POKEMON İSTATİSTİKLERİ

Her Pokémonun bir dizi istatistikleri vardır. HP (İsabet Puanları) bir Pokémon'un yaşam gücüdür. Pokémon'unuzun HP sıfıra vururs a, bayılır ve savaşta artık kullanılamaz. Hız , hangi Pokémonun savaşta ilk hamleyi yapacağına karar verir. Bu uzun savaşlarda kritik olabilir.

Saldırı ve Özel Saldırı , Pokemonunuz tarafından kullanılan hamlelerin gücünü ölçer; bu ne kadar yüksek olursa, rakiplere o kadar çok hasar verirsiniz. Saldırı Fiziksel kategorisindeki hareketlere karşılık gelirken Özel Saldırı Özel hareketlere karşılık gelir.

Benzer şekilde, Savunma ve Özel Savunma diğer Pokémonlardan saldırı alma yeteneğini ölçer; ne kadar yüksek olursa, saldırı sırasında o kadar az İsabet Puanı kaybedilir. Savunma, fiziksel kategorisindeki hamlelere, Özel Savunma ise özel hamlelere karşılık gelir.

Veri seti 13 değişken ve 800 gözlemden oluşmaktadır.

ID: Her pokemon için kimlik

Name: Her pokemonun ismi

Tip1: Her pokemonun bir tipi vardır, bu da saldırılara karşı zayıflığı / direnci belirler.

Tip2: Bazı pokemonlar çift tipe sahiptir.

Total: İstatistiklerin toplamı, bir pokemonun ne kadar güçlü olduğuna dair genel bir rehber.

HP: Vuruş noktaları veya sağlık, bir pokemonun bayılmadan önce dayanabileceği zararı tanımlar.

Attack: Normal saldırılar için temel değiştirici. (örn. Çizik, Yumruk)

Defense: Normal saldırılara karşı temel hasar direnci

SP.Atk: Özel saldırı, özel saldırılar için temel değiştirici. (örneğin yangın patlaması, kabarcık ışını)

SP. Def: Özel saldırılara karşı temel hasar direnci.

Speed: Hangi pokemonun ilk turda saldıracağını belirler.

Generation: Pokemonun kaçıncı nesil olduğu belirtir.

Legendary: Pokemonun efsanevi pokemon olup olmadığı belirtir.

PYTHON İLE VERİ SETİNE İLK BAKIŞ VE GÖRSELLEŞTİRME

Gerekli kütüphaneler;

```
In [80]: import pandas as pd
         import seaborn as sns
         import numpy as np
         import matplotlib.pyplot as plt
         import warnings
         from mpl_toolkits.mplot3d import Axes3D
         import os
         import matplotlib.pyplot as plt
         warnings.filterwarnings('ignore')
```

Veri import;

```
In [81]: pokemon = pd.read_csv("pokemon.csv")
```

```
In [82]: df = pokemon.copy()
```

```
In [83]: df.head()
```

Out[83]:

	#	Name	Type1	Type2	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

Veri setinin ilk dört gözlemine bakılmıştır.

```
In [84]: df.tail()
```

Out[84]:

	#	Name	Type1	Type2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
795	719	Diancie	Rock	Fairy	600	50	100	150	100	150	50	6	True
796	719	DiancieMega Diancie	Rock	Fairy	700	50	160	110	160	110	110	6	True
797	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	60	150	130	70	6	True
798	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	60	170	130	80	6	True
799	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	True

```
In [85]: df.columns
```

```
Out[85]: Index(['#', 'Name', 'Type1', 'Type2', 'Total', 'HP', 'Attack', 'Defense',
                'Sp. Atk', 'Sp. Def', 'Speed', 'Generation', 'Legendary'],
               dtype='object')
```

```
In [86]: df.shape
Out[86]: (800, 13)

In [87]: df.count()
Out[87]: #             800
         Name          800
         Type1         800
         Type2         414
         Total         800
         HP            800
         Attack        800
         Defense       800
         Sp. Atk       800
         Sp. Def       800
         Speed         800
         Generation    800
         Legendary     800
         dtype: int64
```

Veri setinin son dört gözlemine bakılmış ve 800 satır 13 sütundan oluştuğu bilgisine varılmıştır.

```
In [88]: df.info()
         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 800 entries, 0 to 799
         Data columns (total 13 columns):
         #             800 non-null int64
         Name          800 non-null object
         Type1         800 non-null object
         Type2         414 non-null object
         Total         800 non-null int64
         HP            800 non-null int64
         Attack        800 non-null int64
         Defense       800 non-null int64
         Sp. Atk       800 non-null int64
         Sp. Def       800 non-null int64
         Speed         800 non-null int64
         Generation    800 non-null int64
         Legendary     800 non-null bool
         dtypes: bool(1), int64(9), object(3)
         memory usage: 75.9+ KB

In [89]: df.Name = pd.Categorical(df.Name)

In [90]: df.Type1 = pd.Categorical(df.Type1)

In [91]: df.Type2 = pd.Categorical(df.Type2)
```

```
In [92]: df.dtypes

Out[92]: #                int64
         Name            category
         Type1           category
         Type2           category
         Total              int64
         HP                 int64
         Attack             int64
         Defense            int64
         Sp. Atk            int64
         Sp. Def            int64
         Speed              int64
         Generation         int64
         Legendary           bool
         dtype: object
```

Değişken tipleri çıktıda görüldüğü gibidir.

```
In [93]: df.describe(include ='all').T
```

Out[93]:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
#	800	NaN	NaN	NaN	362.814	208.344	1	184.75	364.5	539.25	721
Name	800	800	Zygarde50% Forme	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type1	800	18	Water	112	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type2	414	18	Flying	97	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Total	800	NaN	NaN	NaN	435.103	119.963	180	330	450	515	780
HP	800	NaN	NaN	NaN	69.2588	25.5347	1	50	65	80	255
Attack	800	NaN	NaN	NaN	79.0012	32.4574	5	55	75	100	190
Defense	800	NaN	NaN	NaN	73.8425	31.1835	5	50	70	90	230
Sp. Atk	800	NaN	NaN	NaN	72.82	32.7223	10	49.75	65	95	194
Sp. Def	800	NaN	NaN	NaN	71.9025	27.8289	20	50	70	90	230
Speed	800	NaN	NaN	NaN	68.2775	29.0605	5	45	65	90	180
Generation	800	NaN	NaN	NaN	3.32375	1.66129	1	2	3	5	6
Legendary	800	2	False	735	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Değişkenlerin özetleyici istatistikleri ve eksik gözlem sayıları verilmiştir.

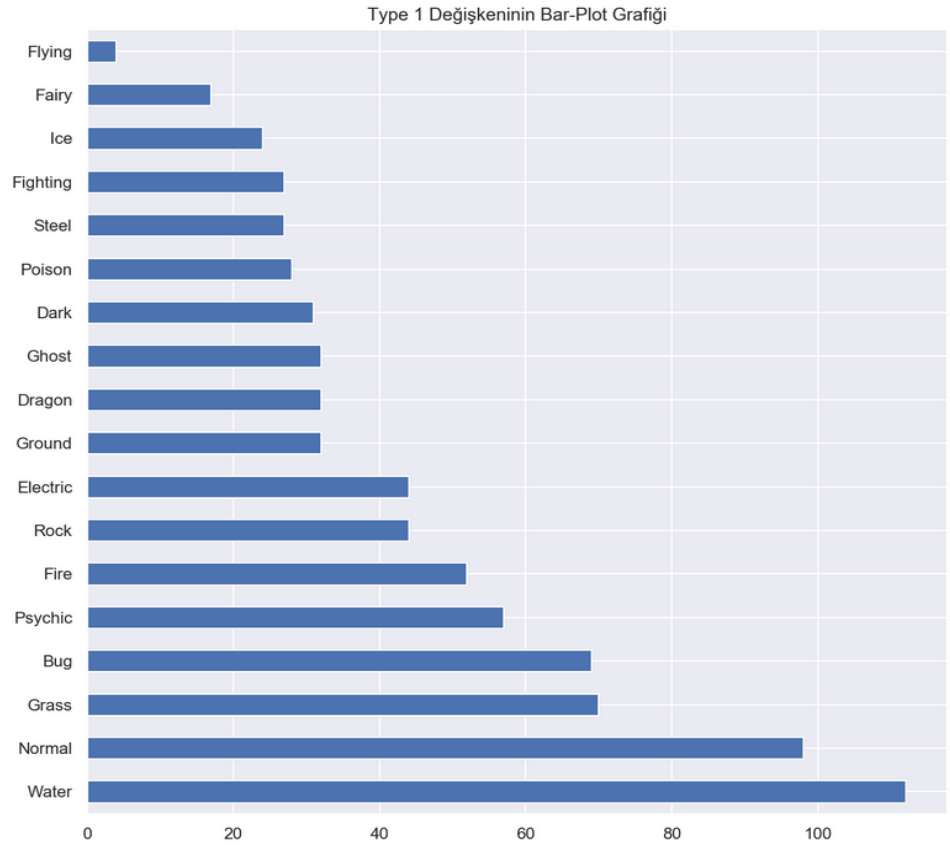
```
In [94]: df.corr()
```

Out[94]:

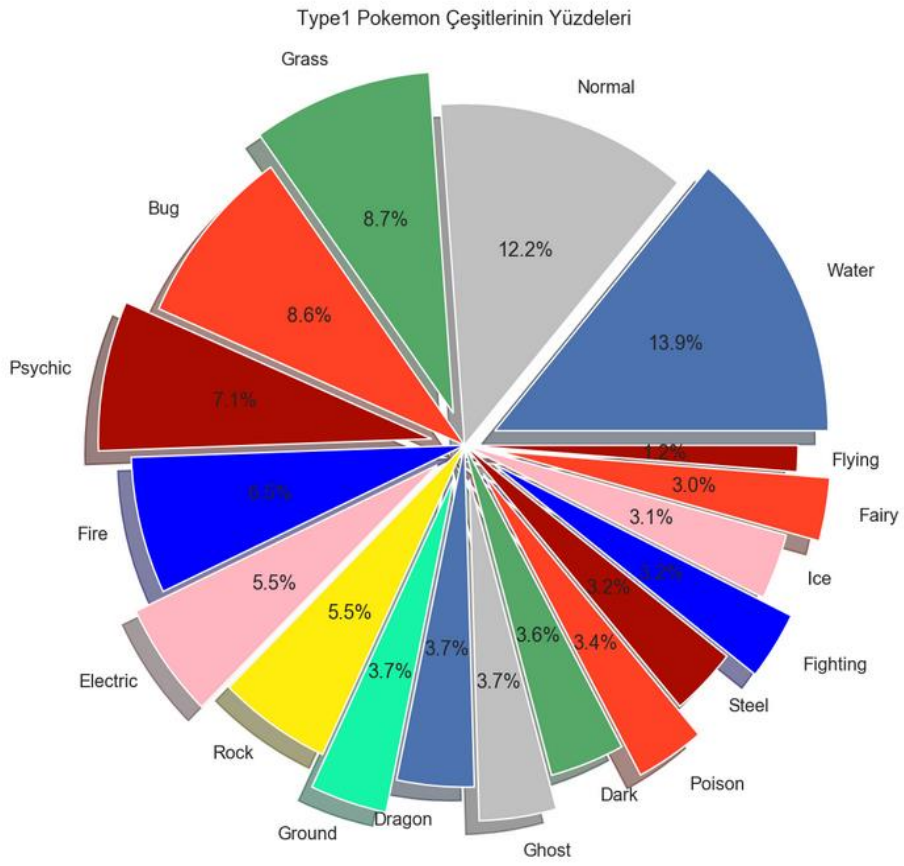
	#	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
#	1.000000	0.119813	0.097614	0.102298	0.094786	0.088759	0.085817	0.010733	0.982516	0.153396
Total	0.119813	1.000000	0.618748	0.736211	0.612787	0.747250	0.717609	0.575943	0.048384	0.501758
HP	0.097614	0.618748	1.000000	0.422386	0.239622	0.362380	0.378718	0.175952	0.058683	0.273620
Attack	0.102298	0.736211	0.422386	1.000000	0.438687	0.396362	0.263990	0.381240	0.051451	0.345408
Defense	0.094786	0.612787	0.239622	0.438687	1.000000	0.223549	0.510747	0.015227	0.042419	0.246377
Sp. Atk	0.088759	0.747250	0.362380	0.396362	0.223549	1.000000	0.506121	0.473018	0.036437	0.448907
Sp. Def	0.085817	0.717609	0.378718	0.263990	0.510747	0.506121	1.000000	0.259133	0.028486	0.363937
Speed	0.010733	0.575943	0.175952	0.381240	0.015227	0.473018	0.259133	1.000000	-0.023121	0.326715
Generation	0.982516	0.048384	0.058683	0.051451	0.042419	0.036437	0.028486	-0.023121	1.000000	0.079794
Legendary	0.153396	0.501758	0.273620	0.345408	0.246377	0.448907	0.363937	0.326715	0.079794	1.000000

Değişkenler arasındaki korelasyon katsayıları verilmiştir.

```
In [96]: df['Type1'].value_counts().plot.barh().set_title('Type 1 Değişkeninin Bar-Plot Grafiği');
```

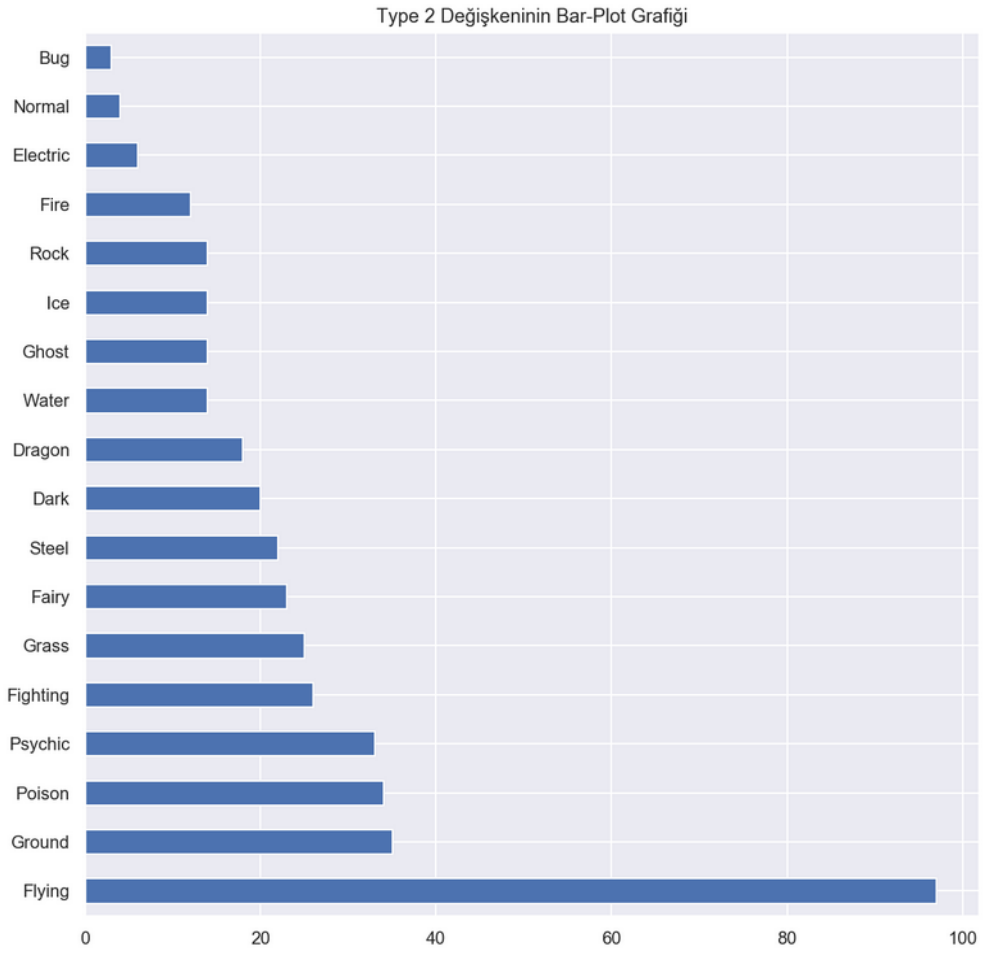


In [115]:
```
labels = ['Water', 'Normal', 'Grass', 'Bug', 'Psychic', 'Fire', 'Electric', 'Rock', 'Ground','Dragon','Ghost','Dark','
Poison','Steel','Fighting','Ice','Fairy','Flying']
sizes = [112, 98, 70, 69, 57, 52, 44, 44, 30,30,30,29,27,26,26,25,24,10]
colors = ['B', 'silver', 'G', '#ff4125', '#aa0b00', '#0000ff','#FFB6C1', '#FFED0D', '#16F5A7','B', 'silver', 'G', '#ff
4125', '#aa0b00', '#0000ff','#FFB6C1','#ff4125', '#aa0b00']
explode = (0.1, 0.0, 0.1, 0, 0.1, 0.0, 0.1, 0, 0.1,0.0,0.1,0,0.1,0,0.1,0.0,0.1,0)
plt.pie(x=sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=0,counterclock=True)
plt.axis('scaled')
plt.title("Type1 Pokemon Çeşitlerinin Yüzdeleri")
fig=plt.gcf()
fig.set_size_inches(9,9)
plt.show()
```

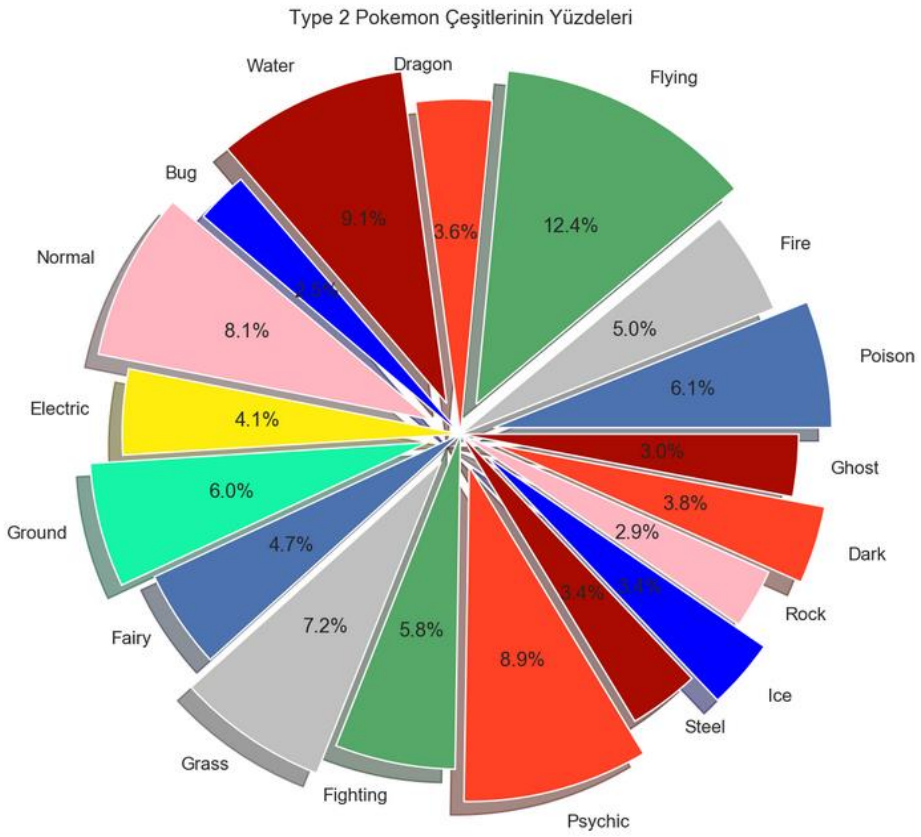


Type1 değişkeni için grafikler incelendiğinde veri setinde water grubu pokemonun diğer pokemon gruplarına göre fazlalık gösterdiği söylenebilmektedir.

In [97]:
```
df['Type2'].value_counts().plot.barh().set_title('Type 2 Değişkeninin Bar-Plot Grafiği');
```

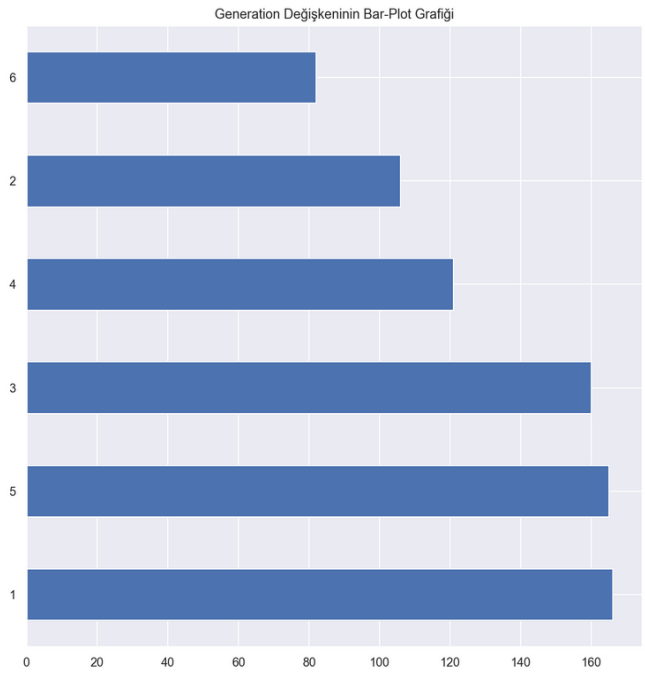


In [113]:
```
labels = ['Poison', 'Fire', 'Flying', 'Dragon', 'Water', 'Bug', 'Normal',
       'Electric', 'Ground', 'Fairy', 'Grass', 'Fighting', 'Psychic',
       'Steel', 'Ice', 'Rock', 'Dark', 'Ghost']
sizes = [49,40,99,29,73,20,65,33,48,38,58,46,71,27,27,23,30,24]
colors = ['B', 'silver', 'G', '#ff4125', '#aa0b00', '#0000ff','#FFB6C1', '#FFED0D', '#16F5A7','B', 'silver', 'G', '#ff
4125', '#aa0b00', '#0000ff','#FFB6C1','#ff4125', '#aa0b00']
explode = (0.1, 0.0, 0.1, 0, 0.1, 0.0, 0.1, 0, 0.1,0.0,0.1,0.0,0.1,0.0,0.1,0.0,0.1,0.0)
plt.pie(x=sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=0,counterclock=True)
plt.axis('scaled')
plt.title("Type 2 Pokemon Çeşitlerinin Yüzdeleri")
fig=plt.gcf()
fig.set_size_inches(9,9)
plt.show()
```



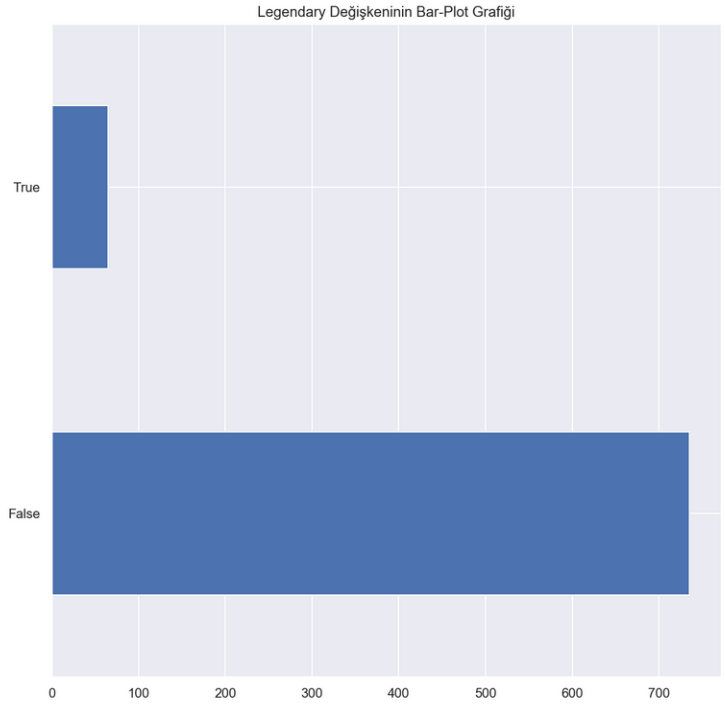
Type2 değişkenine bakıldığında ise flying pokemon grubunun diğer pokemon gruplarına göre fazlalık gösterdiği söylenebilmektedir.

```
In [98]: df['Generation'].value_counts().plot.barh().set_title('Generation Değişkeninin Bar-Plot Grafiği');
```



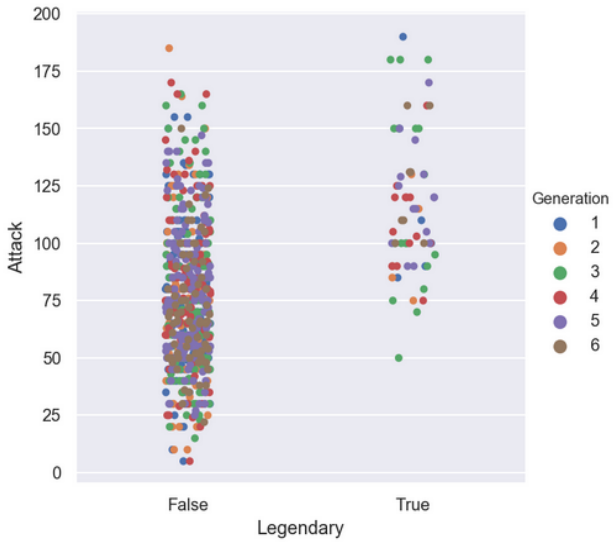
Eski nesil pokemonların (1 ve 2), yeni nesil pokemonlara göre veri setinde daha fazla yer aldığı söylenebilmektedir.

```
In [99]: df['Legendary'].value_counts().plot.barh().set_title('Legendary Değişkeninin Bar-Plot Grafiği');
```



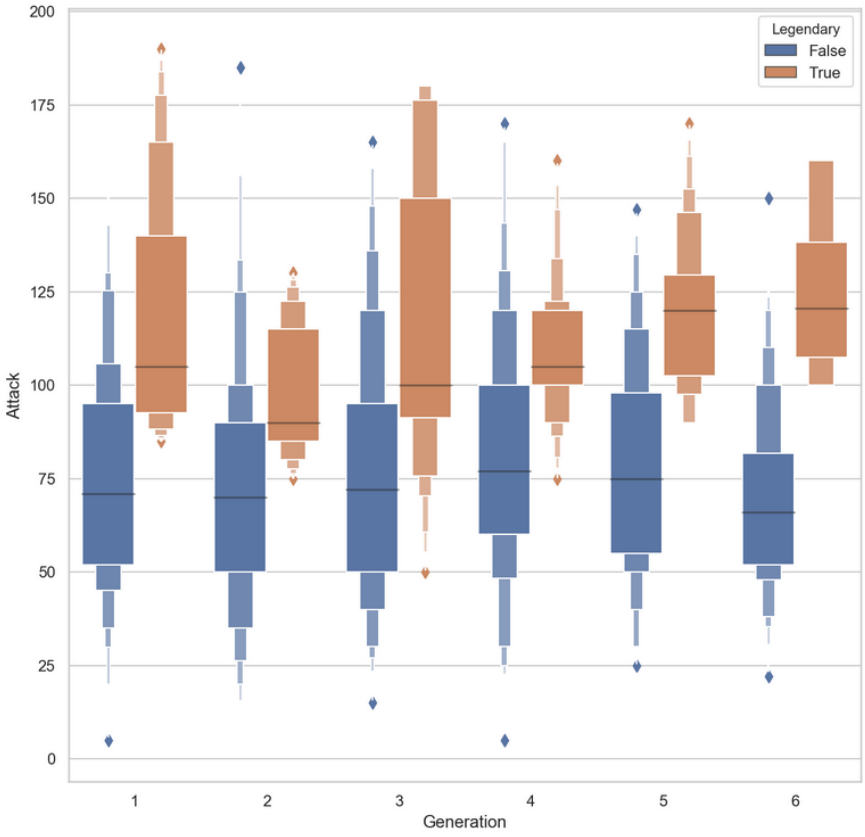
Son olarak efsanevi pokemon olma durumlarına bakıldığında, veri setinde efsanevi olmayan pokemon sayısının oldukça fazla olduğu grafikte görülmektedir.

```
In [102]: %config InlineBackend.figure_format = 'retina'
          sns.catplot(x = 'Legendary', y = 'Attack', hue = "Generation", data = df);
```



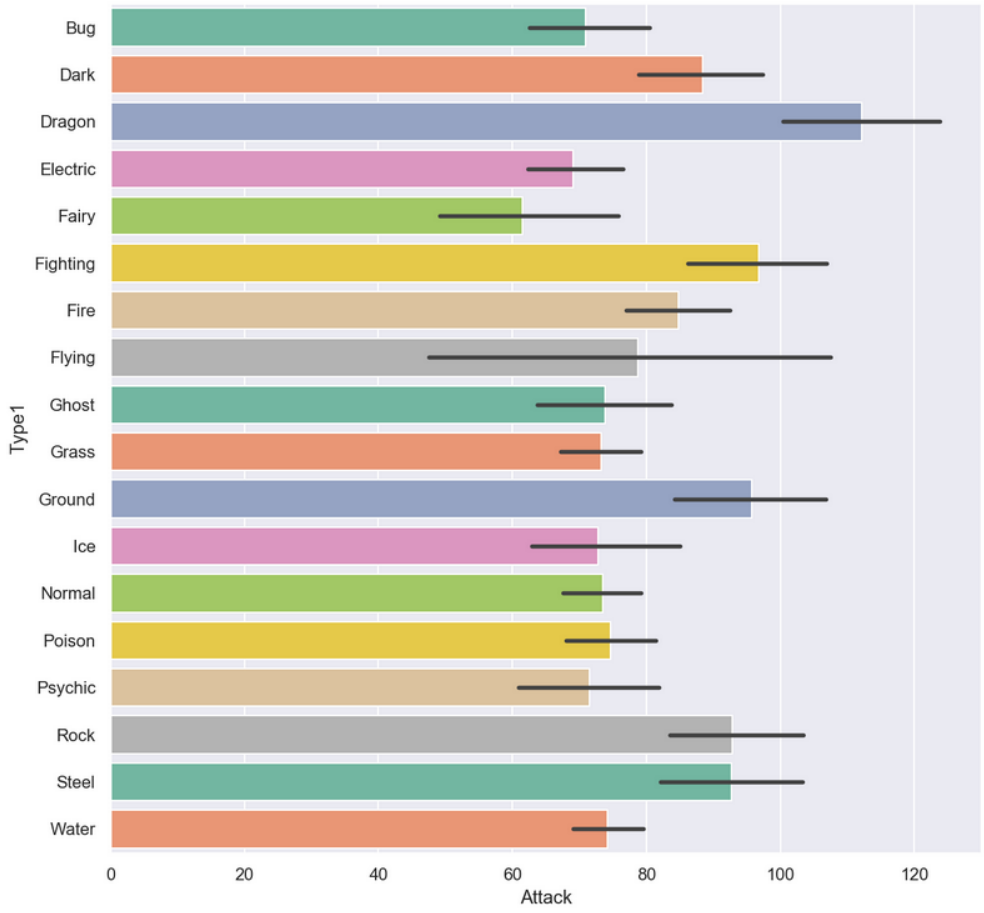
Grafiğe bakıldığında yeni nesil pokemonların (6 ve 5), efsanevi pokemon olmadığı ve saldırı gücünün 25 ile 120 arasında yoğunluk gösterdiği görülmektedir.

```
In [134]: sbr.boxenplot(x="Generation",y="Attack",hue="Legendary",data=df)
          warnings.filterwarnings("ignore")
```



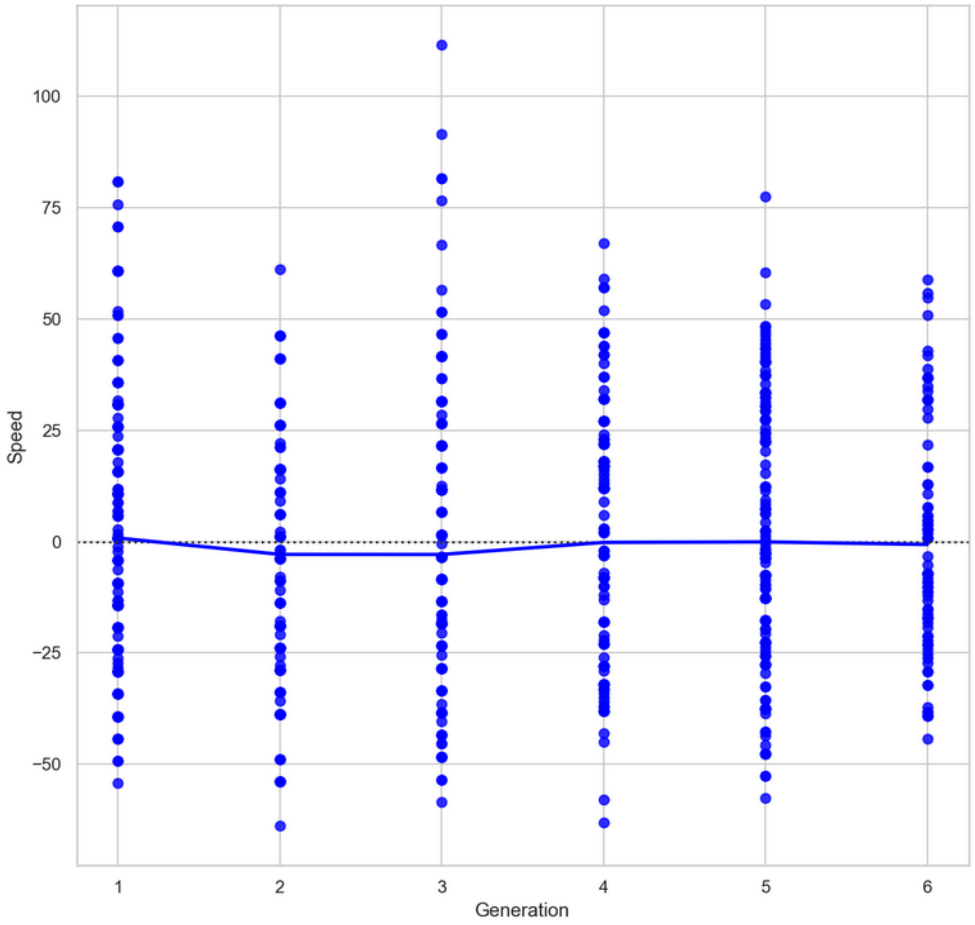
Grafiğe bakıldığında efsanevi ve efsanevi olmayan pokemonların saldırı değişkenine göre aykırı değerlere sahip olduğu görülmektedir. Efsanevi pokemonlardan 1, 2, 3, 4. Jenerasyona sahip olanların saldırıya göre sağa çarpık dağılım gösterdiği, efsanevi olmayan pokemonların ise tüm jenerasyonlara bakıldığında neredeyse simetrik bir dağılım gösterdiği söylenmektedir.

```
In [105]: sbr.set(rc={'figure.figsize':(10,10)})
          sbr.barplot(x=df.Attack,y=df.Type1,data=df,palette="Set2",linewidth=1)
          warnings.filterwarnings("ignore")
```



Grafiğe bakıldığında saldırı gücü olarak Type1 değişkenindeki pokemonlardan en iyi hasarı Dragon tipi pokemon vermektedir.

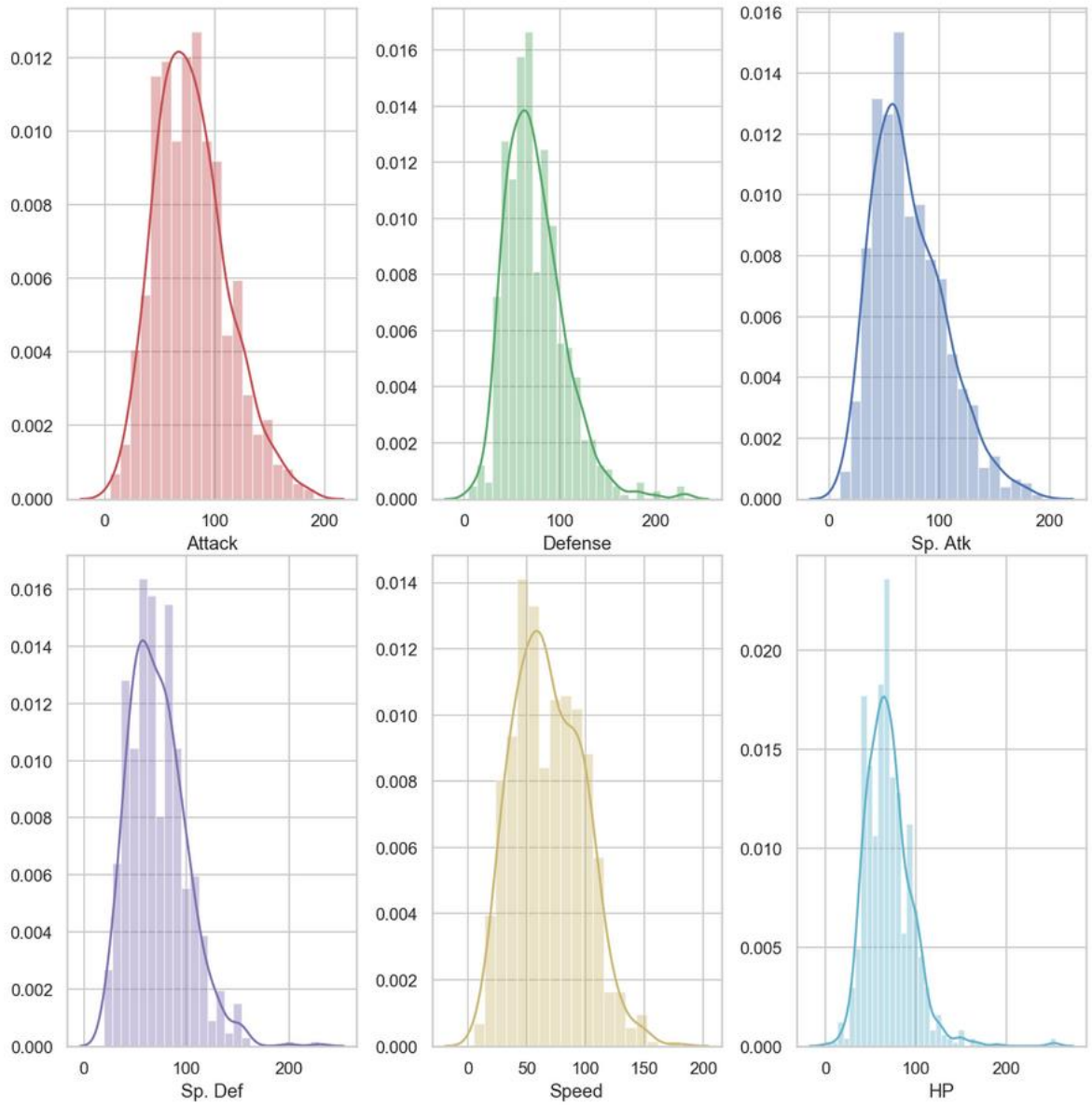
In [133]:
```
sbr.set(style="whitegrid")
sbr.residplot(x=df.Generation,y=df.Speed,lowess=True,color="blue",order=0.01)
warnings.filterwarnings("ignore")
```



Yeni nesil pokemonlar (5 ve 6), diğer nesil pokemonlara göre çok az bir farkla ilk turda saldırı göstermektedir.

In [135]:
```
print("-> İstatistiksel Dağılımlar:")
fig, axes = plt.subplots(nrows=2, ncols=3)
fig.tight_layout()
colors = ['r','g','b','m','y','c']
stats = ['Attack','Defense','Sp. Atk','Sp. Def','Speed','HP']
for idx,clr in enumerate(colors):
    plt.subplot(2, 3, idx+1)
    pltt = sns.distplot(df[stats[idx]],color=clr)
```

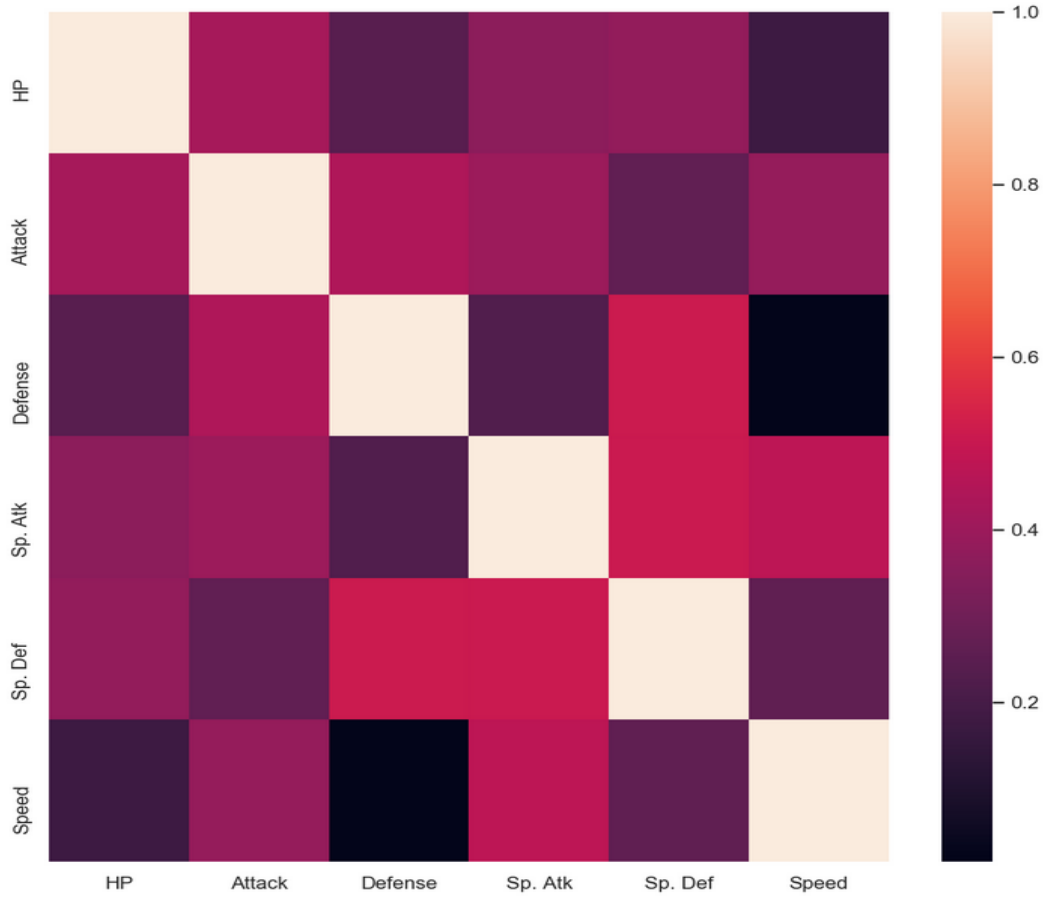
```
-> İstatistiksel Dağılımlar:
```



Değişken tiplerinin dağılım grafikleri yukarıda verilmiştir. Defense, Sp.Def. ve HP değişkenlerinin sağa çarpıklık gösterdiğini, diğer değişkenlerin ise neredeyse normal dağılım gösterdiği görülmektedir.

In [136]:
```
corr = df.iloc[:,5:11].corr()
print("-> Değişkenler Arasındaki Korelasyon Isı Haritası:")
pltt = sns.heatmap(corr, xticklabels=corr.columns.values, yticklabels=corr.columns.values)
```

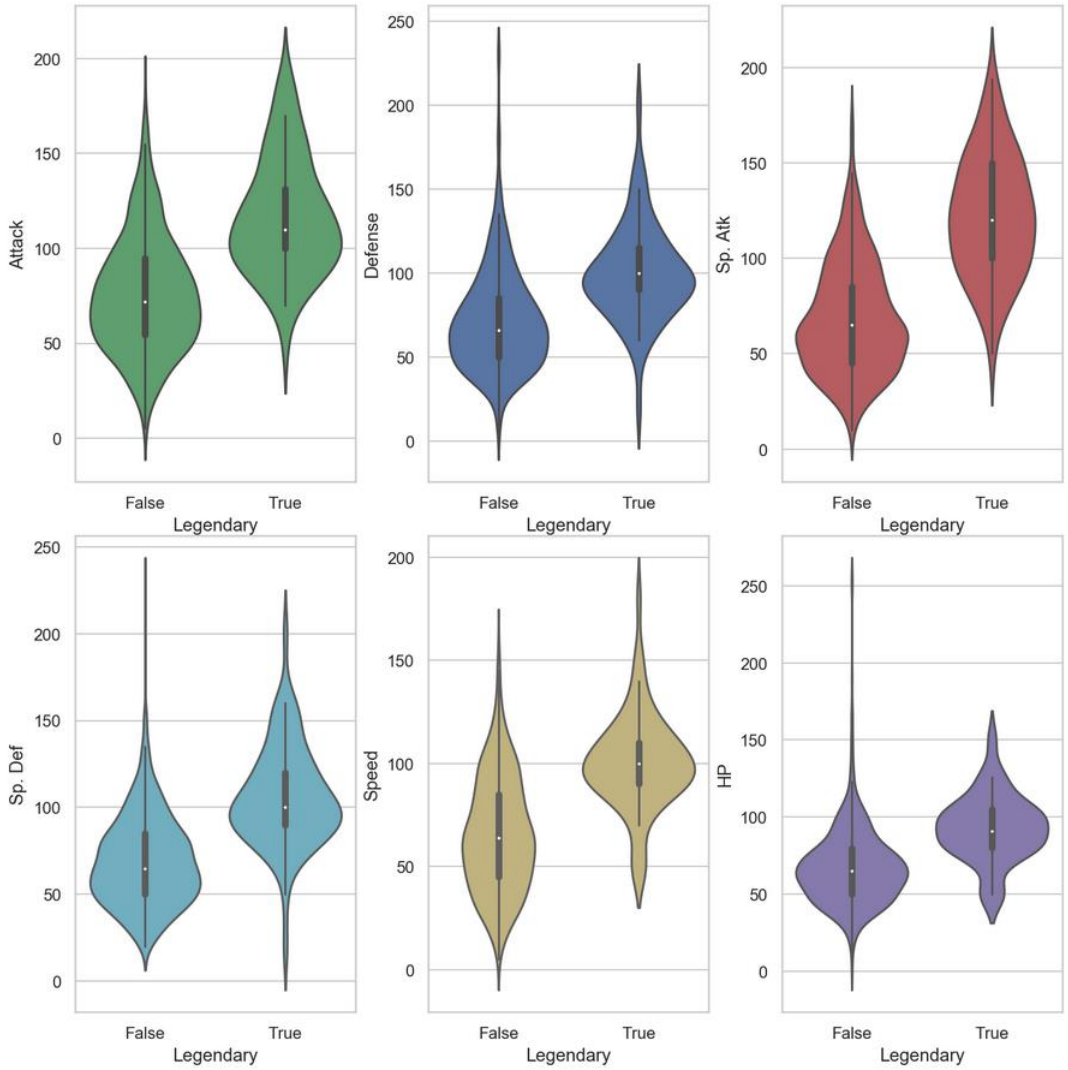
-> Değişkenler Arasındaki Korelasyon Isı Haritası:



Defense ve Speed değişkenleri arasında güçlü bir negatif korelasyon olduğu görülmektedir.

In [137]:
```
print("-> Efsanevi ve Efsanevi Olmayan Pokemonların İstatistiksel Dağılımı")
fig, axes = plt.subplots(nrows=2, ncols=3)
fig.tight_layout()
colors = ['g','b','r','c','y','m']
stats = ['Attack','Defense','Sp. Atk','Sp. Def','Speed','HP']
for idx,clr in enumerate(colors):
    plt.subplot(2, 3, idx+1)
    pltt = sns.violinplot(df['Legendary'],df[stats[idx]],color=clr)
```

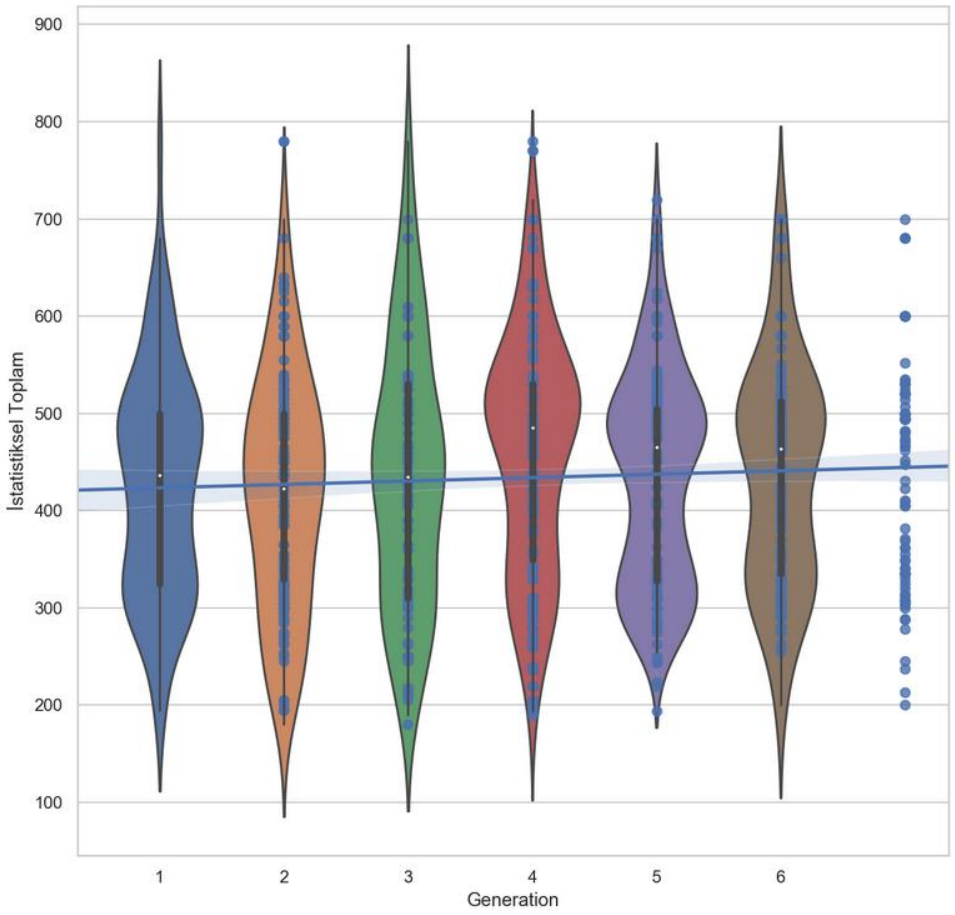
```
-> Efsanevi ve Efsanevi Olmayan Pokemonların İstatistiksel Dağılımı
```



Efsanevi olan pokemonların efsanevi olmayan pokemonlara göre her zaman daha iyi olduğu görülmektedir.

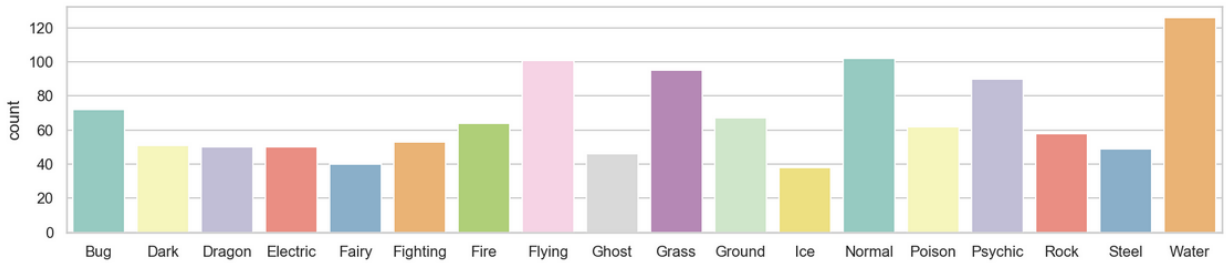
In [138]:
```
print("-> Nesillerin İstatistiksel Toplam Değişkenine Göre Dağılımı")
sum_stats = df["Total"]
pltt = sns.violinplot(df['Generation'],sum_stats)

print("-> Nesillerin İstatistiksel Toplam Değişkenine Göre Regresyonu:")
pltt = sns.regplot(df['Generation'],sum_stats)
pltt = pltt.set_ylabel('İstatistiksel Toplam')
```

```
-> Nesillerin İstatistiksel Toplam Değişkenine Göre Dağılımı
-> Nesillerin İstatistiksel Toplam Değişkenine Göre Regresyonu:
```



Violin grafiğine bakıldığında 1, 4, 5 ve 6. Jenerasyona sahip pokemonların istatistiksel toplam değişkenine göre iki tepeli olduğu görülmektedir.

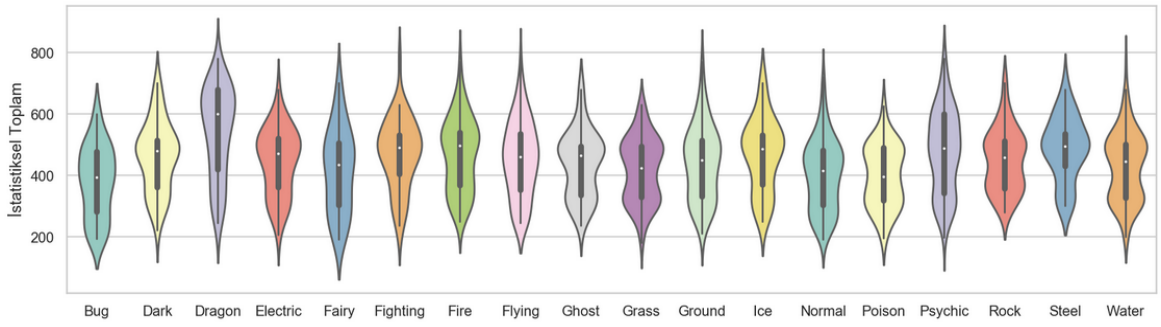
```
In [50]: fig = plt.figure(figsize=(15,3))
         pltt = sns.countplot(df['Type1'].append(df['Type2']), palette="Set3")
```



Tiplere göre grafiğe bakıldığında en çok water grubu pokemonun veri setinde yer aldığı görülmektedir.

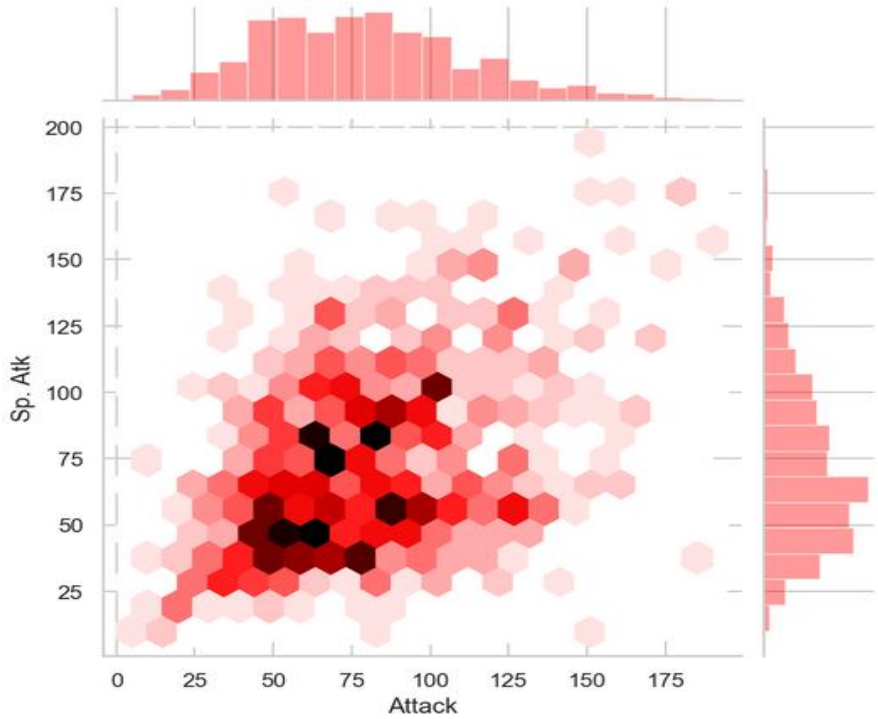
```
In [139]: fig = plt.figure(figsize=(15,4))
          print("-> Türler Arasında Toplam İstatistik Dağılımı:")
          pltt = sns.violinplot(df['Type1'].append(df['Type2']),sum_stats.append(sum_stats),palette="Set3")
          pltt = pltt.set_ylabel('İstatistiksel Toplam')
```

```
-> Türler Arasında Toplam İstatistik Dağılımı:
```



Dragon tipi pokemonların Poisson tipi pokemonlara göre istatistiksel toplam değerinin daha yüksek olduğu görülmektedir ancak Poisson tipi pokemonların varyansı daha büyüktür.

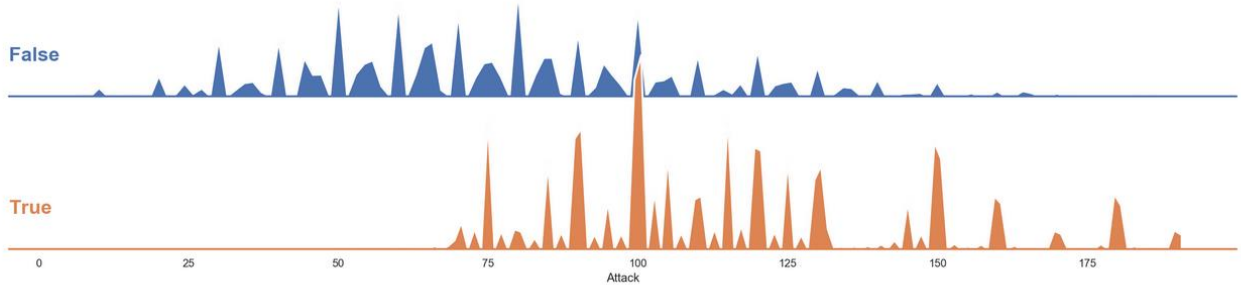
```
In [35]: pltt = sns.jointplot(df['Attack'],df['Sp. Atk'], kind="hex", color='red')
         #aynı grafiğin farklı kodu
         #sbr.jointplot(x="Attack",y="Sp. Atk",data=df,kind="hex",color="green");
         #warnings.filterwarnings("ignore")
```



Grafiğe bakıldığında iki değişken arasında ki korelasyon en çok iki değerinde 50 olduğu bölgelerde yoğunlaşmıştır. Attack ve Sp.Atk. değişkenlerinin arasında yer yer negatif korelasyona da rastlandığı görülmektedir.

In [36]:
```
print("-> Efsanevi Pokemon ve Efsanevi Olmayan Pokemonların Saldırı Dağılımı")
sns.set(style="white", rc={"axes.facecolor": (0, 0, 0, 0)})
g = sns.FacetGrid(df, row="Legendary", hue="Legendary", aspect=6)
g.map(sns.kdeplot, "Attack", clip_on=False, shade=True, alpha=1, lw=1.5, bw=.2)
g.map(sns.kdeplot, "Attack", clip_on=False, color="w", lw=2, bw=.2)
g.map(plt.axhline, y=0, lw=2, clip_on=False)
def label(x, color, label):
    ax = plt.gca()
    ax.text(0, .2, label, size=20, fontweight="bold", color=color, ha="left", va="center", transform=ax.transAxes)
g.map(label, "Attack")
g.fig.subplots_adjust(hspace=-.25)
g.set_titles("")
g.set(yticks=[])
g.despine(bottom=True, left=True)
plt.show()
```

```
-> Efsanevi Pokemon ve Efsanevi Olmayan Pokemonların Saldırı Dağılımı
```



Efsanevi pokemonların saldırı gücünün efsanevi olmayan pokemonlara göre daha iyi olduğu görülmektedir.

PYTHON İLE MAKİNE ÖĞRENMESİ

In [64]:
```
from sklearn.preprocessing import LabelEncoder

df["Types Num"] = 1+df.iloc[:,3:4].count(axis=1)
df["Stats Median"] = df.iloc[:,5:11].median(axis=1)
df["MaxMin Ratio"] = df.iloc[:,5:11].max(axis=1)/df.iloc[:,5:11].min(axis=1)
df["Max Stat"] = df.iloc[:,5:11].idxmax(axis=1)
df["Min Stat"] = df.iloc[:,5:11].idxmin(axis=1)
df["Is Mega"] = df.iloc[:,1].apply(lambda x: x.find('Mega')>0)
df.head(15)
```

Out[64]:

	#	Name	Type1	Type2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Types Num	Stats Median	MaxMin Ratio	Max Stat	Min Stat	Is Mega
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False	2	49.0	1.444444	Sp. Atk	HP	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False	2	62.5	1.333333	Sp. Atk	HP	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False	2	82.5	1.250000	Sp. Atk	HP	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False	2	110.0	1.537500	Defense	HP	True
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False	1	51.0	1.666667	Speed	HP	False
5	5	Charmeleon	Fire	NaN	405	58	64	58	80	65	80	1	False	1	64.5	1.379310	Sp. Atk	HP	False
6	6	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	False	2	84.5	1.397436	Sp. Atk	HP	False
7	6	CharizardMega Charizard X	Fire	Dragon	634	78	130	111	130	85	100	1	False	2	105.5	1.666667	Attack	HP	True
8	6	CharizardMega Charizard Y	Fire	Flying	634	78	104	78	159	115	100	1	False	2	102.0	2.038462	Sp. Atk	HP	True
9	7	Squirtle	Water	NaN	314	44	48	65	50	64	43	1	False	1	49.0	1.511628	Defense	Speed	False
10	8	Wartortle	Water	NaN	405	59	63	80	65	80	58	1	False	1	64.0	1.379310	Defense	Speed	False
11	9	Blastoise	Water	NaN	530	79	83	100	85	105	78	1	False	1	84.0	1.346154	Sp. Def	Speed	False
12	9	BlastoiseMega Blastoise	Water	NaN	630	79	103	120	135	115	78	1	False	1	109.0	1.730769	Sp. Atk	Speed	True
13	10	Caterpie	Bug	NaN	195	45	30	35	20	20	45	1	False	1	32.5	2.250000	HP	Sp. Atk	False
14	11	Metapod	Bug	NaN	205	50	20	55	25	25	30	1	False	1	27.5	2.750000	Defense	Attack	False

6 tane yeni değişken veri setine eklenmiştir. Bunlar kısaca;

Types Num: Pokemonun kaç tipe sahip olduğu

Stats Media: Medyan değeri

MaxMin Ratio: Maksimum ve minimum değer oranı

Max Stat: Maksimum değere sahip değişken

Min Stat: Minimum değere sahip değişken

İs Mega: Pokemonun Mega özel gücünün olup olmadığı

Burada gözetimli makine öğrenmesi uygulanmaktadır. Pokemonların efsanevi pokemon olup olmama durumunu en iyi hangi modelin açıklayabildiği araştırılmaktadır.

In [56]:
```
# Label strings
df["Max Stat"] = df[["Max Stat"]].apply(LabelEncoder().fit_transform)
df["Min Stat"] = df[["Min Stat"]].apply(LabelEncoder().fit_transform)
```

In [57]:
```
from sklearn.model_selection import cross_val_score
from sklearn import linear_model
from sklearn import svm
from sklearn import ensemble
```

In [58]:
```
#gözetimli öğrenme yapıyoruz.
clf_list = [linear_model.LogisticRegression(),svm.SVC(),ensemble.RandomForestClassifier(random_state=123, n_estimators=1000)]
model_list = ['Logistic Regression', 'SVM', 'Random Forest Classifier']
features_arr = ["Total","Is Mega","Stats Median","MaxMin Ratio","Types Num"]
```

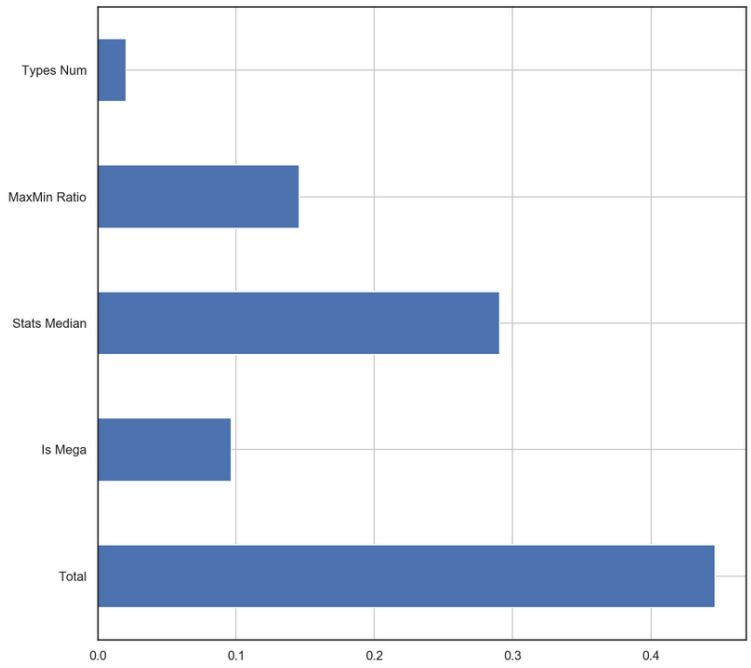
Lojistik regresyon, SVM ve Random Forest modelleri kullanılmaktadır. Başlamak için birkaç parametre seçilmiştir.

In [59]:
```
for idx,clf in enumerate(clf_list):
    print("Trying Model ", model_list[idx])
    scores = cross_val_score(clf, df[features_arr].values, df["Legendary"].values, cv=10)
    print('>> Mean CV score is: ', round(np.mean(scores),3))
```

```
Trying Model  Logistic Regression
>> Mean CV score is:  0.949
Trying Model  SVM
>> Mean CV score is:  0.944
Trying Model  Random Forest Classifier
>> Mean CV score is:  0.959
```

Çıktıya bakıldığında en iyi sonucu veren Random Forest algoritmasıdır.

In [60]:
```
clf = clf_list[2] # en iyi olan random forest
clf.fit(df[features_arr].values, df["Legendary"].values)

feat_importances = pd.Series(clf.feature_importances_, index=features_arr)
feat_importances.plot(kind='barh',grid=True)
plt.show()
```



Random Forest modelinde en önemli değişkenin (başka bir ifadeyle en yüksek tahmin gücüne sahip olan değişken) istatistiksel toplam değeri olduğunu söyleyebiliriz.

```
In [61]: scores = cross_val_score(clf, df[features_arr+["Max Stat","Min Stat"]].values, df["Legendary"].values, cv=10)
         print('>> Mean CV score is: ', round(np.mean(scores),3))

         >> Mean CV score is:  0.965
```

Modelin içerisine maksimum ve minimum değere sahip değişkenler eklendiğinde Random Forest daha iyi bir sonuç vermektedir.

Pokemonun efsanevi pokemon olmasındaki en önemli etkenler; gücü (istatistiksel toplam değeri), maksimum değere sahip değişkeni ve minimum değere sahip değişkeni olduğunu söylemek mümkündür.