

# Analyzing 160,000 Wine Spectator Reviews

Justin Meisenhelter

Data Source: <https://www.kaggle.com/zynicide/wine-reviews>



## Objective:

- Explore Dataset, find anomalies
- Which regions produce the 'best' wine
- Cursory analysis of bias in reviewers
- Make wine pairing easier



# Data Dictionary

Columns:

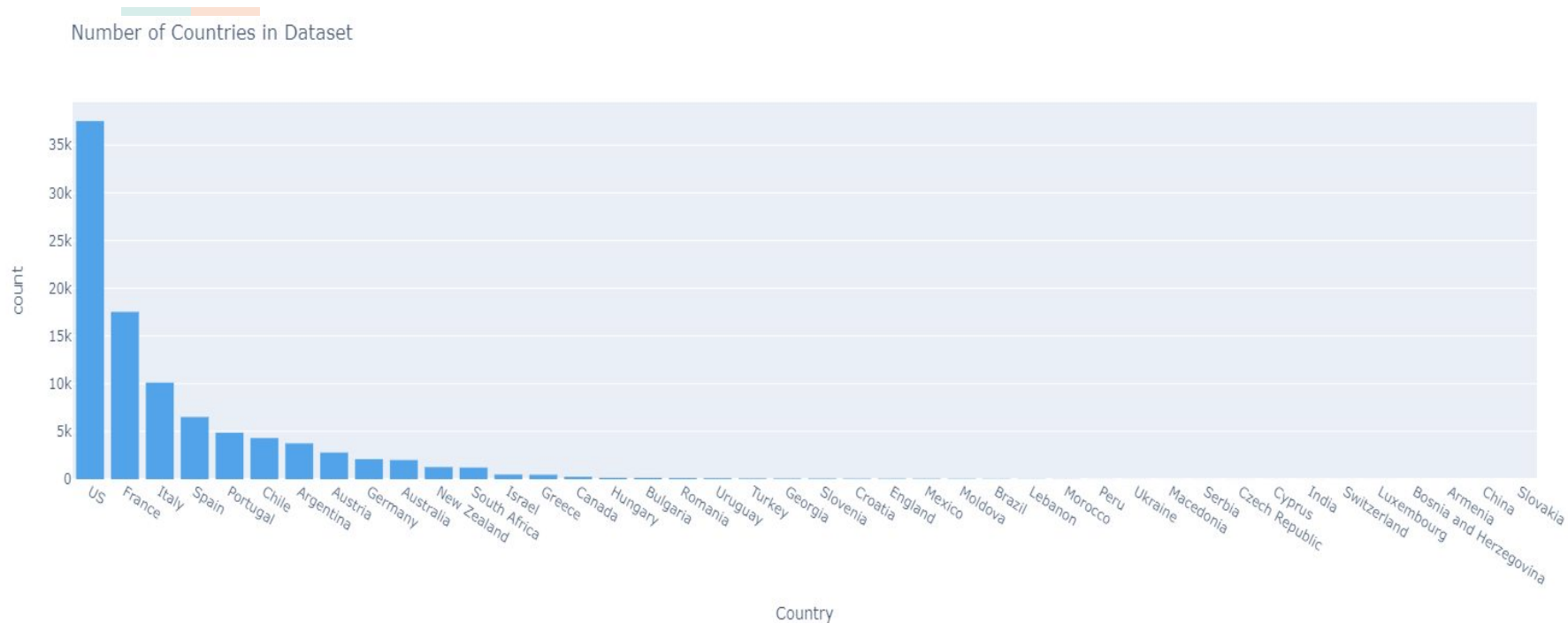
- **Country** : Country the wine is from
- **Description** : Full text review, includes tasting notes
- **Designation**: Name of the wine
- **Points**: Numerical review score from 50-100
- **Price**: Cost of bottle of wine (US Dollars)
- **Province** : The province or state the wine is from
- **Region\_1**: Name of wine growing region in **Province**
- **Taster\_Name**: Name of the person conducting the review

---

# Data Breakdown

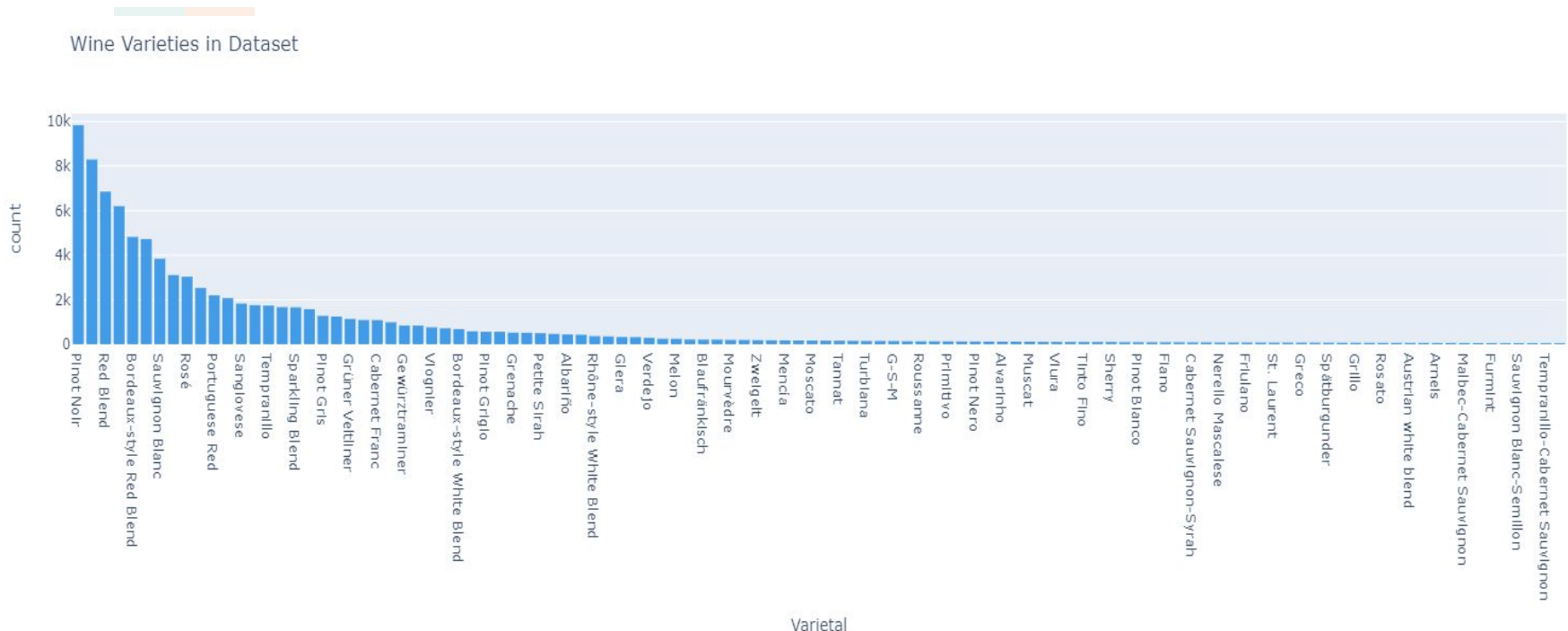
Understanding the Shape of The Data

# Represented Countries



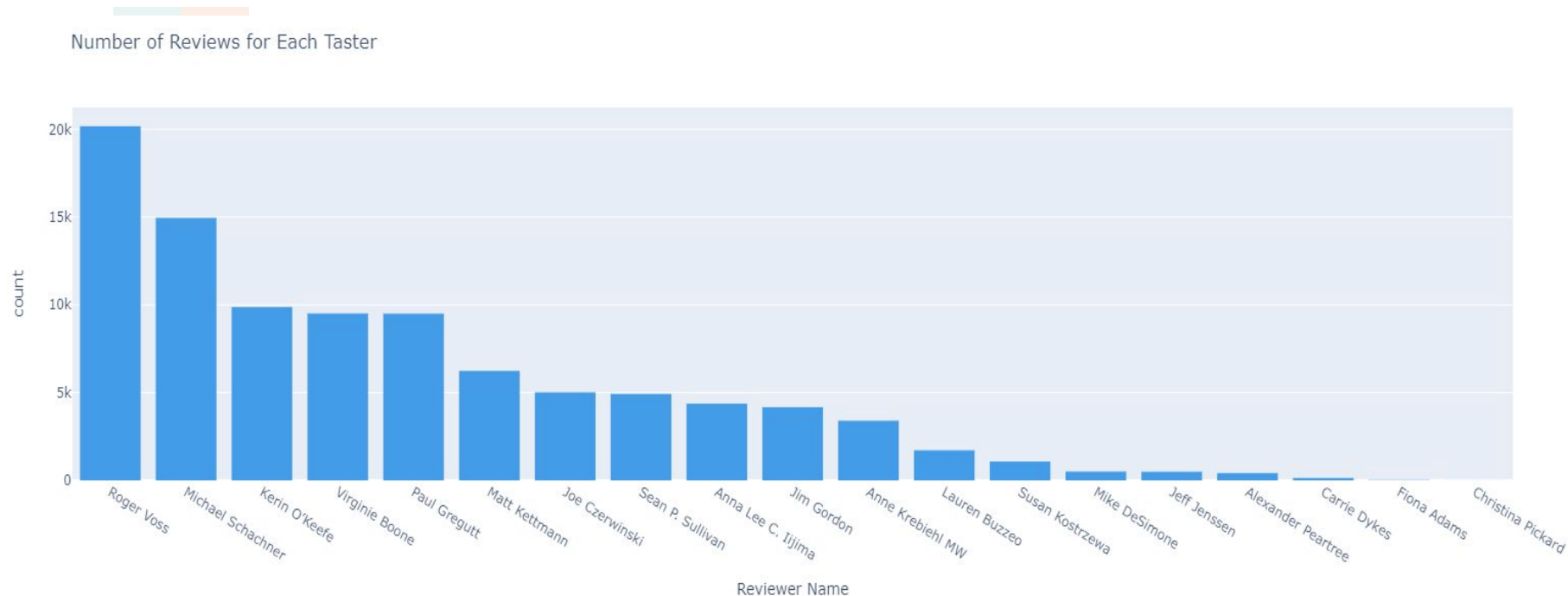
The vast majority of our data is concentrated in a small sample of countries

# Represented Wine Varietals



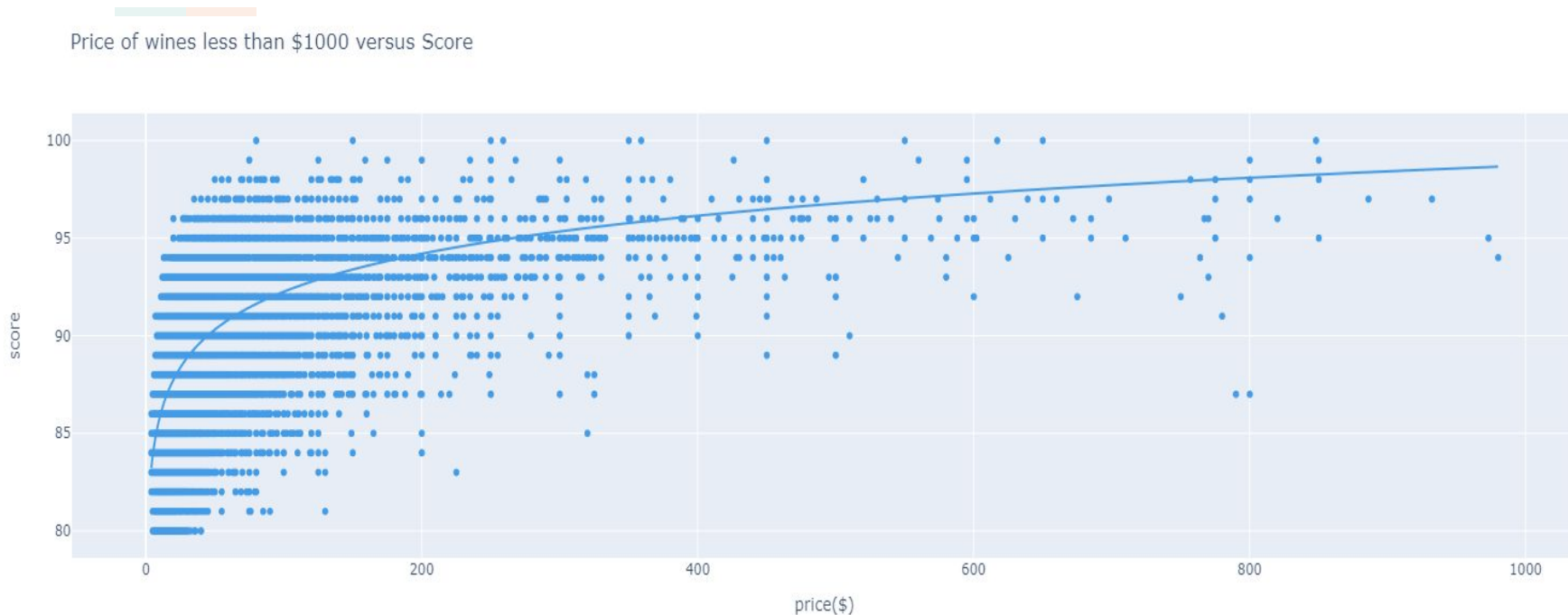
Most of the reviews concentrate around a few dozen varieties

# Number of Reviews by Each Taster



There are 19 total Tasters, and about a dozen prolific ones.

# Correlation of Price and Review Score

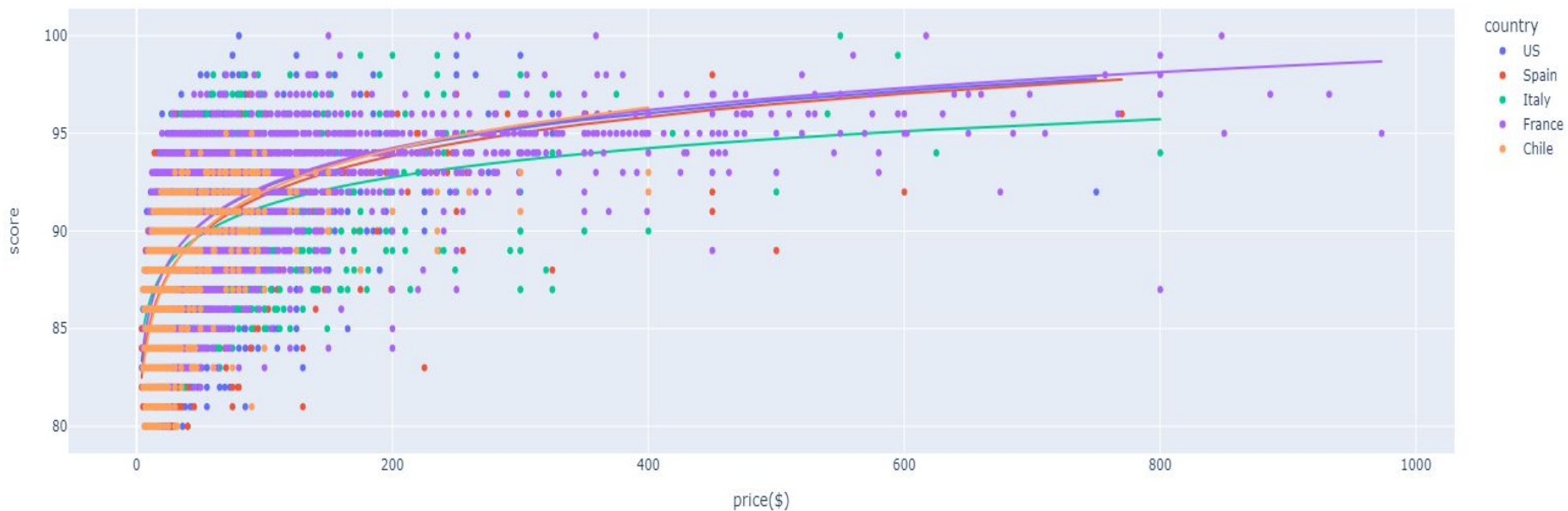


A logarithmic relationship seems to exist between these two observations



# Price Versus Review Score by Country

Price of wines less than \$1000 versus Score of the 5 most Reviewed Countries



---

# Categorizing Price and Review Score

# Defining 'Price' and 'Review Score' Buckets

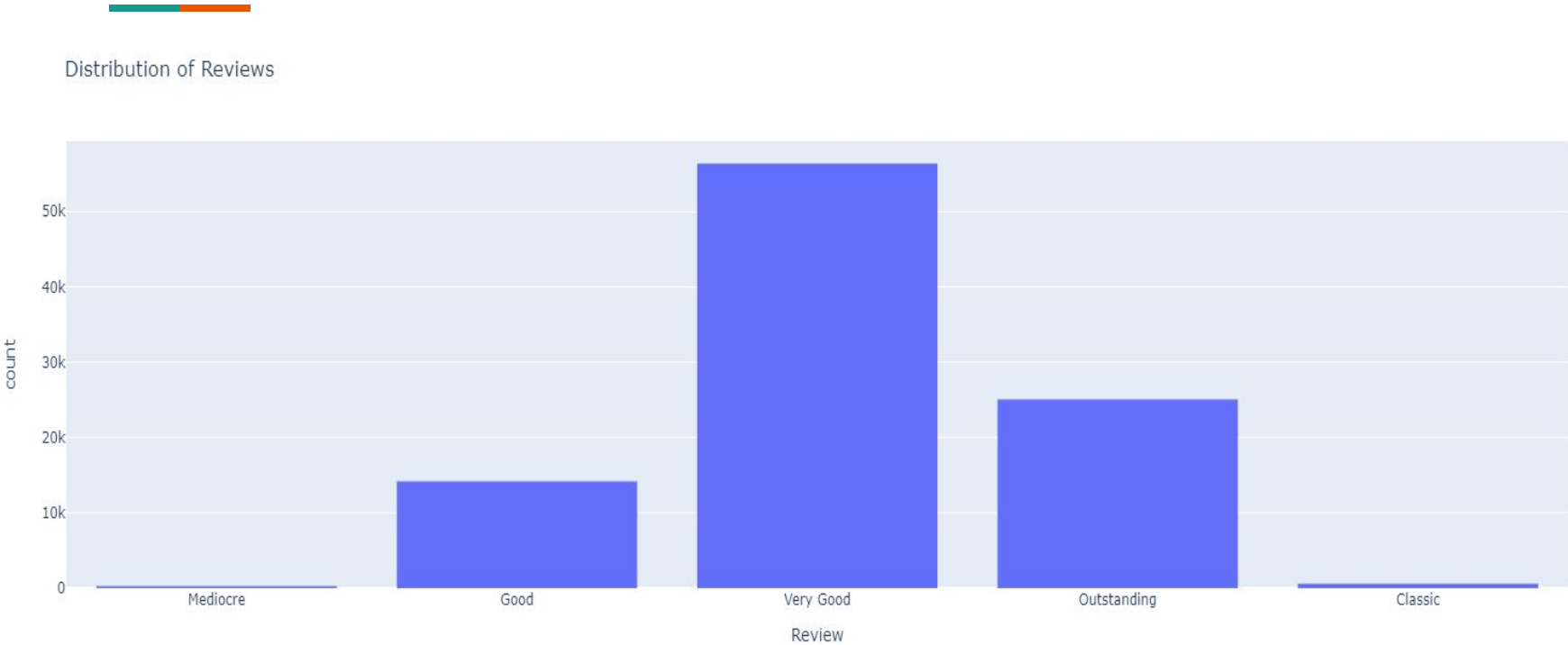


- Cheap: 4-17(\$)
- Inexpensive: 17-25(\$)
- Moderate: 26-42(\$)
- Pricey: 43-79(\$)
- Expensive: 80 - 175(\$)
- Outlandish: 175(\$) +
- 95-100 Classic: a great wine
- 90-94 Outstanding: a wine of superior character and style
- 85-89 Very good: a wine with special qualities
- 80-84 Good: a solid, well-made wine
- 75-79 Mediocre: a drinkable wine that may have minor flaws
- 50-74 Not recommended

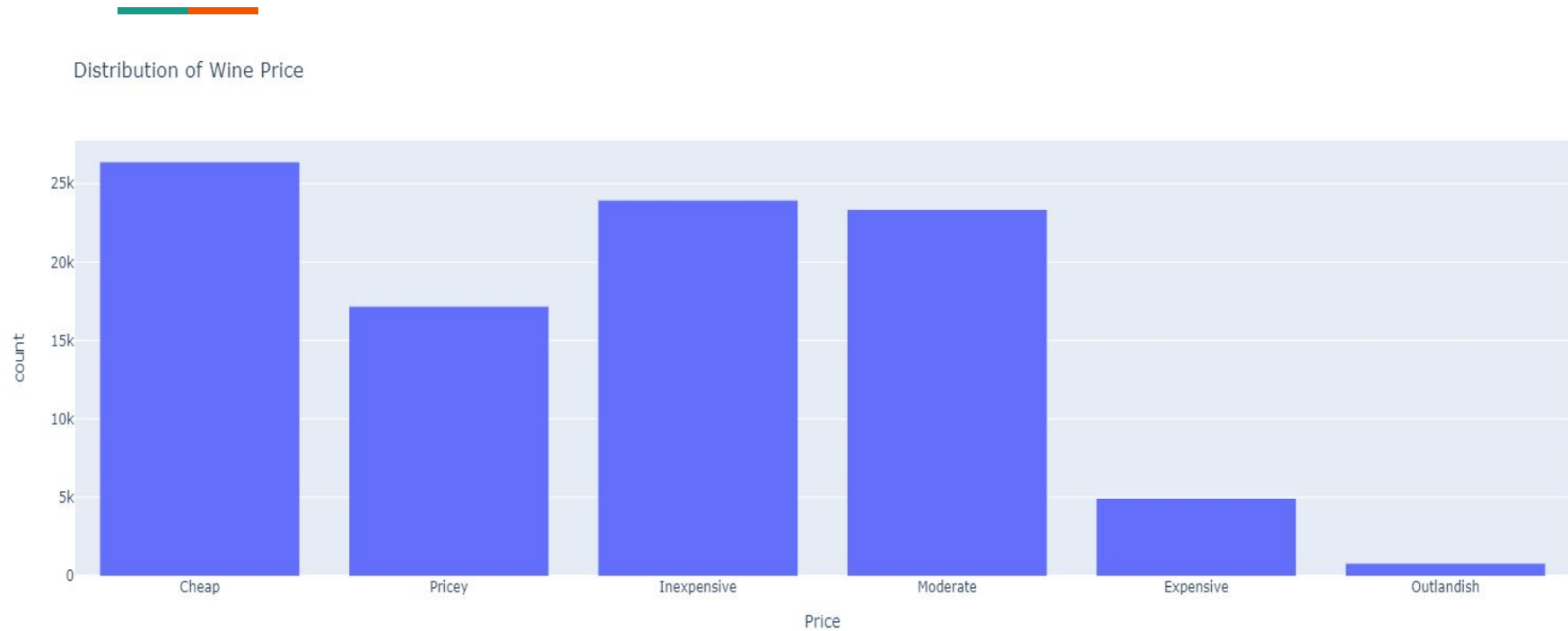
Price broken down by quantile.

Review scores breakdown from Wine Spectator

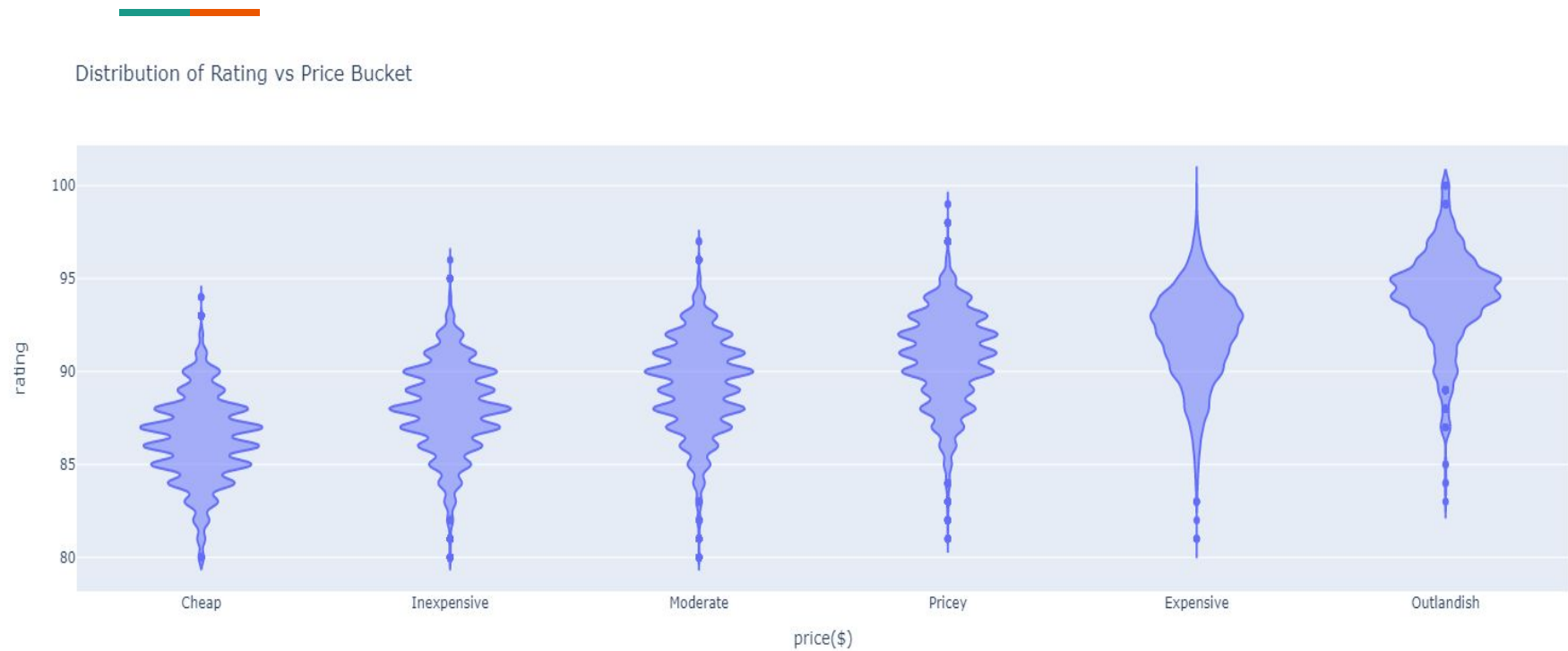
# Distribution of Reviews



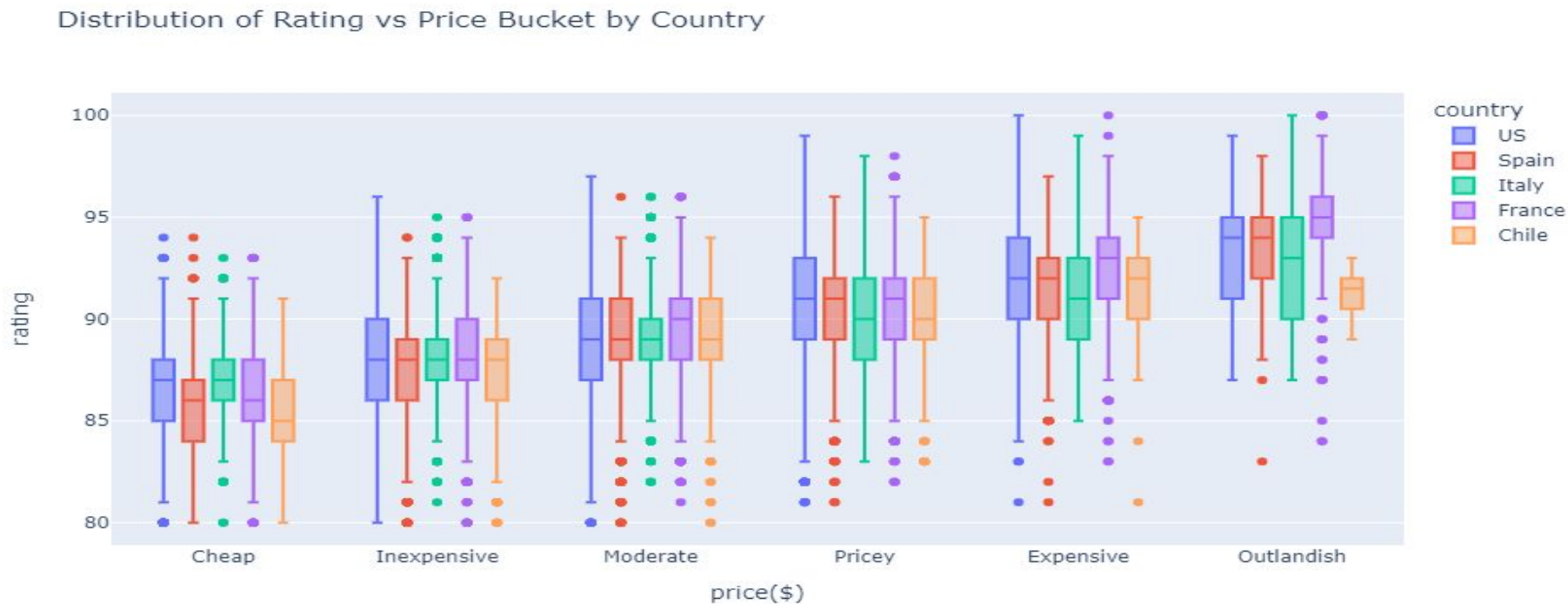
# Distribution of Price



# Distribution of Price Versus Review Score



# Review Score Distribution by Price Bucket and Country



---

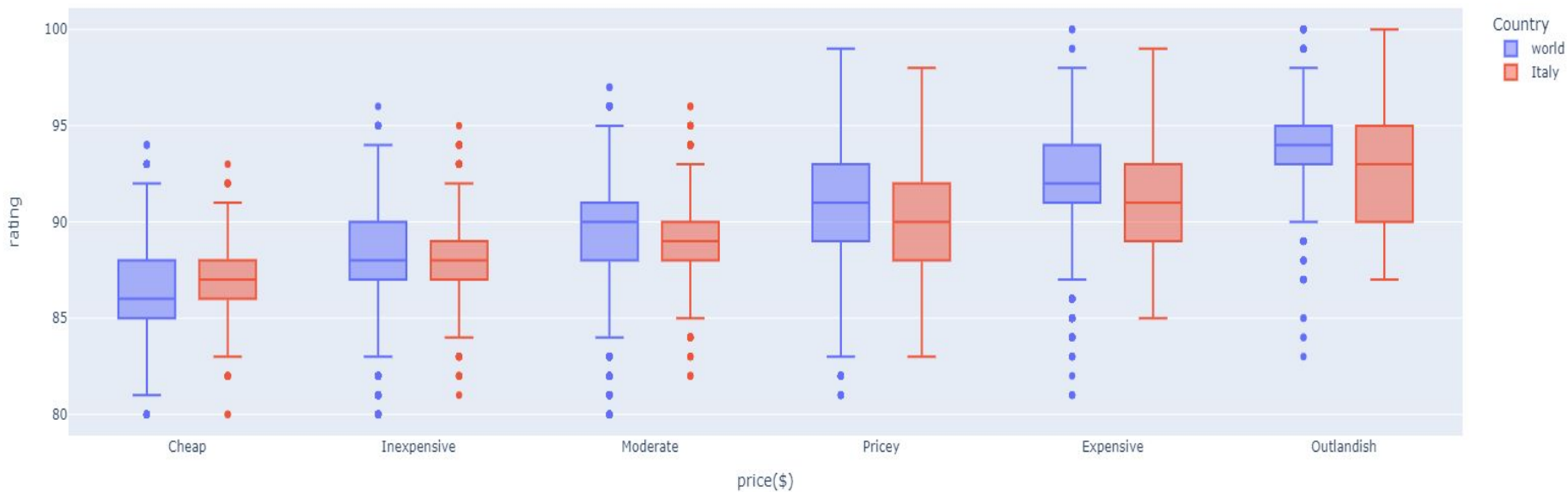
# Italy, Are you OK?

An Analysis of Italian Wines' Review Scores



# Italian Wine Reviews Compared with the Rest of the World

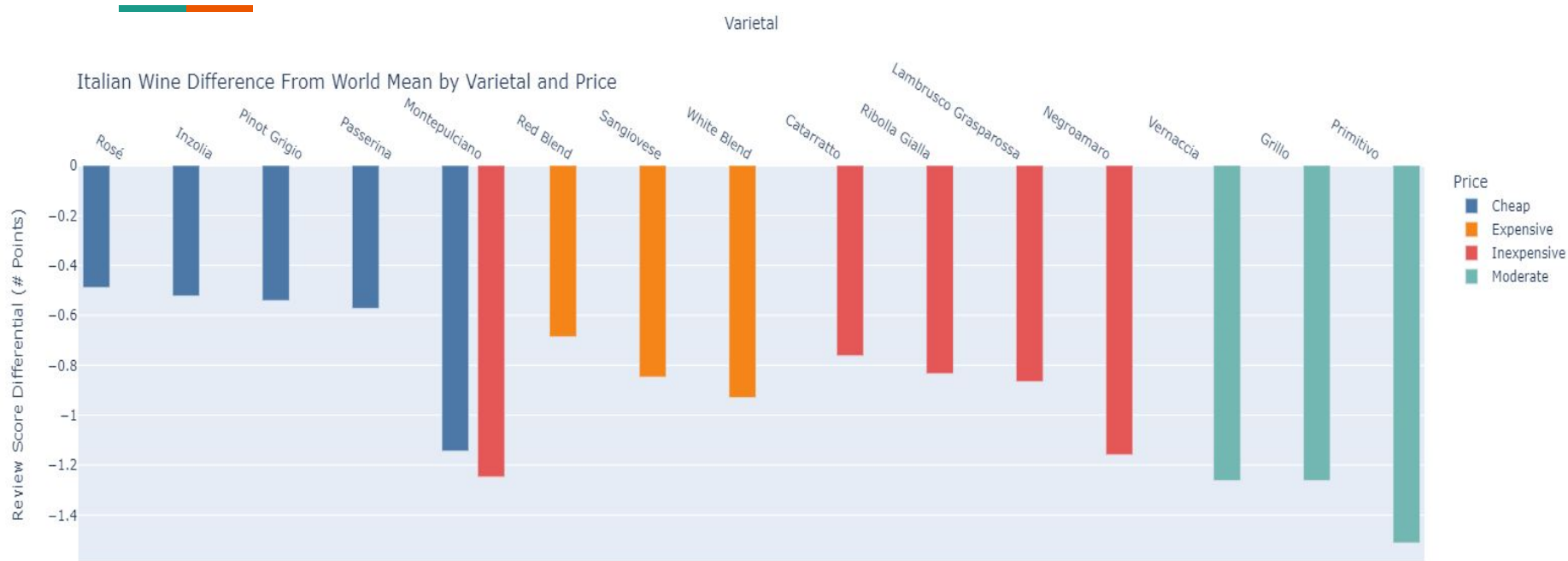
Comparison of Review scores of Italy versus the rest of the world



# Lowest Rated Italian Wine Regions



# Lowest Rated Italian Wine Varieties



**Conclusion: There are a handful of Italian regions and varietals pulling down their review scores.**

Regions:

- Central Italy
- Northeastern Italy
- Northwestern Italy
- Tuscany
- Veneto
- Other

Varietals:

- Montepulciano
- Primitivo
- Negroamaro
- Tuscany
- Grillo
- Vernaccia

---

# Finding Value in Wine Pairings

An Exploration into what regions produce good value

# A Few Definitions



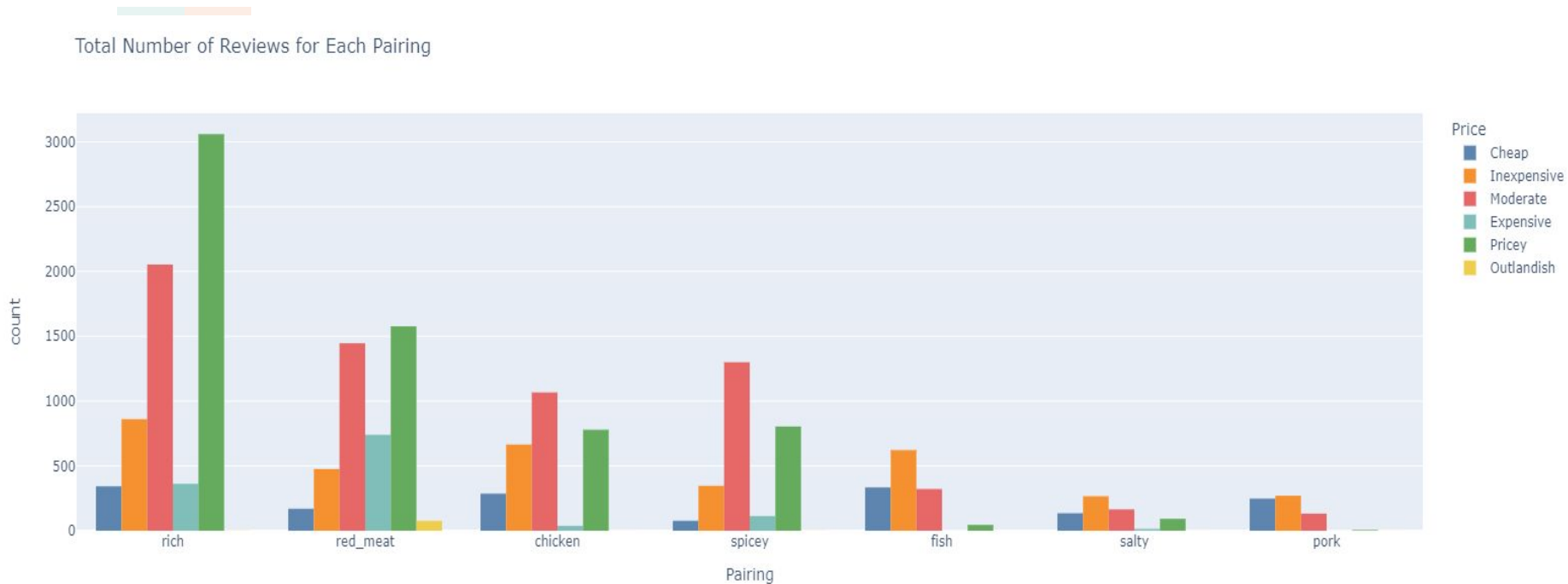
## Value:

- Value in wine will be defined as a wine's review score divided by its price.
- This gives us a figure of merit in review score per dollars spent
- A higher relative number will be a greater value wine

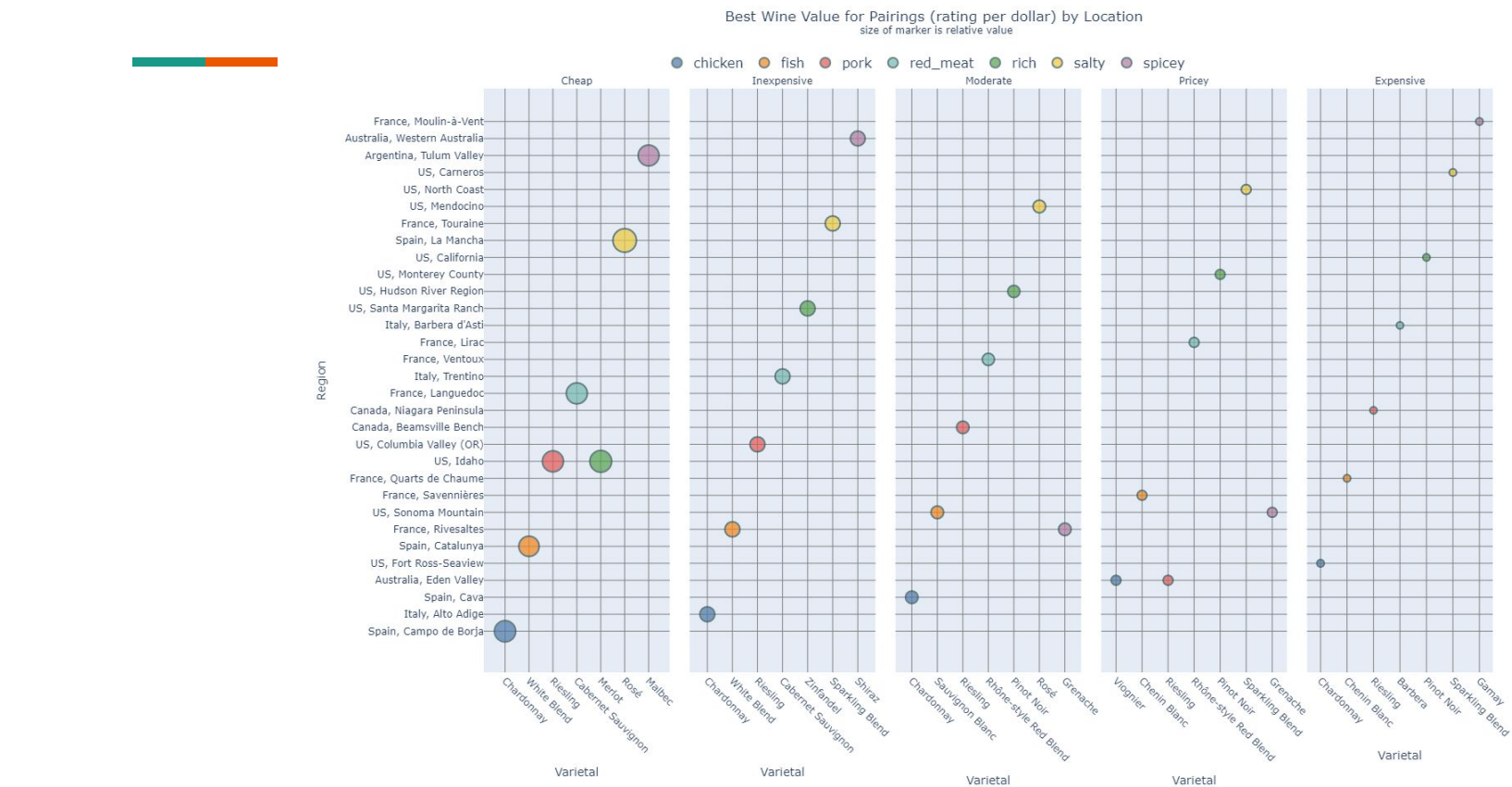
## Wine Pairings:

- Wine pairing is more art than science
- I will use my own experience in the field to define the 7 most common entree pairings
  - Chicken
  - Fish
  - Pork
  - Red Meat
  - Rich
  - Salty
  - Spicy
- See Appendix for a list of each wine appropriate for each pairing (and possibly an argument)

# Number of Reviews for each Pairing

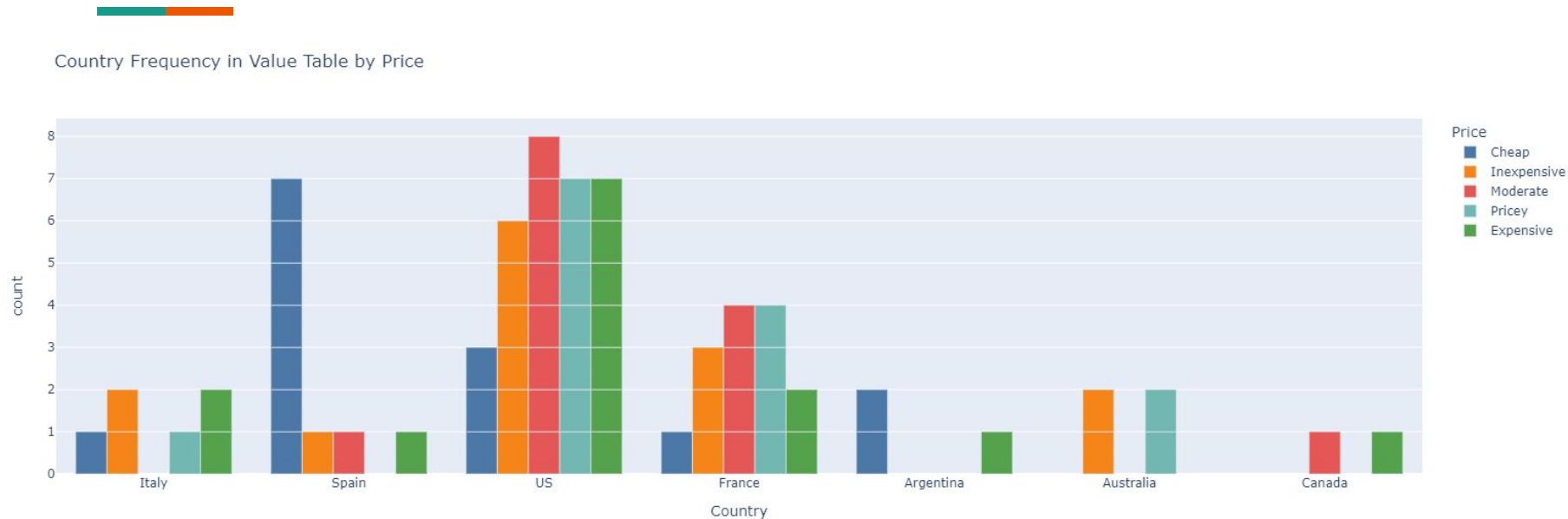


# Value Based Wine Selection





# Which Countries Produce the Highest Value Wine



The United States and France seem to produce the highest value wine

---

# Bias in Reviews

Are any of our reviewers biased against certain countries?

# Bias Criteria



- A Reviewer must have a statistically significant number of reviews in a given country
- Must have a large population size to compare reviews against
- Correct for wine review score and price. For example, If a reviewer only reviews 'Inexpensive' wine, the scores will not reflect the population mean

# How Wine Spectator Makes This Difficult



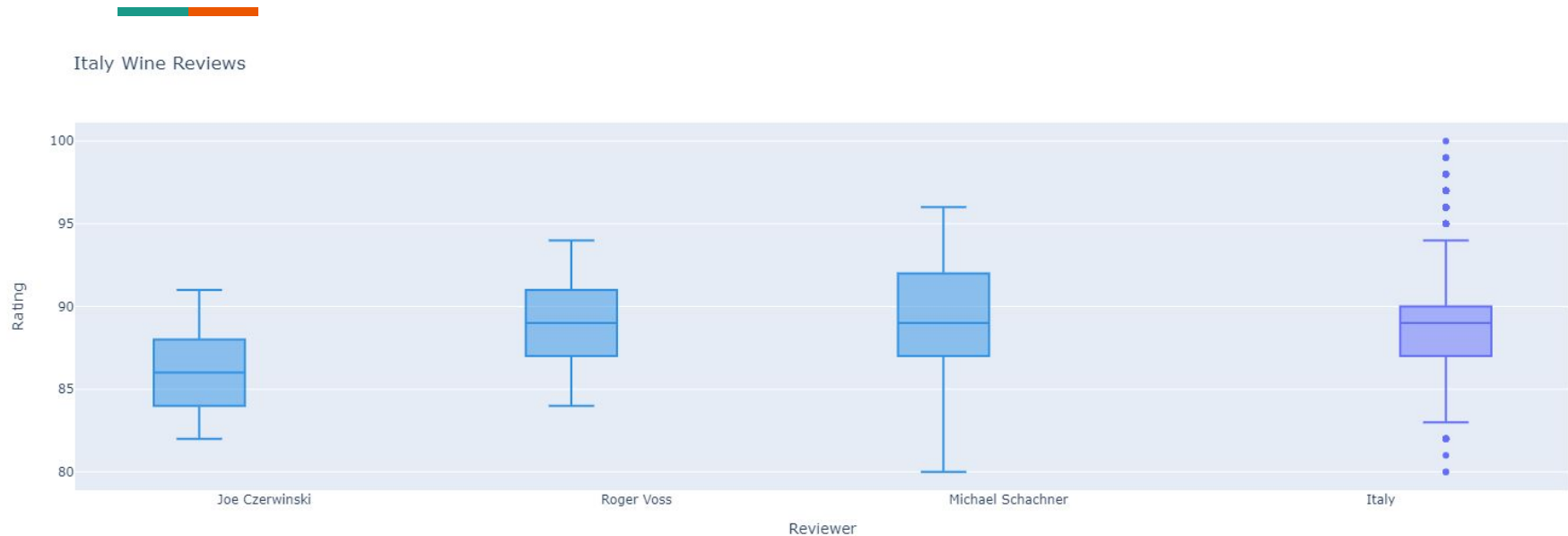
- Many countries have a primary reviewer who conducts more than 95% of a countries reviews
- There are only 10 Reviewers with a significant number of reviews to perform statistical analysis
- Most reviewers concentrate their reviews in narrow price ranges outside of their most reviewed country

# What can we do?



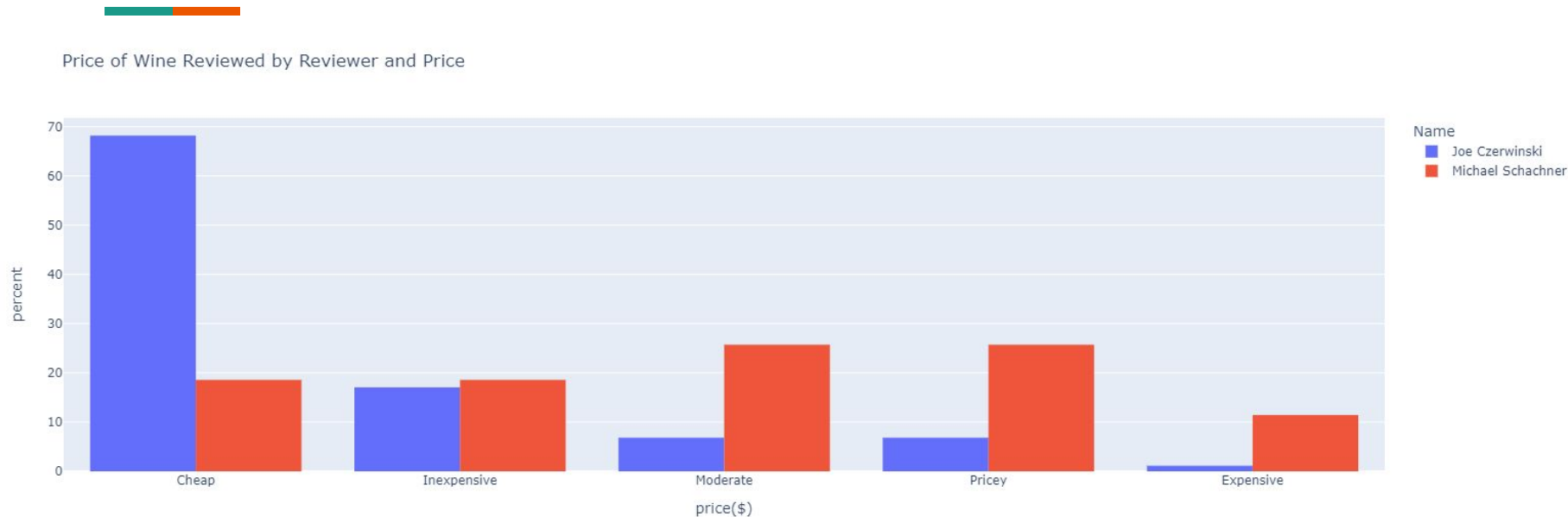
- There are two countries with enough reviewers to perform simple bias exploration (Italy and The US)
- We can look at the distribution of a single reviewer again both countries and attempt to correct for price of wine reviewed

# Italian Wine Reviews



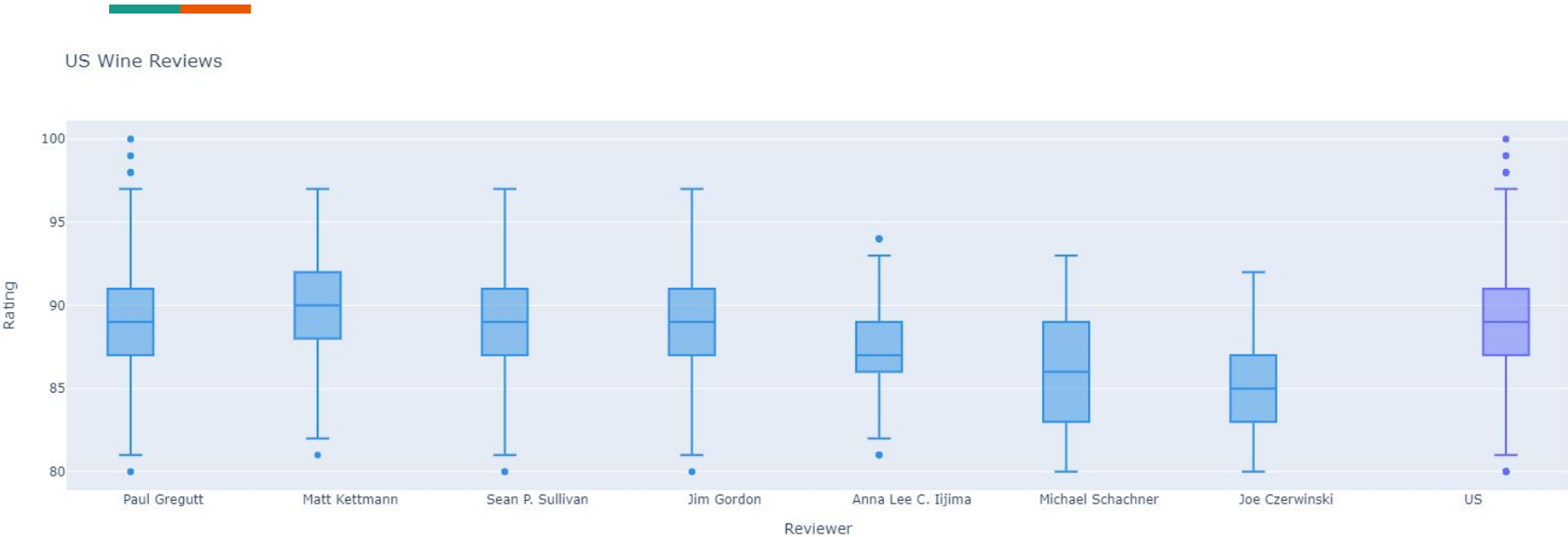
Joe Czerwinski seems to be biased against Italian wines, Roger Voss' reviews seem to line up with the general Italian wine reviews, while Michael Schachner seems to have a slight bias favoring Italian wines.

# Italian Wine Reviews by Price Bucket



Both tasters bias' can be explained by their lop-sided selection of cheap wine in Joe Czerwinski's case, and more expensive wines in Michael Schachner's case

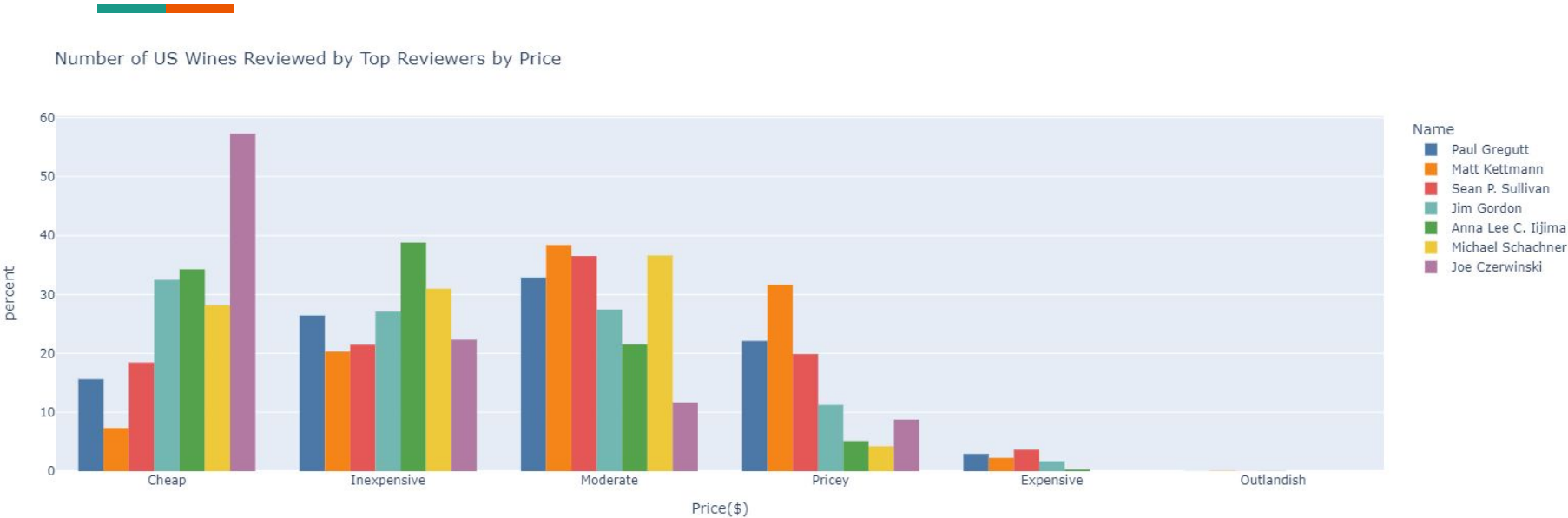
# US Wine Reviews



Anna Iijima, Michael Schachner, and Joe Czerwinski all seem to review US wine lower than the mean



# US Wine Reviews by Price Bucket



Wouldn't you know it, all three of the last slides reviewers reviewed less expensive wine

**All apparent biases can be  
explained by wine price  
selection**

—

# Wine Pairings List



- **Fish :** 'Sauvignon Blanc', 'Pinot Gris', 'White Blend', 'Chenin Blanc', 'Albariño', 'Pinot Blanc'
- **Red Meat:** 'Cabernet Sauvignon', 'Bordeaux-style Red Blend', 'Nebbiolo', 'Rhône-style Red Blend', 'Cabernet Franc', 'Barbera', 'Verdejo', 'Petit Verdot'
- **Salty:** 'Rosé', 'Sparkling Blend', 'Champagne Blend', 'Glera'
- **Spicey :** 'Syrah', 'Malbec', 'Tempranillo', 'Gamay', 'Shiraz', 'Tempranillo Blend', 'Grenache', 'Petite Sirah', 'Garnacha'
- **Rich:** 'Pinot Noir', 'Merlot', 'Sangiovese', 'Zinfandel', 'Carmenère', 'Torrontés'
- **Pork:** 'Riesling', 'Grüner Veltliner', 'Gewürztraminer', 'Blafränkisch'
- **Chicken:** 'Chardonnay', 'Portuguese White', 'Viognier', 'Bordeaux-style White Blend', 'Rhône-style White Blend'

## Code for Represented Countries Graph

```
fig = px.histogram(x=reviewsClean['country'],
                  labels = {
                      'x' : 'Country'
                  },
                  color_discrete_sequence=px.colors.qualitative.Dark24,
                  title = 'Number of Countries in Dataset'
                )
fig.update_xaxes(categoryorder = 'total descending')
fig
```

## Code for Represented Varietals Graph

```
a = reviewsClean[reviewsClean['variety'].map(reviewsClean['variety'].value_counts()) > 50]
fig = px.histogram(a, x='variety',
    labels = {
        'variety' : 'Varietal'
    },
    color_discrete_sequence=px.colors.qualitative.Dark24,
    title = 'Wine Varieties in Dataset'
)
fig.update_xaxes(categoryorder = 'total descending')
fig
```


## Code for Tasters Graph

```
fig = px.histogram(reviewsClean, x='taster_name',  
    labels = {  
        'taster_name': 'Reviewer Name'  
    },  
    color_discrete_sequence=px.colors.qualitative.Dark24,  
    title = 'Number of Reviews for Each Taster'  
)  
fig.update_xaxes(categoryorder = 'total descending')  
fig
```

## Code for Price V Score Graph

```
cheaperWinesDF = reviewsClean.loc[reviewsClean['price'] < 1000]
fig = px.scatter(x=cheaperWinesDF['price'], y=cheaperWinesDF['points'],
                 trendline="ols",
                 trendline_options = dict(log_x = True),
                 labels = {
                     'x' : 'price($)',
                     'y' : 'score'
                 },
                 color_discrete_sequence=px.colors.qualitative.Dark24,
                 title = 'Price of wines less than $1000 versus Score'
                 )
fig
```

## Code of bucketing price and points



```
# Create Buckets
cut_labels = ['Cheap', 'Inexpensive', 'Moderate', 'Pricey', 'Expensive', 'Outlandish']
cut_bins = [4, 17, 25, 42, 79, 175, 3300]
# start with fresh DF
reviewsCleanBuckets = reviewsClean
# create series for new row
price_buckets = pd.cut(reviewsCleanBuckets['price'], bins=cut_bins,
labels=cut_labels)
# concat to avoid assigning into a copy of a DF
reviewsCleanBuckets = pd.concat([reviewsCleanBuckets, price_buckets], axis = 1)
# Rename columns
reviewsCleanBuckets.columns = ['Unnamed: 0', 'country', 'description', 'designation',
'points',
'price', 'province', 'region_1', 'region_2', 'taster_name',
'taster_twitter_handle', 'title', 'variety', 'winery', 'price_bucket']
#drop unneeded column
reviewsCleanBuckets.drop('Unnamed: 0', axis = 1)
```

```
cut_labels = ['Not Recommended', 'Mediocre', 'Good', 'Very Good', 'Outstanding',
'Classic']
cut_bins = [50, 75, 80, 85, 90, 95, 100]
# start with fresh DF
b = reviewsCleanBuckets
# create series for new row
point_buckets = pd.cut(b['points'], bins=cut_bins, labels=cut_labels)
# concat to avoid assigning into a copy of a DF
b = pd.concat([b, point_buckets], axis = 1)
# Rename columns
b.columns = ['Unnamed: 0', 'country', 'description', 'designation', 'points',
'price', 'province', 'region_1', 'region_2', 'taster_name',
'taster_twitter_handle', 'title', 'variety', 'winery', 'price_bucket', 'review_bucket']
```



## Code of Review Distribution Graph



```
fig = px.histogram(b, x="review_bucket",
    category_orders = {
        'review_bucket': ['Not Recommended', 'Mediocre', 'Good', 'Very Good',
        'Outstanding', 'Classic']
    },
    labels = {
        'review_bucket': 'Review'
    },
    title = 'Distribution of Reviews'
)
fig.show()
```

## Code of Price Distribution Graph



```
fig = px.histogram(reviewsCleanBuckets, x="price_bucket",
                   category_orders = {
                       'price_bucket_bucket': ['Cheap', 'Inexpensive', 'Moderate', 'Pricey',
                                                'Expensive', 'Outlandish']
                   },
                   labels = {
                       'price_bucket': 'Price'
                   },
                   title = 'Distribution of Wine Price'
                   )
fig.show()
```

## Code of Price versus review Distribution Graph



```
fig = px.violin(reviewsCleanBuckets, x="price_bucket", y="points",
               category_orders = {
                   'price_bucket': ['Cheap', 'Inexpensive', 'Moderate', 'Pricey', 'Expensive',
                                   'Outlandish']
               },
               labels = {
                   'price_bucket': 'price($)',
                   'points': 'rating'
               },
               title = 'Distribution of Rating vs Price Bucket'
            )
fig.show()
```

## Code of Price Versus Review Score by Country Graph




```
popCountryFilter = ['US', 'France', 'Italy', 'Spain', 'Chile']
popCountryDF =
cheaperWinesDF.loc[cheaperWinesDF['country'].isin(popCountryFilter)]
fig = px.scatter(popCountryDF, x='price', y='points', color = "country",
                 trendline="ols",
                 trendline_options = dict(log_x = True),
                 labels = {
                     'price': 'price($)',
                     'points': 'score'
                 },
                 title = 'Price of wines less than $1000 versus Score of the 5 most
Reviewed Countries'
)
```

## Code of Italy Vs World Review Graph




```
fig = px.box(italyDF, x="price_bucket", y="points", color = 'is_italy',
             category_orders = {
                 'price_bucket': ['Cheap', 'Inexpensive', 'Moderate', 'Pricey', 'Expensive',
                                   'Outlandish']
             },
             labels = {
                 'price_bucket': 'price($)',
                 'points': 'rating'
             },
             title = 'Comparison of Review scores of Italy versus the rest of the world'
             )
fig.update_layout(legend_title = 'Country')
fig.show()
```

## Code of Italy Region Reviews Graph



```
#Create working DF
onlyItalyDF = italyDF.loc[italyDF['country'] == 'Italy']
# group by price and province, aggregate points
dfh = onlyItalyDF.groupby(['price_bucket', 'province']).agg(
    average_points = ('points', 'mean')
).dropna()
#collapse index, get the top 3 for each group
h = dfh.groupby(level=0, group_keys=False).apply(
    lambda x: x.sort_values('average_points').head(3)).reset_index()
h = h.merge(italyDF.groupby('price_bucket')['points'].agg('mean'), how = 'inner', on = 'price_bucket')
# create difference column (world average minus italy average)
h['difference'] = h['average_points'] - h['points']
h.sort_values('difference', ascending = False).head(10)
fig = px.bar(h.loc[h['price_bucket']!= 'Outlandish'].sort_values('difference').head(10), x = 'province', y = 'difference', color = 'price_bucket',
    labels = {
        'difference': 'Review Score Differential (# Points)',
        'province': 'Region'
    },
    barmode = 'group',
    color_discrete_sequence=px.colors.qualitative.T10,
    title = 'Italian Wine Review Score Difference Form World Mean by Region and Price')
fig.update_layout(legend_title = 'Price', xaxis = {'side': 'top'})
fig.update_traces(width=.2)
fig
```

## Code of Italy Varietal Reviews Graph



```
#group by varietal and price
dfg = onlyItalyDF.sort_values('points').groupby(['price_bucket', 'variety']).agg(
    average_points = ('points', 'mean'),
    count = ('variety', 'count')
).dropna()
# take the 5 worst rated varietals for each price bucket
g = dfg.loc[dfg['count'] > 5].groupby(level=0, group_keys=False).apply(
    lambda x: x.sort_values(('average_points'),).head(5)).reset_index()
#compare to world average by price bucket
g = g.merge(italyDF.groupby('price_bucket')['points'].agg('mean'), how = 'inner', on = 'price_bucket')
g['difference'] = g['average_points'] - g['points']
g.sort_values('difference').head(10).sort_values('price_bucket')
fig = px.bar(g.loc[g['price_bucket'] != 'Outlandish'], sort_values('difference', ascending = False).head(16),
    x = 'variety', y = 'difference', color = 'price_bucket',
    labels = {
        'difference': 'Review Score Differential (# Points)',
        'variety': 'Varietal'
    },
    barmode='group',
    color_discrete_sequence=px.colors.qualitative.T10,
    title = 'Italian Wine Difference From World Mean by Varietal and Price')
fig.update_layout(legend_title = 'Price', xaxis = {'side': 'top'})
fig.update_traces(width=.3)
fig
```

## Code for pairings by price bucket Graph

```
pairingsDict = {'fish': ['Sauvignon Blanc', 'Pinot Gris', 'White Blend', 'Chenin Blanc', 'Albariño', 'Pinot Blanc'],
'red_meat': ['Cabernet Sauvignon', 'Bordeaux-style Red Blend', 'Nebbiolo', 'Rhône-style Red Blend', 'Cabernet Franc', 'Barbera', 'Verdejo', 'Petit Verdot'],
'salty': ['Rosé', 'Sparkling Blend', 'Champagne Blend', 'Glera'],
'spicy': ['Syrah', 'Malbec', 'Tempranillo', 'Gamay', 'Shiraz', 'Tempranillo Blend', 'Grenache', 'Petite Sirah', 'Garnacha'],
'rich': ['Pinot Noir', 'Merlot', 'Sangiovese', 'Zinfandel', 'Carmenère', 'Torrontés'],
'pork': ['Riesling', 'Grüner Veltliner', 'Gewürztraminer', 'Blafränkisch'],
'chicken': ['Chardonnay', 'Portuguese White', 'Viognier', 'Bordeaux-style White Blend', 'Rhône-style White Blend']}

pairings = []
for i in topVarDF.variety:
    for j in pairingsDict:
        if i in pairingsDict[j]:
            pairings.append(j)
pairings = pd.Series(pairings)
topVarDF = pd.concat([topVarDF, pairings], axis = 1)
topVarDF.columns = ['country', 'description', 'designation', 'points', 'price', 'province', 'region_1', 'region_2',
'taster_name', 'taster_twitter_handle', 'title', 'variety', 'winery', 'price_bucket', 'pairing']
fig = px.histogram(topVarDF.dropna(), x = 'pairing', color = 'price_bucket',
color_discrete_sequence=px.colors.qualitative.T10,
labels = {
'pairing': 'Pairing',
'price_bucket': 'Price'
}, title = 'Total Number of Reviews for Each Pairing',
barmode = 'group'
)
fig.update_xaxes(categoryorder = 'total descending')
fig
```



## Code for pairings by price bucket Graph



```
valueDF['location'] = valueDF['country'] + ', ' + valueDF['region_1']
fig = px.scatter(valueDF, y = 'location', x = 'variety', color = 'pairing', facet_col = 'price_bucket',
                size = 'value',
                labels = {
                    'location': 'Region',
                    'pairing': '',
                    'price_bucket': 'Price',
                    'variety': 'Varietal',
                },
                title = 'Best Wine Value for Pairings (rating per dollar) by Location <br><sup>size of marker is relative value</sup>',
                color_discrete_sequence=px.colors.qualitative.T10,
                height = 900,
                width = 1500)
fig.update_xaxes(matches=None)
fig.update_layout(title_x=0.5, title_y=.96,
                  legend=dict(
                      orientation="h",
                      yanchor="bottom",
                      y=1.02,
                      xanchor="right",
                      x=.7,
                      font=dict(
                          size=16
                      )
                  ))
fig.update_traces(marker=dict(line=dict(width=2,
                                         color='DarkSlateGrey')),
                  selector=dict(mode='markers'))
fig.for_each_annotation(lambda a: a.update(text=a.text.split("=")[-1]))
fig.update_xaxes(showgrid=True, gridwidth=1, gridcolor='gray')
fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='gray')
fig
```

## Code for countries with highest value Graph



```
fig = px.bar(valueDF, x = 'country', color = 'price_bucket',
             labels = {
                 'price_bucket': 'Price',
                 'country': 'Country'
             },
             barmode = 'group',
             color_discrete_sequence=px.colors.qualitative.T10,
             title = 'Country Frequency in Value Table by Price'
             )
fig
```

## Code for Italian wine reviewers Graph



```
graphDF1 = topReviewersDF.loc[(topReviewersDF['taster_name'].isin(['Joe Czerwinski', 'Michael Schachner', 'Roger Voss']))&
    (topReviewersDF['country'] == 'Italy')]
graphDF2 = reviewsClean.loc[reviewsClean['country'] == 'Italy']

fig = px.box(graphDF1, x = 'taster_name', y="points",
    labels = {
        'points': 'Rating',
        'taster_name': 'Reviewer'
    },
    color_discrete_sequence=px.colors.qualitative.Dark24,
    title = 'Italy Wine Reviews'
)
fig2 = px.box(graphDF2, x='country', y='points',
    labels = {
        'country': 'Total Reviws'
    }
)
fig.add_trace(fig2.data[0])

fig.show()
```

## Code for Italian wine reviewers by Price Graph



```
graphDF1 = popReviewsCleanBuckets.loc[(popReviewsCleanBuckets['taster_name'].isin(
    ['Joe Czerwinski', 'Michael Schachner'])) & (popReviewsCleanBuckets['country'] == 'Italy')]
fig = px.histogram(graphDF1, x = 'price_bucket', color = 'taster_name',
    category_orders = {
        'price_bucket': ['Cheap', 'Inexpensive', 'Moderate', 'Pricey', 'Expensive', 'Outlandish']
    },
    labels = {
        'price_bucket': 'price($)',
        'taster_name': 'Name'
    },
    barmode = 'group',
    histnorm = 'percent',
    title = 'Price of Wine Reviewed by Reviewer and Price'
)
fig.show()
```

## Code for US wine reviewers Graph



```
USReviewers = ['Anna Lee C. Iijima', 'Jim Gordon',
               'Joe Czerwinski', 'Matt Kettmann', 'Michael Schachner', 'Paul Gregutt', 'Sean P. Sullivan']
graphDF1 = popReviewsCleanBuckets.loc[(popReviewsCleanBuckets['taster_name'].isin(USReviewers))&
                                       (popReviewsCleanBuckets['country'] == 'US')]
graphDF2 = reviewsClean.loc[reviewsClean['country'] == 'US']
fig = px.box(graphDF1, x = 'taster_name', y="points",

              labels = {
                  'points': 'Rating',
                  'taster_name': 'Reviewer'
              },
              color_discrete_sequence=px.colors.qualitative.Dark24,
              title = 'US Wine Reviews'
            )
fig2 = px.box(graphDF2, x='country', y='points',
              labels = {
                  'country': 'Total Reviws'
              }
            )
fig.add_trace(fig2.data[0])

fig.show()
```

## Code for US wine reviewers by price Graph



```
USReviewers = ['Anna Lee C. Iijima', 'Jim Gordon',
               'Joe Czerwinski', 'Matt Kettmann', 'Michael Schachner', 'Paul Gregutt', 'Sean P. Sullivan']
graphDF1 = popReviewsCleanBuckets.loc[(popReviewsCleanBuckets['taster_name'].isin(USReviewers))&
                                       (popReviewsCleanBuckets['country'] == 'US')]
graphDF2 = reviewsClean.loc[reviewsClean['country'] == 'US']
fig = px.box(graphDF1, x = 'taster_name', y="points",

              labels = {
                  'points': 'Rating',
                  'taster_name': 'Reviewer'
              },
              color_discrete_sequence=px.colors.qualitative.Dark24,
              title = 'US Wine Reviews'
            )
fig2 = px.box(graphDF2, x='country', y='points',
              labels = {
                  'country': 'Total Reviws'
              }
            )
fig.add_trace(fig2.data[0])

fig.show()
```