Exploring Data

In [166…

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import math
```

In [167…

```
pd.set_option('display.max_columns', None)
df = pd.read_csv("df_arabica_clean.csv")
df.head()
```

Out[167…

	Unnamed: 0	ID	Country of Origin	Farm Name	Lot Number	Mill	ICO Number	Company
0	0	0	Colombia	Finca El Paraiso	CQU2022015	Finca El Paraiso	NaN	Coffee Quality Union
1	1	1	Taiwan	Royal Bean Geisha Estate	The 2022 Pacific Rim Coffee Summit,T037	Royal Bean Geisha Estate	NaN	Taiwan Coffee Laboratory
2	2	2	Laos	OKLAO coffee farms	The 2022 Pacific Rim Coffee Summit,LA01	oklao coffee processing plant	NaN	Taiwan Coffee Laboratory
3	3	3	Costa Rica	La Cumbre	CQU2022017	La Montana Tarrazu MIll	NaN	Coffee Quality Union
4	4	4	Colombia	Finca Santuario	CQU2023002	Finca Santuario	NaN	Coffee Quality Union

In [168…

```
df.columns
```

Out[168…

```
Index(['Unnamed: 0', 'ID', 'Country of Origin', 'Farm Name', 'Lot Number',
       'Mill', 'ICO Number', 'Company', 'Altitude', 'Region', 'Producer',
       'Number of Bags', 'Bag Weight', 'In-Country Partner', 'Harvest Yea
r',
       'Grading Date', 'Owner', 'Variety', 'Status', 'Processing Method',
       'Aroma', 'Flavor', 'Aftertaste', 'Acidity', 'Body', 'Balance',
       'Uniformity', 'Clean Cup', 'Sweetness', 'Overall', 'Defects',
       'Total Cup Points', 'Moisture Percentage', 'Category One Defects',
       'Quakers', 'Color', 'Category Two Defects', 'Expiration',
       'Certification Body', 'Certification Address', 'Certification Contac
t'],
      dtype='object')
```

In [169…

```python
country_count = df["Country of Origin"].value_counts().reset_index()
country_count.columns = ["Country of Origin", "Count"]
country_count
```

Out[169…

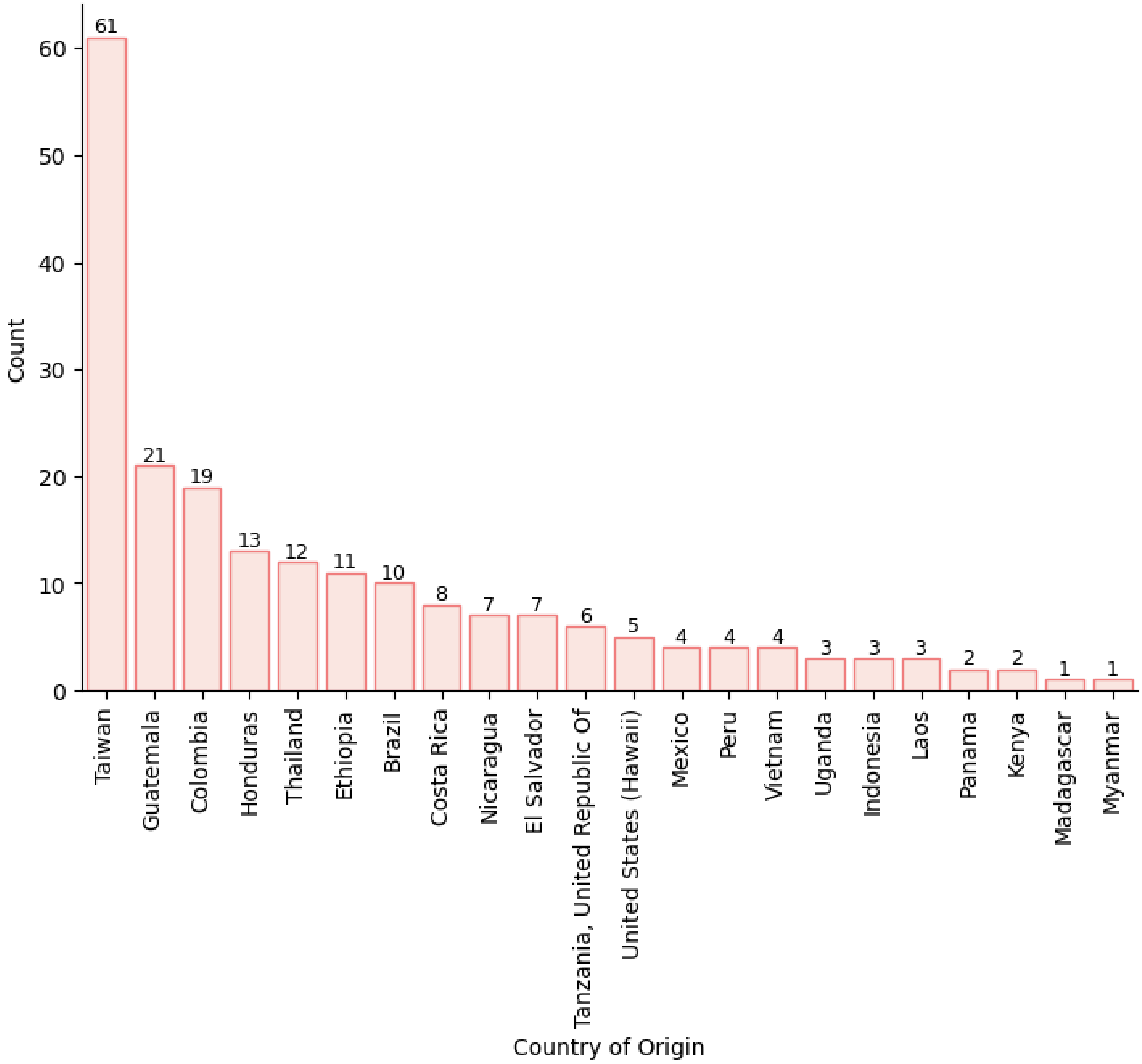
	Country of Origin	Count
0	Taiwan	61
1	Guatemala	21
2	Colombia	19
3	Honduras	13
4	Thailand	12
5	Ethiopia	11
6	Brazil	10
7	Costa Rica	8
8	Nicaragua	7
9	El Salvador	7
10	Tanzania, United Republic Of	6
11	United States (Hawaii)	5
12	Mexico	4
13	Peru	4
14	Vietnam	4
15	Uganda	3
16	Indonesia	3
17	Laos	3
18	Panama	2
19	Kenya	2
20	Madagascar	1
21	Myanmar	1

In [170…

```python
g = sns.catplot(data = country_count, x = "Country of Origin", y = "Count",
plt.xticks(rotation = 90)
ax = g.axes[0,0]
for p in ax.patches:
    value = int(p.get_height())
    ax.text(
        p.get_x() + p.get_width()/2,
        p.get_height(),
        value,
        ha="center", va="bottom",
        fontsize=9
    )
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/se
aborn/axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```



In [171…

```
country_count = df["Country of Origin"].value_counts().reset_index()
country_count.columns = ["Country of Origin", "Count"]
country_bags = df.groupby("Country of Origin", as_index=False)["Number of Ba
country_summary = country_count.merge(country_bags, on="Country of Origin")
country_summary.columns = ["Country of Origin", "Count", "Total Bags"]
country_summary
```

Out[171…

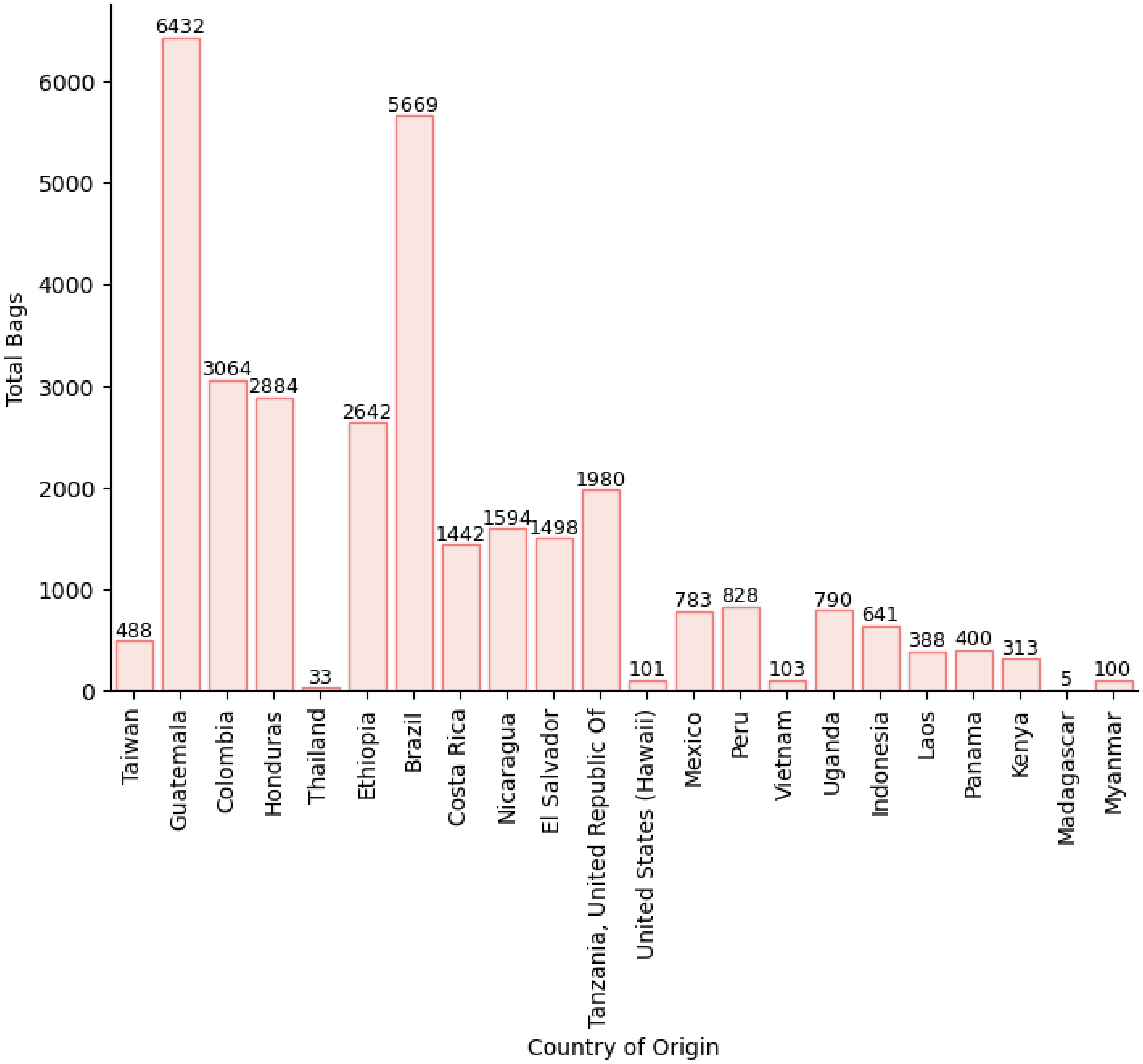
	Country of Origin	Count	Total Bags
0	Taiwan	61	488
1	Guatemala	21	6432
2	Colombia	19	3064
3	Honduras	13	2884
4	Thailand	12	33
5	Ethiopia	11	2642
6	Brazil	10	5669
7	Costa Rica	8	1442
8	Nicaragua	7	1594
9	El Salvador	7	1498
10	Tanzania, United Republic Of	6	1980
11	United States (Hawaii)	5	101
12	Mexico	4	783
13	Peru	4	828
14	Vietnam	4	103
15	Uganda	3	790
16	Indonesia	3	641
17	Laos	3	388
18	Panama	2	400
19	Kenya	2	313
20	Madagascar	1	5
21	Myanmar	1	100

In [172…

```python
g = sns.catplot(data = country_summary, x = "Country of Origin", y = "Total
plt.xticks(rotation = 90)
ax = g.axes[0,0]
for p in ax.patches:
    value = int(p.get_height())
    ax.text(
        p.get_x() + p.get_width()/2,
        p.get_height(),
        value,
        ha="center", va="bottom",
        fontsize=9
    )
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/se
aborn/axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```



In [173…

```python
country_summary["Country of Origin"].value_counts().reset_index()
```

Out [173…

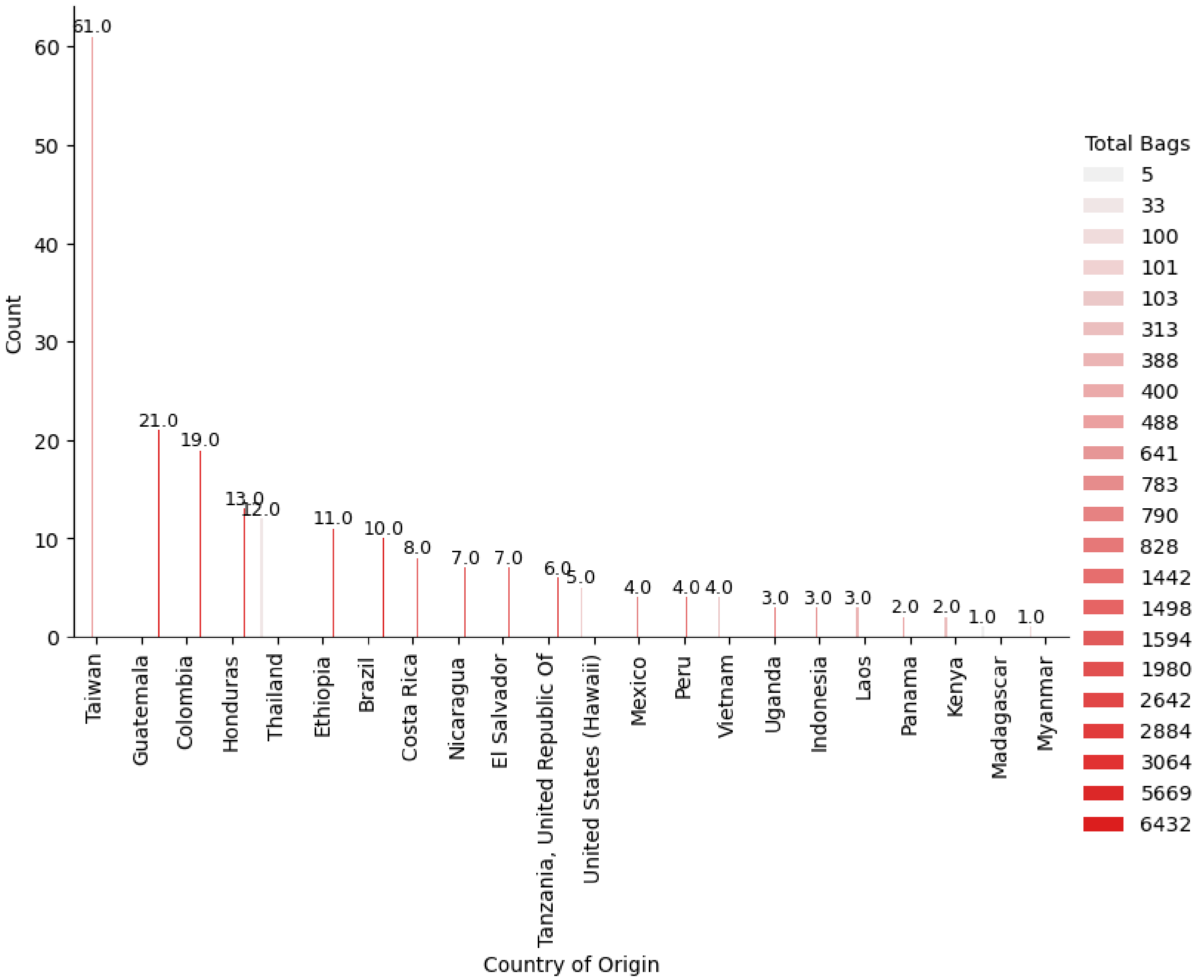
	Country of Origin	count
0	Taiwan	1
1	Guatemala	1
2	Madagascar	1
3	Kenya	1
4	Panama	1
5	Laos	1
6	Indonesia	1
7	Uganda	1
8	Vietnam	1
9	Peru	1
10	Mexico	1
11	United States (Hawaii)	1
12	Tanzania, United Republic Of	1
13	El Salvador	1
14	Nicaragua	1
15	Costa Rica	1
16	Brazil	1
17	Ethiopia	1
18	Thailand	1
19	Honduras	1
20	Colombia	1
21	Myanmar	1

In [174…

```
g = sns.catplot(data = country_summary, x = "Country of Origin", y = "Count'
plt.xticks(rotation = 90)
ax = g.axes[0,0]
for p in ax.patches:
    value = p.get_height()
    if (math.isnan(value)):
        continue
    ax.text(
        p.get_x() + p.get_width()/2,
        p.get_height(),
        value,
        ha="center", va="bottom",
        fontsize=9
    )
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/se
aborn/axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```



In [175…

```
cup_points_mean = df.groupby("Country of Origin", as_index=False)["Total Cup
cup_points_mean.columns = ["Country of Origin", "Cup Points"]
cup_points_mean
```

Out [175…

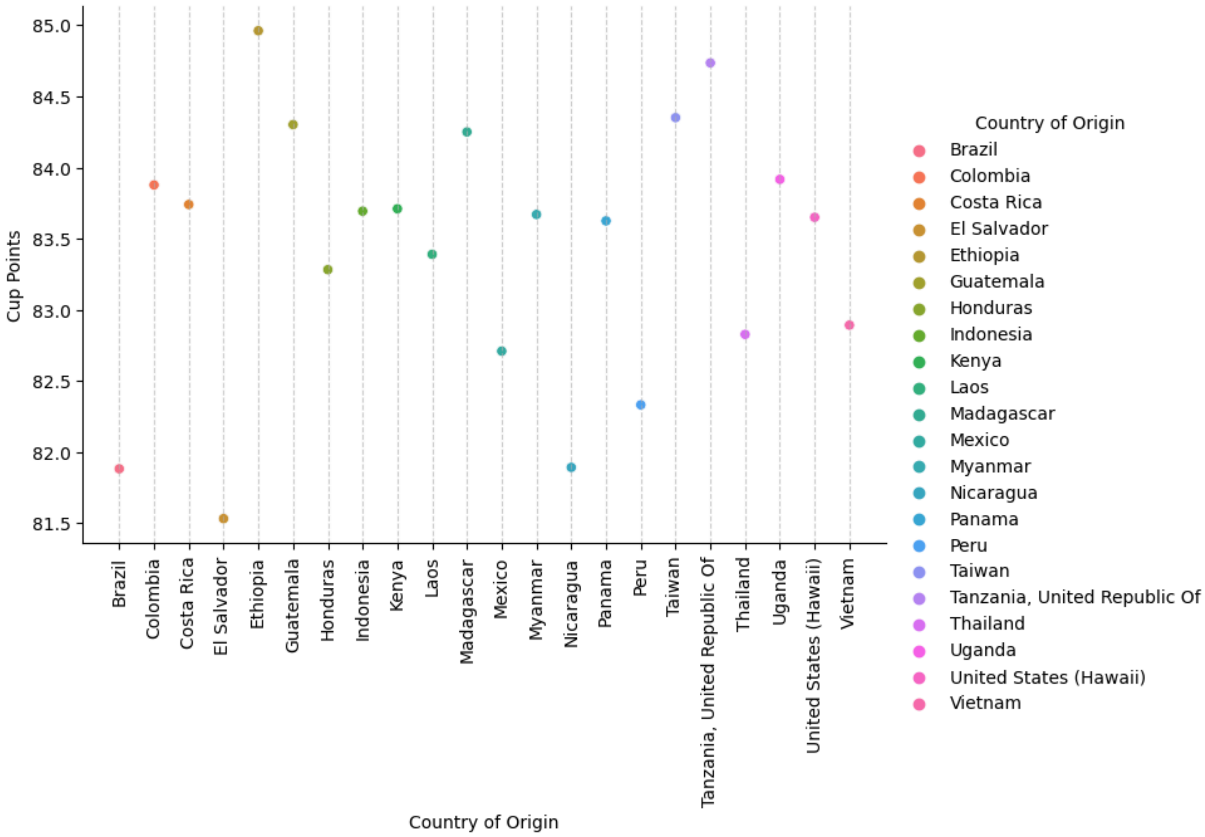
	Country of Origin	Cup Points
0	Brazil	81.883000
1	Colombia	83.877368
2	Costa Rica	83.740000
3	El Salvador	81.532857
4	Ethiopia	84.960909
5	Guatemala	84.301429
6	Honduras	83.282308
7	Indonesia	83.693333
8	Kenya	83.710000
9	Laos	83.390000
10	Madagascar	84.250000
11	Mexico	82.710000
12	Myanmar	83.670000
13	Nicaragua	81.892857
14	Panama	83.625000
15	Peru	82.332500
16	Taiwan	84.350328
17	Tanzania, United Republic Of	84.735000
18	Thailand	82.827500
19	Uganda	83.916667
20	United States (Hawaii)	83.650000
21	Vietnam	82.892500

In [176…

```
sns.relplot(cup_points_mean, x = "Country of Origin", y = "Cup Points", hue
plt.xticks(rotation = 90)
plt.grid(axis="x", linestyle="--", alpha=0.6)
```

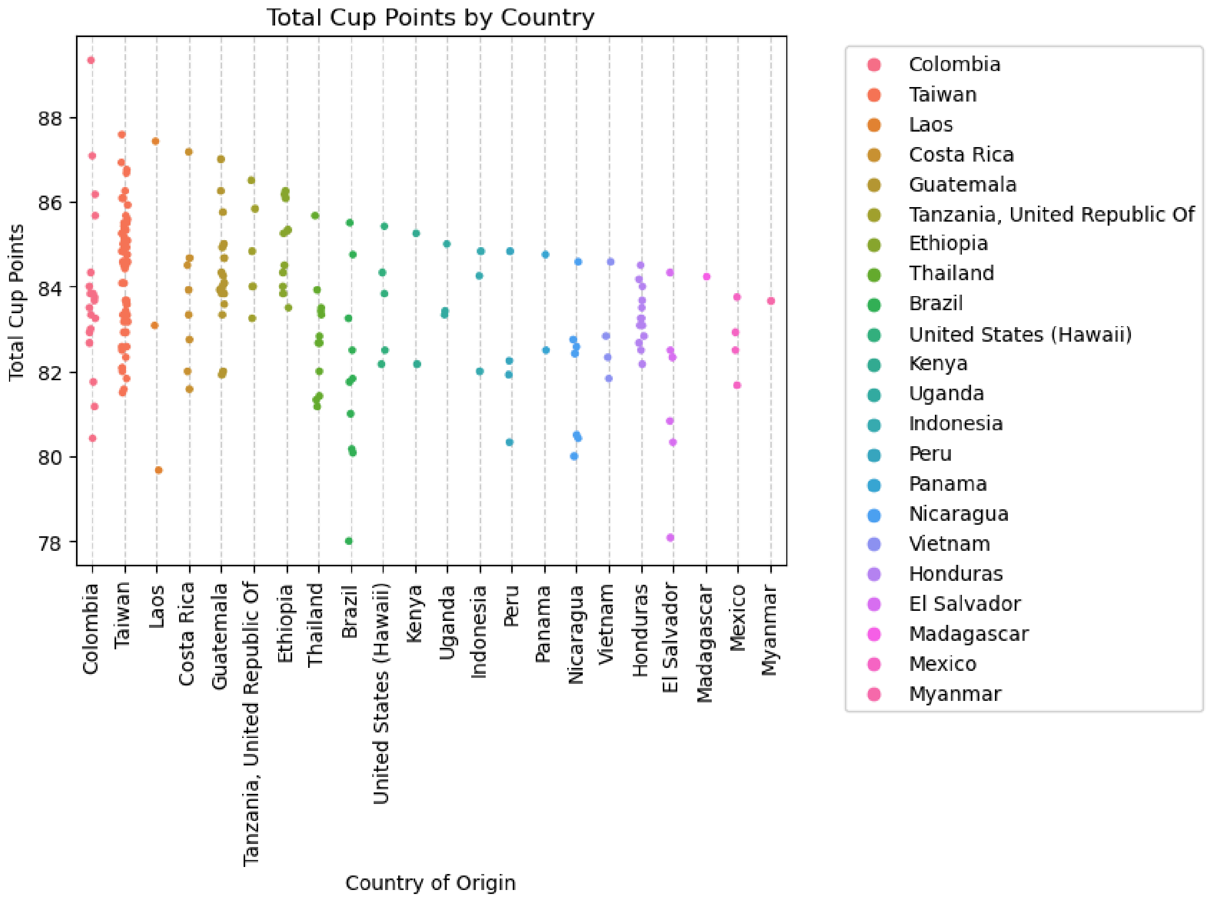
```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/se
aborn/axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```



```
In [177… sns.stripplot(df, x = 'Country of Origin', y = 'Total Cup Points', hue = 'Co
         plt.xticks(rotation = 90)
         plt.legend(bbox_to_anchor=(1.6, 1), loc = 'upper right')
         plt.grid(axis="x", linestyle="--", alpha=0.6)
         plt.title("Total Cup Points by Country")
```

```
Out[177… Text(0.5, 1.0, 'Total Cup Points by Country')
```



In [178…

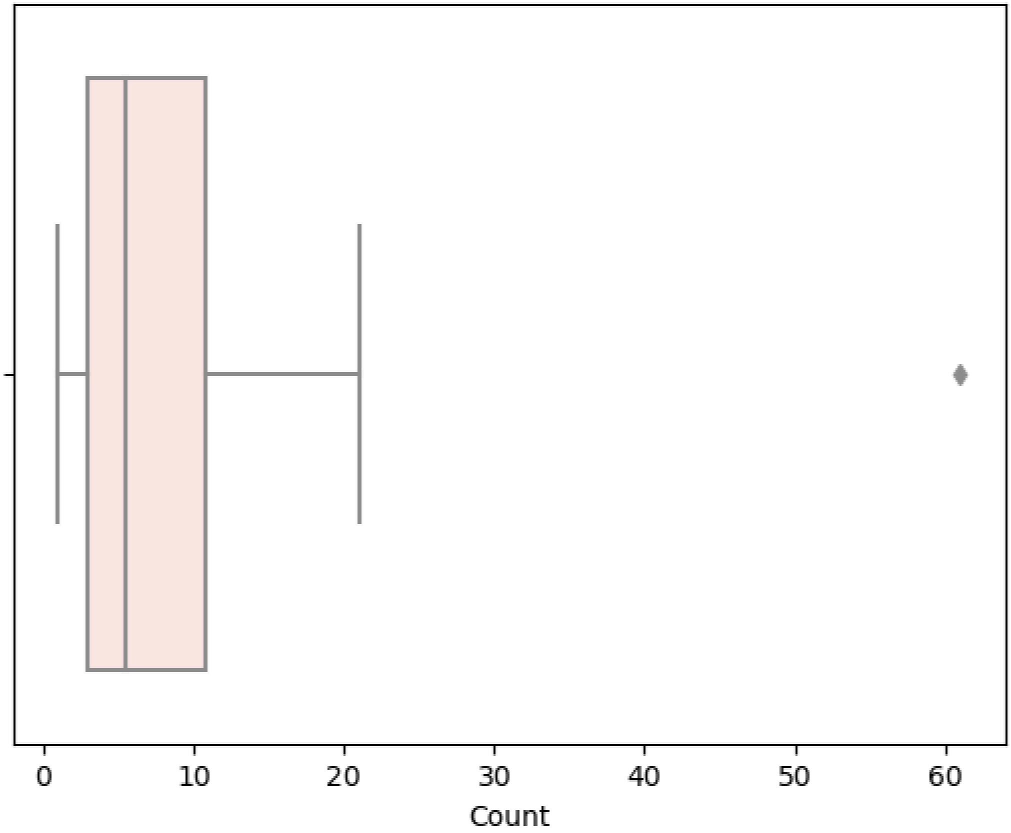
```
df["Total Cup Points"].min()
df["Total Cup Points"].max()
```

Out[178…

```
89.33
```

In [179…

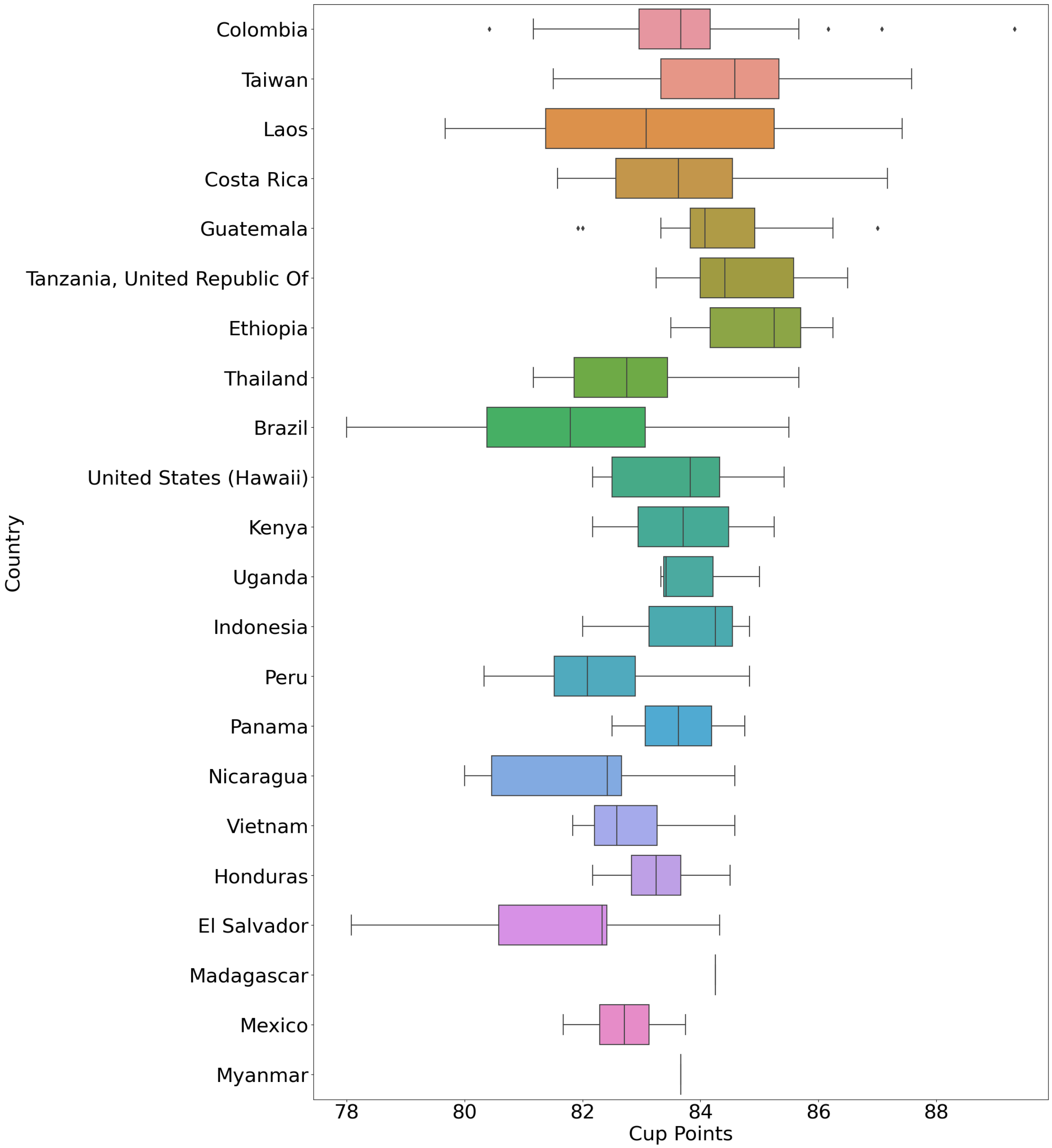
```
df_box = sns.boxplot(country_count, x = "Count", color = "mistyrose")
```



In [180…

```
plt.figure(figsize=(20, 30))
sns.boxplot(data = df, y = "Country of Origin", x = "Total Cup Points")
plt.xlabel("Cup Points", fontsize = 30)
plt.ylabel("Country", fontsize = 30)
plt.xticks(fontsize = 30)
plt.yticks(fontsize = 30)
```

```
Out[180…  (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                  17, 18, 19, 20, 21]),
           [Text(0, 0, 'Colombia'),
            Text(0, 1, 'Taiwan'),
            Text(0, 2, 'Laos'),
            Text(0, 3, 'Costa Rica'),
            Text(0, 4, 'Guatemala'),
            Text(0, 5, 'Tanzania, United Republic Of'),
            Text(0, 6, 'Ethiopia'),
            Text(0, 7, 'Thailand'),
            Text(0, 8, 'Brazil'),
            Text(0, 9, 'United States (Hawaii)'),
            Text(0, 10, 'Kenya'),
            Text(0, 11, 'Uganda'),
            Text(0, 12, 'Indonesia'),
            Text(0, 13, 'Peru'),
            Text(0, 14, 'Panama'),
            Text(0, 15, 'Nicaragua'),
            Text(0, 16, 'Vietnam'),
            Text(0, 17, 'Honduras'),
            Text(0, 18, 'El Salvador'),
            Text(0, 19, 'Madagascar'),
            Text(0, 20, 'Mexico'),
            Text(0, 21, 'Myanmar')])
```



In [181…

```
#df["Certification Address"].value_counts()
countries = ['Taiwan', 'Japan', 'Switzerland', 'Philippines', 'Jakarta',
'Guatemala', 'Madagascar', 'Kenya', 'Panama', 'Laos', 'Indonesia',
'Uganda', 'Vietnam', 'Peru', 'Mexico', 'United States', 'Tanzania',
'El Salvador', 'Nicaragua', 'Costa Rica', 'Brazil', 'Ethiopia',
'Thailand', 'Honduras', 'Colombia', 'Myanmar']

import re

def extract_country(address):
    if pd.isna(address):
        return "Unknown"

    address_str = str(address)
    normalized = address_str.lower()
    # 1. Direct match from known country list
    for c in countries:
        if c.lower() in normalized:
            return c

    # 2. Special fallback / region-based rules
    if "cortes" in normalized or "san pedro sula" in normalized:
        return "Honduras"
    if "rohrmoser" in normalized or "prisma dental" in normalized:
        return "Costa Rica"
    if "instituto de ecología" in normalized:
        return "Mexico"
    if "calle 60a" in normalized or "medellin" in normalized:
        return "Colombia"
    if "izusan" in normalized or "shizuoka" in normalized:
        return "Japan"
    if "atami" in normalized:
        return "Japan"
    if "commerce drive" in normalized:
        return "United States"
    if "del hotel seminole" in normalized:
        return "Nicaragua"
    if "calle pte" in normalized:
        return "El Salvador"

    # 3. REGEX but only for known uppercase country names
    uppercase_countries = ["USA", "JAPAN", "MEXICO"]
    for uc in uppercase_countries:
        if uc in address_str:
            return uc.title()

    return "Unknown"

df["Certification Country"] = df["Certification Address"].apply(extract_cour
df.head()
```

Out[181…

	Unnamed: 0	ID	Country of Origin	Farm Name	Lot Number	Mill	ICO Number	Company
0	0	0	Colombia	Finca El Paraiso	CQU2022015	Finca El Paraiso	NaN	Coffee Quality Union
1	1	1	Taiwan	Royal Bean Geisha Estate	The 2022 Pacific Rim Coffee Summit,T037	Royal Bean Geisha Estate	NaN	Taiwan Coffee Laboratory
2	2	2	Laos	OKLAO coffee farms	The 2022 Pacific Rim Coffee Summit,LA01	oklao coffee processing plant	NaN	Taiwan Coffee Laboratory
3	3	3	Costa Rica	La Cumbre	CQU2022017	La Montana Tarrazu MIll	NaN	Coffee Quality Union
4	4	4	Colombia	Finca Santuario	CQU2023002	Finca Santuario	NaN	Coffee Quality Union

In [182…

```
country_cert = df["Certification Country"].value_counts().reset_index()
country_cert.columns = ["Certification Country", "Certification Count"]
country_cert
```

Out[182…

	Certification Country	Certification Count
0	Taiwan	89
1	Japan	27
2	Guatemala	14
3	Honduras	10
4	Thailand	10
5	Kenya	8
6	Costa Rica	7
7	Ethiopia	6
8	El Salvador	6
9	Mexico	5
10	Switzerland	4
11	Brazil	4
12	Unknown	4
13	Colombia	4
14	Uganda	3
15	Nicaragua	3
16	Philippines	1
17	Jakarta	1
18	United States	1

In [183…

```
df[df["Certification Country"] == "Unknown"][["Certification Address", "Cert
```

Out[183…

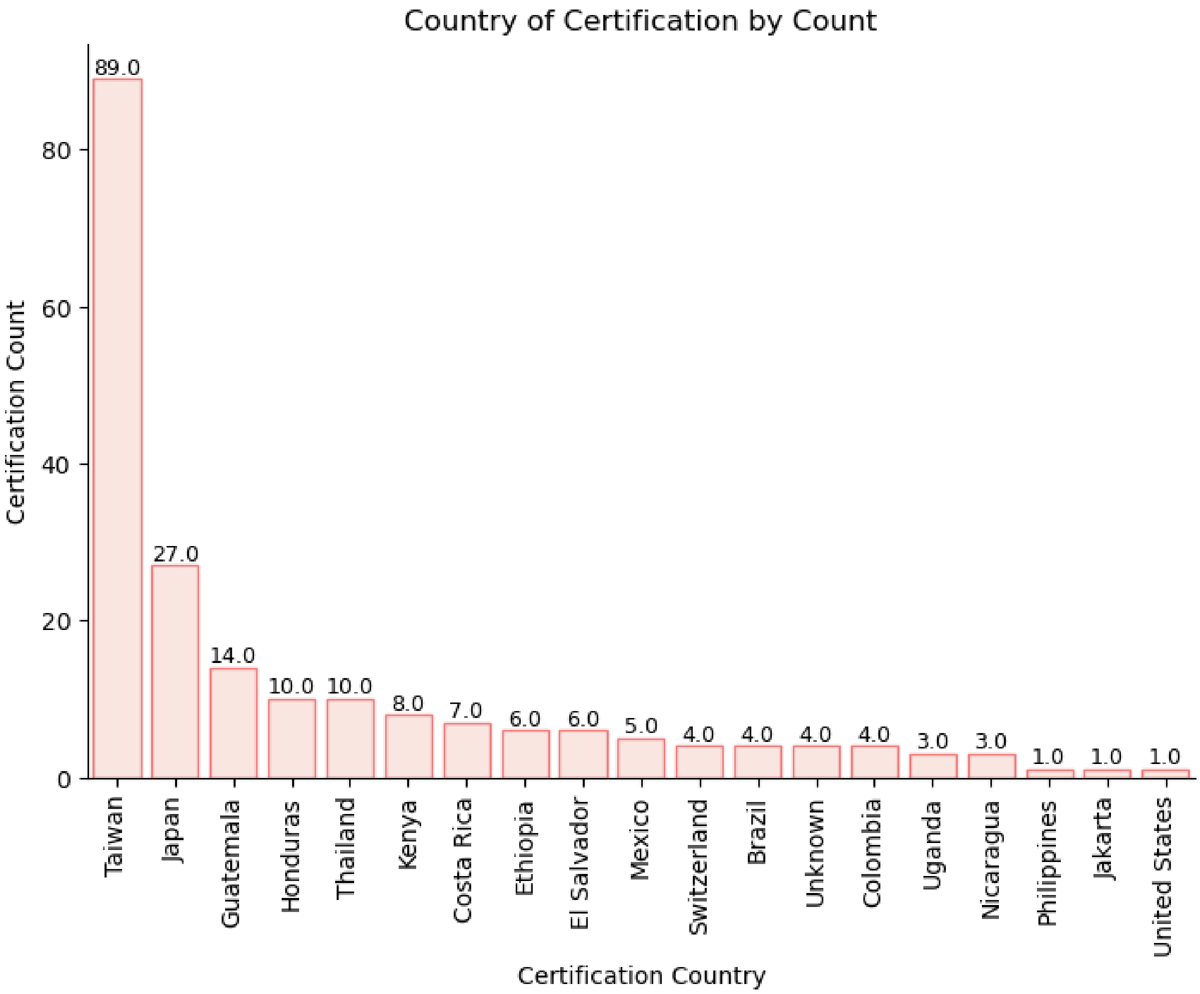
	Certification Address	Certification Country
64	*CURRENTLY NOT ACCEPTING SAMPLES**	Unknown
146	*CURRENTLY NOT ACCEPTING SAMPLES**	Unknown
155	*CURRENTLY NOT ACCEPTING SAMPLES**	Unknown
164	*CURRENTLY NOT ACCEPTING SAMPLES**	Unknown

In [184…

```
g = sns.catplot(data = country_cert, x = "Certification Country", y = "Certi
plt.xticks(rotation = 90)
plt.title("Country of Certification by Count")
ax = g.axes[0,0]
for p in ax.patches:
    value = p.get_height()
    if (math.isnan(value)):
        continue
    ax.text(
        p.get_x() + p.get_width()/2,
        p.get_height(),
        value,
        ha="center", va="bottom",
        fontsize=9
    )
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/se
aborn/axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```



In [185…

```
total_country_bags = df.groupby("Country of Origin")["Number of Bags"].sum()
total_country_bags.columns = ["Country of Origin", "Total Bags"]
bags_by_bin = (
    df.groupby(["Country of Origin", "CupPointsBin"])["Number of Bags"]
      .sum()
      .reset_index()
)
df_merge = bags_by_bin.merge(total_country_bags, on="Country of Origin")
df_merge["Proportion"] = df_merge["Number of Bags"] / df_merge["Total Bags"]
df_prop = df_merge.pivot_table(
    index="Country of Origin",
    columns="CupPointsBin",
    values="Proportion",
    fill_value=0
).reset_index()

df_prop.sort_values("High")
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[185], line 4
      1 total_country_bags = df.groupby("Country of Origin")["Number of Bag
s"].sum().reset_index()
      2 total_country_bags.columns = ["Country of Origin", "Total Bags"]
      3 bags_by_bin = (
----> 4     df.groupby(["Country of Origin", "CupPointsBin"])["Number of Bag
s"]
      5       .sum()
      6       .reset_index()
      7 )
      8 df_merge = bags_by_bin.merge(total_country_bags, on="Country of Orig
in")
      9 df_merge["Proportion"] = df_merge["Number of Bags"] / df_merge["Tota
l Bags"]

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packag
es/pandas/core/frame.py:8252, in DataFrame.groupby(self, by, axis, level, as
_index, sort, group_keys, observed, dropna)
   8249     raise TypeError("You have to supply one of 'by' and 'level'")
   8250 axis = self._get_axis_number(axis)
-> 8252 return DataFrameGroupBy(
   8253     obj=self,
   8254     keys=by,
   8255     axis=axis,
   8256     level=level,
   8257     as_index=as_index,
   8258     sort=sort,
   8259     group_keys=group_keys,
   8260     observed=observed,
   8261     dropna=dropna,
   8262 )

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packag
es/pandas/core/groupby/groupby.py:931, in GroupBy.__init__(self, obj, keys,
axis, level, grouper, exclusions, selection, as_index, sort, group_keys, obs
erved, dropna)
    928 self.dropna = dropna
    930 if grouper is None:
--> 931     grouper, exclusions, obj = get_grouper(
    932         obj,
    933         keys,
    934         axis=axis,
    935         level=level,
    936         sort=sort,
    937         observed=observed,
    938         dropna=self.dropna,
    939     )
    941 self.obj = obj
    942 self.axis = obj._get_axis_number(axis)

File /opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packag
es/pandas/core/groupby/grouper.py:985, in get_grouper(obj, key, axis, level,
sort, observed, validate, dropna)
    983         in_axis, level, gpr = False, gpr, None
    984     else:
--> 985         raise KeyError(gpr)
    986 elif isinstance(gpr, Grouper) and gpr.key is not None:
    987     # Add key to exclusions
    988     exclusions.add(gpr.key)

KeyError: 'CupPointsBin'
```

Final Alluvial

```
In [ ]: import plotly.express as px
        import plotly.offline as py
        py.init_notebook_mode()
        pd.DataFrame.iteritems = pd.DataFrame.items
```

```
In [ ]: def alluvialplot(df, dim_cols:list, color_col:str, title=None):

            # The color column must be a category
            if df[color_col].dtype != 'category':
                df[color_col] = df[color_col].astype('category')

            fig = px.parallel_categories(
                df,
                dimensions=dim_cols,
                color=df[color_col].cat.codes,
                height=1000,
                width=800,
                title=title
            )
            fig.update_traces(line={'shape': 'hspline'})
            fig.update_layout(coloraxis_showscale=False)
            fig.show()
```

```
In [ ]: df_master = df[["Country of Origin","Number of Bags", "Total Cup Points", "C
        df["Cup Points Category"] = pd.qcut(df["Total Cup Points"], q=3, labels=["Lo
        df_master
```

```
In [ ]: df_sorted = df_master.sort_values("Certification Country", ascending = True)
        alluvialplot(df_sorted, dim_cols = ['Country of Origin', 'Certification Cour
```