Working with Textual Data

```
import pandas as pd
import numpy as np

# What are vectorized operations  => aap jiss chiz pe operation kar rahe ho wo vector(set of things) hai

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

```
array([1, 2, 3, 4])
```

```
a*4   ## sare a ke element 4 se multiply ho jayega , ek single operation chalane se vector ke sare element me changes aaya hai isi ko bolte hain
                                    #vectorized operation
```

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

```
# problem in vectorized opertions in vanilla python
s=['cat','mat','fat','rat']     # chack karo ki kon kon se item 'c' se start ho raha hai

[i.startswith('c') for i in s]
```

```
[True, False, False, False]
```

```
# let's see another example
s=['cat','mat',None,'rat']
# chack karo ki kon kon se item 'c' se start ho raha hai
[i.startswith('c') for i in s]    # error aayega qki nonetype object startswith attributes nahi hota hai isi problem ko dur karta hai pandas
                                # and one another thing ki python me string operation slow hota hai
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[23], line 4
      2 s=['cat','mat',None,'rat']
      3 # chack karo ki kon kon se item 'c' se start ho raha hai
----> 4 [i.startswith('c') for i in s]

AttributeError: 'NoneType' object has no attribute 'startswith'
```

```
# let's see how it works in pandas
s=pd.Series(['cat','mat',None,'rat'])

# string accessor(str) isse lagana parta hai jab bhi string ke sath
```

```
kaam karte hain
s.str.startswith('c')


# it is also fast and optimized
```

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

```
# import titanic
df=pd.read_csv('titanic.csv')
df.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                           Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
# actually hum string pe kaam kar rahe hain to sirf textual data like
name ko extract kar lete hain
df['Name']
```

```
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 Function

- `lower/upper/capitalize/title`
- `lower` =>upper case me convert kar dega

```
# if sare name ko lower case me dikhana chahate ho
df['Name'].str.lower()
```

```
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
```

- `upper` => upper case me convert kar dega

```
df['Name'].str.upper()
```

```
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
```

```
df['Name'].str.capitalize()
```

```
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
```

```
df['Name'].str.title()
```

```
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
```

len

```
# len--> sare strings ka length count karke de dega

df['Name'].str.len()
```

```
0      23
1      51
2      22
3      44
4      24
       ..
886    21
887    28
888    40
889    21
890    19
Name: Name, Length: 891, dtype: int64
```

```
# sabse lamba name ko extract karo
df['Name'].str.len().max()  # ye sabse jyada langth ke string ko
return kar dega
```

```
82
```

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

```
0      False
1      False
2      False
3      False
4      False
       ...
886    False
887    False
888    False
889    False
890    False
Name: Name, Length: 891, dtype: bool
```

```
df['Name'][df['Name'].str.len() == 82]    # masking
```

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

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

```
'Penasco y Castellana, Mrs. Victor de Satode (Maria Josefa Perez de
Soto y Vallejo)'
```

```
## strip
"                           jay
".strip()
```

```
'jay'
```

```
df['Name'].str.strip()
```

```
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

```
# split -> get
df["Name"]
```

```
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
```

```
# hum chahate hain ki sare name ka salutation , first name and last name ko alag alag column me store kare
df["Name"].str.split(',')    # ek list milega and list ka first item as a surname hai
```

```
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
```

```
df["Name"].str.split(',').str.get(0)   # abb sare ka surname mil gya
```

```
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
```

```
df['lastname']=df["Name"].str.split(',').str.get(0)
df.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
```

```
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                  Name     Sex   Age
SibSp  \
0                             Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                              Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                            Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Cabin Embarked   lastname
0      0         A/5 21171   7.2500   NaN        S     Braund
1      0          PC 17599  71.2833   C85        C    Cumings
2      0  STON/O2. 3101282   7.9250   NaN        S  Heikkinen
3      0            113803  53.1000  C123        S   Futrelle
4      0            373450   8.0500   NaN        S      Allen
```

```
df["Name"]
```

```
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
```

```
df["Name"].str.split(',').str.get(1)    # abhi yaha pe salutation and
tehn name mil raha hai so we can do again split
```

```
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
```

```
df["Name"].str.split(',').str.get(1).str.split(' ')  # abhi problem ye ho raha hai ki original data me ek extra space  bhi aa raha tha salutation ke
                                                                                # pahle so we can strip
```

```
0                                      [, Mr., Owen, Harris]
1      [, Mrs., John, Bradley, (Florence, Briggs, Tha...
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
```

```
df["Name"].str.split(',').str.get(1).str.strip().str.split(' ')  # phir ek proble ki alag alag bande ke name me multiple word hai
```

```
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
```

```
# so what we can do is hum sirf 1st space ke basis pe hi split karenge
df["Name"].str.split(',').str.get(1).str.strip().str.split(' ',n=1)  # n=1 basically ek space ko split karega
```

```
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
```

```
df["Name"].str.split(',').str.get(1).str.strip().str.split(' ',n=1  ,
expand=True)  # expand Series ko dataframe me convert kar deta hai
```

```
         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 x 2 columns]
```

```
df[['title','firstname']]=df["Name"].str.split(',').str.get(1).str.str
ip().str.split(' ',n=1  , expand=True)
```

```
df.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                 Name     Sex   Age
SibSp  \
0                             Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                              Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                            Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Cabin Embarked   lastname
```

```
title  \
0      0         A/5 21171   7.2500   NaN        S     Braund    Mr.

1      0          PC 17599  71.2833   C85        C    Cumings   Mrs.

2      0  STON/O2. 3101282   7.9250   NaN        S  Heikkinen  Miss.

3      0            113803  53.1000  C123        S   Futrelle   Mrs.

4      0            373450   8.0500   NaN        S      Allen    Mr.


                                 firstname
0                              Owen Harris
1  John Bradley (Florence Briggs Thayer)
2                                   Laina
3          Jacques Heath (Lily May Peel)
4                           William Henry
```

```
# suppose hume pata karna hai ki kitne alag alag title ke log hain
df['title'].value_counts()
```

```
title
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: count, dtype: int64
```

```
# Miss and Ms both are same and Mlle in french is used for Ms so
ambeguity nn aaye we can replace

# replace
df['title']=df['title'].str.replace('Ms.','Miss.')
df['title']=df['title'].str.replace('Mlle.','Miss.')

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

```
title
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: count, dtype: int64
```

filtering

```
# suppose aapko sare passengers ka name find karna hai that starts
wiith a

# Startswith()
df['firstname'].str.startswith('A')
```

```
0      False
1      False
2      False
3      False
4      False
       ...
886    False
887    False
888    False
889    False
890    False
Name: firstname, Length: 891, dtype: bool
```

```
df[df['firstname'].str.startswith('A')]
```

```
     PassengerId  Survived  Pclass                                Name
\
13            14         0       3          Andersson, Mr. Anders Johan

22            23         1       3          McGowan, Miss. Anna "Annie"

35            36         0       1       Holverson, Mr. Alexander Oskar
```

```
38            39         0       3  Vander Planke, Miss. Augusta Maria
61            62         1       1                Icard, Miss. Amelie
..           ...       ...     ...                               ...
842          843         1       1            Serepeca, Miss. Augusta
845          846         0       3                Abbing, Mr. Anthony
866          867         1       2       Duran y More, Miss. Asuncion
875          876         1       3   Najib, Miss. Adele Kiamie "Jane"
876          877         0       3      Gustafsson, Mr. Alfred Ossian

        Sex   Age  SibSp  Parch         Ticket     Fare Cabin Embarked
\
13     male  39.0      1      5         347082  31.2750   NaN        S
22   female  15.0      0      0         330923   8.0292   NaN        Q
35     male  42.0      1      0         113789  52.0000   NaN        S
38   female  18.0      2      0         345764  18.0000   NaN        S
61   female  38.0      0      0         113572  80.0000   B28      NaN
..      ...   ...    ...    ...            ...      ...   ...      ...
842  female  30.0      0      0         113798  31.0000   NaN        C
845    male  42.0      0      0      C.A. 5547   7.5500   NaN        S
866  female  27.0      1      0  SC/PARIS 2149  13.8583   NaN        C
875  female  15.0      0      0           2667   7.2250   NaN        C
876    male  20.0      0      0           7534   9.8458   NaN        S

          lastname  title            firstname
13       Andersson    Mr.         Anders Johan
22         McGowan  Miss.         Anna "Annie"
35       Holverson    Mr.      Alexander Oskar
38   Vander Planke  Miss.        Augusta Maria
61           Icard  Miss.               Amelie
..             ...    ...                  ...
842       Serepeca  Miss.             Augusta
845         Abbing    Mr.             Anthony
```

```
866   Duran y More  Miss.             Asuncion
875          Najib  Miss.  Adele Kiamie "Jane"
876     Gustafsson    Mr.        Alfred Ossian

[95 rows x 15 columns]
```

```
## Startswith()

df[df['firstname'].str.endswith('A')]
```

```
     PassengerId  Survived  Pclass                    Name     Sex  Age
SibSp  \
64            65         0       1  Stewart, Mr. Albert A    male  NaN
0
303          304         1       2    Keane, Miss. Nora A  female  NaN
0

     Parch    Ticket     Fare Cabin Embarked lastname  title firstname

64       0  PC 17605  27.7208   NaN        C  Stewart    Mr.  Albert A

303      0    226593  12.3500  E101        Q    Keane  Miss.    Nora A
```

```
## isdigit/isalpha...


# check ki kisi ka name digit se milke bana hai
df[df['firstname'].str.isdigit()]
```

```
Empty DataFrame
Columns: [PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch,
Ticket, Fare, Cabin, Embarked, lastname, title, firstname]
Index: []
```

applying regex (regular expression)

```
# wo sare name find karo jisme ki john available hai either small me
or capital me

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

```
0      False
1       True
2      False
3      False
```

```
4      False
       ...
886    False
887    False
888    False
889    False
890    False
Name: firstname, Length: 891, dtype: bool
```

```
df[df['firstname'].str.contains('john',case=False)]
```

```
     PassengerId  Survived  Pclass  \
1              2         1       1
41            42         0       2
45            46         0       3
98            99         1       2
112          113         0       3
117          118         0       2
160          161         0       3
162          163         0       3
165          166         1       3
168          169         0       1
188          189         0       3
212          213         0       3
226          227         1       2
227          228         0       3
324          325         0       3
328          329         1       3
401          402         0       3
418          419         0       2
467          468         0       1
527          528         0       1
548          549         0       3
549          550         1       2
550          551         1       1
563          564         0       3
572          573         1       1
574          575         0       3
581          582         1       1
583          584         0       1
586          587         0       2
594          595         0       2
613          614         0       3
624          625         0       3
657          658         0       3
694          695         0       1
698          699         0       1
700          701         1       1
733          734         0       2
760          761         0       3
```

```
765          766          1       1
818          819          0       3
822          823          0       1
825          826          0       3
848          849          0       2
864          865          0       2

                                                     Name     Sex   Age
SibSp  \
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
41   Turpin, Mrs. William John Robert (Dorothy Ann ...  female  27.0
1
45                            Rogers, Mr. William John    male   NaN
0
98                Doling, Mrs. John T (Ada Julia Bone)  female  34.0
0
112                             Barton, Mr. David John    male  22.0
0
117                    Turpin, Mr. William John Robert    male  29.0
1
160                           Cribb, Mr. John Hatfield    male  44.0
0
162                          Bengtsson, Mr. John Viktor    male  26.0
0
165    Goldsmith, Master. Frank John William "Frankie"    male   9.0
0
168                                 Baumann, Mr. John D    male   NaN
0
188                                    Bourke, Mr. John    male  40.0
1
212                             Perkin, Mr. John Henry    male  22.0
0
226                           Mellors, Mr. William John    male  19.0
0
227                    Lovell, Mr. John Hall ("Henry")    male  20.5
0
324                             Sage, Mr. George John Jr    male   NaN
8
328     Goldsmith, Mrs. Frank John (Emily Alice Brown)  female  31.0
1
401                                     Adams, Mr. John    male  26.0
0
418                          Matthews, Mr. William John    male  30.0
0
467                          Smart, Mr. John Montgomery    male  56.0
0
527                                  Farthing, Mr. John    male   NaN
0
```

```
548                            Goldsmith, Mr. Frank John    male  33.0
1
549                         Davies, Master. John Morgan Jr    male   8.0
1
550                           Thayer, Mr. John Borland Jr    male  17.0
0
563                                     Simmons, Mr. John    male   NaN
0
572                      Flynn, Mr. John Irwin ("Irving")    male  36.0
0
574                          Rush, Mr. Alfred George John    male  16.0
0
581  Thayer, Mrs. John Borland (Marian Longstreth M...  female  39.0
1
583                                    Ross, Mr. John Hugo    male  36.0
0
586                               Jarvis, Mr. John Denzil    male  47.0
0
594                               Chapman, Mr. John Henry    male  37.0
1
613                                    Horgan, Mr. John    male   NaN
0
624                           Bowen, Mr. David John "Dai"    male  21.0
0
657                         Bourke, Mrs. John (Catherine)  female  32.0
1
694                                     Weir, Col. John    male  60.0
0
698                               Thayer, Mr. John Borland    male  49.0
1
700  Astor, Mrs. John Jacob (Madeleine Talmadge Force)  female  18.0
1
733                            Berriman, Mr. William John    male  23.0
0
760                                   Garfirth, Mr. John    male   NaN
0
765                Hogeboom, Mrs. John C (Anna Andrews)  female  51.0
1
818                     Holm, Mr. John Fredrik Alexander    male  43.0
0
822                      Reuchlin, Jonkheer. John George    male  38.0
0
825                                    Flynn, Mr. John    male   NaN
0
848                                    Harper, Rev. John    male  28.0
0
864                               Gill, Mr. John William    male  24.0
0
```

	Parch	Ticket	Fare	Cabin	Embarked	lastname	title
1	0	PC 17599	71.2833	C85	C	Cumings	Mrs.
41	0	11668	21.0000	NaN	S	Turpin	Mrs.
45	0	S.C./A.4. 23567	8.0500	NaN	S	Rogers	Mr.
98	1	231919	23.0000	NaN	S	Doling	Mrs.
112	0	324669	8.0500	NaN	S	Barton	Mr.
117	0	11668	21.0000	NaN	S	Turpin	Mr.
160	1	371362	16.1000	NaN	S	Cribb	Mr.
162	0	347068	7.7750	NaN	S	Bengtsson	Mr.
165	2	363291	20.5250	NaN	S	Goldsmith	Master.
168	0	PC 17318	25.9250	NaN	S	Baumann	Mr.
188	1	364849	15.5000	NaN	Q	Bourke	Mr.
212	0	A/5 21174	7.2500	NaN	S	Perkin	Mr.
226	0	SW/PP 751	10.5000	NaN	S	Mellors	Mr.
227	0	A/5 21173	7.2500	NaN	S	Lovell	Mr.
324	2	CA. 2343	69.5500	NaN	S	Sage	Mr.
328	1	363291	20.5250	NaN	S	Goldsmith	Mrs.
401	0	341826	8.0500	NaN	S	Adams	Mr.
418	0	28228	13.0000	NaN	S	Matthews	Mr.
467	0	113792	26.5500	NaN	S	Smart	Mr.
527	0	PC 17483	221.7792	C95	S	Farthing	Mr.
548	1	363291	20.5250	NaN	S	Goldsmith	Mr.
549	1	C.A. 33112	36.7500	NaN	S	Davies	Master.
550	2	17421	110.8833	C70	C	Thayer	Mr.
563	0	SOTON/OQ 392082	8.0500	NaN	S	Simmons	

```
Mr.
572      0          PC 17474   26.3875      E25        S      Flynn
Mr.
574      0       A/4. 20589    8.0500      NaN        S       Rush
Mr.
581      1            17421  110.8833      C68        C     Thayer
Mrs.
583      0            13049   40.1250      A10        C       Ross
Mr.
586      0           237565   15.0000      NaN        S     Jarvis
Mr.
594      0      SC/AH 29037   26.0000      NaN        S    Chapman
Mr.
613      0           370377    7.7500      NaN        Q     Horgan
Mr.
624      0            54636   16.1000      NaN        S      Bowen
Mr.
657      1           364849   15.5000      NaN        Q     Bourke
Mrs.
694      0           113800   26.5500      NaN        S       Weir
Col.
698      1            17421  110.8833      C68        C     Thayer
Mr.
700      0          PC 17757  227.5250  C62 C64        C      Astor
Mrs.
733      0            28425   13.0000      NaN        S   Berriman
Mr.
760      0           358585   14.5000      NaN        S   Garfirth
Mr.
765      0            13502   77.9583      D11        S   Hogeboom
Mrs.
818      0           C 7075    6.4500      NaN        S       Holm
Mr.
822      0            19972    0.0000      NaN        S   Reuchlin
Jonkheer.
825      0           368323    6.9500      NaN        Q      Flynn
Mr.
848      1           248727   33.0000      NaN        S     Harper
Rev.
864      0           233866   13.0000      NaN        S       Gill
Mr.

                                        firstname
1          John Bradley (Florence Briggs Thayer)
41   William John Robert (Dorothy Ann Wonnacott)
45                                  William John
98                        John T (Ada Julia Bone)
112                                 David John
117                           William John Robert
```

```
160                                  John Hatfield
162                                    John Viktor
165                  Frank John William "Frankie"
168                                          John D
188                                            John
212                                     John Henry
226                                   William John
227                            John Hall ("Henry")
324                                 George John Jr
328                Frank John (Emily Alice Brown)
401                                            John
418                                   William John
467                                 John Montgomery
527                                            John
548                                      Frank John
549                                   John Morgan Jr
550                                  John Borland Jr
563                                            John
572                           John Irwin ("Irving")
574                              Alfred George John
581      John Borland (Marian Longstreth Morris)
583                                       John Hugo
586                                    John Denzil
594                                     John Henry
613                                            John
624                                David John "Dai"
657                                John (Catherine)
694                                            John
698                                   John Borland
700        John Jacob (Madeleine Talmadge Force)
733                                   William John
760                                            John
765                           John C (Anna Andrews)
818                          John Fredrik Alexander
822                                     John George
825                                            John
848                                            John
864                                   John William
```

```
# find lastnames(surname) with start and end char vowel

df.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
```

```
                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                           Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Cabin Embarked   lastname
title  \
0      0         A/5 21171   7.2500   NaN        S     Braund    Mr.

1      0          PC 17599  71.2833   C85        C    Cumings   Mrs.

2      0  STON/O2. 3101282   7.9250   NaN        S  Heikkinen  Miss.

3      0            113803  53.1000  C123        S   Futrelle   Mrs.

4      0            373450   8.0500   NaN        S      Allen    Mr.


                                firstname
0                             Owen Harris
1  John Bradley (Florence Briggs Thayer)
2                                  Laina
3          Jacques Heath (Lily May Peel)
4                           William Henry
```

```
df[df['lastname'].str.contains('^[aeiouAEIOU].+[aeiouAEIOU]$')]
# andar bala ^ -> start ko batata hai and $ --> last of string ko
batata hai dot batata hai ki aage ke sare and + batata hai ki sare
string
                    #jo bich me hai usse add karke rakhana hai
```

```
     PassengerId  Survived  Pclass  \
30            31         0       1
49            50         0       3
207          208         1       3
210          211         0       3
353          354         0       3
493          494         0       1
518          519         1       2
784          785         0       3
840          841         0       3
```

```
                                                    Name     Sex   Age
SibSp  \
30                             Uruchurtu, Don. Manuel E    male  40.0
0
49       Arnold-Franchi, Mrs. Josef (Josefine Franchi)  female  18.0
1
207                          Albimona, Mr. Nassef Cassem    male  26.0
0
210                                     Ali, Mr. Ahmed    male  24.0
0
353                           Arnold-Franchi, Mr. Josef    male  25.0
1
493                             Artagaveytia, Mr. Ramon    male  71.0
0
518  Angle, Mrs. William A (Florence "Mary" Agnes H...  female  36.0
1
784                                   Ali, Mr. William    male  25.0
0
840                          Alhomaki, Mr. Ilmari Rudolf    male  20.0
0

     Parch               Ticket     Fare Cabin Embarked         lastname
title  \
30       0            PC 17601  27.7208   NaN        C       Uruchurtu
Don.
49       0              349237  17.8000   NaN        S  Arnold-Franchi
Mrs.
207      0                2699  18.7875   NaN        C         Albimona
Mr.
210      0  SOTON/O.Q. 3101311   7.0500   NaN        S              Ali
Mr.
353      0              349237  17.8000   NaN        S  Arnold-Franchi
Mr.
493      0            PC 17609  49.5042   NaN        C    Artagaveytia
Mr.
518      0              226875  26.0000   NaN        S           Angle
Mrs.
784      0  SOTON/O.Q. 3101312   7.0500   NaN        S              Ali
Mr.
840      0    SOTON/O2 3101287   7.9250   NaN        S         Alhomaki
Mr.

                                    firstname
30                                   Manuel E
49                   Josef (Josefine Franchi)
207                             Nassef Cassem
210                                     Ahmed
353                                     Josef
493                                     Ramon
```

```
518  William A (Florence "Mary" Agnes Hughes)
784                                  William
840                              Ilmari Rudolf

# agar consonant bala chahaiye hota
df[df['lastname'].str.contains('^[^aeiouAEIOU].+[^aeiouAEIOU]$')]   # square bracket ke andar bala ^ symbol negetion ka kaam kar raha hai ki hume
                                                    #vowel nahi consonant chahaiye
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	lastname	title
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0						
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0						
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0						
5	6	0	3	Moran, Mr. James	male	NaN	0	0						
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0						
..	...	...	...	...	...	...	...	...						
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0						
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0						
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2						
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0						
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0						

```
\
0        0         A/5 21171   7.2500   NaN        S     Braund    Mr.

1        0          PC 17599  71.2833   C85        C    Cumings   Mrs.

2        0  STON/O2. 3101282   7.9250   NaN        S  Heikkinen  Miss.

5        0            330877   8.4583   NaN        Q      Moran    Mr.

6        0             17463  51.8625   E46        S   McCarthy    Mr.

..     ...               ...      ...   ...      ...        ...    ...

884      0   SOTON/OQ 392076   7.0500   NaN        S   Sutehall    Mr.

887      0            112053  30.0000   B42        S     Graham  Miss.

888      2        W./C. 6607  23.4500   NaN        S   Johnston  Miss.

889      0            111369  30.0000  C148        C       Behr    Mr.

890      0            370376   7.7500   NaN        Q     Dooley    Mr.


                                  firstname
0                               Owen Harris
1    John Bradley (Florence Briggs Thayer)
2                                     Laina
5                                     James
6                                 Timothy J
..                                      ...
884                               Henry Jr
887                         Margaret Edith
888               Catherine Helen "Carrie"
889                            Karl Howell
890                                Patrick

[671 rows x 15 columns]
```

```
# slicing
df['Name']
```

```
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
```

```
# har name ka 4 character chahaiye
df['Name'].str[:4]
```

```
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
```

```
# step
df['Name'].str[::2]
```

```
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
```

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

```
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
```

```
Date-and-Time-in-Pandas
```

```
import numpy as np

import pandas as pd
```

Timestamp Object

Time stamps reference particular moments in time (e.g., Oct 24th, 2022 at 7:00pm)

Creating Timestamp objects

- YYYY/MM/DD (is the standard format)

```
# kisi bhi ek particular time ko aap time stamp bula sakte ho

#creating a timestamp
pd.Timestamp('2025/2/24')   # output me time bhi aayega (hh:mm:ss) since humne kuch specify nahi kya hai to midnight assume kar lya jata hai

Timestamp('2025-02-24 00:00:00')

type(pd.Timestamp('2025/2/24'))

pandas._libs.tslibs.timestamps.Timestamp

# variation
pd.Timestamp('2025-2-24')

Timestamp('2025-02-24 00:00:00')

pd.Timestamp('2025,2,24')

---------------------------------------------------------------------------
DateParseError                              Traceback (most recent call last)
Cell In[17], line 1
----> 1 pd.Timestamp('2025,2,24')

File timestamps.pyx:1865, in pandas._libs.tslibs.timestamps.Timestamp.__new__()

File conversion.pyx:364, in pandas._libs.tslibs.conversion.convert_to_tsobject()
```

```
File conversion.pyx:641, in
pandas._libs.tslibs.conversion.convert_str_to_tsobject()

File parsing.pyx:336, in
pandas._libs.tslibs.parsing.parse_datetime_string()

File parsing.pyx:666, in pandas._libs.tslibs.parsing.dateutil_parse()

DateParseError: Unknown datetime string format, unable to parse:
2025,2,24
```

```
pd.Timestamp('2025, 2, 24')   #isme comma ke baad ek space dena
compulsor hai
```

```
Timestamp('2025-02-24 00:00:00')
```

```
#only year
pd.Timestamp('2025')
```

```
Timestamp('2025-01-01 00:00:00')
```

```
# using text
pd.Timestamp('24th february 2025')
```

```
Timestamp('2025-02-24 00:00:00')
```

providing time also

```
pd.Timestamp('24th february 2025 5:05pm')
```

```
Timestamp('2025-02-24 17:05:00')
```

```
pd.Timestamp('24th february 2025 5:05PM')
```

```
Timestamp('2025-02-24 17:05:00')
```

```
# using datetime.datetime obj
import datetime as dt
```

```
dt.datetime(2025,2,24,5,7,26)    # ye python ke module se kar rahe
hain
```

```
datetime.datetime(2025, 2, 24, 5, 7, 26)
```

```
# combining both
x=pd.Timestamp(dt.datetime(2025,2,24,5,7,26))
x
```

```
Timestamp('2025-02-24 05:07:26')
```

```
# fetching attributes
print(x.year)
```

```
print(x.month)
print(x.day)
print(x.hour)
print(x.minute)
print(x.second)
```

```
2025
2
24
5
7
26
```

```
# why separate objects to handle data and time when python already has datetime functionality?
```

- syntax wise datetime is very convenient
- But the performance takes a hit while working with huge data. List vs Numpy Array
- The weaknesses of Python's datetime format inspired the NumPy team to add a set of native time series data type to NumPy.
- The datetime64 dtype encodes dates as 64-bit integers, and thus allows arrays of dates to be represented very compactly.

```
import numpy as np
date = np.array('2015-07-04', dtype=np.datetime64)
date
```

```
array('2015-07-04', dtype='datetime64[D]')
```

```
date + np.arange(12)
```

```
array(['2015-07-04', '2015-07-05', '2015-07-06', '2015-07-07',
       '2015-07-08', '2015-07-09', '2015-07-10', '2015-07-11',
       '2015-07-12', '2015-07-13', '2015-07-14', '2015-07-15'],
      dtype='datetime64[D]')
```

- Because of the uniform type in NumPy datetime64 arrays, this type of operation can be accomplished much more quickly than if we were working directly with Python's datetime objects, especially as arrays get large
- Pandas Timestamp object combines the ease-of-use of python datetime with the efficient storage and vectorized interface of numpy.datetime64
- From a group of these Timestamp objects, Pandas can construct a DatetimeIndex that can be used to index data in a Series or DataFrame

DatetimeIndex Object

A collection of pandas timestamp

```
# from string
pd.DatetimeIndex(['2025/2/24','2026/8/14','2027/3/29'])
```

```
DatetimeIndex(['2025-02-24', '2026-08-14', '2027-03-29'],
dtype='datetime64[ns]', freq=None)
```

```
type(pd.DatetimeIndex(['2025/2/24','2026/8/14','2027/3/29']))
```

```
pandas.core.indexes.datetimes.DatetimeIndex
```

```
type(pd.DatetimeIndex(['2025/2/24','2026/8/14','2027/3/29'])[0])
```

```
pandas._libs.tslibs.timestamps.Timestamp
```

```
# using python datetime object
pd.DatetimeIndex([dt.datetime(2025,2,24),dt.datetime(2026,8,14),dt.dat
etime(2027,3,29)])
```

```
DatetimeIndex(['2025-02-24', '2026-08-14', '2027-03-29'],
dtype='datetime64[ns]', freq=None)
```

```
# using pd.timestamp
dt_index=pd.DatetimeIndex([pd.Timestamp(2025,2,24),pd.Timestamp(2026,8
,14),pd.Timestamp(2027,3,29)])
dt_index
```

```
DatetimeIndex(['2025-02-24', '2026-08-14', '2027-03-29'],
dtype='datetime64[ns]', freq=None)
```

```
# using datetime as series index
pd.Series([1,2,3],index=dt_index)
```

```
2025-02-24    1
2026-08-14    2
2027-03-29    3
dtype: int64
```

date_range function

```
# generate daily dates in a given range
pd.date_range(start='2025/2/24',end='2025/3/31',freq='D')   # iss
range ke sare dates ko print kar dega and freq --> specifies frequency
jo ki as a step
                                                            # size ke
jaisa kaam karta hai
```

```
DatetimeIndex(['2025-02-24', '2025-02-25', '2025-02-26', '2025-02-27',
               '2025-02-28', '2025-03-01', '2025-03-02', '2025-03-03',
               '2025-03-04', '2025-03-05', '2025-03-06', '2025-03-07',
               '2025-03-08', '2025-03-09', '2025-03-10', '2025-03-11',
               '2025-03-12', '2025-03-13', '2025-03-14', '2025-03-15',
               '2025-03-16', '2025-03-17', '2025-03-18', '2025-03-19',
```

```
               '2025-03-20', '2025-03-21', '2025-03-22', '2025-03-23',
               '2025-03-24', '2025-03-25', '2025-03-26', '2025-03-27',
               '2025-03-28', '2025-03-29', '2025-03-30', '2025-03-
31'],
              dtype='datetime64[ns]', freq='D')

# suppose 1 din chor ke 1 din chahaiye to just frequency ko 2D kar do
pd.date_range(start='2025/2/24',end='2025/3/31',freq='2d')

DatetimeIndex(['2025-02-24', '2025-02-26', '2025-02-28', '2025-03-02',
               '2025-03-04', '2025-03-06', '2025-03-08', '2025-03-10',
               '2025-03-12', '2025-03-14', '2025-03-16', '2025-03-18',
               '2025-03-20', '2025-03-22', '2025-03-24', '2025-03-26',
               '2025-03-28', '2025-03-30'],
              dtype='datetime64[ns]', freq='2D')

pd.date_range(start='2025/2/24',end='2025/3/31',freq='3d')

DatetimeIndex(['2025-02-24', '2025-02-27', '2025-03-02', '2025-03-05',
               '2025-03-08', '2025-03-11', '2025-03-14', '2025-03-17',
               '2025-03-20', '2025-03-23', '2025-03-26', '2025-03-
29'],
              dtype='datetime64[ns]', freq='3D')

# business days --> mon to friday
pd.date_range(start='2025/2/24',end='2025/3/31',freq='B')  # just freq
ko B kar do

DatetimeIndex(['2025-02-24', '2025-02-25', '2025-02-26', '2025-02-27',
               '2025-02-28', '2025-03-03', '2025-03-04', '2025-03-05',
               '2025-03-06', '2025-03-07', '2025-03-10', '2025-03-11',
               '2025-03-12', '2025-03-13', '2025-03-14', '2025-03-17',
               '2025-03-18', '2025-03-19', '2025-03-20', '2025-03-21',
               '2025-03-24', '2025-03-25', '2025-03-26', '2025-03-27',
               '2025-03-28', '2025-03-31'],
              dtype='datetime64[ns]', freq='B')

# W -> one day per week
pd.date_range(start='2025/2/24',end='2025/3/31',freq='W')  # ye
bydefault sare sunday hai aap apne according change bhi kar sakte hain

DatetimeIndex(['2025-03-02', '2025-03-09', '2025-03-16', '2025-03-23',
               '2025-03-30'],
              dtype='datetime64[ns]', freq='W-SUN')

pd.date_range(start='2025/2/24',end='2025/3/31',freq='W-THU')

DatetimeIndex(['2025-02-27', '2025-03-06', '2025-03-13', '2025-03-20',
               '2025-03-27'],
              dtype='datetime64[ns]', freq='W-THU')
```

```
# h  --> har hour ka datastamp
pd.date_range(start='2025/2/24',end='2025/3/31',freq='h')
```

```
DatetimeIndex(['2025-02-24 00:00:00', '2025-02-24 01:00:00',
               '2025-02-24 02:00:00', '2025-02-24 03:00:00',
               '2025-02-24 04:00:00', '2025-02-24 05:00:00',
               '2025-02-24 06:00:00', '2025-02-24 07:00:00',
               '2025-02-24 08:00:00', '2025-02-24 09:00:00',
               ...
               '2025-03-30 15:00:00', '2025-03-30 16:00:00',
               '2025-03-30 17:00:00', '2025-03-30 18:00:00',
               '2025-03-30 19:00:00', '2025-03-30 20:00:00',
               '2025-03-30 21:00:00', '2025-03-30 22:00:00',
               '2025-03-30 23:00:00', '2025-03-31 00:00:00'],
              dtype='datetime64[ns]', length=841, freq='h')
```

```
# har 6 hour ka time stamp chahaiye to
pd.date_range(start='2025/2/24',end='2025/3/31',freq='6h')
```

```
DatetimeIndex(['2025-02-24 00:00:00', '2025-02-24 06:00:00',
               '2025-02-24 12:00:00', '2025-02-24 18:00:00',
               '2025-02-25 00:00:00', '2025-02-25 06:00:00',
               '2025-02-25 12:00:00', '2025-02-25 18:00:00',
               '2025-02-26 00:00:00', '2025-02-26 06:00:00',
               ...
               '2025-03-28 18:00:00', '2025-03-29 00:00:00',
               '2025-03-29 06:00:00', '2025-03-29 12:00:00',
               '2025-03-29 18:00:00', '2025-03-30 00:00:00',
               '2025-03-30 06:00:00', '2025-03-30 12:00:00',
               '2025-03-30 18:00:00', '2025-03-31 00:00:00'],
              dtype='datetime64[ns]', length=141, freq='6h')
```

```
# M --> Month end
pd.date_range(start='2025/2/24',end='2025/3/31',freq='M')
```

```
C:\Users\jayra\AppData\Local\Temp\ipykernel_18700\3928703045.py:2:
FutureWarning: 'M' is deprecated and will be removed in a future
version, please use 'ME' instead.
  pd.date_range(start='2025/2/24',end='2025/3/31',freq='M')
```

```
DatetimeIndex(['2025-02-28', '2025-03-31'], dtype='datetime64[ns]',
freq='ME')
```

```
pd.date_range(start='2025/2/24',end='2025/3/31',freq='ME')   # upar
bale me sirf m likhne se warning aa raha tha uska updated me hai
```

```
DatetimeIndex(['2025-02-28', '2025-03-31'], dtype='datetime64[ns]',
freq='ME')
```

```
# MS -> month start
pd.date_range(start='2025/2/24',end='2025/3/31',freq='MS')
```

```
DatetimeIndex(['2025-03-01'], dtype='datetime64[ns]', freq='MS')
```

```
# A -> year end
pd.date_range(start='2025/2/24',end='2030/3/31',freq='A')
```

```
C:\Users\jayra\AppData\Local\Temp\ipykernel_18700\189764669.py:2:
FutureWarning: 'A' is deprecated and will be removed in a future
version, please use 'YE' instead.
  pd.date_range(start='2025/2/24',end='2030/3/31',freq='A')
```

```
DatetimeIndex(['2025-12-31', '2026-12-31', '2027-12-31', '2028-12-31',
               '2029-12-31'],
              dtype='datetime64[ns]', freq='YE-DEC')
```

```
pd.date_range(start='2025/2/24',end='2030/3/31',freq='YE')
```

```
DatetimeIndex(['2025-12-31', '2026-12-31', '2027-12-31', '2028-12-31',
               '2029-12-31'],
              dtype='datetime64[ns]', freq='YE-DEC')
```

```
# YS -> Year start
pd.date_range(start='2025/2/24',end='2030/3/31',freq='YS')
```

```
DatetimeIndex(['2026-01-01', '2027-01-01', '2028-01-01', '2029-01-01',
               '2030-01-01'],
              dtype='datetime64[ns]', freq='YS-JAN')
```

```
# using periods (number of results)
```

```
pd.date_range(start='2025/2/24',periods=25,freq='D')  # suru se leke
25 din tak dikhao
```

```
DatetimeIndex(['2025-02-24', '2025-02-25', '2025-02-26', '2025-02-27',
               '2025-02-28', '2025-03-01', '2025-03-02', '2025-03-03',
               '2025-03-04', '2025-03-05', '2025-03-06', '2025-03-07',
               '2025-03-08', '2025-03-09', '2025-03-10', '2025-03-11',
               '2025-03-12', '2025-03-13', '2025-03-14', '2025-03-15',
               '2025-03-16', '2025-03-17', '2025-03-18', '2025-03-19',
               '2025-03-20'],
              dtype='datetime64[ns]', freq='D')
```

```
pd.date_range(start='2025/2/24',periods=25,freq='h')  # hourly dikha
dega
```

```
DatetimeIndex(['2025-02-24 00:00:00', '2025-02-24 01:00:00',
               '2025-02-24 02:00:00', '2025-02-24 03:00:00',
               '2025-02-24 04:00:00', '2025-02-24 05:00:00',
               '2025-02-24 06:00:00', '2025-02-24 07:00:00',
               '2025-02-24 08:00:00', '2025-02-24 09:00:00',
               '2025-02-24 10:00:00', '2025-02-24 11:00:00',
               '2025-02-24 12:00:00', '2025-02-24 13:00:00',
               '2025-02-24 14:00:00', '2025-02-24 15:00:00',
```

```
               '2025-02-24 16:00:00', '2025-02-24 17:00:00',
               '2025-02-24 18:00:00', '2025-02-24 19:00:00',
               '2025-02-24 20:00:00', '2025-02-24 21:00:00',
               '2025-02-24 22:00:00', '2025-02-24 23:00:00',
               '2025-02-25 00:00:00'],
              dtype='datetime64[ns]', freq='h')

# jo upar kye ho sab kuch periods ke sath bhi kar sakte ho
```

to_datetime function

converts an existing objects to pandas timestamp/datetimeindex object

```
# simple series example

s= pd.Series(['2023/1/1','2022/1/1','2021/1/1'])
s

0    2023/1/1
1    2022/1/1
2    2021/1/1
dtype: object

# suppose hume sare year ko nikalna hai
s.str.split('/')

0    [2023, 1, 1]
1    [2022, 1, 1]
2    [2021, 1, 1]
dtype: object

s.str.split('/').str.get(0)

0    2023
1    2022
2    2021
dtype: object

# hum normal tarike se to find kar lenge but lets see something
intresting

# simply jo series hai use convert kar do datetime me

pd.to_datetime(s)

0   2023-01-01
1   2022-01-01
2   2021-01-01
dtype: datetime64[ns]

pd.to_datetime(s).dt.year
```

```
0    2023
1    2022
2    2021
dtype: int32
```

```
pd.to_datetime(s).dt.month
```

```
0    1
1    1
2    1
dtype: int32
```

```
pd.to_datetime(s).dt.day
```

```
0    1
1    1
2    1
dtype: int32
```

```
pd.to_datetime(s).dt.day_name()
```

```
0      Sunday
1    Saturday
2      Friday
dtype: object
```

```
pd.to_datetime(s).dt.month_name()
```

```
0    January
1    January
2    January
dtype: object
```

```
# with errors
s= pd.Series(['2023/1/1','2022/1/1','2021/130/1'])
s
```

```
0      2023/1/1
1      2022/1/1
2    2021/130/1
dtype: object
```

```
pd.to_datetime(s)  # yaha pe error aa jayega qki month ka value 130
nahi ho sakta hai
```

```
---------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
Cell In[153], line 1
```

```
----> 1 pd.to_datetime(s)

File ~\anaconda3\Lib\site-packages\pandas\core\tools\
datetimes.py:1067, in to_datetime(arg, errors, dayfirst, yearfirst,
utc, format, exact, unit, infer_datetime_format, origin, cache)
   1065         result = arg.map(cache_array)
   1066     else:
-> 1067         values = convert_listlike(arg._values, format)
   1068         result = arg._constructor(values, index=arg.index,
name=arg.name)
   1069 elif isinstance(arg, (ABCDataFrame, abc.MutableMapping)):

File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:433,
in _convert_listlike_datetimes(arg, format, name, utc, unit, errors,
dayfirst, yearfirst, exact)
    431 # `format` could be inferred, or user didn't ask for mixed-
format parsing.
    432 if format is not None and format != "mixed":
--> 433     return _array_strptime_with_fallback(arg, name, utc,
format, exact, errors)
    435 result, tz_parsed = objects_to_datetime64(
    436     arg,
    437     dayfirst=dayfirst,
   (...)
    441     allow_object=True,
    442 )
    444 if tz_parsed is not None:
    445     # We can take a shortcut since the datetime64 numpy array
    446     # is in UTC

File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:467,
in _array_strptime_with_fallback(arg, name, utc, fmt, exact, errors)
    456 def _array_strptime_with_fallback(
    457     arg,
    458     name,
   (...)
    462     errors: str,
    463 ) -> Index:
    464     """
    465     Call array_strptime, with fallback behavior depending on
'errors'.
    466     """
--> 467     result, tz_out = array_strptime(arg, fmt, exact=exact,
errors=errors, utc=utc)
    468     if tz_out is not None:
    469         unit = np.datetime_data(result.dtype)[0]

File strptime.pyx:501, in
pandas._libs.tslibs.strptime.array_strptime()
```

```
File strptime.pyx:451, in
pandas._libs.tslibs.strptime.array_strptime()

File strptime.pyx:583, in
pandas._libs.tslibs.strptime._parse_with_format()

ValueError: time data "2021/130/1" doesn't match format "%Y/%m/%d", at
position 2. You might want to try:
    - passing `format` if your strings have a consistent format;
    - passing `format='ISO8601'` if your strings are all ISO8601 but
not necessarily in exactly the same format;
    - passing `format='mixed'`, and the format will be inferred for
each element individually. You might want to use `dayfirst` alongside
this.
```

```
pd.to_datetime(s,errors='coerce')    # ye pure code me error nn aa
jaye so hum errors parrameter pass karte hain jo ki agar kahi
                                     #problem aata hai to usse not a
time(NaT) bata deta hai
 #(ye ho sakte hai ki galti se kisi nn month ki value ko kuch aur likh
dya ho or typing mistake ho sakta hai to aaise problem me bachne ke
lye ye
  #   karte hain)
```

```
0   2023-01-01
1   2022-01-01
2          NaT
dtype: datetime64[ns]
```

```
# abb sare operation use kar sakte hain
pd.to_datetime(s,errors='coerce').dt.year
```

```
0    2023.0
1    2022.0
2       NaN
dtype: float64
```

```
df=pd.read_csv('expense_data.csv')
df.shape
```

```
(277, 11)
```

```
df.head()
```

```
             Date                Account        Category
Subcategory  \
```

```
0  3/2/2022 10:11  CUB - online payment             Food           NaN

1  3/2/2022 10:11  CUB - online payment            Other           NaN

2  3/1/2022 19:50  CUB - online payment             Food           NaN

3  3/1/2022 18:56  CUB - online payment  Transportation           NaN

4  3/1/2022 18:22  CUB - online payment             Food           NaN


                Note    INR Income/Expense  Note.1  Amount Currency
Account.1
0           Brownie   50.0        Expense     NaN    50.0      INR
50.0
1  To lended people  300.0        Expense     NaN   300.0      INR
300.0
2            Dinner   78.0        Expense     NaN    78.0      INR
78.0
3             Metro   30.0        Expense     NaN    30.0      INR
30.0
4            Snacks   67.0        Expense     NaN    67.0      INR
67.0
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 277 entries, 0 to 276
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            277 non-null    object
 1   Account         277 non-null    object
 2   Category        277 non-null    object
 3   Subcategory     0 non-null      float64
 4   Note            273 non-null    object
 5   INR             277 non-null    float64
 6   Income/Expense  277 non-null    object
 7   Note.1          0 non-null      float64
 8   Amount          277 non-null    float64
 9   Currency        277 non-null    object
 10  Account.1       277 non-null    float64
dtypes: float64(5), object(6)
memory usage: 23.9+ KB
```

```
# yaha pe abhi date bala column object hai so isse date time me
convert karna hoga
df["Date"]=pd.to_datetime(df['Date'])
df['Date']
```

```
0     2022-03-02 10:11:00
1     2022-03-02 10:11:00
2     2022-03-01 19:50:00
3     2022-03-01 18:56:00
4     2022-03-01 18:22:00
              ...
272   2021-11-22 14:16:00
273   2021-11-22 14:16:00
274   2021-11-21 17:07:00
275   2021-11-21 15:50:00
276   2021-11-21 13:30:00
Name: Date, Length: 277, dtype: datetime64[ns]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 277 entries, 0 to 276
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype         
---  ------          --------------  -----         
 0   Date            277 non-null    datetime64[ns]
 1   Account         277 non-null    object        
 2   Category        277 non-null    object        
 3   Subcategory     0 non-null      float64       
 4   Note            273 non-null    object        
 5   INR             277 non-null    float64       
 6   Income/Expense  277 non-null    object        
 7   Note.1          0 non-null      float64       
 8   Amount          277 non-null    float64       
 9   Currency        277 non-null    object        
 10  Account.1       277 non-null    float64       
dtypes: datetime64[ns](1), float64(5), object(5)
memory usage: 23.9+ KB
```

```
# now you can see ye date object se datetime me convert ho gya hai
```

dt accessor

Accessor object for datetimelike properties of the Series values.

```
df['Date'].dt.year
```

```
0      2022
1      2022
2      2022
3      2022
4      2022
       ...
272    2021
273    2021
```

```
274    2021
275    2021
276    2021
Name: Date, Length: 277, dtype: int32
```

```
df['Date'].dt.month
```

```
0       3
1       3
2       3
3       3
4       3
       ..
272    11
273    11
274    11
275    11
276    11
Name: Date, Length: 277, dtype: int32
```

```
df['Date'].dt.month_name()
```

```
0         March
1         March
2         March
3         March
4         March
         ...
272    November
273    November
274    November
275    November
276    November
Name: Date, Length: 277, dtype: object
```

```
df['Date'].dt.day
```

```
0       2
1       2
2       1
3       1
4       1
       ..
272    22
273    22
274    21
275    21
276    21
Name: Date, Length: 277, dtype: int32
```

```
df['Date'].dt.day_name()
```

```
0      Wednesday
1      Wednesday
2        Tuesday
3        Tuesday
4        Tuesday
         ...
272       Monday
273       Monday
274       Sunday
275       Sunday
276       Sunday
Name: Date, Length: 277, dtype: object
```

```
df['Date'].dt.is_month_end    # batayega ki ye month end hai yaa nahi
```

```
0      False
1      False
2      False
3      False
4      False
       ...
272    False
273    False
274    False
275    False
276    False
Name: Date, Length: 277, dtype: bool
```

```
df['Date'].dt.is_month_start
```

```
0      False
1      False
2       True
3       True
4       True
       ...
272    False
273    False
274    False
275    False
276    False
Name: Date, Length: 277, dtype: bool
```

```
df['Date'].dt.is_quarter_end    # quarter end hai kya
```

```
0      False
1      False
2      False
3      False
4      False
       ...
```

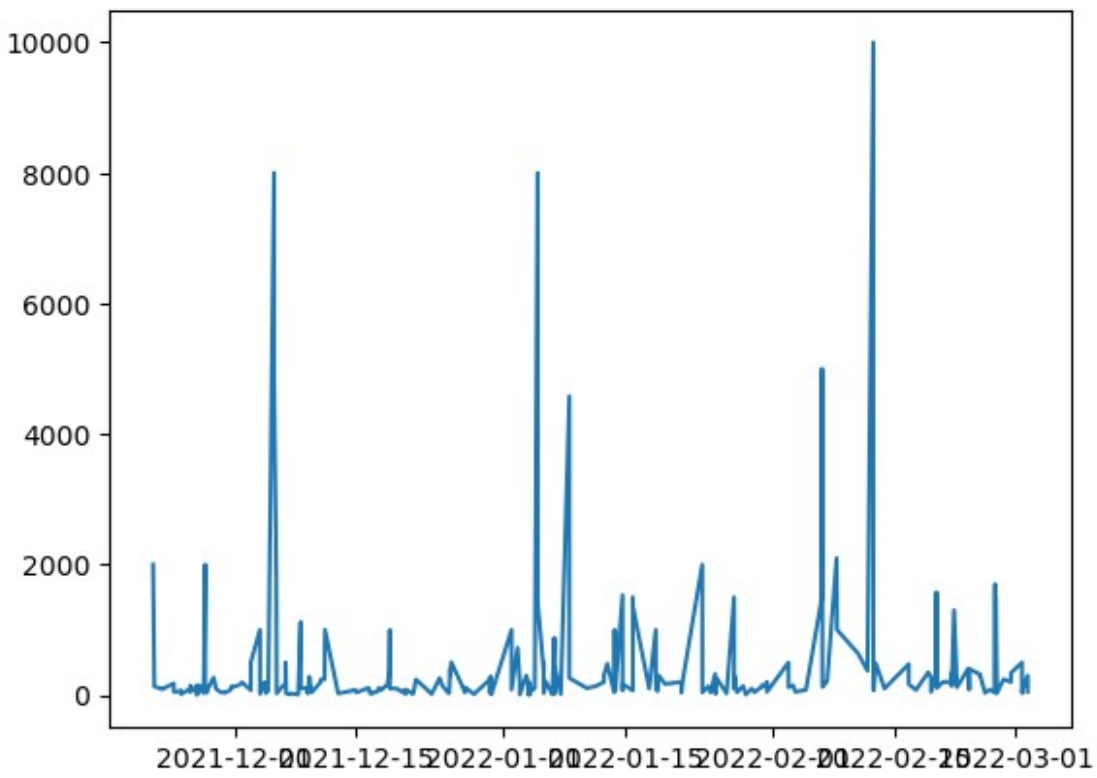
```
272    False
273    False
274    False
275    False
276    False
Name: Date, Length: 277, dtype: bool
```

```
df['Date'].dt.is_quarter_start
```

```
0      False
1      False
2      False
3      False
4      False
       ...
272    False
273    False
274    False
275    False
276    False
Name: Date, Length: 277, dtype: bool
```

```
# plot graph
import matplotlib.pyplot as plt

# suppose i have to create a graph between date and expenses (ki kon
se date pe kitna kharcha kya hai)
plt.plot(df['Date'],df['INR'])      #(x,y)->axis
```

```
[<matplotlib.lines.Line2D at 0x20d3a032c30>]
```




```
# day name wise bar chart/month wise bar chart
df['day_name']=df['Date'].dt.day_name()

df.head()
```

```
                  Date                 Account        Category
Subcategory  \
0 2022-03-02 10:11:00  CUB - online payment            Food
NaN
1 2022-03-02 10:11:00  CUB - online payment           Other
NaN
2 2022-03-01 19:50:00  CUB - online payment            Food
NaN
3 2022-03-01 18:56:00  CUB - online payment  Transportation
NaN
4 2022-03-01 18:22:00  CUB - online payment            Food
NaN

                Note    INR Income/Expense  Note.1  Amount Currency
Account.1  \
0           Brownie   50.0        Expense     NaN    50.0      INR
50.0
1  To lended people  300.0        Expense     NaN   300.0      INR
300.0
2            Dinner   78.0        Expense     NaN    78.0      INR
78.0
```

```
3             Metro   30.0        Expense     NaN    30.0       INR
30.0
4            Snacks   67.0        Expense     NaN    67.0       INR
67.0
```

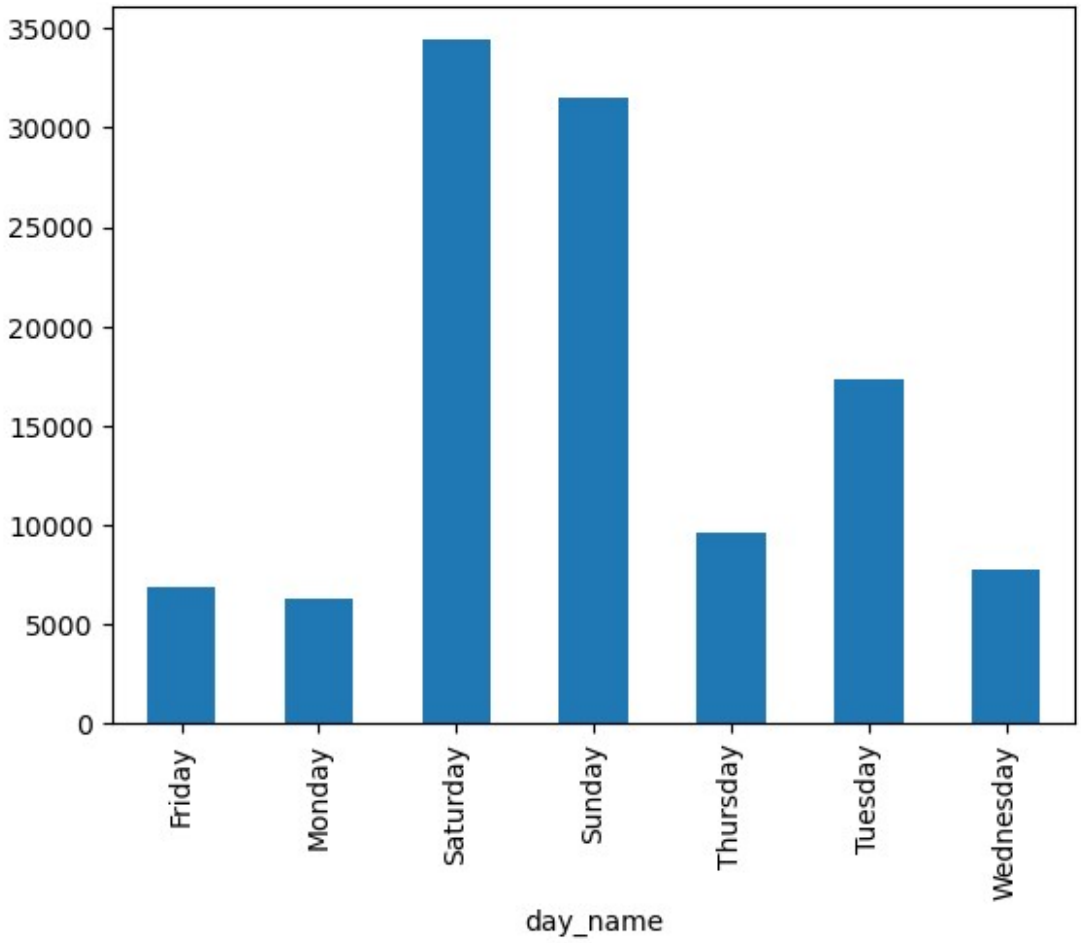
```
    day_name
0  Wednesday
1  Wednesday
2    Tuesday
3    Tuesday
4    Tuesday
```

```
df.groupby('day_name')['INR'].sum()
```

```
day_name
Friday        6910.00
Monday        6248.95
Saturday     34421.02
Sunday       31542.40
Thursday      9570.51
Tuesday      17344.65
Wednesday     7740.47
Name: INR, dtype: float64
```

```
df.groupby('day_name')['INR'].sum().plot(kind='bar')
```

```
<Axes: xlabel='day_name'>
```




```
df['month_name']=df['Date'].dt.month_name()

df.head()
```

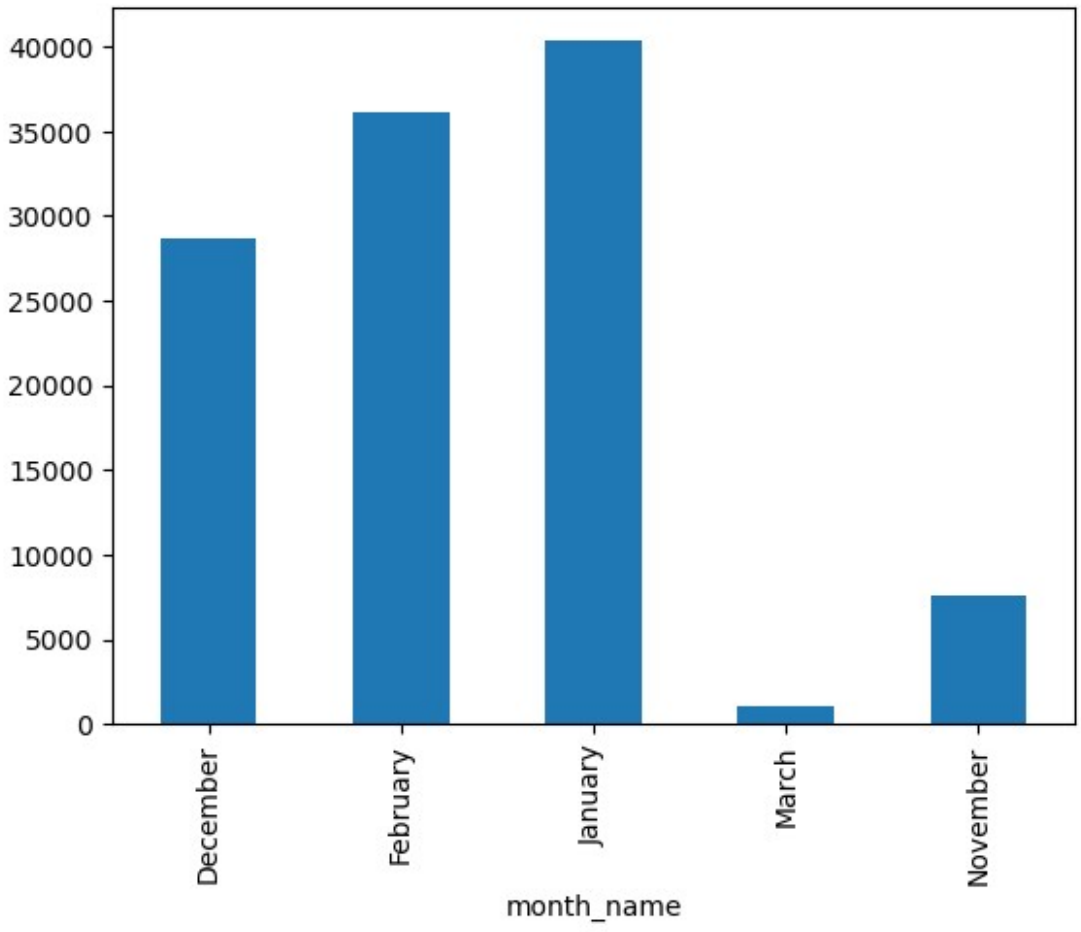
```
                  Date                 Account        Category
Subcategory  \
0 2022-03-02 10:11:00  CUB - online payment            Food
NaN
1 2022-03-02 10:11:00  CUB - online payment           Other
NaN
2 2022-03-01 19:50:00  CUB - online payment            Food
NaN
3 2022-03-01 18:56:00  CUB - online payment  Transportation
NaN
4 2022-03-01 18:22:00  CUB - online payment            Food
NaN

                Note    INR Income/Expense  Note.1  Amount Currency
Account.1  \
0           Brownie   50.0        Expense     NaN    50.0      INR
50.0
```

```
1  To lended people  300.0        Expense     NaN   300.0       INR
300.0
2            Dinner   78.0        Expense     NaN    78.0       INR
78.0
3             Metro   30.0        Expense     NaN    30.0       INR
30.0
4            Snacks   67.0        Expense     NaN    67.0       INR
67.0
```

```
    day_name month_name
0  Wednesday      March
1  Wednesday      March
2    Tuesday      March
3    Tuesday      March
4    Tuesday      March
```

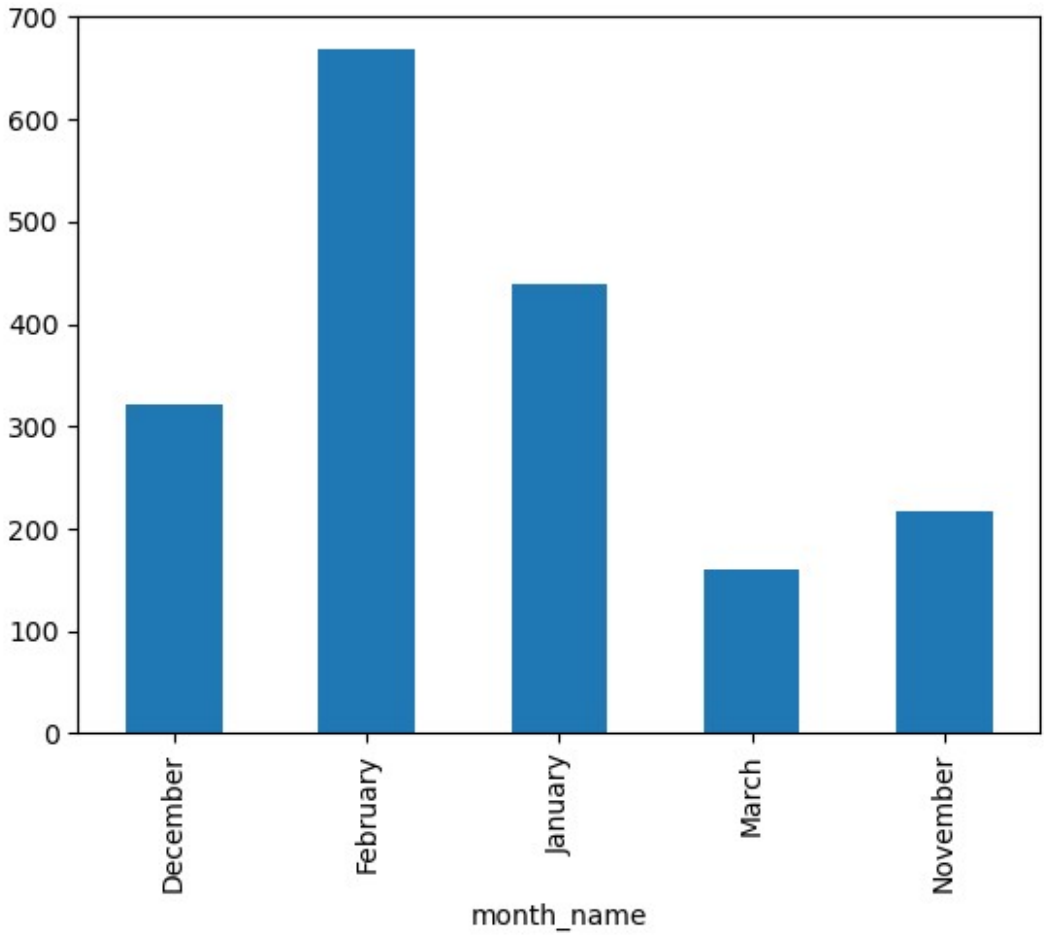
```
df.groupby('month_name')['INR'].sum().plot(kind='bar')
```

```
<Axes: xlabel='month_name'>
```




```
df.groupby('month_name')['INR'].mean().plot(kind='bar')
```

```
<Axes: xlabel='month_name'>
```




```
# batao ye month ke end me kitna kharcha karta hai
df['Date'].dt.is_month_end
```

```
0      False
1      False
2      False
3      False
4      False
       ...
272    False
273    False
274    False
275    False
276    False
Name: Date, Length: 277, dtype: bool
```

```
df[df['Date'].dt.is_month_end]
```

```
                    Date                Account        Category
Subcategory  \
7   2022-02-28 11:56:00  CUB - online payment            Food
NaN
8   2022-02-28 11:45:00  CUB - online payment           Other
NaN
61  2022-01-31 08:44:00  CUB - online payment  Transportation
NaN
62  2022-01-31 08:27:00  CUB - online payment           Other
NaN
63  2022-01-31 08:26:00  CUB - online payment  Transportation
NaN
242 2021-11-30 14:24:00  CUB - online payment            Gift
NaN
243 2021-11-30 14:17:00  CUB - online payment            Food
NaN
244 2021-11-30 10:11:00  CUB - online payment            Food
NaN

                    Note     INR Income/Expense  Note.1  Amount
Currency  \
7                  Pizza  339.15        Expense     NaN  339.15
INR
8            From kumara  200.00         Income     NaN  200.00
INR
61           Vnr to apk   50.00        Expense     NaN   50.00
INR
62             To vicky  200.00        Expense     NaN  200.00
INR
63       To ksr station  153.00        Expense     NaN  153.00
INR
242    Bharath birthday  115.00        Expense     NaN  115.00
INR
243  Lunch with company  128.00        Expense     NaN  128.00
INR
244           Breakfast   70.00        Expense     NaN   70.00
INR

     Account.1 day_name month_name
7       339.15   Monday   February
8       200.00   Monday   February
61       50.00   Monday    January
62      200.00   Monday    January
63      153.00   Monday    January
242     115.00  Tuesday   November
243     128.00  Tuesday   November
244      70.00  Tuesday   November
```