In [1]:

```
#Import Libraries
import numpy as np #linear algebra
import pandas as pd #data processing, csv file I/O (e.g pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [4]:

```
df_train=pd.read_csv('DigiDB_digimonlist.csv')
df_train.head()
#pd.read_csv('DigiDB_digimonlist.csv')
#pd.read_csv('DigiDB_supportlist.csv')
```

Out[4]:

	Number	Digimon	Stage	Type	Attribute	Memory	Equip Slots	Lv 50 HP	Lv50 SP	Lv50 Atk	Lv50 Def	Lv50 Int	Lv50 Spd
0	1	Kuramon	Baby	Free	Neutral	2	0	590	77	79	69	68	95
1	2	Pabumon	Baby	Free	Neutral	2	0	950	62	76	76	69	68
2	3	Punimon	Baby	Free	Neutral	2	0	870	50	97	87	50	75
3	4	Botamon	Baby	Free	Neutral	2	0	690	68	77	95	76	61
4	5	Poyomon	Baby	Free	Neutral	2	0	540	98	54	59	95	86

In [5]:

```
#WE will display the number of rows and column
df_train.shape
```

Out[5]:

```
(249, 13)
```

In [6]:

```
df_train.describe()
```

Out[6]:

	Number	Memory	Equip Slots	Lv 50 HP	Lv50 SP	Lv50 Atk	Lv50 Def	Lv50 Int	Lv50 Spd
count	249.000000	249.000000	249.000000	249.000000	249.000000	249.000000	249.000000	249.000000	249.000000
mean	125.000000	11.987952	1.574297	1210.883534	109.779116	124.518072	116.377510	112.638554	120.401606
std	72.024301	6.616501	0.854012	326.102384	32.454115	45.639372	32.132696	41.562888	32.633339
min	1.000000	2.000000	0.000000	530.000000	50.000000	52.000000	59.000000	50.000000	61.000000
25%	63.000000	6.000000	1.000000	990.000000	84.000000	89.000000	93.000000	79.000000	92.000000
50%	125.000000	12.000000	1.000000	1180.000000	104.000000	119.000000	113.000000	104.000000	119.000000
75%	187.000000	18.000000	2.000000	1480.000000	132.000000	153.000000	138.000000	138.000000	143.000000
max	249.000000	25.000000	3.000000	2080.000000	203.000000	318.000000	213.000000	233.000000	218.000000

In [7]:

```
df_train.head()
```

Out[7]:

	Number	Digimon	Stage	Type	Attribute	Memory	Equip Slots	Lv 50 HP	Lv50 SP	Lv50 Atk	Lv50 Def	Lv50 Int	Lv50 Spd
0	1	Kuramon	Baby	Free	Neutral	2	0	590	77	79	69	68	95
1	2	Pabumon	Baby	Free	Neutral	2	0	950	62	76	76	69	68
2	3	Punimon	Baby	Free	Neutral	2	0	870	50	97	87	50	75
3	4	Botamon	Baby	Free	Neutral	2	0	690	68	77	95	76	61
4	5	Poyomon	Baby	Free	Neutral	2	0	540	98	54	59	95	86

In [8]:

```
#Displaying the columns in our dataset
df_train.columns
```

Out[8]:

```
Index(['Number', 'Digimon', 'Stage', 'Type', 'Attribute', 'Memory',
       'Equip Slots', 'Lv 50 HP', 'Lv50 SP', 'Lv50 Atk', 'Lv50 Def',
       'Lv50 Int', 'Lv50 Spd'],
      dtype='object')
```

In [9]:

```
df_train.info()
#This command gives basic information about each column in dataset
```

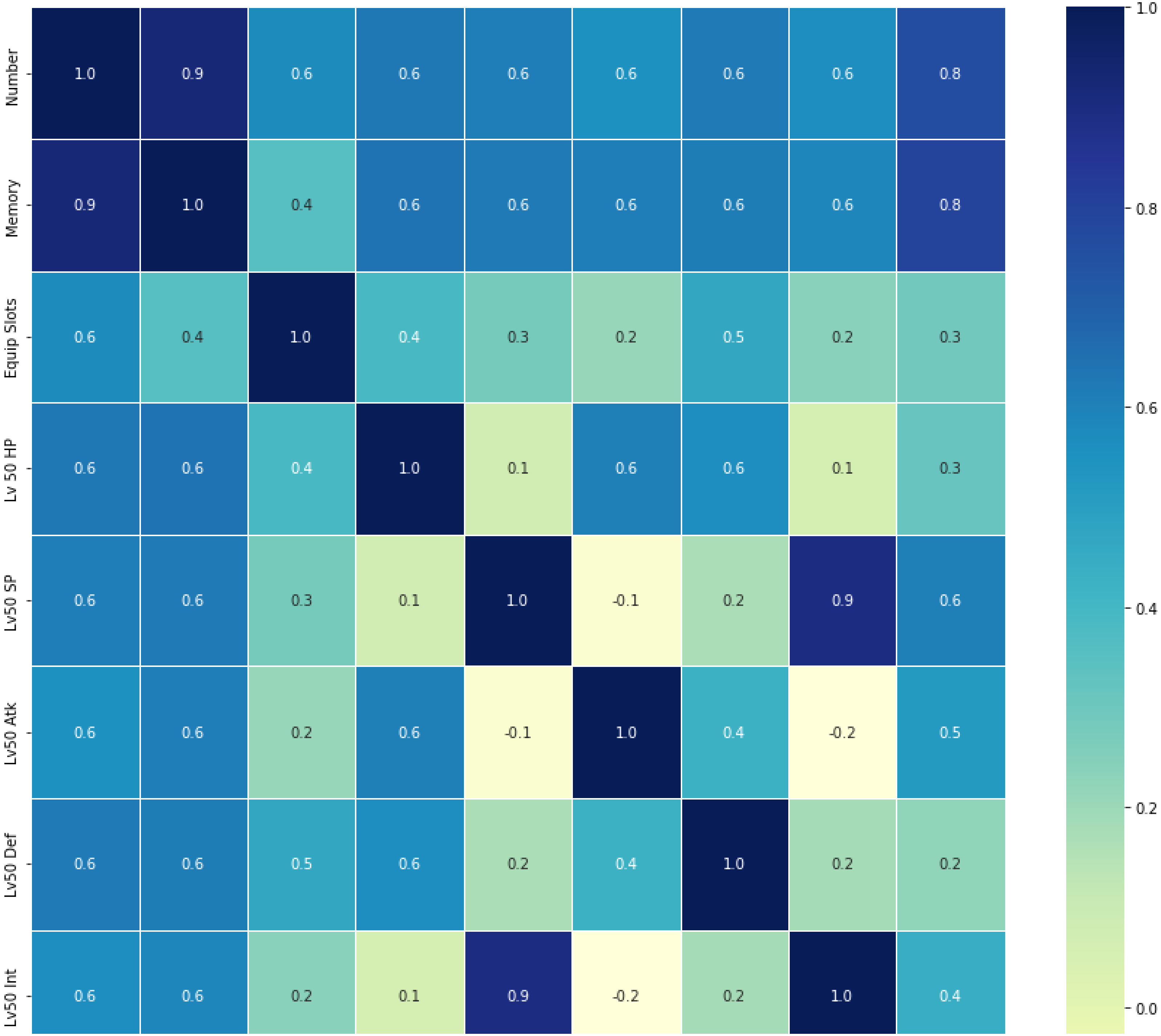
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 249 entries, 0 to 248
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Number       249 non-null    int64 
 1   Digimon      249 non-null    object
 2   Stage        249 non-null    object
 3   Type         249 non-null    object
 4   Attribute    249 non-null    object
 5   Memory       249 non-null    int64 
 6   Equip Slots  249 non-null    int64 
 7   Lv 50 HP     249 non-null    int64 
 8   Lv50 SP      249 non-null    int64 
 9   Lv50 Atk     249 non-null    int64 
 10  Lv50 Def     249 non-null    int64 
 11  Lv50 Int     249 non-null    int64 
 12  Lv50 Spd     249 non-null    int64 
dtypes: int64(9), object(4)
memory usage: 25.4+ KB
```

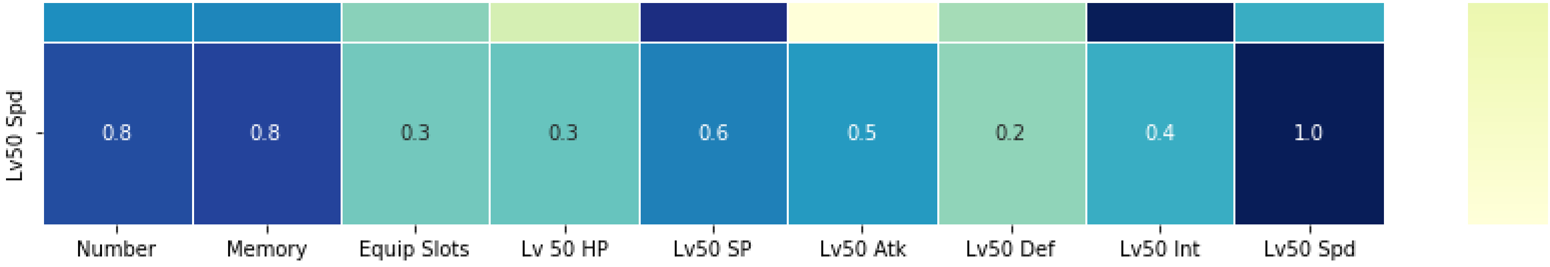
In [10]:

```
f,ax=plt.subplots(figsize=(15,15))
sns.heatmap(df_train.corr(),annot=True, linewidths=1,fmt='.1f',ax=ax,cmap='YlGnBu')
```

Out[10]:

```
<AxesSubplot:>
```


Number 1.0 0.9 0.6 0.6 0.6 0.6 0.6 0.6 0.8
Memory 0.9 1.0 0.4 0.6 0.6 0.6 0.6 0.6 0.8
Equip Slots 0.6 0.4 1.0 0.4 0.3 0.2 0.5 0.2 0.3
Lv 50 HP 0.6 0.6 0.4 1.0 0.1 0.6 0.6 0.1 0.3
Lv50 SP 0.6 0.6 0.3 0.1 1.0 -0.1 0.2 0.9 0.6
Lv50 Atk 0.6 0.6 0.2 0.6 -0.1 1.0 0.4 -0.2 0.5
Lv50 Def 0.6 0.6 0.5 0.6 0.2 0.4 1.0 0.2 0.2
Lv50 Int 0.6 0.6 0.2 0.1 0.9 -0.2 0.2 1.0 0.4
1.0
0.8
0.6
0.4
0.2
0.0



we see that,there are no null values in a row.if there were any missing values ,than we would need to choose a strategy for dealing with them:

In [11]:
```
#let's look for the maximum HP
df_train['Lv 50 HP'].max()
```

Out[11]:
```
2080
```

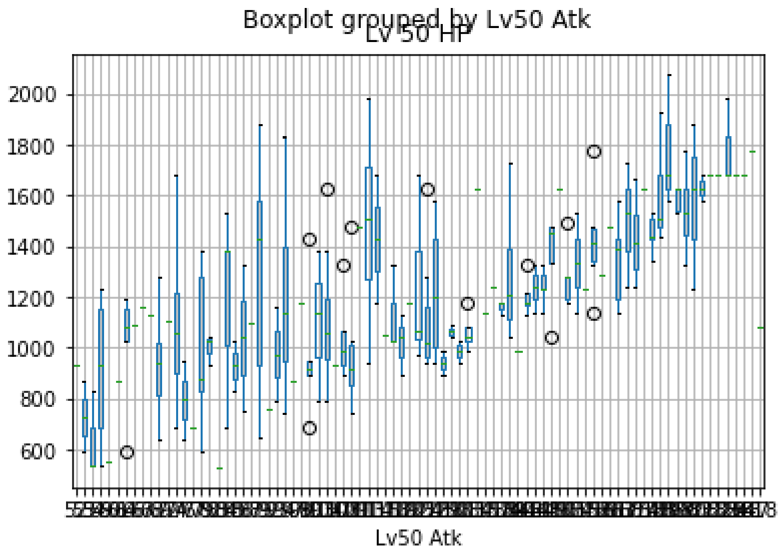
VISUAL EXPLORATORY DATA ANALYSIS

Box plots: visualize basic statistics like outliers , min/max or quantities

```
* I want to compare level of digimons and their attack
```

In [12]:
```
df_train.boxplot(column='Lv 50 HP', by='Lv50 Atk')
```

Out[12]:
```
<AxesSubplot:title={'center':'Lv 50 HP'}, xlabel='Lv50 Atk'>
```

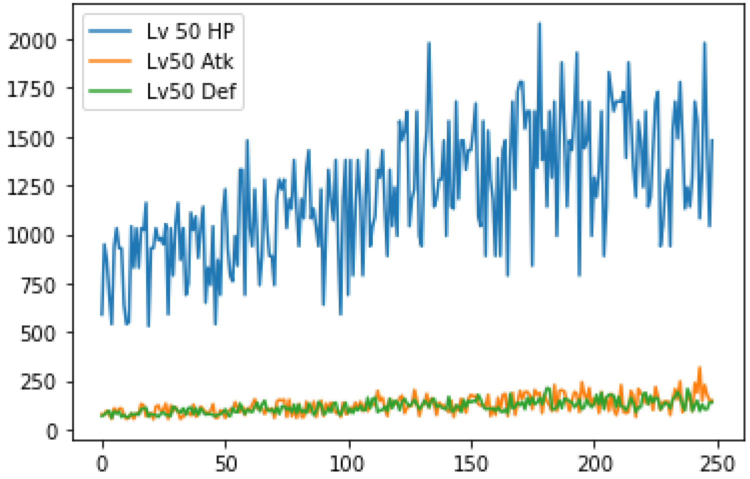


In [14]:

```
data= df_train.loc[:,['Lv 50 HP','Lv50 Atk', 'Lv50 Def']]
data.plot()
```

Out[14]:

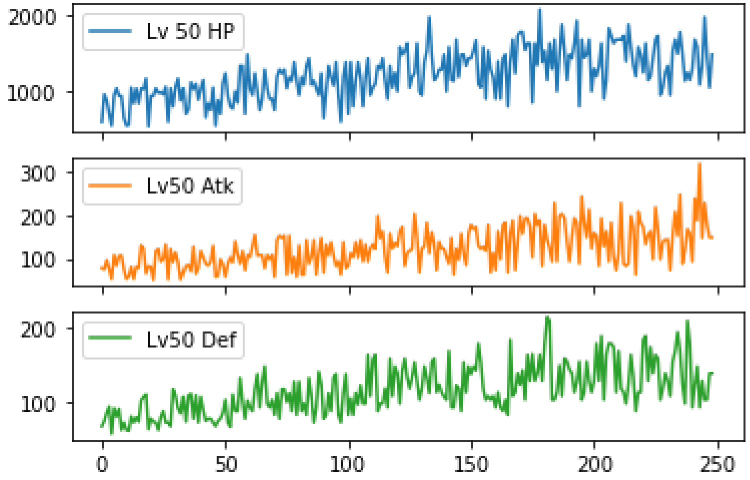
```
<AxesSubplot:>
```



In [15]:

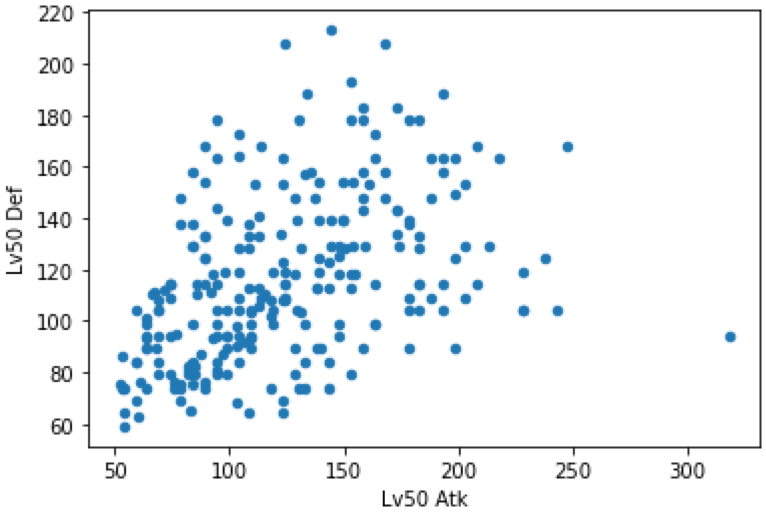
```
#subplots
data.plot(subplots=True)
```

```
Out[15]: array([<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>], dtype=object)
```



```
In [16]: #scatter plot
         data.plot(kind ='scatter',x='Lv50 Atk', y='Lv50 Def')
```

```
Out[16]: <AxesSubplot:xlabel='Lv50 Atk', ylabel='Lv50 Def'>
```

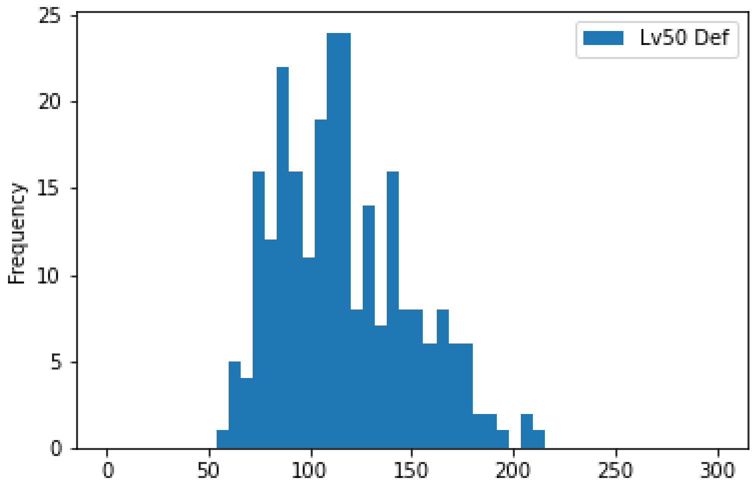


```
In [24]: #hist plot

         data.plot.hist(stacked=True, bins=50 , range=(0,300), y='Lv50 Def')
```

Out[24]:

```
<AxesSubplot:ylabel='Frequency'>
```



In [25]:

```
df_train['Type'].unique()
```

Out[25]:

```
array(['Free', 'Vaccine', 'Virus', 'Data'], dtype=object)
```

In [26]:

```
#Now i want to know , what type of digimons have the biggest attack ?
pd.crosstab(df_train['Type'], df_train['Lv50 Atk'])
```

Out[26]:

Lv50 Atk	52	53	54	59	60	61	64	66	67	68	...	198	203	208	213	218	228	238	243	247	318
Type																					
Data	0	0	0	1	0	0	3	0	0	0	...	2	0	0	1	0	1	0	0	0	0
Free	0	0	3	0	1	0	0	0	1	0	...	2	0	0	0	0	0	0	1	0	0
Vaccine	0	1	0	2	0	0	2	0	0	0	...	0	1	1	0	0	1	0	0	0	1
Virus	1	1	0	1	0	1	2	1	0	1	...	0	2	1	0	1	1	1	0	1	0

4 rows × 83 columns

In [27]:

```
df_train[df_train['Type']=='Data']['Lv50 Atk'].max()
```

Out[27]:

```
228
```

```
In [28]: df_train[df_train['Type'] == 'Free']['Lv50 Atk'].max()
```

```
Out[28]: 243
```

```
In [29]: df_train[df_train['Type'] == 'Vaccine']['Lv50 Atk'].max()
```

```
Out[29]: 318
```

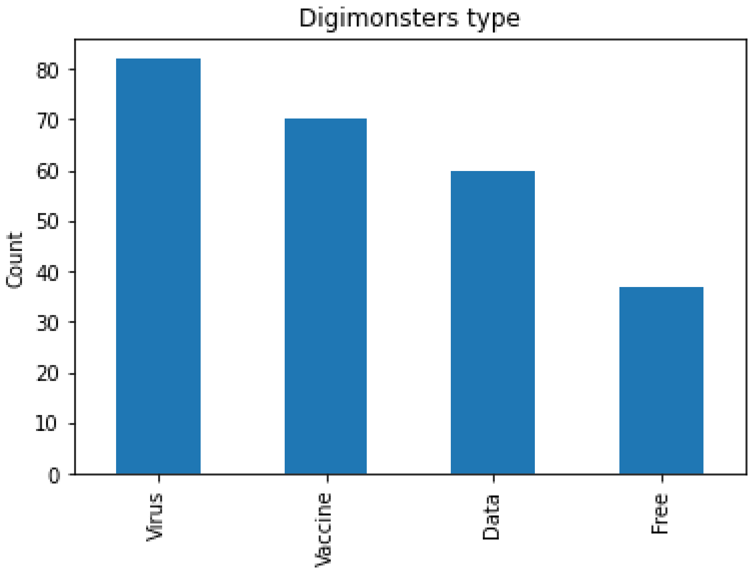
```
In [30]: df_train[df_train['Type'] == 'Virus']['Lv50 Atk'].max()
```

```
Out[30]: 247
```

The next step, i will plot the type of digimons and their amount.

```
In [31]: digimons_movelist = pd.read_csv('DigiDB_digimonlist.csv')
```

```
In [32]: digimons_movelist['Type'].value_counts().plot(kind='bar')
         plt.title('Digimonsters type')
         plt.ylabel('Count')
         plt.show()
```



In [ ]:

```

```