In [236]:

```
## importing required library
import warnings

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

warnings.filterwarnings("ignore")
%matplotlib inline
pd.set_option("display.max_columns", None)
```

In [237]:

```
wine = pd.read_csv("wine reviews_small.csv")
```

In [238]:

```
## Droping unwanted column
wine.drop("Unnamed: 0", axis=1, inplace=True)
```

EDA

In [239]:

```
for feature in wine.columns:
    print("The unique values in ", feature, "is", wine[feature].nunique())
```

```
The unique values in  country is 41
The unique values in  description is 29169
The unique values in  designation is 13454
The unique values in  points is 21
The unique values in  price is 242
The unique values in  province is 330
The unique values in  region_1 is 964
The unique values in  region_2 is 17
The unique values in  taster_name is 19
The unique values in  taster_twitter_handle is 15
The unique values in  title is 29107
The unique values in  variety is 478
The unique values in  winery is 9881
```

In [240]:

```
## Creating variable having all the numerical feature
numeric_feature = [feature for feature in wine.columns if wine[feature].dtype != "O"]
numeric_feature
```

Out[240]:

```
['points', 'price']
```

In [241]:

```
## Creating variable having catogorical Feature names
cat_feature = [feature for feature in wine.columns if wine[feature].dtype == "O"]
cat_feature
```

Out[241]:

```
['country',
 'description',
```

```
 'designation',
 'province',
 'region_1',
 'region_2',
 'taster_name',
 'taster_twitter_handle',
 'title',
 'variety',
 'winery']
```

In [242]:

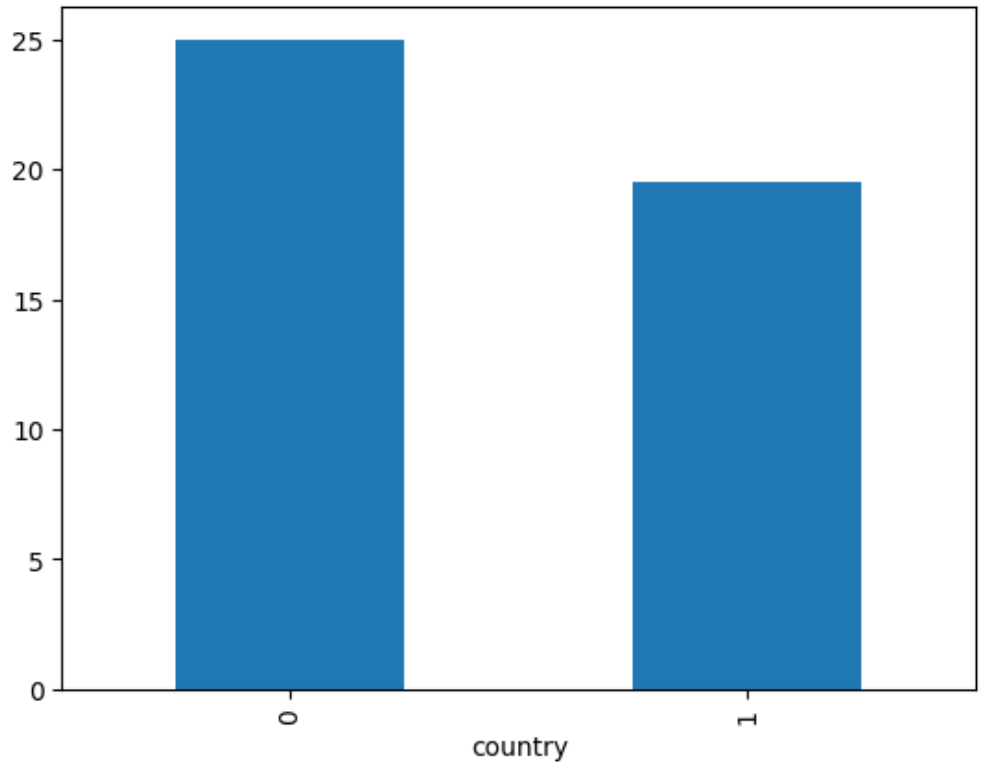
```
## feature with Nan or missing Values
feature_with_nan = [
    feature for feature in wine.columns if wine[feature].isnull().sum() > 0
]
feature_with_nan
```

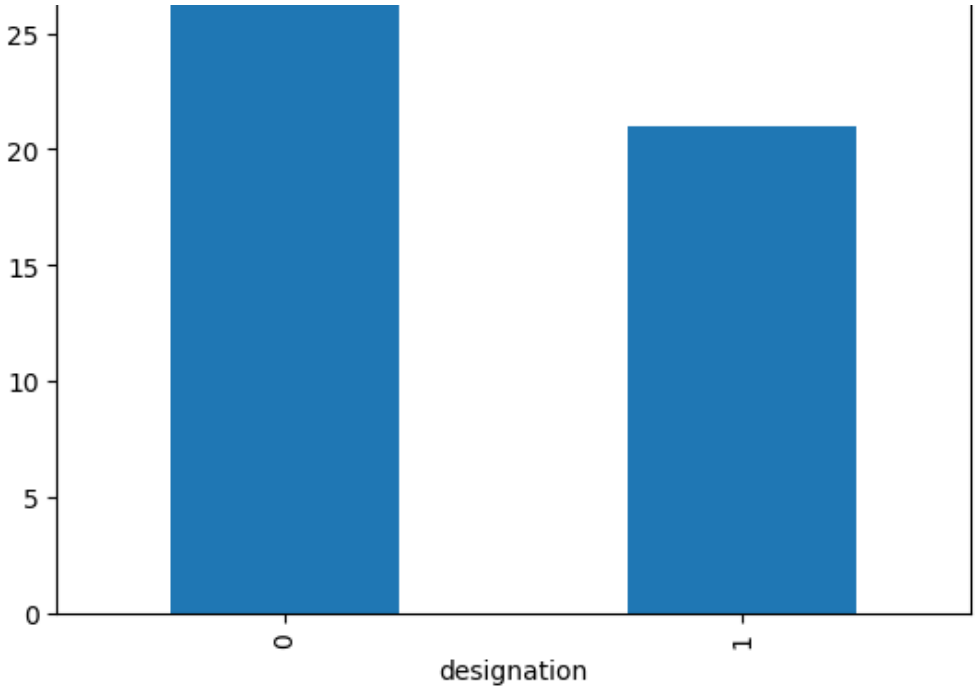
Out[242]:

```
['country',
 'designation',
 'price',
 'province',
 'region_1',
 'region_2',
 'taster_name',
 'taster_twitter_handle']
```

In [243]:

```
## checking the relation between missing values and dependent feature
feature_with_nans = ["country", "designation"]
data = wine.copy()
for feature in feature_with_nans:
    data[feature] = np.where(data[feature].isnull(), 1, 0)
    data.groupby(feature)["price"].median().plot.bar()
    plt.show()
```





The Nan values Have no realtion with output feature so we can replace the values by Median

In [244]:

```
wine.isnull().sum()
## data having more missing values so we drop the some of unwanted columns
data = wine.copy()
data.drop(["region_2", "taster_twitter_handle", "region_1"], axis=1, inplace=True)
```

In [245]:

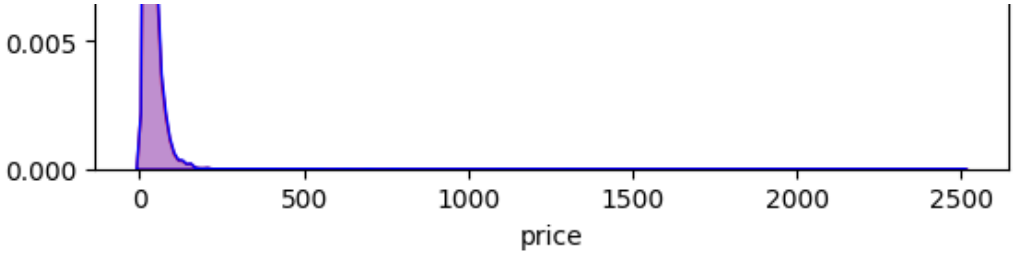
```
## price feature missing values is replaced by median
data["price"] = data["price"].fillna(data["price"].median())
```

In [246]:

```
## their is no change in distribution
sns.kdeplot(data["price"], color="r", label="without nan", shade=True)
sns.kdeplot(wine["price"], color="b", label="with nan", shade=True)
```

Out[246]:

```
<AxesSubplot:xlabel='price', ylabel='Density'>
```



In [247]:

```
data.dropna(subset=["country"], inplace=True)
```

In [248]:

```
## missing values is replace as 'Missing'
data["designation"].fillna("Missing", inplace=True)
data["taster_name"].fillna("Missing", inplace=True)
data.drop_duplicates().isnull().sum()
```

Out[248]:

```
country        0
description    0
designation    0
points         0
price          0
province       0
taster_name    0
title          0
variety        0
winery         0
dtype: int64
```

In [249]:

```
## exporting the Clean Csv
data.to_csv("wine_review_clean.csv")
```

In [250]:

```
## Now the Data is clean so WE CAN START ANALYSISNG
## univarient analysis
```

univarient analysis

In [251]:

```
data.head()
```

Out[251]:

	country	description	designation	points	price	province	taster_name	title	variety	winery
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	25.0	Sicily & Sardinia	Kerin O'Keefe	Nicosia 2013 Vulkà Bianco (Etna)	White Blend	Nicosia
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	Roger Voss	Quinta dos Avidagos 2011 Avidagos Red (Douro)	Portuguese Red	Quinta dos Avidagos
2	US	Tart and snappy, the flavors of lime flesh and	Missing	87	14.0	Oregon	Paul Gregutt	Rainstorm 2013 Pinot Gris (Willamette	Pinot Gris	Rainstorm

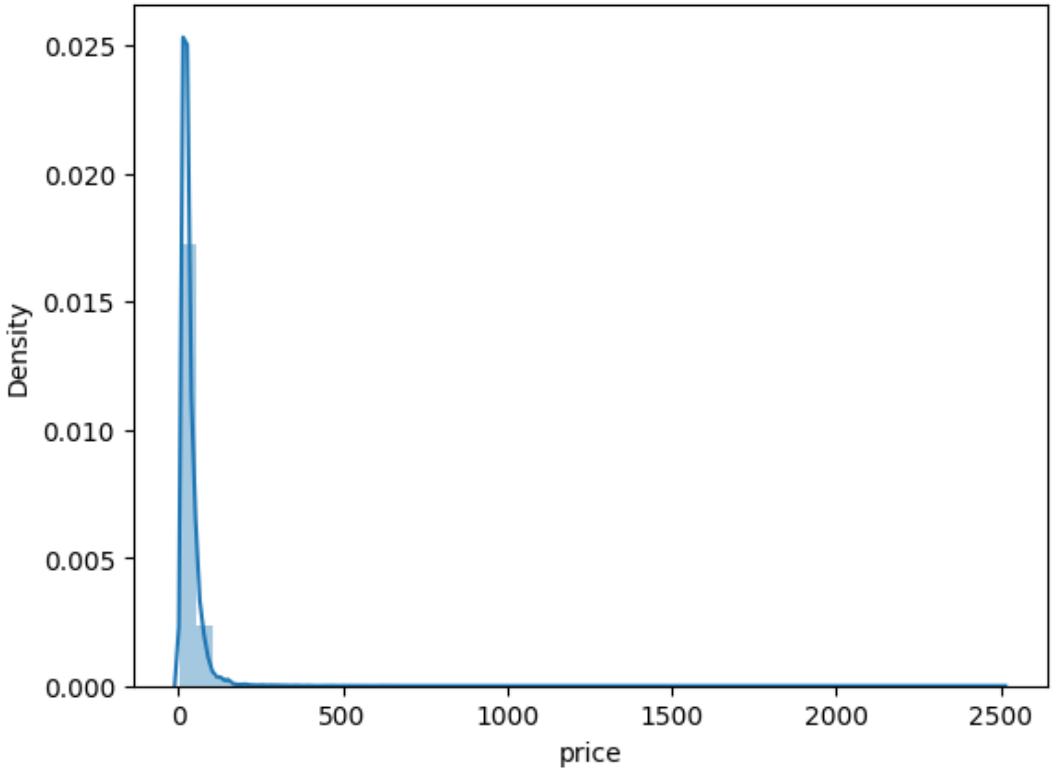
	country	description	designation	points	price	province	taster_name	Valley)	variety	winery
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Alexander Peartree	St. Julian 2013 Reserve Late Harvest Riesling ...	Riesling	St. Julian
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Paul Gregutt	Sweet Cheeks 2012 Vintner's Reserve Wild Child...	Pinot Noir	Sweet Cheeks

In [252]:

```
# distribution of numeric variable
sns.distplot(data["price"])
```

Out[252]:

```
<AxesSubplot:xlabel='price', ylabel='Density'>
```



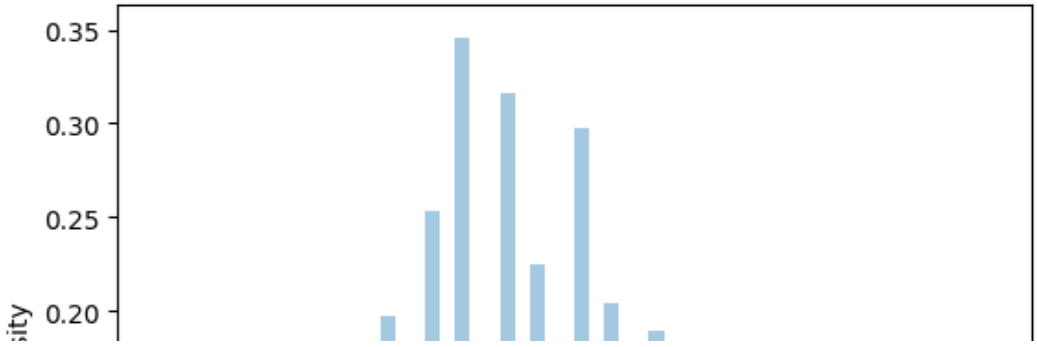
The feature price is right skewed

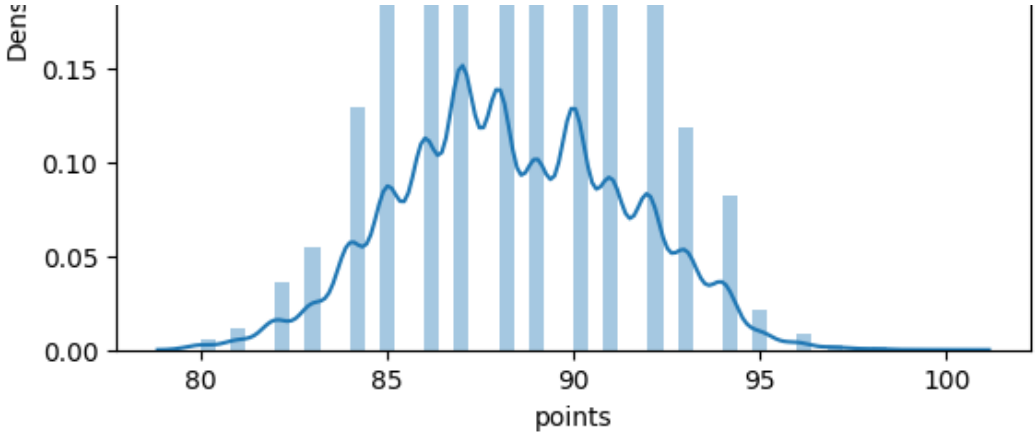
In [253]:

```
sns.distplot(data["points"])
```

Out[253]:

```
<AxesSubplot:xlabel='points', ylabel='Density'>
```

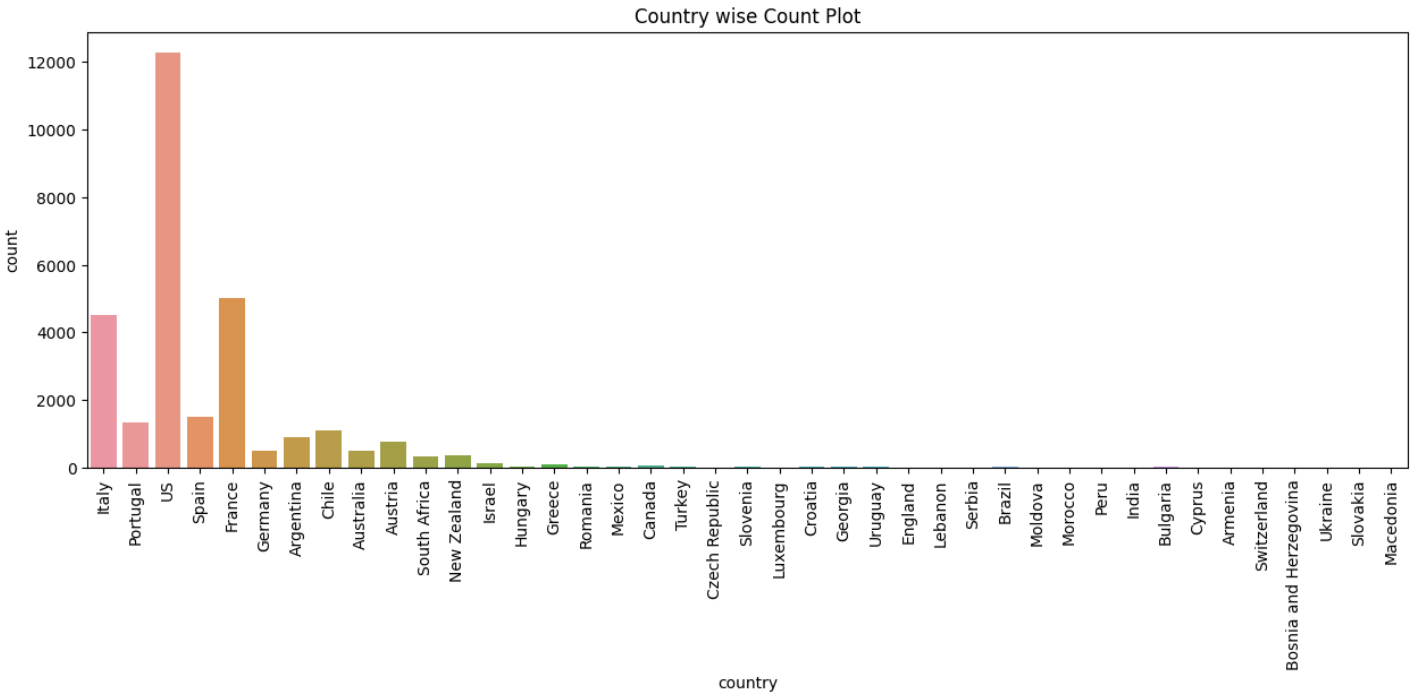




the points feature more or less follows normal distribution

In [254]:

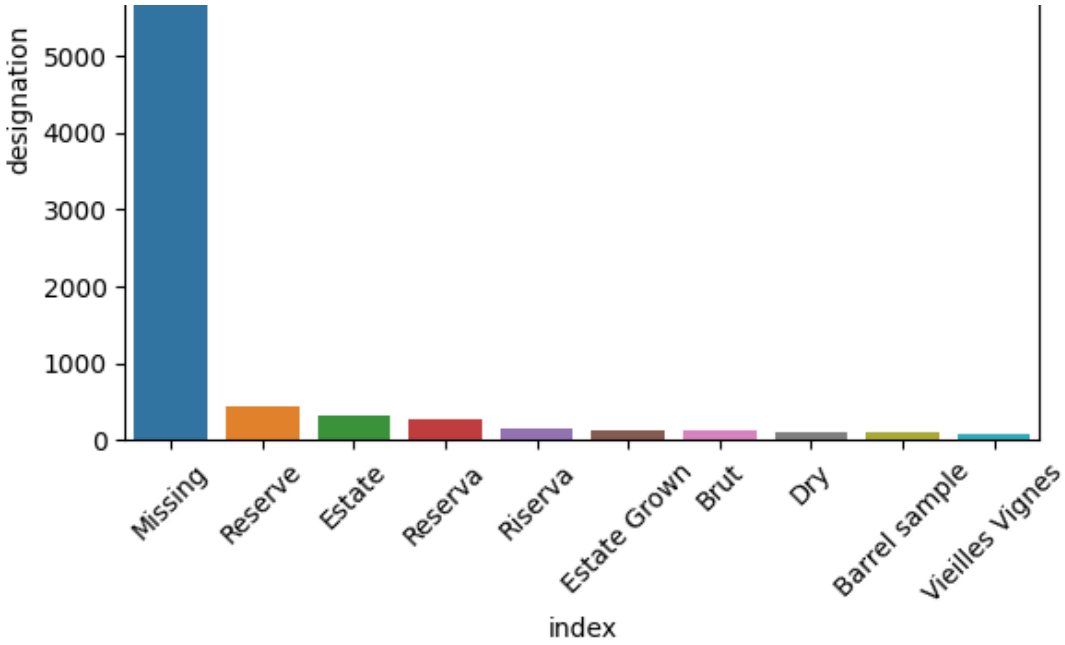
```
# country feature count plot
plt.figure(figsize=(15, 5))
sns.countplot(data["country"]).set_title("Country wise Count Plot")
a = plt.xticks(rotation=90)
```



In this reviews the most of the wine is from United States over 12000 and after that France

In [255]:

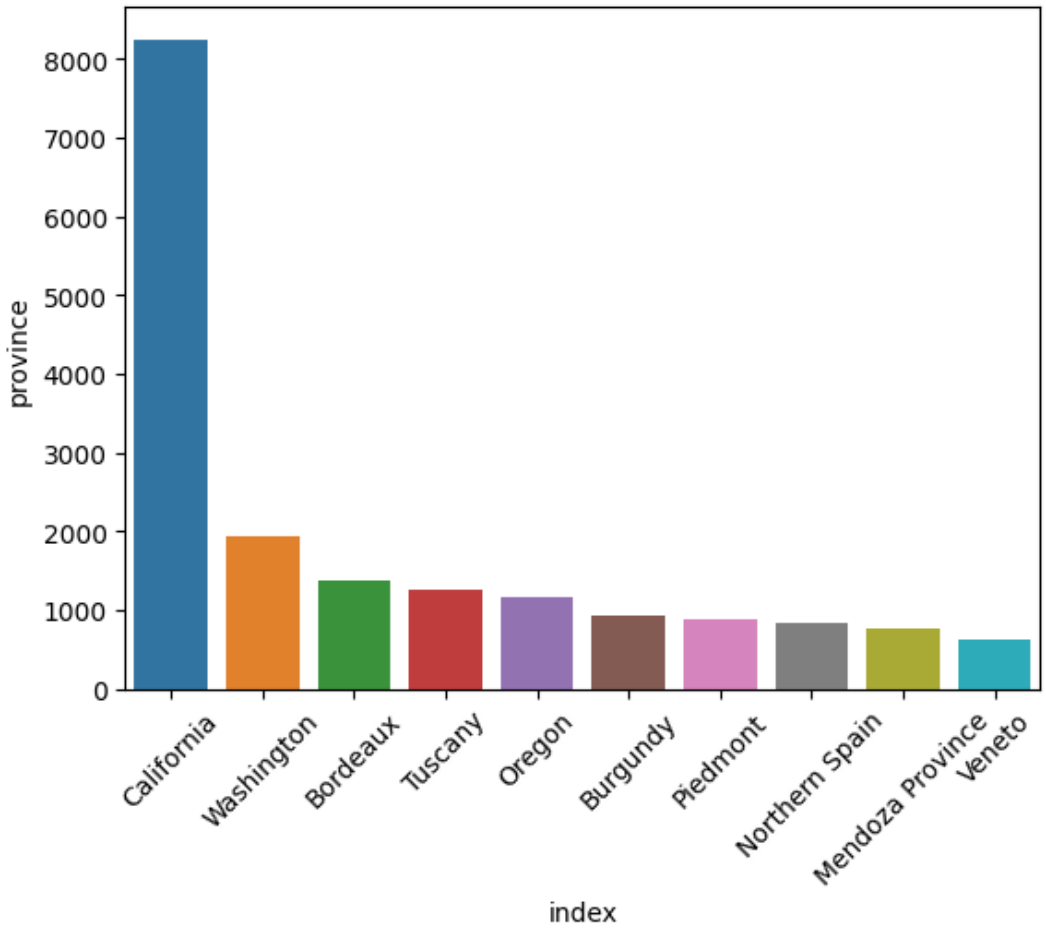
```
# Designation: the vineyard within the winery where the grapes that made the wine are fr
om
desig = data["designation"].value_counts().reset_index()
sns.barplot(data=desig.head(10), x="index", y="designation")
a = plt.xticks(rotation=45)
```



So we can see the above plot it shows the designation has more missing values after tha 'reserver designation' is more in the review Dataset

In [256]:

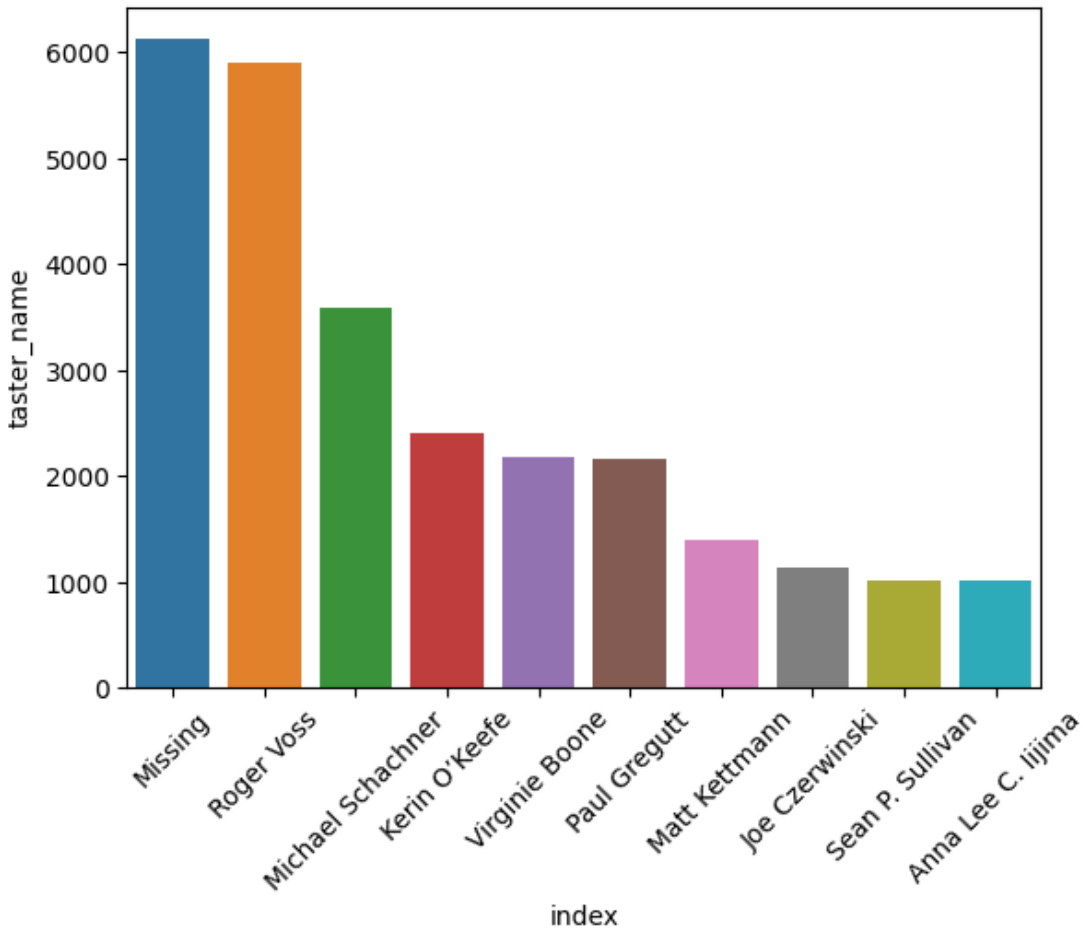
```
prov = data["province"].value_counts().reset_index()
sns.barplot(data=prov.head(10), x="index", y="province")
a = plt.xticks(rotation=45)
```



The Data set of wine reviews the state california is having more counts and next is Washington

In [257]:

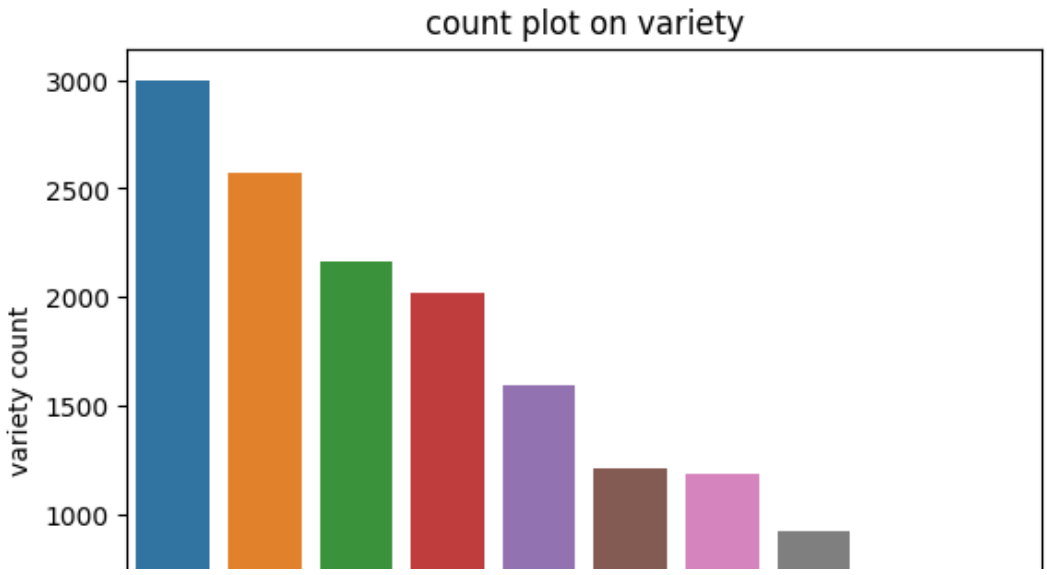
```
taster_name = data["taster_name"].value_counts().reset_index()
sns.barplot(data=taster_name.head(10), x="index", y="taster_name")
a = plt.xticks(rotation=45)
```

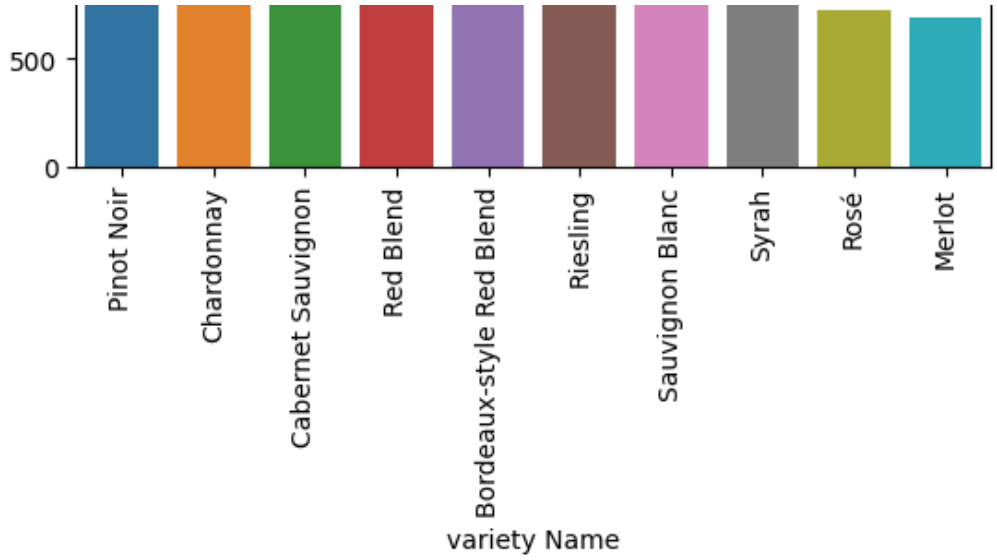


The plot show that many people tast the wine are not provide the name so many values are missing

In [258]:

```
variety = data["variety"].value_counts().reset_index()
sns.barplot(data=variety.head(10), x="index", y="variety").set_title(
    "count plot on variety"
)
plt.xlabel("variety Name")
plt.ylabel("variety count")
a = plt.xticks(rotation=90)
```

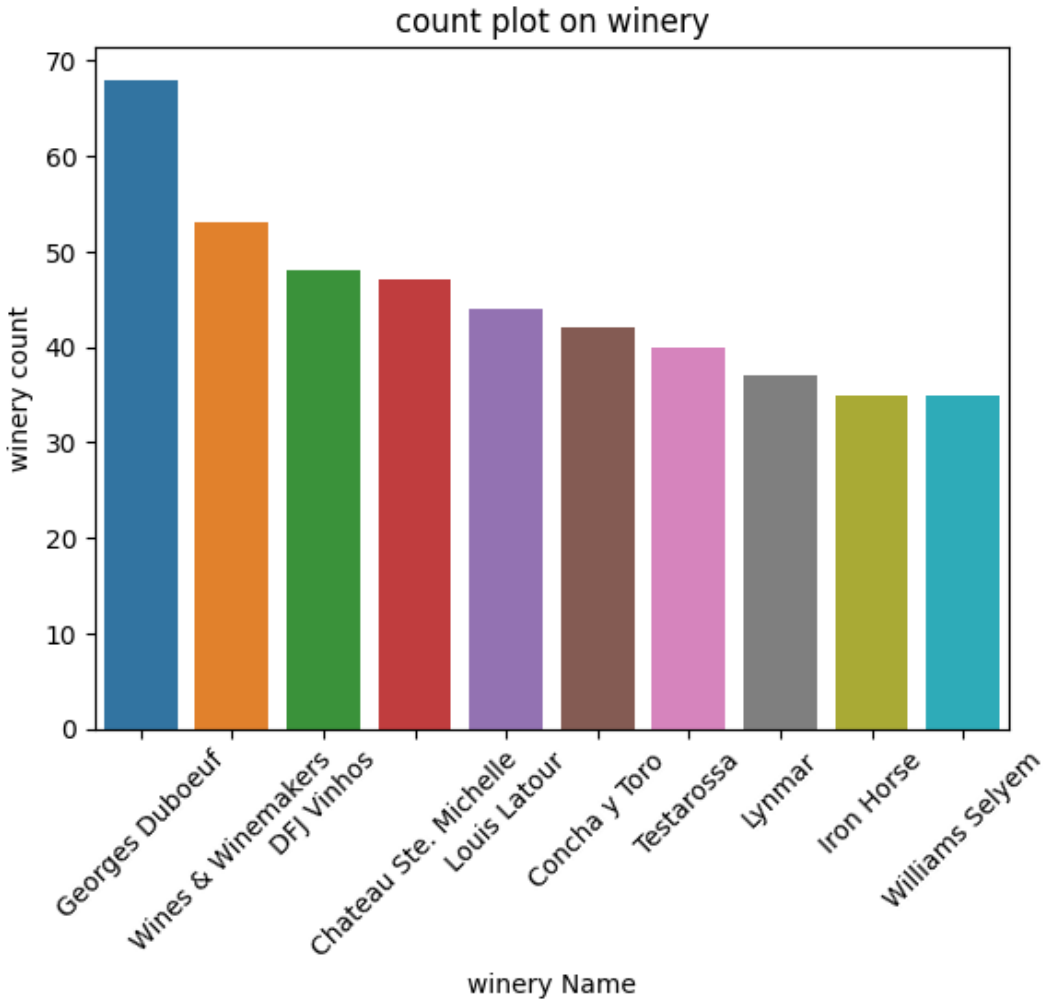




The variety pinot Noir wine is tasted many time in the wine review data over 3000 times

In [259]:

```
winery = data["winery"].value_counts().reset_index()
sns.barplot(data=winery.head(10), x="index", y="winery").set_title(
    "count plot on winery"
)
plt.xlabel("winery Name")
plt.ylabel("winery count")
a = plt.xticks(rotation=45)
```



The winery Georges Duboeuf is the most tasted wine based on this wine Dataset

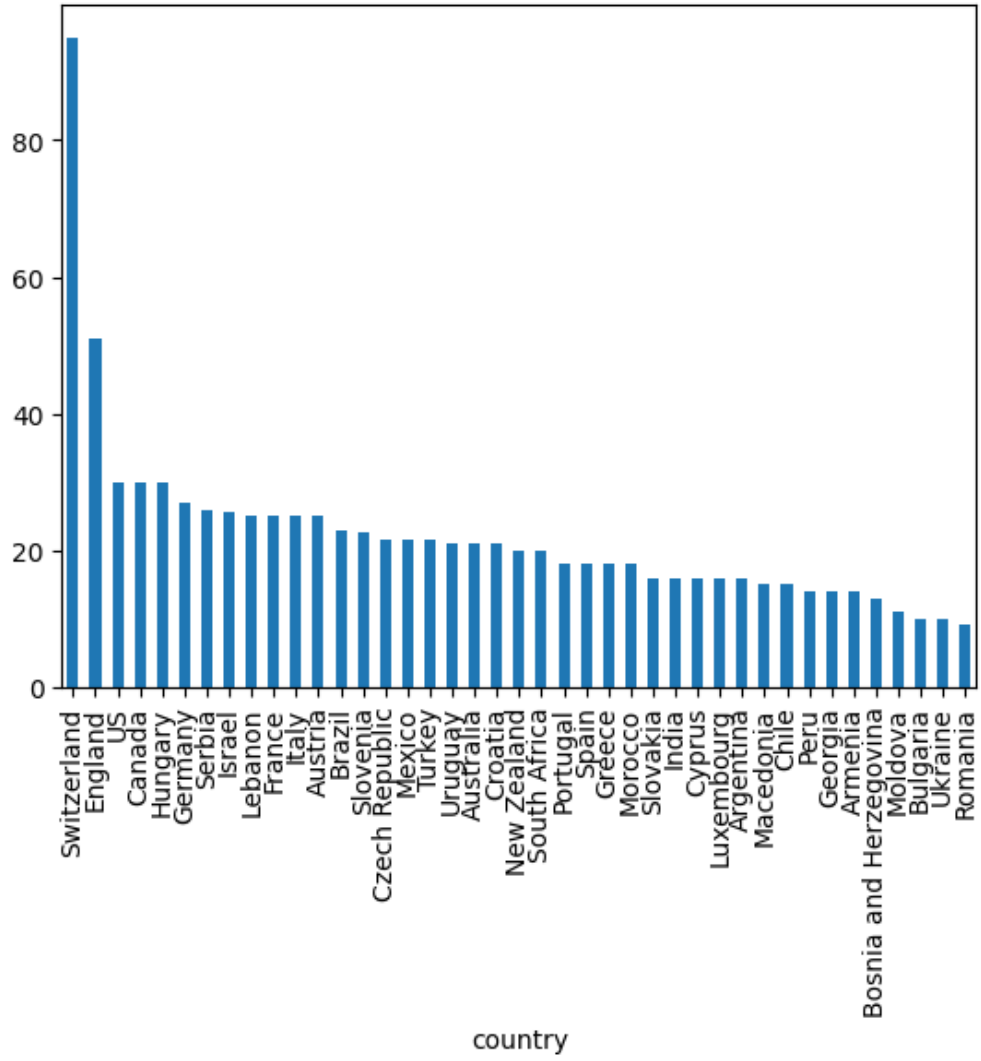
Multivariant Analysis

In [260]:

```
data.groupby("country")["price"].median().sort_values(ascending=False).plot.bar()
```

Out[260]:

<AxesSubplot:xlabel='country'>



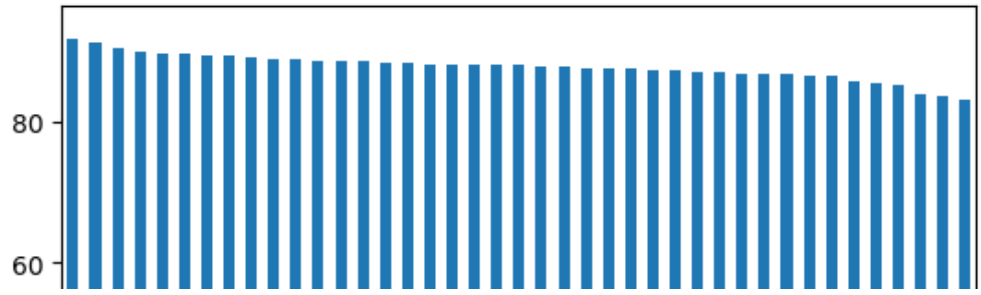
The Swizerland wine are most costly among the other country

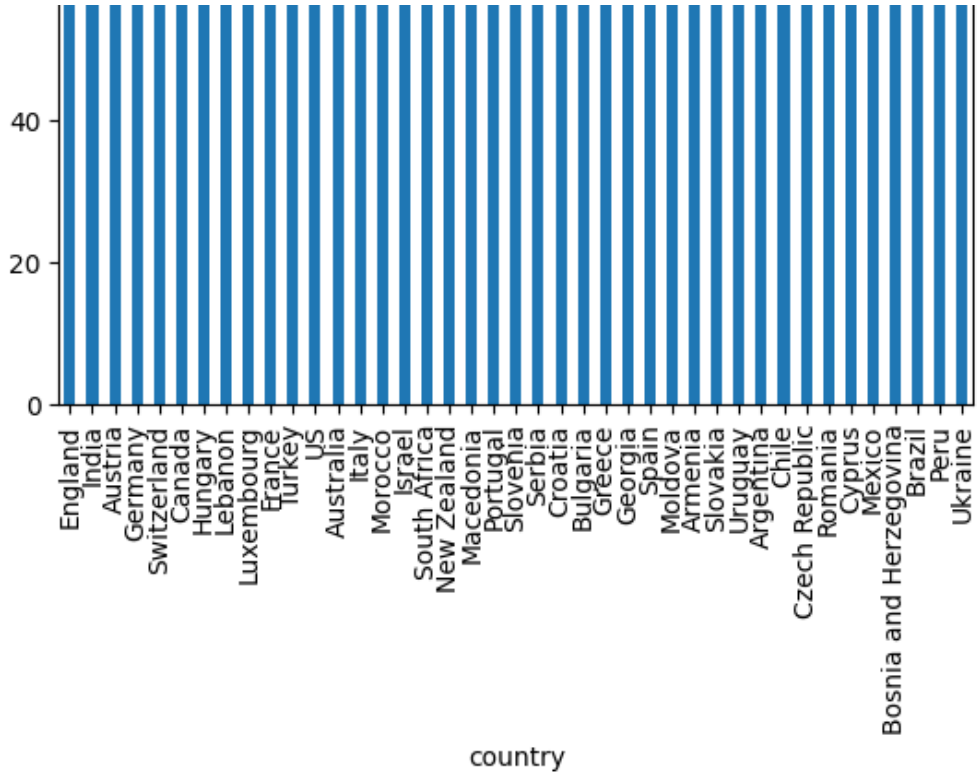
In [261]:

```
data.groupby("country")["points"].mean().sort_values(ascending=False).plot.bar()
```

Out[261]:

<AxesSubplot:xlabel='country'>





England is the country which having high average points

In [262]:

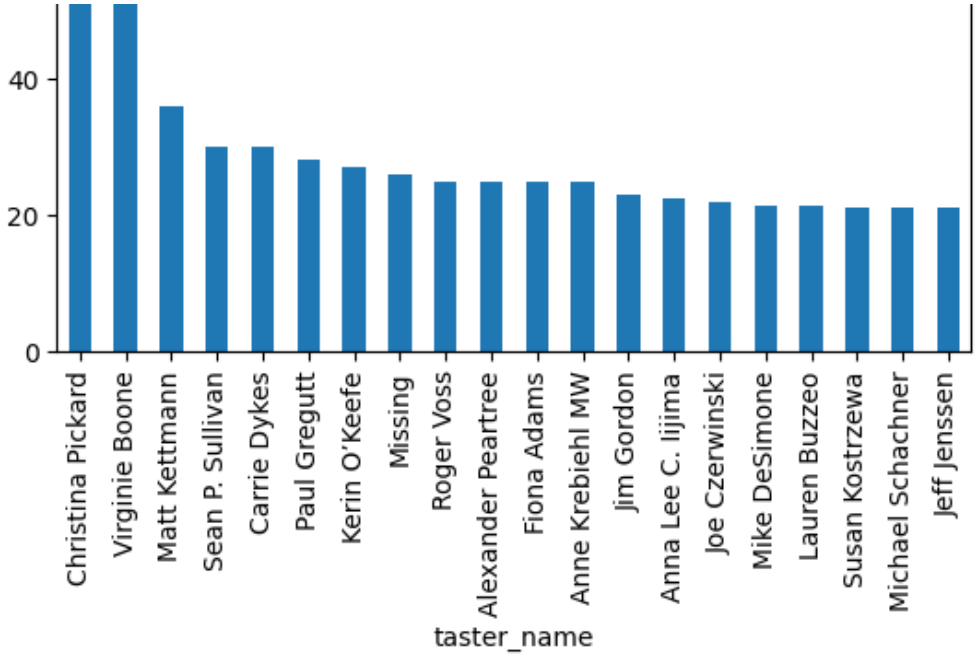
```
## varibale with 50 or less tha 50 unique categories
cat_feature_50 = [feature for feature in data.columns if data[feature].dtype == "O"]
cat_feature_50
```

Out[262]:

```
['country',
 'description',
 'designation',
 'province',
 'taster_name',
 'title',
 'variety',
 'winery']
```

In [263]:

```
for feature in cat_feature_50:
    if data[feature].nunique() < 50:
        datas = (
            data.groupby(feature)["price"]
            .median()
            .sort_values(ascending=False)
            .plot.bar()
        )
```



The taster_name feature is about the people who tast the wine the most wine tasted by Christina pickard

In [264]:

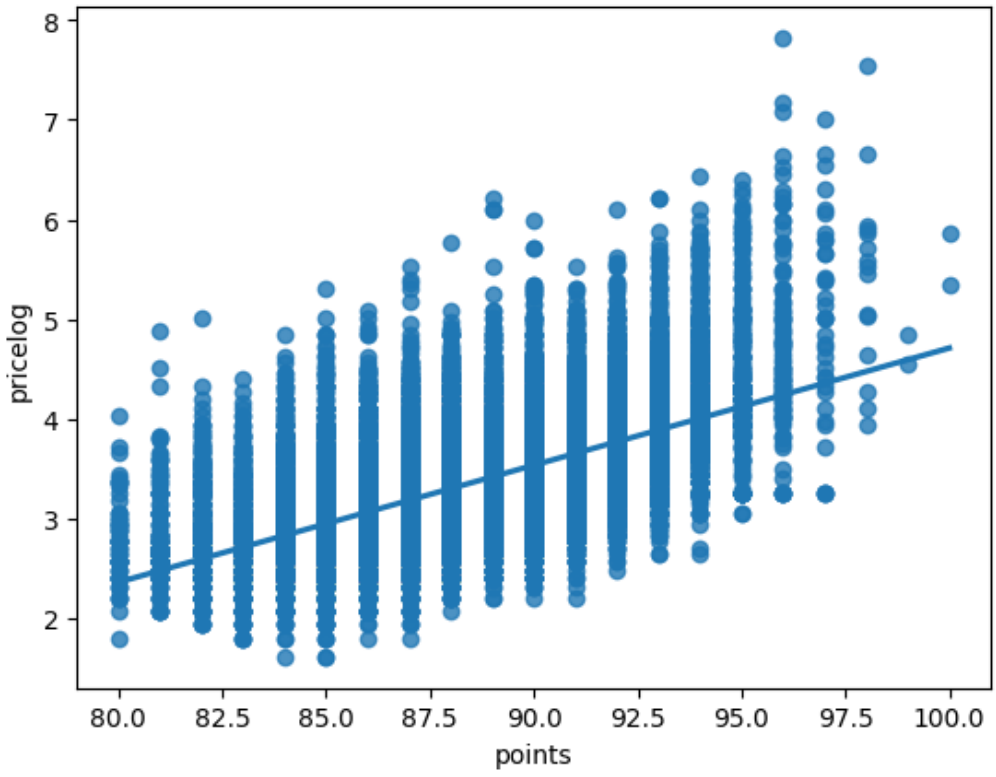
```
## log Trasnform
data["pricelog"] = np.log1p(data["price"])
```

In [265]:

```
##vReg plot
sns.regplot(y=data["pricelog"], x=data["points"])
```

Out[265]:

```
<AxesSubplot:xlabel='points', ylabel='pricelog'>
```

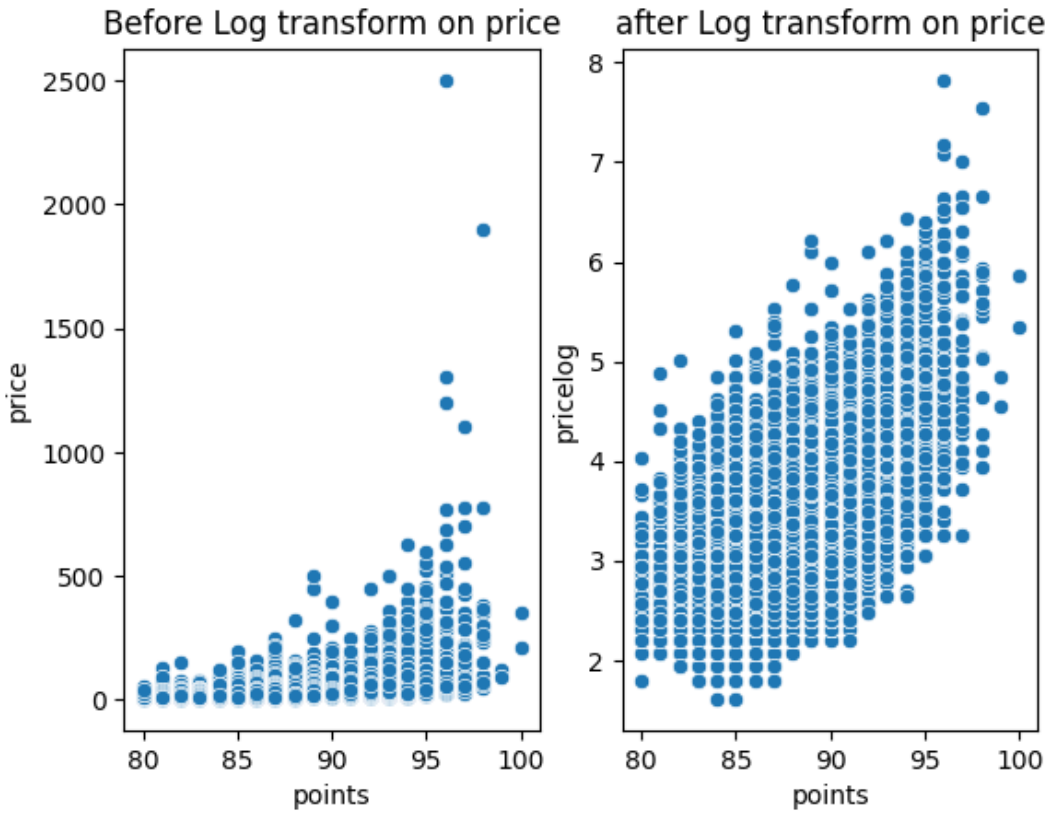


In [266]:

```
fig, ax = plt.subplots(1, 2)
sns.scatterplot(data=data, y="price", x="points", ax=ax[0]).set_title(
    "Before Log transform on price"
)
sns.scatterplot(data=data, y="pricelog", x="points", ax=ax[1]).set_title(
    "after Log transform on price"
)
```

Out[266]:

```
Text(0.5, 1.0, 'after Log transform on price')
```



In [267]:

```
## correlatio before log transform
print("before applying log")
print(data[["points", "price"]].corr())
## correlatio after log transform
print("after applying log")
print(data[["points", "pricelog"]].corr())
```

```
before applying log
          points     price
points  1.000000  0.400483
price   0.400483  1.000000
after applying log
            points  pricelog
points    1.000000  0.585642
pricelog  0.585642  1.000000
```

There is a strong relation between the (price and point) The correlation of price and points is 40% before log transform The correlation of price and points is 58% after log transform it shows that the log transform make the distribution close to normal

In [268]:

```
top_5_country = list(
    data["country"]
    .value_counts()
```

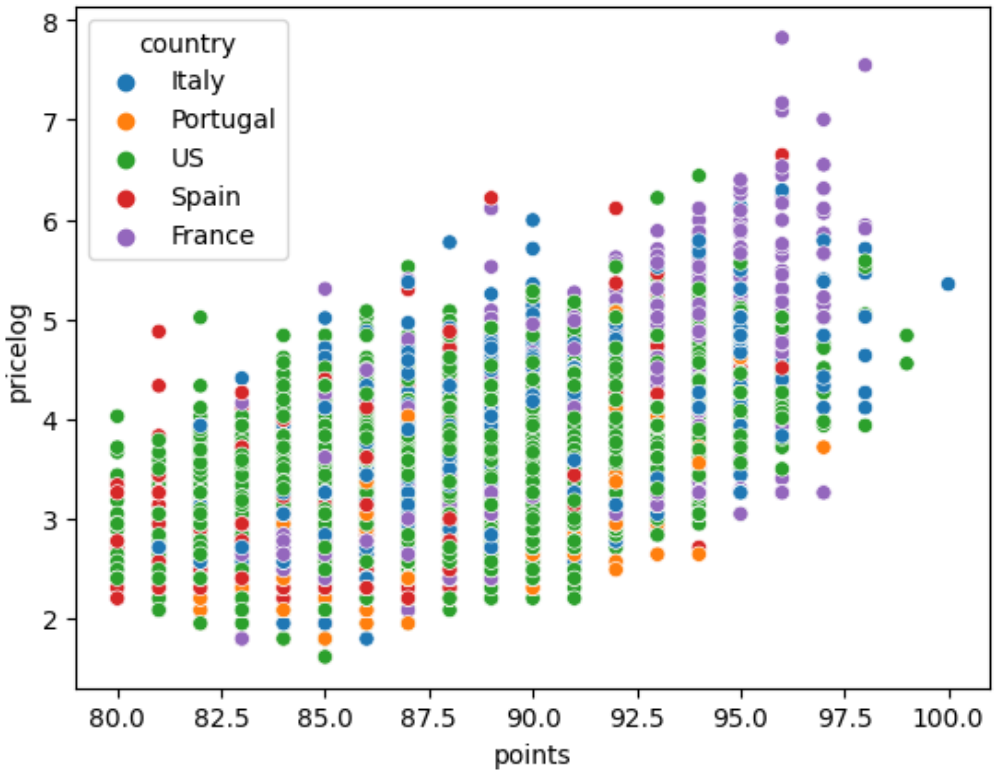
```
    .sort_values(ascending=False)
    .reset_index()["index"]
    .head()
)
top_5_country
top_5_co = data[data["country"].isin(top_5_country)]
```

In [269]:

```
## Top 5 country in sactter plot
sns.scatterplot(data=data, x="points", y="pricelog", hue=top_5_co["country"])
```

Out[269]:

```
<AxesSubplot:xlabel='points', ylabel='pricelog'>
```



Top 5 countrys points plotted in scatter plot it shows mostly the points are from US

In [270]:

```
for feature in data.columns:
    print("The unique values in ", feature, "is", data[feature].nunique())
```

```
The unique values in  country is 41
The unique values in  description is 29154
The unique values in  designation is 13445
The unique values in  points is 21
The unique values in  price is 242
The unique values in  province is 330
The unique values in  taster_name is 20
The unique values in  title is 29092
The unique values in  variety is 474
The unique values in  winery is 9873
The unique values in  pricelog is 242
```

In [271]:

```
data.groupby(["country", "winery", "variety", "province"])[
    "price"
].median().sort_values(ascending=False).reset_index()
```

Out[271]:

	country	winery	variety	province	price
0	France	Château Pétrus	Bordeaux-style Red Blend	Bordeaux	2500.0
1	Spain	Marco Abella	Carignan	Catalonia	770.0
2	France	Château Haut-Brion	Bordeaux-style Red Blend	Bordeaux	765.0
3	France	Château La Mission Haut-Brion	Bordeaux-style White Blend	Bordeaux	698.0
4	US	Yao Ming	Cabernet Sauvignon	California	625.0
...	...	...	...	...	...
18622	Argentina	Terrenal	Malbec	Mendoza Province	5.0
18623	France	Belle Made For You	Cabernet Sauvignon	France Other	5.0
18624	Argentina	Broke Ass	Malbec-Syrah	Mendoza Province	4.0
18625	Spain	Felix Solis	Syrah	Central Spain	4.0
18626	US	Dancing Coyote	White Blend	California	4.0

18627 rows × 5 columns

In [272]:

```
data.groupby(["country", "winery", "variety", "province"])["points"].mean().sort_values(
    ascending=False
).reset_index()
```

Out[272]:

	country	winery	variety	province	points
0	US	Quilceda Creek	Cabernet Sauvignon	Washington	99.0
1	France	Château La Mission Haut-Brion	Bordeaux-style White Blend	Bordeaux	97.0
2	Italy	Passopisciaro	Nerello Mascalese	Sicily & Sardinia	97.0
3	Italy	Tenuta dell'Ornellaia	Red Blend	Tuscany	97.0
4	France	Château Haut-Brion	Bordeaux-style Red Blend	Bordeaux	96.5
...	...	...	...	...	...
18622	US	Hermes	Nebbiolo	Ohio	80.0
18623	US	California's Jewel	Zinfandel	California	80.0
18624	France	Mont Tauch	Red Blend	Languedoc-Roussillon	80.0
18625	US	Pianetta	Cabernet Sauvignon	California	80.0
18626	Spain	Reula	Tempranillo-Merlot	Northern Spain	80.0

18627 rows × 5 columns

In [273]:

```
data.groupby("province")["points"].mean().sort_values(ascending=False)
```

Out[273]:

```
province
Mittelrhein                94.000000
Eisenberg                  93.000000
Santa Cruz                 92.500000
Tokaji                     91.714286
England                    91.636364
                             ...
San Jose                   82.500000
```

```
Middle and South Dalmatia    82.000000
Molina                       82.000000
Serra do Sudeste             82.000000
Table wine                   81.000000
Name: points, Length: 330, dtype: float64
```

Sampling

In [274]:

```
country_stata = data.groupby("country", group_keys=False).apply(
    lambda x: x.sample(100, replace=True)
)
country_stata.head(2)
```

Out[274]:

	country	description	designation	points	price	province	taster_name	title	variety	winery	pricelog
13606	**Argentina**	**With its tropical melony aromas and crisp pala...**	**Missing**	**89**	**14.0**	**Mendoza Province**	**Michael Schachner**	**Gauchezco 2010 Torrontés (Mendoza)**	**Torrontés**	**Gauchezco**	**2.708050**
9006	**Argentina**	**Earthy and meaty, with aromas of dark cherry, ...**	**Alberto Furque**	**88**	**15.0**	**Mendoza Province**	**Michael Schachner**	**Bodega Aconquija 2005 Alberto Furque Syrah (Uc...**	**Syrah**	**Bodega Aconquija**	**2.772589**

In [275]:

```
country_stata.groupby("country")["price"].mean().sort_values(ascending=False)
```

Out[275]:

```
country
Switzerland               102.80
France                     64.02
England                    56.21
Hungary                    55.34
Lebanon                    41.38
Germany                    38.80
US                         35.60
Canada                     35.28
Italy                      34.76
Austria                    32.63
Israel                     32.34
Serbia                     29.72
Australia                  29.23
Uruguay                    28.59
New Zealand                27.67
Croatia                    26.71
Argentina                  26.68
Spain                      26.03
Czech Republic             24.96
Slovenia                   24.77
Turkey                     24.46
Portugal                   24.40
Mexico                     24.40
South Africa               22.54
Greece                     22.47
```

```
Morocco                    22.44
Brazil                     22.39
Chile                      21.22
Moldova                    18.93
Georgia                    17.72
Slovakia                   16.00
Luxembourg                 16.00
Cyprus                     16.00
India                      15.76
Macedonia                  15.00
Armenia                    14.00
Peru                       13.15
Bosnia and Herzegovina     13.00
Bulgaria                   12.60
Romania                    12.44
Ukraine                    10.00
Name: price, dtype: float64
```

After sampling equal proportion the 'Switzerland' wine are More Costly

In [276]:

```
country_stata.groupby("country")["points"].mean().sort_values(ascending=False)
```

Out[276]:

```
country
England                    91.60
India                      90.88
Austria                    90.50
Germany                    89.73
Switzerland                89.44
Lebanon                    89.37
Hungary                    89.30
Canada                     89.15
Luxembourg                 89.00
Australia                  88.81
France                     88.79
Turkey                     88.75
Italy                      88.63
US                         88.47
Israel                     88.34
New Zealand                88.18
Morocco                    88.14
Macedonia                  88.00
South Africa               87.96
Portugal                   87.91
Slovenia                   87.90
Serbia                     87.86
Croatia                    87.77
Bulgaria                   87.72
Greece                     87.59
Georgia                    87.42
Moldova                    87.25
Argentina                  87.21
Spain                      87.20
Armenia                    87.00
Slovakia                   87.00
Uruguay                    86.88
Czech Republic             86.70
Chile                      86.58
Romania                    86.31
Cyprus                     85.68
Mexico                     85.35
Bosnia and Herzegovina     85.00
Brazil                     83.94
Peru                       83.59
Ukraine                    83.00
```

```
Name: points, dtype: float64
```

After sampling england is having highest average points (or) we can say that the wine from England are get high points form the taster in the wine review Dataset

In [277]:

```
# relation Between province and Price
data.groupby("province")["price"].median().sort_values(ascending=False)
```

Out[277]:

```
province
Switzerland                 160.0
Puente Alto                 103.5
Santa Cruz                   95.0
Apalta                       82.0
Middle and South Dalmatia    65.0
                            ...
Dealurile Munteniei           8.0
Alenquer                      8.0
Molina                        8.0
Viile Timisului               7.0
Recas                         7.0
Name: price, Length: 330, dtype: float64
```

The above Data having unequal proportion of province so we have to equal the proption for that we can use stratified Sampling

In [278]:

```
# Appling Stratified Sampling
province_stata = data.groupby("province", group_keys=False).apply(
    lambda x: x.sample(1000, replace=True)
)
province_stata.head(2)
```

Out[278]:

	country	description	designation	points	price	province	taster_name	title	variety	winery	priceIc
22328	Chile	Oceanic aromas of grass, scallion, baby garlic...	Missing	90	22.0	Aconcagua Costa	Michael Schachner	Errazuriz 2015 Sauvignon Blanc (Aconcagua Costa)	Sauvignon Blanc	Errazuriz	3.13549
2205	Chile	Nutty aromas of popcorn, buttered toast, peach...	Missing	88	20.0	Aconcagua Costa	Michael Schachner	Arboleda 2014 Chardonnay (Aconcagua Costa)	Chardonnay	Arboleda	3.04452

In [279]:

```
# after sampling grouping Province with Points to check which province have high average
point
province_stata.groupby("province")["points"].median().sort_values(ascending=False)
```

Out[279]:

```
province
Eisenberg                   94.0
Mittelrhein                 94.0
```

```
Nasnik                        93.0
Santa Cruz                    93.0
Puente Alto                   92.0
                              ...
Canterbury                    82.0
Middle and South Dalmatia     82.0
Serra do Sudeste              82.0
Vale dos Vinhedos             81.0
Table wine                    81.0
Name: points, Length: 330, dtype: float64
```

In [280]:

```
# after sampling grouping Province with Price to check which province have high average price
province_stata.groupby("province")["price"].median().sort_values(ascending=False)
```

Out[280]:

```
province
Switzerland         160.0
Puente Alto         120.0
Olifants River      100.0
Santa Cruz           95.0
Apalta               82.0
                    ...
Molina                8.0
Alenquer              8.0
Recas                 7.0
Primorska             7.0
Viile Timisului       7.0
Name: price, Length: 330, dtype: float64
```

Findings in this analysis

Univariate

1 The Nan values Have no realtion with output feature so we can replace the values by Median

2 The points feature more or less follows normal distribution

3 In this reviews the most of the wine is from 'United States' over 12000 and after that 'France'

4 The designation has more missing values after tha 'reserver designation' is more in the review Dataset

5 The Data set of wine reviews the state 'California' is having more counts and second is 'Washington'

6 Many people tast the wine are not provide the name so many values are missing in Taster_Name feature

7 The variety 'Pinot Noir' wine is tasted many time in the wine review data over 3000 times

8 The winery 'Georges Duboeuf' is the most tasted wine based on this wine Dataset

Multivariate

1 The 'Swizerland' wine are most costly among the other country

2 England is the country which having high average points

3 The 'Price' Feature is Right Skewed so we can perform Log transform

4 There is a strong relation between the (price and point)

The correlation of price and points is 40% before log transform The correlation of price and points is 58% after log transform it shows that the log transform make the distribution close to normal

Sampling

1 The Data having unequal proportion of province so we have to equal the proption for that we can use stratified Sampling

2 After sampling equal proportion the 'Switzerland' wine are More Costly

3 After sampling england is having highest average points (or)

we can say that the wine from England are get high points form the taster in the wine review Dataset

In [ ]: