In [281…
```
%time
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import zscore
```

```
CPU times: user 4 µs, sys: 0 ns, total: 4 µs
Wall time: 8.11 µs
```

Data Cleaning

In [282…
```
df = pd.read_csv("../data/merged_data_cleaned.csv")
df.head()
```

Out[282…

	Unnamed: 0	Species	Owner	Country.of.Origin	Farm.Name	Lot.Number	Mi
0	0	Arabica	metad plc	Ethiopia	metad plc	NaN	meta p
1	1	Arabica	metad plc	Ethiopia	metad plc	NaN	meta p
2	2	Arabica	grounds for health admin	Guatemala	san marcos barrancas "san cristobal cuch	NaN	Na
3	3	Arabica	yidnekachew dabessa	Ethiopia	yidnekachew dabessa coffee plantation	NaN	wolens
4	4	Arabica	metad plc	Ethiopia	metad plc	NaN	meta p

5 rows × 44 columns

We have a lot of columns with data that are irrelevant for our analysis. We'll drop them to reduce dimensionality of the dataset

In [283…
```
df = df.drop(["Unnamed: 0", "Farm.Name", "Lot.Number", "Mill", "ICO.Numbe
```

In [284…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1339 entries, 0 to 1338
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Species               1339 non-null   object 
 1   Country.of.Origin     1338 non-null   object 
 2   Variety               1113 non-null   object 
 3   Processing.Method     1169 non-null   object 
 4   Aroma                 1339 non-null   float64
 5   Flavor                1339 non-null   float64
 6   Aftertaste            1339 non-null   float64
 7   Acidity               1339 non-null   float64
 8   Body                  1339 non-null   float64
 9   Balance               1339 non-null   float64
 10  Uniformity            1339 non-null   float64
 11  Clean.Cup             1339 non-null   float64
 12  Sweetness             1339 non-null   float64
 13  Cupper.Points         1339 non-null   float64
 14  Total.Cup.Points      1339 non-null   float64
 15  Moisture              1339 non-null   float64
 16  Category.One.Defects  1339 non-null   int64  
 17  Quakers               1338 non-null   float64
 18  Color                 1069 non-null   object 
 19  Category.Two.Defects  1339 non-null   int64  
 20  altitude_mean_meters  1109 non-null   float64
dtypes: float64(14), int64(2), object(5)
memory usage: 219.8+ KB
```

In [285…

```
df.isna().sum()
```

Out[285…

```
Species                   0
Country.of.Origin         1
Variety                 226
Processing.Method       170
Aroma                     0
Flavor                    0
Aftertaste                0
Acidity                   0
Body                      0
Balance                   0
Uniformity                0
Clean.Cup                 0
Sweetness                 0
Cupper.Points             0
Total.Cup.Points          0
Moisture                  0
Category.One.Defects      0
Quakers                   1
Color                   270
Category.Two.Defects      0
altitude_mean_meters    230
dtype: int64
```

We replace rows with a lot of missing nominal data with the mode of the column, to retain the variance of the rest of the row data

In [286…

```
df = df.fillna({'Variety': df['Variety'].mode()[0]})
df = df.fillna({'Processing.Method': df['Processing.Method'].mode()[0]})
```

```
df = df.fillna({'Color': df['Color'].mode()[0]})
```

We drop the rows with a single row missing nominal data. Especially "Country of Origin", since we can not just put in a mode value, since it might create significant wrong data

In [287…
```
df = df.dropna(how='any')
```

In [288…
```
df.isna().sum()
```

Out[288…
```
Species                 0
Country.of.Origin       0
Variety                 0
Processing.Method       0
Aroma                   0
Flavor                  0
Aftertaste              0
Acidity                 0
Body                    0
Balance                 0
Uniformity              0
Clean.Cup               0
Sweetness               0
Cupper.Points           0
Total.Cup.Points        0
Moisture                0
Category.One.Defects    0
Quakers                 0
Color                   0
Category.Two.Defects    0
altitude_mean_meters    0
dtype: int64
```

No more missing values!

Outliers

In [289…
```
df.describe()
```

Out[289…

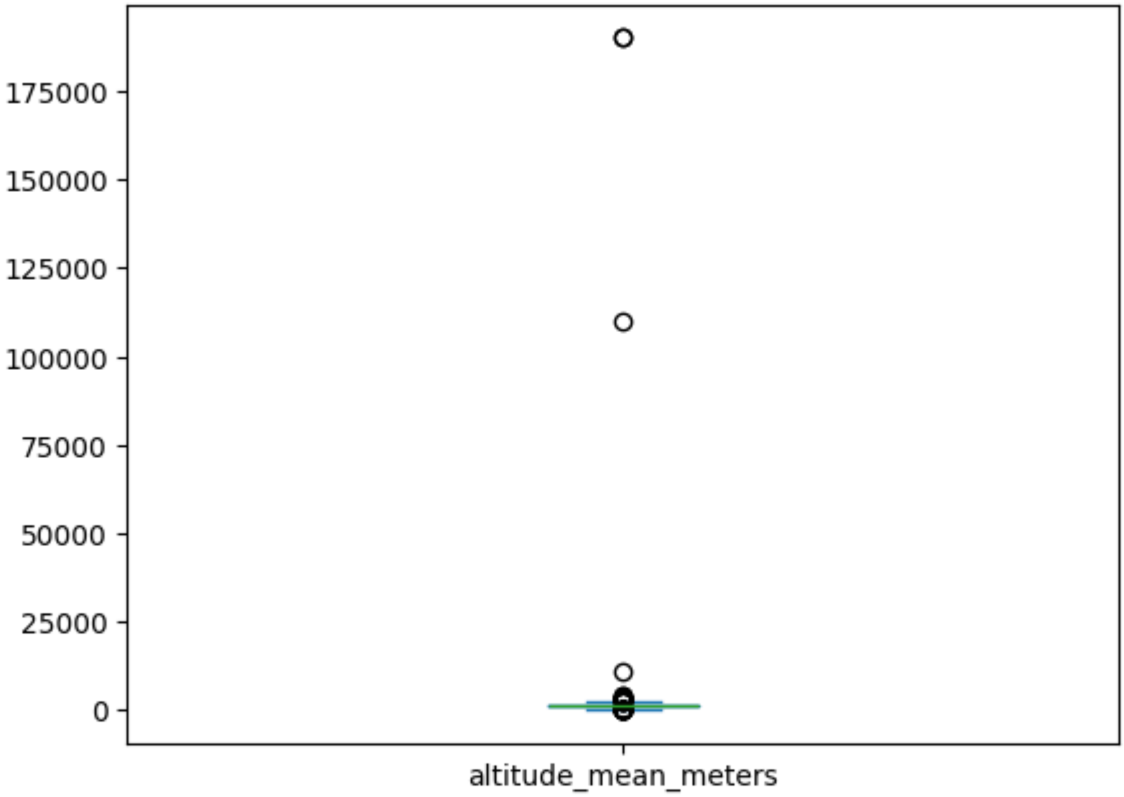
	Aroma	Flavor	Aftertaste	Acidity	Body	Balance
count	1108.000000	1108.00000	1108.000000	1108.000000	1108.000000	1108.000000
mean	7.570569	7.52056	7.394269	7.528953	7.506670	7.505542
std	0.383837	0.40059	0.405867	0.386075	0.366717	0.419311
min	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000
25%	7.420000	7.33000	7.250000	7.330000	7.330000	7.330000
50%	7.580000	7.58000	7.420000	7.500000	7.500000	7.500000
75%	7.750000	7.75000	7.580000	7.750000	7.670000	7.750000
max	8.750000	8.83000	8.670000	8.750000	8.580000	8.750000

We have some columns with very high standard deviations

In [290…
```
fig = plt.figure()
```

```
df.altitude_mean_meters.plot.box()
```

Out[290… <Axes: >



The column contains significant outliers. Since its only a couple of rows, we'll drop them

In [291…
```
# We'll remove the outlying rows based on z-score
df = df[np.abs(zscore(df['altitude_mean_meters'])) < 1]
```

In [292…
```
df
```

Out[292…

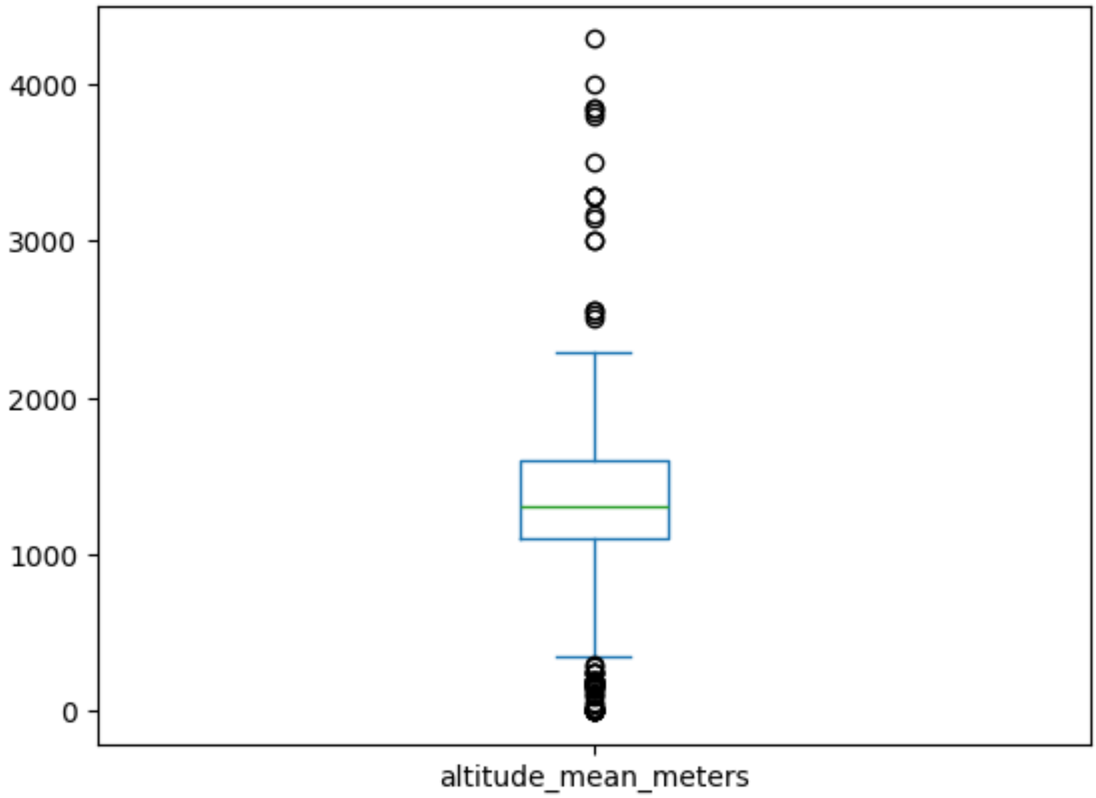
	Species	Country.of.Origin	Variety	Processing.Method	Aroma	Flavor	Aftertaste
0	Arabica	Ethiopia	Caturra	Washed / Wet	8.67	8.83	8.67
1	Arabica	Ethiopia	Other	Washed / Wet	8.75	8.67	8.50
2	Arabica	Guatemala	Bourbon	Washed / Wet	8.42	8.50	8.42
3	Arabica	Ethiopia	Caturra	Natural / Dry	8.17	8.58	8.42
4	Arabica	Ethiopia	Other	Washed / Wet	8.25	8.50	8.25
...	...	...	...	...	...	...	...
1331	Robusta	India	Caturra	Washed / Wet	7.67	7.67	7.50
1332	Robusta	India	Caturra	Natural / Dry	7.58	7.42	7.42
1333	Robusta	United States	Arusha	Natural / Dry	7.92	7.50	7.42
1335	Robusta	Ecuador	Caturra	Washed / Wet	7.50	7.67	7.75
1336	Robusta	United States	Caturra	Natural / Dry	7.33	7.33	7.17

1104 rows × 21 columns

In [293…

```
df.altitude_mean_meters.plot.box()
```

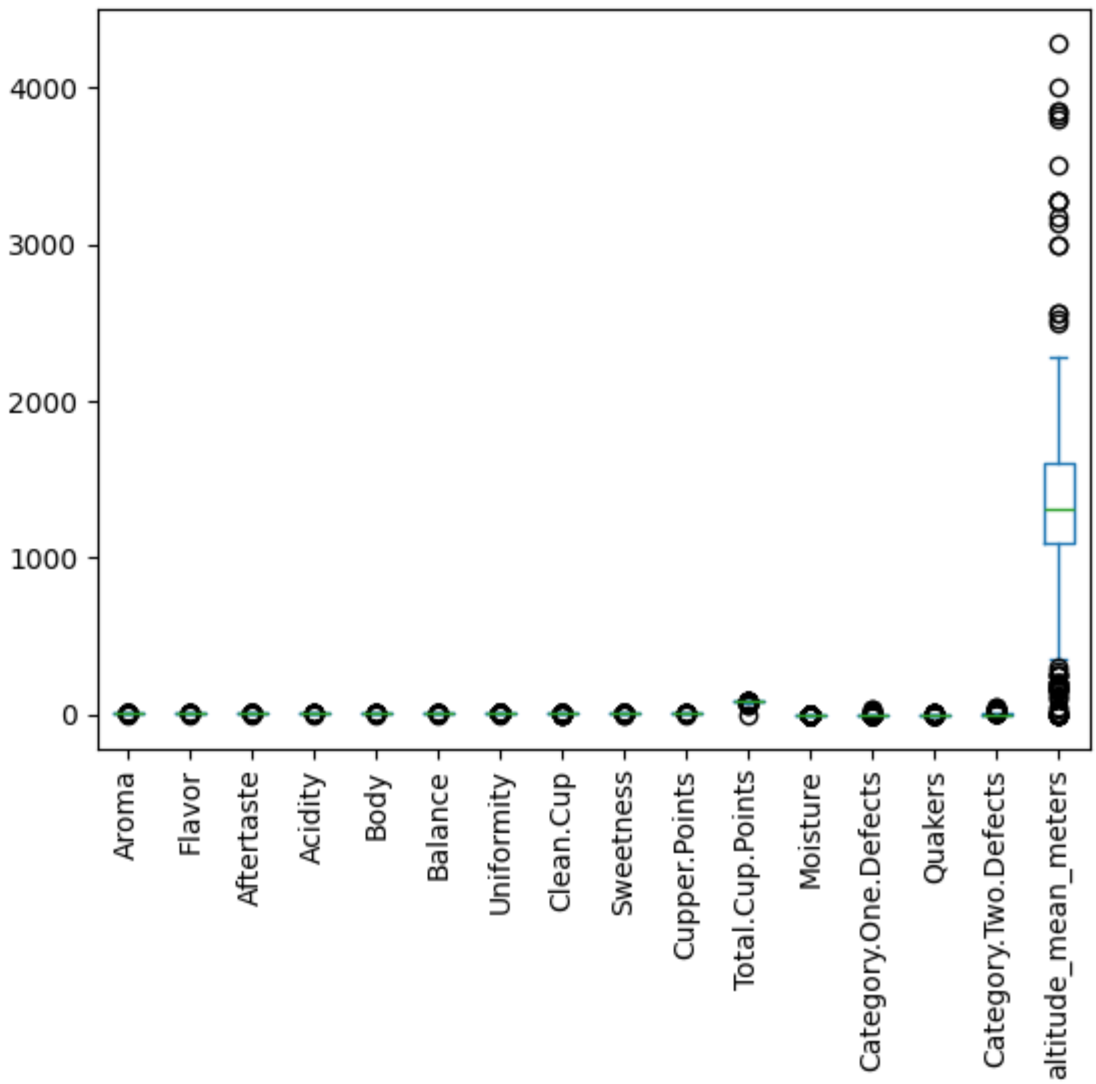
Out[293… <Axes: >



In [294…

```
df.plot.box(rot=90)
```

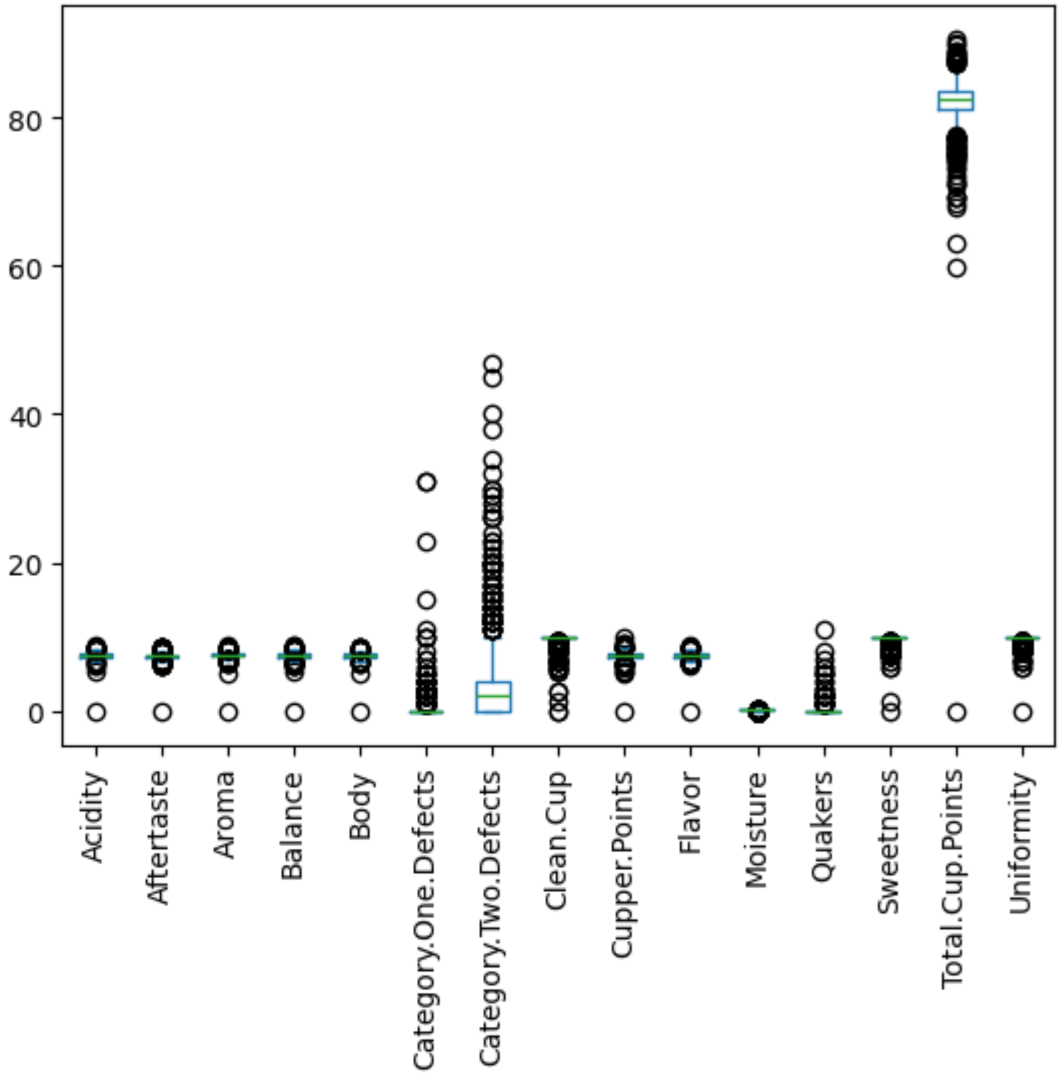
Out[294… <Axes: >



Even after removing the worst outliers, the mean altitude still distributes over large values. We'll exclude it in the plot to dentify other problematic features

In [295…
```
df[df.columns.difference(['altitude_mean_meters'])].plot.box(rot=90)
```

Out[295…
```
<Axes: >
```



It seems a lot of the features contain unnatural zero-values. We'll replace the zero values, with the median of the feature

In [296…

```
df['Acidity'] = df['Acidity'].replace(0, df['Acidity'].median())
df['Aftertaste'] = df['Aftertaste'].replace(0, df['Aftertaste'].median())
df['Aroma'] = df['Aroma'].replace(0, df['Aroma'].median())
df['Balance'] = df['Balance'].replace(0, df['Balance'].median())
df['Body'] = df['Body'].replace(0, df['Body'].median())
df['Clean.Cup'] = df['Clean.Cup'].replace(0, df['Clean.Cup'].median())
df['Cupper.Points'] = df['Cupper.Points'].replace(0, df['Cupper.Points'].
df['Flavor'] = df['Flavor'].replace(0, df['Flavor'].median())
df['Moisture'] = df['Moisture'].replace(0, df['Moisture'].median())
df['Sweetness'] = df['Sweetness'].replace(0, df['Sweetness'].median())
df['Uniformity'] = df['Uniformity'].replace(0, df['Uniformity'].median())
```

Except the Total cup points. We'll drop the row since it is our target value, and an unnatural zero might mess with correlations

In [297…

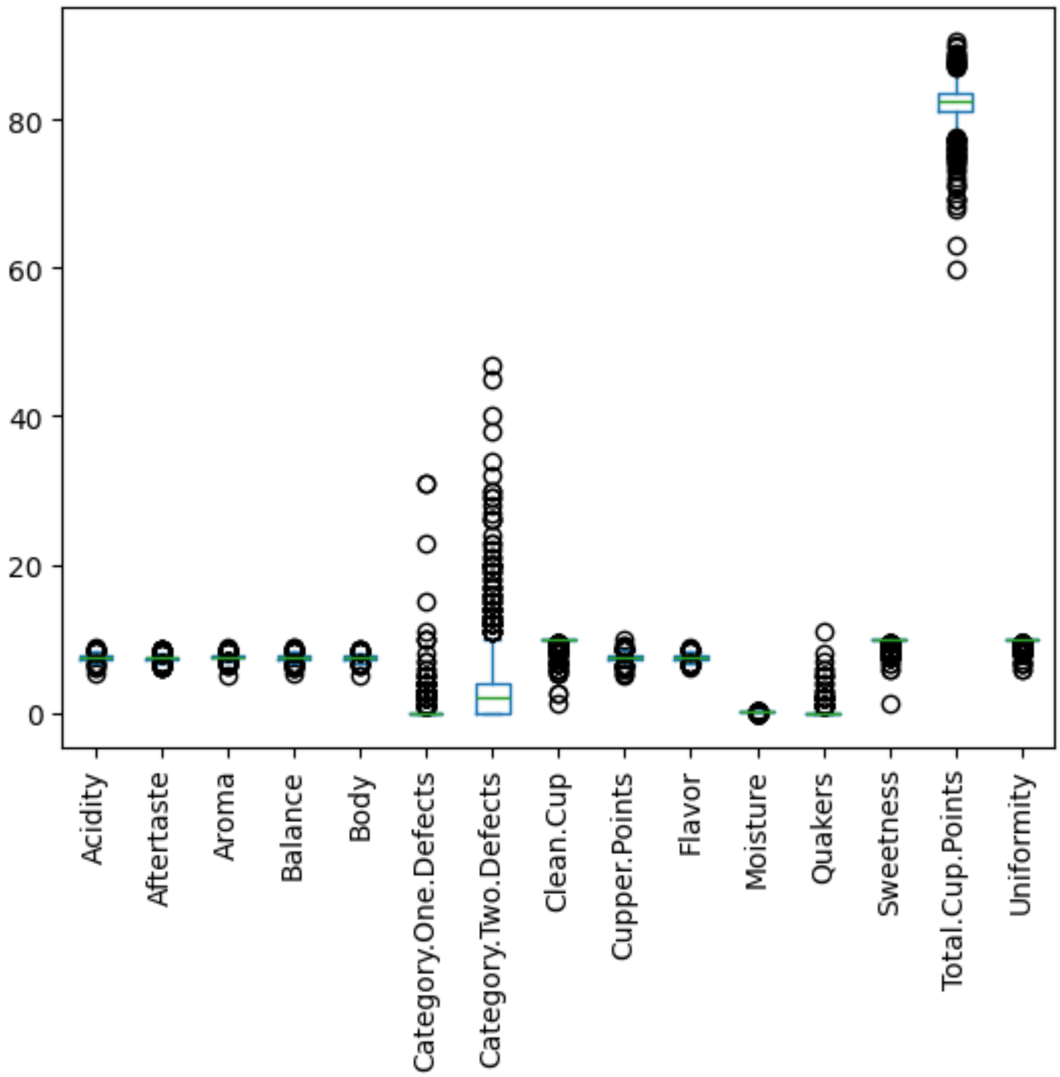
```
df = df[df['Total.Cup.Points'] != 0]
```

In [298…

```
df[df.columns.difference(['altitude_mean_meters'])].plot.box(rot=90)
```

Out[298…

```
<Axes: >
```



In [299…
```
df.describe()
```

Out[299…

	Aroma	Flavor	Aftertaste	Acidity	Body	Balanc
count	1103.000000	1103.000000	1103.000000	1103.000000	1103.000000	1103.00000
mean	7.578368	7.527824	7.401496	7.535739	7.513654	7.51266
std	0.309181	0.331268	0.340065	0.313192	0.289467	0.35402
min	5.080000	6.170000	6.170000	5.250000	5.170000	5.25000
25%	7.420000	7.330000	7.250000	7.330000	7.330000	7.33000
50%	7.580000	7.580000	7.420000	7.500000	7.500000	7.50000
75%	7.750000	7.750000	7.580000	7.750000	7.670000	7.75000
max	8.750000	8.830000	8.670000	8.750000	8.580000	8.75000

In [300…
```
df.to_csv('cleaned_dataset_no_zeros.csv', index=False)
```

The datset is now ready for analysis!