

## Step 1+2: Installing Libraries + Datasets

```
In [ ]: import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns

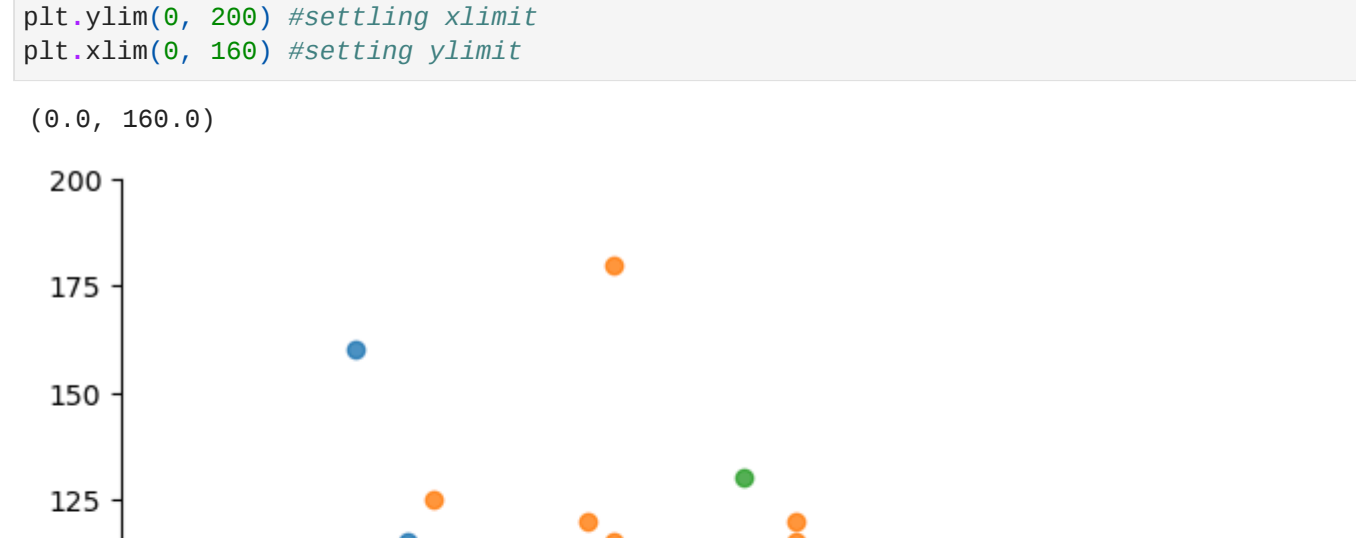
In [12]: df = pd.read_csv('/Users/ellienu/Downloads/Pokemon.csv', index_col = 0, encoding = 'latin')
df.head()
```

	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Stage	Legendary
#												
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	2	False
3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	3	False
4	Charmeleon	Fire	NaN	309	39	52	43	60	50	65	1	False
5	Charmander	Fire	NaN	405	58	64	58	80	65	80	2	False

## Step3: Seaborn's Plotting Functions


```
In [14]: sns.lmplot(x = 'Attack', y = 'Defense', data = df)

Out[14]: <seaborn.axisgrid.FacetGrid at 0x16531f510>
```



```
In [16]: sns.lmplot(x = 'Attack', y = 'Defense', data = df,
fit_reg = False, #no regression line
hue = 'Stage') #color by evolution stage

Out[16]: <seaborn.axisgrid.FacetGrid at 0x165568ad0>
```

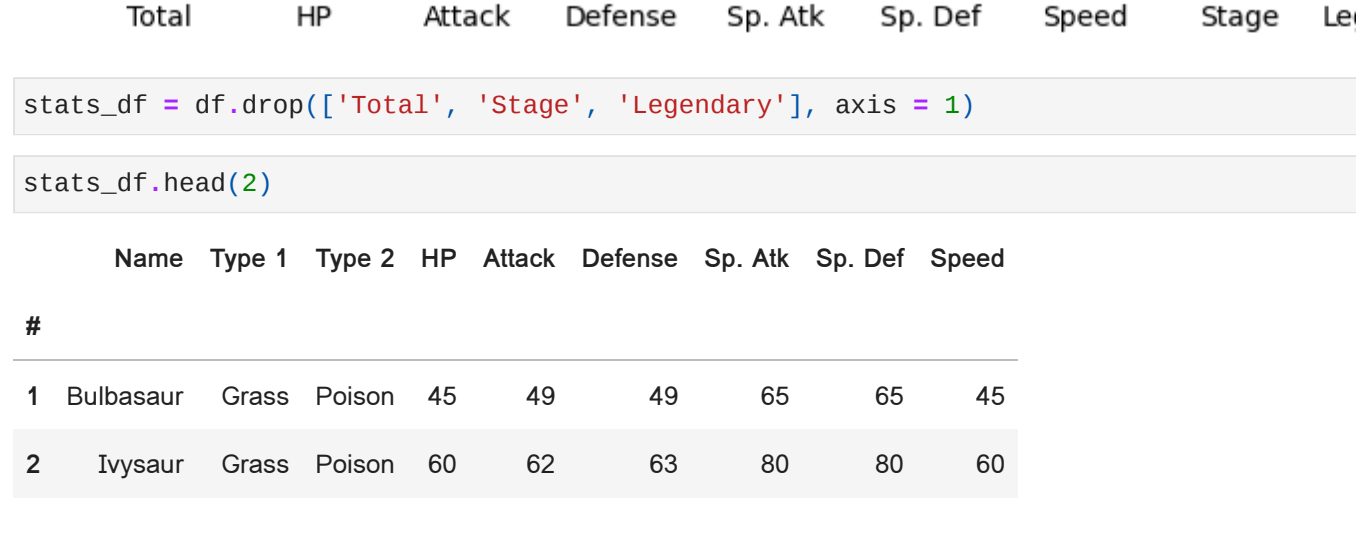


## Step 4: Customizing with Matplotlib

```
In [19]: sns.lmplot(x = 'Attack', y = 'Defense', data = df,
fit_reg = False, #no regression line
hue = 'Stage') #color by evolution stage

plt.ylim(0, 200) #setting ylimit
plt.xlim(0, 160) #setting xlimit

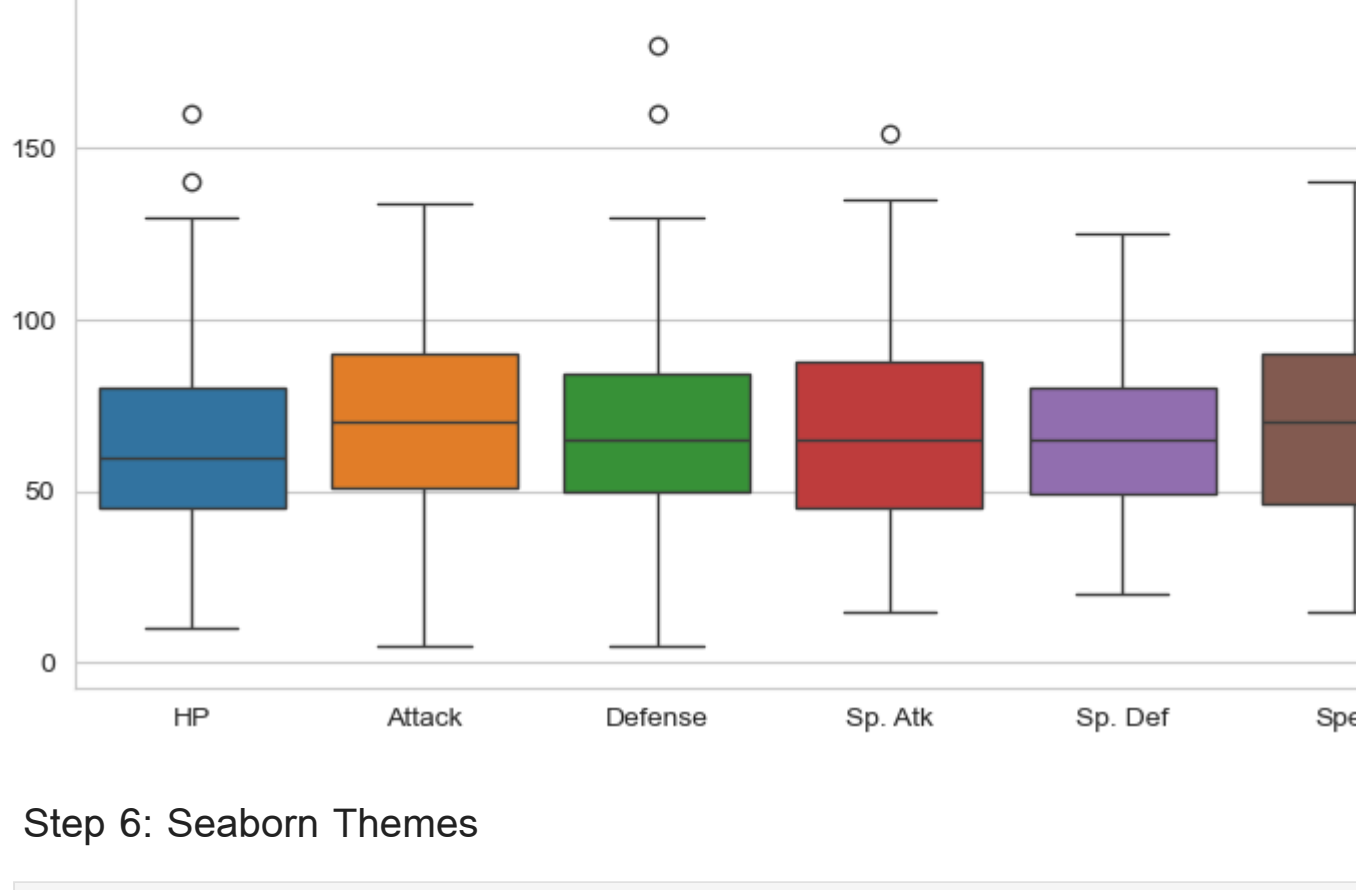
Out[19]: (0.0, 160.0)
```



## Step5: The Role of Pandas

```
In [24]: #Boxplot
plt.figure(figsize=(9,6)) #set plot dimensions
sns.boxplot(data = df)

Out[24]: <Axes: >
```



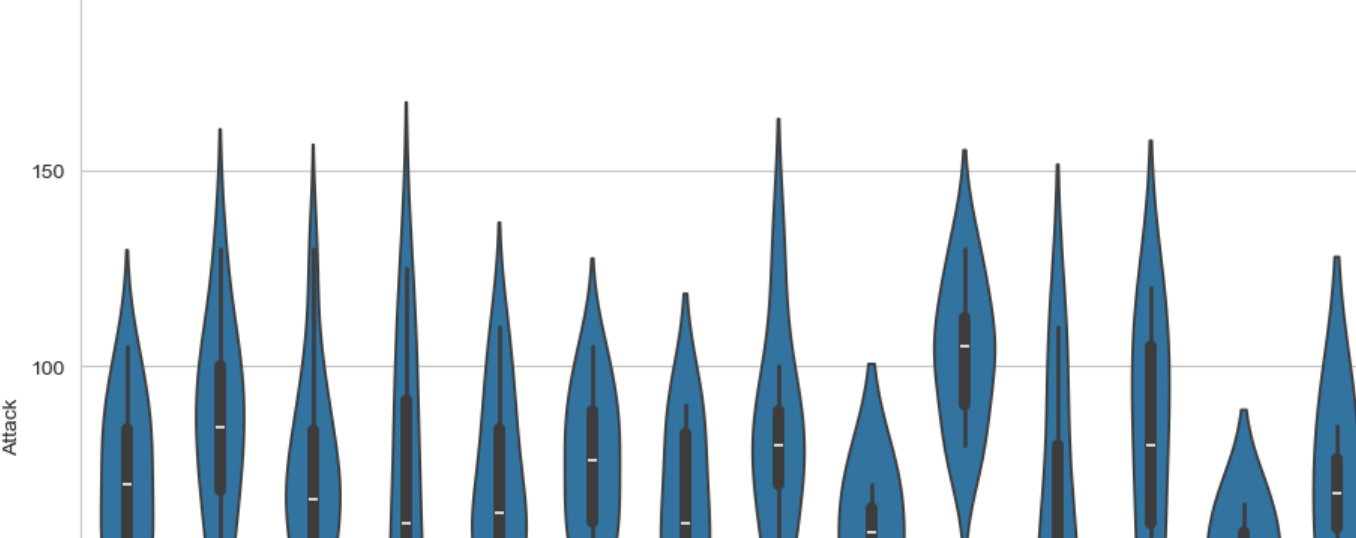
```
In [26]: stats_df = df.drop(['Total', 'Stage', 'Legendary'], axis = 1)

In [28]: stats_df.head(2)
```

	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed
#									
1	Bulbasaur	Grass	Poison	45	49	49	65	65	45
2	Ivysaur	Grass	Poison	60	62	63	80	80	60

```
In [72]: #New boxplot using stats_df
plt.figure(figsize = (9,6)) #set plot dimensions
sns.boxplot(data = stats_df)

Out[72]: <Axes: >
```

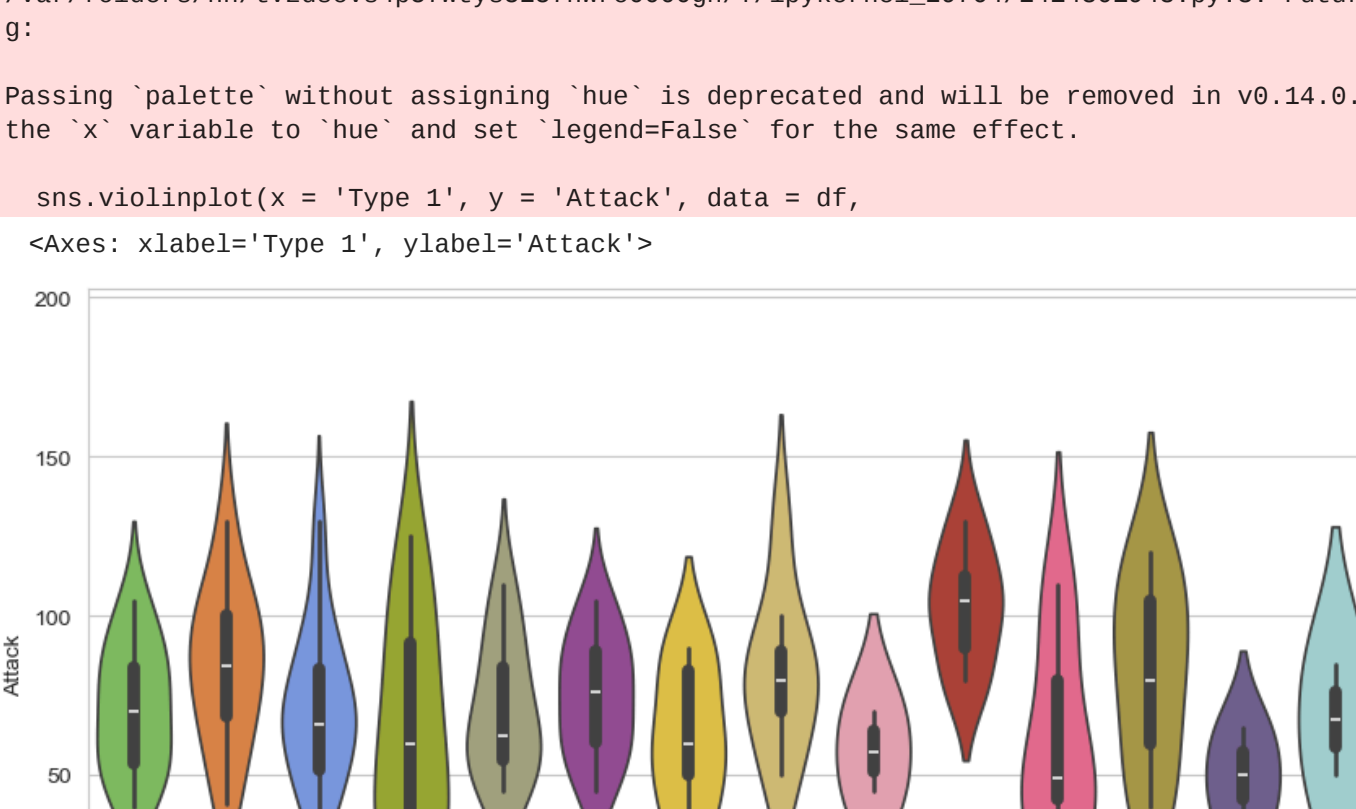


## Step 6: Seaborn Themes

```
In [34]: sns.set_style('whitegrid')

plt.figure(figsize = (12,8)) #set plot dimensions
sns.violinplot(x = 'Type 1', y = 'Attack', data = df)

Out[34]: <Axes: xlabel='Type 1', ylabel='Attack'>
```



## Step 7: Color Palettes

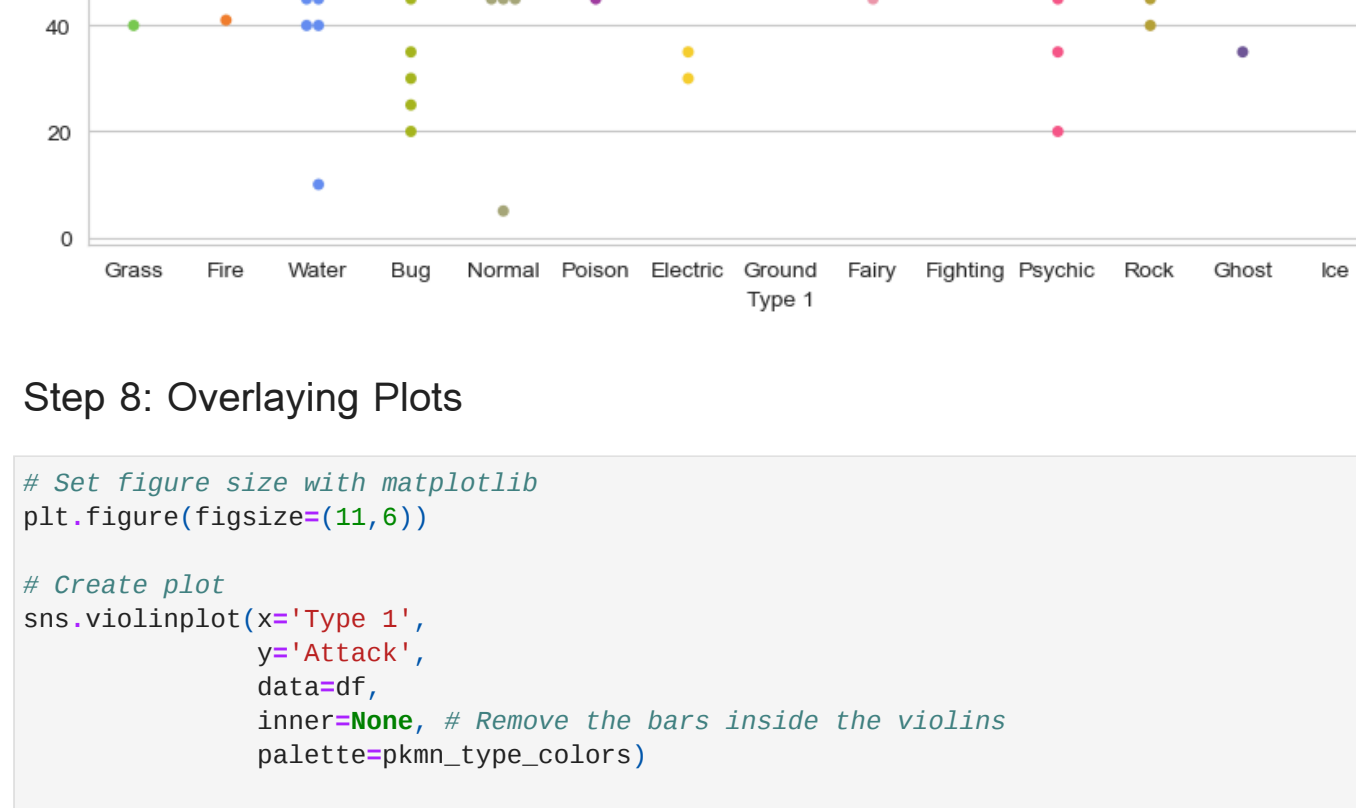
```
In [35]: pkmn_type_colors = {'Grass': '#77C850', # Grass
'Fire': '#F08080', # Fire
'Water': '#A9A9A9', # Water
'Bug': '#A8B820', # Bug
'Normal': '#A9A9A9', # Normal
'Poison': '#A9A9A9', # Poison
'Electric': '#F08080', # Electric
'Ground': '#E0C080', # Ground
'Fairy': '#E0C080', # Fairy
'Fighting': '#C08080', # Fighting
'Psychic': '#F08080', # Psychic
'Rock': '#808080', # Rock
'Ghost': '#808080', # Ghost
'Ice': '#808080', # Ice
'Dragon': '#7030A0', # Dragon
}

In [38]: #Violin plot with Pokemon color palette
plt.figure(figsize = (11, 6)) #set plot dimensions
sns.violinplot(x = 'Type 1', y = 'Attack', data = df,
palette = pkmn_type_colors) #set color palette

/var/folders/hn/tvzdvscvs4p57wtys5257hwrc0000gn/T/ipykernel_29764/4033301071.py:3: FutureWarning:
g:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign
the 'x' variable to 'hue' and set 'legend=False' for the same effect.

sns.violinplot(x = 'Type 1', y = 'Attack', data = df,

Out[38]: <Axes: xlabel='Type 1', ylabel='Attack'>
```

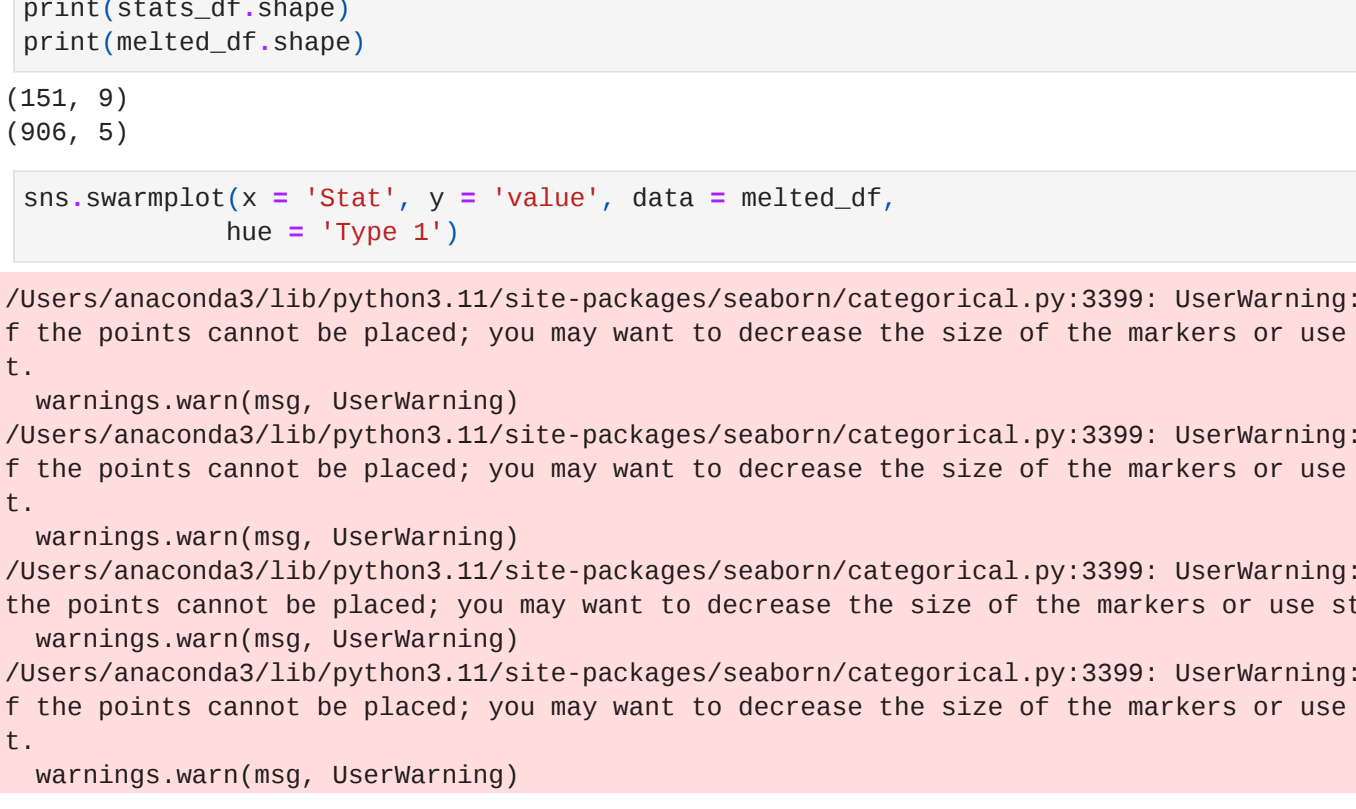


```
In [73]: #swarm plot with Pokemon color palette
plt.figure(figsize = (11, 6)) #set plot dimensions
sns.swarmplot(x = 'Type 1', y = 'Attack', data = df,
palette = pkmn_type_colors)

/var/folders/hn/tvzdvscvs4p57wtys5257hwrc0000gn/T/ipykernel_29764/4033301071.py:3: FutureWarning:
g:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign
the 'x' variable to 'hue' and set 'legend=False' for the same effect.

sns.swarmplot(x = 'Type 1', y = 'Attack', data = df,

Out[73]: <Axes: xlabel='Type 1', ylabel='Attack'>
```



## Step 8: Overlaying Plots

```
In [75]: # Set figure size with matplotlib
plt.figure(figsize=(11,6))

# Create plot
sns.violinplot(x='Type 1',
y='Attack',
data=df,
inner=None, # Remove the bars inside the violins
palette=pkmn_type_colors)


sns.swarmplot(x='Type 1',
y='Attack',
data=df,
color='k', # Make points black
alpha=0.7) # and slightly transparent

# Set title with matplotlib
plt.title('Attack by Type')

/var/folders/hn/tvzdvscvs4p57wtys5257hwrc0000gn/T/ipykernel_29764/149021041.py:5: FutureWarning:
g:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign
the 'x' variable to 'hue' and set 'legend=False' for the same effect.

sns.violinplot(x='Type 1',

Out[75]: <Axes: xlabel='Type 1', ylabel='Attack'>
```



## Step 9: Putting It all Together

```
In [76]: #Melt DataFrame
melted_df = pd.melt(stats_df,
id_vars = ['Name', 'Type 1', 'Type 2'], #variables to keep
var_name = 'Stat') #name of melted variable

In [46]: print(stats_df.shape)

(151, 9)
(906, 5)

In [47]: sns.swarmplot(x = 'Stat', y = 'value', data = melted_df,
hue = 'Type 1')

/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 20.5% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 11.3% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 7.3% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 15.2% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 23.8% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 13.9% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 17.9% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 10.6% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 18.5% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)
/Users/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:3399: UserWarning: 25.2% o
f the points cannot be placed; you may want to decrease the size of the markers or use stripplo
t.
warnings.warn(msg, UserWarning)

Out[47]: <Axes: xlabel='Stat', ylabel='value'>
```



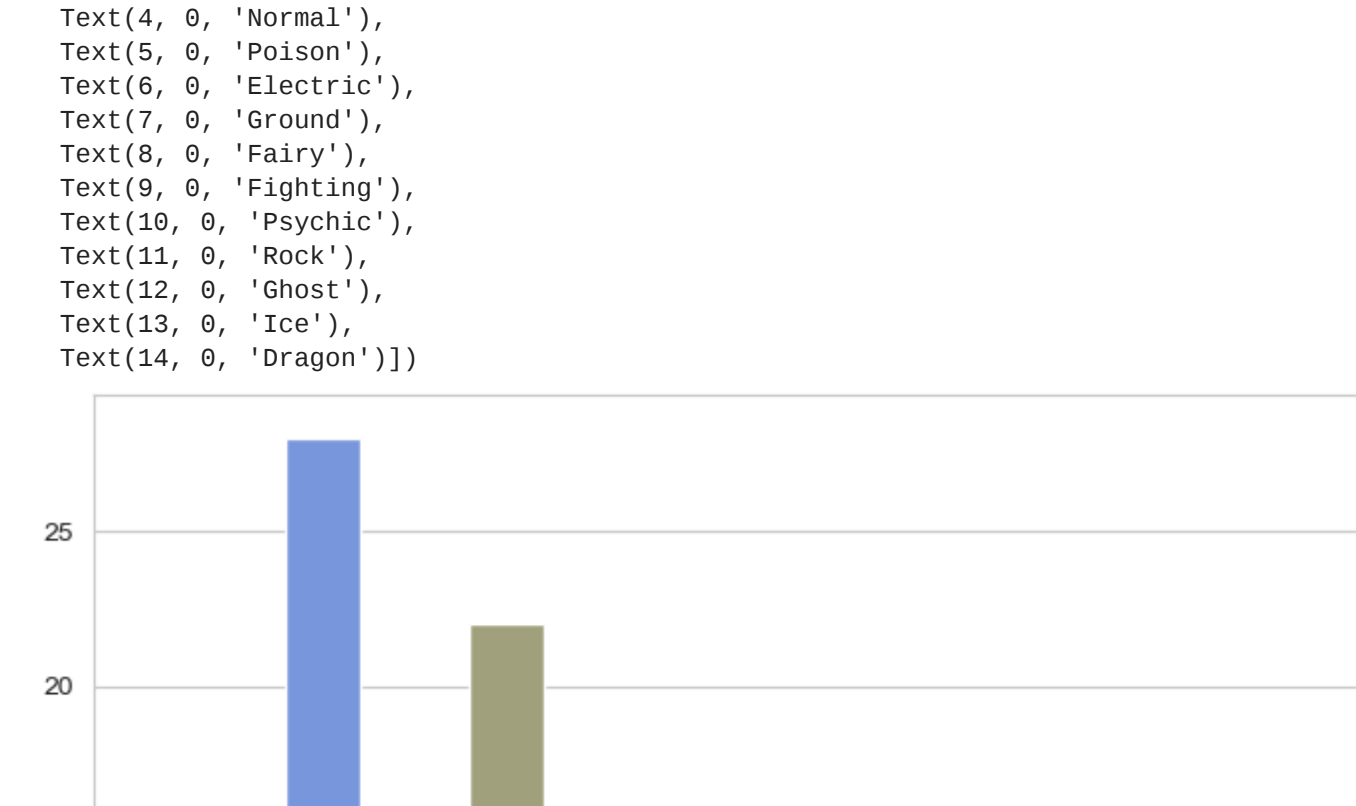
```
In [61]: plt.figure(figsize = (12, 8))

sns.swarmplot(x = 'Stat',
y = 'value',
data = melted_df,
hue = 'Type 1',
dodge = True, # 2. Separate points by hue
palette = pkmn_type_colors) #3. Use Pokemon palette

#4. Adjust the y-axis
plt.ylim(0, 260)

#5. Place legend to the right
plt.legend(bbox_to_anchor = (1,1), loc = 2)

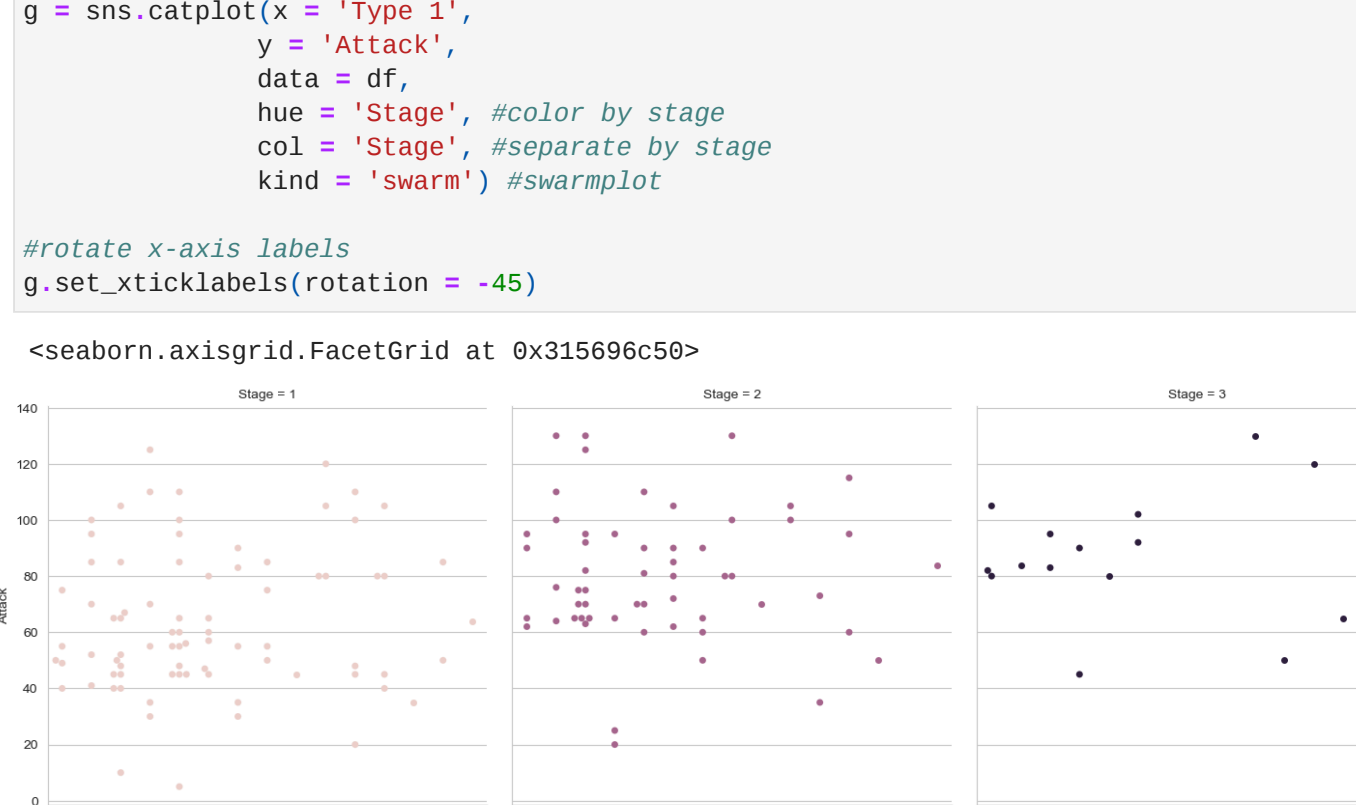
Out[61]: <matplotlib.legend.Legend at 0x314605490>
```



## Histogram

```
In [77]: #Distribution Plot
plt.figure(figsize = (9,6))
sns.histplot(df, x = 'Attack',
kde = True) #plot the density curve too

Out[77]: <Axes: xlabel='Attack', ylabel='Count'>
```



## 10 - Bar Plot

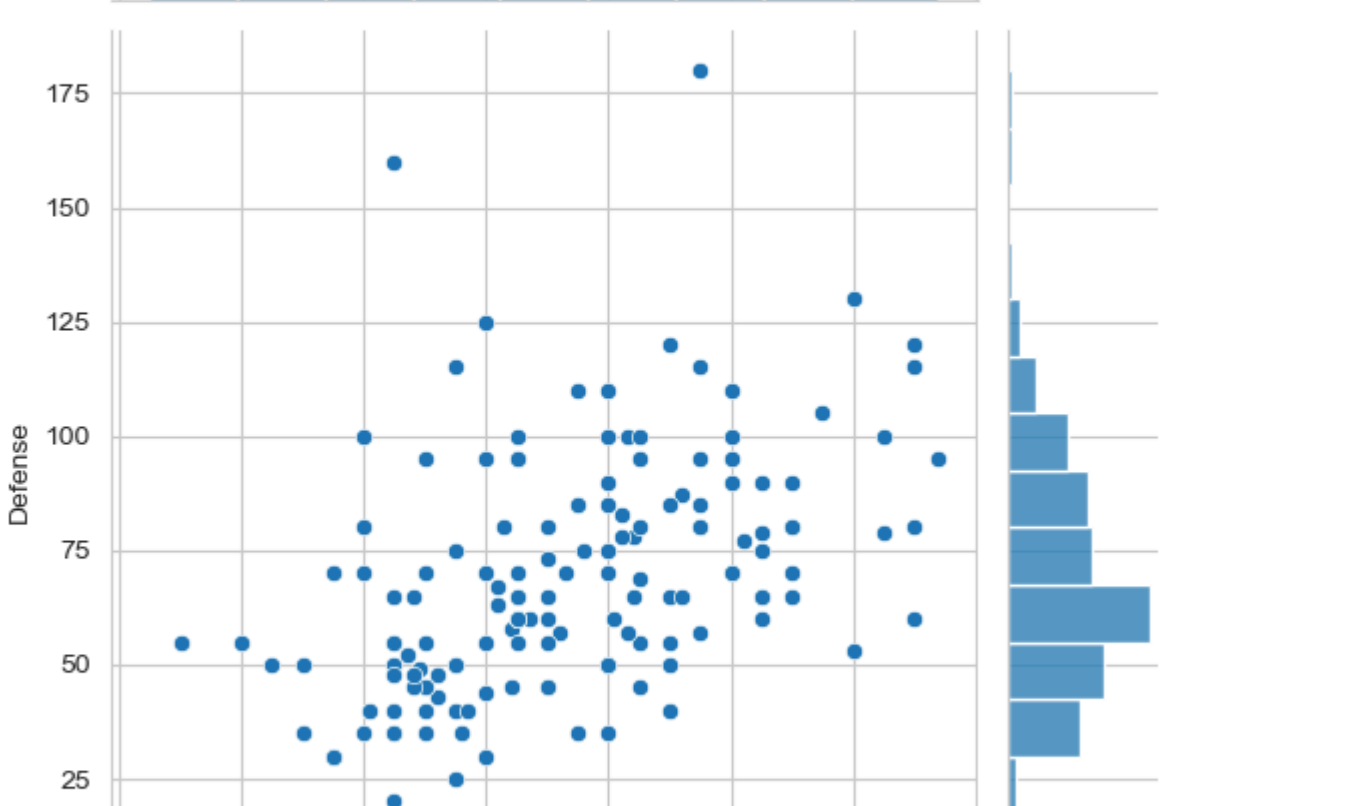
```
In [66]: plt.figure(figsize = (9,6))
sns.countplot(x = 'Type 1', data = df, palette = pkmn_type_colors)

plt.xticks(rotation = -45)

/var/folders/hn/tvzdvscvs4p57wtys5257hwrc0000gn/T/ipykernel_29764/1299667369.py:2: FutureWarning:
g:
Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign
the 'x' variable to 'hue' and set 'legend=False' for the same effect.

sns.countplot(x = 'Type 1', data = df, palette = pkmn_type_colors)

Out[66]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14],
Text(0, 0, 'Grass'),
Text(1, 0, 'Fire'),
Text(2, 0, 'Water'),
Text(3, 0, 'Bug'),
Text(4, 0, 'Normal'),
Text(5, 0, 'Poison'),
Text(6, 0, 'Electric'),
Text(7, 0, 'Ground'),
Text(8, 0, 'Fairy'),
Text(9, 0, 'Fighting'),
Text(10, 0, 'Psychic'),
Text(11, 0, 'Rock'),
Text(12, 0, 'Ghost'),
Text(13, 0, 'Ice'),
Text(14, 0, 'Dragon'))]
```




## 10- CatPlot(Categorical Plot)

```
In [68]: g = sns.catplot(x = 'Type 1',
y = 'Attack',
data = df,
hue = 'Stage', #color by stage
col = 'Stage', #separate by stage
kind = 'swarm') #swarmplot

#rotate x-axis labels
g.set_xticklabels(rotation = -45)

Out[68]: <seaborn.axisgrid.FacetGrid at 0x315696c50>
```



## Density Plot

```
In [69]: sns.kdeplot(x = 'Attack', y = 'Defense', data = df)

Out[69]: <Axes: xlabel='Attack', ylabel='Defense'>
```



## 10 - Joint Distribution Plot

```
In [71]: sns.jointplot(x = 'Attack', y = 'Defense', data = df)

Out[71]: <seaborn.axisgrid.JointGrid at 0x3159c1c10>
```



```
In [ ]:
```