

Pokemon Dataset

Sebastian Méndez

2023-11-20

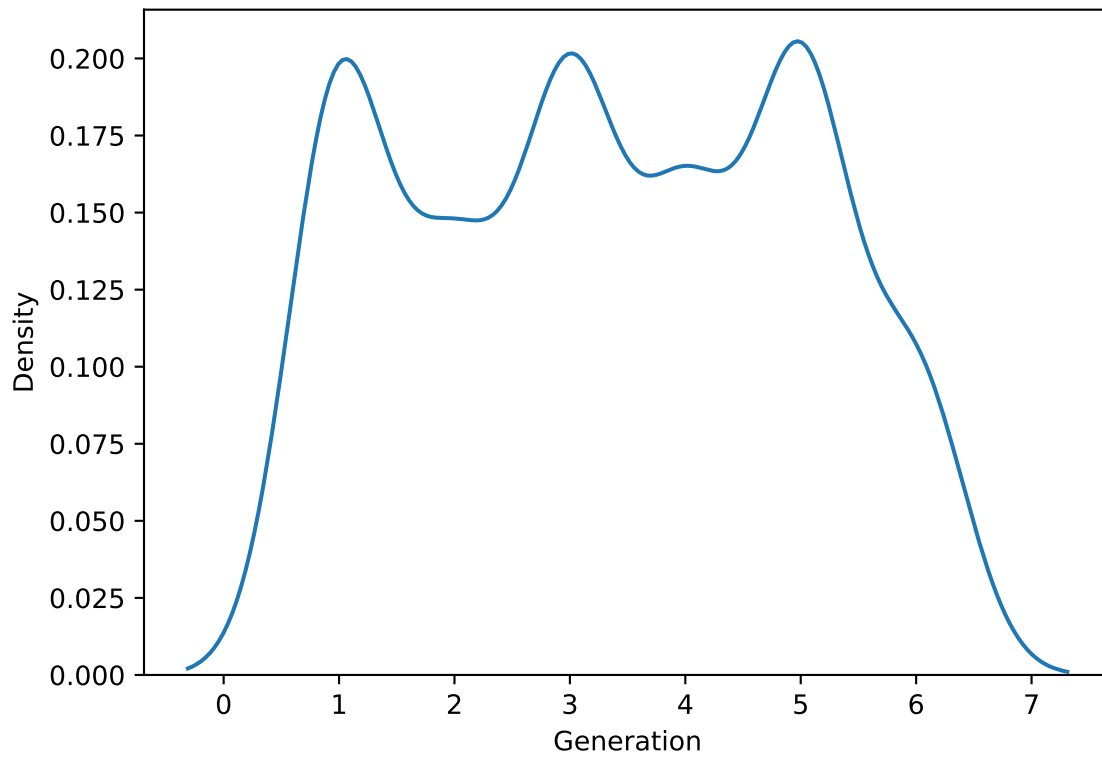
¡Pokemon Dataset!

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib
matplotlib.use('pdf')
Pokedex = pd.read_csv("C:/Users/57301/Desktop/Universidad/Universidad 2023-2 (ECCI)/Electiva/Final assignm
```

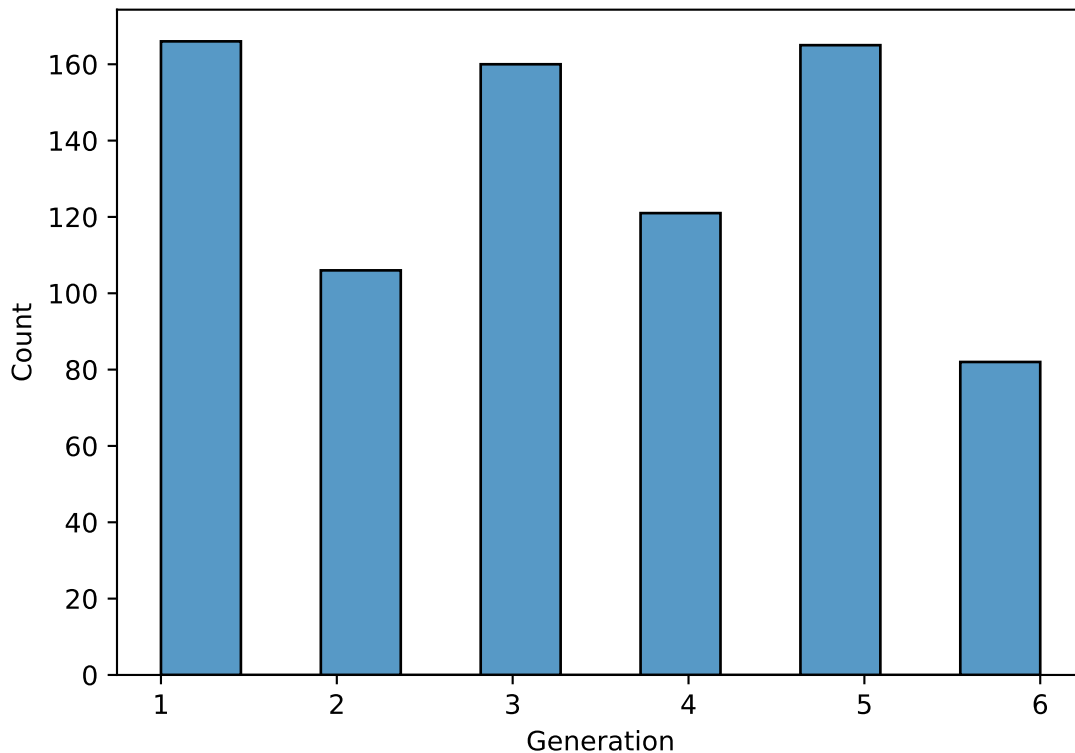
¿What is the distribution of the generation?

Para ver la distribución de la generación, podemos usar dos de los mejores diagramas para expresarlo, es decir, el histograma y el diagrama de densidad, que son bastante similares entre si; a continuación, podremos ver ambos.

```
sns.kdeplot(Pokedex['Generation'])
plt.show()
```



```
sns.histplot(Pokedex['Generation'], kde=False)
plt.show()
```



Como se puede observar con el histograma que las generaciones 1, 3 y 5 son aquellas con mayor cantidad de pokémones, teniendo así alrededor de 166, 159 y 164, mientras que la generación que cuenta con una menor cantidad de Pokémons es la 6, con alrededor de 81 pokémones.

Posteriormente, con el diagrama de densidad se confirman los datos del histograma, mostrándonos así como se concentra más densidad de datos sobre el pico de las generaciones 1, 3 y 5, mientras que disminuye mientras se va acercando a la sexta generación.

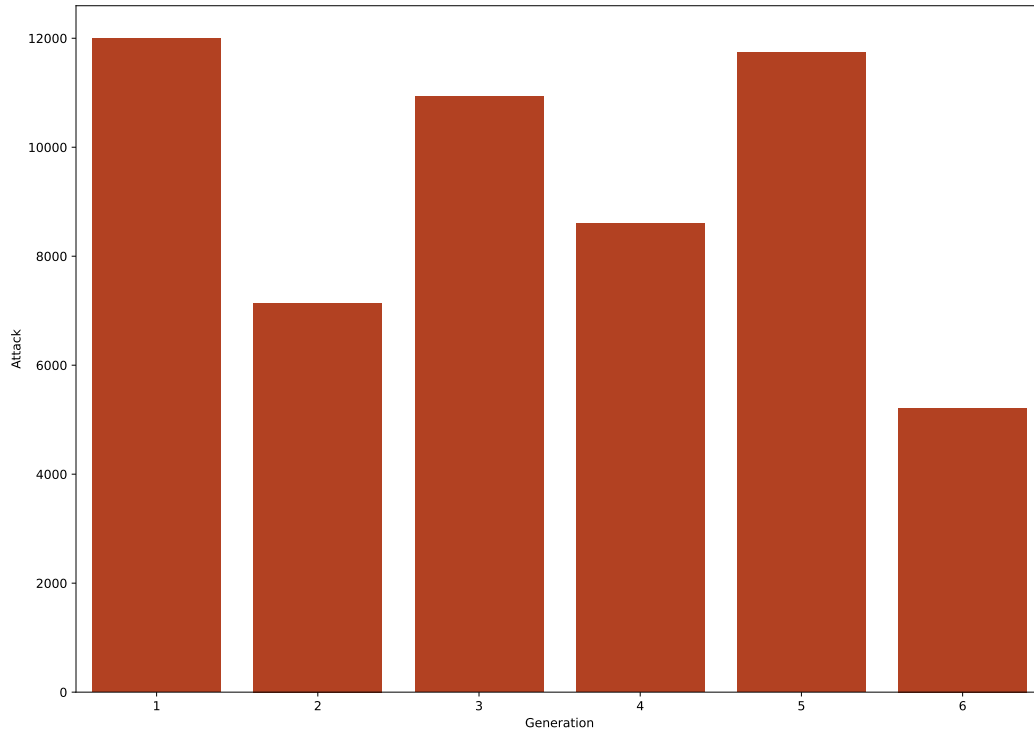
¿In what generation it is more likely to find a non legendary Pokemon with a high attack?

Para la solución a esta pregunta, se usará un diagrama de barras, pero para ello, primero, se creará un subdataset, el cual solo contenga datos de pokémones No Legendarios, lo cual se hace a través de un borrado de filas con condicional, donde la condición será eliminar aquellas filas que en su columna 'Legendary' tengan un valor 'True' como se ve a continuación:

```
Nolegendary = Pokedex.drop(Pokedex[Pokedex['Legendary']==True].index)
plt.figure(figsize=(14, 10))
sns.barplot(
    x='Generation',
    y='Attack',
    data=Nolegendary,
    estimator=sum,
    ci=None,
    color='#CA330A');
```

```
## <string>:1: FutureWarning:
##
## The 'ci' parameter is deprecated. Use 'errorbar=None' for the same effect.
```

```
plt.show()
```



Como se puede observar en la gráfica, en el eje 'x' tenemos la generación y en el eje 'y' tenemos el poder de ataque de los pokemones. Este gráfico nos permite visualizar cual generación es más fuerte si se suma el poder de ataque de todos los pokemones que le pertenece a cada una. Es decir, la generación con más poder de ataque de "No Legendarios" sería la generación 1, seguida de la 5, posteriormente la 3, 4, 2 y 6.

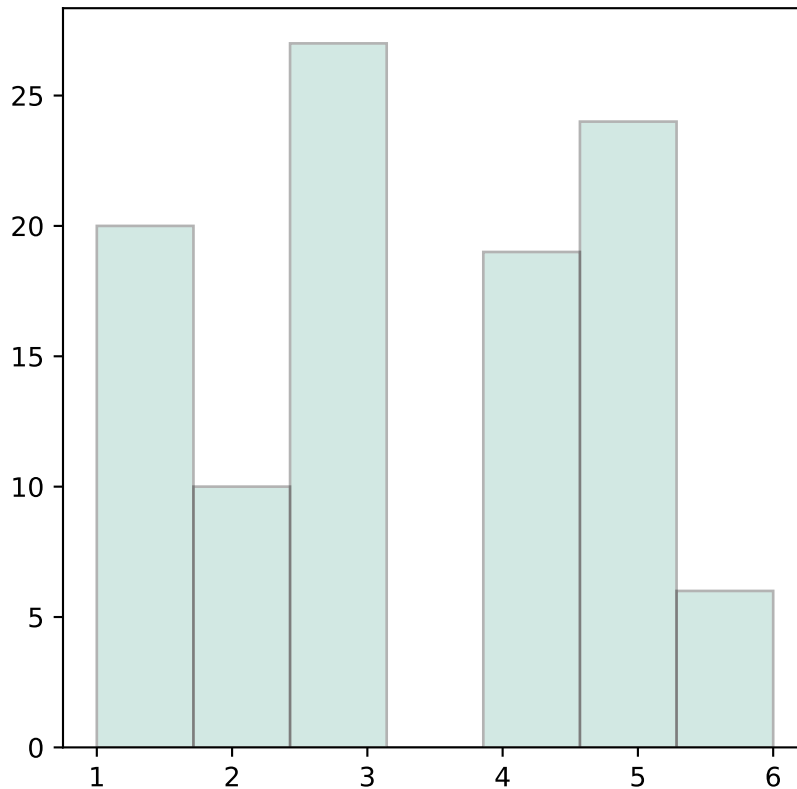
Sin embargo, seamos más específicos; veamos cuantos pokemones tiene cada generación con un alto poder de ataque.

```
print(Pokedex['Attack'].max() - Pokedex['Attack'].mean())
```

```
## 110.99875
```

El método que utilicé para definir si el poder de ataque es fuerte o no, fue sacar el máximo poder de ataque y restarle el promedio de poder de ataque, obteniendo un valor de 110.99, el cual vamos a redondear a 110.

```
StrongNL = Nolegendary.drop(Nolegendary[Nolegendary['Attack']<110].index)
fig, ax = plt.subplots(figsize = (5, 5))
ax.hist(StrongNL['Generation'], bins=7, edgecolor="black", color="#69b3a2", alpha=0.3)
plt.show()
```



Para llegar a este gráfico se creo un subdataset de “NoLegendary”, el cual tenga como condición de borrado a aquellas filas que en la columna de ataque tengan un valor menor a 110, que es el valor obtenido con la operación que se enseña más arriba.

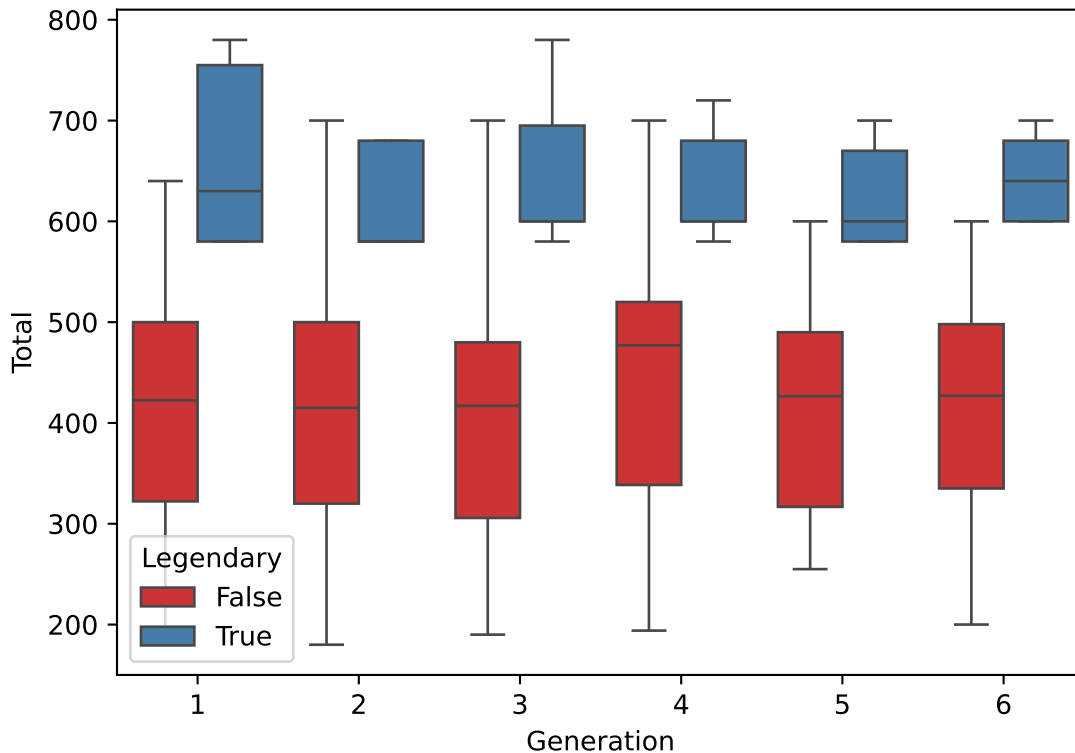
Analizando el subdataset llamado “StrongNL”, el cual fue generado según la explicación del anterior parrafo, se puede observar que la cantidad de pokemones con alto poder de ataque para las generaciones 1, 2, 3, 4, 5 y 6 son 20, 10, 27, 19, 24 y 6 respectivamente. Esto quiere decir que la generación en la cual es más común encontrar pokemones con alto poder de ataque es en la número 3, con 27 pokemones que superan un valor de poder de ataque de 110, mientras que la generación menos común para este parametro es la número 6.

¿How the stats distribution differ if we compare legendary vs non-legendary Pokemon?

Esta pregunta se interpreta como un cambio en la densidad de los valores totales de estadísticas y como varían según si estos son o no son Pokemones legendarios, por lo cual se procederá a usar un boxplot para su evaluación.

```
sns.boxplot(x="Generation",
            y="Total",
            hue="Legendary",
            data=Pokedex,
            palette="Set1",
            width=0.8)

plt.show()
```

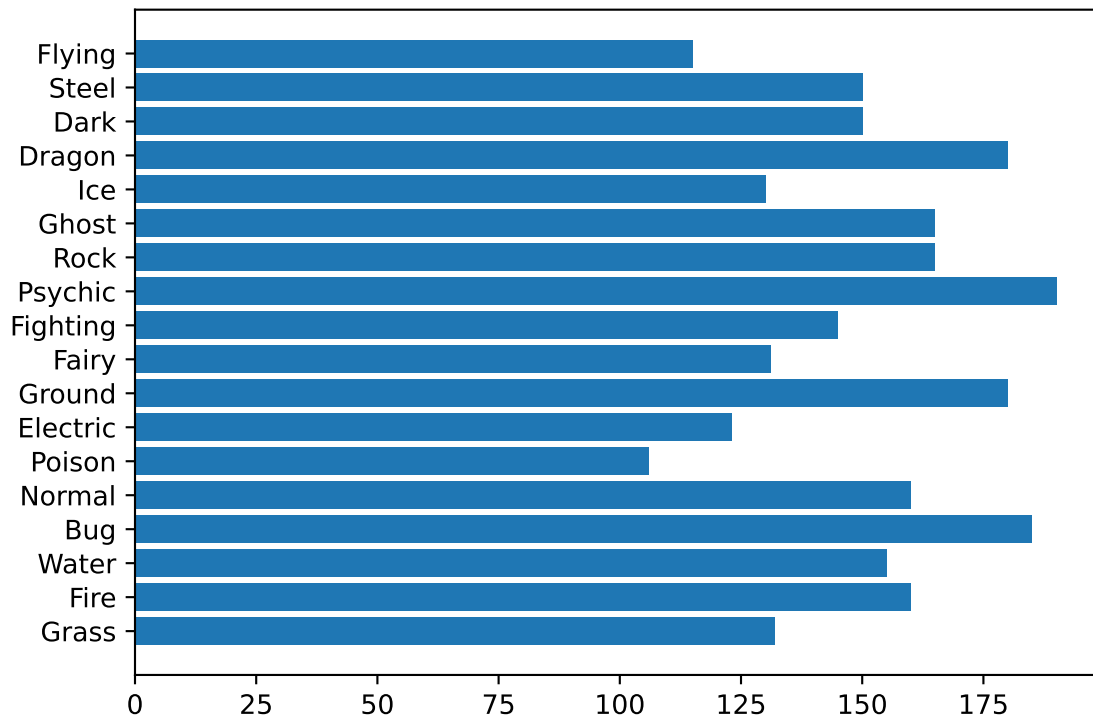


Además de lo pedido por el punto, para tener un análisis más completo de la gráfica, añadí una variable más, la cual me permite analizar generación a generación la distribución de las estadísticas. Sin lugar a duda y como evidentemente se puede interpretar en esta gráfica, la distribución en los datos muestra mayor densidad en los Pokemon legendarios. El boxplot nos muestra como todos los valores máximos de cada generación de Pokemones legendarios es mayor a los valores máximos de cada generación de pokemones no legendarios. Del mismo modo, la mediana de las estadísticas totales de los no legendarios es similar en cada una de las generaciones, pero aún así, superadas por la mediana de cada uno de las generaciones de Pokemones legendarios. Incluso para remarcar la diferencia entre estadísticas, se puede observar que en cada una de las generaciones, el valor mínimo de estadísticas de Pokemones legendarios es superior al 75% de los valores totales de estadística de los Pokemones no legendarios.

¿Is there any pattern between stats?

Se revisa un gráfico de barras, el cual relacionará el tipo principal de Pokemon (eje y) con el poder de ataque del Pokemon (eje x).

```
plt.barh(Pokedex['Type 1'], Pokedex['Attack'])
plt.show()
```



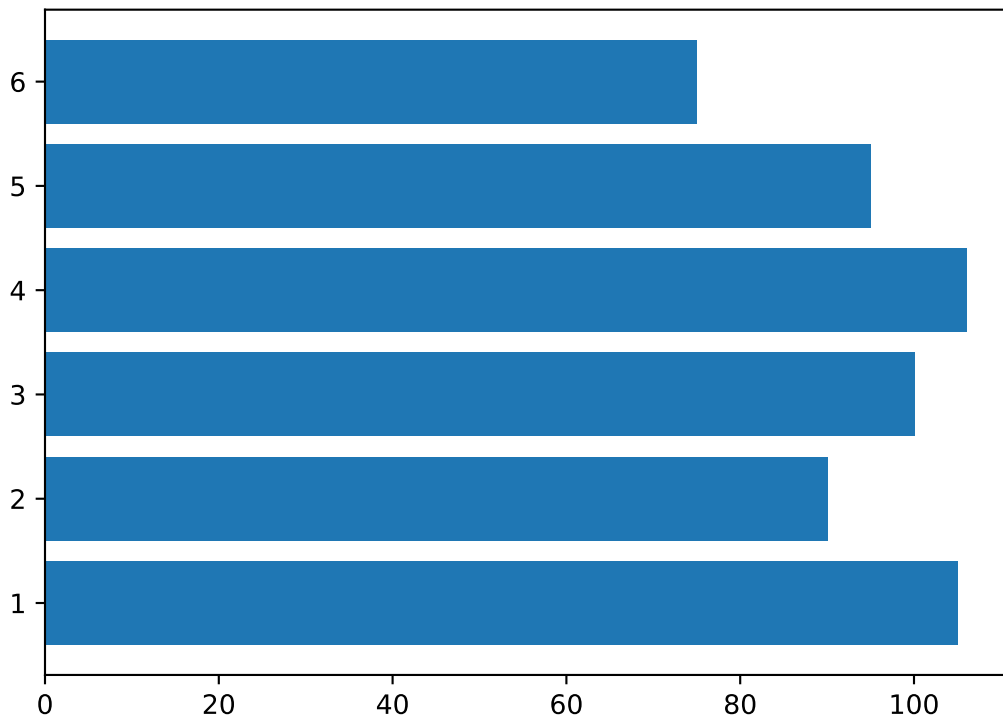
Al realizar una observación se logra percibir que los tipo veneno no tienen ningún Pokémon que supere los 125 de ataque, así que se procede a verificar este dato, generando un dataset que contenga solo pokemones tipo veneno.

```
PoisonPokemon=Pokedex.drop(Pokedex[Pokedex['Type 1']!='Poison'].index)
PoisonPokemon['Attack'].max()
```

```
## 106
```

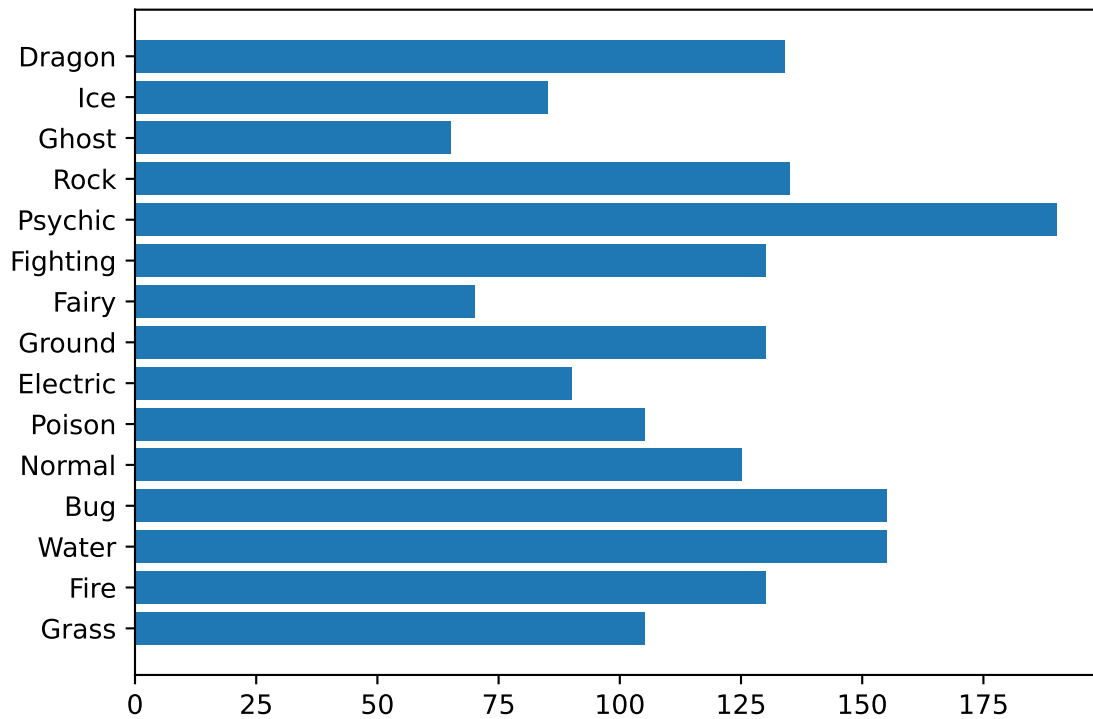
Se observa como el pokemón con más daño dentro de los tipo veneno tiene 106 de daño, un valor mucho menor al de los demás.

```
plt.barh(PoisonPokemon['Generation'], PoisonPokemon['Attack'])
plt.show()
```



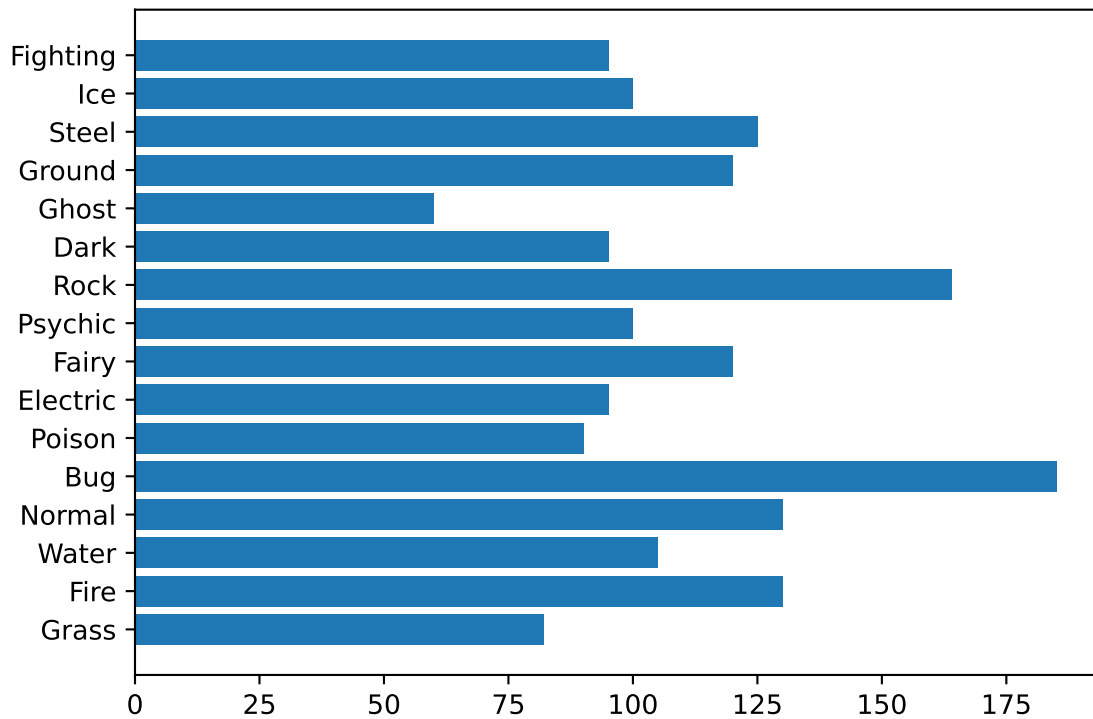
Observando esta gráfica que representa el poder de ataque de los Pokemon tipo veneno en cada una de las generaciones, se ve como en ninguna de ellas se superan los 110 de poder de ataque. A continuación se comparará el poder de ataque de cada una de las generaciones con respecto a su tipo principal.

```
Generation1 = Pokedex.drop(Pokedex[Pokedex['Generation']!=1].index)
plt.barh(Generation1['Type 1'], Generation1['Attack'])
plt.show()
```

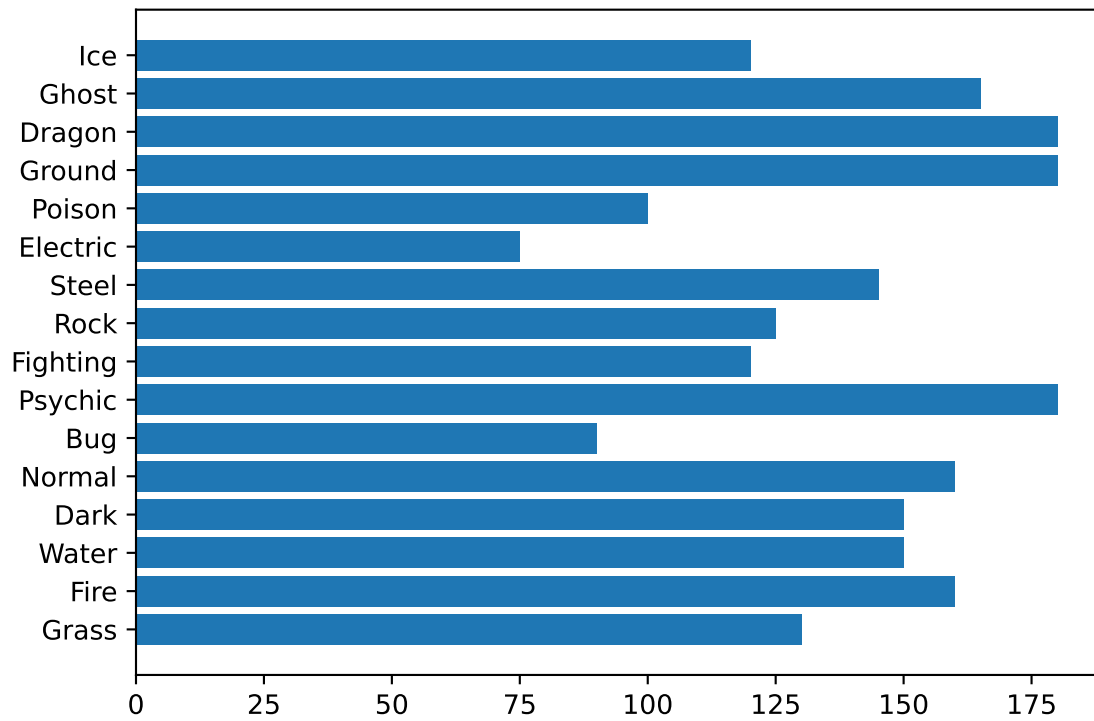
En la primera generación se puede observar que hay muchas muestras de pokémones tipo veneno, sin embargo, pese a su bajo poder de ataque, en la generación 1, estos no son los más débiles.

```
Generation2 = Pokedex.drop(Pokedex[Pokedex['Generation']!=2].index)
plt.barh(Generation2['Type 1'], Generation2['Attack'])
plt.show()
```



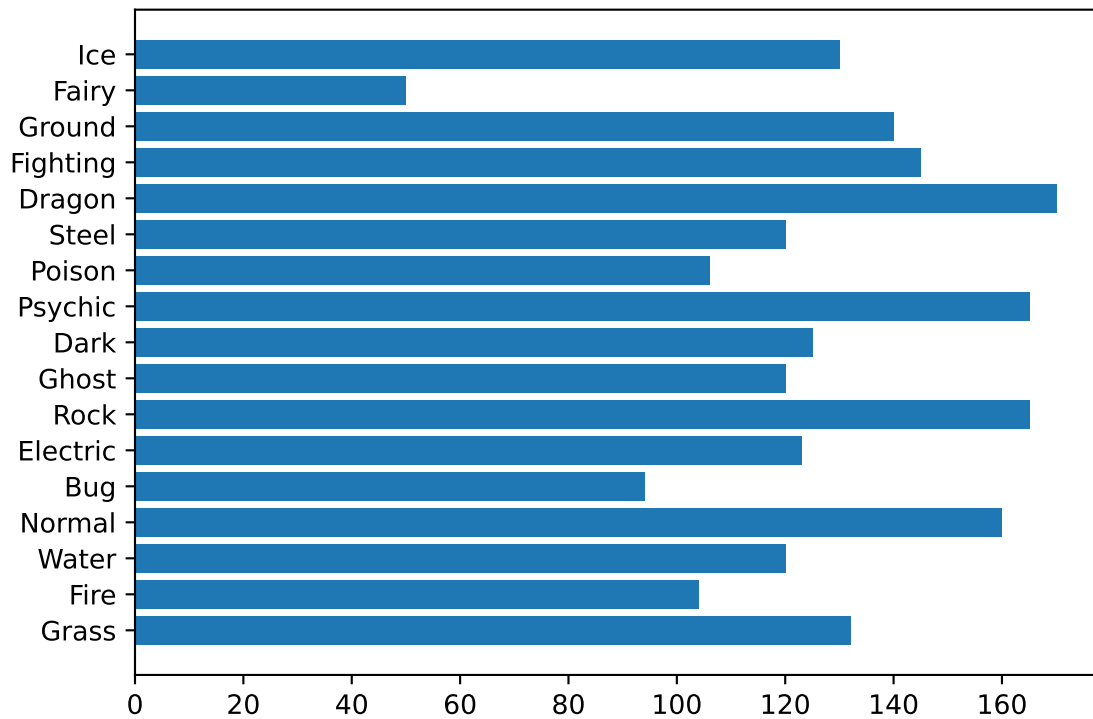
En la generación uno, los tipo veneno se encontraban por encima de dos tipos más de pokemones, en esta segunda generación, solo se encuentran por encima de los pokemones tipo “Grass”, teniendo una sola muestra de Pokémon tipo veneno para la misma.

```
Generation3 = Pokedex.drop(Pokedex[Pokedex['Generation']!=3].index)
plt.barh(Generation3['Type 1'], Generation3['Attack'])
plt.show()
```



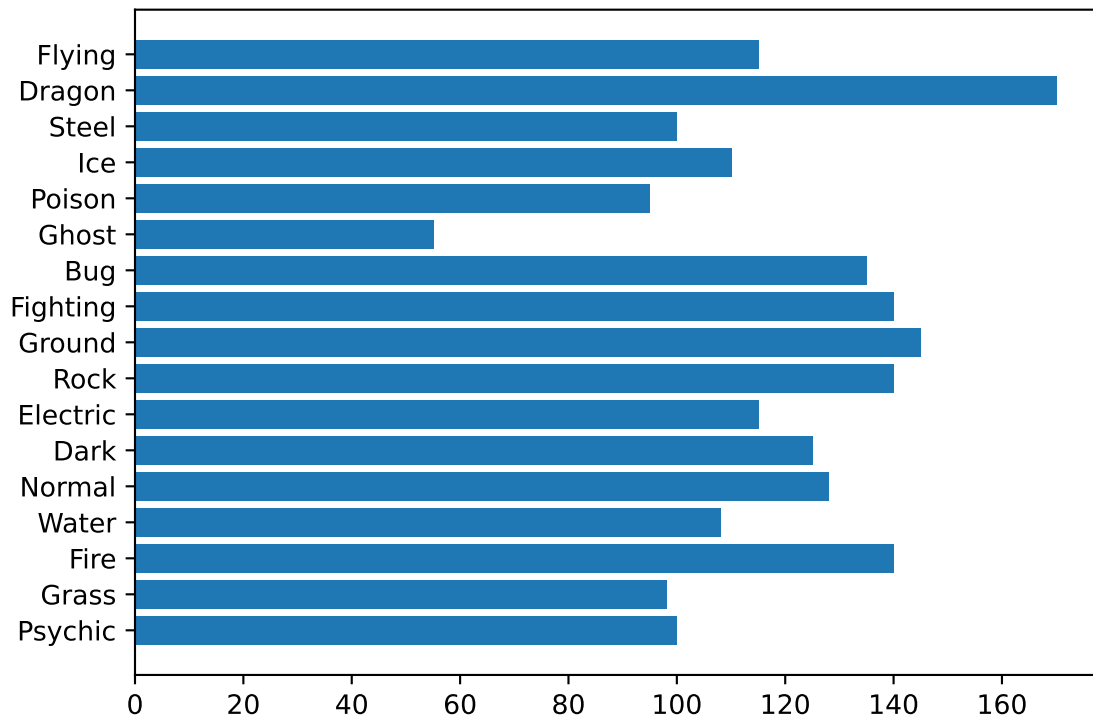
En este caso, de nuevo los pokemón tipo veneno se encuentran dentro de los 3 más debiles en cuanto a poder de ataque se refiere.

```
Generation4 = Pokedex.drop(Pokedex[Pokedex['Generation']!=4].index)
plt.barh(Generation4['Type 1'], Generation4['Attack'])
plt.show()
```



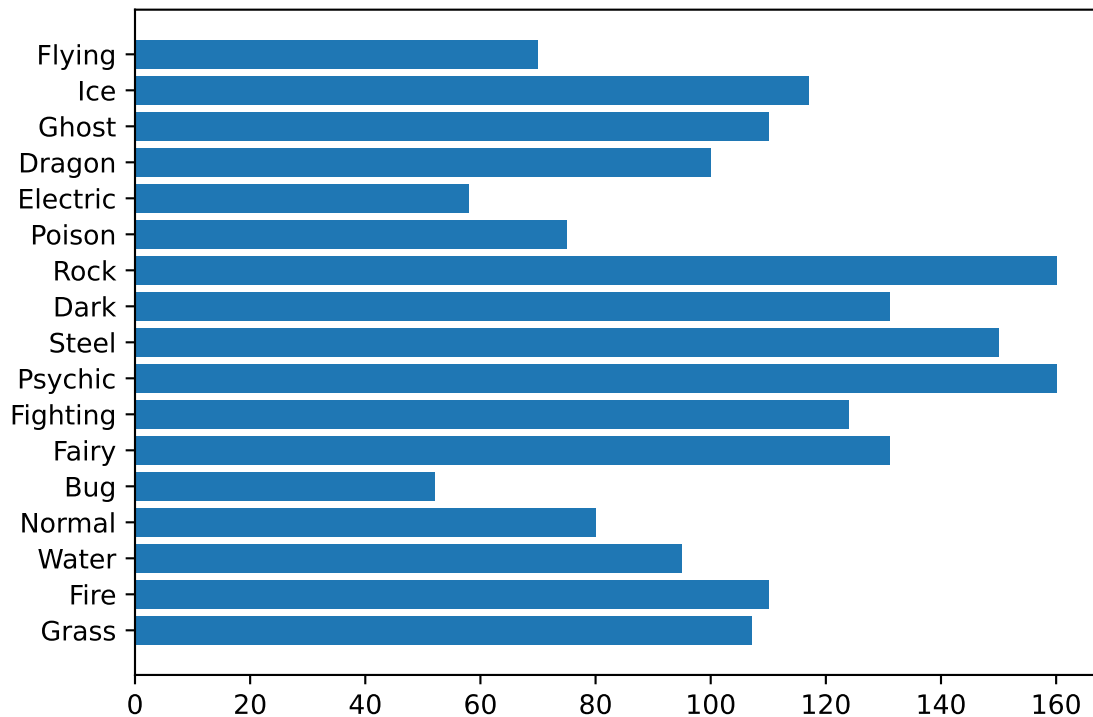
Este gráfico nos permite evidenciar lo mismo que el anterior, Pokemones tipo veneno ocupando la parte baja de las posiciones de poder de ataque, nuevamente solo por encima de dos tipos.

```
Generation5 = Pokedex.drop(Pokedex[Pokedex['Generation']!=5].index)
plt.barh(Generation5['Type 1'], Generation5['Attack'])
plt.show()
```



Para este caso se vuelve a segunda generación, donde el tipo veneno solo se encuentra únicamente por encima de los tipos fantasma

```
Generation6 = Pokedex.drop(Pokedex[Pokedex['Generation']!=6].index)
plt.barh(Generation6['Type 1'], Generation6['Attack'])
plt.show()
```

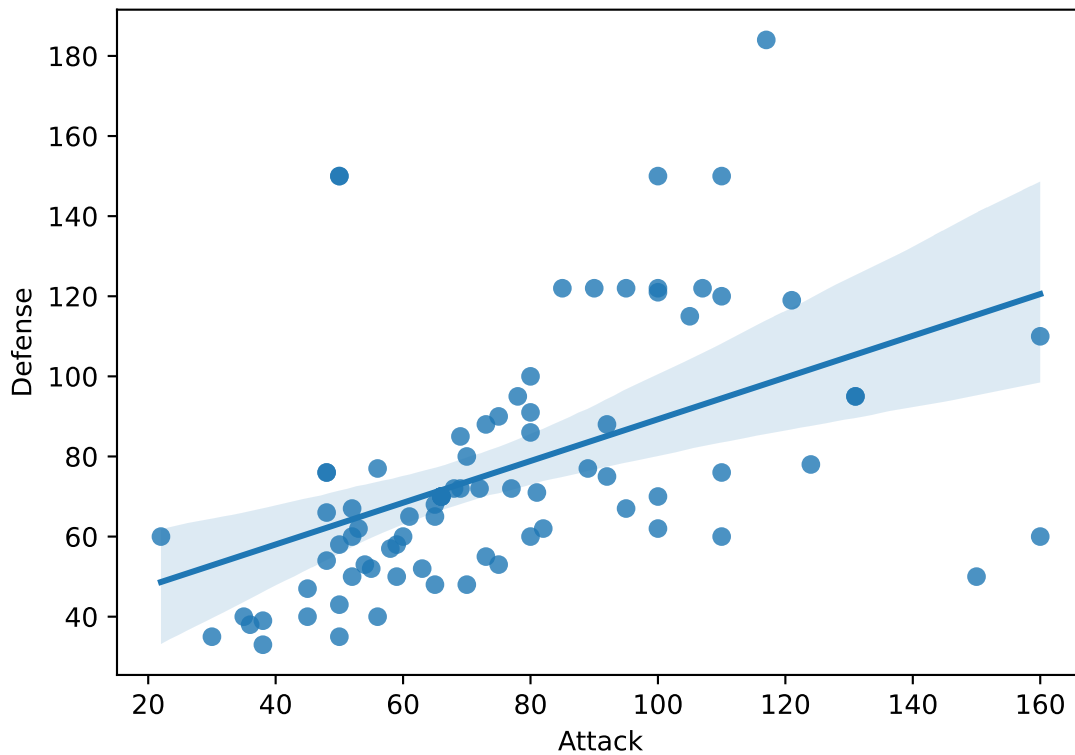


Al terminar de revisar todas las gráficas, se puede observar que pese a que los tipo veneno tiene el valor máximo de poder de ataque más pequeño en términos generales, es decir, revisando todas las generaciones a la vez. Sin embargo, al ver estas gráficas, se puede observar como patrón que los Pokémon tipo eléctrico, “bug” y veneno son aquellos que menor poder de ataque tienen al revisar generación por generación, donde suelen ser mínimo dos de estos el top 3 de menor poder de ataque.

Además podemos encontrar que hasta la tercera generación, existía un patrón el cual colocaba a sus respectivos Pokémon por generación con mayor poder que los demás, es decir, primera, segunda y tercera generación de Psíquicos son los que desarrollan mayor poder de ataque. Este patrón se rompe durante las generaciones 4 y 5, pero vuelve a darse en la generación 6. Por esto podemos concluir que tenemos una tendencia neutra, en la cual no crece ni decrece en grandes proporciones el poder de los Pokémon tipo psíquicos durante las 3 primeras generaciones.

Como amante de los juegos de todo tipo, siempre existe el dilema de que para que haya un balance en un personaje, debe existir un patrón donde su defensa tiene que ser inversamente proporcional a su ataque, por lo cual quise hacer uso de un diagrama de dispersión para ponerlo a prueba, en este caso en la sexta generación.

```
Generation6 = Pokedex.drop(Pokedex[Pokedex['Generation']!=6].index)
sns.regplot(x=Generation6["Attack"], y=Generation6["Defense"])
plt.show()
```



Como se puede observar, las estadísticas de ataque y defensa tienen un coeficiente de correlación positivo, sin embargo, son muy pocos aquellos que se acercan a seguirlo, es decir, se tiene muchos valores dispersos. Se puede concluir de esta gráfica que no se presenta una relación de proporcionalidad entre la defensa y el ataque, dejando de seguir un patrón, bien sea por que varíe con otras de las variables como la vida o el tipo de Pokémon o por que a cambio de un gran ataque y una gran defensa a la vez, se imponga alguna debilidad al Pokémon.