```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib as mpl
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecat
  import pandas.util.testing as tm
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491h

Enter your authorization code:
··········
Mounted at /content/drive
```

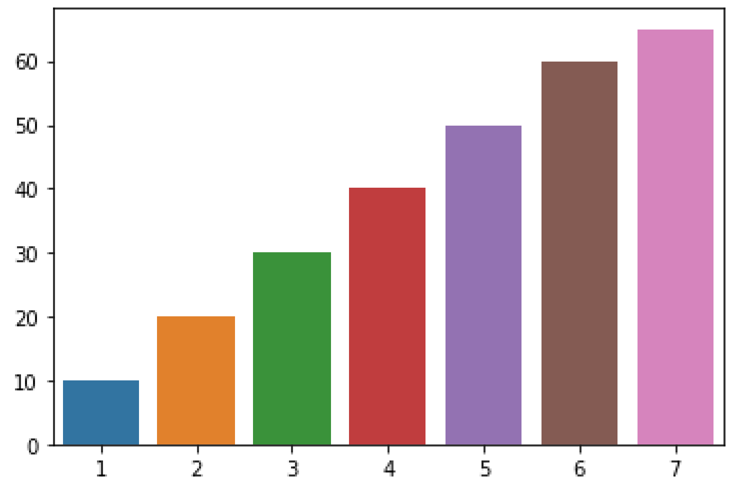
Bar Plot

Bar Plot shows the relationship between a numerical variable and a categorical variable.

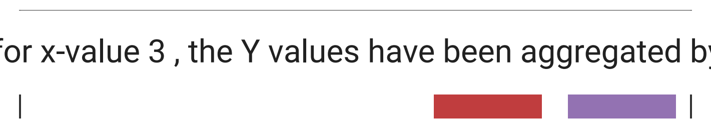
```
# Recover default matplotlib settings
mpl.rcParams.update(mpl.rcParamsDefault)
%matplotlib inline
```

```
x1 = [1,2,3,4,5,6,7]
```

```
y1 = [10,20,30,40,50,60,65]
sns.barplot(x=x1,y=y1)
plt.show()
```



```
x1 = [1,2,3,3,3,4,5]
y1 = [10,20,30,40,50,60,65]
sns.barplot(x=x1,y=y1)
plt.show()
```

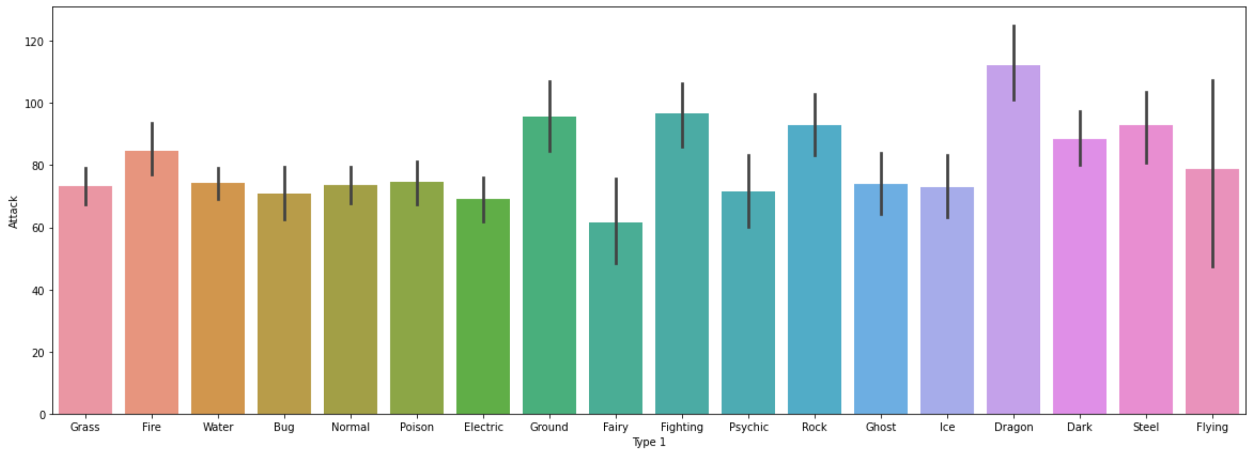


Notice for x-value 3 , the Y values have been aggregated by performing mean of 30,40,50

```
pokemon = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seaborn
pokemon.head(10)
```

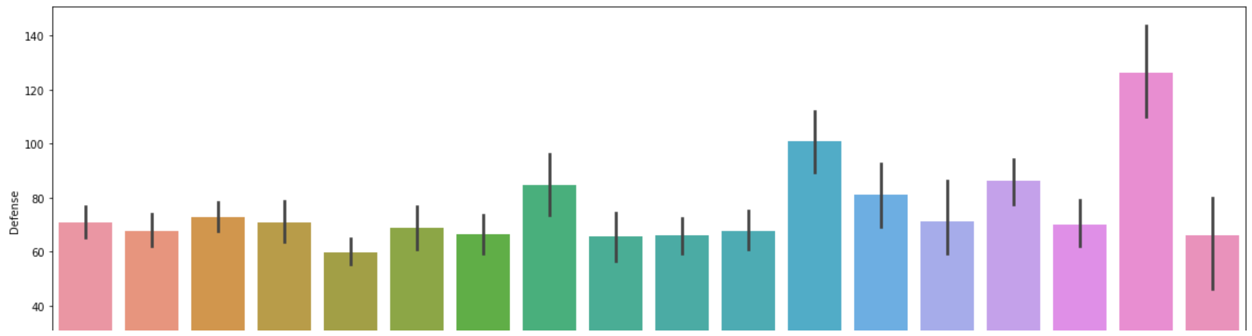
	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False	318
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False	309
5	5	Charmeleon	Fire	NaN	58	64	58	80	65	80	1	False	405
6	6	Charizard	Fire	Flying	78	84	78	109	85	100	1	False	534
7	6	CharizardMega Charizard X	Fire	Dragon	78	130	111	130	85	100	1	False	634

```
plt.figure(figsize=(20,7))
sns.barplot(x=pokemon['Type 1'], y= pokemon['Attack'])
plt.show()
```



As per the above Bar plot pokemon with Type-1 as $Dragon$ are best attackers

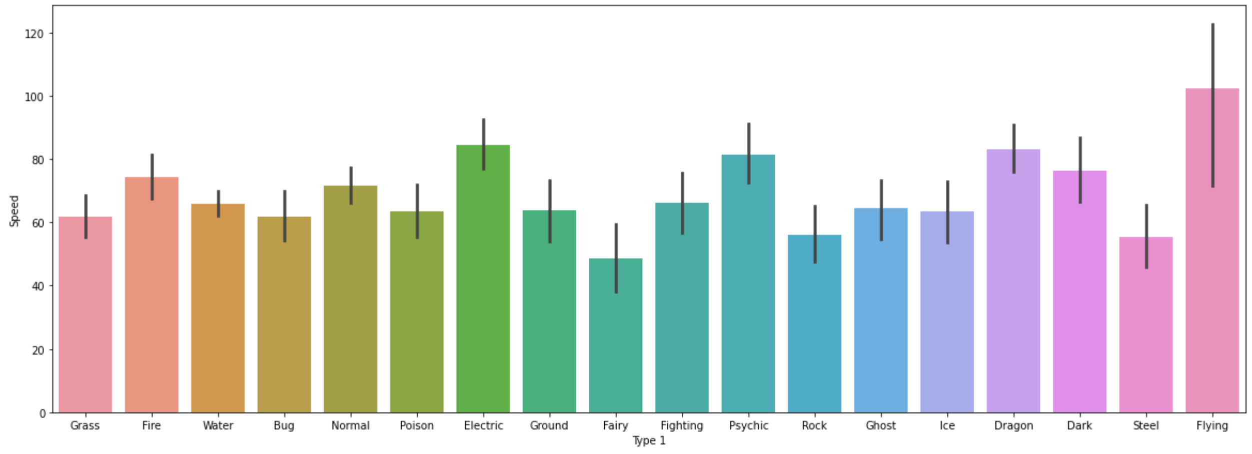
```
plt.figure(figsize=(20,7))
sns.barplot(x=pokemon['Type 1'], y= pokemon['Defense'])
plt.show()
```



```
# Sorted Bar plot
plt.figure(figsize=(20,7))
order = pokemon.groupby(['Type 1']).mean().sort_values('Defense' , ascending = False).i
sns.barplot(x=pokemon['Type 1'], y= pokemon['Defense'] , order=order)
plt.show()
```

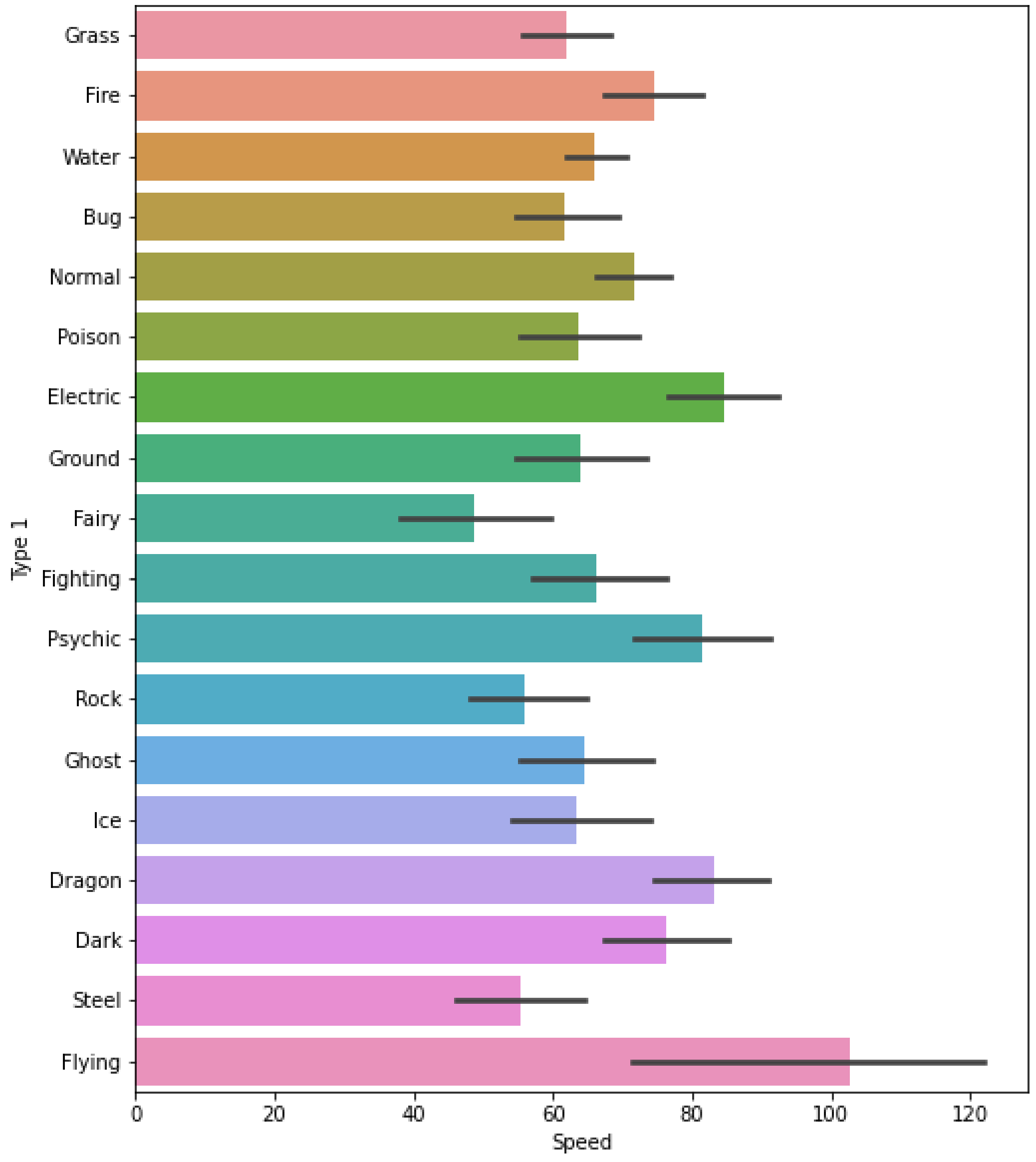
Pokemon with Type-2 as Steel have best defence

```
plt.figure(figsize=(20,7))
sns.barplot(x=pokemon['Type 1'], y= pokemon['Speed'])
plt.show()
```

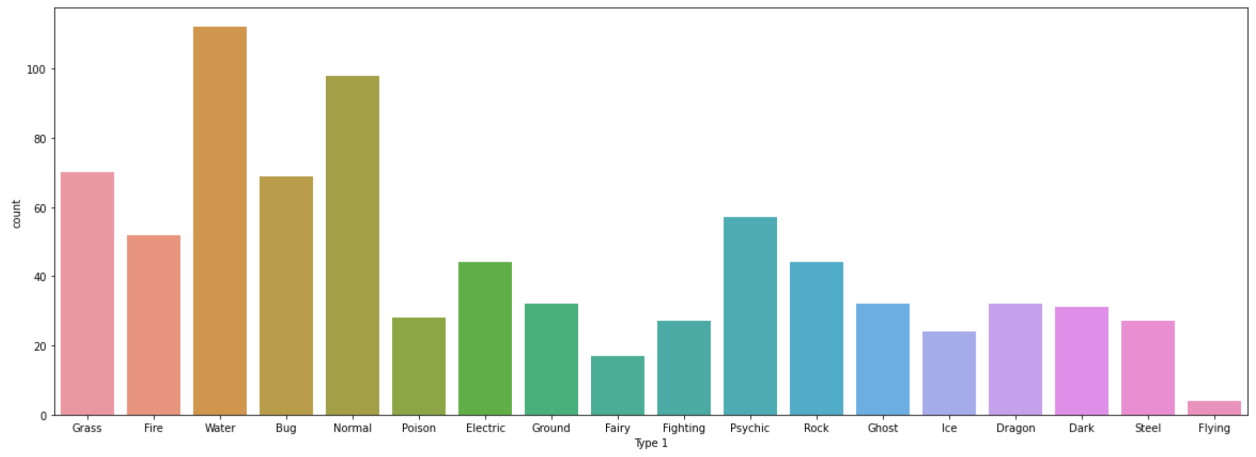


Pokemons with Type-1 as Flying are the fastest

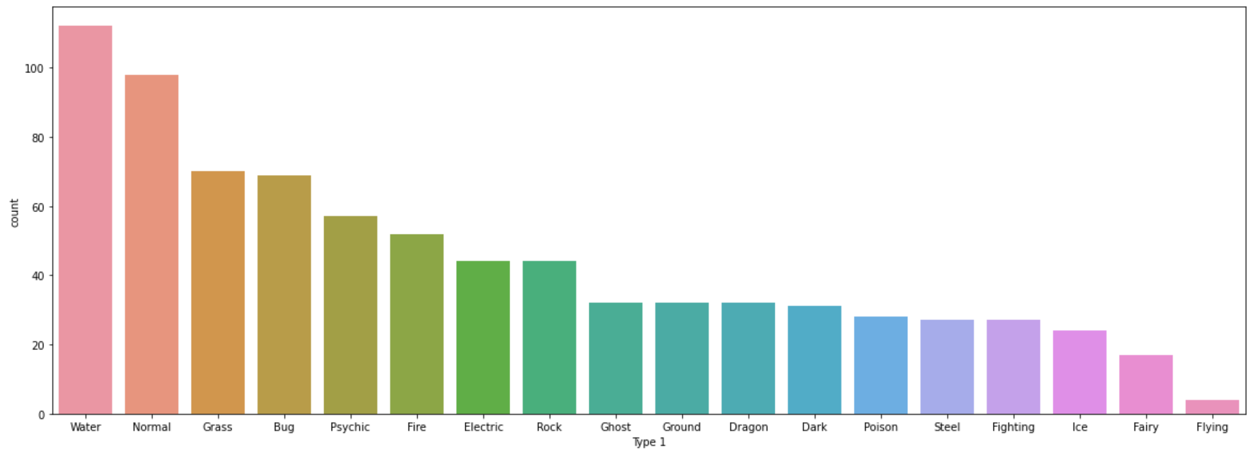
```
# Horizontal Bar plot
plt.figure(figsize=(8,10))
plt.figure(figsize=(8,10))
sns.barplot(x=pokemon['Speed'], y= pokemon['Type 1'])
plt.show()
```



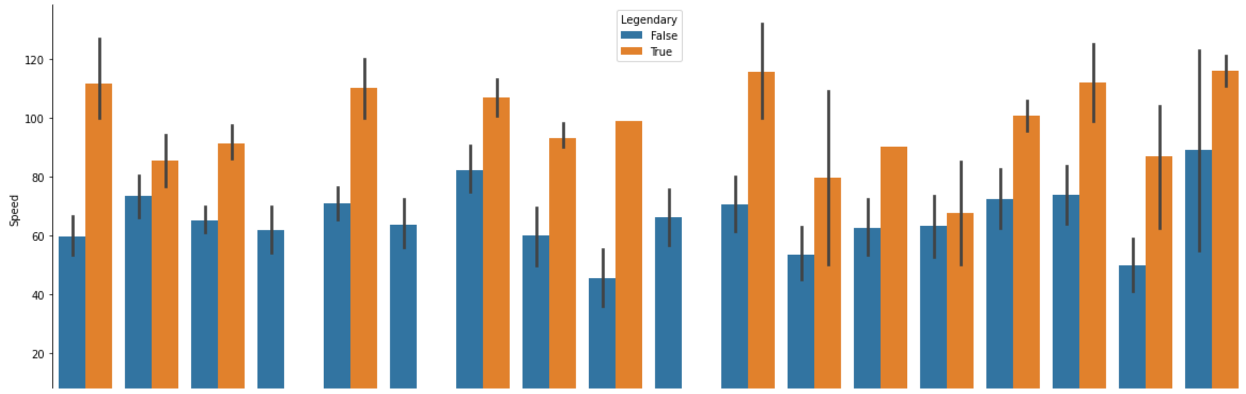
```
plt.figure(figsize=(20,7))
sns.countplot(x=pokemon['Type 1'])
plt.show()
```



```
plt.figure(figsize=(20,7))
sns.countplot(x=pokemon['Type 1'] , order = pokemon['Type 1'].value_counts().index)
plt.show()
```



```
plt.figure(figsize=(20,7))
sns.barplot(x=pokemon['Type 1'], y= pokemon['Speed'],hue=pokemon['Legendary'])
sns.despine() # right and top lines can be removed
plt.show()
```



Pokemon with Type-1 as water & Normal are most common.

```
#Changing the background of bar plot
plt.figure(figsize=(20,7))
sns.set(rc={"axes.facecolor":"#283747", "axes.grid":False,'xtick.labelsize':14,'ytick.l
sns.countplot(x=pokemon['Type 1'] , order = pokemon['Type 1'].value_counts().index)
plt.show()
```

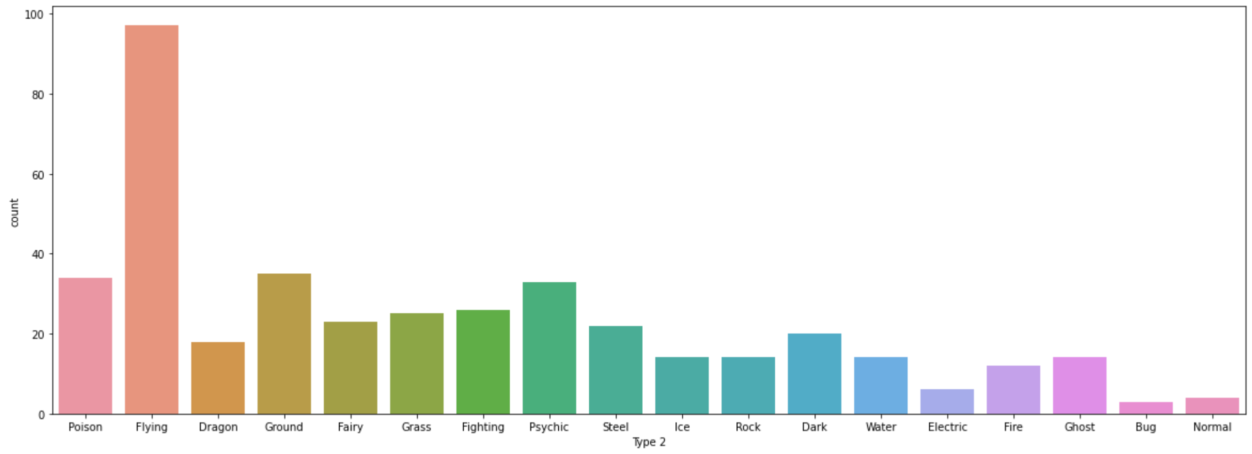
100
80
60
count

```
plt.figure(figsize=(20,7))
sns.set(rc={"axes.facecolor":"#283747", "axes.grid":False,'xtick.labelsize':14,'ytick.l
plt.gcf().text(.5, .93, "Bar Plot", fontsize = 60, color='Black' ,ha='center', va='cent
sns.countplot(x=pokemon['Type 1'] , order = pokemon['Type 1'].value_counts().index , pa
plt.show()
```

Bar Plot

```
mpl.rcParams.update(mpl.rcParamsDefault)
%matplotlib inline
```

```
plt.figure(figsize=(20,7))
sns.countplot(x=pokemon['Type 2'])
plt.show()
```

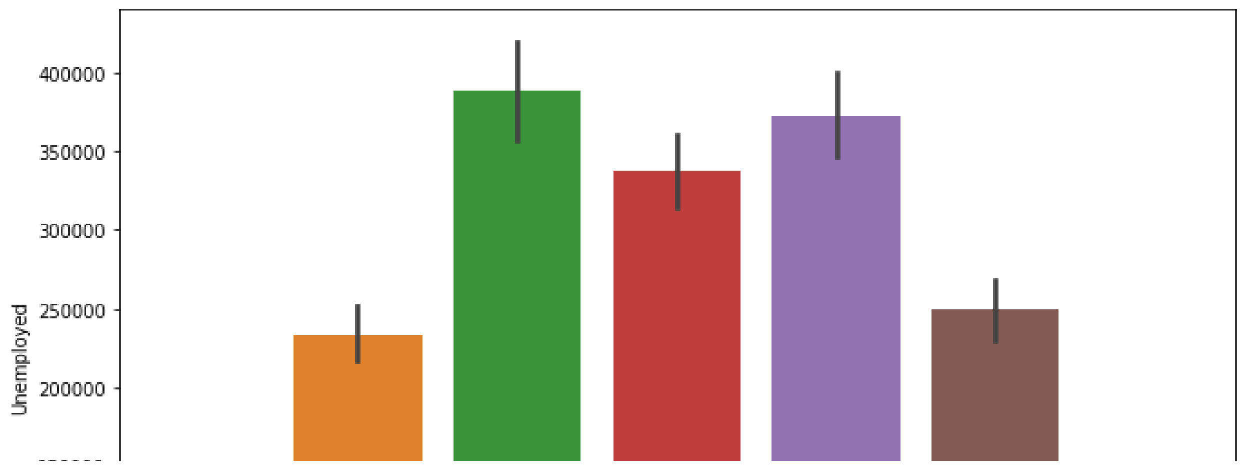


As per above Count plot Type-2 Flying pokemon are most common

```
employment = pd.read_excel("/content/drive/My Drive/Python DataScience/Visualization/Se
employment.head(10)
```

	Age	Gender	Period	Unemployed
0	16 to 19 years	Men	2005-01-01	91000
1	20 to 24 years	Men	2005-01-01	175000
2	25 to 34 years	Men	2005-01-01	194000
3	35 to 44 years	Men	2005-01-01	201000
4	45 to 54 years	Men	2005-01-01	207000
5	55 to 64 years	Men	2005-01-01	101000
6	65 years and over	Men	2005-01-01	33000
7	16 to 19 years	Women	2005-01-01	38000
8	20 to 24 years	Women	2005-01-01	90000
9	25 to 34 years	Women	2005-01-01	142000

```
plt.figure(figsize=(11,7))
sns.barplot(x="Age",y="Unemployed", data=employment)
plt.show()
```

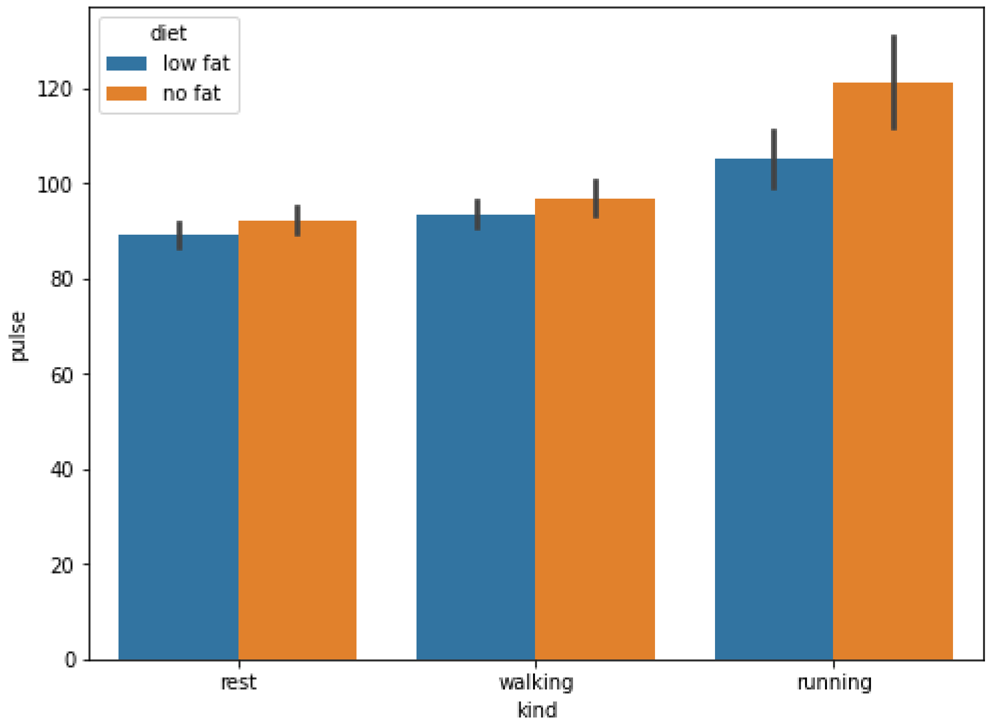


```
exercise = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seabor
exercise.head(10)
```

	id	diet	pulse	time	kind
0	1	low fat	85	1 min	rest
1	1	low fat	85	15 min	rest
2	1	low fat	88	30 min	rest
3	2	low fat	90	1 min	rest
4	2	low fat	92	15 min	rest
5	2	low fat	93	30 min	rest
6	3	low fat	97	1 min	rest
7	3	low fat	97	15 min	rest
8	3	low fat	94	30 min	rest
9	4	low fat	80	1 min	rest

```
#exercise.head()
```

```
# Show groups with different colors using "hue"
plt.figure(figsize=(8,6))
sns.barplot(x=exercise.kind , y=exercise.pulse ,hue=exercise.diet)
plt.show()
```

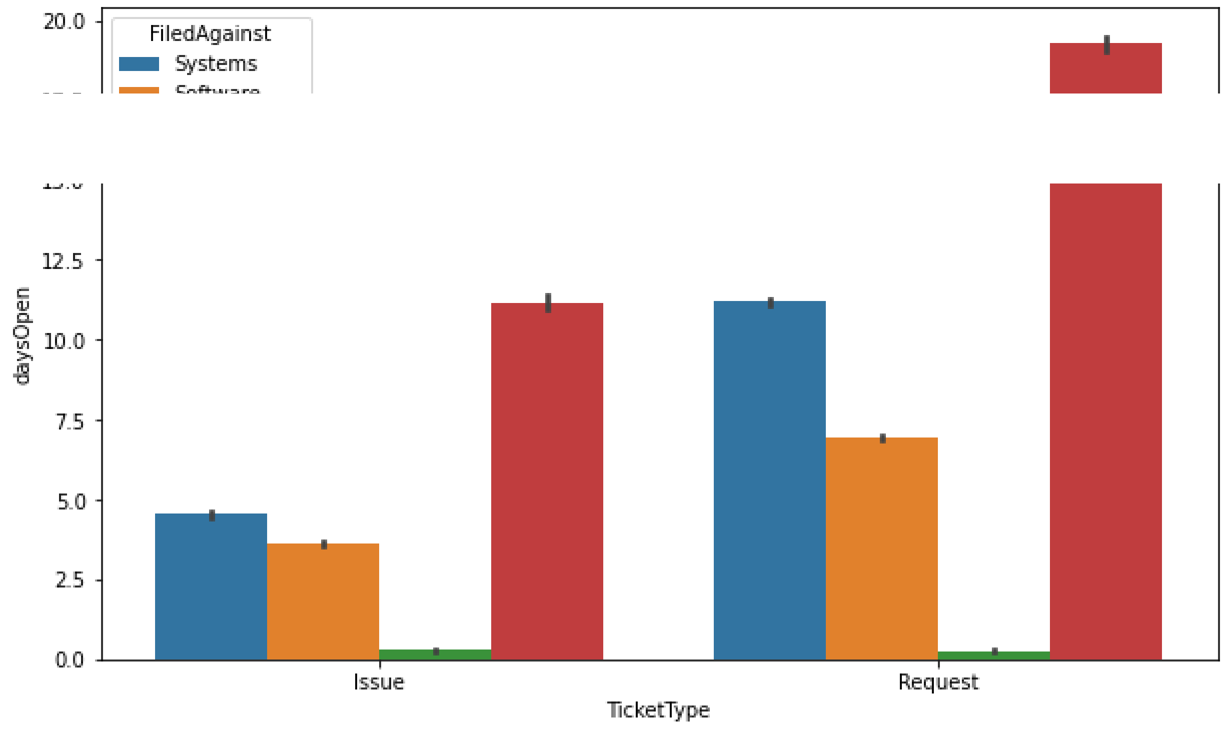


```
helpdesk = pd.read_csv("/content/drive/My Drive/Python DataScience/Visualization/Seabor
helpdesk.head(10)
```

	ticket	requestor	RequestorSeniority	ITOwner	FiledAgainst	TicketType	Severity	Priority	daysOpen	Satisfacti
0	1	1929	1 - Junior	50	Systems	Issue	2 - Normal	0 - Unassigned	3	1 - Unsatisfi
1	2	1587	2 - Regular	15	Software	Request	1 - Minor	1 - Low	5	1 - Unsatisfi
2	3	925	2 - Regular	15	Access/Login	Request	2 - Normal	0 - Unassigned	0	0 - Unknov
3	4	413	4 - Management	22	Systems	Request	2 -	0 -	20	0 - Unknov
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Normal	[illegible]	[illegible]	[illegible]

```
#helpdesk.head()
```

```
# Show groups with different colors using "hue"
plt.figure(figsize=(10,6))
sns.barplot(x=helpdesk.TicketType , y=helpdesk.daysOpen , hue=helpdesk.FiledAgainst)
#sns.despine() # right and top lines can be removed
plt.show()
```


FiledAgainst
Systems
Software
20.0
12.5
10.0
7.5
5.0
2.5
0.0
daysOpen
Issue
Request
TicketType