In [16]:
```
!pip install altair
```

```
Looking in indexes: https://pypi.org/simple, https://pip.repos.neuron.amazonaws.com
Requirement already satisfied: altair in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (4.1.0)
Requirement already satisfied: toolz in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from alt
air) (0.11.1)
Requirement already satisfied: pandas>=0.18 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (f
rom altair) (1.1.5)
Requirement already satisfied: jsonschema in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (fro
m altair) (3.2.0)
Requirement already satisfied: entrypoints in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (fr
om altair) (0.3)
Requirement already satisfied: jinja2 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from al
tair) (2.11.3)
Requirement already satisfied: numpy in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from alt
air) (1.19.5)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-p
ackages (from pandas>=0.18->altair) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (f
rom pandas>=0.18->altair) (2021.1)
Requirement already satisfied: six>=1.5 in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-packages (from
python-dateutil>=2.7.3->pandas>=0.18->altair) (1.15.0)
```

In [130]:
```
import altair as alt
import pandas as pd
```

In [131]:
```
csv_url = 'https://gist.githubusercontent.com/armgilles/194bcff35001e7eb53a2a8b441e8b2c6/raw/92200bc0a673d5ce2110aaad45
```

In [132]:
```
df = pd.read_csv(csv_url, na_filter=False)
```

In [133]:
```
df
```

Out[133]:

	#	Name	Type 1	Type 2	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		309	39	52	43	60	50	65	1	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
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

800 rows × 13 columns

We have extracted the data that we will need for vis.

In [134]:
```
df.drop(['#','Type 2','Total','HP','Sp. Atk','Sp. Def','Speed','Generation','Legendary'], inplace=True, axis=1)
df.sample()
```

Out[134]:

	Name	Type 1	Attack	Defense
304	Kirlia	Psychic	35	35

In [135]:
```
df
```

Out[135]:

	Name	Type 1	Attack	Defense
0	Bulbasaur	Grass	49	49
1	Ivysaur	Grass	62	63
2	Venusaur	Grass	82	83
3	VenusaurMega Venusaur	Grass	100	123
4	Charmander	Fire	52	43
...	...	...	...	...
795	Diancie	Rock	100	150
796	DiancieMega Diancie	Rock	160	110
797	HoopaHoopa Confined	Psychic	110	60
798	HoopaHoopa Unbound	Psychic	160	60
799	Volcanion	Fire	110	120

800 rows × 4 columns

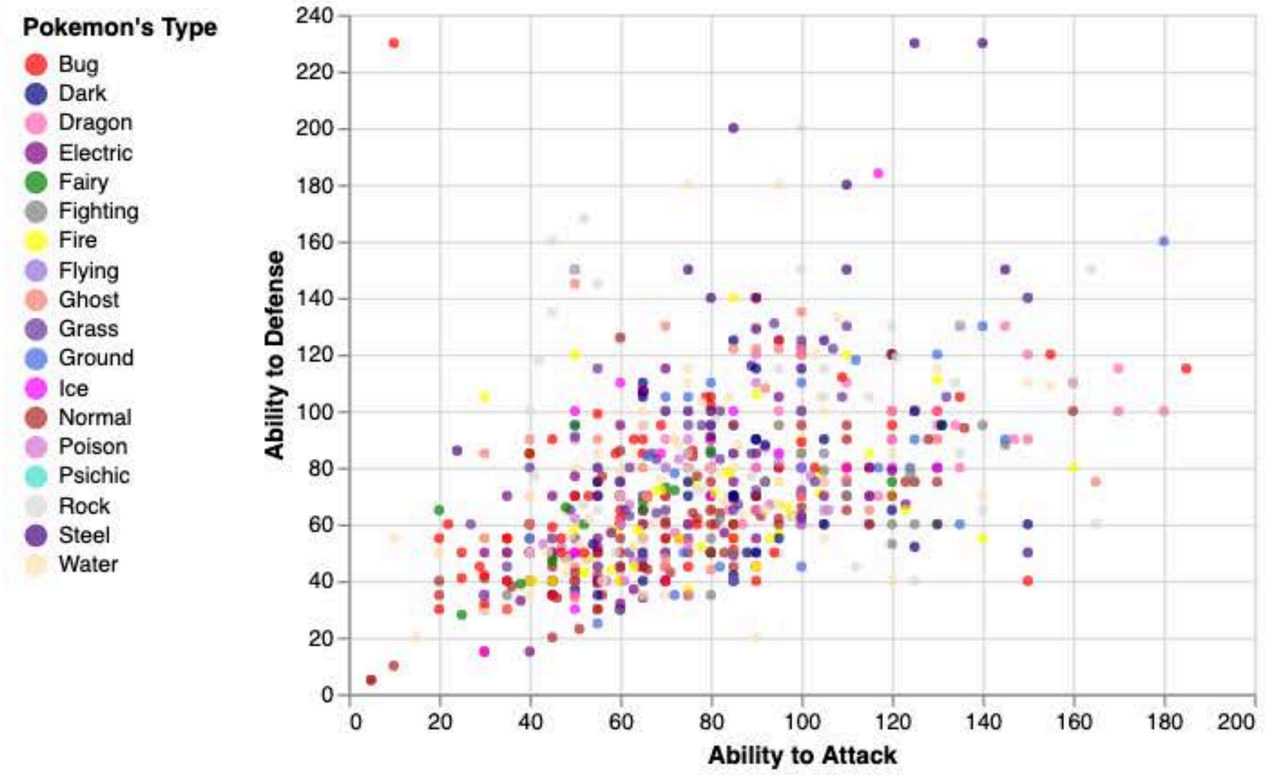
Assigning a unique colour to each new parameter from the list we are requesting.

As we can see it works, but put data manually is not the most efficient way to do so.

In [137]:
```
circle1 = alt.Chart(df).mark_circle(size = 20).encode(
    alt.X('Attack:Q',
          title = 'Ability to Attack'),
    alt.Y('Defense:Q',
          title = 'Ability to Defense'),
    alt.Color('Type 1:N',
              scale = alt.Scale(domain = ['Bug','Dark','Dragon','Electric','Fairy','Fighting','Fire','Flying','Ghost',
                                          'Grass','Ground','Ice','Normal','Poison','Psichic','Rock','Steel','Water'],
                                range = ['red','navy','hotpink','purple','green','gray','yellow','mediumpurple','salmon
                                orient = 'left')
             )
).interactive()

circle1
```

Out[137]:



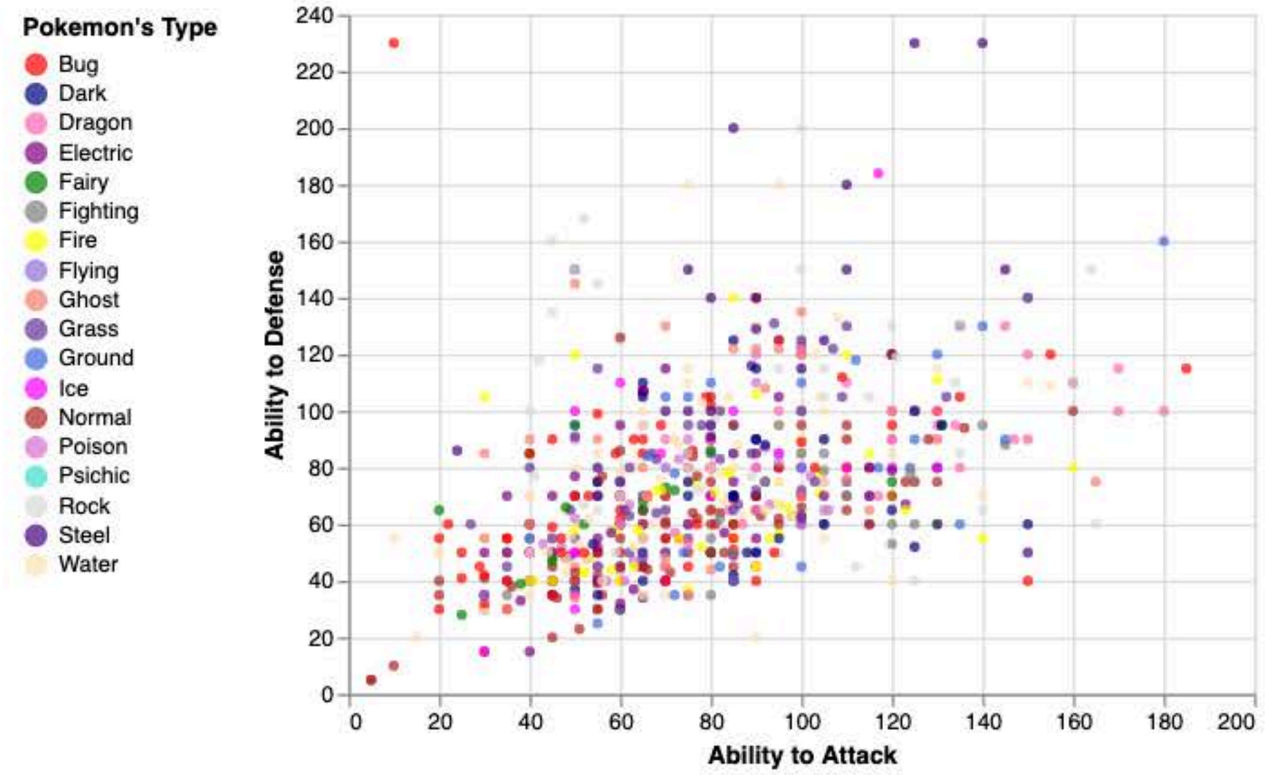
Assigning a unique colour to each new parameter from the list we are requesting.

As we can see it works, but put data manually is not the most efficient way to do so.

In [137]:
```
circle1 = alt.Chart(df).mark_circle(size = 20).encode(
    alt.X('Attack:Q',
          title = 'Ability to Attack'),
    alt.Y('Defense:Q',
          title = 'Ability to Defense'),
    alt.Color('Type 1:N',
              scale = alt.Scale(domain = ['Bug','Dark','Dragon','Electric','Fairy','Fighting','Fire','Flying','Ghost',
                                          'Grass','Ground','Ice','Normal','Poison','Psichic','Rock','Steel','Water'],
                                range = ['red','navy','hotpink','purple','green','gray','yellow','mediumpurple','salmon
                                orient = 'left')
             )
).interactive()

circle1
```

Out[137]:



In Altair it is possible to assign a unique colour to each new parameter from the requested list.

In [138]:
```
circle2 = alt.Chart(df).mark_circle(size = 20).encode(
    alt.X('Attack:Q',
          title = 'Ability to Attack'),
    alt.Y('Defense:Q',
          title = 'Ability to Defense'),
    alt.Color('Type 1:N')
).interactive()

circle2
```

Out[138]:

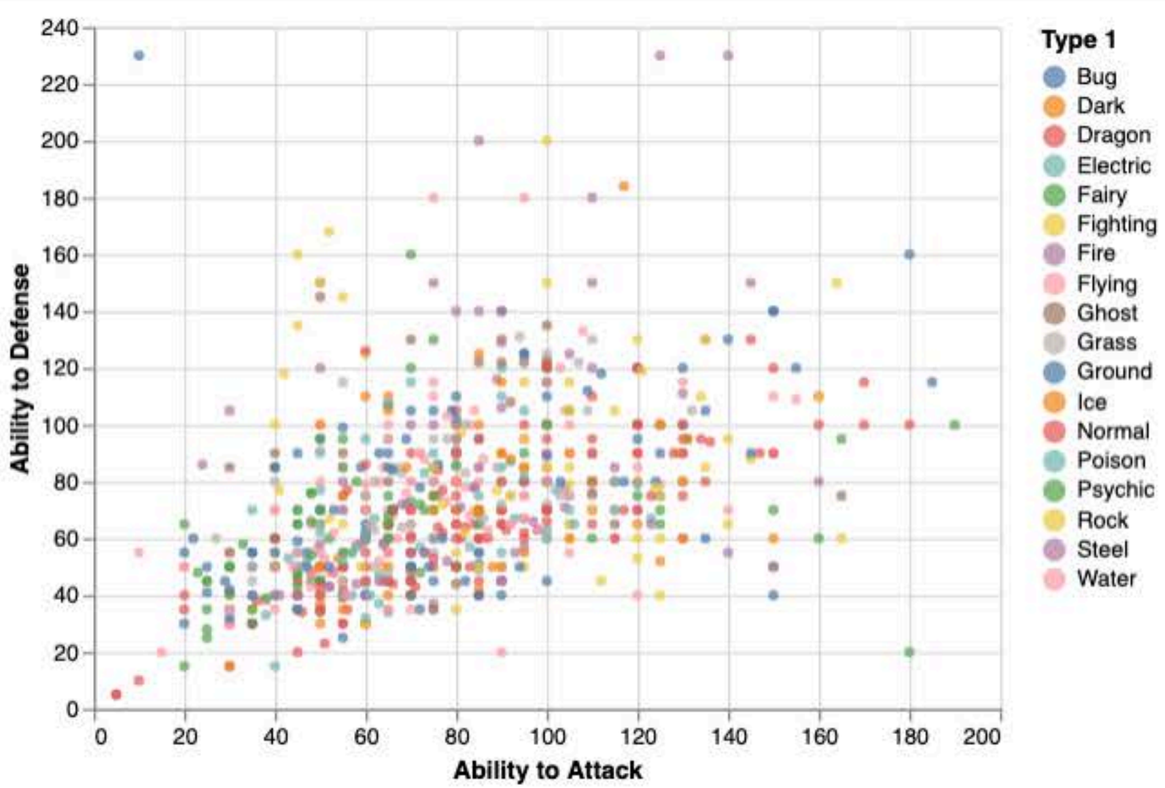


Chart divide Pokemon by their ability to Defence

the bar chart have also been plotted for comparison. but as we can see, this type of graph is not very suitable for this case, because we need two graphs to reporesent the data.

In [139]:
```
bars1 = alt.Chart(df).mark_bar().encode(
    y = 'Defense:N',
    color = alt.Color('Type 1:N',legend = alt.Legend(title = 'Pokemon\'s Type')),
    x = 'count(Type 1):Q',
).interactive()

bars1
```

Out[139]:

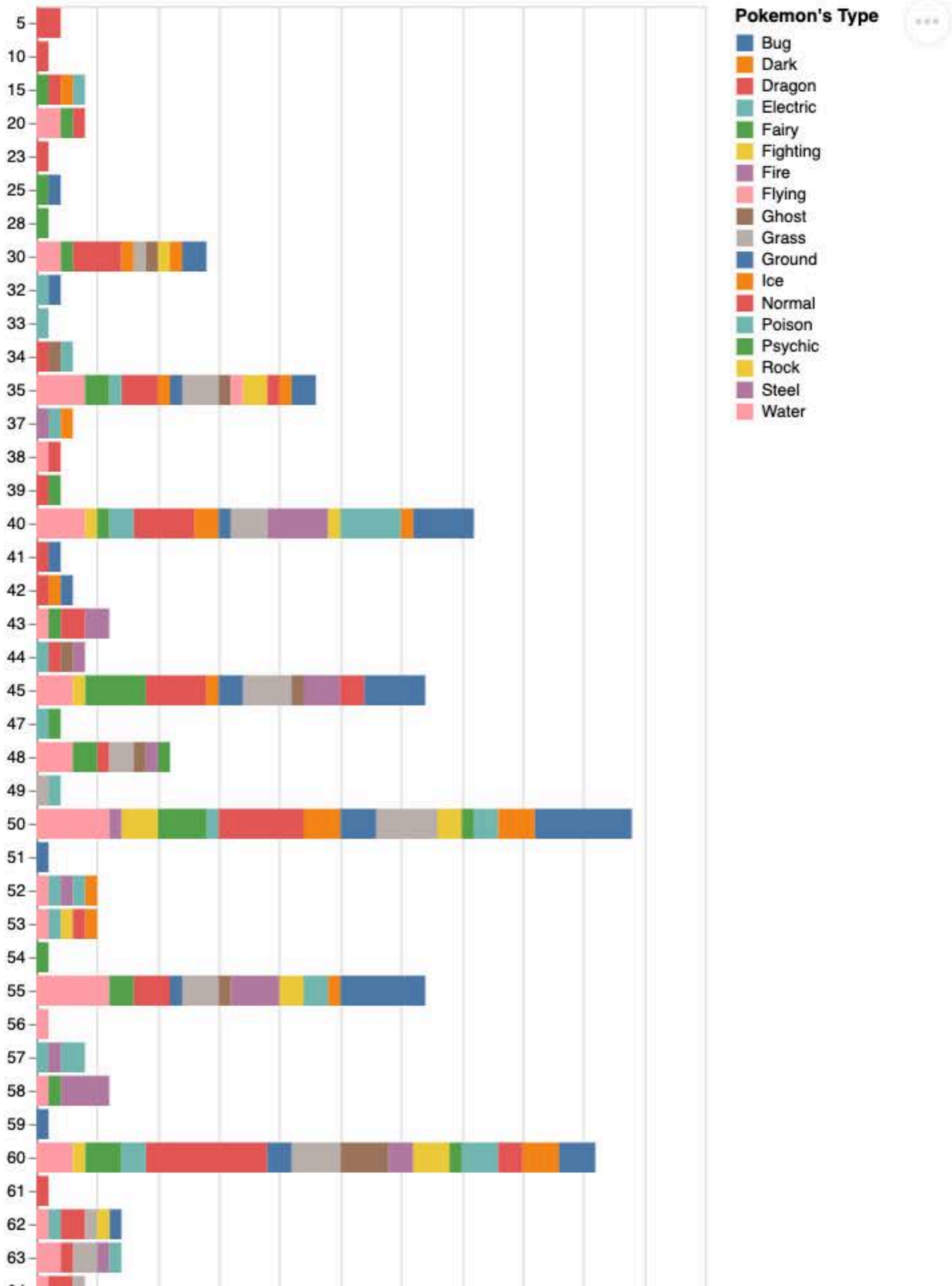
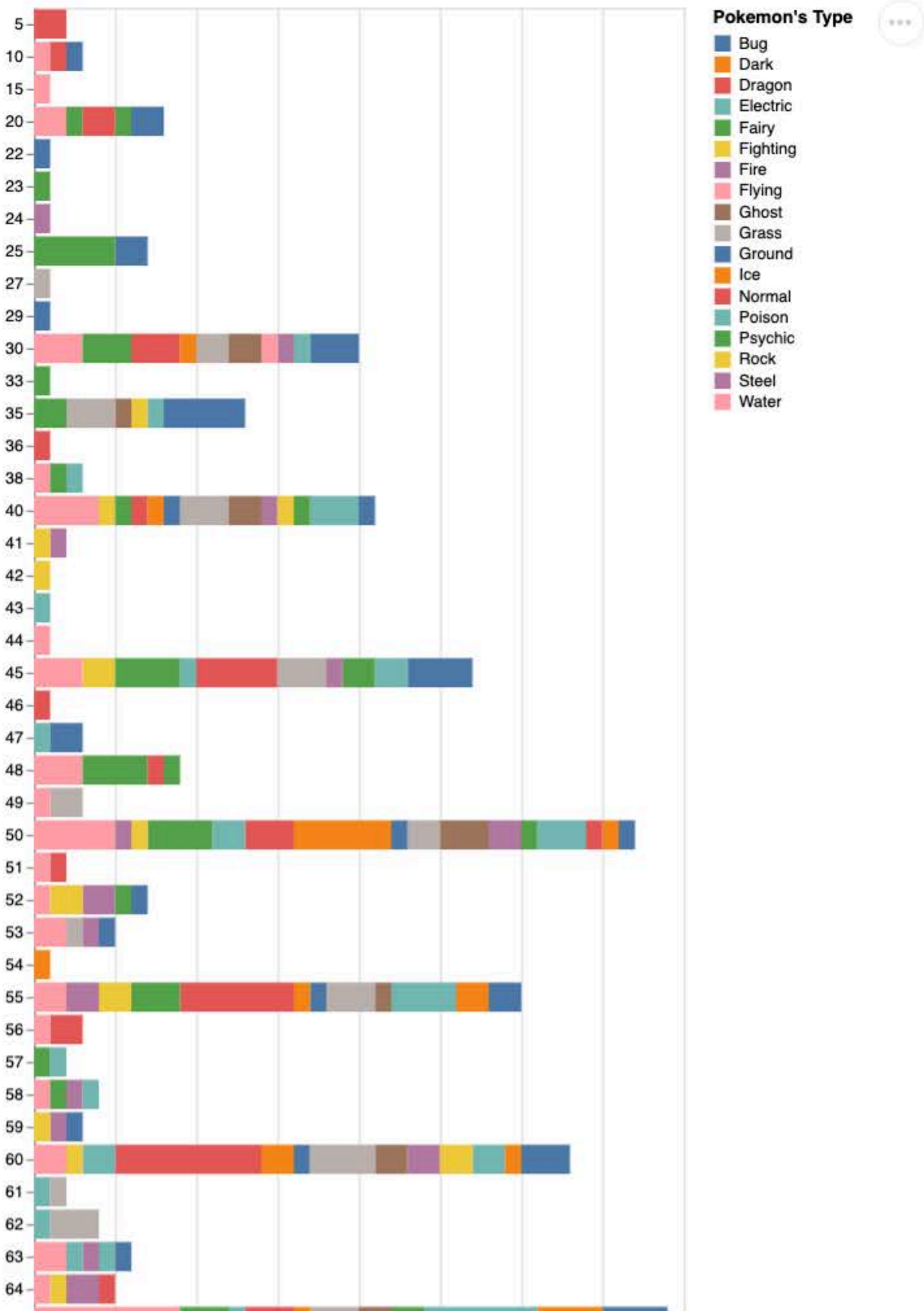


Chart divide Pokemon by their ability to Attack

In [140]:
```
bars2 = alt.Chart(df).mark_bar().encode(
    y = 'Attack:N',
    color = alt.Color('Type 1:N',legend = alt.Legend(title = 'Pokemon\'s Type')),
    x = 'count(Type 1):Q',
).interactive()

bars2
```

Out[140]:



In [141]:
```
circle2 & bars1 & bars2
```

Out[141]: