Mohammed Aljubori

Final Project

First Gen Pokemon Data Visualization
I have chosen to use Pokemon data because a lot of people enjoy the subject and it was fun for me to make.

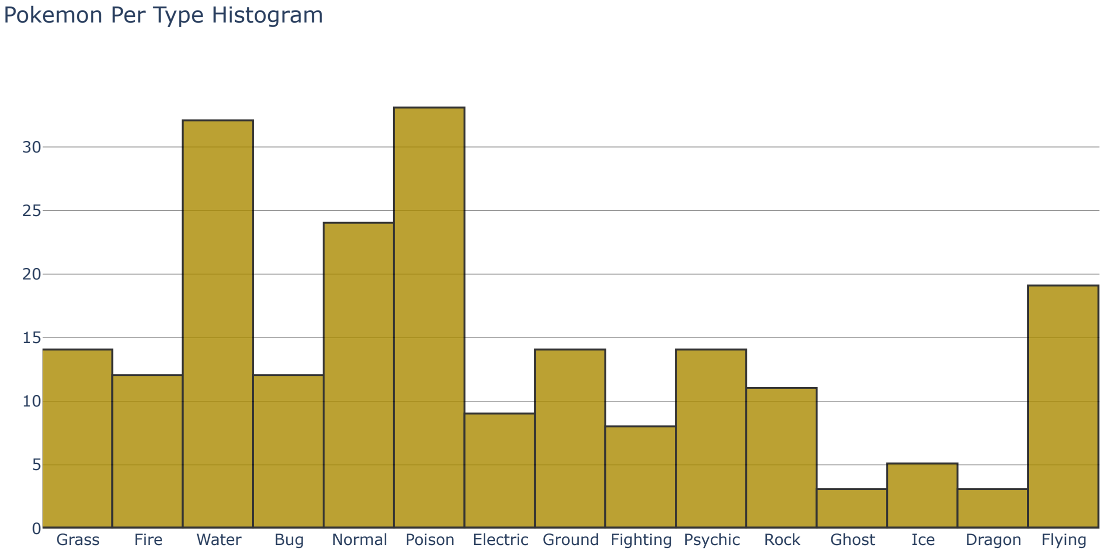
In [1]:

```
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly.express as px
import plotly.graph_objs as go
import plotly.figure_factory as FF
from plotly import tools
import numpy as np
import pandas as pd
init_notebook_mode(connected=True)
```

Viz 1.
This histogram compares the counts of each pokemon per type
The most common type in the first generation was the Poison type

In [2]:

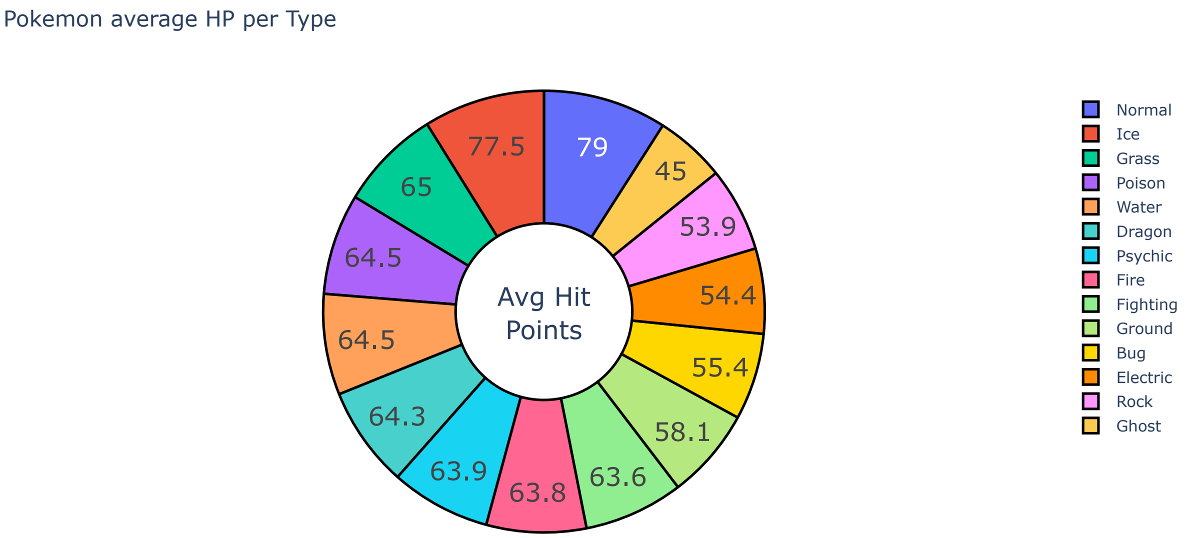
```
df = pd.read_csv('FirstGenPokemon.csv',sep='\s*,\s*',header=0, encoding='ascii', engine='python')
x2 = df['Type1'].str.title()
trace = go.Histogram(x=x2, histnorm="", marker_color='#AA8A00', opacity=0.8, hoverlabel_bgcolor='Silver',
                    marker_line_color='Black', marker_line_width=1.5)
layout = go.Layout(title='Pokemon Per Type Histogram', plot_bgcolor='white', bargroupgap=0, bargap=0)
fig2 = go.Figure(data=[trace], layout=layout)

iplot(fig2)
```



Viz 2.
This Pie chart shows the average hit points per type
The normal type has the highest average and the ghost type has the lowest average

In [25]:

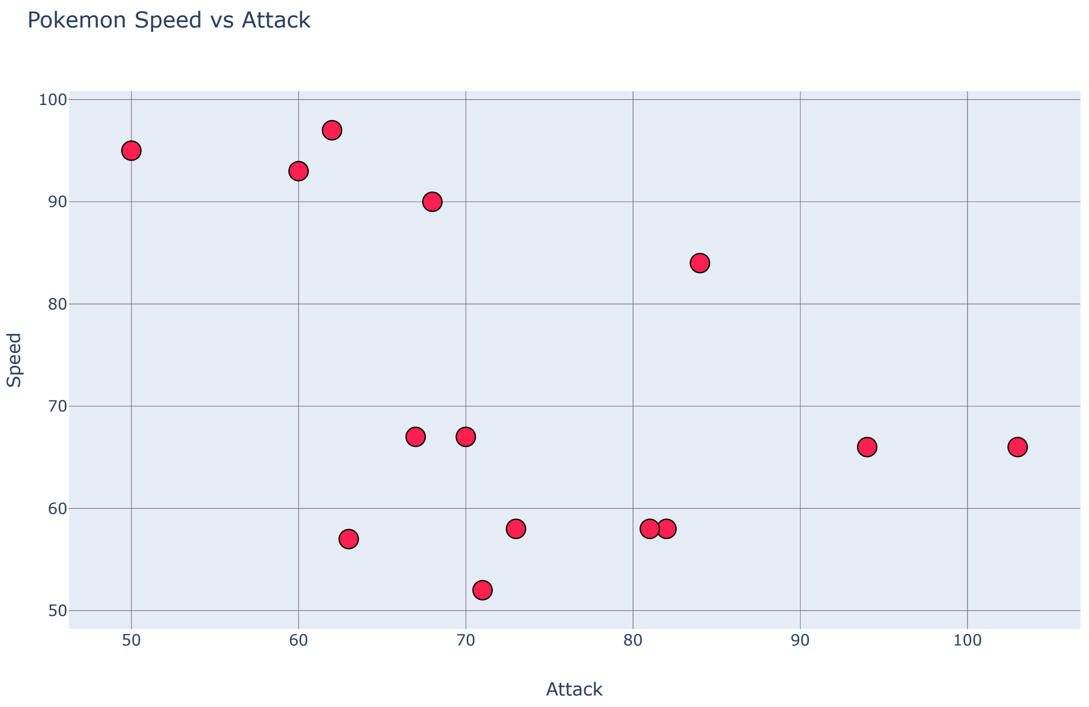
```
ch = df.iloc[0:151]

gkk = ch.groupby(['Type1'])['HP'].mean()
dp = gkk.reset_index()
colors = ['gold', 'mediumturquoise', 'darkorange', 'lightgreen']

fig = go.Figure(data=[go.Pie(labels=dp["Type1"].str.title(),
                             values=dp['HP'].round(1))])
fig.update_traces(hoverinfo='label+percent', textinfo='value', textfont_size=20,
                  marker=dict(colors=colors, line=dict(color='#000000', width=2)))
fig.update_traces(hole=.4, hoverinfo="label", name='')
fig.update_layout(title_text="Pokemon average HP per Type",
                  annotations=[dict(text='Avg Hit'+'<br>'+'Points', x=0.5, y=0.5, font_size=20, showarrow=False)])
fig.show()
```



Viz 3.
This scatter plot compares the general speed of pokemon vs the general attack per type most types are not similar for attack and speed with the exception of Rock and Ground types

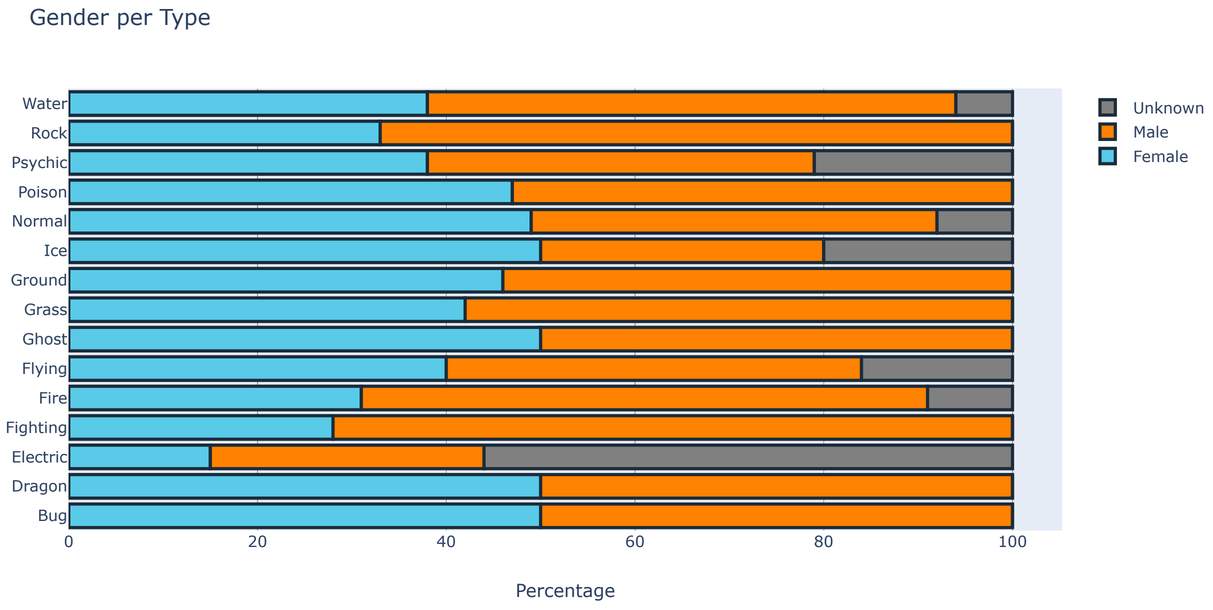
In [26]:

```
GA =df.iloc[0:151]
gap = GA.groupby(['Type1'], as_index=False)['Attack'].mean()
temp = GA.groupby(['Type1'], as_index=False)['Speed'].mean()
gap['Speed'] = temp['Speed'].astype(int)
gap.sort_values(by=['Attack'])


trace1 = go.Scatter(
    x=gap['Attack'].round(0),
    y=gap['Speed'],
    marker_color='#ff2052',
    marker_showscale=False,
    marker_size=15,
    marker_line_color='Black',
    marker_line_width=1,
    mode='markers',
    name='Attack Vs Speed',
    hoverlabel_bordercolor='Black',
    hoverlabel_font_color='White',
    text=
    'Type: ' + gap['Type1'].str.title()+ '<br>'
  + 'Attack: '+ gap['Attack'].round(0).astype("str") + '<br>'
  + 'Speed: ' + gap['Speed'].astype('str'),

)
layout = go.Layout(height=600, width=950, title='Pokemon Speed vs Attack',yaxis=go.layout.YAxis(automargin=True,
                                                                                        title=go.layout.yaxis.Title(text='Speed')),
 xaxis=go.layout.XAxis(title=go.layout.xaxis.Title(
                        text='Attack'))
                  )
fig = go.Figure(data=[trace1], layout=layout)
iplot(fig)
```



Viz 4.
This stacked horizontal bar chart shows the chance of a certain type of pokemon to be male, female or unkown

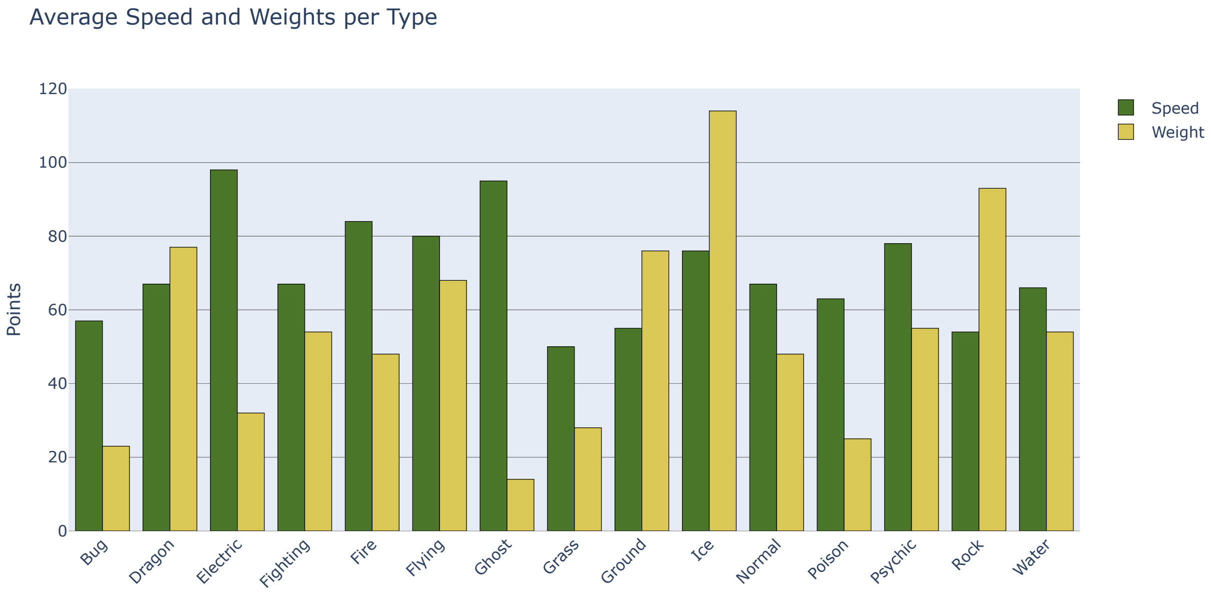
```
In [27]: male = df.groupby(['Type1'], as_index=False)['Male_Pct'].mean().round()
         female = df.groupby(['Type1'], as_index=False)['Female_Pct'].mean().round()
         unk = pd.DataFrame([male.Type1,female.Female_Pct,male.Male_Pct]).transpose()
         unk['Unknown_Pct'] = 100 - (unk['Male_Pct'] + unk['Female_Pct'])


         fig = go.Figure()
         fig.add_trace(go.Bar(
             y=female['Type1'].str.title(),
             x=female['Female_Pct'],
             name='Female',
             orientation='h',
             marker=dict(
                 color='#59cae8',
                 line=dict(color='#1b2b3a', width=2.5)
             )
         ))
         fig.add_trace(go.Bar(
             y=male['Type1'].str.title(),
             x=male['Male_Pct'],
             name='Male',
             orientation='h',
             marker=dict(
                 color='#ff8201',
                 line=dict(color='#1b2b3a', width=2.5)
             )
         ))
         fig.add_trace(go.Bar(
             y=unk['Type1'].str.title(),
             x=unk['Unknown_Pct'],
             name='Unknown',
             orientation='h',
             marker=dict(
                 color='Gray',
                 line=dict(color='#1b2b3a', width=2.5)
             )
         ))
         fig.update_layout(barmode='stack', title='Gender per Type', xaxis=go.layout.XAxis(title=go.layout.xaxis.Title(text='Percentage')))
         fig.show()
```



Viz5.
This grouped bar chart compares the average speed and weight points per type

In [28]:

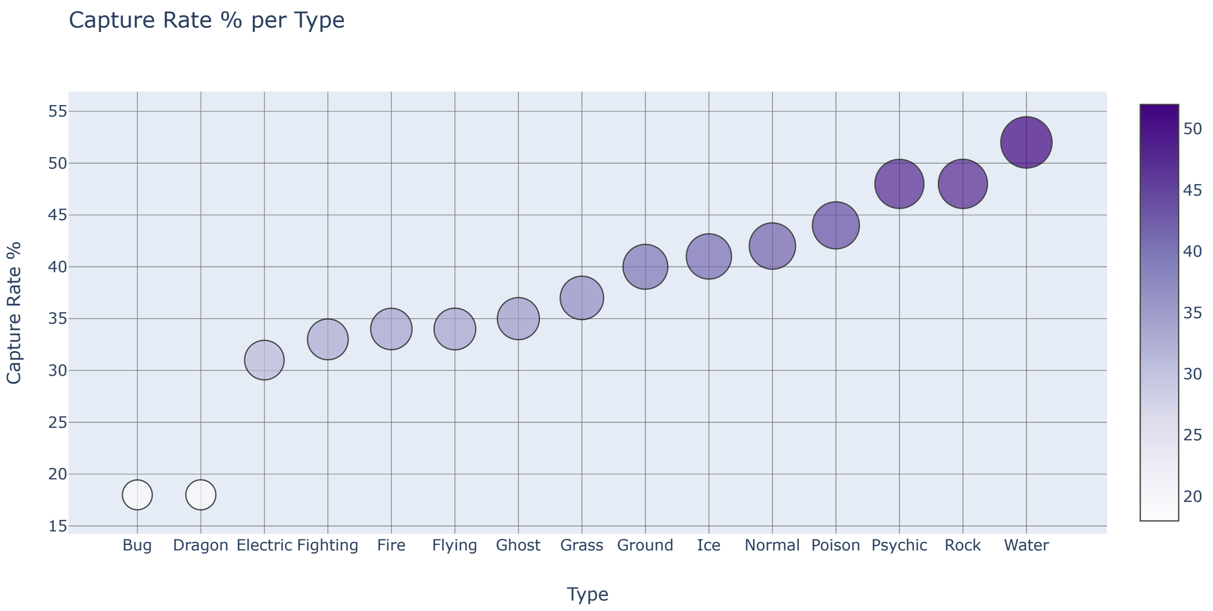
```
speed = df.groupby(['Type1'], as_index=False)['Speed'].mean().round()
weight = df.groupby(['Type1'], as_index=False)['Weight(kg)'].mean().round()


fig = go.Figure()
fig.add_trace(go.Bar(
    x=speed['Type1'].str.title(),
    y=speed['Speed'],
    name='Speed',
    marker_color='#4A7729',
    opacity=1,
    marker_line_color='black'

))
fig.add_trace(go.Bar(
    x=weight['Type1'].str.title(),
    y=weight['Weight(kg)'],
    name='Weight',
    marker_color='#D9C756',
    opacity=1,
    marker_line_color='black'

))

# Here we modify the tickangle of the xaxis, resulting in rotated labels.
fig.update_layout(barmode='group', xaxis_tickangle=-45,title='Average Speed and Weights per Type',
                  yaxis=go.layout.YAxis(title=go.layout.yaxis.Title(text='Points')))
fig.show()
```



Viz6.
This bubble chart shows the percentage of capturing a pokemon per type, the color and size of the bubble correspond to the %

In [29]:

```
cap = df.groupby(['Type1'], as_index=False)['Capt_Rate'].mean()
sortedz=cap.sort_values(by=['Capt_Rate'])
size = sortedz['Capt_Rate']/255
values = list(range(15,200,30))

fig = go.Figure(data=[go.Scatter(
    x=cap['Type1'].str.title(),
    y=size.round(2)*100,
    mode='markers',
    marker=dict(
        size=size,
        sizemode='area',
        sizeref=2.*max(size)/(40.**2),
        sizemin=4,
        color=size.round(2)*100,
        colorscale="Purples",
        showscale=True,
        line_color='Black'
    )

)])
fig.update_layout(
    title=go.layout.Title(
        text="Capture Rate % per Type",
        xref="paper",
        x=0
    ),
    xaxis=go.layout.XAxis(
        title=go.layout.xaxis.Title(
            text="Type"
        )
    ),
    yaxis=go.layout.YAxis(
        title=go.layout.yaxis.Title(
            text="Capture Rate %",
        )
    )
)

fig.show()
```



In [ ]: