

```
In [147]: x.mode()[0]
```

```
Out[147]: 0    2  
         1    3  
         Name: 0, dtype: int32
```

```
In [148]: st.mode(x)
```

```
Out[148]: 0
```

---Structuring ---

Metacharacter []

```
In [30]: import re  
        txt = "The rain in Spain"  
        #Find all lower case characters alphabetically between "a" and "m":  
        x = re.findall("[a-zA-Z]{1,3}", txt)  
        print(x)
```

```
['The', 'rai', 'n', 'in', 'Spa', 'in']
```

```
In [31]: x = re.search("ai", txt)  
        print(x)
```

```
<re.Match object; span=(5, 7), match='ai'>
```

```
In [36]: x = re.match("The", txt)  
        print(x)
```

```
<re.Match object; span=(0, 3), match='The'>
```

```
In [11]: x = re.split("pa", txt)  
        print(x)
```

```
['The rain in S', 'in']
```

```
In [37]: x = re.sub("pa", "ta", txt)
print(x)
```

The rain in Stain

---One or more + and * ---

```
In [232]: txt = "The rain in Pakistan"
#Find all lower case characters alphabetically between "a" and "m":
x = re.findall("[a-m]+", txt)
print(x)
```

[' r', 'n ', 'n ', 'p', 'n']

```
In [233]: txt = "The rain in Pakistan"
#Find all lower case characters alphabetically between "a" and "m":
x = re.findall("[a-m]+", txt)
print(x)
```

['T', '', '', ' r', '', '', 'n ', '', 'n Sp', '', '', 'n', '']

--- metacharacter \ ---

```
In [235]: txt = "That wi3ll be 59 do78911a6rs"
#Find all digit characters:
x = re.findall("\d", txt)
print(x)
```

['3', '59', '789', '6']

```
In [241]: txt = "That wi3ll be 59 dolla6rs"
#Find all digit characters:
x = re.findall("\D+", txt)
print(x)
```

['That wi', 'll be ', ' dolla', 'rs']

```
In [38]: txt = "That wi3ll be 59 dolla6rs"
#Find all digit characters:
x = re.findall("\d+", txt)
print(x)
```

```
['3', '59', '6']
```

--- metacharacter '.' ---

```
In [246]: import re
txt = "hello planet hegapo heabo he23o 45896"
#Search for a sequence that starts with "he", followed by two (any) characters, and an "o":
x = re.findall("he..o", txt)
print(x)
```

```
['45896']
```

--- metacharacter '^' start with given character ---

```
In [14]: txt = "helllo jello planet hello world"
#Check if the string starts with 'hello':
x = re.findall("^hello", txt)
if x:
    print("Yes, the string starts with 'hello'")
else:
    print("No match")
```

```
No match
```

```
In [39]: txt = "hello planet hello world"
#Check if the string starts with 'hello':
x = re.findall("^hello world", txt)
if x:
    print("Yes, the string starts with 'hello'")
else:
    print("No match")
```

No match

--- metacharacter '^' with '[']' start with given character ---

```
In [21]: txt = "hello plane4t he3llo world"
#Check if the string starts with 'hello':
x = re.findall(r'^hello \s', txt)
print(x)
```

['p', 'a', 'n', '4', 't', '3', 'w', 'r', 'd']

```
In [47]: # --- metacharacter $ ---

txt = "hello planet"
#Check if the string ends with 'planet':
x = re.findall("planet$", txt)
if x:
    print("Yes, the string ends with 'planet'")
else:
    print("No match")
```

Yes, the string ends with 'planet'

```
In [48]: txt = "hello planet world"
#Check if the string ends with 'planet':
x = re.findall("planet$", txt)
if x:
    print("Yes, the string ends with 'planet'")
else:
    print("No match")
```

No match

```
In [22]: # --- metachracter * ---
txt = "hello planet heo helo"
#Search for a sequence that starts with "he", followed by 0 or more (any) >> characters, and an "o":
x = re.findall("he[a-z]+o", txt)
print(x)
```

['hello', 'heo']

```
In [57]: # --- metachracter + ---
txt = "hello planet"
#Search for a sequence that starts with "he", followed by 1 or more (any) characters, and an "o":
x = re.findall("he.+o", txt)
print(x)
```

['hello']

```
In [23]: # --- metachracter ? ---
txt = "hello planet heo"
#Search for a sequence that starts with "he", followed by 0 or 1 (any) character, and an "o":
x = re.findall("he.?o", txt)
print(x)
```

['heo']

```
In [24]: # --- metachracter ? ---
txt = "hello planet"
#Search for a sequence that starts with "he", followed by 0 or 1 (any) character, and an "o":
x = re.findall('[l]{1,3}', txt)
print(x)

['ll', 'l']
```

```
In [63]: # --- metachracter {} ---
txt = "hello planet"
#Search for a sequence that starts with "he", followed exactly 2 (any) characters, and an "o":
x = re.findall("he.{2}o", txt)
print(x)

['hello']
```

```
In [71]: # --- metachracter {} ---
txt = "hello planet"
#Search for a sequence that starts with "he", followed exactly 2 (any) characters, and an "o":
x = re.findall('[a-z]{2}', txt)
print(x)

['he', 'll', 'pl', 'an', 'et']
```

```
In [25]: # --- metachracter | ---
txt = "The rain in Spain falls mainly in the plain!"
#Check if the string contains either "falls" or "stays":
x = re.findall("falls|stays", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")

['falls']
Yes, there is at least one match!
```

```
In [91]: # --- special sequence \A
txt = "The rain in Spain"
#Check if the string starts with "The":
x = re.findall(r"\AThe", txt)
print(x)
if x:
    print("Yes, there is a match!")
else:
    print("No match")
```

```
['The']
Yes, there is a match!
```

```
In [90]: # --- special sequence \b
txt = "The rain in Spain"
#Check if "ain" is present at the end of a WORD:
x = re.findall(r"ain\b", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['ain', 'ain']
Yes, there is at least one match!
```

```
In [27]: # --- special sequence \b
txt = "The rain in Spain ain"
#Check if "ain" is present at the end of a WORD:
x = re.findall(r"\bain", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['ain']
Yes, there is at least one match!
```

```
In [86]: # --- special sequence \b
txt = "The rain in Spain"
#Check if "ain" is present at the end of a WORD:
x = re.findall(r"\bpai", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
[]
No match
```

```
In [87]: # --- special sequence \b begning or end of word
txt = "The rain in Spain"
#Check if "ain" is present at the end of a WORD:
x = re.findall(r"\bSpain", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['Spain']
Yes, there is at least one match!
```

```
In [93]: # --- special sequence \B
txt = "The rain in Spain"
#Check if "ain" is present, but NOT at the beginning of a word:
x = re.findall(r"\Bain", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['ain', 'ain']
Yes, there is at least one match!
```



```
In [94]: # --- special sequence \d
txt = "The rain in Spain"
#Check if the string contains any digits (numbers from 0-9):
x = re.findall("\d", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
[]
No match
```

```
In [97]: # --- special sequence \d
txt = "pakistan currency is beigned devalued compared with dollar today rate is $1 = Rs.228"
#Check if the string contains any digits (numbers from 0-9):
x = re.findall("\d", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['1', '2', '2', '8']
Yes, there is at least one match!
```

```
In [98]: # --- special sequence \d
txt = "pakistan currency is beigned devalued compared with dollar today rate is $1 = Rs.228"
#Check if the string contains any digits (numbers from 0-9):
x = re.findall("\d+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['1', '228']
Yes, there is at least one match!
```

```
In [99]: # --- special sequence \D
txt = "The rain in Spain"
#Return a match at every no-digit character:
x = re.findall("\D", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

['T', 'h', 'e', ' ', 'r', 'a', 'i', 'n', ' ', 'i', 'n', ' ', 'S', 'p', 'a', 'i', 'n']
Yes, there is at least one match!

```
In [100]: # --- special sequence \D
txt = "one 1, two 2, and one hundred two 102"
#Return a match at every no-digit character:
x = re.findall("\D", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

['o', 'n', 'e', ' ', '1', ',', ' ', 't', 'w', 'o', ' ', '2', ',', ' ', 'a', 'n', 'd', ' ', ' ', 'o', 'n', 'e', ' ', 'h', 'u', 'n', 'd', 'r', 'e', 'd', ' ', ' ', 't', 'w', 'o', ' ', '1', '0', '2']
Yes, there is at least one match!

```
In [101]: # --- special sequence \D
txt = "one 1, two 2, and one hundred two 102"
#Return a match at every no-digit character:
x = re.findall("\D+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

['one ', ', ', 'two ', ', ', 'and one hundred two ']
Yes, there is at least one match!

```
In [102]: # --- special sequence \s
txt = "The rain in Spain"
#Return a match at every white-space character:
x = re.findall("\s", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
[' ', ' ', ' ', ' ']
```

Yes, there is at least one match!

```
In [103]: # --- special sequence--- \S
txt = "The rain in Spain"
#Return a match at every NON white-space character:
x = re.findall("\S", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['T', 'h', 'e', 'r', 'a', 'i', 'n', 'i', 'n', 'S', 'p', 'a', 'i', 'n']
```

Yes, there is at least one match!

```
In [104]: # --- special sequence--- \S
txt = "The rain in Spain"
#Return a match at every NON white-space character:
x = re.findall("\S+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['The', 'rain', 'in', 'Spain']
```

Yes, there is at least one match!

```
In [105]: # --- special sequence--- \w
txt = "The rain in Spain"
#Return a match at every word character (characters from a to Z, digits from 0-9, and the underscore _ character)
x = re.findall("\w", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['T', 'h', 'e', 'r', 'a', 'i', 'n', 'i', 'n', 'S', 'p', 'a', 'i', 'n']
Yes, there is at least one match!
```

```
In [106]: # --- special sequence--- \w
txt = "The rain in Spain"
#Return a match at every word character (characters from a to Z, digits from 0-9, and the underscore _ character)
x = re.findall("\w+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['The', 'rain', 'in', 'Spain']
Yes, there is at least one match!
```

```
In [108]: # --- special sequence--- \W
txt = "The rain in Spain *** $10"
#Return a match at every NON word character (characters NOT between a and Z. Like "!", "?" white-space etc.):
x = re.findall("\W", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
[' ', ' ', ' ', ' ', ' ', '*', '*', '*', ' ', '$']
Yes, there is at least one match!
```

```
In [109]: # --- special sequence--- \W
txt = "The rain in Spain *** $10"
#Return a match at every NON word character (characters NOT between a and Z. Like "!", "?" white-space etc.):
x = re.findall("\W+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
[' ', ' ', ' ', ' ', ' *** $']
Yes, there is at least one match!
```

```
In [110]: # --- special sequence--- \Z
txt = "The rain in Spain"
#Check if the string ends with "Spain":
x = re.findall("Spain\Z", txt)
print(x)
if x:
    print("Yes, there is a match!")
else:
    print("No match")
```

```
['Spain']
Yes, there is a match!
```

--- Set operation [] ---

```
In [113]: txt = "The rain in Spain"
#Check if the string has any a, r, or n characters:
x = re.findall('[arn]', txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['r', 'a', 'n', 'n', 'a', 'n']
Yes, there is at least one match!
```

```
In [114]: txt = "The rain in Spain"
#Check if the string has any a, r, or n characters:
x = re.findall('[arn]+', txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['ra', 'n', 'n', 'a', 'n']
Yes, there is at least one match!
```

```
In [115]: txt = "The rain in Spain"
#Check if the string has any characters between a and n:
x = re.findall("[a-n]", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['h', 'e', 'a', 'i', 'n', 'i', 'n', 'a', 'i', 'n']
Yes, there is at least one match!
```

```
In [116]: txt = "The rain in Spain"
#Check if the string has any characters between a and n:
x = re.findall("[a-n]+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['he', 'ain', 'in', 'ain']
Yes, there is at least one match!
```

```
In [117]: txt = "The rain in Spain"
#Check if the string has other characters than a, r, or n:
x = re.findall("[^arn]", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['T', 'h', 'e', ' ', 'i', ' ', 'i', ' ', 'S', 'p', 'i']
Yes, there is at least one match!
```

```
In [118]: txt = "The rain in Spain"
#Check if the string has other characters than a, r, or n:
x = re.findall("[^arn]+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['The ', 'i', ' i', ' Sp', 'i']
Yes, there is at least one match!
```

```
In [120]: txt = "The rain in Spain"
#Check if the string has any 0, 1, 2, or 3 digits:
x = re.findall("[i0123S]", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['i', 'i', 'S', 'i']
Yes, there is at least one match!
```

```
In [121]: txt = "The rain in Spain"
#Check if the string has any 0, 1, 2, or 3 digits:
x = re.findall("[0123]", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
[]
No match
```

```
In [122]: txt = "8 times before 11:45 AM"
#Check if the string has any digits:
x = re.findall("[0-9]", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['8', '1', '1', '4', '5']
Yes, there is at least one match!
```



```
In [123]: txt = "8 times before 11:45 AM"
#Check if the string has any digits:
x = re.findall("[0-9]+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['8', '11', '45']
```

Yes, there is at least one match!

```
In [124]: txt = "8 times before 11:45 AM"
#Check if the string has any characters from a to z Lower case, and A to Z upper case:
x = re.findall("[a-zA-Z]", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['t', 'i', 'm', 'e', 's', 'b', 'e', 'f', 'o', 'r', 'e', 'A', 'M']
```

Yes, there is at least one match!

```
In [125]: txt = "8 times before 11:45 AM"
#Check if the string has any characters from a to z Lower case, and A to Z upper case:
x = re.findall("[a-zA-Z]+", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
['times', 'before', 'AM']
```

Yes, there is at least one match!

```
In [126]: txt = "8 times before 11:45 AM"
#Check if the string has any + characters:
x = re.findall("[+]", txt)
print(x)
if x:
    print("Yes, there is at least one match!")
else:
    print("No match")
```

```
[]
No match
```

--- Real example of dataset ---

```
In [1]: import pandas as pd
import numpy as np
```

--- Titanic Dataset ---

```
In [1]: import pandas as pd
dfT = pd.read_csv('./dataset/titanic_train.csv')
```

```
In [3]: dfT.isnull().sum()
```

```
Out[3]: PassengerId      0
Survived      0
Pclass      0
Name      0
Sex      0
Age      177
SibSp      0
Parch      0
Ticket      0
Fare      0
Cabin      687
Embarked      2
dtype: int64
```

```
In [2]: dfT.head()
```

```
Out[2]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
In [4]: dfT['Title'] = dfT['Name'].str.extract('([A-Za-z]+\.)', expand=False)
```

```
In [7]: dfT.Title.value_counts()
```

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

```
In [9]: dfT['Age'].fillna(dfT.groupby('Title')['Age'].transform('mean'), inplace=True) # fill with respect to Title group
```

```
In [10]: dfT.head(20)
```

```
Out[10]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	NaN	S	Mr.
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C	Mrs.
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	NaN	S	Miss.
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S	Mrs.
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	NaN	S	Mr.
5	6	0	3	Moran, Mr. James	male	32.368090	0	0	330877	8.4583	NaN	Q	Mr.
6	7	0	1	McCarthy, Mr. Timothy J	male	54.000000	0	0	17463	51.8625	E46	S	Mr.
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.000000	3	1	349909	21.0750	NaN	S	Master.
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.000000	0	2	347742	11.1333	NaN	S	Mrs.
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.000000	1	0	237736	30.0708	NaN	C	Mrs.
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.000000	1	1	PP 9549	16.7000	G6	S	Miss.
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.000000	0	0	113783	26.5500	C103	S	Miss.
12	13	0	3	Saunderscock, Mr. William Henry	male	20.000000	0	0	A/5. 2151	8.0500	NaN	S	Mr.
13	14	0	3	Andersson, Mr. Anders Johan	male	39.000000	1	5	347082	31.2750	NaN	S	Mr.
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.000000	0	0	350406	7.8542	NaN	S	Miss.

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title	
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.000000	0	0	248706	16.0000	NaN	S	Mrs.
16	17	0	3	Rice, Master. Eugene	male	2.000000	4	1	382652	29.1250	NaN	Q	Master.
17	18	1	2	Williams, Mr. Charles Eugene	male	32.368090	0	0	244373	13.0000	NaN	S	Mr.
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.000000	1	0	345763	18.0000	NaN	S	Mrs.
19	20	1	3	Masselmani, Mrs. Fatima	female	35.898148	0	0	2649	7.2250	NaN	C	Mrs.

```
In [54]: # Trying to get last integers of ticket
# dfT['Ticket_num'] = dfT['Ticket'].str.extract('(\d+\b)' , expand=False)
```

```
In [57]: #Findall function is not suitable for dataframe

x = re.findall('[A-Za-z]+\.',dfT['Name'][0])    #find all method, it will run only for one row
```

```
In [153]: x
```

```
Out[153]: ['Mr. ']
```

--Regex on weather data ---

```
In [13]: dfW = pd.read_csv('./dataset/weather_data.csv')
```

```
In [16]: import numpy as np
dfW.temperature.replace('-9999',np.NaN,inplace=True)
```

```
In [17]: dfW.temperature.unique()
```

```
Out[17]: array(['32', '--9999', '28', nan, '32 #', '34FA', '40'], dtype=object)
```

In [20]: dfW

Out[20]:

	day	temperature	windspeed	event
0	1/1/2017	32	6us	Rain
1	1/4/2017	NaN	9	Sunny
2	1/5/2017	28	-7777	Snow
3	1/6/2017	NaN	7	NaN
4	1/7/2017	32 #	-7777	Rain
5	1/8/2017	NaN	-7777	Sunny
6	1/9/2017	NaN	-7777	NaN
7	1/10/2017	34FA	8yyy	Cloudy
8	1/11/2017	40	12	Sunny

In [19]: dfW.temperature.unique()

Out[19]: array(['32', nan, '28', '32 #', '34FA', '40'], dtype=object)

In [18]: dfW.temperature.replace('--9999',np.NaN,inplace=True)

In [23]: dfW.windspeed.replace('-7777',np.NaN,inplace=True)

In [29]: dfW['tempNew'] = dfW.temperature.str.extract('([0-9]+)',expand=False)

In [34]: dfW.tempNew = dfW.tempNew.astype(float)

In [35]: dfW.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   day         9 non-null     object
1   temperature 5 non-null     object
2   windspeed   5 non-null     object
3   event       7 non-null     object
4   tempNew     5 non-null     float64
dtypes: float64(1), object(4)
memory usage: 488.0+ bytes
```

In [58]: *# Removing unnecessary character from temperature using replace function*

```
dfW['newTemp2'] = dfW['temperature'].replace('[^0-9-]', '', regex=True)
```

In [54]: *# Removing unnecessary character from temperature using str.extract function*

```
dfW['newTemp'] = dfW['temperature'].str.extract('([-]?[0-9]+)', expand=False)
```


In [59]: dfW

Out[59]:

	day	temperature	windspeed	event	newTemp2
0	1/1/2017	32	6us	Rain	32
1	1/4/2017	--9999	9	Sunny	--9999
2	1/5/2017	28	-7777	Snow	28
3	1/6/2017	-9999	7	NaN	-9999
4	1/7/2017	32 #	-7777	Rain	32
5	1/8/2017	-9999	-7777	Sunny	-9999
6	1/9/2017	-9999	-7777	NaN	-9999
7	1/10/2017	34FA	8yyy	Cloudy	34
8	1/11/2017	40	12	Sunny	40

In [64]: dfW['newTemp'] = dfW['newTemp'].astype('float32')

In [65]: dfW.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   day              9 non-null     object
1   temperature      9 non-null     object
2   windspeed        9 non-null     object
3   event            7 non-null     object
4   newTemp          9 non-null     float32
5   newTemp2         9 non-null     object
dtypes: float32(1), object(5)
memory usage: 524.0+ bytes
```

In [171]: dfW

Out[171]:

	day	temperature	windspeed	event	newTemp	newTemp2
0	1/1/2017	32	6us	Rain	32.0	32
1	1/4/2017	-9999	9	Sunny	-9999.0	-9999
2	1/5/2017	28	-7777	Snow	28.0	28
3	1/6/2017	-9999	7	NaN	-9999.0	-9999
4	1/7/2017	32 #	-7777	Rain	32.0	32
5	1/8/2017	-9999	-7777	Sunny	-9999.0	-9999
6	1/9/2017	-9999	-7777	NaN	-9999.0	-9999
7	1/10/2017	34FA	8yyy	Cloudy	34.0	34
8	1/11/2017	40	12	Sunny	40.0	40

--- Intellectual Capical Dataset ---

In [2]: dfP = pd.read_csv('./dataset/pakistan_intellectual_capital.csv')

In [111]: dfP.shape

Out[111]: (1142, 13)