Pokemon Visualization

In this assignment, we will answer pokemon questions using graphs and charts!

In [3]:
```
#importing the dataset
import pandas as pd

pokemon=pd.read_csv("C:\\Users\\sujoydutta\\Downloads\\pokemon.csv")
pokemon.head()
```

Out[3]:

	abilities	against_bug	against_dark	against_dragon	against_electric	against_fairy	against_fight	against_fi
0	['Overgrow', 'Chlorophyll']	1.0	1.0	1.0	0.5	0.5	0.5	2
1	['Overgrow', 'Chlorophyll']	1.0	1.0	1.0	0.5	0.5	0.5	2
2	['Overgrow', 'Chlorophyll']	1.0	1.0	1.0	0.5	0.5	0.5	2
3	['Blaze', 'Solar Power']	0.5	1.0	1.0	1.0	0.5	1.0	0
4	['Blaze', 'Solar Power']	0.5	1.0	1.0	1.0	0.5	1.0	0

5 rows × 41 columns

In [4]:
```
#examining the dataset
pokemon.info()
```

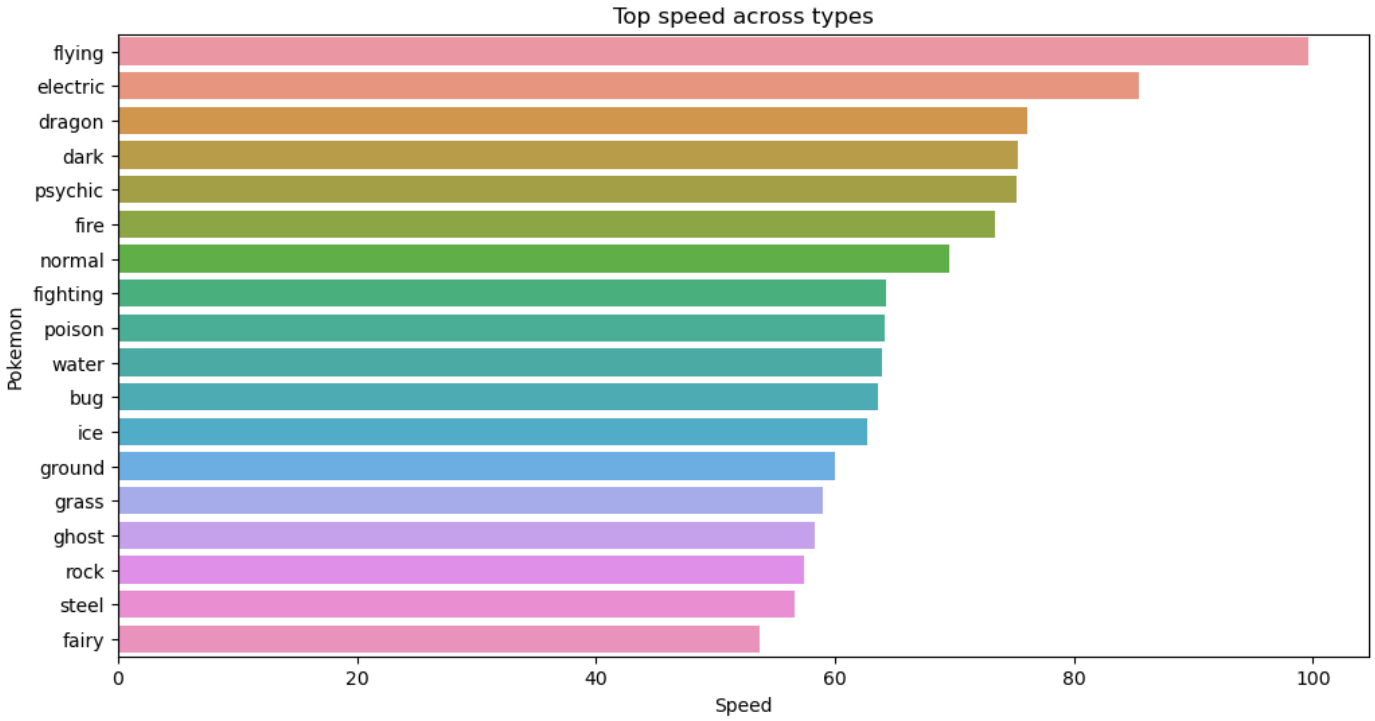
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 801 entries, 0 to 800
Data columns (total 41 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   abilities          801 non-null    object
 1   against_bug        801 non-null    float64
 2   against_dark       801 non-null    float64
 3   against_dragon     801 non-null    float64
 4   against_electric   801 non-null    float64
 5   against_fairy      801 non-null    float64
 6   against_fight      801 non-null    float64
 7   against_fire       801 non-null    float64
 8   against_flying     801 non-null    float64
 9   against_ghost      801 non-null    float64
 10  against_grass      801 non-null    float64
 11  against_ground     801 non-null    float64
 12  against_ice        801 non-null    float64
 13  against_normal     801 non-null    float64
 14  against_poison     801 non-null    float64
 15  against_psychic    801 non-null    float64
 16  against_rock       801 non-null    float64
 17  against_steel      801 non-null    float64
 18  against_water      801 non-null    float64
 19  attack             801 non-null    int64
 20  base_egg_steps     801 non-null    int64
 21  base_happiness     801 non-null    int64
```

```
 22  base_total          801 non-null    int64
 23  capture_rate        801 non-null    object
 24  classfication       801 non-null    object
 25  defense             801 non-null    int64
 26  experience_growth   801 non-null    int64
 27  height_m            781 non-null    float64
 28  hp                  801 non-null    int64
 29  japanese_name       801 non-null    object
 30  name                801 non-null    object
 31  percentage_male     703 non-null    float64
 32  pokedex_number      801 non-null    int64
 33  sp_attack           801 non-null    int64
 34  sp_defense          801 non-null    int64
 35  speed               801 non-null    int64
 36  type1               801 non-null    object
 37  type2               417 non-null    object
 38  weight_kg           781 non-null    float64
 39  generation          801 non-null    int64
 40  is_legendary        801 non-null    int64
dtypes: float64(21), int64(13), object(7)
memory usage: 256.7+ KB
```

In [6]:
```
# Calculating the mean speed across type
typespeed = pokemon.groupby('type1')['speed'].mean().sort_values(ascending=False)
typespeed
```

Out[6]:
```
type1
flying      99.666667
electric    85.410256
dragon      76.111111
dark        75.310345
psychic     75.150943
fire        73.346154
normal      69.533333
fighting    64.285714
poison      64.187500
water       63.921053
bug         63.569444
ice         62.739130
ground      59.968750
grass       59.025641
ghost       58.333333
rock        57.422222
steel       56.583333
fairy       53.666667
Name: speed, dtype: float64
```

In [10]:
```
# Plotting the top mean speed across type
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(12, 6))
sns.barplot(x=typespeed.values, y=typespeed.index)
plt.title('Top speed across types')
plt.xlabel('Speed')
plt.ylabel('Pokemon')
plt.show()
```

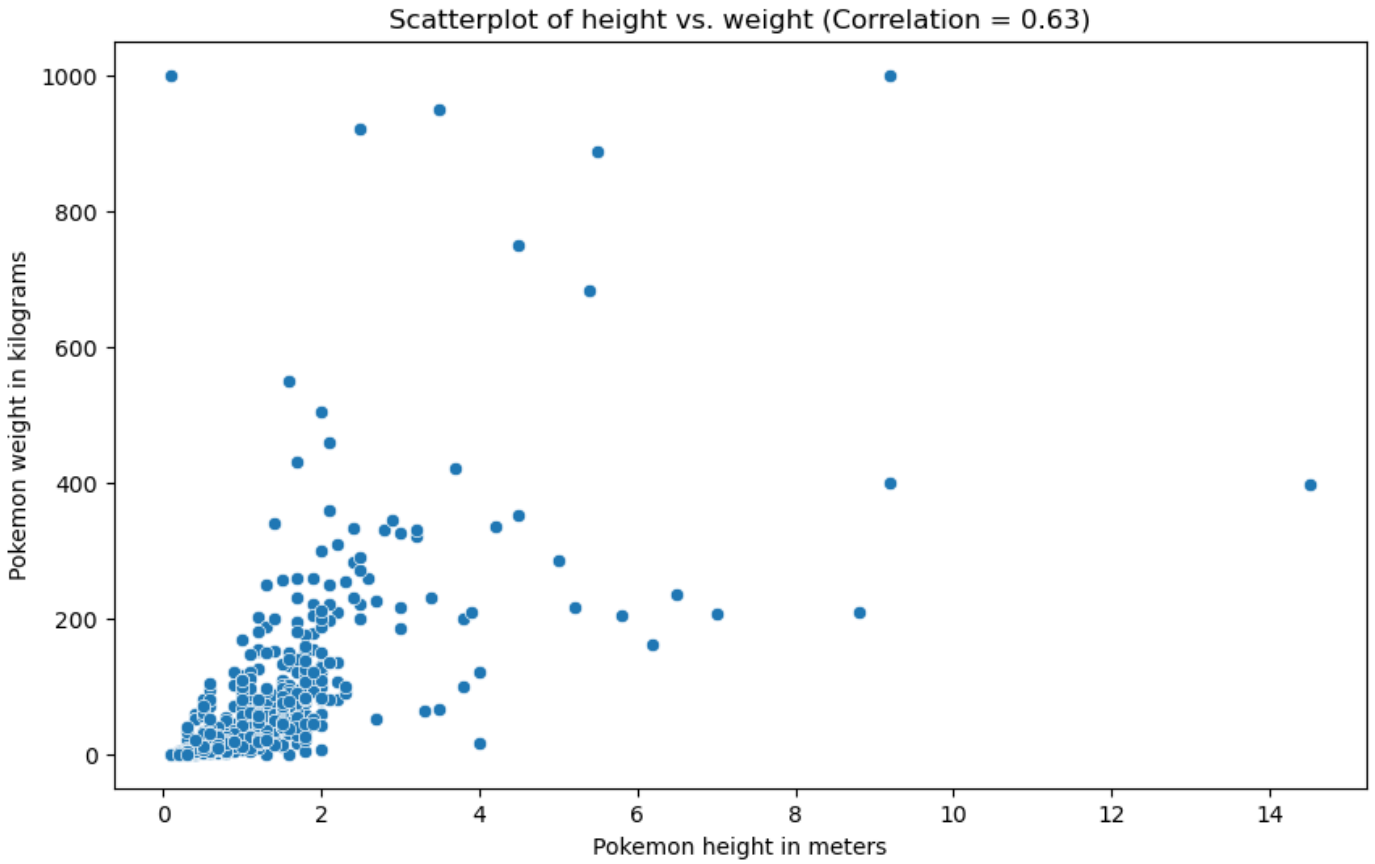


In [12]:
```
# Seeing correlation between height  and weight

correlation = pokemon[['height_m', 'weight_kg']].corr().iloc[0, 1]
correlation
```

Out[12]:
```
0.6265511437853188
```

In [13]:
```
# Building a scatterplot to visualize the relationship between height  and weight
plt.figure(figsize=(10, 6))
sns.scatterplot(x='height_m', y='weight_kg', data=pokemon)
plt.title(f'Scatterplot of height vs. weight (Correlation = {correlation:.2f})')
plt.xlabel('Pokemon height in meters')
plt.ylabel('Pokemon weight in kilograms')
plt.show()
```



In [14]:
```
pokemon.columns
```

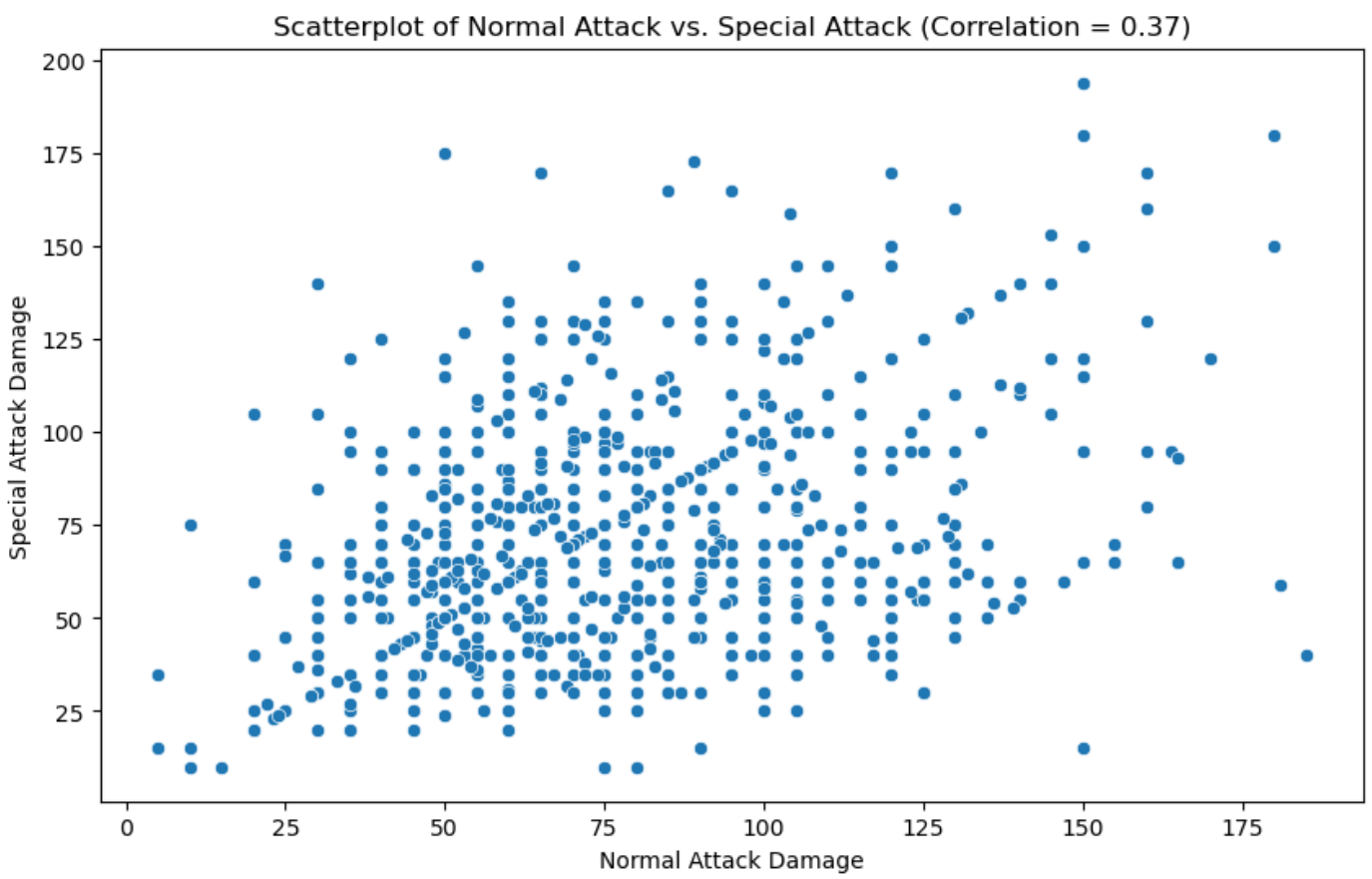
Out[14]:
```
Index(['abilities', 'against_bug', 'against_dark', 'against_dragon',
       'against_electric', 'against_fairy', 'against_fight', 'against_fire',
       'against_flying', 'against_ghost', 'against_grass', 'against_ground',
       'against_ice', 'against_normal', 'against_poison', 'against_psychic',
       'against_rock', 'against_steel', 'against_water', 'attack',
       'base_egg_steps', 'base_happiness', 'base_total', 'capture_rate',
       'classfication', 'defense', 'experience_growth', 'height_m', 'hp',
       'japanese_name', 'name', 'percentage_male', 'pokedex_number',
       'sp_attack', 'sp_defense', 'speed', 'type1', 'type2', 'weight_kg',
       'generation', 'is_legendary'],
      dtype='object')
```

In [15]:
```
# Seeing correlation between attack  and sp attack

correlation = pokemon[['attack', 'sp_attack']].corr().iloc[0, 1]
correlation
```

Out[15]:
```
0.3681539995496
```

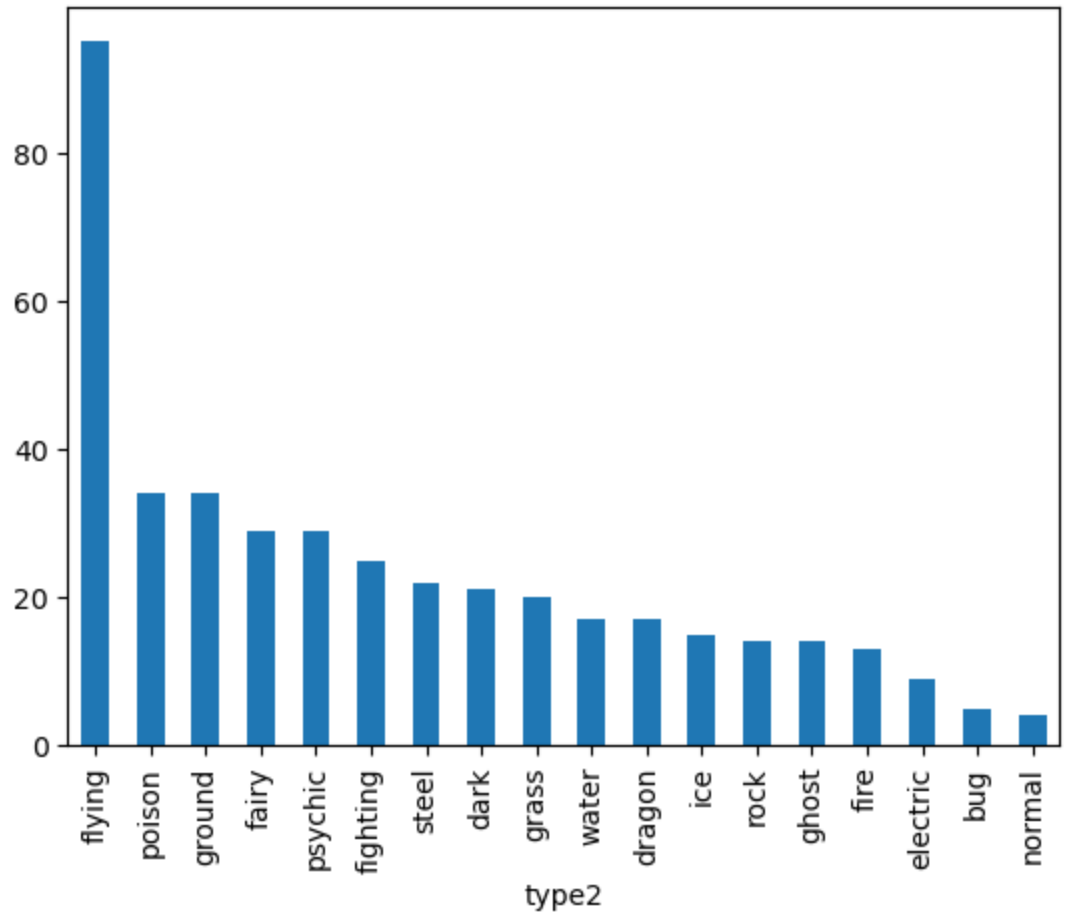
In [16]:
```
# Building a scatterplot to visualize the relationship between attack  and special attac
plt.figure(figsize=(10, 6))
sns.scatterplot(x='attack', y='sp_attack', data=pokemon)
plt.title(f'Scatterplot of Normal Attack vs. Special Attack (Correlation = {correlation:
plt.xlabel('Normal Attack Damage')
plt.ylabel('Special Attack Damage')
plt.show()
```



In [21]:
```
#seeing most common pokemon types

pokemon['type2'].value_counts().plot(kind="bar")


plt.show()
```



In [23]:
```
#pokemon weight summary stats
pokemon.weight_kg.describe()
```

Out[23]:
```
count    781.000000
mean      61.378105
std      109.354766
min        0.100000
25%        9.000000
50%       27.300000
75%       64.800000
max      999.900000
Name: weight_kg, dtype: float64
```