

Donovan Orozco

Pokemones

```
In [1]: import pandas as pd
```

```
In [4]: df = pd.read_csv("pokemon.csv")  
df.head()
```

Out [4]:

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

5 rows × 41 columns

```
In [5]: columna = list(df.columns)
columna
```

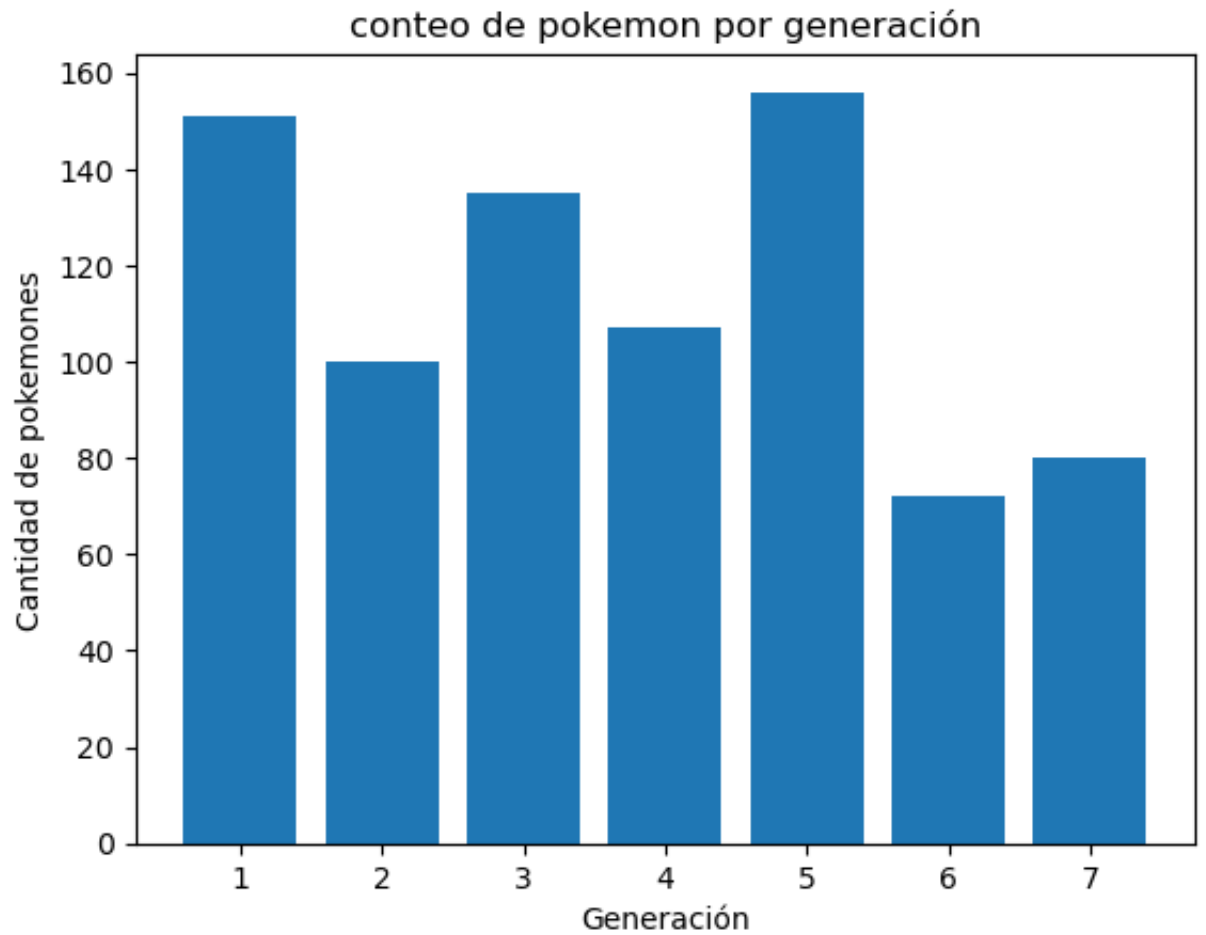
```
Out[5]: ['abilities',
         'against_bug',
         'against_dark',
         'against_dragon',
         'against_electric',
         'against_fairy',
         'against_fight',
         'against_fire',
         'against_flying',
         'against_ghost',
         'against_grass',
         'against_ground',
         'against_ice',
         'against_normal',
         'against_poison',
         'against_psychic',
         'against_rock',
         'against_steel',
         'against_water',
         'attack',
         'base_egg_steps',
         'base_happiness',
         'base_total',
         'capture_rate',
         'classification',
         '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']
```

Realizar una visualización por ejercicio que respondan cada cuestionamiento:

1. ¿Cuántos nuevos pokemon hay por generación?
2. ¿Cuántos pokemon legendarios hay por generación?
3. Visualización por peso y altura.
4. ¿Cuál es el tipo más común de pokemon?
5. ¿Cuáles son las combinaciones de tipo más comunes?
6. ¿Qué pokemon son los mejores en términos de sus stats? (attack, defense, sp_attack, sp_defense, speed, hp)
7. ¿Cómo cambiaron los stats en promedio conforme avanzaban las generaciones?
8. ¿Un pokemon es más fuerte entre más difícil es de capturar?

```
In [6]: import matplotlib.pyplot as plt
#Ejercicio1
df2 = df [['name', 'generation'] ].groupby("generation").count()

plt.bar(df2.index, df2['name'])
plt.title("conteo de pokemon por generación")
plt.xlabel("Generación")
plt.ylabel("Cantidad de pokemones")
plt.show()
```

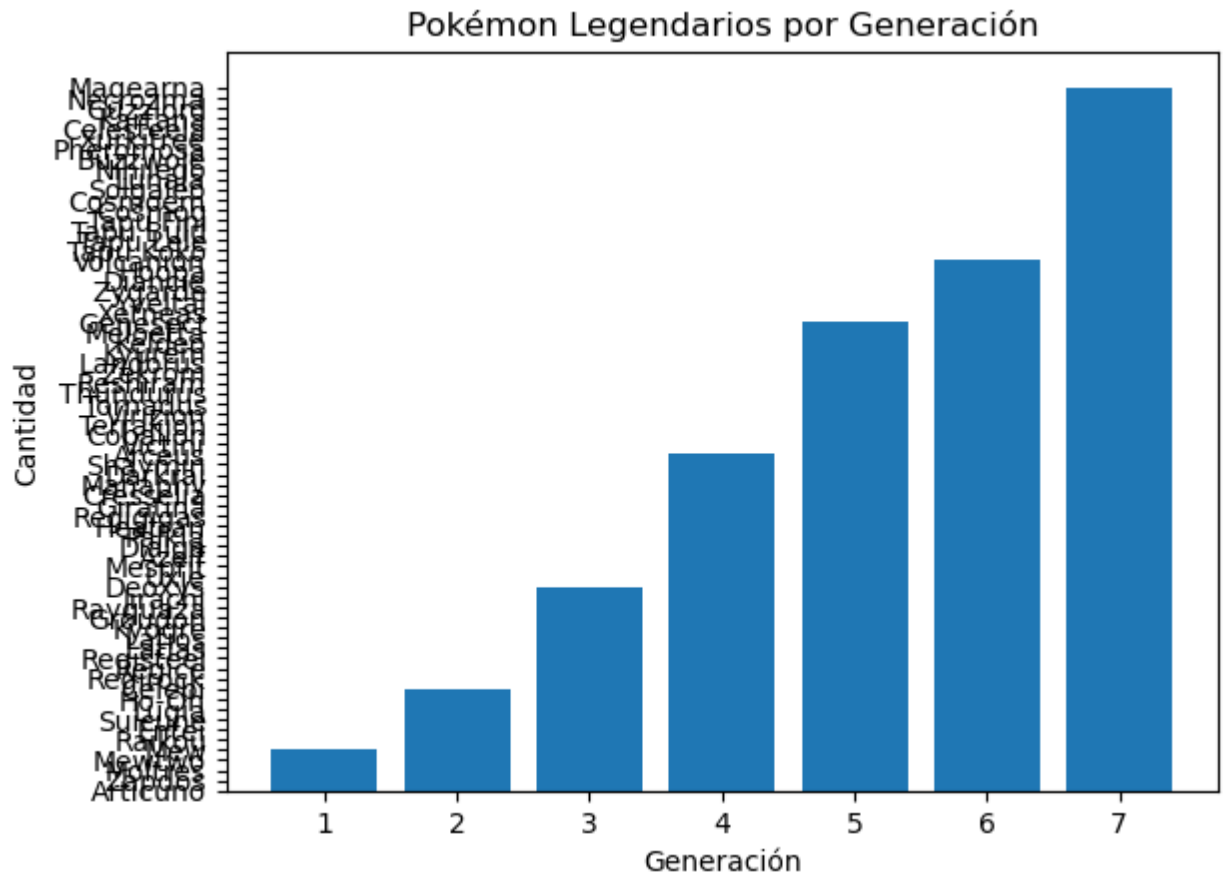


In [9]:

```
# 2
df3 = df[df["is_legendary"] == 1][["name", "generation", "is_legendary"]]

plt.bar(df3["generation"], df3["name"])
plt.title("Pokémon Legendarios por Generación")
plt.xlabel("Generación")
plt.ylabel("Cantidad")
plt.show()

print(df3)
```

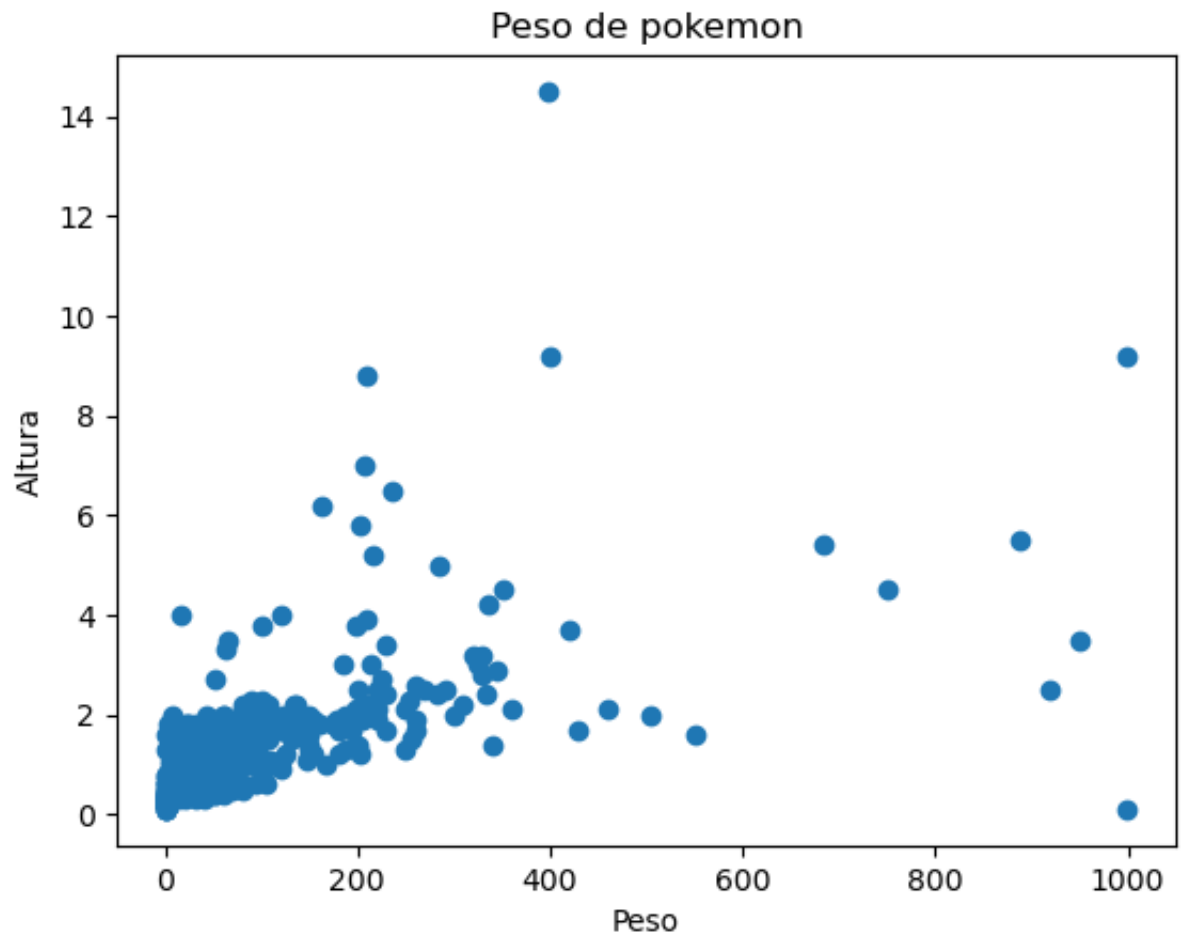


	name	generation	is_legendary
143	Articuno	1	1
144	Zapdos	1	1
145	Moltres	1	1
149	Mewtwo	1	1
150	Mew	1	1
..
796	Celesteela	7	1
797	Kartana	7	1
798	Guzzlord	7	1
799	Necrozma	7	1
800	Magearna	7	1

[70 rows x 3 columns]

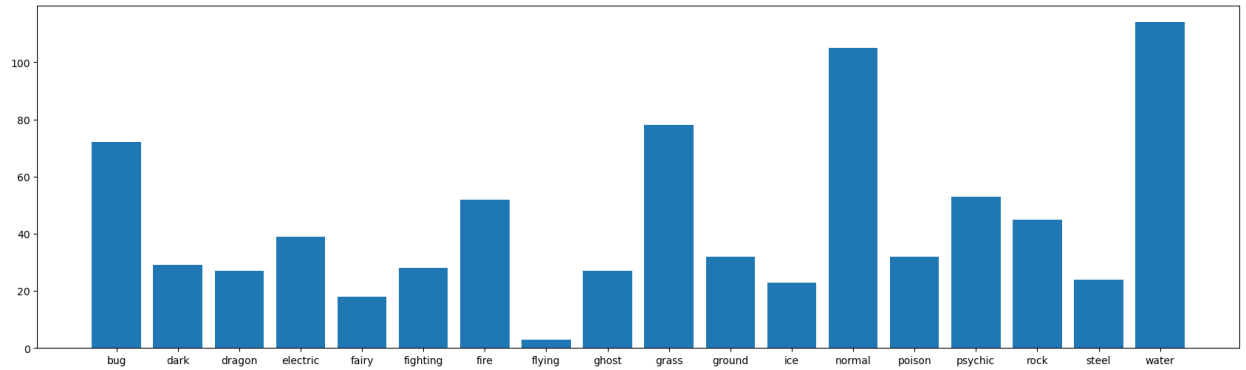
```
In [10]: #3
df4 = df[["weight_kg", "height_m"]]

plt.scatter(df4["weight_kg"], df4["height_m"])
plt.title("Peso de pokemon")
plt.xlabel("Peso")
plt.ylabel("Altura")
plt.show()
```



```
In [15]: #4
df5 = df[["name", "type1", "generation"]].groupby("type1").count()
df5

plt.figure(figsize=(21,6))
plt.bar(df5.index, df5["name"])
plt.show()
```



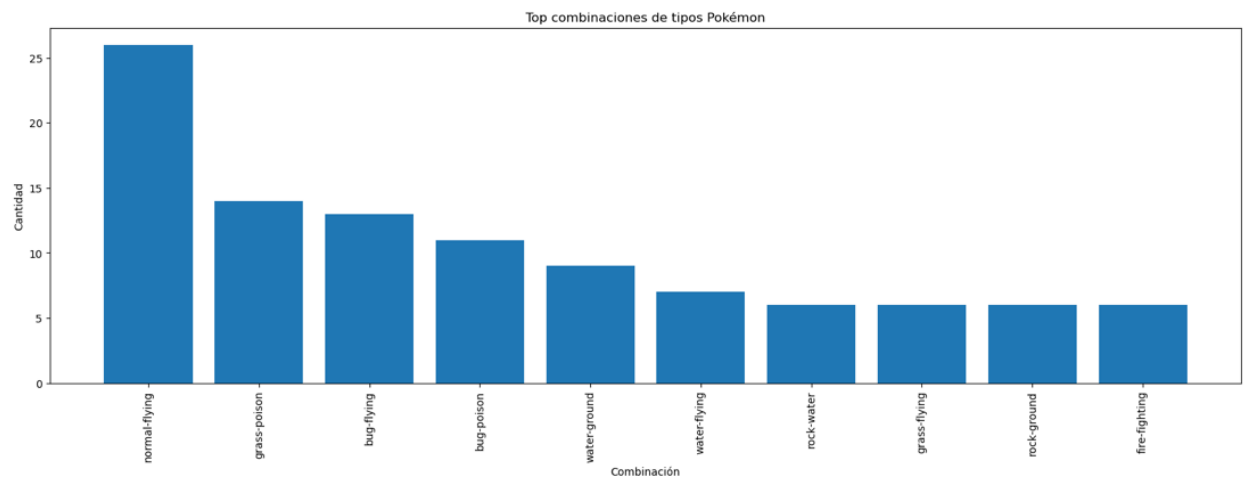
```
In [17]: #5
df6 = df[~df["type2"].isna()]
df6.loc[:, "type"] = df6.loc[:, "type1"] + "-" + df6.loc[:, "type2"]
df7 = df6[["name", "type"]].groupby("type").count().sort_values("name")

plt.figure(figsize=(20,6))
plt.bar(df7.index[:10], df7["name"][:10])
plt.title("Top combinaciones de tipos Pokémon")
plt.xlabel("Combinación")
plt.ylabel("Cantidad")
plt.xticks(rotation=90)
plt.show()
```

/var/folders/8v/tvt7vss914x18vfsvmp23mnr0000gn/T/ipykernel_83935/2496435743.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

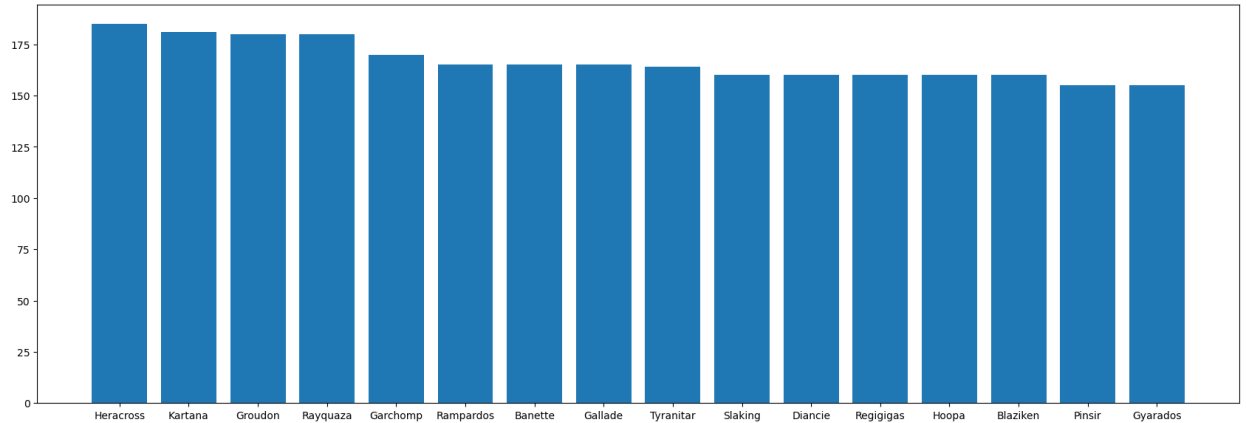
```
df6.loc[:, "type"] = df6.loc[:, "type1"] + "-" + df6.loc[:, "type2"]
]
```




```
In [18]: #6
df8 = df[["name", "attack"]].sort_values("attack", ascending = False)
plt.figure(figsize = (21,7))

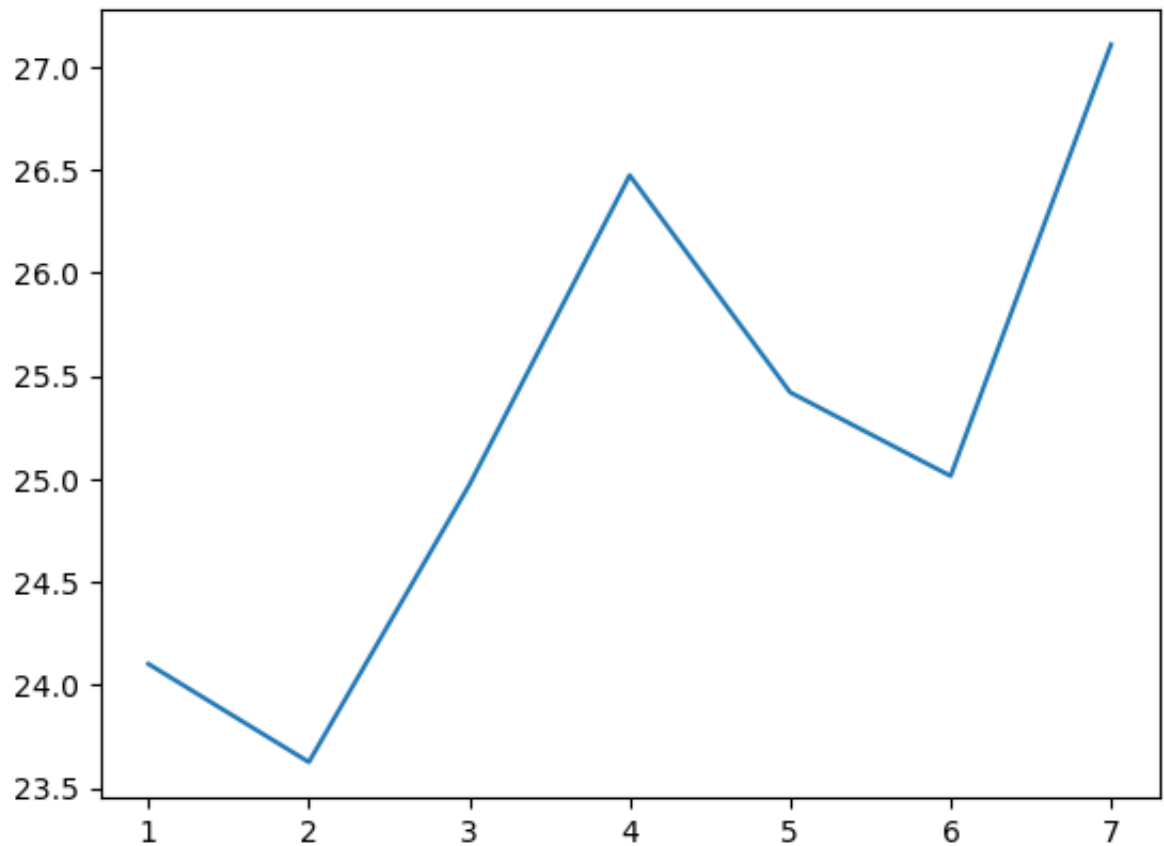
plt.bar(df8["name"][:16],df8["attack"][:16])
plt.show
```

```
Out[18]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
In [20]: #7
def stat_promedio(fila):
    return (fila["attack"] + fila["defense"]) / 6
df["stat_promedio"] = df.apply(stat_promedio, axis=1)
```

```
In [22]: df2 = df[["stat_promedio", "generation"]].groupby("generation").mean(  
plt.plot( df2.index, df2["stat_promedio"])  
plt.show()
```



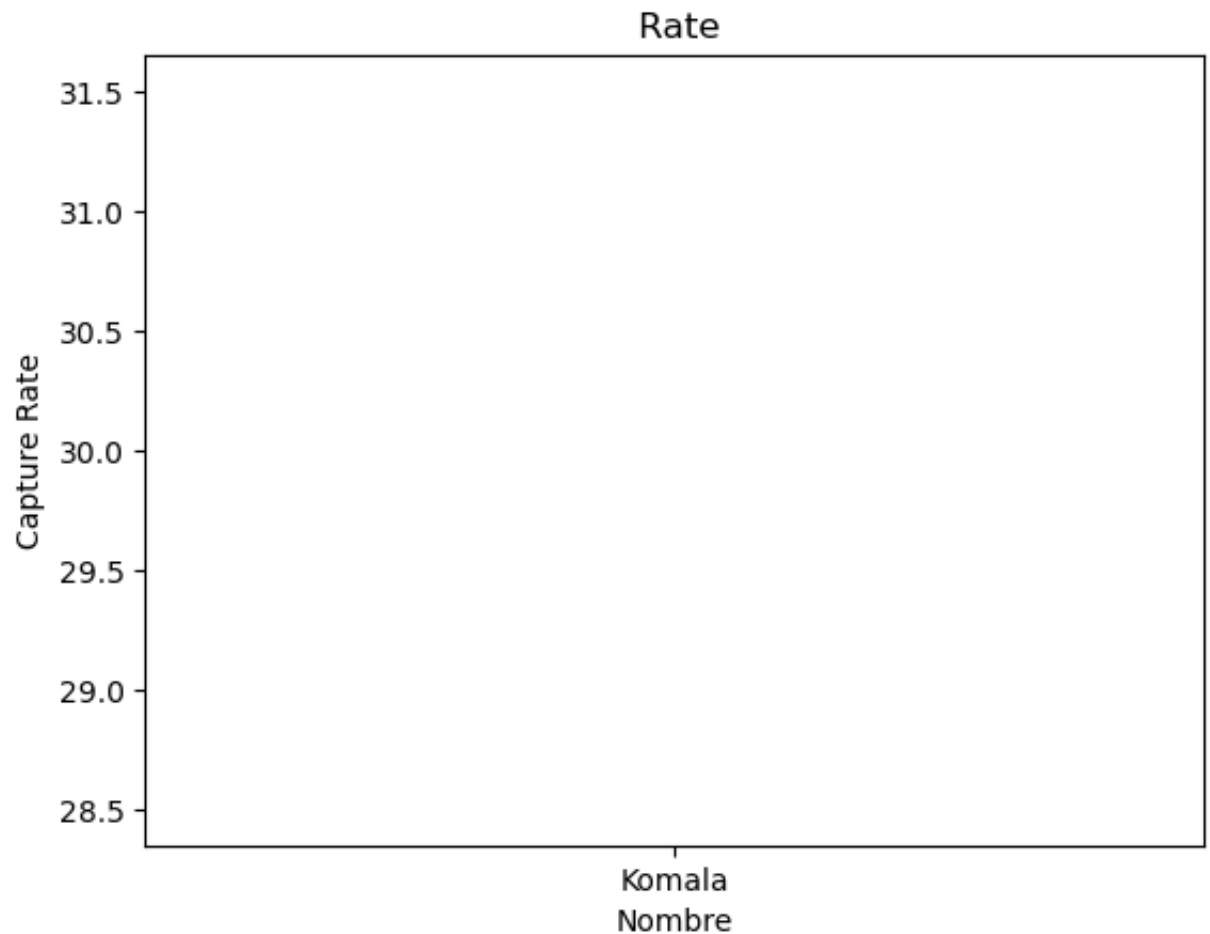
```
In [23]: #8
df["capture_rate"].values
```

'120', '45', '150', '80', '255', '45', '30', '30', '45', '45',
 '190', '75', '205', '155', '255', '90', '45', '45', '45', '45',
 ',
 '255', '60', '45', '200', '225', '45', '190', '90', '200', '45',
 ',
 '30', '125', '190', '75', '255', '120', '45', '255', '60', '60',
 ',
 '25', '225', '45', '45', '45', '3', '3', '3', '3', '3', '3', '
 3',
 '3', '3', '3', '45', '3', '3', '45', '45', '45', '45', '45', '
 45',
 '45', '45', '45', '255', '120', '45', '255', '127', '255', '45',
 ',
 '235', '120', '45', '255', '75', '45', '45', '45', '45', '120',
 ',
 '45', '45', '120', '45', '200', '190', '75', '190', '75', '190',
 ',
 '75', '45', '125', '60', '190', '60', '45', '30', '190', '75',
 '120', '225', '60', '255', '90', '255', '145', '130', '30', '1
 00',
 '45', '45', '45', '150', '75', '45', '140', '60', '120', '45',

```
In [24]: df6 =df[df["capture_rate"] == '30 (Meteorite)255 (Core)']["name"]
df.at[774, "capture_rate"] =30
```

```
In [26]: datos = df.loc[774][["name", "capture_rate"]]  
  
print(datos)  
  
plt.plot(datos["name"], datos["capture_rate"])  
plt.title(" Rate")  
plt.xlabel("Nombre")  
plt.ylabel("Capture Rate")  
plt.show()
```

```
name          Komala  
capture_rate    30  
Name: 774, dtype: object
```



```
In [ ]:
```

