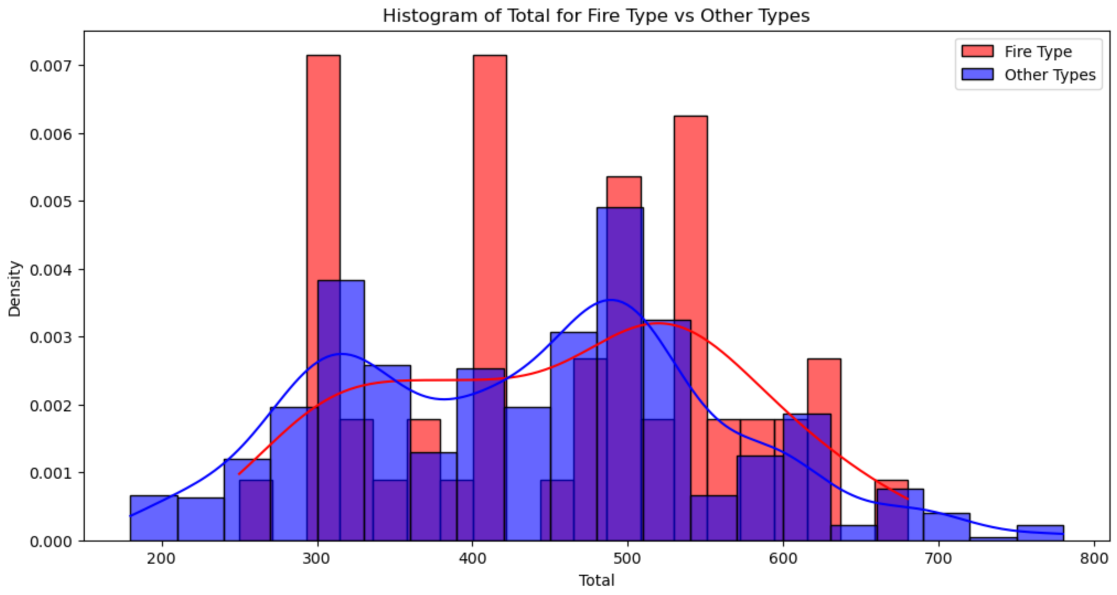
RodriguezDSC530-T301FinalProject

March 1, 2025

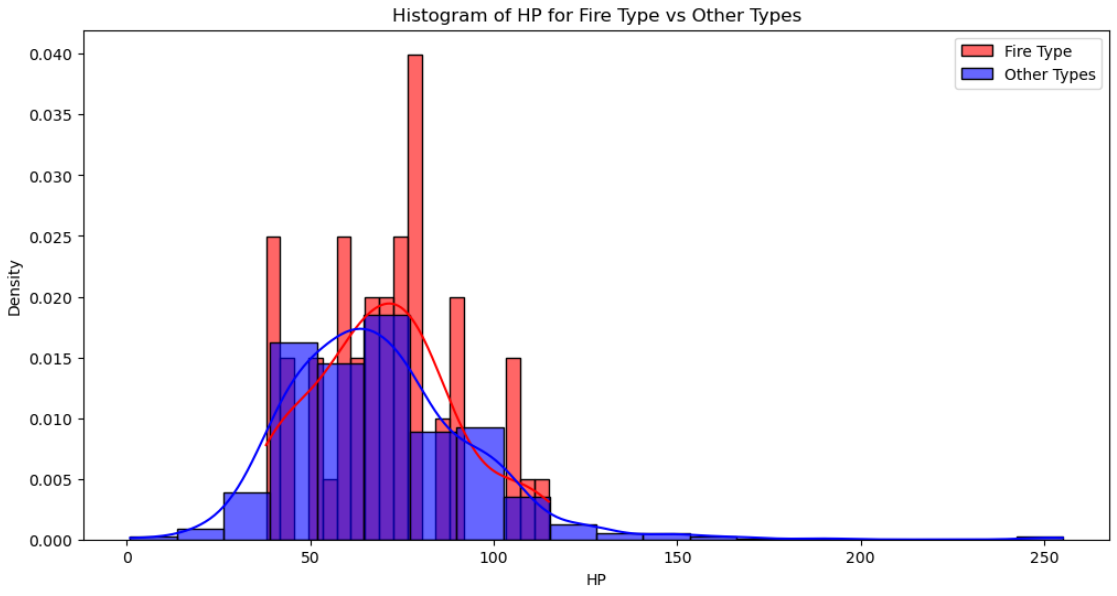
[7]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import statsmodels.api as sm
from scipy import stats
from scipy.stats import pareto
from mpl_toolkits.mplot3d import Axes3D

# Load the dataset
df = pd.read_csv('Pokemon.csv')

# Create a Fire-type dummy variable where 1 = Fire and 0 = Other
df['Fire_Type'] = (df['Type 1'] == 'Fire').astype(int)

# Filter fire-type Pokemon and others
fire_type = df[df['Type 1'] == 'Fire']
other_types = df[df['Type 1'] != 'Fire']

# Descriptive statistics function
def descriptive_stats(series):
    return {
        'Mean': series.mean(),
        'Mode': series.mode()[0],
        'Spread (Std Dev)': series.std(),
        'Tails':{
            'Skewness': series.skew(),
            'Kurtosis': series.kurtosis()
        }
    }

# Variables to analyze
variables = ['Total', 'HP', 'Attack', 'Defense', 'Speed']

# Histograms and descriptive statistics
for var in variables:
    plt.figure(figsize=(12, 6))
    sns.histplot(fire_type[var], bins=20, color='red', kde=True, stat='density',
label='Fire Type', alpha=0.6)
    sns.histplot(other_types[var], bins=20, color='blue', kde=True,␣
 ↪stat='density',
label='Other Types', alpha=0.6)
    plt.title(f'Histogram of {var} for Fire Type vs Other Types')
    plt.xlabel(var)
    plt.ylabel('Density')
    plt.legend()

    plt.savefig(f'histogram_{var}.png', bbox_inches='tight') # Save as PNG for␣
 ↪Powerpoint

    plt.show()

    stats_fire = descriptive_stats(fire_type[var])
    stats_other = descriptive_stats(other_types[var])

    print(f"Descriptive statistics for {var} (Fire Type):", stats_fire)
    print(f"Descriptive statistics for {var} (Other Types):", stats_other)
    print("\n")

# Filter for Legendary Pokemon
legendary_pokemon = df[df['Legendary'] == True]

# Count the number of Legendary Pokemon by type
legendary_count_by_type = legendary_pokemon['Type 1'].value_counts()

# Count Fire-type Legendary Pokemon
fire_legendaries_count = legendary_count_by_type.get('Fire', 0)

# Prepare data for comparison
legendary_count_by_type = legendary_count_by_type.reset_index()
legendary_count_by_type.columns = ['Type', 'Count']

# Filter out Fire-type for separate comparison
other_legendaries = legendary_count_by_type[legendary_count_by_type['Type'] !=␣
 ↪'Fire']

# Create a DataFrame to include Fire-type count
comparison_data = pd.DataFrame({
    'Type': ['Fire'] + other_legendaries['Type'].tolist(),
    'Count': [fire_legendaries_count] + other_legendaries['Count'].tolist()
})

# Plot the comparison of each legendary type to fire
plt.figure(figsize=(10, 6))
```

```
sns.barplot(x='Type', y='Count', data=comparison_data)
plt.title('Comparison of Fire Type Legendary Pokemon to Other Legendary Types')
plt.xlabel('Legendary Type')
plt.ylabel('Count')
plt.xticks(rotation=45)

plt.savefig('legendary_comparison.png', bbox_inches='tight') # Save as PNG for␣
 ↪Powerpoint
plt.show()

# PMF comparison for Attack
plt.figure(figsize=(12, 6))
sns.histplot(fire_type['Attack'], bins=20, stat='probability', color='red',␣
 ↪label='Fire Type', alpha=0.6)
sns.histplot(other_types['Attack'], bins=20, stat='probability', color='blue',␣
 ↪label='Other Types', alpha=0.6)
plt.title('PMF of Attack for Fire Type vs Other Types')
plt.xlabel('Attack')
plt.ylabel('Probability')
plt.legend()

plt.savefig('pmf_attack.png', bbox_inches='tight') # Save as PNG for Powerpoint
plt.show()

# CDF for the Total variable
plt.figure(figsize=(12, 6))
sns.ecdfplot(fire_type['Total'], label='Fire Type', color='red')
sns.ecdfplot(other_types['Total'], label='Other Types', color='blue')
plt.title('CDF of the Total variable for Fire Type vs Other Types')
plt.xlabel('Total Stats')
plt.ylabel('Cumulative Probability')
plt.legend()

plt.savefig('cdf_total.png', bbox_inches='tight') # Save as PNG for Powerpoint
plt.show()

# Analytical distribution (Pareto Distribution)
# Utilizing Parteo because I know there is a large number of low-stat Pokemon␣
 ↪and fewer high-stat Pokemon
alpha = 2 # Shape parameter
x = np.linspace(df['Total'].min(), df['Total'].max(), 100)
m = df['Total'].min() # Scale paramter (xmin)
pdf = pareto.pdf(x, b=alpha, scale=m)

plt.figure(figsize=(12, 6))
plt.plot(x, pdf, color='blue', label='Parteo Distribution', lw=2)
plt.title('Analytical Pareto Distribution for Total Stats')
plt.xlabel('Total Stats')
plt.ylabel('Density')

plt.savefig('pareto_distribution.png', bbox_inches='tight') # Save as PNG for␣
  ↪Powerpoint
plt.show()

# Scatter plots for correlation
plt.figure(figsize=(12, 6))
sns.scatterplot(data=df, x = 'Attack', y='Speed', hue='Type 1', alpha=0.7)
plt.title('Scatter Plot of Attack vs Speed')
plt.xlabel('Attack')
plt.ylabel('Speed')

plt.savefig('scatter_attack_speed.png', bbox_inches='tight') # Save as PNG for␣
  ↪Powerpoint
plt.show()

# 3D Scatter plot for Total Stats vs. Attack vs. Special Attack
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')

# Get data for plotting
x = df['Attack']
y = df['Sp. Atk']
z = df['Total']

# Color mapping for types
type_colors = {type_name: idx for idx, type_name in enumerate(df['Type 1'].
  ↪unique())}
colors = df['Type 1'].map(type_colors)

# Scatter plot
scatter = ax.scatter(x, y, z, c=df['Type 1'].astype('category').cat.codes,␣
  ↪cmap='viridis', alpha=0.6, s=50)
ax.set_xlabel('Attack')
ax.set_ylabel('Special Attack')
ax.set_zlabel('Total Stats')
ax.set_title('Total Stats vs. Attack vs. Special Attack for All Pokemon')

# Create a legend
handles = []
for type_name, color_idx in type_colors.items():
    handles.append(plt.Line2D([0], [0], marker='o', color='w', label=type_name,
                              markerfacecolor=scatter.cmap(color_idx /␣
  ↪len(type_colors)), markersize=10))
```

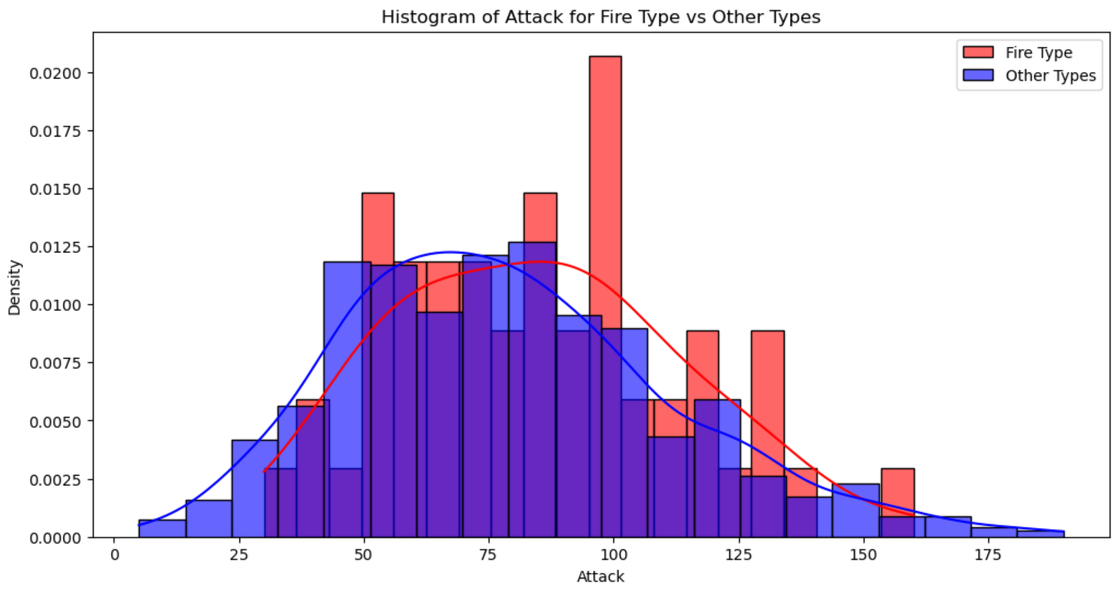
```
ax.legend(handles=handles, title="Types", bbox_to_anchor=(1.05, 1), loc='upper␣
 ↪left')

plt.savefig('3d_scatter_total_attack_spatk.png', bbox_inches='tight') # Save as␣
 ↪PNG for Powerpoint
plt.show()

# Correlation and covariance
covariance = df['Attack'].cov(df['Speed'])
pearson_corr = df['Attack'].corr(df['Speed'])
print(f"Covariance between Attack and Speed: {covariance}")
print(f"Pearson's Correlation between Attack and Speed: {pearson_corr}")

# Setup for the hypothesis test and regression analysis
# Null Hypothesis: Fire-type Pokemon do not have significantly higher Total␣
 ↪Stats than Other Types

# Prepare data for hypothesis testing
fire_total = fire_type['Total']
other_total = other_types['Total']

# Conduct Hypothesis Test
# Using two-sample t-test to compare means
t_stat, p_value = stats.ttest_ind(fire_total, other_total, equal_var=False) #␣
 ↪Welch's t-test
print(f"T-statistic: {t_stat}, P-value: {p_value}")

# Regression Analysis
X = sm.add_constant(df['Fire_Type']) # Idependent variable
y = df['Total'] # Depedent variable

model = sm.OLS(y, X).fit()
print(model.summary())
```



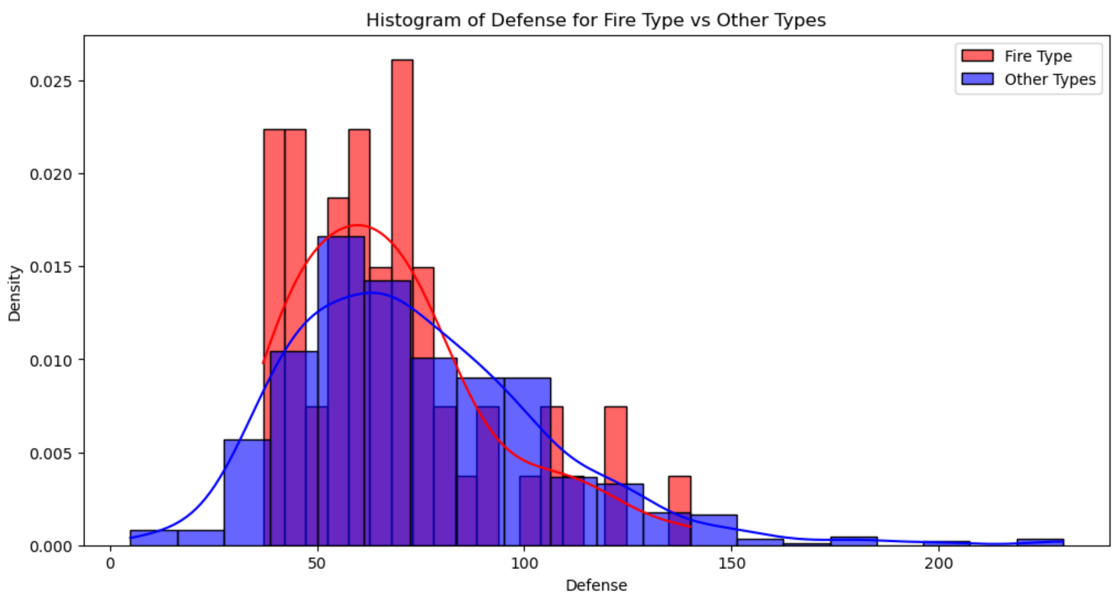
```
Descriptive statistics for Total (Fire Type): {'Mean': 458.0769230769231,
'Mode': 405, 'Spread (Std Dev)': 109.76049615338435, 'Tails': {'Skewness':
-0.06275679026772849, 'Kurtosis': -1.0239261014213268}}
Descriptive statistics for Total (Other Types): {'Mean': 433.5053475935829,
'Mode': 600, 'Spread (Std Dev)': 120.54507496592437, 'Tails': {'Skewness':
0.17086580618913044, 'Kurtosis': -0.4769044392960442}}
```



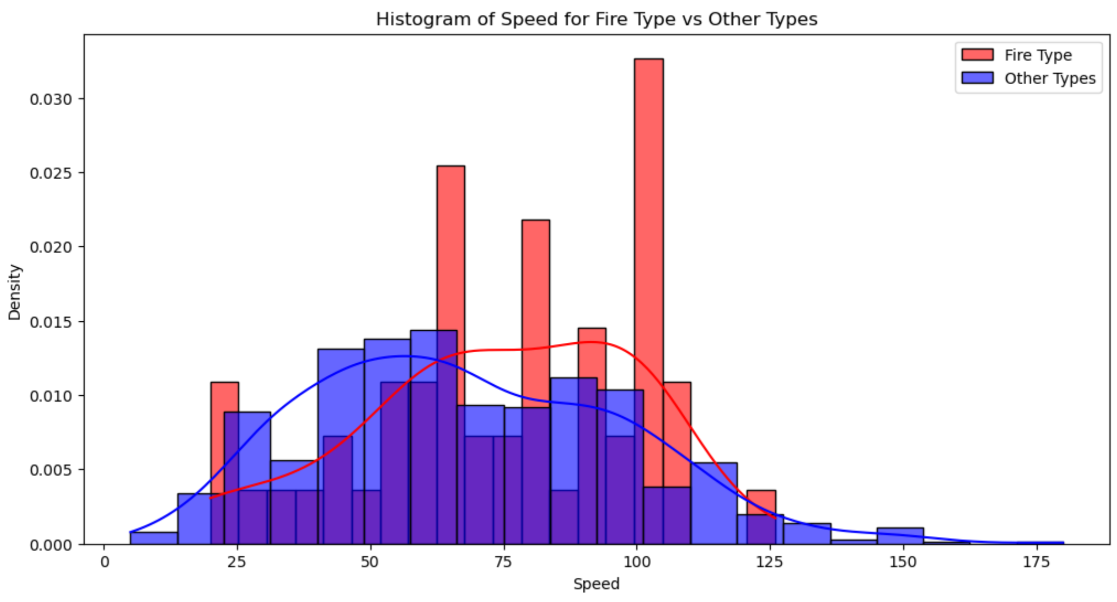
```
Descriptive statistics for HP (Fire Type): {'Mean': 69.90384615384616, 'Mode':
78, 'Spread (Std Dev)': 19.404122884506883, 'Tails': {'Skewness':
0.3039610282727519, 'Kurtosis': -0.2911384071933947}}
Descriptive statistics for HP (Other Types): {'Mean': 69.21390374331551, 'Mode':
60, 'Spread (Std Dev)': 25.9166043698439, 'Tails': {'Skewness':
1.5985398621070595, 'Kurtosis': 7.272507700719929}}
```



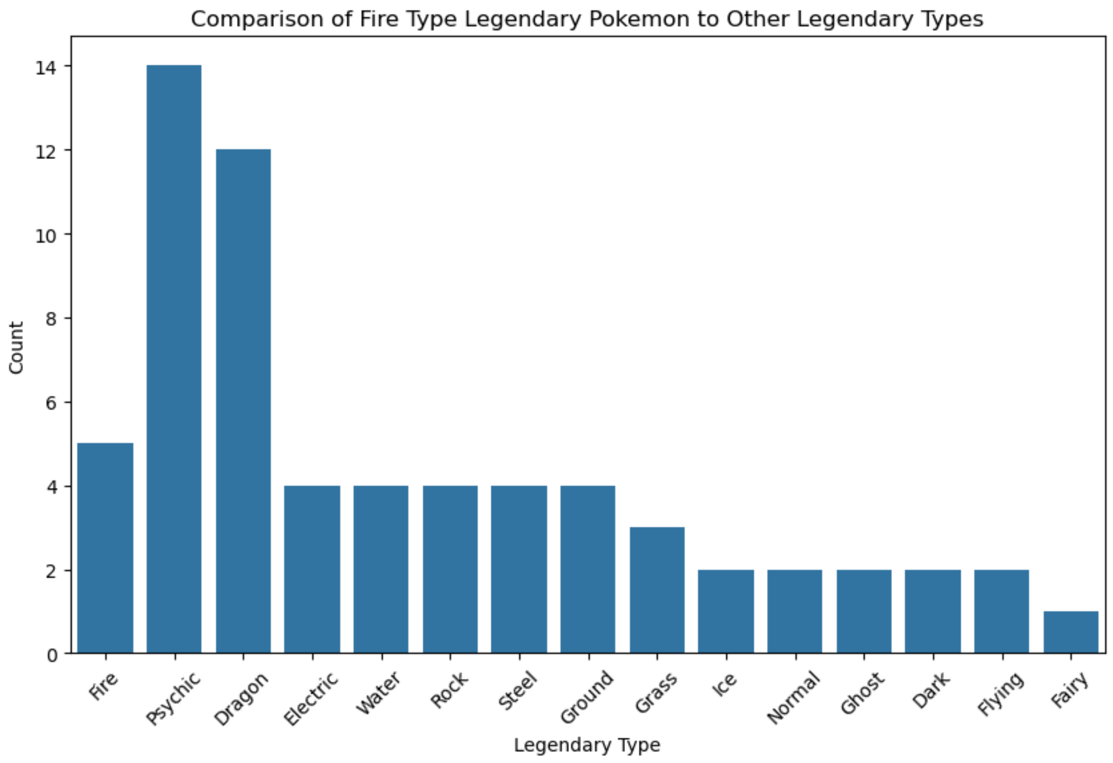
```
Descriptive statistics for Attack (Fire Type): {'Mean': 84.76923076923077,
'Mode': 85, 'Spread (Std Dev)': 28.769275130832632, 'Tails': {'Skewness':
0.35047789702850574, 'Kurtosis': -0.3116964723469362}}
Descriptive statistics for Attack (Other Types): {'Mean': 78.60026737967914,
'Mode': 65, 'Spread (Std Dev)': 32.677677841484005, 'Tails': {'Skewness':
0.5709496878117126, 'Kurtosis': 0.20170275142975402}}
```

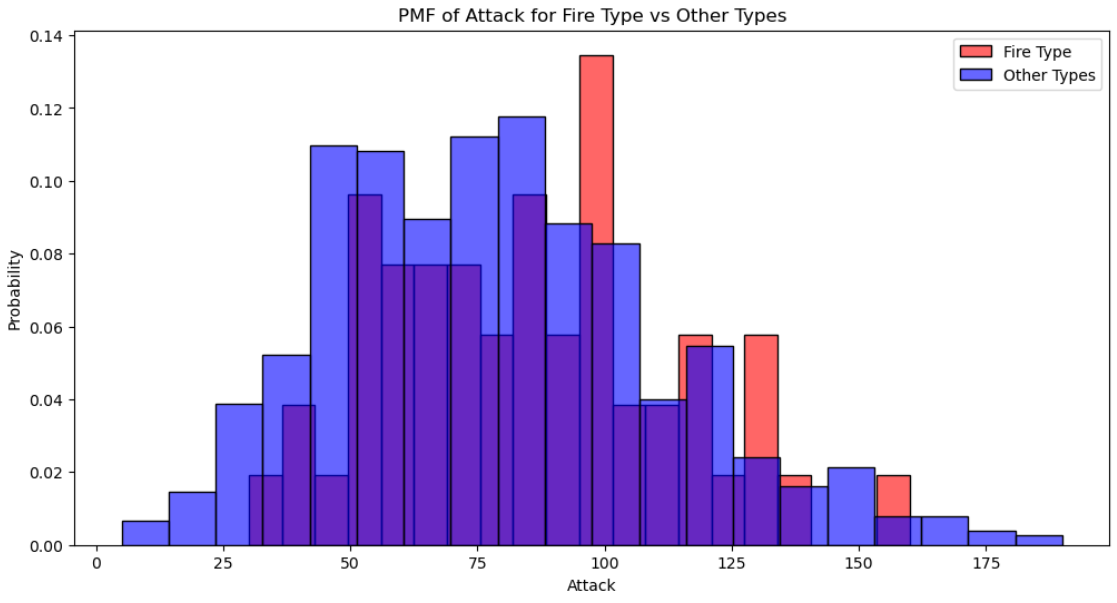


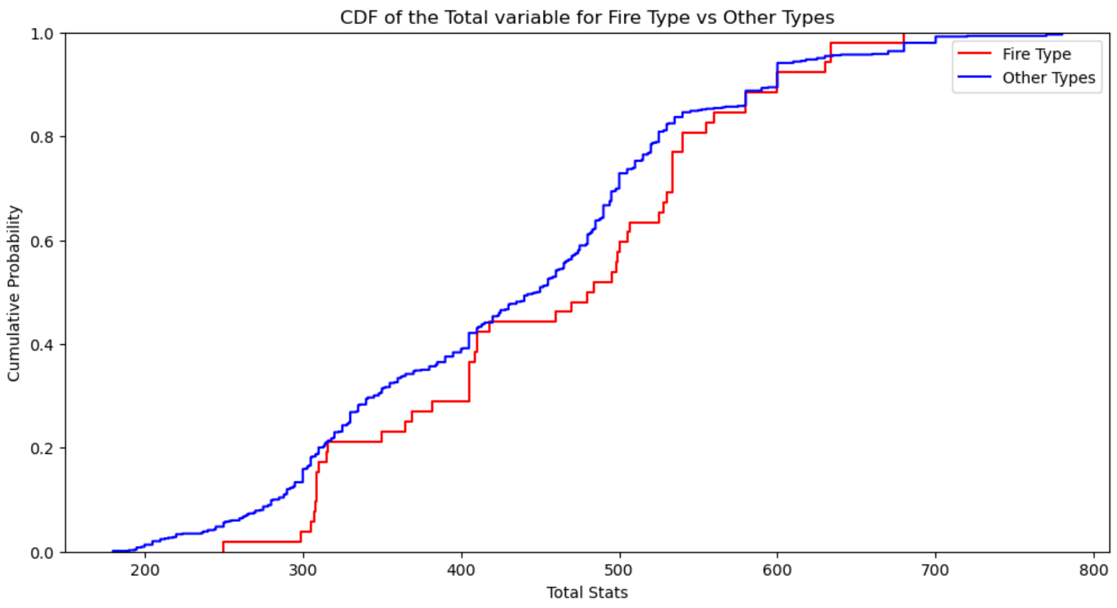
```
Descriptive statistics for Defense (Fire Type): {'Mean': 67.76923076923077,
'Mode': 40, 'Spread (Std Dev)': 23.658199577309748, 'Tails': {'Skewness':
1.0421343583467988, 'Kurtosis': 0.864818392064095}}
Descriptive statistics for Defense (Other Types): {'Mean': 74.26470588235294,
'Mode': 70, 'Spread (Std Dev)': 31.609218408569724, 'Tails': {'Skewness':
1.1409562118900907, 'Kurtosis': 2.6773860877083426}}
```

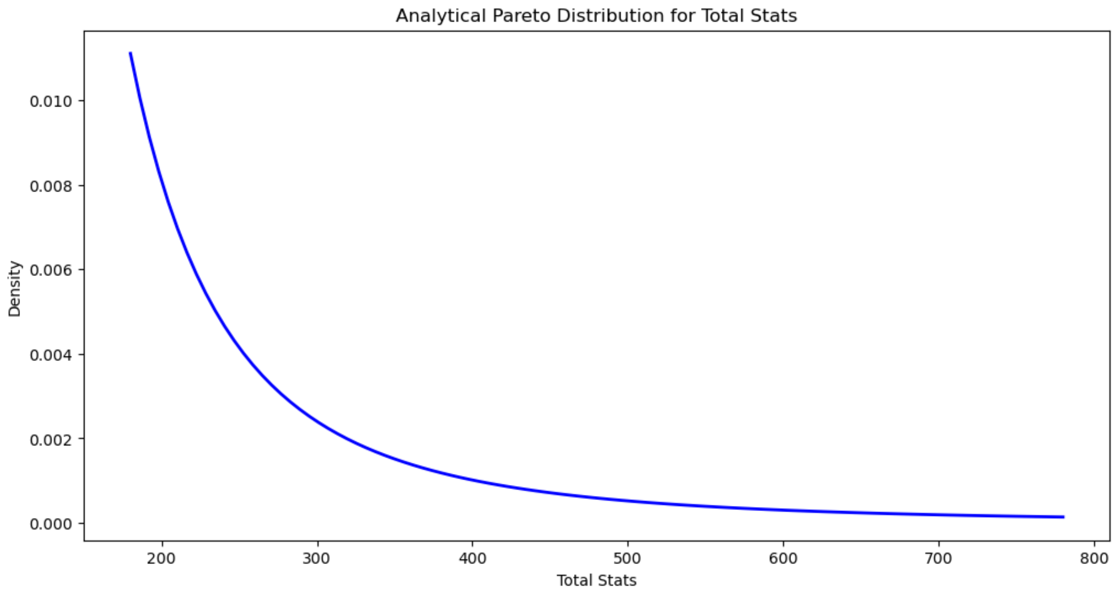


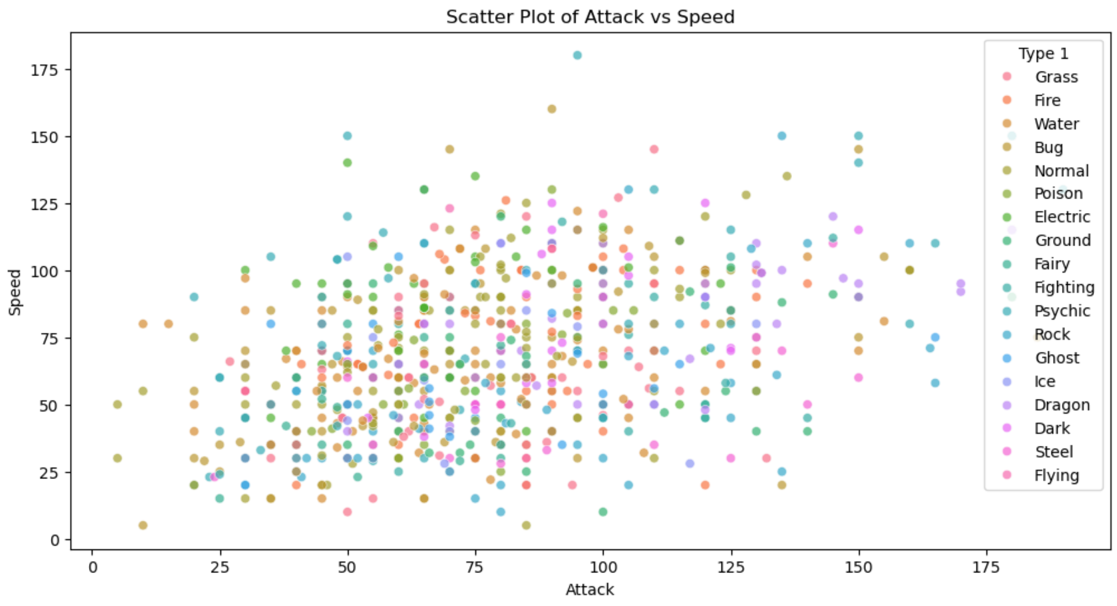
```
Descriptive statistics for Speed (Fire Type): {'Mean': 74.4423076923077, 'Mode':
100, 'Spread (Std Dev)': 25.24578276685657, 'Tails': {'Skewness':
-0.4338328655600905, 'Kurtosis': -0.37915722843837685}}
Descriptive statistics for Speed (Other Types): {'Mean': 67.84893048128342,
'Mode': 50, 'Spread (Std Dev)': 29.27380618299583, 'Tails': {'Skewness':
0.4041451460666016, 'Kurtosis': -0.19883477831773222}}
```

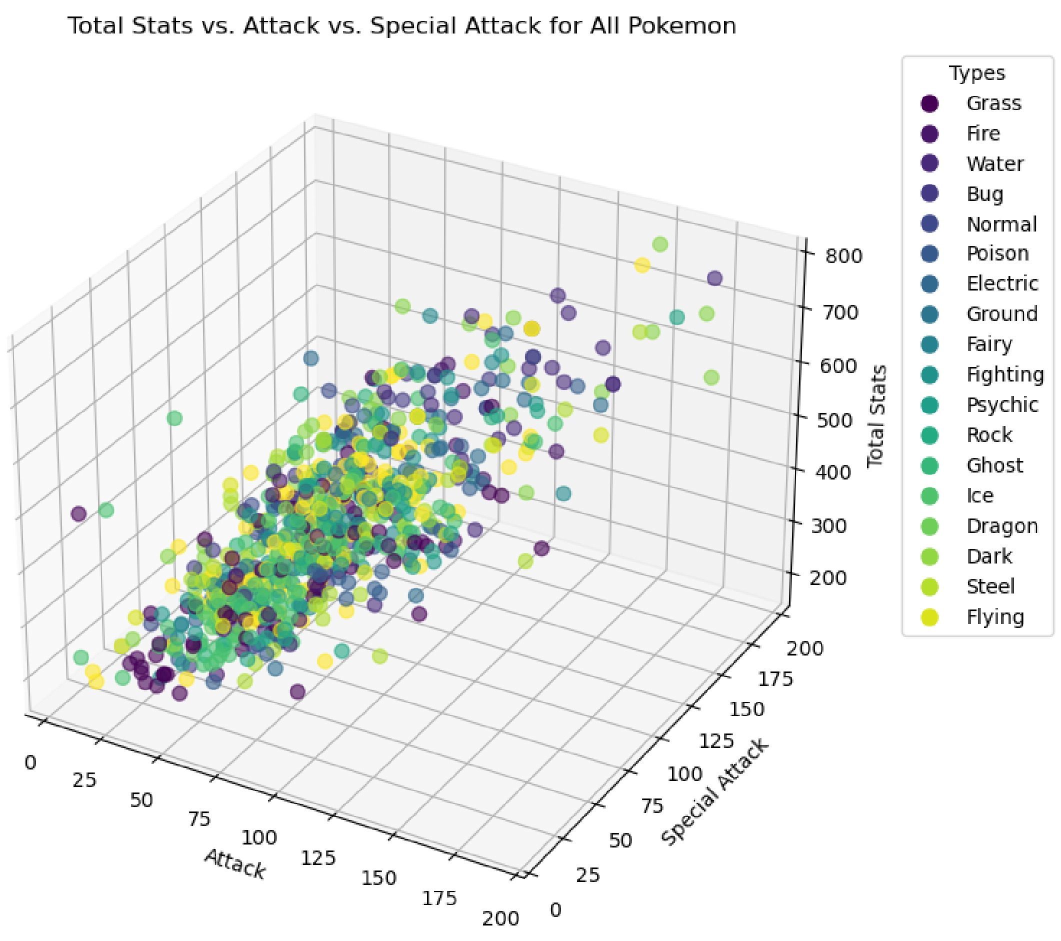



PMF of Attack for Fire Type vs Other Types
Fire Type
Other Types
Probability
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
25
50
75
100
125
150
175
Attack


CDF of the Total variable for Fire Type vs Other Types
Fire Type
Other Types
Cumulative Probability
1.0
0.8
0.6
0.4
0.2
0.0
200
300
400
500
600
700
800
Total Stats


Analytical Pareto Distribution for Total Stats
Density
0.000
0.002
0.004
0.006
0.008
0.010
Total Stats
200
300
400
500
600
700
800


Scatter Plot of Attack vs Speed
Speed
0
25
50
75
100
125
150
175
Attack
0
25
50
75
100
125
150
175
Type 1
Grass
Fire
Water
Bug
Normal
Poison
Electric
Ground
Fairy
Fighting
Psychic
Rock
Ghost
Ice
Dragon
Dark
Steel
Flying



```
Covariance between Attack and Speed: 359.59539737171457
Pearson's Correlation between Attack and Speed: 0.3812397392410896
T-statistic: 1.5506143905831495, P-value: 0.12626350091288374
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Total   R-squared:                       0.003
Model:                            OLS   Adj. R-squared:                  0.001
Method:                 Least Squares   F-statistic:                     2.042
Date:                Sat, 01 Mar 2025   Prob (F-statistic):              0.153
Time:                        20:54:52   Log-Likelihood:                -4963.4
No. Observations:                 800   AIC:                             9931.
Df Residuals:                     798   BIC:                             9940.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        433.5053      4.383     98.897      0.000     424.901     442.110
```

```
Fire_Type     24.5716     17.193      1.429      0.153      -9.178      58.321
==============================================================================
Omnibus:                       17.895   Durbin-Watson:                   1.574
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               11.756
Skew:                           0.159   Prob(JB):                      0.00280
Kurtosis:                       2.499   Cond. No.                         4.07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

```
[ ]:
```