Name of activity (Assignment: Visualizations): 11.3 Exercise Visualization
Your name: Huilin Chang, Gladies
Your UVA computing ID:hc5hq

Exercise: Visualizations

line plot: Micron Stock Price Trend Chart

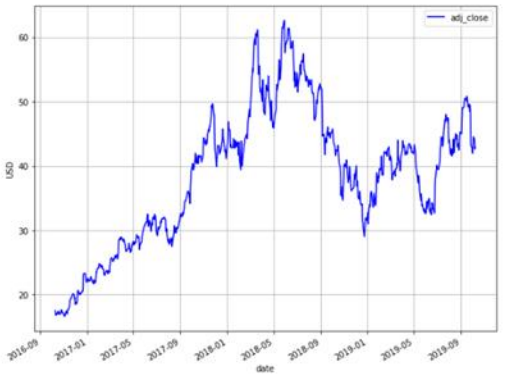
I tried to understand the Micron daily stock price using a trend chart to see the changes of the stock price over time.

Parameters:

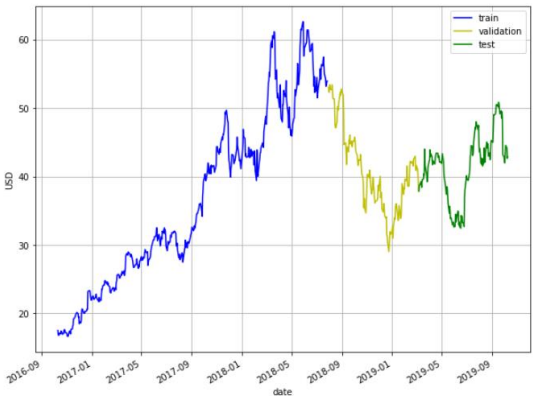
x axis: date

y axis: USD

```
[5]: # Plot adjusted close over time
rcParams['figure.figsize'] = 10, 8 # width 10, height 8

ax = df.plot(x='date', y='adj_close', style='b-', grid=True)
ax.set_xlabel("date")
ax.set_ylabel("USD")

[5]: Text(0, 0.5, 'USD')
```




```
#Plot adjusted close over time
rcParams['figure.figsize'] = 10, 8 # width 10, height 8

ax = train.plot(x='date', y='adj_close', style='b-', grid=True)
ax = cv.plot(x='date', y='adj_close', style='y-', grid=True, ax=ax)
ax = test.plot(x='date', y='adj_close', style='g-', grid=True, ax=ax)
ax.legend(['train', 'validation', 'test'])
ax.set_xlabel("date")
ax.set_ylabel("USD")

Text(0, 0.5, 'USD')
```



bar plot

I tried to understand the distribution of player's height in FIFA. To do this I used a bar plot to count the number of participants' height.

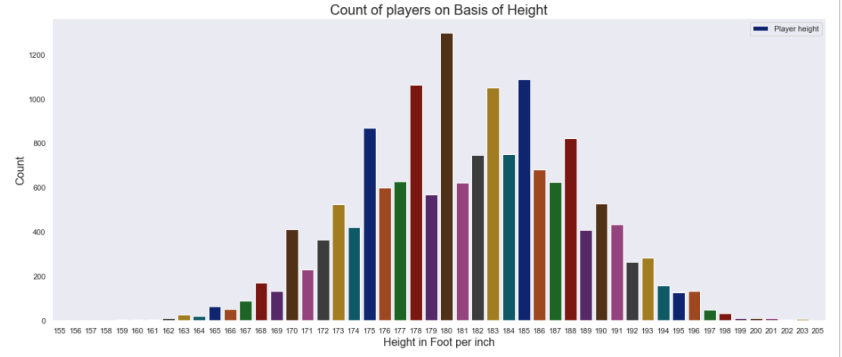
Parameters:

x axis: Height

y axis: Counts

Players Height distribution -180cm

```
# Height of Players

plt.figure(figsize = (20, 8))
ax = sns.countplot(x = 'Height', data = df, palette = 'dark')
ax.set_title(label = 'Count of players on Basis of Height', fontsize = 20)
ax.set_xlabel(xlabel = 'Height in Foot per inch', fontsize = 16)
ax.set_ylabel(ylabel = 'Count', fontsize = 16)
plt.legend(labels=['Player height'])
plt.show()
```



I tried to understand the distribution of player's position in FIFA. To do this I used a bar plot to count the number of participants' position in.

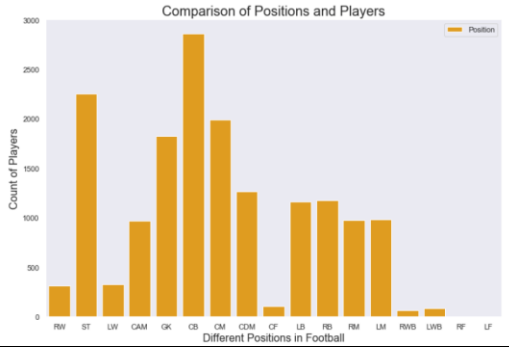
Parameters:

x axis: Height

y axis: Counts

Different position acquired by the players

```
plt.figure(figsize = (12, 8))
sns.set(style = 'dark', palette = 'colorblind', color_codes = True)
ax = sns.countplot('Position', data = df, color = 'orange')
ax.set_xlabel(xlabel = 'Different Positions in Football', fontsize = 16)
ax.set_ylabel(ylabel = 'Count of Players', fontsize = 16)
ax.set_title(label = 'Comparison of Positions and Players', fontsize = 20)
plt.legend(labels=['Position'])
plt.show()
```



I tried to understand the ranking of countries by estimating which country has more participants in FIFA. To do this I used a bar plot to count the number of participants.

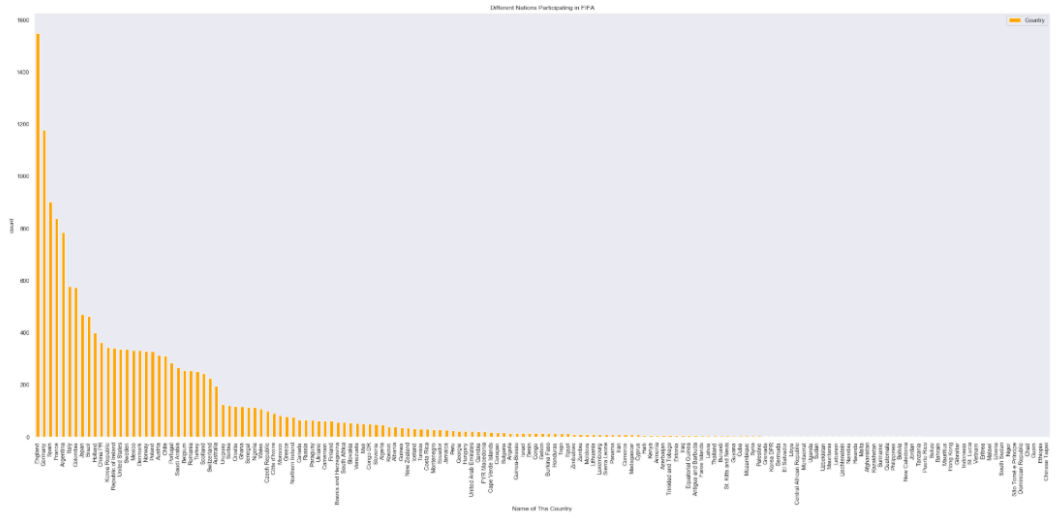
Parameters:

x axis: Country

y axis: Counts

Top three national participating- England, Germany and Spain

```
# To show Different nations participating in the FIFA 2019

df['Country'].value_counts().plot.bar(color = 'orange', figsize = (35, 15 ))
plt.title('Different Nations Participating in FIFA')
plt.xlabel('Name of The Country')
plt.ylabel('count')
plt.legend()
plt.show()
```



histogram

(interactive plot)

I'm interested in trying to understand the overall rating distribution from different countries. I used Histogram plots to compare the distribution of each country' performance.

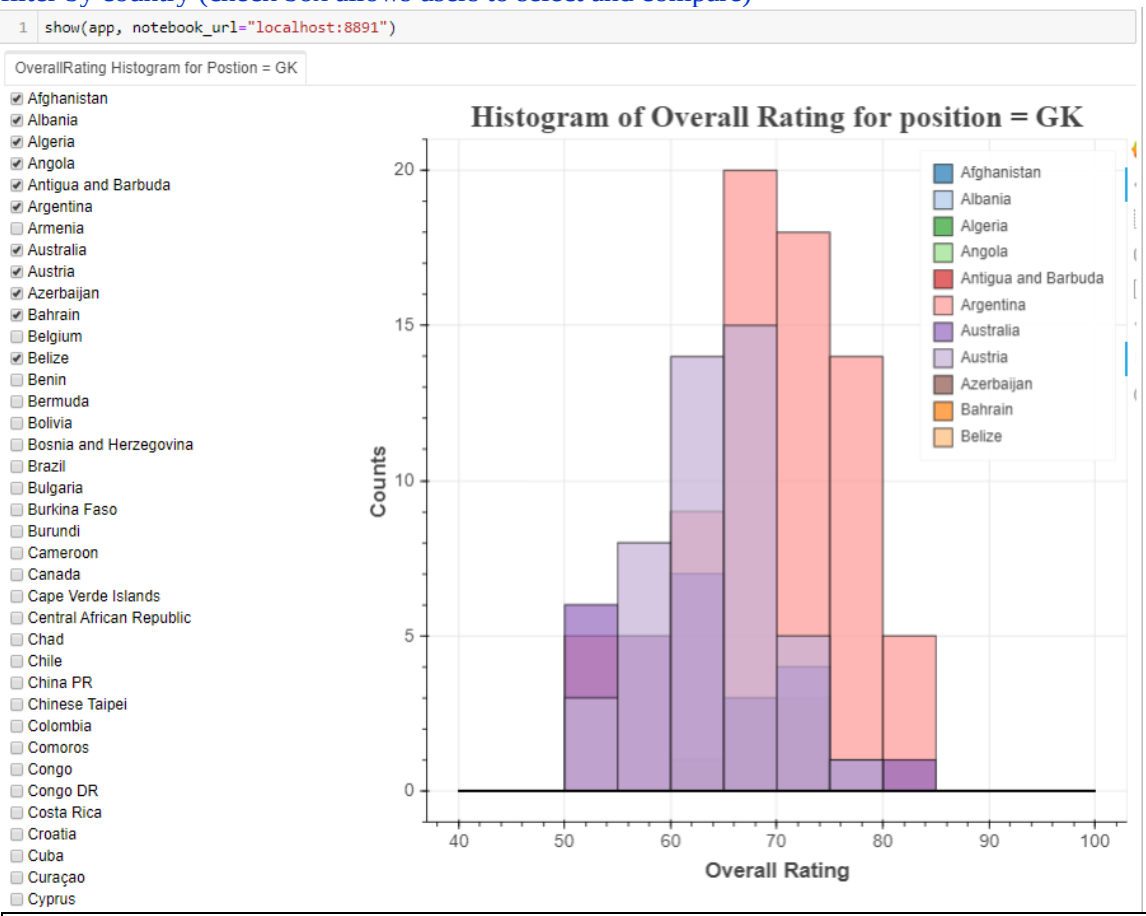
Parameters:

x axis: Overall Rating

y axis: Counts

filter by country (check box allows users to select and compare)

```
In [25]: 1 show(app, notebook_url="localhost:8891")
```



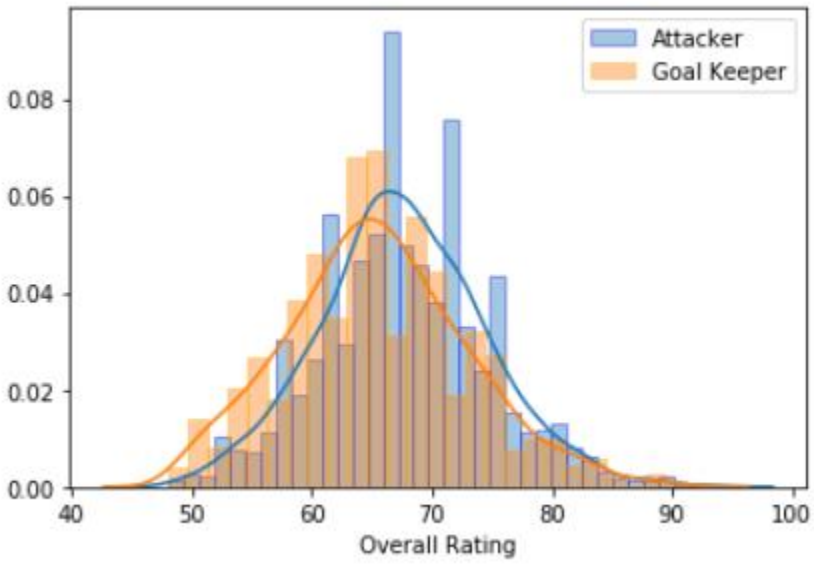
I'm interested in trying to understand the overall rating distribution from position group. I used Histogram plots to compare the distribution of players' position performance.

```
:
1 f_fuko = sample.loc[sample['Position Group']=='Attacker']['Overall Rating']
2 m_fuko = sample.loc[sample['Position Group']=='Goal Keeper']['Overall Rating']
3 # f1_fuko = sample.loc[sample['Position Group']=='Midfieder']['Overall Rating']
4 # m1_fuko = sample.loc[sample['Position Group']=='Defender']['Overall Rating']
5 sns.distplot(f_fuko, hist=True, kde=True, rug=False, hist_kws={'edgecolor':'blue'}, label='Attacker')
6 sns.distplot(m_fuko, hist=True, kde=True, rug=False, hist_kws={'edgecolor':'orange'}, label='Goal Keeper')
7 # sns.distplot(f1_fuko, hist=True, kde=True, rug=False, hist_kws={'edgecolor':'green'}, label='Midfieder')
8 # sns.distplot(m1_fuko, hist=True, kde=True, rug=False, hist_kws={'edgecolor':'pink'}, label='Defender')
9 plt.legend()
```

x axis: Overall Rating

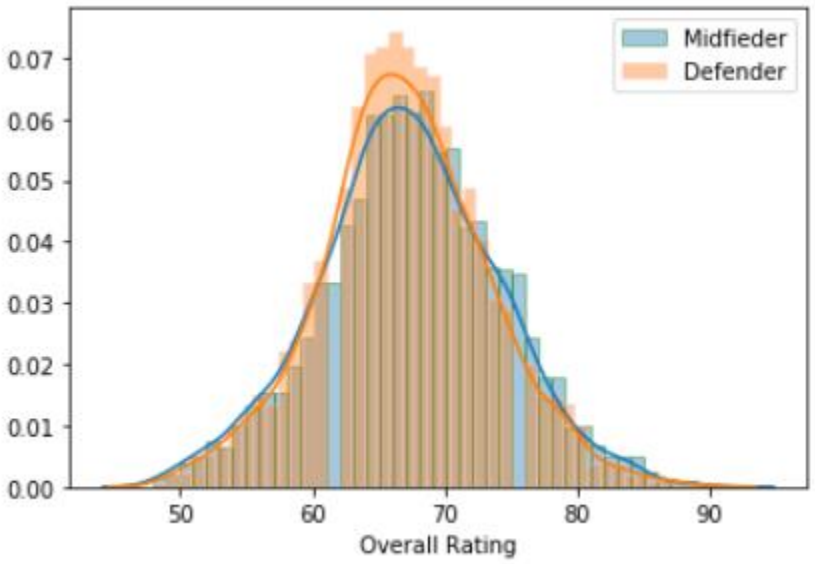
y axis: Counts

filter by Attacked and Goal Keeper




```
# f_fuko = sample.loc[sample['Position Group']=='Attacker']['Overall Rating']
# m_fuko = sample.loc[sample['Position Group']=='Goal Keeper']['Overall Rating']
f1_fuko = sample.loc[sample['Position Group']=='Midfieder']['Overall Rating']
m1_fuko = sample.loc[sample['Position Group']=='Defender']['Overall Rating']
# sns.distplot(f_fuko, hist=True, kde=True, rug=False, hist_kws={'edgecolor':'blue'}, label='Attacker')
# sns.distplot(m_fuko, hist=True, kde=True, rug=False, hist_kws={'edgecolor':'orange'}, label='Goal Keeper')
sns.distplot(f1_fuko, hist=True, kde=True, rug=False, hist_kws={'edgecolor':'green'}, label='Midfieder')
sns.distplot(m1_fuko, hist=True, kde=True, rug=False, hist_kws={'edgecolor':'pink'}, label='Defender')
plt.legend()
```

x axis: Overall Rating
y axis: Counts
filter by Midfieder and Defender



Scatter plot (interactive)

I tried to understand the attributes of FIFA players needed to obtain a higher performance score. A scatter plot was used with variables x, y as attributes.

Parameters – x, y, player features which allow users to select from drop down

Color- player's features which allow users to select from drop down

Size - player's features which allow users to select from drop down

```
In [1]:
import os
os.getcwd()
```

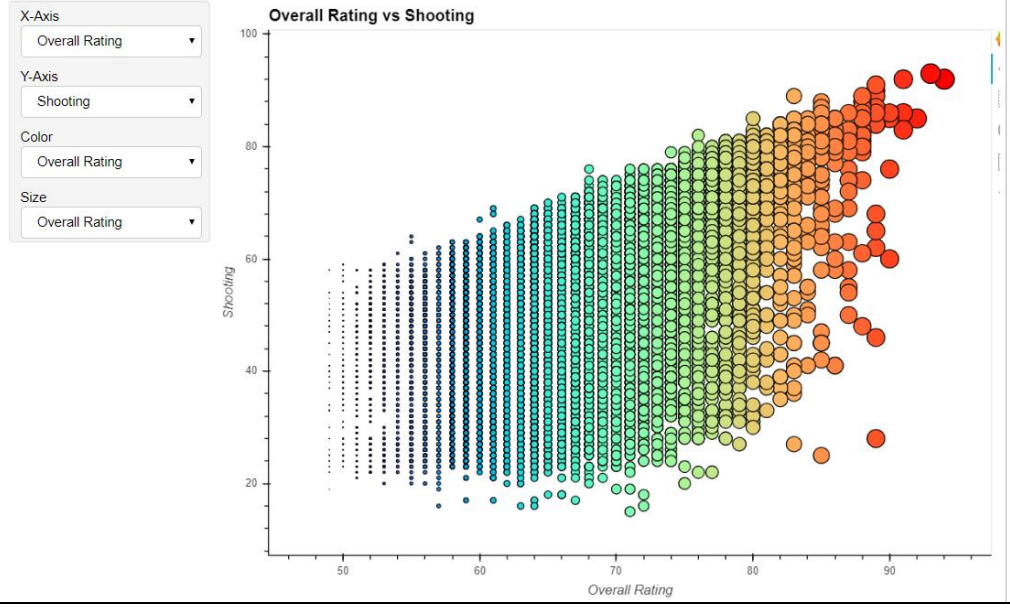
```
Out[1]: 'C:\\Users\\gladies\\bokeh_app'
```

```
In [2]:
import holoviews as hv
import panel as pn
import panel.widgets as pnw
import pandas as pd
from bokeh.palettes import Category20
```

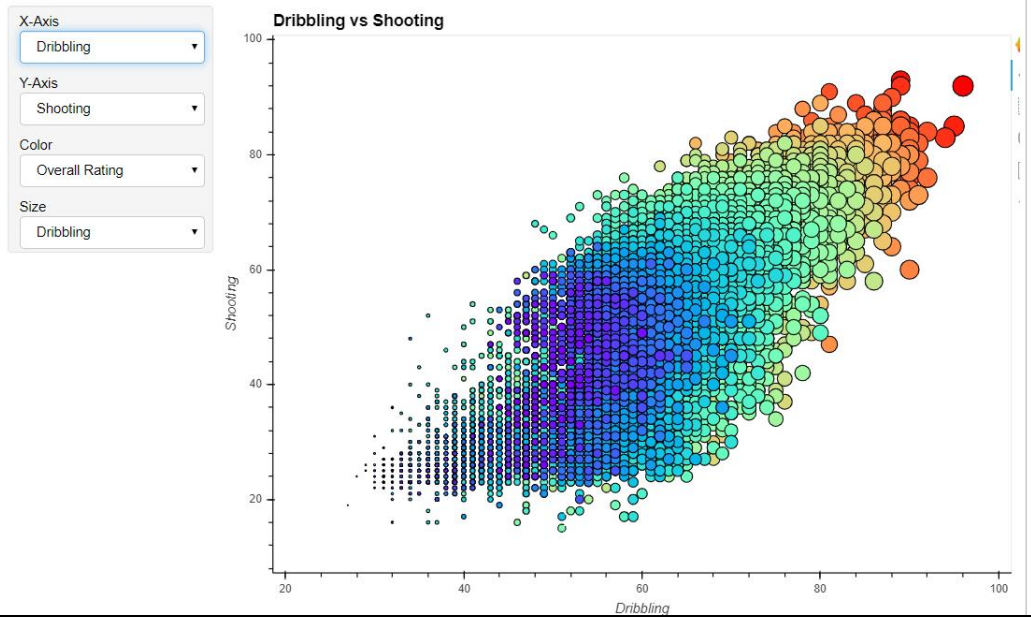
```
In [3]:
df=pd.read_csv('FIFA_1112.csv')
```

```
In [4]:
"""
An example demonstrating how to put together a cross-selector app based
on the FIFA dataset.
"""
hv.extension("bokeh","matplotlib")
import holoviews as hv
import panel as pn
import panel.widgets as pnw

from bokeh.sampledata.autompg import autompg

columns = sorted(df.columns)
discrete = [x for x in columns if df[x].dtype == object]
continuous = [x for x in columns if x not in discrete]
quantileable = [x for x in continuous if len(df[x].unique()) > 20]

x = pnw.Select(name='X-Axis', value='Overall Rating', options=quantileable)
y = pnw.Select(name='Y-Axis', value='Pace', options=quantileable)
size = pnw.Select(name='Size', value='None', options=['None'] + quantileable)
color = pnw.Select(name='Color', value='None', options=['None'] + quantileable)

@pn.depends(x.param.value, y.param.value, color.param.value, size.param.value)
def create_figure(x, y, color, size):
    opts = dict(cmap='rainbow', width=800, height=600, padding=0.1, line_color='black')
    if color != 'None':
        opts['color'] = color
    if size != 'None':
        opts['size'] = hv.dim(size).norm()*20
    return hv.Points(df, [x, y], label="%s vs %s" % (x.title(), y.title())).opts(**opts)

widgets = pn.WidgetBox(x, y, color, size, width=200)

pn.Row(widgets, create_figure).servable('Cross-selector')
```

Out[4]:



Out[4]:



box plot

I tried to understand the statistical distribution (medium, 25 quantile, 75 quantile) of top position group by using box plot.

Parameters

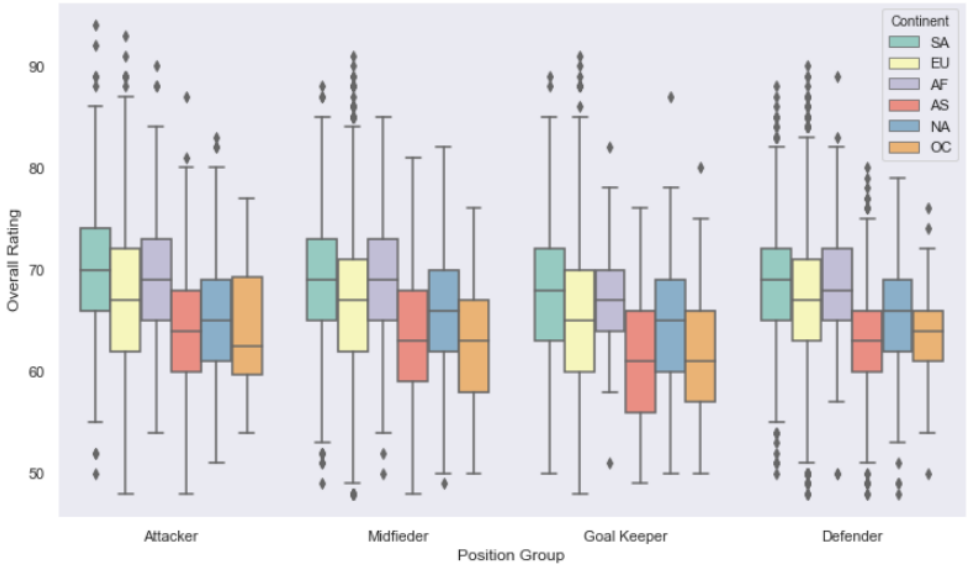
x axis: Position Group, y-axis: Overall Rating

filter by Continent

We can do either side by side or separate by each continent

In [38]:

```
ax = sns.boxplot(x="Position Group", y="Overall Rating", hue="Continent",
...                 data=df, palette="Set3")
```



In [41]:
```
 g = sns.catplot(x="Position Group", y="Overall Rating",
...                 hue="Continent", col="Continent",
...                 data=df, kind="box",
...                 height=4, aspect=.7);
g.set_xticklabels(rotation=30)
```

Out[41]: <seaborn.axisgrid.FacetGrid at 0x1b5681dd710>