In [1]:

```
import pandas as pd
import numpy as np
```

What are vectorized operations

vectorized operations are a way to perform operations on entire arrays of data at once, which is faster than doing them one at a time.

In [2]:

```
a = np.array([1,2,3,4])
a * 4
```

Out[2]:

```
array([ 4,  8, 12, 16])
```

In [3]:

```
# problem in vectorized opertions in vanilla python
s = ['cat','mat',None,'rat']
[i.startswith('c') for i in s]
## Throws an error , because Startswith only works on strings
```

In [4]:

```
# How pandas solves this issue?

s = pd.Series(['cat','mat',None,'rat'])
s
```

Out[4]:

```
0     cat
1     mat
2    None
3     rat
dtype: object
```

here , **str = string accesor**

In [5]:

```
s.str.startswith('c') # Fast and optimized for larger datasets.
```

Out[5]:

```
0     True
1    False
2     None
3    False
dtype: object
```

In [6]:

```
# import titanic dataset
df =pd.read_csv("titanic.csv")
```

In [7]:
```
df.head(1)
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	

In [8]:
```
df['Name']
```

Out[8]:
```
0                                Braund, Mr. Owen Harris
1      Cumings, Mrs. John Bradley (Florence Briggs Th...
2                                 Heikkinen, Miss. Laina
3           Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                               Allen, Mr. William Henry
                             ...
886                                Montvila, Rev. Juozas
887                         Graham, Miss. Margaret Edith
888             Johnston, Miss. Catherine Helen "Carrie"
889                                Behr, Mr. Karl Howell
890                                  Dooley, Mr. Patrick
Name: Name, Length: 891, dtype: object
```

Common Functions

lower/upper/capitalize/title

In [9]:
```
# Upper
df['Name'].str.upper() # converts into Capital Words
```

Out[9]:
```
0                                BRAUND, MR. OWEN HARRIS
1      CUMINGS, MRS. JOHN BRADLEY (FLORENCE BRIGGS TH...
2                                 HEIKKINEN, MISS. LAINA
3           FUTRELLE, MRS. JACQUES HEATH (LILY MAY PEEL)
4                               ALLEN, MR. WILLIAM HENRY
                             ...
886                                MONTVILA, REV. JUOZAS
887                         GRAHAM, MISS. MARGARET EDITH
888             JOHNSTON, MISS. CATHERINE HELEN "CARRIE"
889                                BEHR, MR. KARL HOWELL
890                                  DOOLEY, MR. PATRICK
Name: Name, Length: 891, dtype: object
```

In [10]:
```
# lower
df['Name'].str.lower() # converts into small Words
```

Out[10]:
```
0                                 braund, mr. owen harris
1       cumings, mrs. john bradley (florence briggs th...
2                                  heikkinen, miss. laina
3            futrelle, mrs. jacques heath (lily may peel)
4                                allen, mr. william henry
                             ...                        
886                                 montvila, rev. juozas
887                          graham, miss. margaret edith
888              johnston, miss. catherine helen "carrie"
889                                 behr, mr. karl howell
890                                   dooley, mr. patrick
Name: Name, Length: 891, dtype: object
```

In [11]:
```
# title
df['Name'].str.title() # converts into starting Letter of Word to Capital
```

Out[11]:
```
0                                 Braund, Mr. Owen Harris
1       Cumings, Mrs. John Bradley (Florence Briggs Th...
2                                  Heikkinen, Miss. Laina
3            Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                                Allen, Mr. William Henry
                             ...                        
886                                 Montvila, Rev. Juozas
887                          Graham, Miss. Margaret Edith
888              Johnston, Miss. Catherine Helen "Carrie"
889                                 Behr, Mr. Karl Howell
890                                   Dooley, Mr. Patrick
Name: Name, Length: 891, dtype: object
```

In [12]:
```
# lets try to find the longest name In the passengers
df['Name'].str.len().max()
```

Out[12]:
```
82
```

In [13]:
```
df['Name'][df['Name'].str.len()== 82]
```

Out[13]:
```
307    Penasco y Castellana, Mrs. Victor de Satode (M...
Name: Name, dtype: object
```

In [14]:
```
df['Name'][df['Name'].str.len()== 82].values[0]
```

Out[14]:
```
'Penasco y Castellana, Mrs. Victor de Satode (Maria Josefa Perez de Soto y Va
llejo)'
```

strip

In [16]:

```
'           jack  '.strip()
```

Out[16]:

```
'jack'
```

In [17]:

```
df['Name'].str.strip() # removes spaces
```

Out[17]:

```
0                                Braund, Mr. Owen Harris
1      Cumings, Mrs. John Bradley (Florence Briggs Th...
2                                 Heikkinen, Miss. Laina
3           Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                               Allen, Mr. William Henry
                             ...
886                                Montvila, Rev. Juozas
887                         Graham, Miss. Margaret Edith
888             Johnston, Miss. Catherine Helen "Carrie"
889                                Behr, Mr. Karl Howell
890                                  Dooley, Mr. Patrick
Name: Name, Length: 891, dtype: object
```

split

In [19]:

```
# split
df['Name'].str.split(',')
```

Out[19]:

```
0                             [Braund,  Mr. Owen Harris]
1      [Cumings,  Mrs. John Bradley (Florence Briggs ...
2                              [Heikkinen,  Miss. Laina]
3        [Futrelle,  Mrs. Jacques Heath (Lily May Peel)]
4                            [Allen,  Mr. William Henry]
                             ...
886                             [Montvila,  Rev. Juozas]
887                      [Graham,  Miss. Margaret Edith]
888          [Johnston,  Miss. Catherine Helen "Carrie"]
889                             [Behr,  Mr. Karl Howell]
890                               [Dooley,  Mr. Patrick]
Name: Name, Length: 891, dtype: object
```

In [21]:

```
# Split -> get
df['Name'].str.split(',').str.get(0)
```

Out[21]:

```
0         Braund
1        Cumings
2      Heikkinen
3       Futrelle
4          Allen
         ...
886     Montvila
887       Graham
888     Johnston
889         Behr
890       Dooley
Name: Name, Length: 891, dtype: object
```

In [22]:
```
df['Name'].str.split(',').str.get(1)
```

Out[22]:
```
0                                   Mr. Owen Harris
1        Mrs. John Bradley (Florence Briggs Thayer)
2                                       Miss. Laina
3                Mrs. Jacques Heath (Lily May Peel)
4                                 Mr. William Henry
                           ...
886                                     Rev. Juozas
887                            Miss. Margaret Edith
888                  Miss. Catherine Helen "Carrie"
889                                 Mr. Karl Howell
890                                     Mr. Patrick
Name: Name, Length: 891, dtype: object
```

In [23]:
```
df['last_name'] = df['Name'].str.split(',').str.get(0)
```

In [25]:
```
df.head(1)
```

Out[25]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	

In [29]:
```
# it is used to split the Name column of the DataFrame df into two columns
# FirstName and LastName.

df['Name'].str.split(',').str.get(1).str.strip().str.split(' ',n=1, expand=Tru
```

Out[29]:

	0	1
0	Mr.	Owen Harris
1	Mrs.	John Bradley (Florence Briggs Thayer)
2	Miss.	Laina
3	Mrs.	Jacques Heath (Lily May Peel)
4	Mr.	William Henry
...	...	...
886	Rev.	Juozas
887	Miss.	Margaret Edith
888	Miss.	Catherine Helen "Carrie"
889	Mr.	Karl Howell
890	Mr.	Patrick

891 rows × 2 columns

In [30]:
```
df[['title','firstname']]= df['Name'].str.split(',').str.get(1).str.strip().st
```

In [31]:
```
df.head(1)
```

Out[31]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	

In [33]:
```
# Number of titles
df['title'].value_counts()
```

Out[33]:
```
Mr.          517
Miss.        182
Mrs.         125
Master.       40
Dr.            7
Rev.           6
Mlle.          2
Major.         2
Col.           2
the            1
Capt.          1
Ms.            1
Sir.           1
Lady.          1
Mme.           1
Don.           1
Jonkheer.      1
Name: title, dtype: int64
```

replace

In [36]:
```
df['title'] = df['title'].str.replace('Ms.','Miss.')
df['title'] = df['title'].str.replace('Mlle.','Miss.')
```

```
C:\Users\user\AppData\Local\Temp/ipykernel_15952/1805277261.py:1: FutureWarni
ng: The default value of regex will change from True to False in a future ver
sion.
  df['title'] = df['title'].str.replace('Ms.','Miss.')
C:\Users\user\AppData\Local\Temp/ipykernel_15952/1805277261.py:2: FutureWarni
ng: The default value of regex will change from True to False in a future ver
sion.
  df['title'] = df['title'].str.replace('Mlle.','Miss.')
```

In [37]:

```
df['title'].value_counts()
```

Out[37]:

```
Mr.          517
Miss.        185
Mrs.         125
Master.       40
Dr.            7
Rev.           6
Major.         2
Col.           2
Don.           1
Mme.           1
Lady.          1
Sir.           1
Capt.          1
the            1
Jonkheer.      1
Name: title, dtype: int64
```

Filtering

In [38]:
```
# startswith/endswith
df[df['firstname'].str.startswith('A')]
```

Out[38]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2750
22	23	1	3	McGowan, Miss. Anna "Annie"	female	15.0	0	0	330923	8.0292
35	36	0	1	Holverson, Mr. Alexander Oskar	male	42.0	1	0	113789	52.0000
38	39	0	3	Vander Planke, Miss. Augusta Maria	female	18.0	2	0	345764	18.0000
61	62	1	1	Icard, Miss. Amelie	female	38.0	0	0	113572	80.0000
...	...	...	...	...	...	...	...	...	...	...
842	843	1	1	Serepeca, Miss. Augusta	female	30.0	0	0	113798	31.0000
845	846	0	3	Abbing, Mr. Anthony	male	42.0	0	0	C.A. 5547	7.5500
866	867	1	2	Duran y More, Miss. Asuncion	female	27.0	1	0	SC/PARIS 2149	13.8583
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15.0	0	0	2667	7.2250
876	877	0	3	Gustafsson, Mr. Alfred Ossian	male	20.0	0	0	7534	9.8458

95 rows × 15 columns

In [40]:

```
# endswith
df[df['firstname'].str.endswith('z')]
```

Out[40]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
69	70	0	3	Kink, Mr. Vincenz	male	26.0	2	0	315151	8.6625	Na
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B5 B5 B5
721	722	0	3	Jensen, Mr. Svend Lauritz	male	17.0	1	0	350048	7.0542	Na

In [41]:

```
# isdigit/isalpha...
df[df['firstname'].str.isdigit()]
```

Out[41]:

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke

regex

In [42]:

```
# applying regex
# contains
# search john -> both case
df[df['firstname'].str.contains('john',case=False)]
```

Out[42]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	7
41	42	0	2	Turpin, Mrs. William John Robert (Dorothy Ann ...	female	27.0	1	0	11668	2
45	46	0	3	Rogers, Mr. William John	male	NaN	0	0	S.C./A.4. 23567	
98	99	1	2	Doling, Mrs. John T (Ada Julia Bone)	female	34.0	0	1	231919	2

In [44]:

```
# find lastnames with start and end char vowel ( aeiou)
df[df['last_name'].str.contains('^[^aeiouAEIOU].+[^aeiouAEIOU]$')]
```

Out[44]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
...	...	...	...	...	...	...	...	...	...	...
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500

671 rows × 15 columns

slicing

In [45]:

```
df['Name'].str[:4] # first 4 characters
```

Out[45]:

```
0      Brau
1      Cumi
2      Heik
3      Futr
4      Alle
       ... 
886    Mont
887    Grah
888    John
889    Behr
890    Dool
Name: Name, Length: 891, dtype: object
```

In [46]:

```
df['Name'].str[::2] # alternate characters
```

Out[46]:

```
0                     Ban,M.Oe ars
1      Cmns r.Jh rde Foec rgsTae)
2                      Hiknn is an
3          Ftel,Ms aqe et Ll a el
4                     Aln r ila er
                  ...              
886                    Mnvl,Rv uzs
887                 Gaa,Ms.Mrae dt
888           Jhso,Ms.CteieHln"are
889                    Bh,M.Kr oel
890                     Doe,M.Ptik
Name: Name, Length: 891, dtype: object
```

In [47]:

```
df['Name'].str[::-1] # reverse
```

Out[47]:

```
0                                sirraH newO .rM ,dnuarB
1      )reyahT sggirB ecnerolF( yeldarB nhoJ .srM ,sg...
2                                 aniaL .ssiM ,nenikkieH
3           )leeP yaM yliL( htaeH seuqcaJ .srM ,ellertuF
4                               yrneH mailliW .rM ,nellA
                             ...                        
886                                sazouJ .veR ,alivtnoM
887                         htidE teragraM .ssiM ,maharG
888             "eirraC" neleH enirehtaC .ssiM ,notsnhoJ
889                                llewoH lraK .rM ,rheB
890                                  kcirtaP .rM ,yelooD
Name: Name, Length: 891, dtype: object
```

In [ ]:

```

```