In [1]:

```
# William Barker
# DSC630
# Week 1

# The dataset we will be exploring is a list of pokemon and their different types and stat
# We'll try and see which type has the best stats and which generation has the best stats.

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
# Import our data
df = pd.read_csv('Pokemon.csv')
df.head()
```

Out[2]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False

In [6]:

```
# I chose to create a new data frame to make making a bar graph simpler
dfbar = df.groupby('Type 1', as_index=False)['Total'].mean()
dfbar
```

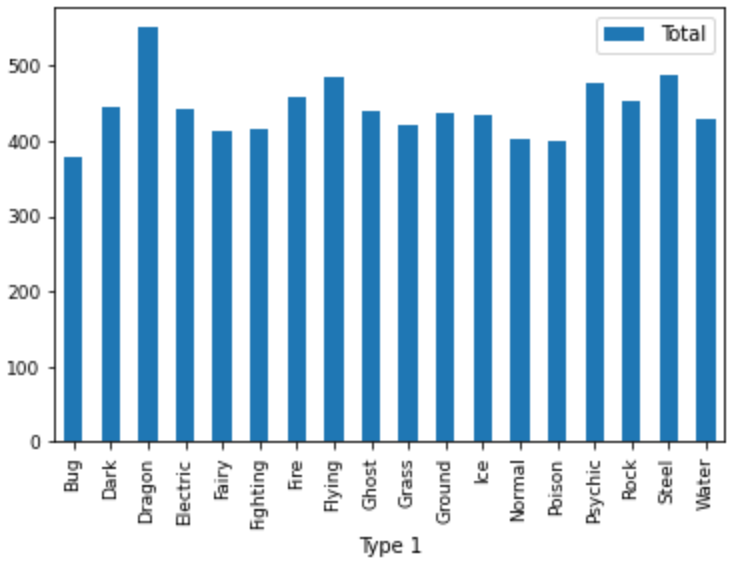
Out[6]:

	Type 1	Total
0	Bug	378.927536
1	Dark	445.741935
2	Dragon	550.531250
3	Electric	443.409091
4	Fairy	413.176471
5	Fighting	416.444444
6	Fire	458.076923
7	Flying	485.000000
8	Ghost	439.562500
9	Grass	421.142857
10	Ground	437.500000
11	Ice	433.458333
12	Normal	401.683673
13	Poison	399.142857
14	Psychic	475.947368
15	Rock	453.750000

	Type 1	Total
16	Steel	487.703704
17	Water	430.455357

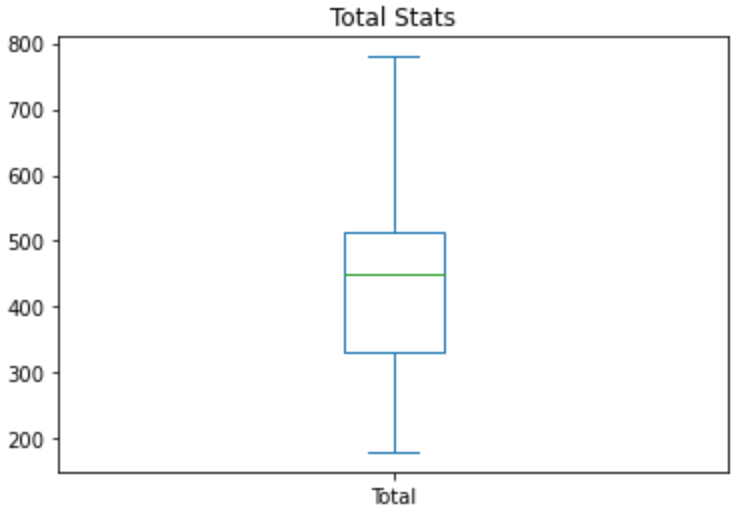
In [7]:

```
# Plotting a bar graph that shows each type and their average stats. It shows dragons as h
# average and bug as having the lowest.
bargraph = dfbar.plot.bar(x = 'Type 1', y = 'Total', fontsize='9')
```



In [9]:

```
# Plotting a boxplot to show the range of pokemons total stats
boxplot = df['Total'].plot(kind='box', title='Total Stats')
plt.show()
```



In [11]:

```
# Creating a new dataframe of the average different stats of pokemon from each generation
bivariate = df.groupby('Generation').mean()[['HP', 'Attack', 'Defense', 'Sp. Atk', 'Sp. De
bivariate
```

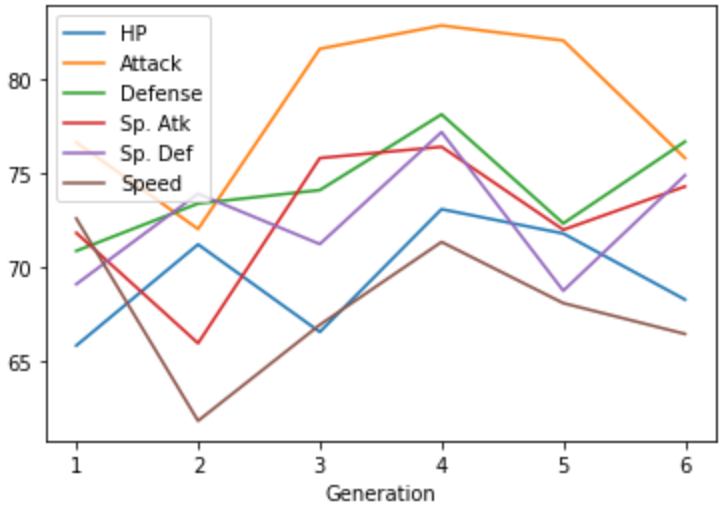
Out[11]:

Generation	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed
1	65.819277	76.638554	70.861446	71.819277	69.090361	72.584337
2	71.207547	72.028302	73.386792	65.943396	73.905660	61.811321

Generation	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed
3	66.543750	81.625000	74.100000	75.806250	71.225000	66.925000
4	73.082645	82.867769	78.132231	76.404959	77.190083	71.338843
5	71.787879	82.066667	72.327273	71.987879	68.739394	68.078788
6	68.268293	75.804878	76.682927	74.292683	74.890244	66.439024

In [13]:
```
# Plotting a line graph showing the averages of each stat over sic generations of pokemon
bivariate.plot.line()
```

Out[13]:
```
<AxesSubplot:xlabel='Generation'>
```

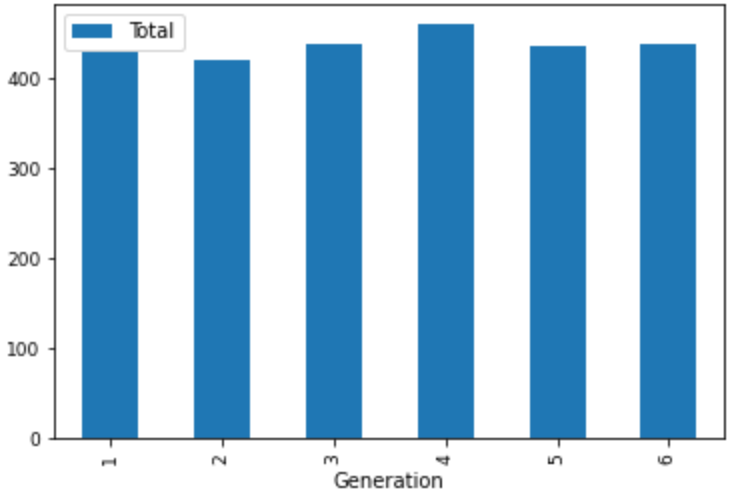


In [14]:
```
# Creating a new data frame of the total stats for pokemon of each generation
dfbar2 = df.groupby('Generation', as_index=False)['Total'].mean()
dfbar2
```

Out[14]:

	Generation	Total
0	1	426.813253
1	2	418.283019
2	3	436.225000
3	4	459.016529
4	5	434.987879
5	6	436.378049

In [15]:
```
# Plotting a bar graph of the total stats of pokemon from each generation.
bargraph2 = dfbar2.plot.bar(x = 'Generation', y = 'Total', fontsize='9')
```



In [16]:
```
# Conclusion
# The questions we chose to answer were which pokemon type tends to have the best stats ar
# generation tends to have the best stats. Our bar graphs were able to show us that on ave
# tend to be the strongest and bug pokemon tend to be the weakest. They also showed us tha
# had the strongest pokemon while generation two had the weakest or lowest average stats.
# the sole determinate on whether a pokemon is good or bad of course, with type combos and
# playing an important role, but this was still interesting to see!
```

In [ ]:

In [ ]: