")[
    ["country", "province", "variety", "winery", "designation", "region_1","region_2" ]]
data_all_wine = [data1_wine_review, data2_wine_review]
wine_name_list = ["winemag-data_first150k.csv", "winemag-data-130k-v2.csv"]
i=0
for datax in data_all_wine:
    print("\n" + wine_name_list[i] + "的频数聚合分析:")
    data1_number = datax
    data1_number = data1_number.dropna(axis=0,how='all')

    data1_country=data1_number["country"].value_counts(sort=True)
    data1_country=data1_country.head(5)
    data1_country_name=data1_country.index.tolist()
    data1_country_num=data1_country.values

    index=np.arange(5)
```

```
    plt.figure(figsize=(15,10))
    plt.bar(index,data1_country_num, 0.5, label="num")
    plt.xticks(index,data1_country_name)
    for a,b in zip(index,data1_country_num):
        plt.text(a, b+0.05, '%.0f' % b, ha='center', va= 'bottom',fontsize=11)
    plt.title("country")

    plt.show()
    i+=1
```

winemag-data_first150k.csv的频数聚合分析:



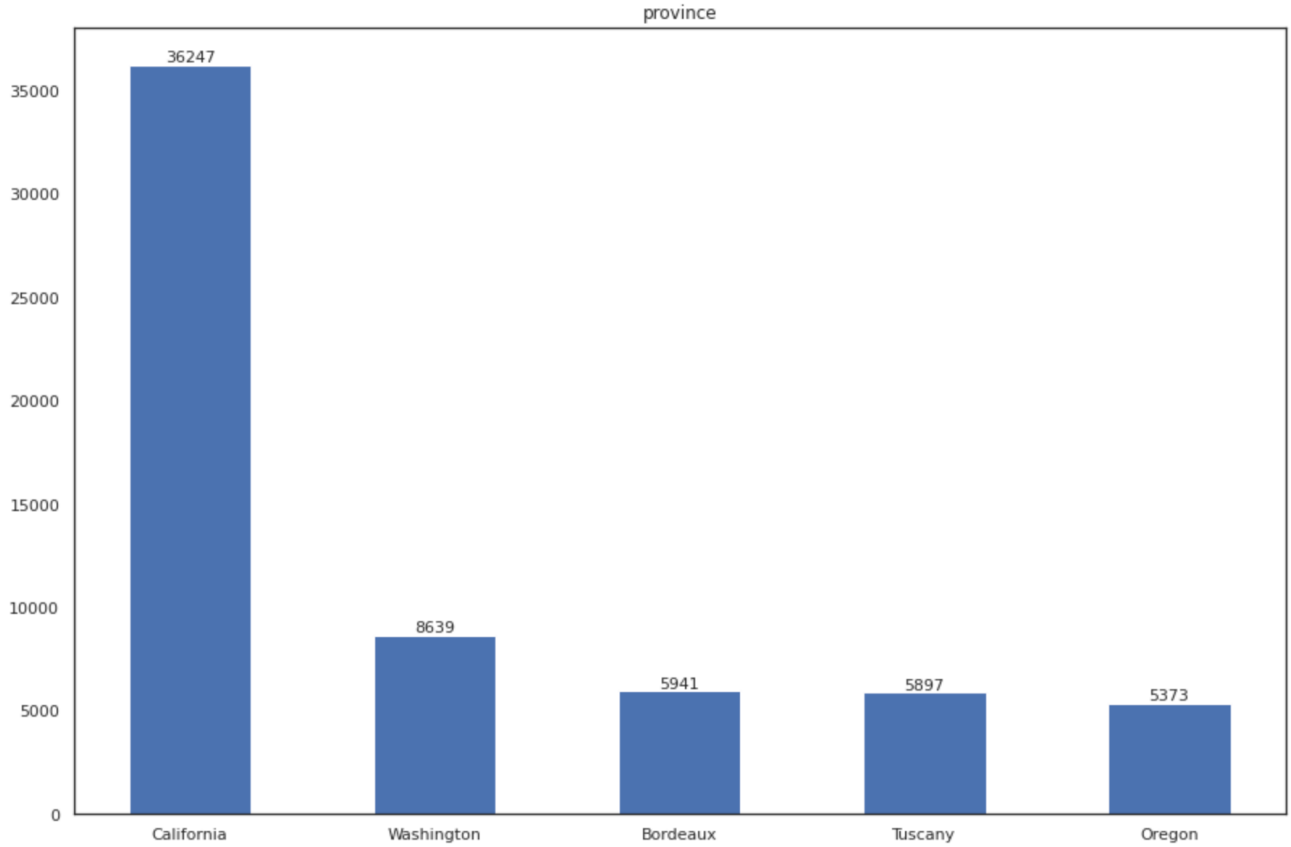
winemag-data-130k-v2.csv的频数聚合分析:



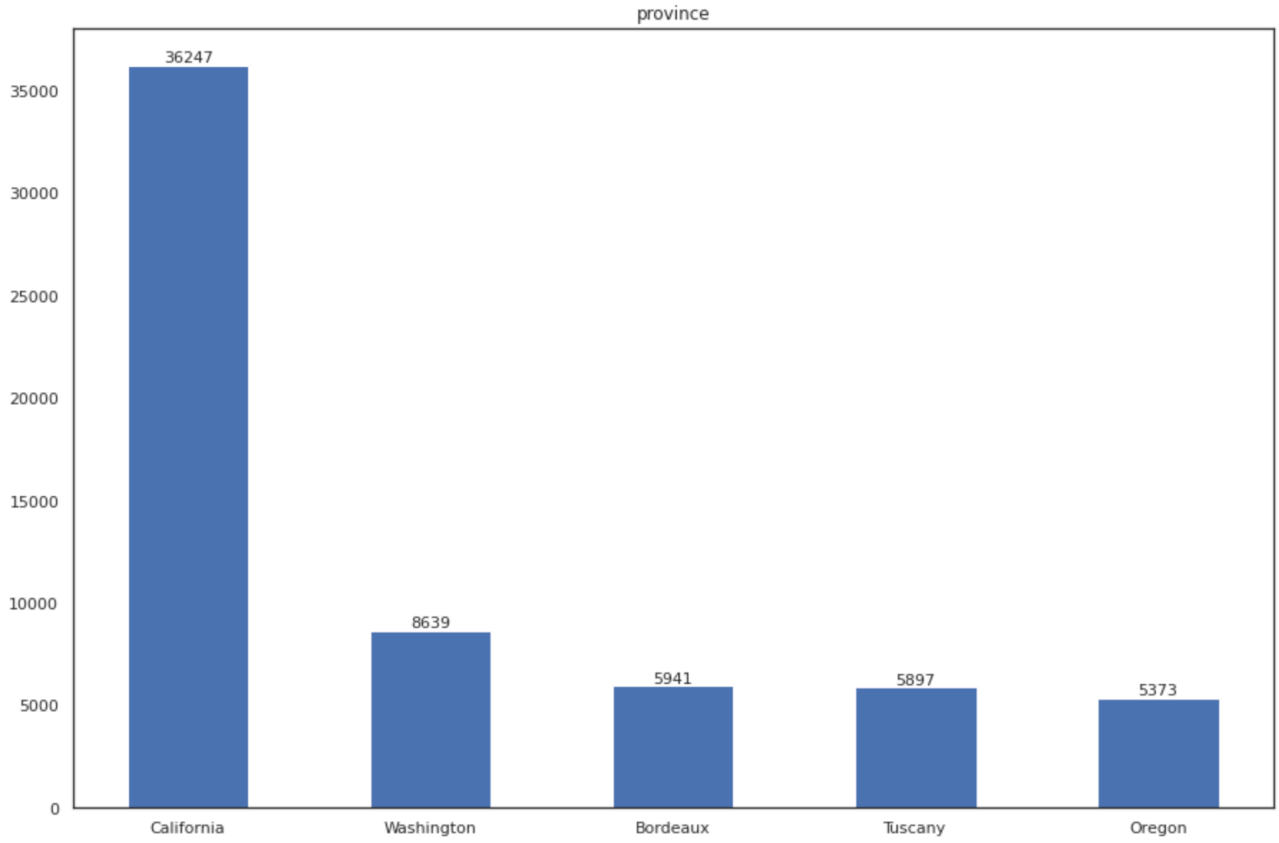
[39]:

```
i=0
for datax in data_all_wine:
    print("\n" + wine_name_list[i] + "的频数聚合分析:")

    data1_province=data1_number["province"].value_counts(sort=True)
    data1_province=data1_province.head(5)
    data1_province_name=data1_province.index.tolist()
    data1_province_num=data1_province.values

    index=np.arange(5)

    plt.figure(figsize=(15,10))
    plt.bar(index,data1_province_num, 0.5, label="num")
    plt.xticks(index,data1_province_name)
    for a,b in zip(index,data1_province_num):
        plt.text(a, b+0.05, '%.0f' % b, ha='center', va= 'bottom',fontsize=11)
    plt.title("province")

    plt.show()
    i+=1
```

winemag-data_first150k.csv的频数聚合分析:



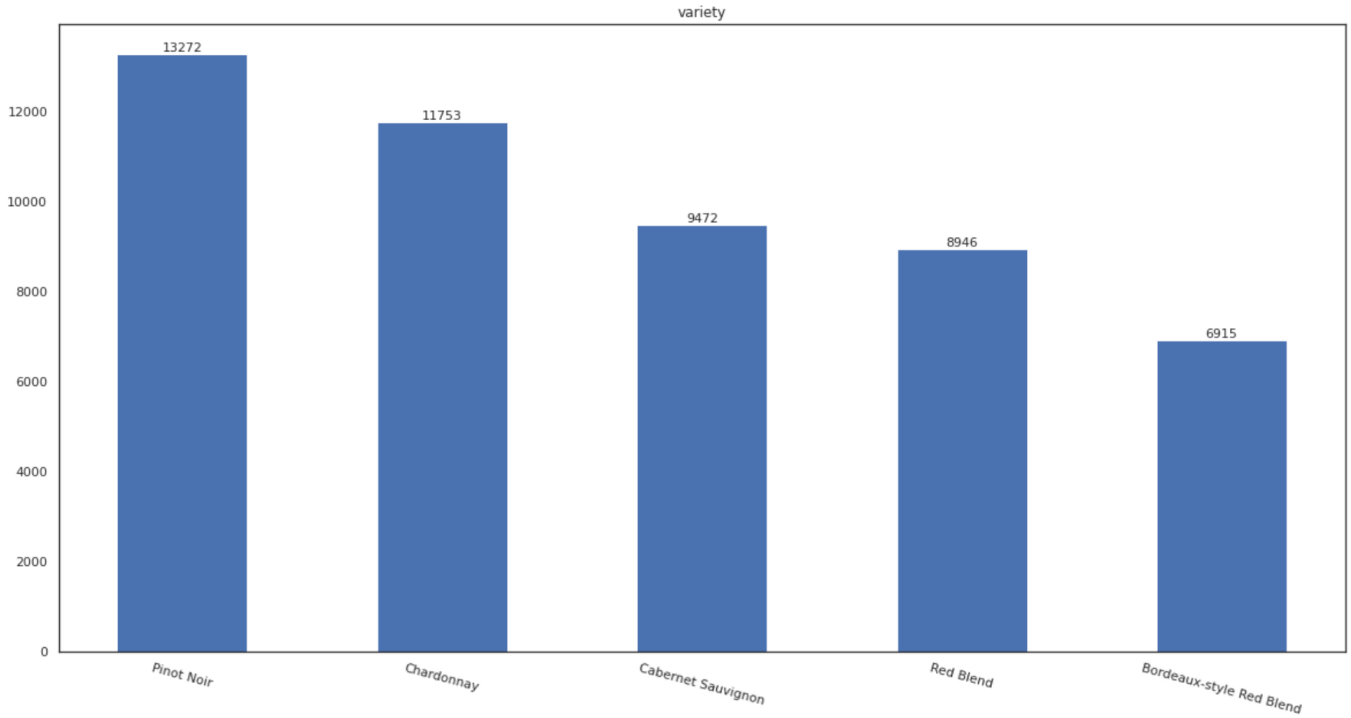
winemag-data-130k-v2.csv的频数聚合分析:



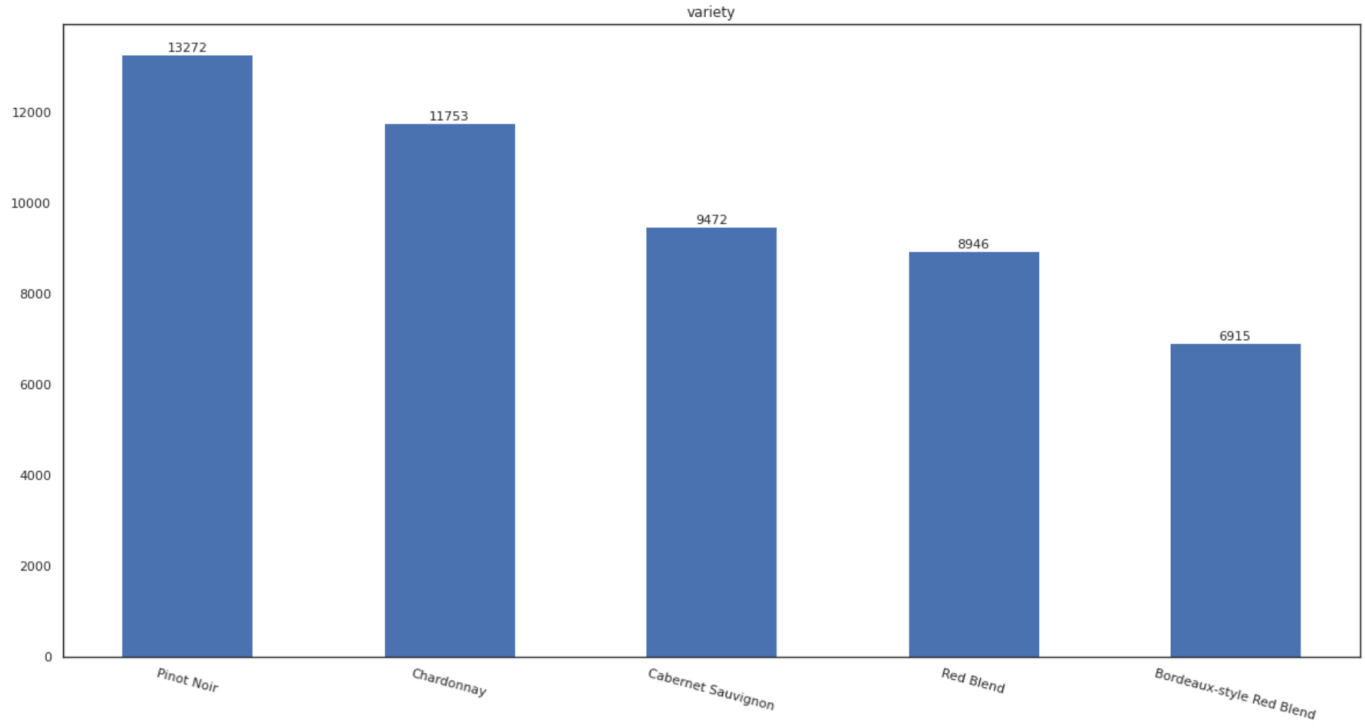
[41]:

```
i=0
for datax in data_all_wine:
    print("\n" + wine_name_list[i] + "的频数聚合分析:")

    data1_variety=data1_number["variety"].value_counts(sort=True)
    data1_variety=data1_variety.head(5)
    data1_variety_name=data1_variety.index.tolist()
    data1_variety_num=data1_variety.values

    index=np.arange(5)

    plt.figure(figsize=(20,10))
    plt.bar(index,data1_variety_num, 0.5, label="num")
    plt.xticks(index,data1_variety_name,rotation=-15)
    for a,b in zip(index,data1_variety_num):
        plt.text(a, b+0.05, '%.0f' % b, ha='center', va= 'bottom',fontsize=11)
    plt.title("variety")

    plt.show()
    i+=1
```

winemag-data_first150k.csv的频数聚合分析:



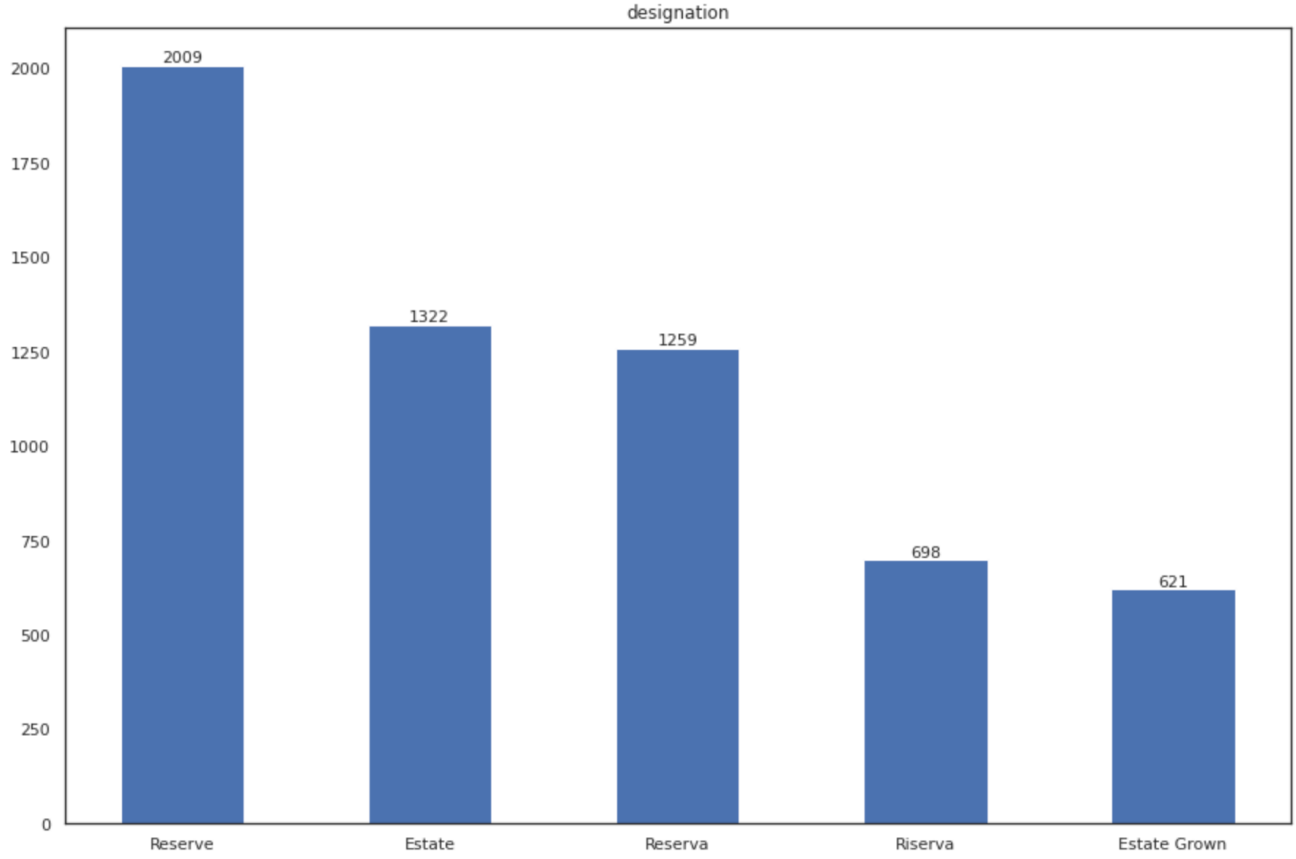
winemag-data-130k-v2.csv的频数聚合分析:



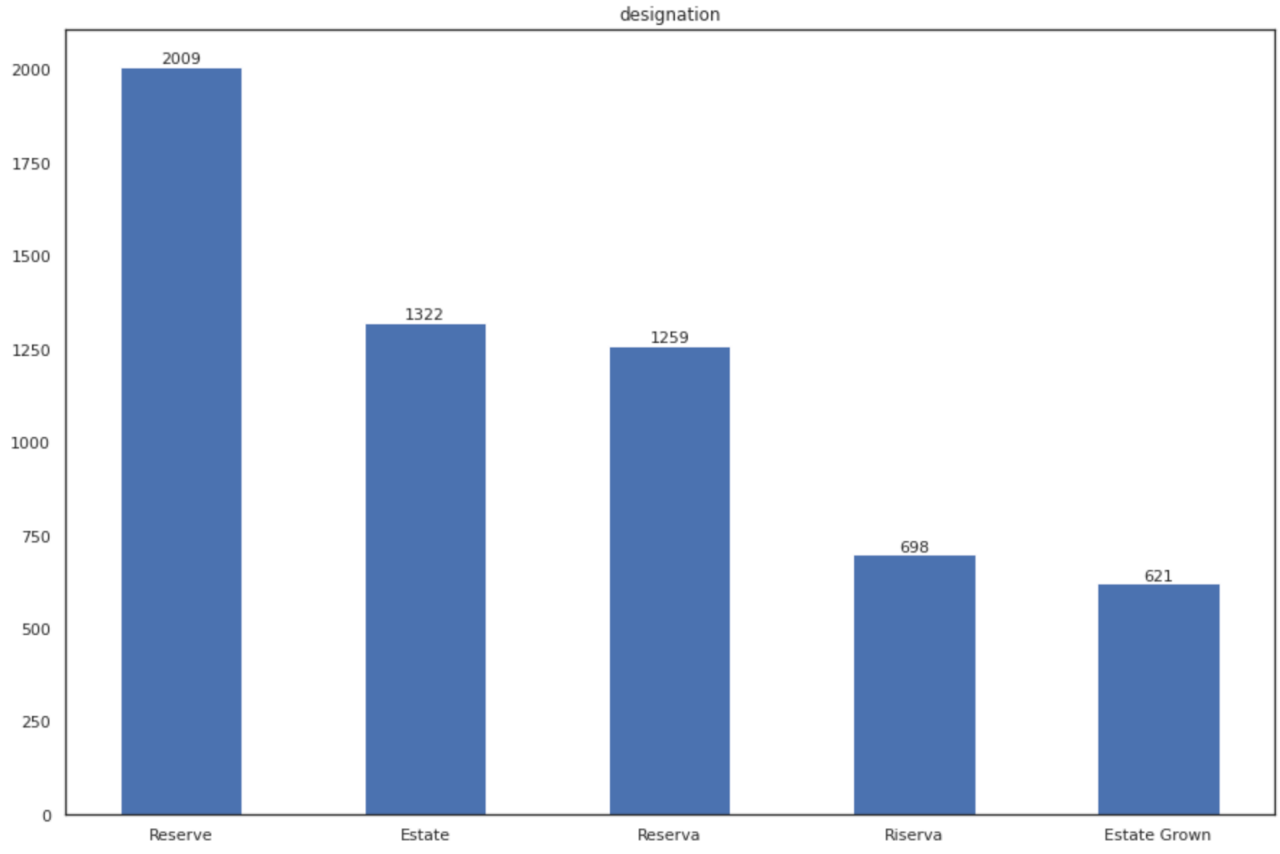
[43]:

```
i=0
for datax in data_all_wine:
    print("\n" + wine_name_list[i] + "的频数聚合分析:")

    data1_designation=data1_number["designation"].value_counts(sort=True)
    data1_designation=data1_designation.head(5)
    data1_designation_name=data1_designation.index.tolist()
    data1_designation_num=data1_designation.values

    index=np.arange(5)

    plt.figure(figsize=(15,10))
    plt.bar(index,data1_designation_num, 0.5, label="num")
    plt.xticks(index,data1_designation_name)
    for a,b in zip(index,data1_designation_num):
        plt.text(a, b+0.05, '%.0f' % b, ha='center', va= 'bottom',fontsize=11)
    plt.title("designation")

    plt.show()
    i+=1
```

winemag-data_first150k.csv的频数聚合分析：



winemag-data-130k-v2.csv的频数聚合分析：

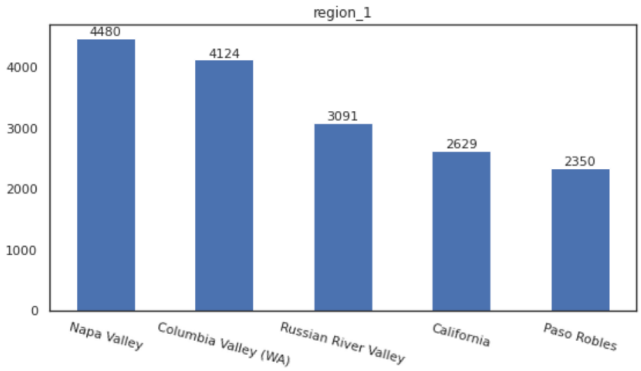


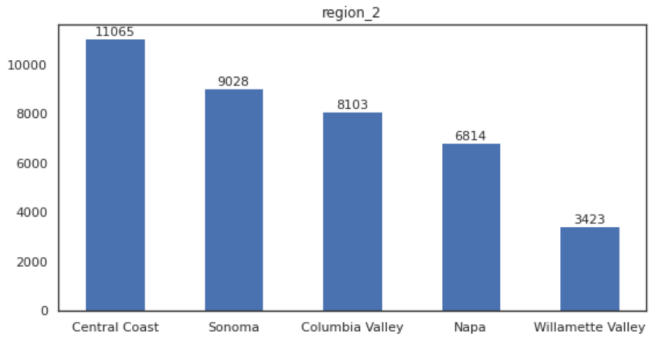
[38]:

```
i=0
for datax in data_all_wine:
    print("\n" + wine_name_list[i] + "的频数聚合分析:")

    data1_region_1=data1_number["region_1"].value_counts(sort=True)
    data1_region_1=data1_region_1.head(5)
    data1_region_1_name=data1_region_1.index.tolist()
    data1_region_1_num=data1_region_1.values

    data1_region_2=data1_number["region_2"].value_counts(sort=True)
    data1_region_2=data1_region_2.head(5)
    data1_region_2_name=data1_region_2.index.tolist()
    data1_region_2_num=data1_region_2.values

    index=np.arange(5)

    plt.figure(figsize=(20,10))
    ax5 = plt.subplot(2,2,1)
    ax6 = plt.subplot(2,2,2)
    plt.sca(ax5)
    plt.bar(index,data1_region_1_num, 0.5, label="num")
    plt.xticks(index,data1_region_1_name,rotation=-15)
    for a,b in zip(index,data1_region_1_num):
        plt.text(a, b+0.05, '%.0f' % b, ha='center', va= 'bottom',fontsize=11)
    plt.title("region_1")
```

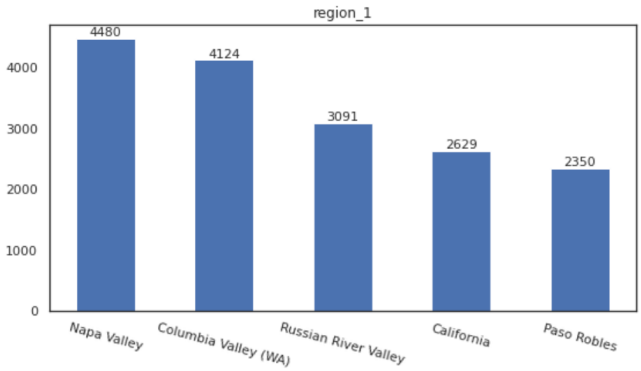
```
    plt.sca(ax6)
    plt.bar(index,data1_region_2_num, 0.5, label="num")
    plt.xticks(index,data1_region_2_name)
    for a,b in zip(index,data1_region_2_num):
        plt.text(a, b+0.05, '%.0f' % b, ha='center', va= 'bottom',fontsize=11)
    plt.title("region_2")

    plt.show()
    i+=1
```

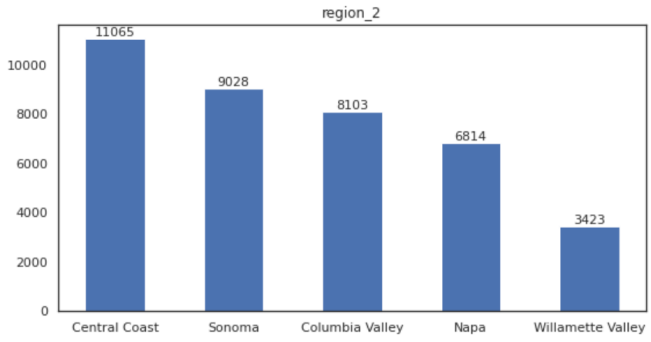
winemag-data_first150k.csv的频数聚合分析:





winemag-data-130k-v2.csv的频数聚合分析:





数值属性以及缺失值的个数

[4]:
```
winedata.head(5)
```

[4]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	variety	winery
0	0	US	This tremendous 100% varietal wine hails from ...	Martha's Vineyard	96	235.0	California	Napa Valley	Napa	Cabernet Sauvignon	Heitz
1	1	Spain	Ripe aromas of fig, blackberry and cassis are ...	Carodorum Selección Especial Reserva	96	110.0	Northern Spain	Toro	NaN	Tinta de Toro	Bodega Carmen Rodríguez
2	2	US	Mac Watson honors the memory of a wine once ma...	Special Selected Late Harvest	96	90.0	California	Knights Valley	Sonoma	Sauvignon Blanc	Macauley
3	3	US	This spent 20 months in 30% new French oak, an...	Reserve	96	65.0	Oregon	Willamette Valley	Willamette Valley	Pinot Noir	Ponzi
4	4	France	This is the top wine from La Bégude, named aft...	La Brûlade	95	66.0	Provence	Bandol	NaN	Provence red blend	Domaine de la Bégude

[24]:
```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.spatial.distance import pdist
from math import ceil
import numpy as np
%matplotlib inline

data1_wine_review = pd.read_csv("../input/wine-reviews/winemag-data_first150k.csv", encoding="utf-8")[
    ["price", "points"]]
data2_wine_review = pd.read_csv("../input/wine-reviews/winemag-data-130k-v2.csv", encoding="utf-8")[
    ["price", "points"]]
data_all_wine = [data1_wine_review, data2_wine_review]
wine_name_list = ["winemag-data_first150k.csv", "winemag-data-130k-v2.csv"]


def fiveNumber(nums):

    Minimum=min(nums)
    Maximum=max(nums)
    Q1=np.percentile(nums,25)
    Median=np.median(nums)
    Q3=np.percentile(nums,75)

    IQR=Q3-Q1
    lower_limit=Q1-1.5*IQR #下限值
    upper_limit=Q3+1.5*IQR #上限值

    return "Minimum: "+str(Minimum) +',  ' + 'Q1: ' +str(Q1) +',  ' + 'Median: ' +str(Median) + ',  ' + 'Q3: ' +s
```

```
print("Wine Reviews数据集的数值属性有：points和price")
i = 0
attribute_list = ["price", "points"]
for datax in data_all_wine:
    print("\n" + wine_name_list[i] + "的五数概括：")
    d = pd.DataFrame(data=datax["price"]) #price属性有空值,转成DataFrame格式处理该空值
    d=d.dropna(axis=0, how='any')
    d=d.values
    d=d.flatten()
    m=fiveNumber(d)
    points_five1=fiveNumber(datax["points"])
    print("points缺省值数量："+str(datax[["points"]].isnull().sum()[0])+ ";   五数概括："+str(points_five1)+\
          "\nprice缺省值数量："+str(datax[["price"]].isnull().sum()[0])+";   五数概括："+str(m))
    i+=1
```

Wine Reviews数据集的数值属性有：points和price

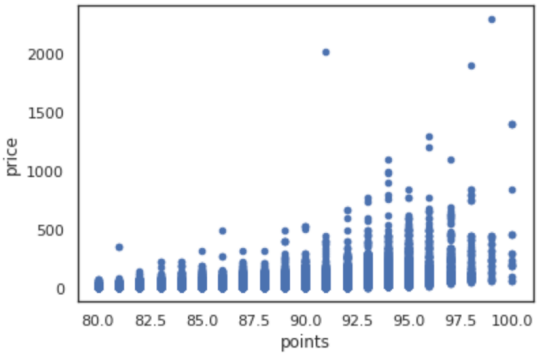
winemag-data_first150k.csv的五数概括：
points缺省值数量：0； 五数概括：Minimum: 80, Q1: 86.0, Median: 88.0, Q3: 90.0, Maximum: 100
price缺省值数量：13695； 五数概括：Minimum: 4.0, Q1: 16.0, Median: 24.0, Q3: 40.0, Maximum: 2300.0

winemag-data-130k-v2.csv的五数概括：
points缺省值数量：0； 五数概括：Minimum: 80, Q1: 86.0, Median: 88.0, Q3: 91.0, Maximum: 100
price缺省值数量：8996； 五数概括：Minimum: 4.0, Q1: 17.0, Median: 25.0, Q3: 42.0, Maximum: 3300.0

数据可视化分析

[4]:
```
#葡萄酒价格与评分间关系（散点图）
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="white", color_codes=True)
%matplotlib inline

winedata.plot(kind="scatter", x="points", y="price")
```

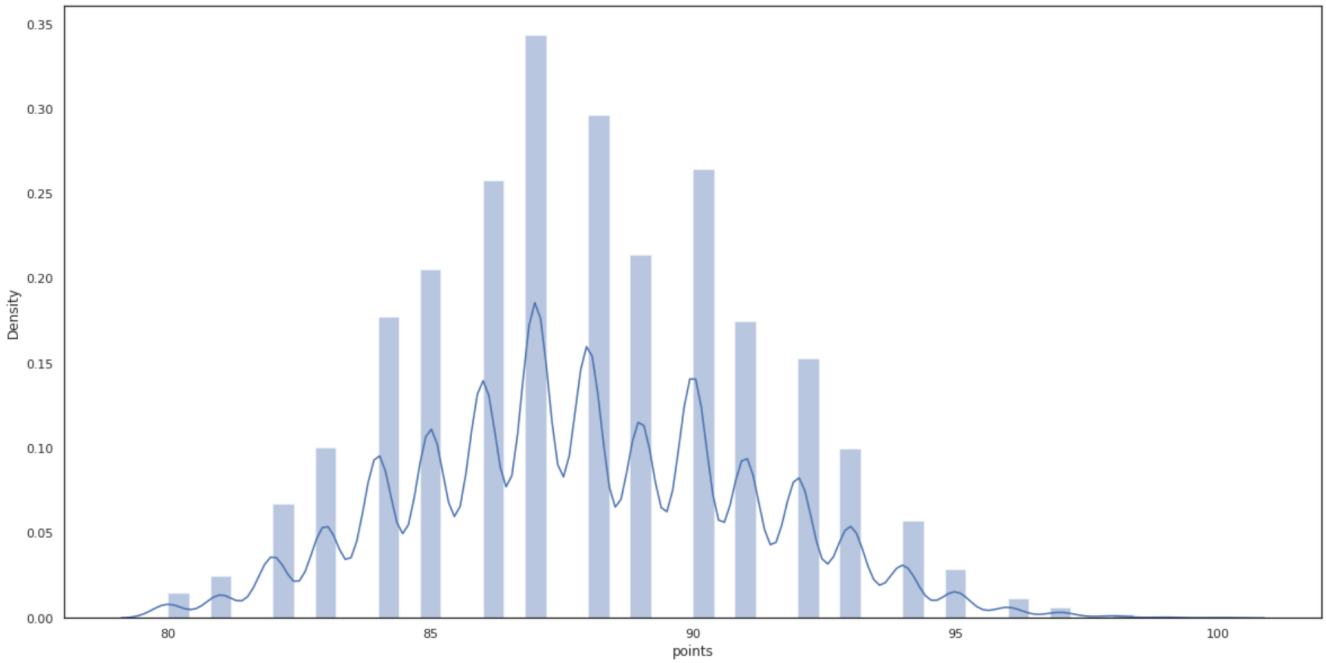
[4]: <AxesSubplot:xlabel='points', ylabel='price'>



直方图检查数据分布

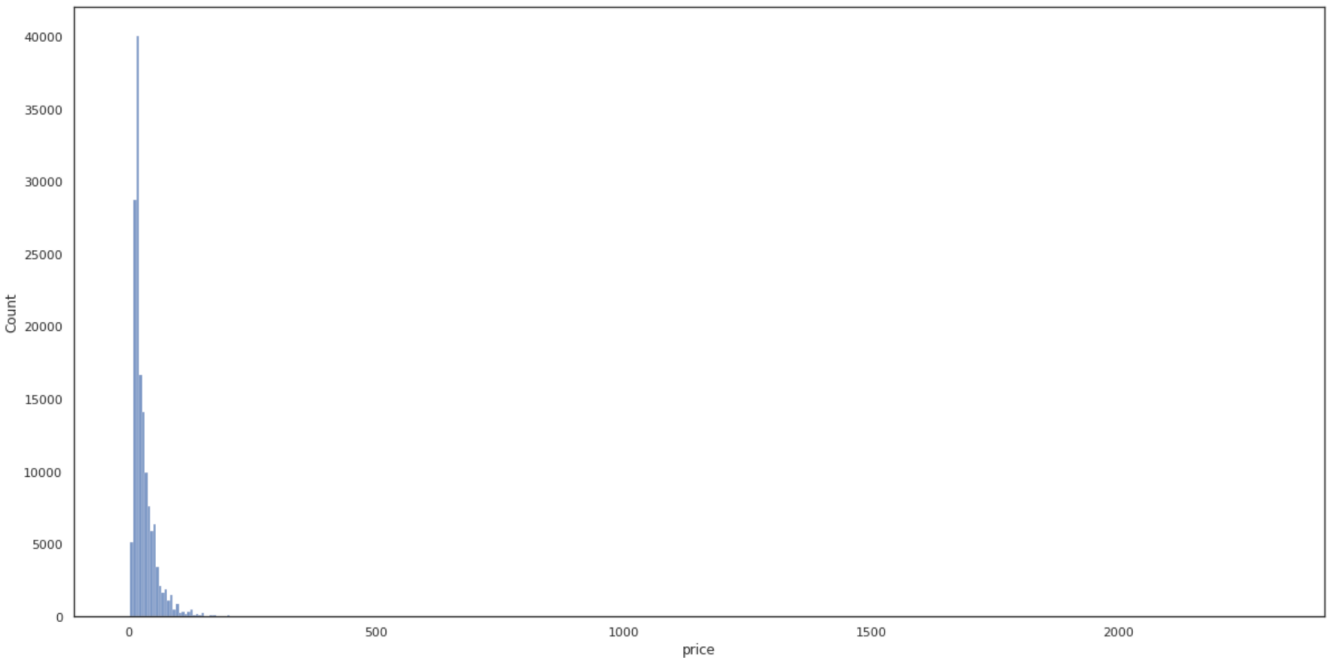
[49]:
```
#葡萄酒评分直方图分布，删除缺失数据后进行绘制
point1=winedata['points'].dropna()
plt.figure(figsize=(20,10))
sns.distplot(point1)
```

[49…] <AxesSubplot:xlabel='points', ylabel='Density'>




```
#葡萄酒价格直方图分布，删除缺失数据后进行绘制
price1=winedata['price'].dropna()
plt.figure(figsize=(20,10))
sns.histplot(price1,bins=400)
```

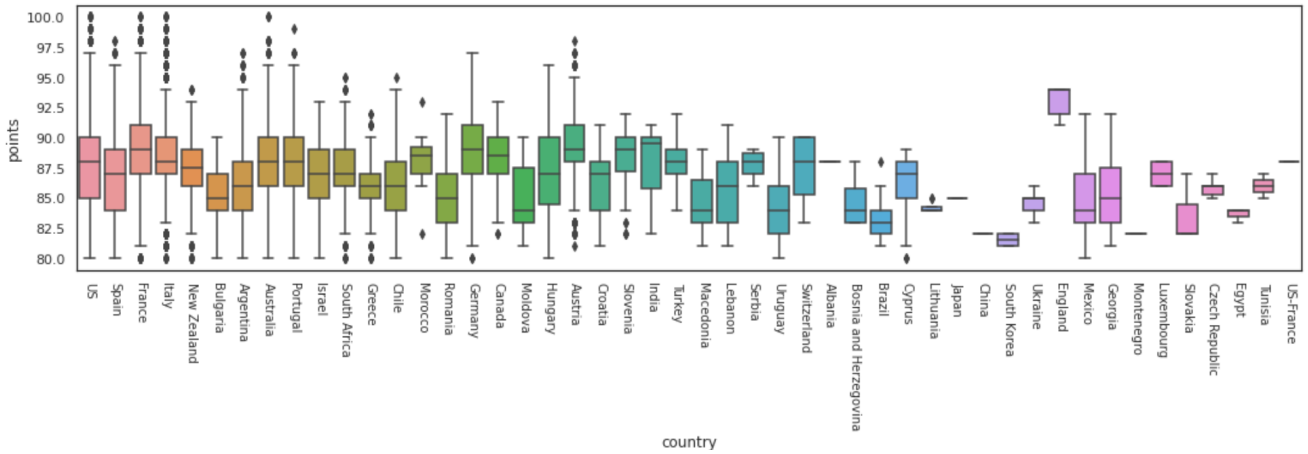
[52…] <AxesSubplot:xlabel='price', ylabel='Count'>



可以观察到，葡萄酒评分的数据比较集中，接近正态分布。葡萄酒价格的数据集中在500以下，结合散点图可以观察到存在一些1000～2000的较大的数据，但是数量很少，有时可以当作异常数据处理。

```
#葡萄酒评分按国家分类的箱图
fig=plt.figure(figsize=(18,4))
plt.tick_params(axis='x',labelsize=10)
plt.xticks(rotation=-90)
sns.boxplot(x='country',y='points',data=winedata)
```

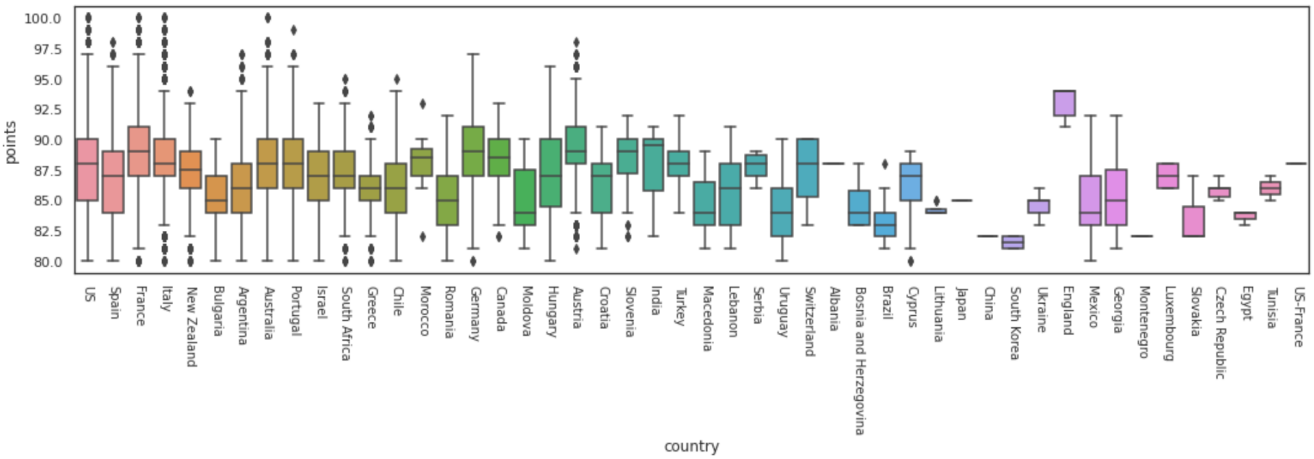
[9]: <AxesSubplot:xlabel='country', ylabel='points'>



[5]:

```
#葡萄酒价格按国家分类的箱图
fig=plt.figure(figsize=(18,4))
plt.tick_params(axis='x',labelsize=10)
plt.xticks(rotation=-90)
sns.boxplot(x='country',y='points',data=winedata)
```

[5]: <AxesSubplot:xlabel='country', ylabel='points'>



数据缺失的处理

由于上述直方图和箱图是已经删除缺失值状态下绘制的，下面分别通过用最高评率值来填补缺失值、通过属性的相关关系来填补缺失值、通过数据对象之间的相似性来填补缺失值三种策略对缺失值进行处理。

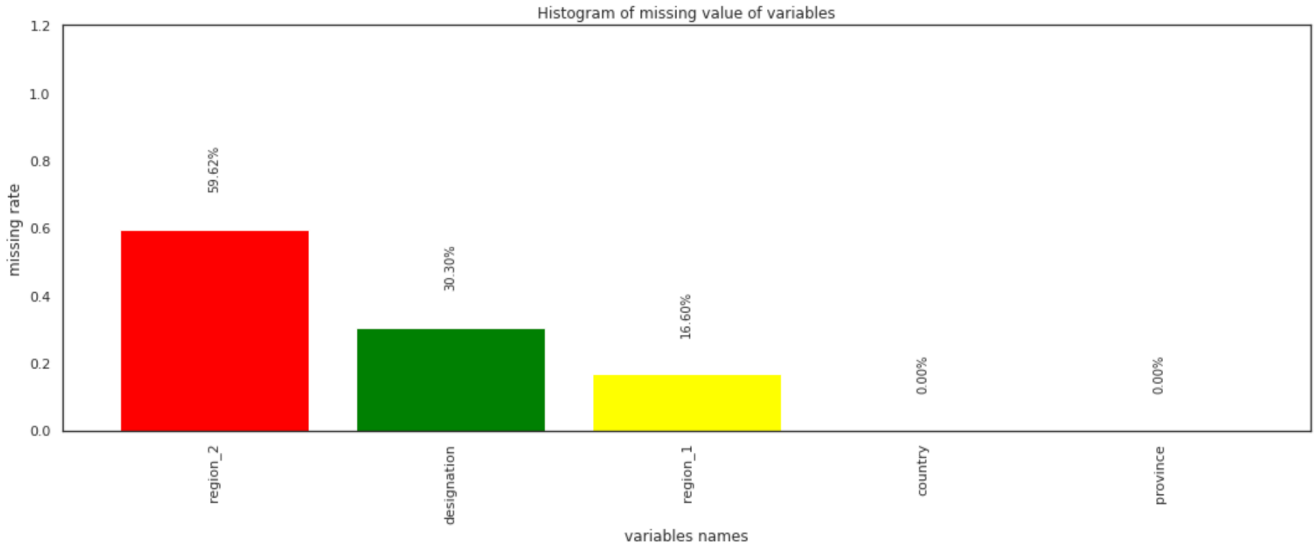
[7]:
```
# 判断各变量中是否存在缺失值
winedata.isnull().any(axis = 0)
# 各变量中缺失值的数量
winedata.isnull().sum(axis = 0)
# 各变量中缺失值的比例
winedata.isnull().sum(axis = 0)/winedata.shape[0]
```

[7]:
```
Unnamed: 0     0.000000
country        0.000033
description    0.000000
designation    0.303021
points         0.000000
price          0.090737
province       0.000033
region_1       0.166037
region_2       0.596151
variety        0.000000
winery         0.000000
dtype: float64
```

可视化缺失值的分布

[56]:
```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.spatial.distance import pdist
from math import ceil
import numpy as np
%matplotlib inline

# 统计缺失值数量
missing=winedata.isnull().sum().reset_index().rename(columns={0:'missNum'})
# 计算缺失比例
missing['missRate']=missing['missNum']/winedata.shape[0]
# 按照缺失率排序显示
miss_analy=missing[missing.missRate>0].sort_values(by='missRate',ascending=False)
# miss_analy 存储的是每个变量缺失情况的数据框

import matplotlib.pyplot as plt
import pylab as pl

fig = plt.figure(figsize=(18,6))
plt.bar(np.arange(miss_analy.shape[0]), list(miss_analy.missRate.values), align = 'center'
    ,color=['red','green','yellow','steelblue'])

plt.title('Histogram of missing value of variables')
plt.xlabel('variables names')
plt.ylabel('missing rate')
# 添加x轴标签，并旋转90度
plt.xticks(np.arange(miss_analy.shape[0]),list(miss_analy['index']))
pl.xticks(rotation=90)
# 添加数值显示
```

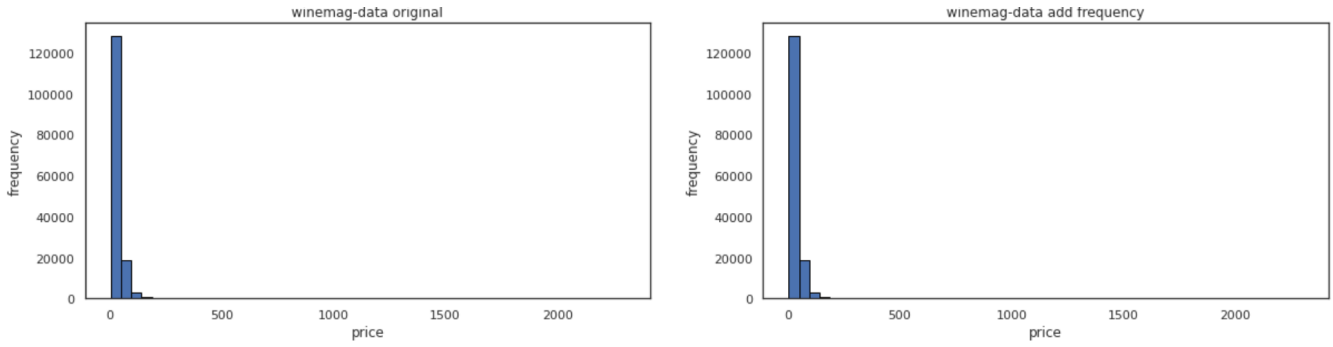
```
# 添加数值显示
for x,y in enumerate(list(miss_analy.missRate.values)):
    plt.text(x,y+0.12,'{:.2%}'.format(y),ha='center',rotation=90)
plt.ylim([0,1.2])

plt.show()
```



[60]:

```
#用最高频率值来填补缺失值
df=winedata
df.fillna(value = {'price': df['price'].mode()[0], # 使用价格的众数替换缺失价格
}, inplace = True )
price2=df['price']
plt.figure(figsize=(20,10))
ax1=plt.subplot(2,2,1)
ax2=plt.subplot(2,2,2)

plt.sca(ax1)
plt.hist(price1, bins=50,  edgecolor = 'black', histtype='bar', align='mid', orientation='vertical')
plt.xlabel("price")
plt.ylabel("frequency")
plt.title("winemag-data original")

plt.sca(ax2)
plt.hist(price2, bins=50,  edgecolor = 'black', histtype='bar', align='mid', orientation='vertical')
plt.xlabel("price")
plt.ylabel("frequency")
plt.title("winemag-data add frequency")
```

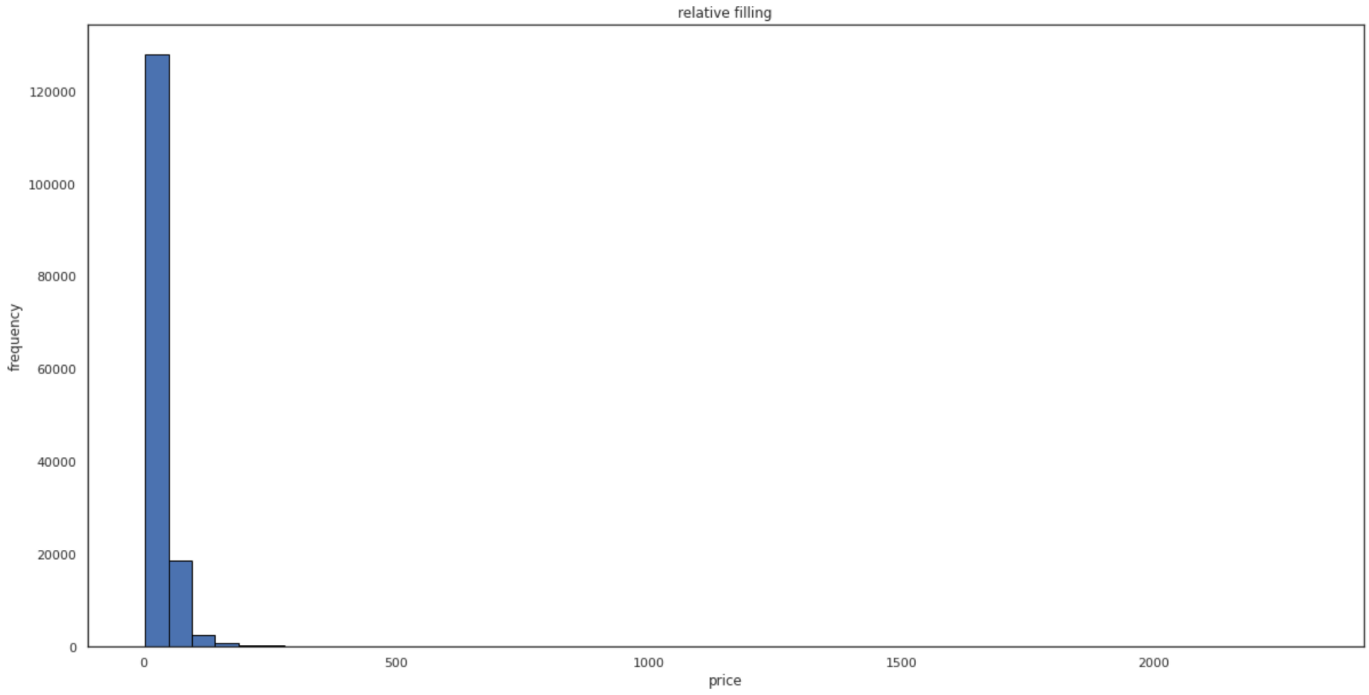
[60… Text(0.5, 1.0, 'winemag-data add frequency')



```
#用属性的相关关系填补缺失值
#由之前的散点图得知，葡萄酒评分与价格间存在一定的属性关系，因此可以用相关关系来填补缺失值
data4_1=winedata
data4_1=data4_1.dropna(axis=0, how='any')
points1=data4_1["points"]
price1=data4_1["price"]

cos1 = np.vstack([points1,price1])
p1 = 1 - pdist(cos1,'cosine')
print("winemag-data_first150k.csv: PLCC="+str(points1.corr(price1,method="pearson"))+"  Cosine similarity="+str(p

print("可见两属性相关性并不是很强，但是可以观察到，评分高的葡萄酒相对来说价格会高")
#以每个度数的中值来填充结果
xx = winedata[["points", "price"]].groupby(by="points").median()
xx=xx.values
xx=xx.flatten()

data_add1=winedata
dataadd_g1=pd.DataFrame()

for i in range(80,101):
    data_ad1=data_add1.loc[data_add1['points'].isin([i])].fillna(xx[i-80])
    if(i==80):
        data_add_g1=data_ad1
    else:
        data_add_g1=pd.concat([data_add_g1,data_ad1],axis=0)
prices1=data_add_g1[["price"][0]]
```

```
plt.figure(figsize=(20,10))
plt.hist(prices1, bins=50,  edgecolor = 'black', histtype='bar', align='mid', orientation='vertical')
plt.xlabel("price")
plt.ylabel("frequency")
plt.title("relative filling")

plt.show()
```

```
winemag-data_first150k.csv: PLCC=0.43368208558406635  Cosine similarity=[0.82371587]
可见两属性相关性并不是很强，但是可以观察到，评分高的葡萄酒相对来说价格会高
```



[64]:
```
#通过数据对象之间的相似性来填补缺失值
#通过观察数据可得，缺失价格的葡萄酒大多是France和Italy两个国家，因此采用两个国家众数填充
from scipy import stats
France = winedata['price']
mode = stats.mode(France)
print(mode[0][0])

Italy = winedata['price']
mode2 = stats.mode(Italy)
print(mode2[0][0])
```

```
20.0
20.0
```

经过计算可以发现，两个国家的葡萄酒价格众数都为20，因此可以直接填充众数20作为对象相似性填充的价格数值

[66]:
```
price3=winedata['price'].fillna(winedata['price'].mode())
plt.figure(figsize=(20,10))
plt.title("similarity filling")
sns.histplot(price3,bins=400)
```

[66…
```
<AxesSubplot:title={'center':'similarity filling'}, xlabel='price', ylabel='Count'>
```