Data Manipulation and Visualization Project

Food

Emma Trivini Bellini

INDEX

- PROJECT GUIDELINES
- DISCOVERY (Problem Identification and Objectives)
- DATA SELECTION
- DATA CLEANING AND DATA TRANSFORMATION
    - Data exploration
    - Data cleaning
    - Qualitative variables
- DATA EXPLORATION AND DATA VISUALIZATION
- CONCLUSIONS

PROJECT GUIDELINES

Goal

Analyze a dataset of 130,000 wine reviews to produce data-driven recommendations for a wine marketplace that connects small local producers with global buyers—prioritizing high quality and competitive pricing.

Scope & Objectives

- Country-level insights: compute average score & price by country to map geographical strengths and pricing trends.
- Italian market deep-dive: analyze price distributions, province-level patterns, and price–score relationships.
- European context: assess production across Europe to identify key regions and specialties.
- Key insights: detect spot-price outliers and investigate correlations among variables.

Approach

Rigorous data analysis + clear visualizations to ensure transparent reasoning and actionable outcomes.

Deliverable

Data-driven recommendations for product assortment and pricing strategy to support small producers and global buyers.

1. DISCOVERY (Problem Identification and Objectives)

Wine appreciation depends on factors such as origin, price, and quality. Small producers often lack visibility and access to global markets.

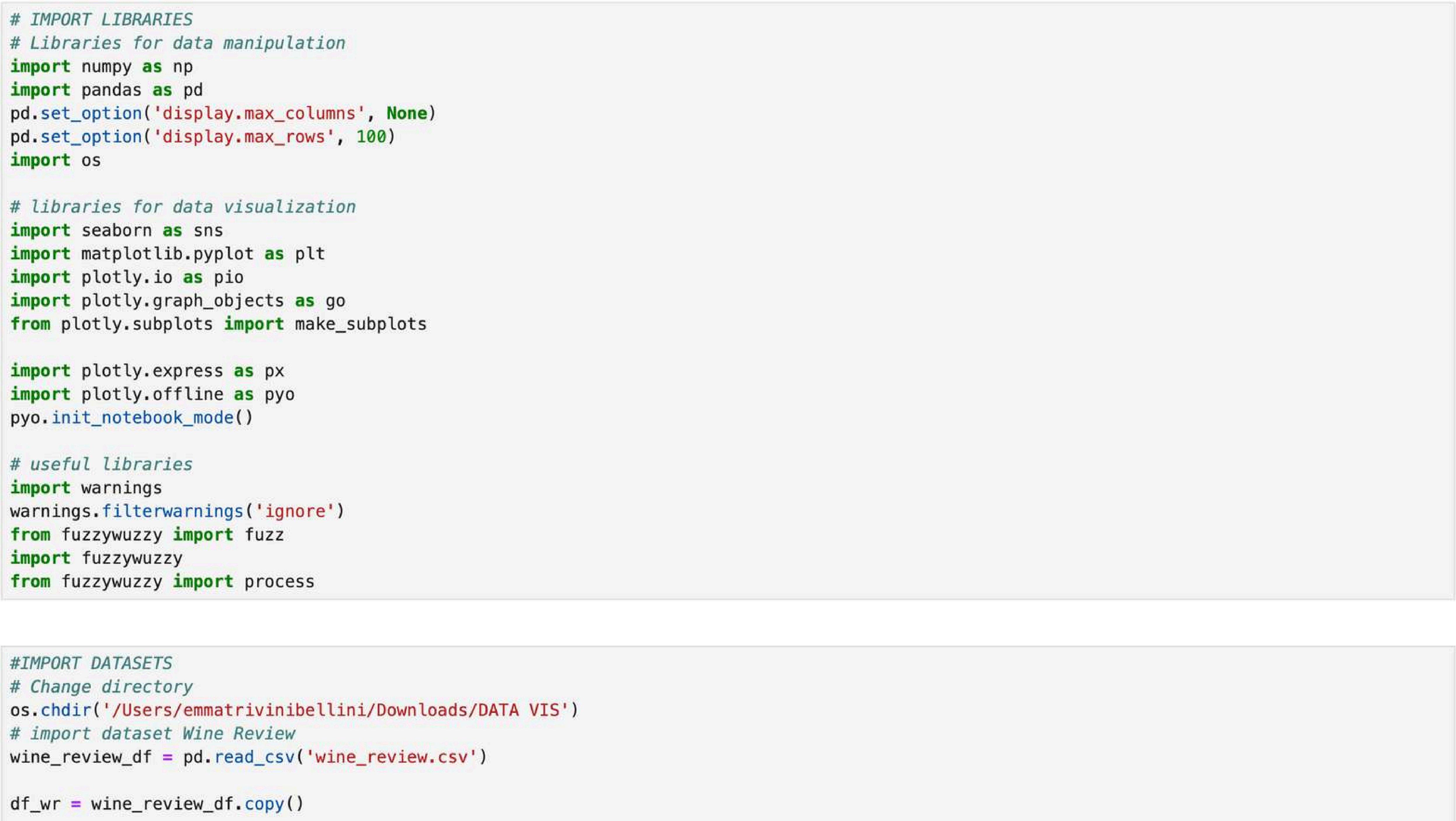
This project uses data analysis to design a marketplace that promotes the diversity and excellence of small-scale wine production.

By exploring data on variety, origin, vineyard, price, and reviews, the analysis aims to reveal regional strengths, pricing patterns, and consumer preferences.

The goal is to connect small producers with global buyers through a curated, data-driven selection that celebrates wine diversity and meets market demand.

2. DATA SELECTION

The dataset of the Wine Review ***is available on Kaggle and it contains 130,000 wine reviews, including details on variety, origin, vineyard, price, and description.***

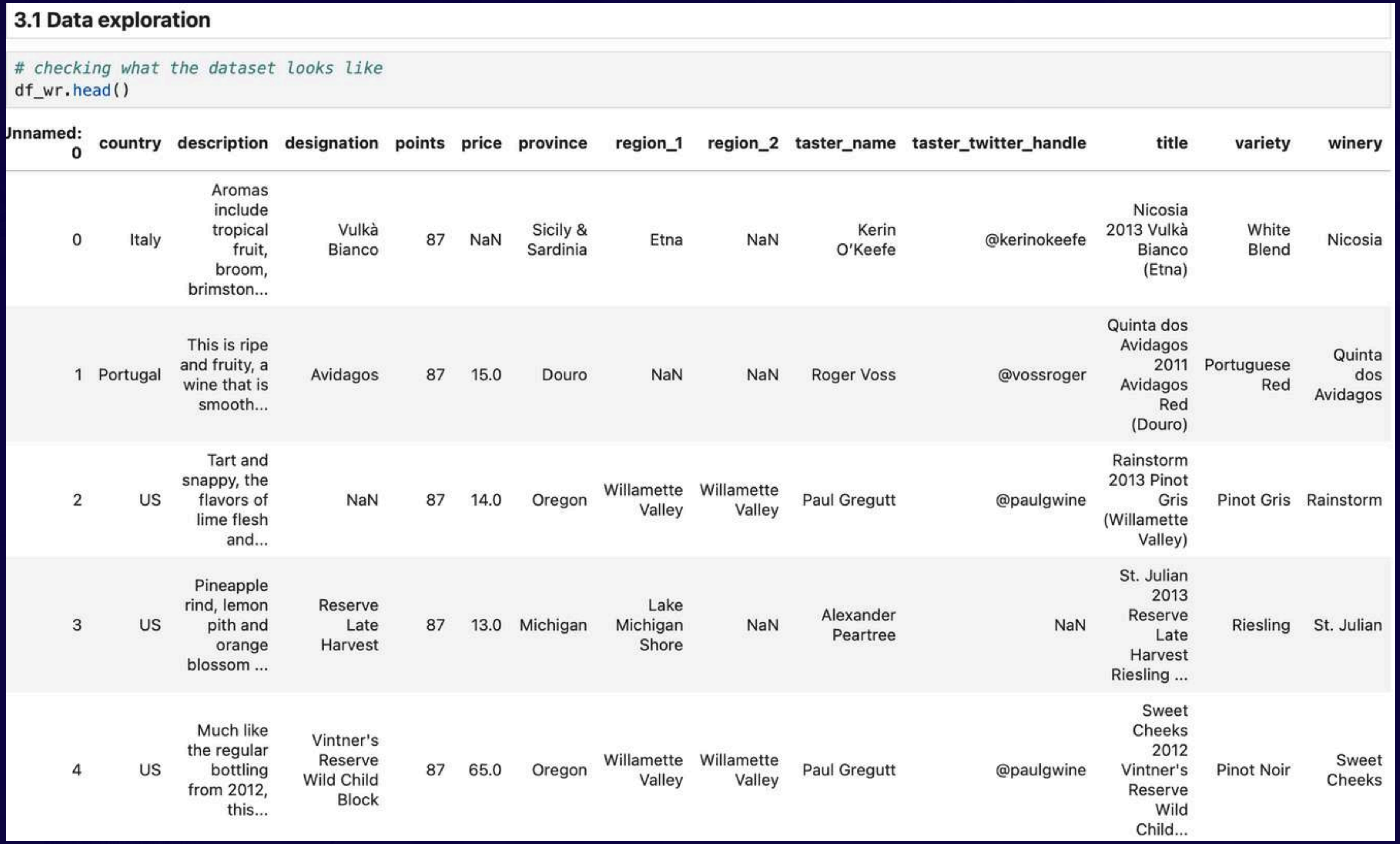


```
# IMPORT LIBRARIES
# Libraries for data manipulation
import numpy as np
import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 100)
import os

# libraries for data visualization
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.io as pio
import plotly.graph_objects as go
from plotly.subplots import make_subplots

import plotly.express as px
import plotly.offline as pyo
pyo.init_notebook_mode()

# useful libraries
import warnings
warnings.filterwarnings('ignore')
from fuzzywuzzy import fuzz
import fuzzywuzzy
from fuzzywuzzy import process
```

```
#IMPORT DATASETS
# Change directory
os.chdir('/Users/emmatrivinibellini/Downloads/DATA VIS')
# import dataset Wine Review
wine_review_df = pd.read_csv('wine_review.csv')

df_wr = wine_review_df.copy()
```

3. DATA CLEANING AND DATA TRANSFORMATION

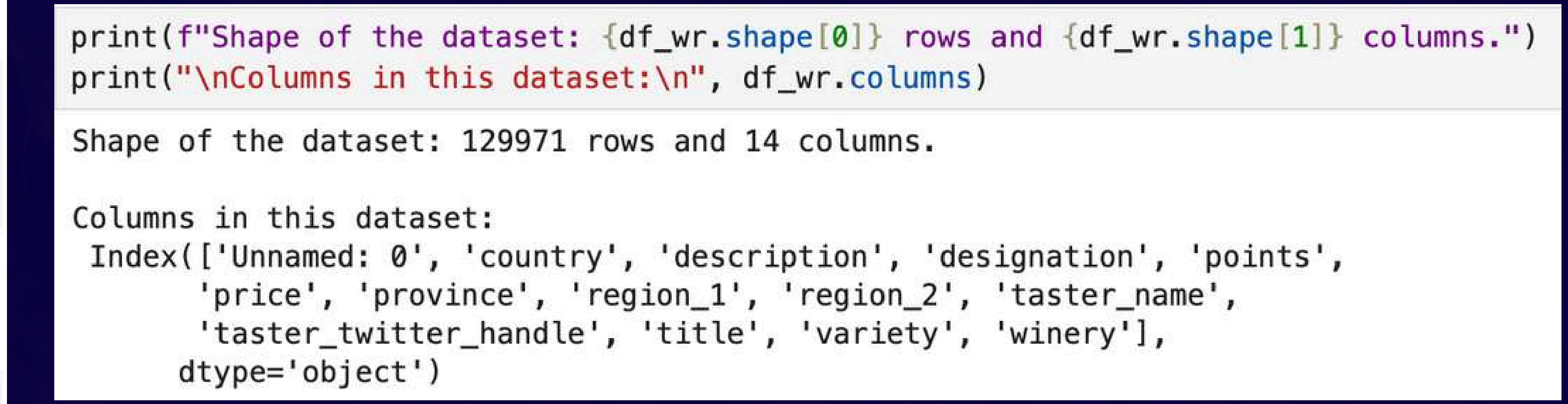
Before actually starting the analysis, it's a good practice to check if the data are ready or not. They might have some invalid values, missing values, mistyped and so on. In this section, I will investigate if data are consistent and ready to be processed.



3.1 Data exploration

```
# checking what the dataset looks like
df_wr.head()
```

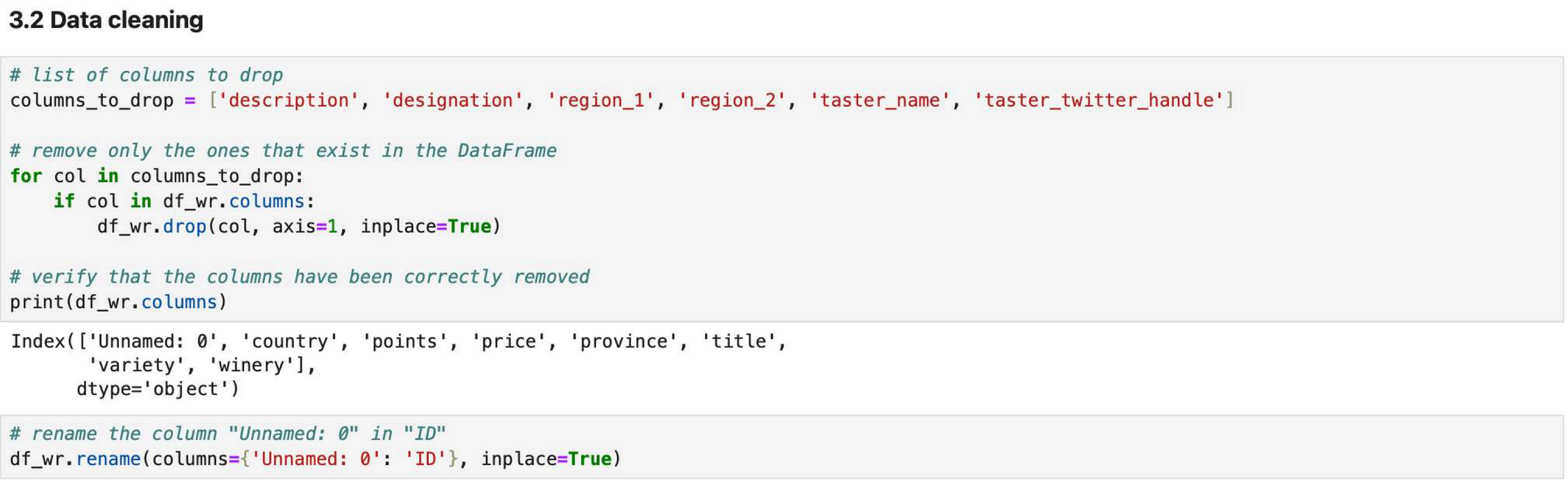
Jnnamed: 0	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title	variety	winery
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O’Keefe	@kerinokeefe	Nicosia 2013 Vulkà Bianco (Etna)	White Blend	Nicosia
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	@vossroger	Quinta dos Avidagos 2011 Avidagos Red (Douro)	Portuguese Red	Quinta dos Avidagos
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Rainstorm 2013 Pinot Gris (Willamette Valley)	Pinot Gris	Rainstorm
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexander Peartree	NaN	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	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Sweet Cheeks 2012 Vintner's Reserve Wild Child...	Pinot Noir	Sweet Cheeks



```
print(f"Shape of the dataset: {df_wr.shape[0]} rows and {df_wr.shape[1]} columns.")
print("\nColumns in this dataset:\n", df_wr.columns)
```

```
Shape of the dataset: 129971 rows and 14 columns.

Columns in this dataset:
 Index(['Unnamed: 0', 'country', 'description', 'designation', 'points',
       'price', 'province', 'region_1', 'region_2', 'taster_name',
       'taster_twitter_handle', 'title', 'variety', 'winery'],
      dtype='object')
```

3. DATA CLEANING AND DATA TRANSFORMATION



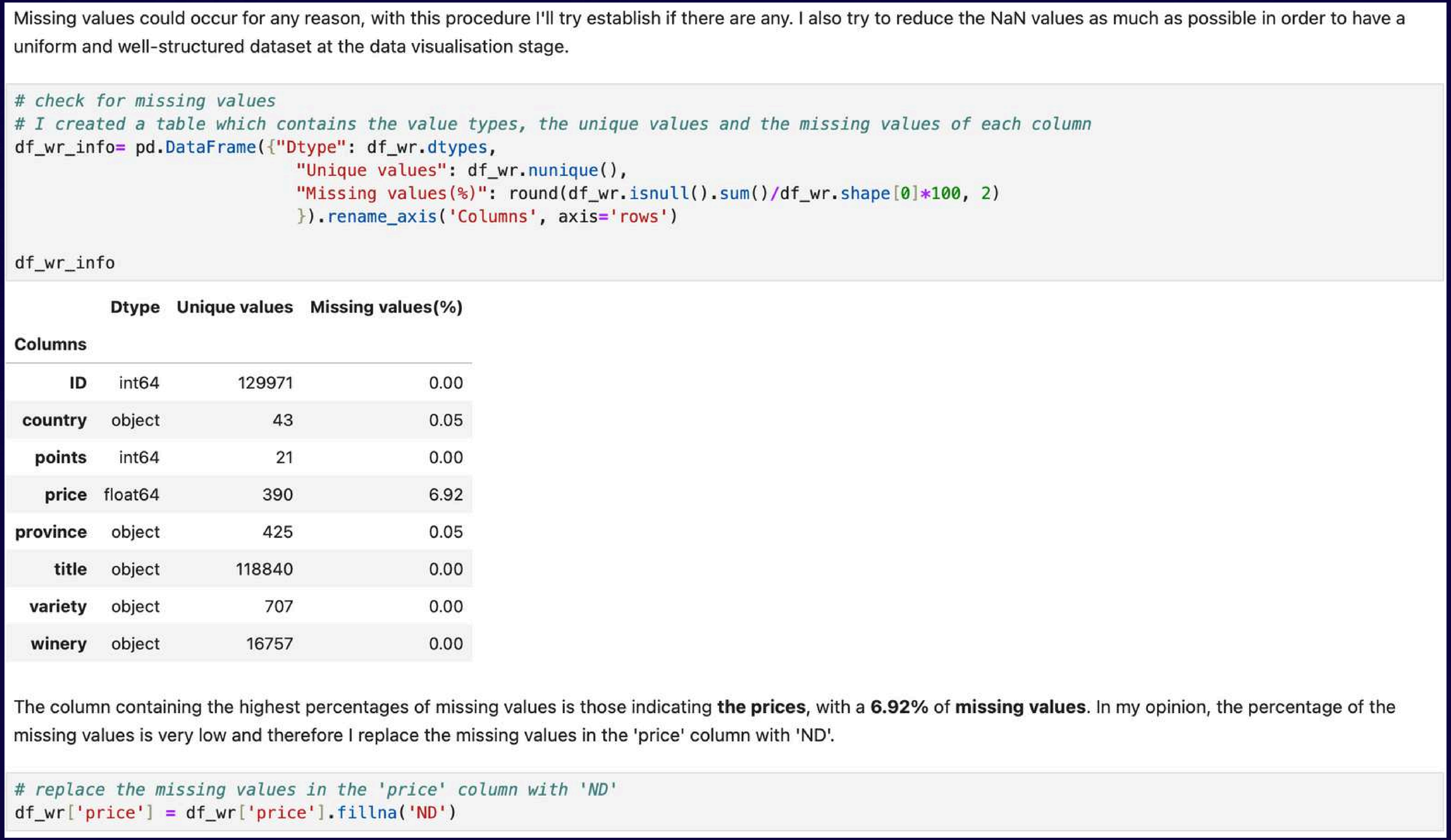
3.2 Data cleaning

```python
# list of columns to drop
columns_to_drop = ['description', 'designation', 'region_1', 'region_2', 'taster_name', 'taster_twitter_handle']

# remove only the ones that exist in the DataFrame
for col in columns_to_drop:
    if col in df_wr.columns:
        df_wr.drop(col, axis=1, inplace=True)

# verify that the columns have been correctly removed
print(df_wr.columns)
```

```
Index(['Unnamed: 0', 'country', 'points', 'price', 'province', 'title',
       'variety', 'winery'],
      dtype='object')
```

```python
# rename the column "Unnamed: 0" in "ID"
df_wr.rename(columns={'Unnamed: 0': 'ID'}, inplace=True)
```

3. DATA CLEANING AND DATA TRANSFORMATION



Missing values could occur for any reason, with this procedure I'll try establish if there are any. I also try to reduce the NaN values as much as possible in order to have a uniform and well-structured dataset at the data visualisation stage.

```
# check for missing values
# I created a table which contains the value types, the unique values and the missing values of each column
df_wr_info= pd.DataFrame({"Dtype": df_wr.dtypes,
                            "Unique values": df_wr.nunique(),
                            "Missing values(%)": round(df_wr.isnull().sum()/df_wr.shape[0]*100, 2)
                            }).rename_axis('Columns', axis='rows')

df_wr_info
```

	Dtype	Unique values	Missing values(%)
Columns			
ID	int64	129971	0.00
country	object	43	0.05
points	int64	21	0.00
price	float64	390	6.92
province	object	425	0.05
title	object	118840	0.00
variety	object	707	0.00
winery	object	16757	0.00

The column containing the highest percentages of missing values is those indicating **the prices**, with a **6.92%** of **missing values**. In my opinion, the percentage of the missing values is very low and therefore I replace the missing values in the 'price' column with 'ND'.

```
# replace the missing values in the 'price' column with 'ND'
df_wr['price'] = df_wr['price'].fillna('ND')
```

3. DATA CLEANING AND DATA TRANSFORMATION

In this analysis, we create two datasets to address different levels of granularity and ensure data accuracy.

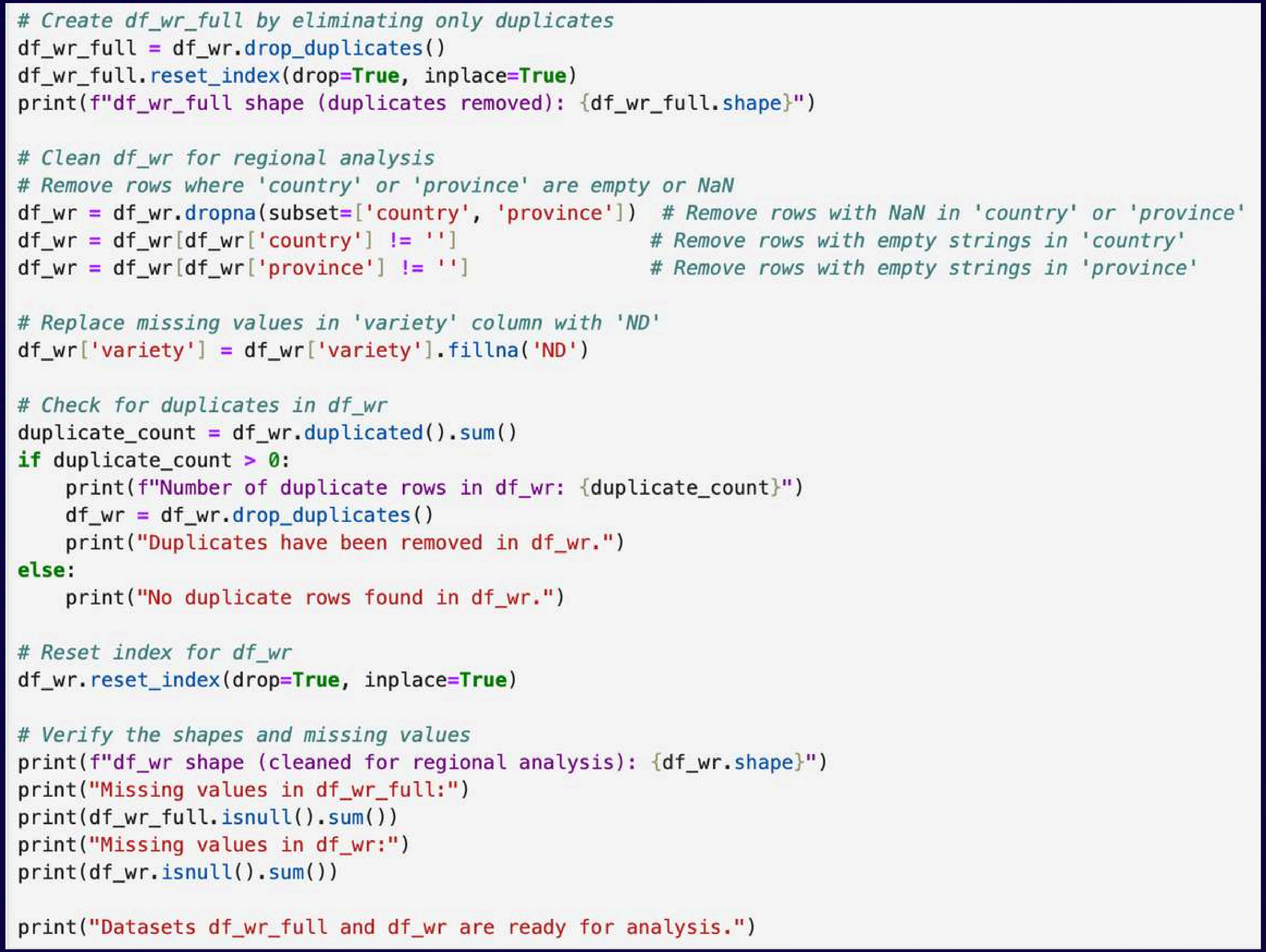
The first dataset, df_wr_full, retains all valid rows except duplicates, even if some information (such as province information) is missing.

This dataset is ideal for global or country-level analyses, such as studying trends across countries, correlations, or distributions where province data is not required.

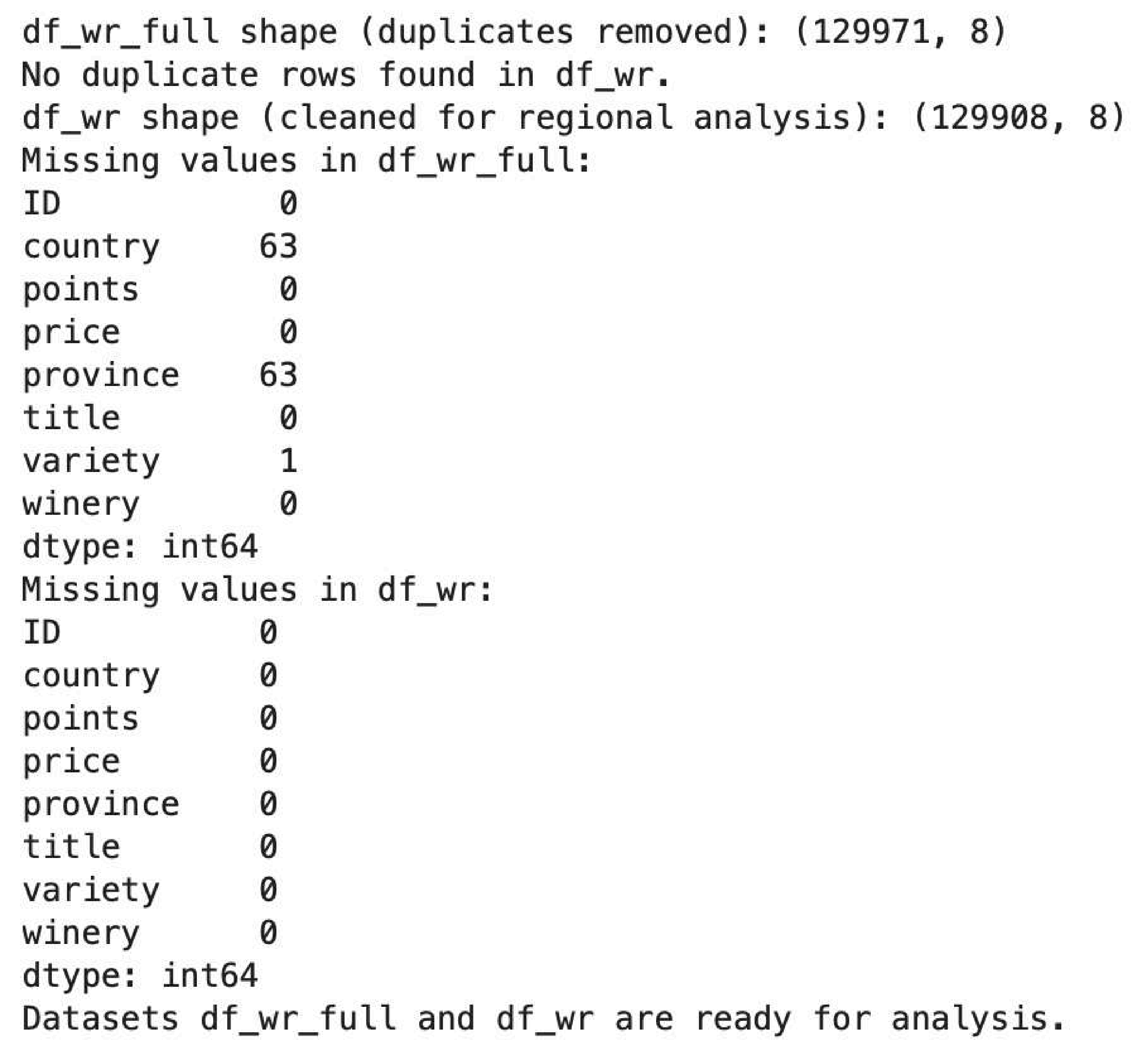
The second dataset, df_wr, filters out rows with missing province information, focusing only on entries with complete regional data.

It is specifically used for regional or province-level analyses, ensuring accurate insights at a more granular level. This separation allows us to maximize data usage for broader trends while maintaining precision for detailed, localized studies.

3. DATA CLEANING AND DATA TRANSFORMATION

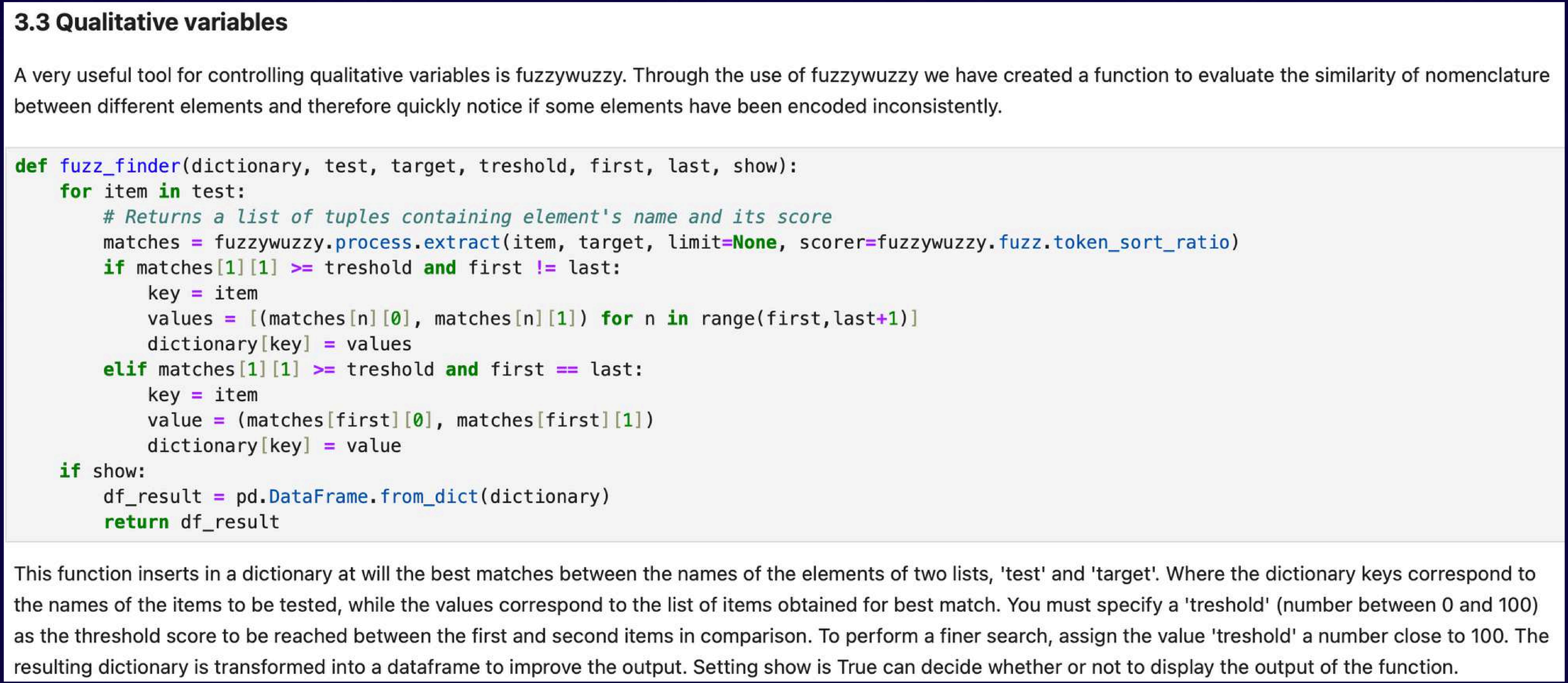


```python
# Create df_wr_full by eliminating only duplicates
df_wr_full = df_wr.drop_duplicates()
df_wr_full.reset_index(drop=True, inplace=True)
print(f"df_wr_full shape (duplicates removed): {df_wr_full.shape}")

# Clean df_wr for regional analysis
# Remove rows where 'country' or 'province' are empty or NaN
df_wr = df_wr.dropna(subset=['country', 'province'])  # Remove rows with NaN in 'country' or 'province'
df_wr = df_wr[df_wr['country'] != '']                # Remove rows with empty strings in 'country'
df_wr = df_wr[df_wr['province'] != '']               # Remove rows with empty strings in 'province'

# Replace missing values in 'variety' column with 'ND'
df_wr['variety'] = df_wr['variety'].fillna('ND')

# Check for duplicates in df_wr
duplicate_count = df_wr.duplicated().sum()
if duplicate_count > 0:
    print(f"Number of duplicate rows in df_wr: {duplicate_count}")
    df_wr = df_wr.drop_duplicates()
    print("Duplicates have been removed in df_wr.")
else:
    print("No duplicate rows found in df_wr.")

# Reset index for df_wr
df_wr.reset_index(drop=True, inplace=True)

# Verify the shapes and missing values
print(f"df_wr shape (cleaned for regional analysis): {df_wr.shape}")
print("Missing values in df_wr_full:")
print(df_wr_full.isnull().sum())
print("Missing values in df_wr:")
print(df_wr.isnull().sum())

print("Datasets df_wr_full and df_wr are ready for analysis.")
```



```
df_wr_full shape (duplicates removed): (129971, 8)
No duplicate rows found in df_wr.
df_wr shape (cleaned for regional analysis): (129908, 8)
Missing values in df_wr_full:
ID           0
country     63
points       0
price        0
province    63
title        0
variety      1
winery       0
dtype: int64
Missing values in df_wr:
ID          0
country     0
points      0
price       0
province    0
title       0
variety     0
winery      0
dtype: int64
Datasets df_wr_full and df_wr are ready for analysis.
```

3. DATA CLEANING AND DATA TRANSFORMATION



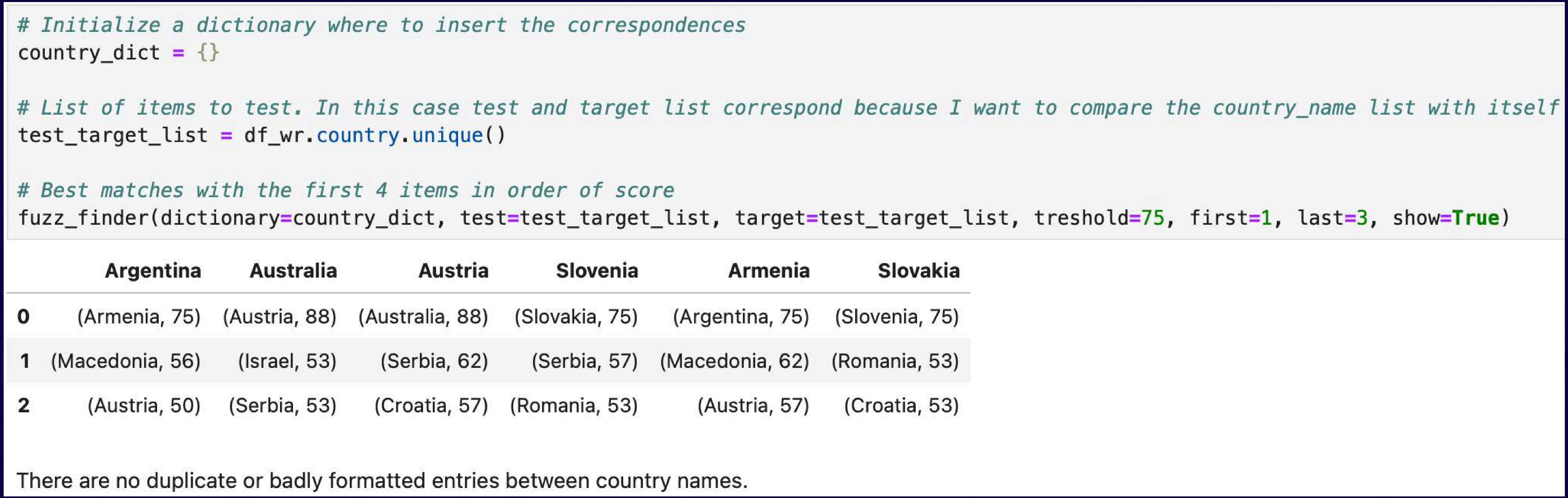
3.3 Qualitative variables

A very useful tool for controlling qualitative variables is fuzzywuzzy. Through the use of fuzzywuzzy we have created a function to evaluate the similarity of nomenclature between different elements and therefore quickly notice if some elements have been encoded inconsistently.

```python
def fuzz_finder(dictionary, test, target, treshold, first, last, show):
    for item in test:
        # Returns a list of tuples containing element's name and its score
        matches = fuzzywuzzy.process.extract(item, target, limit=None, scorer=fuzzywuzzy.fuzz.token_sort_ratio)
        if matches[1][1] >= treshold and first != last:
            key = item
            values = [(matches[n][0], matches[n][1]) for n in range(first,last+1)]
            dictionary[key] = values
        elif matches[1][1] >= treshold and first == last:
            key = item
            value = (matches[first][0], matches[first][1])
            dictionary[key] = value
    if show:
        df_result = pd.DataFrame.from_dict(dictionary)
        return df_result
```

This function inserts in a dictionary at will the best matches between the names of the elements of two lists, 'test' and 'target'. Where the dictionary keys correspond to the names of the items to be tested, while the values correspond to the list of items obtained for best match. You must specify a 'treshold' (number between 0 and 100) as the threshold score to be reached between the first and second items in comparison. To perform a finer search, assign the value 'treshold' a number close to 100. The resulting dictionary is transformed into a dataframe to improve the output. Setting show is True can decide whether or not to display the output of the function.

3. DATA CLEANING AND DATA TRANSFORMATION



```python
# Initialize a dictionary where to insert the correspondences
country_dict = {}

# List of items to test. In this case test and target list correspond because I want to compare the country_name list with itself
test_target_list = df_wr.country.unique()

# Best matches with the first 4 items in order of score
fuzz_finder(dictionary=country_dict, test=test_target_list, target=test_target_list, treshold=75, first=1, last=3, show=True)
```

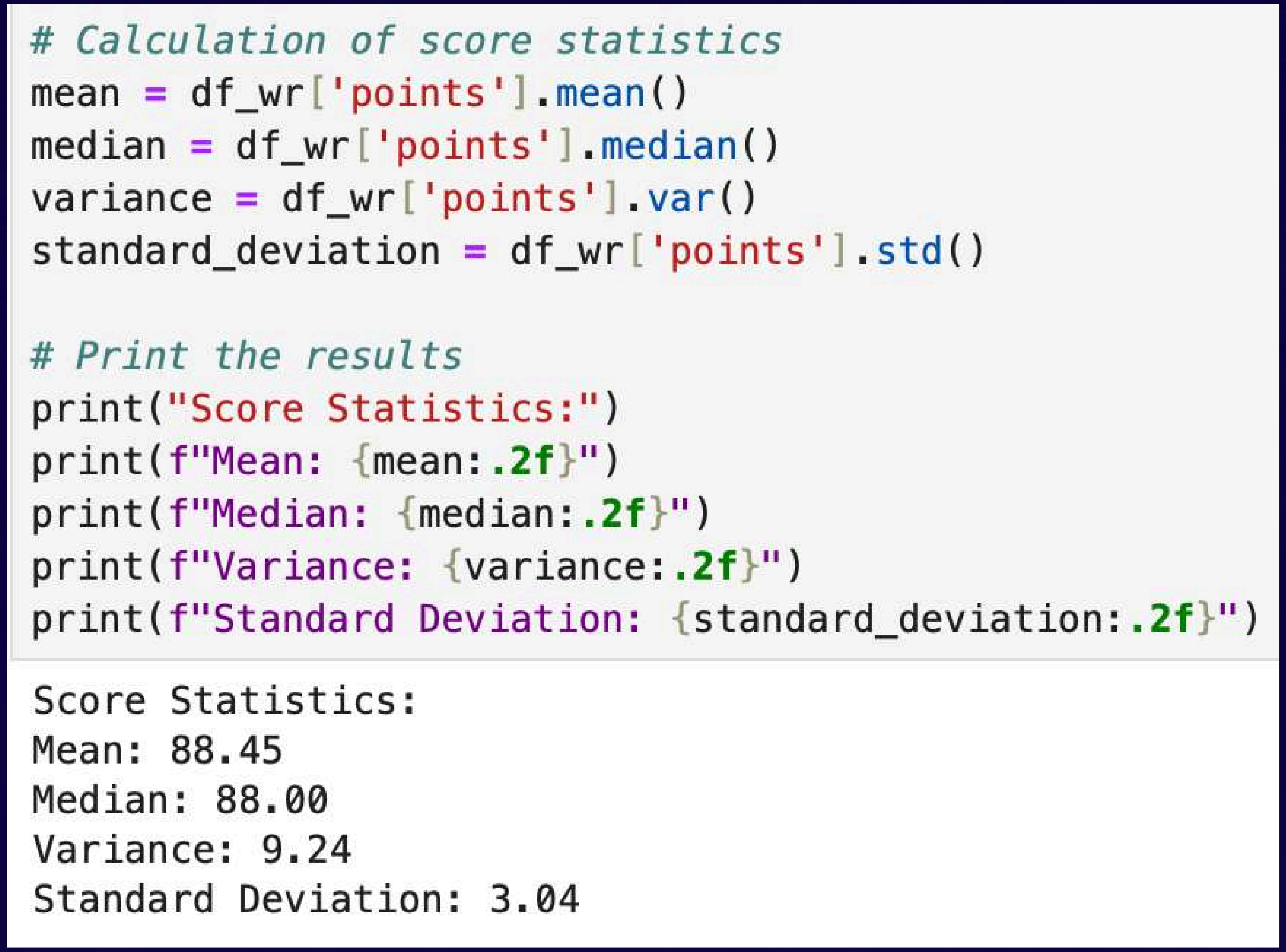
	Argentina	Australia	Austria	Slovenia	Armenia	Slovakia
0	(Armenia, 75)	(Austria, 88)	(Australia, 88)	(Slovakia, 75)	(Argentina, 75)	(Slovenia, 75)
1	(Macedonia, 56)	(Israel, 53)	(Serbia, 62)	(Serbia, 57)	(Macedonia, 62)	(Romania, 53)
2	(Austria, 50)	(Serbia, 53)	(Croatia, 57)	(Romania, 53)	(Austria, 57)	(Croatia, 53)

There are no duplicate or badly formatted entries between country names.

4. DATA EXPLORATION AND DATA VISUALIZATION

Before actually starting the analysis, it's a good practice to check if the data are ready or not.

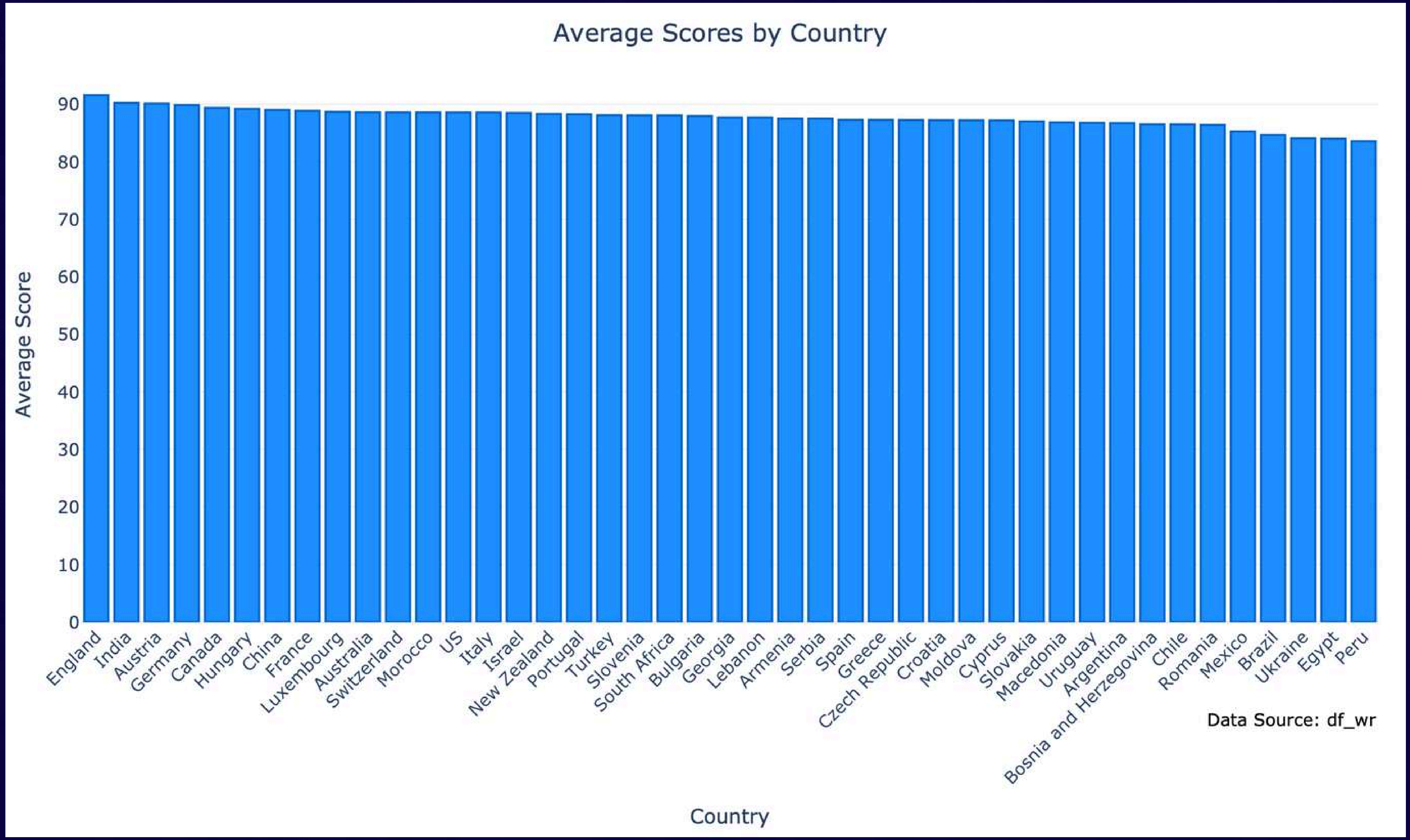
They might have some invalid values, missing values, mistyped and so on. In this section, I will investigate if data are consistent and ready to be processed.



```python
# Calculation of score statistics
mean = df_wr['points'].mean()
median = df_wr['points'].median()
variance = df_wr['points'].var()
standard_deviation = df_wr['points'].std()

# Print the results
print("Score Statistics:")
print(f"Mean: {mean:.2f}")
print(f"Median: {median:.2f}")
print(f"Variance: {variance:.2f}")
print(f"Standard Deviation: {standard_deviation:.2f}")
```

```
Score Statistics:
Mean: 88.45
Median: 88.00
Variance: 9.24
Standard Deviation: 3.04
```

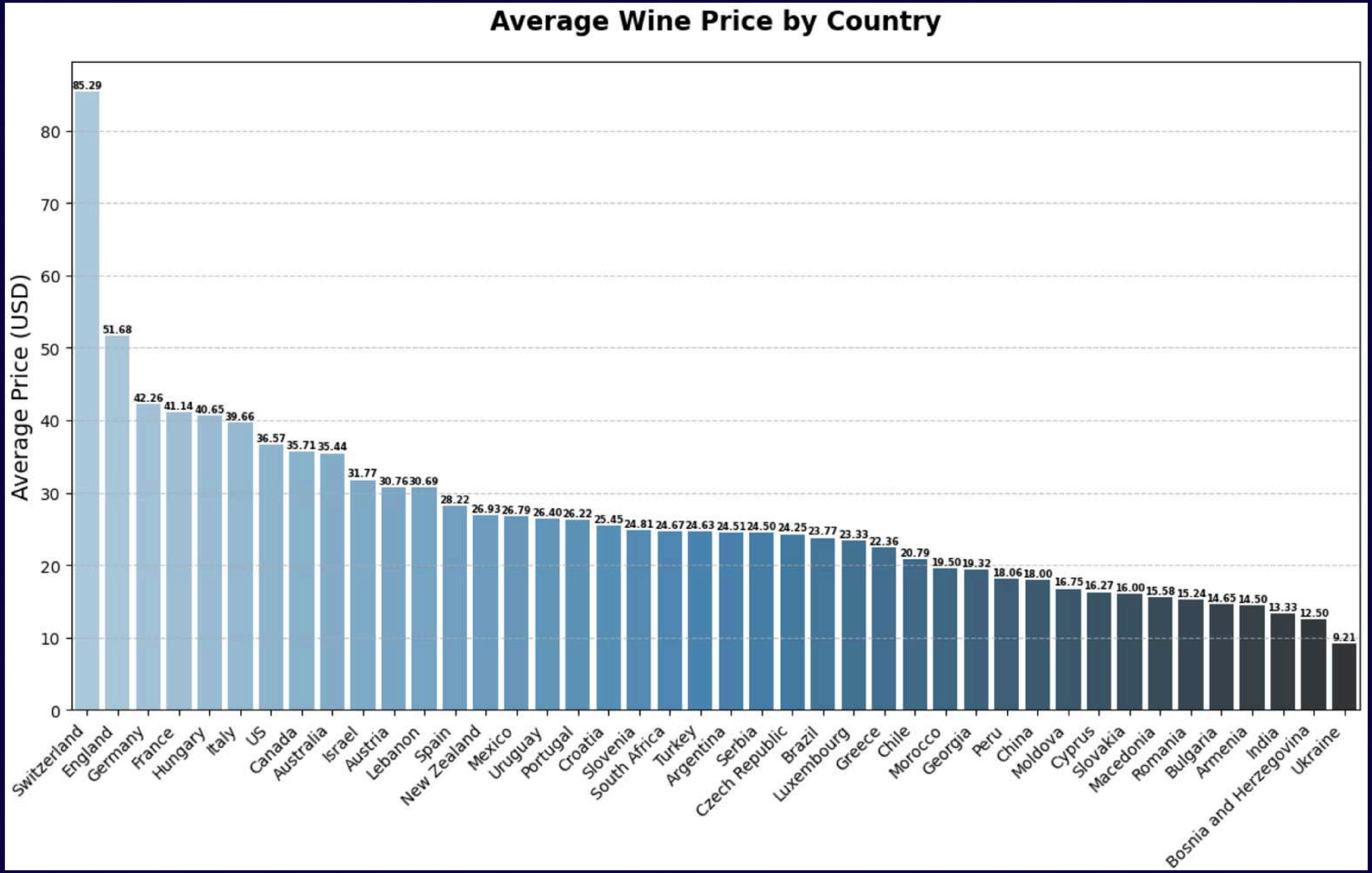
Average Score per Country



Data Source: df_wr

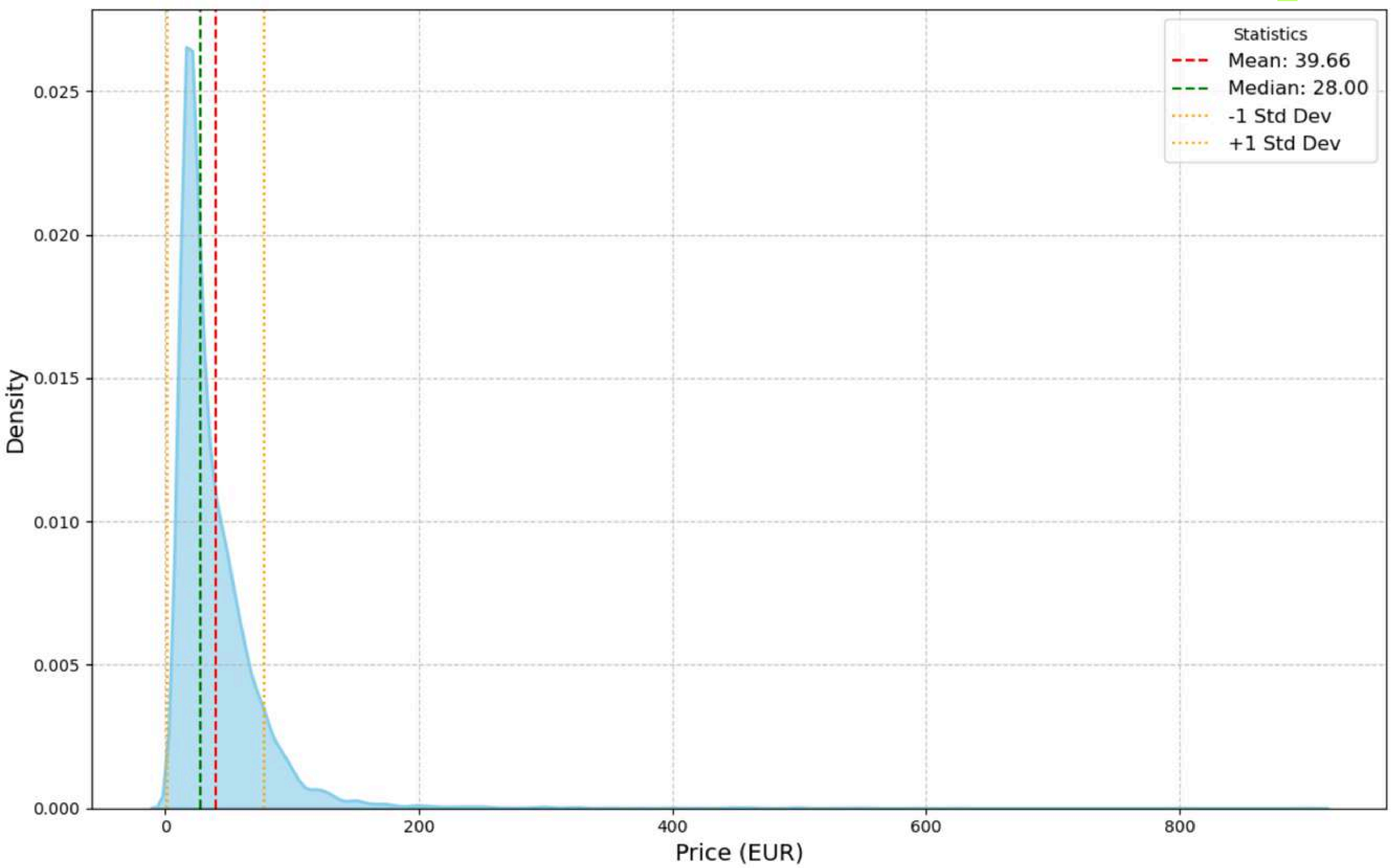
Distributed Excellence: There is a clear trend towards excellence, with all the analyzed countries surpassing the threshold of 80 points. This suggests a constant commitment to producing high-quality wines at an international level. The bar chart presented provides an effective visualization of the average scores obtained by a sample of countries. The data was processed using statistical techniques to calculate the average score for each nation. The choice of a bar chart makes it easy to compare the performance of different countries and helps identify the nations with the highest and lowest scores. The visualization is clear and intuitive, making it easy for even a non-expert audience to understand the results.

Average wine price per Country



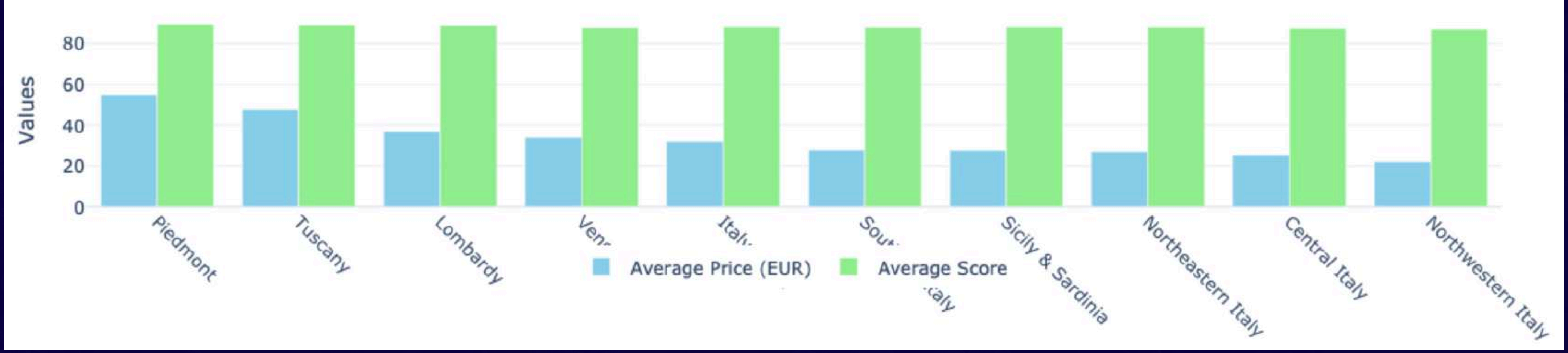
The chart shows the average wine price divided by country. There is considerable variability in average prices between different countries. Among the countries with the highest prices, Switzerland stands out, with a very high price largely due to the cost of living, and the limited variety offered, which makes it difficult to compete with countries like France, Italy, and Portugal, which offer more affordable prices based on the variety. This comparison provides an interesting insight into how wine prices can be influenced by various factors, including international recognition, production costs, and product positioning.

Distribution of Italian wine prices



The chart shows the distribution of Italian wine prices. We can observe significant variability, with prices ranging from just a few euros to several hundred. The averagestands at € 39.66, indicating the mean price of a bottle of Italian wine. The median, at € 28, reveals that half of the wines are priced below this threshold. The fact that the average price (€39.66) is higher than the median price (€28) indicates a positively skewed distribution. This suggests that the Italian wine market includes a number of high-priced outliers, such as premium or luxury wines, which significantly raise the average price. However, the median provides a better representation of the typical price, as it is not affected by these extreme values. This highlights the wide range of options in the Italian wine market, catering to both everyday consumers and those seeking exclusive, high-end products. The high variance (1447.99) suggests substantial dispersion around the mean, indicating a wide heterogeneity in the offerings.

Average Wine Prices and Scores by Province in Italy



This graph displays the average wine prices (in EUR) and average wine scores across various provinces in Italy. Here's a detailed analysis:

Average Scores (Green Bars): The average scores for wines in all provinces are consistently high, hovering around 90 points. This suggests that wines from all listed provinces are generally of high quality, without significant variation in scoring.

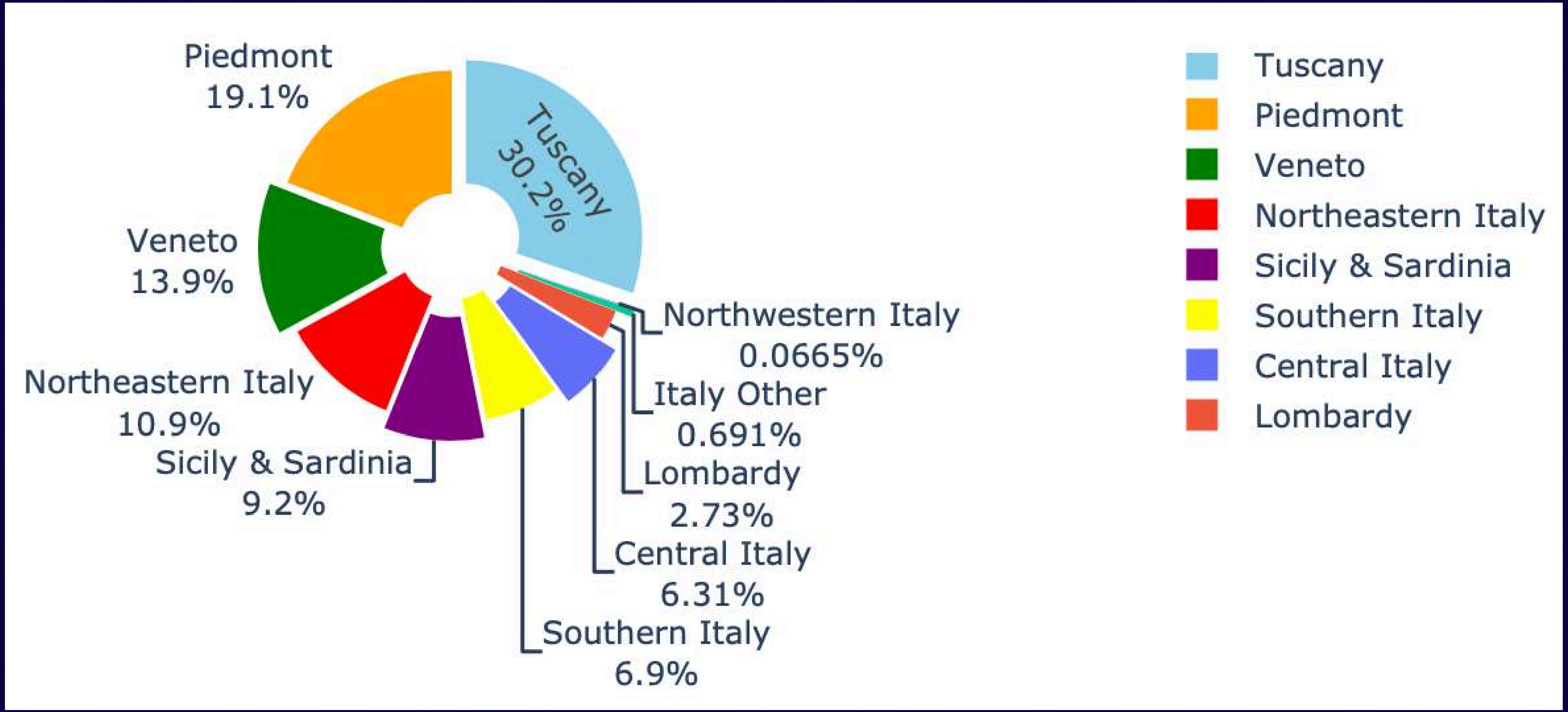
Average Prices (Blue Bars): The average wine prices vary considerably across provinces:

- *Piedmont has the highest average wine prices among the provinces.*
- *Tuscany also has relatively high wine prices, although slightly lower than Piedmont.*
- *Lombardy, Veneto and Italy Other have moderate average prices.*
- *Regions such as Southern Italy, Sicily & Sardinia, Northeastern Italy, Central Italy, and Northwestern Italy have lower average prices compared to Piedmont and Tuscany.*

Comparison:

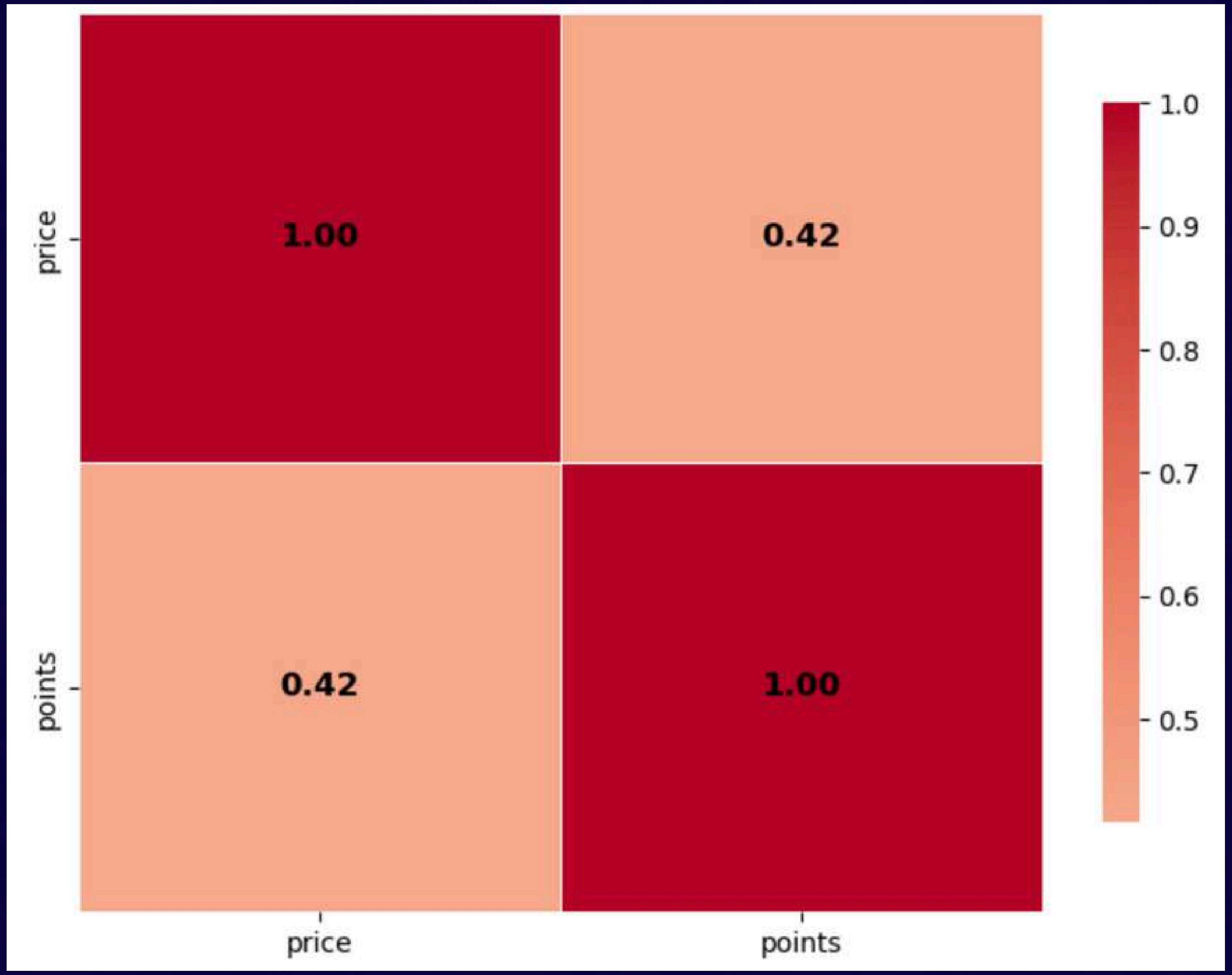
- *Despite the variation in prices, the quality scores (green bars) remain consistently high, suggesting that even wines from provinces with lower prices maintain excellent quality.*
- *Piedmont and Tuscany might represent premium wine regions where higher prices align with prestige or demand rather than drastic differences in quality.*

Distribution of bottles sold by province in Italy



The pie chart effectively conveys the proportion of wine bottles sold across different provinces in Italy, making it visually appealing and easy to understand. Tuscany's dominance (30.2%) is immediately clear, followed by regions such as Piedmont (19.1%) and Veneto (13.9%), showcasing their importance in Italy's wine production and distribution. The use of percentages and a distinct color palette aids in interpreting the data at a glance.

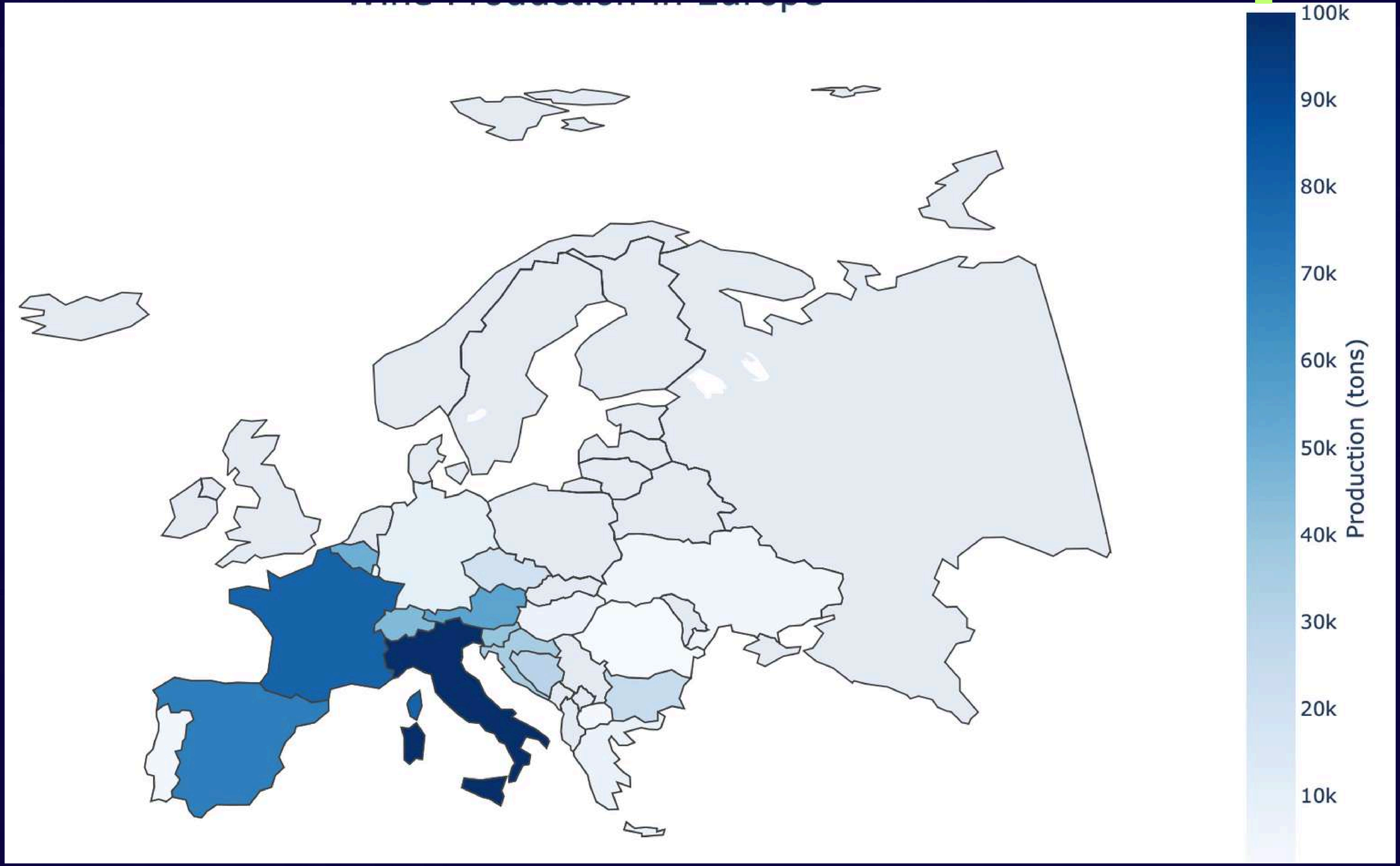
Correlation between price and wine scores in Italy



The heatmap provides a clear visual representation of the correlation between wine prices and scores. The value of 0.42 for the correlation between price and score is accurately displayed, indicating a moderate positive relationship. The value of 0.42 suggests that higher-priced wines tend to have higher scores, but the relationship is not very strong or perfect. This means:

- *Price does not guarantee quality, as there are many affordable wines with high scores.*
- *Other factors, such as production methods, grape variety, or regional prestige, likely play a significant role in determining both the price and quality.*

Wine Production in Europe



This map provides an overview of wine production across Europe. It highlights how Italy, France, and Spain dominate the scene, but also other countries like Germany and Portugal have significant production. Italy confirms its position as the largest wine producer in Europe, followed by France and Spain. Production is primarily concentrated in Southern European countries, gradually decreasing as we move northward. This distribution is influenced by climatic, historical, and cultural factors.

5. CONCLUSIONS

The results highlight widespread excellence in the international wine industry, with Italy confirming its leadership in both production and variety.

However, significant regional disparities emerge, along with a correlation between price and quality.

Market segmentation, where regions specialize in high-end production and others in more affordable wines, offers consumers a broad selection and provides producers the opportunity to position themselves in specific niches to meet the demands of an increasingly discerning clientele.

On the international stage, Italy remains a point of reference, but the competition is fierce, requiring continuous investment in quality and innovation. Additionally, international competition necessitates constant adaptation to emerging consumption trends and greater attention to sustainability.

Thank you for your attention

Click here to view the full Python notebook

Emma Trivini Bellini