

Pandas :

- Pandas is a library in python which is used to perform data analysis.
- Pandas will reduce complexity and make our work very easy.
- It can be applicable for any type of data(ordered or unordered)
- The output will be in the format of tabular form. That tabular form is named as DataFrame.
- This can be applied for different file formats like csv, excel, pdf, json and etc.

Main Key words in creating our data using pandas:

- **Series:**
 - This series is a 1-d array.
 - The output will be in the format of row by row(line to line).
- **Data Frame:**
 - In this the data will be represented in the format of rows and columns in a tabular arrangement

Importing the Pandas library:

In [3]:



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

In [4]:



```
### Creating data using Series:
### pd.Series(List)

a = pd.Series([1,2,3,4])
a
```

Out[4]:

```
0    1
1    2
2    3
3    4
dtype: int64
```

In [5]:



```
### Creating Series using Numpy array:  
### pd.Series(np.array(list))  
  
a = pd.Series(np.array([1,2,3,4]))  
a
```

Out[5]:

```
0    1  
1    2  
2    3  
3    4  
dtype: int32
```

In [6]:



```
### Creating Data Series using Dictionary:  
  
d = {'a' : 0, "b" : 1, "c" : 2}  
d
```

Out[6]:

```
{'a': 0, 'b': 1, 'c': 2}
```

In [9]:



```
a = pd.Series(d)  
a
```

Out[9]:

```
a    0  
b    1  
c    2  
dtype: int64
```

In [11]:



```
b = pd.Series({'a' : 0, "b" : 1, "c" : 2})  
b
```

Out[11]:

```
a    0  
b    1  
c    2  
dtype: int64
```

In [14]:



```
### changing the index values in series:

a = pd.Series([1,2,3,45],index = ['z','x','y','a'])
a
```

Out[14]:

```
z      1
x      2
y      3
a     45
dtype: int64
```

In [17]:



```
a = pd.Series([1,2,'a',45,6],index = np.arange(1,6))
a
```

Out[17]:

```
1      1
2      2
3      a
4     45
5      6
dtype: object
```

In [18]:



```
a.dtype  ### o --- object datatype --> combination of numbers and alphabets.
```

Out[18]:

```
dtype('O')
```

In [19]:



```
#### Creating data using dataframe:

a = pd.DataFrame([1,2,3,4])
a
```

Out[19]:

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

In [21]:



```
##### creating data with multiple columns:

a = pd.DataFrame([["Niharikaa",12,2020],["Lavanya",11,2019],["Gayathri",10,2018]])
a
```

Out[21]:

	0	1	2
0	Niharikaa	12	2020
1	Lavanya	11	2019
2	Gayathri	10	2018

In [22]:



```
### creating multiple columns with column names:

a = pd.DataFrame([["Niharikaa",12,2020],["Lavanya",11,2019],["Gayathri",10,2018]],columns = 
a
```

Out[22]:

	Name	Age	year
0	Niharikaa	12	2020
1	Lavanya	11	2019
2	Gayathri	10	2018

In [24]:



```
### Creating dataframe by dictionary :

d = {"Name" : ["Niharikaa","Mounica","Lavanya","Gayathri"],
      "Branch" : ["Ece","Cse","Ece","Ece"],
      "Year": [2020,2019,2018,2017]}
a = pd.DataFrame(d,index = ['a','b','c','d'])
a
```

Out[24]:

	Name	Branch	Year
a	Niharikaa	Ece	2020
b	Mounica	Cse	2019
c	Lavanya	Ece	2018
d	Gayathri	Ece	2017

In [25]:



```
### creating dataframe by using multiple dictionaries:

data = [{'a':10,'b':20,"c":30},{"a":30,"b":40,"c":20}]
a = pd.DataFrame(data)
a
```

Out[25]:

	a	b	c
0	10	20	30
1	30	40	20

In [27]:



```
### Converting data Series into dataframe:
## Nan --- not a number.

d = {"one" : pd.Series([1,2,3],index = ['a','b','c']),
     "Two" : pd.Series([4,5,6,7],index = ["a","b","c","d"])}
a = pd.DataFrame(d)
a
```

Out[27]:

	one	Two
a	1.0	4
b	2.0	5
c	3.0	6
d	NaN	7

In [31]:



```
a = [1,2,3,4]
a = pd.DataFrame(a,dtype = 'float64')
a
```

Out[31]:

	0
0	1.0
1	2.0
2	3.0
3	4.0

In [32]:



```
d = {"Name" : ["Niharikaa", "Mounica", "Lavanya", "Gayathri"],
     "Branch" : ["Ece", "Cse", "Ece", "Ece"],
     "Year": [2020, 2019, 2018, 2017]}
a = pd.DataFrame(d, index = ['a', 'b', 'c', 'd'])
a
```

Out[32]:

	Name	Branch	Year
a	Niharikaa	Ece	2020
b	Mounica	Cse	2019
c	Lavanya	Ece	2018
d	Gayathri	Ece	2017

In [34]:



```
### getting all the vaules form the data frame:
a.values ### variable_name.values
```

Out[34]:

```
array([[ 'Niharikaa', 'Ece', 2020],
       [ 'Mounica', 'Cse', 2019],
       [ 'Lavanya', 'Ece', 2018],
       [ 'Gayathri', 'Ece', 2017]], dtype=object)
```

In [36]:



```
### getting all the index vaules form the data frame:
a.index ### variable_name.index
```

Out[36]:

```
Index(['a', 'b', 'c', 'd'], dtype='object')
```

In [37]:



```
### checking the nan values in a dataframe:
```

```
a.isnull()
```

Out[37]:

	Name	Branch	Year
a	False	False	False
b	False	False	False
c	False	False	False
d	False	False	False

In [38]:



```
d = {"one" : pd.Series([1,2,3],index = ['a','b','c']),  
     "Two" : pd.Series([4,5,6,7],index = ["a","b","c","d"])}  
a = pd.DataFrame(d)  
a
```

Out[38]:

	one	Two
a	1.0	4
b	2.0	5
c	3.0	6
d	NaN	7

In [39]:



```
a.isnull()
```

Out[39]:

	one	Two
a	False	False
b	False	False
c	False	False
d	True	False

In [40]:



```
a.notnull()
```

Out[40]:

	one	Two
a	True	True
b	True	True
c	True	True
d	False	True

In [41]:



```
d = {"Name" : ["Niharikaa", "Mounica", "Lavanya", "Gayathri"],
      "Branch" : ["Ece", "Cse", "Ece", "Ece"],
      "Year": [2020, 2019, 2018, 2017]}
a = pd.DataFrame(d, index = ['a', 'b', 'c', 'd'])
a
```

Out[41]:

	Name	Branch	Year
a	Niharikaa	Ece	2020
b	Mounica	Cse	2019
c	Lavanya	Ece	2018
d	Gayathri	Ece	2017

In [43]:



```
#### accessing the elements from the dataframe :

a["Name"] ### particular column
```

Out[43]:

```
a    Niharikaa
b      Mounica
c      Lavanya
d    Gayathri
Name: Name, dtype: object
```

In [44]:



```
a["Name"][ 'c' ] ### getting particular value
```

Out[44]:

```
'Lavanya'
```


In [45]:



```
a[0:3] ### first three columns
```

Out[45]:

	Name	Branch	Year
a	Niharikaa	Ece	2020
b	Mounica	Cse	2019
c	Lavanya	Ece	2018

In [47]:



```
a[-1:] #### last column
```

Out[47]:

	Name	Branch	Year
d	Gayathri	Ece	2017

iloc and loc:

- loc --- for accessing the rows using the index datatype other than integer.
- iloc --- for accessing the rows using the index datatype integer.

In [48]:



```
a
```

Out[48]:

	Name	Branch	Year
a	Niharikaa	Ece	2020
b	Mounica	Cse	2019
c	Lavanya	Ece	2018
d	Gayathri	Ece	2017

In [49]:



```
a.loc['c']
```

Out[49]:

```
Name      Lavanya
Branch      Ece
Year      2018
Name: c, dtype: object
```

In [51]:



```
a.iloc[2]
```

Out[51]:

```
Name      Lavanya
Branch      Ece
Year      2018
Name: c, dtype: object
```

In [52]:



```
a
```

Out[52]:

	Name	Branch	Year
a	Niharikaa	Ece	2020
b	Mounica	Cse	2019
c	Lavanya	Ece	2018
d	Gayathri	Ece	2017

Data Cleaning:

In [54]:

```
### reading the dataset to our jupyter notebook:
### pd.read_csv(filename)

data = pd.read_csv("matches.csv")

data
```

Out[54]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied
0	1	2008	Bangalore	18-04-2008	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0
1	2	2008	Chandigarh	19-04-2008	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	0
2	3	2008	Delhi	19-04-2008	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	0
3	4	2008	Mumbai	20-04-2008	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	0

- excel : pd.read_excel(filename)
- table : pd.read_table(filename)
- sql database : pd.read_sql(query,connection_object)
- json : pd.read_json(json_string/filename)
- html :pd.read_html(url)

In [57]:

```
### viewing our dataset:  
  
data.head() ### first five rows
```

Out[57]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl
0	1	2008	Bangalore	18-04-2008	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2	2008	Chandigarh	19-04-2008	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	
2	3	2008	Delhi	19-04-2008	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	
3	4	2008	Mumbai	20-04-2008	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	
4	5	2008	Kolkata	20-04-2008	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat	normal	

In [58]:

```
data.tail() ### last five rows
```

Out[58]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl
691	692	2018	Pune	20-05-2018	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	field	normal	
692	693	2018	Mumbai	22-05-2018	Sunrisers Hyderabad	Chennai Super Kings	Chennai Super Kings	field	normal	
693	694	2018	Kolkata	23-05-2018	Kolkata Knight Riders	Rajasthan Royals	Rajasthan Royals	field	normal	
694	695	2018	Kolkata	25-05-2018	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
695	696	2018	Mumbai	27-05-2018	Sunrisers Hyderabad	Chennai Super Kings	Chennai Super Kings	field	normal	

In [59]:



```
data.head(20) ###
```

Out[59]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result
0	1	2008	Bangalore	18-04-2008	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal
1	2	2008	Chandigarh	19-04-2008	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal
2	3	2008	Delhi	19-04-2008	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal
3	4	2008	Mumbai	20-04-2008	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal
4	5	2008	Kolkata	20-04-2008	Deccan Chargers	Kolkata Knight Riders	Deccan Chargers	bat	normal
5	6	2008	Jaipur	21-04-2008	Kings XI Punjab	Rajasthan Royals	Kings XI Punjab	bat	normal
6	7	2008	Hyderabad	22-04-2008	Deccan Chargers	Delhi Daredevils	Deccan Chargers	bat	normal
7	8	2008	Chennai	23-04-2008	Chennai Super Kings	Mumbai Indians	Mumbai Indians	field	normal
8	9	2008	Hyderabad	24-04-2008	Deccan Chargers	Rajasthan Royals	Rajasthan Royals	field	normal
9	10	2008	Chandigarh	25-04-2008	Kings XI Punjab	Mumbai Indians	Mumbai Indians	field	normal
10	11	2008	Bangalore	26-04-2008	Royal Challengers Bangalore	Rajasthan Royals	Rajasthan Royals	field	normal
11	12	2008	Chennai	26-04-2008	Kolkata Knight Riders	Chennai Super Kings	Kolkata Knight Riders	bat	normal
12	13	2008	Mumbai	27-04-2008	Mumbai Indians	Deccan Chargers	Deccan Chargers	field	normal
13	14	2008	Chandigarh	27-04-2008	Delhi Daredevils	Kings XI Punjab	Delhi Daredevils	bat	normal

	id	season	city	date	team1	team2	toss_winner	toss_decision	result
14	15	2008	Bangalore	28-04-2008	Chennai Super Kings	Royal Challengers Bangalore	Chennai Super Kings	bat	normal
15	16	2008	Kolkata	29-04-2008	Kolkata Knight Riders	Mumbai Indians	Kolkata Knight Riders	bat	normal
16	17	2008	Delhi	30-04-2008	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal
17	18	2008	Hyderabad	01-05-2008	Deccan Chargers	Kings XI Punjab	Kings XI Punjab	field	normal
18	19	2008	Jaipur	01-05-2008	Rajasthan Royals	Kolkata Knight Riders	Rajasthan Royals	bat	normal
19	20	2008	Chennai	02-05-2008	Chennai Super Kings	Delhi Daredevils	Chennai Super Kings	bat	normal

◀
▶

In [60]:



```
data.tail(20)
```

Out[60]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result
676	677	2018	Kolkata	09-05-2018	Mumbai Indians	Kolkata Knight Riders	Kolkata Knight Riders	field	no
677	678	2018	Delhi	10-05-2018	Delhi Daredevils	Sunrisers Hyderabad	Delhi Daredevils	bat	no
678	679	2018	Jaipur	11-05-2018	Chennai Super Kings	Rajasthan Royals	Chennai Super Kings	bat	no
679	680	2018	Indore	12-05-2018	Kolkata Knight Riders	Kings XI Punjab	Kings XI Punjab	field	no
680	681	2018	Delhi	12-05-2018	Delhi Daredevils	Royal Challengers Bangalore	Royal Challengers Bangalore	field	no
681	682	2018	Pune	13-05-2018	Sunrisers Hyderabad	Chennai Super Kings	Chennai Super Kings	field	no
682	683	2018	Mumbai	13-05-2018	Mumbai Indians	Rajasthan Royals	Rajasthan Royals	field	no
683	684	2018	Indore	14-05-2018	Kings XI Punjab	Royal Challengers Bangalore	Royal Challengers Bangalore	field	no
684	685	2018	Kolkata	15-05-2018	Rajasthan Royals	Kolkata Knight Riders	Kolkata Knight Riders	field	no
685	686	2018	Mumbai	16-05-2018	Mumbai Indians	Kings XI Punjab	Kings XI Punjab	field	no
686	687	2018	Bengaluru	17-05-2018	Royal Challengers Bangalore	Sunrisers Hyderabad	Sunrisers Hyderabad	field	no
687	688	2018	Delhi	18-05-2018	Delhi Daredevils	Chennai Super Kings	Chennai Super Kings	field	no
688	689	2018	Jaipur	19-05-2018	Rajasthan Royals	Royal Challengers Bangalore	Rajasthan Royals	bat	no
689	690	2018	Hyderabad	19-05-2018	Sunrisers Hyderabad	Kolkata Knight Riders	Sunrisers Hyderabad	bat	no
690	691	2018	Delhi	20-05-2018	Delhi Daredevils	Mumbai Indians	Delhi Daredevils	bat	no

	id	season	city	date	team1	team2	toss_winner	toss_decision	result
691	692	2018	Pune	20-05-2018	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	field	no
692	693	2018	Mumbai	22-05-2018	Sunrisers Hyderabad	Chennai Super Kings	Chennai Super Kings	field	no
693	694	2018	Kolkata	23-05-2018	Kolkata Knight Riders	Rajasthan Royals	Rajasthan Royals	field	no
694	695	2018	Kolkata	25-05-2018	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders	field	no
695	696	2018	Mumbai	27-05-2018	Sunrisers Hyderabad	Chennai Super Kings	Chennai Super Kings	field	no

In [61]:

```
data.shape
```

Out[61]:

```
(696, 18)
```

In [62]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 696 entries, 0 to 695
Data columns (total 18 columns):
id                696 non-null int64
season            696 non-null int64
city              689 non-null object
date              696 non-null object
team1             696 non-null object
team2             696 non-null object
toss_winner       696 non-null object
toss_decision     696 non-null object
result           696 non-null object
dl_applied        696 non-null int64
winner            693 non-null object
win_by_runs       696 non-null int64
win_by_wickets    696 non-null int64
player_of_match   693 non-null object
venue             696 non-null object
umpire1           695 non-null object
umpire2           695 non-null object
umpire3           60 non-null object
dtypes: int64(5), object(13)
memory usage: 98.0+ KB
```


In [63]:



```
data.describe() ### statistical analysis
```

Out[63]:

	id	season	dl_applied	win_by_runs	win_by_wickets
count	696.000000	696.000000	696.000000	696.000000	696.000000
mean	348.500000	2012.965517	0.027299	13.472701	3.349138
std	201.062179	3.069266	0.163070	23.607994	3.411398
min	1.000000	2008.000000	0.000000	0.000000	0.000000
25%	174.750000	2010.000000	0.000000	0.000000	0.000000
50%	348.500000	2013.000000	0.000000	0.000000	3.000000
75%	522.250000	2016.000000	0.000000	19.000000	6.000000
max	696.000000	2018.000000	1.000000	146.000000	10.000000



```
data.apply(pd.Series.value_counts)
```

Out[68]:

[illegible]

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl
TM Dilshan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
UT Yadav	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Umar Gul	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
V Kohli	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
V Sehwag	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
VA Kulkarni	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
VK Sharma	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
VR Aaron	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Vidarbha Cricket Association Stadium, Jamtha	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Vineet Kulkarni	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Virender Kumar Sharma	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Visakhapatnam	NaN	NaN	11.0	NaN	NaN	NaN	NaN	NaN	NaN	
WP Saha	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
WPUJC Vaas	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Wankhede Stadium	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Washington Sundar	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
YC Barde	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
YK Pathan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
YS Chahal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Yeshwant Barde	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Yuvraj Singh	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Z Khan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
bat	NaN	NaN	NaN	NaN	NaN	NaN	NaN	283.0	NaN	
field	NaN	NaN	NaN	NaN	NaN	NaN	NaN	413.0	NaN	
no result	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	3.0	
normal	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	686.0	
tie	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	7.0	

1575 rows × 18 columns

In [69]:



```
data.columns ### getting all the column names
```

Out[69]:

```
Index(['id', 'season', 'city', 'date', 'team1', 'team2', 'toss_winner',  
      'toss_decision', 'result', 'dl_applied', 'winner', 'win_by_runs',  
      'win_by_wickets', 'player_of_match', 'venue', 'umpire1', 'umpire2',  
      'umpire3'],  
      dtype='object')
```

In [72]:



```
data['toss_decision'].value_counts()
```

Out[72]:

```
field    413  
bat       283  
Name: toss_decision, dtype: int64
```

In [73]:



```
data["season"].value_counts()
```

Out[73]:

```
2013    76  
2012    74  
2011    73  
2018    60  
2016    60  
2014    60  
2010    60  
2017    59  
2015    59  
2008    58  
2009    57  
Name: season, dtype: int64
```

In [79]:



```
data.columns = np.arange(1,19)
```

In [80]:

```
data
```

Out[80]:

	1	2	3	4	5	6	7	8	9	10	11	12
0	1	2008	Bangalore	18-04-2008	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0	Kolkata Knight Riders	140
1	2	2008	Chandigarh	19-04-2008	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	0	Chennai Super Kings	33
2	3	2008	Delhi	19-04-2008	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	0	Delhi Daredevils	0
3	4	2008	Mumbai	20-04-2008	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	0	Royal Challengers Bangalore	0

In [82]:

```
data[18].isnull().value_counts()
```

Out[82]:

```
True      636
False      60
Name: 18, dtype: int64
```

In [86]:

```
data = data.dropna()
```

In [87]:

```
data
```

Out[87]:

	1	2	3	4	5	6	7	8	9	10	11	12	1
636	637	2018	Mumbai	07-04-2018	Mumbai Indians	Chennai Super Kings	Chennai Super Kings	field	normal	0	Chennai Super Kings	0	
637	638	2018	Mohali	08-04-2018	Delhi Daredevils	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	
638	639	2018	Kolkata	08-04-2018	Royal Challengers Bangalore	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	
639	640	2018	Hyderabad	09-04-2018	Rajasthan Royals	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	0	Sunrisers Hyderabad	0	

In [89]:

```
data.fillna('x')
```

Out[89]:

	1	2	3	4	5	6	7	8	9	10	11	12	1
636	637	2018	Mumbai	07-04-2018	Mumbai Indians	Chennai Super Kings	Chennai Super Kings	field	normal	0	Chennai Super Kings	0	
637	638	2018	Mohali	08-04-2018	Delhi Daredevils	Kings XI Punjab	Kings XI Punjab	field	normal	0	Kings XI Punjab	0	
638	639	2018	Kolkata	08-04-2018	Royal Challengers Bangalore	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	0	Kolkata Knight Riders	0	
639	640	2018	Hyderabad	09-04-2018	Rajasthan Royals	Sunrisers Hyderabad	Sunrisers Hyderabad	field	normal	0	Sunrisers Hyderabad	0	

In [90]:

```
data = pd.read_csv("matches.csv")
```

In [91]:

```
data.fillna('x')
```

Out[91]:

	id	season	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied
0	1	2008	Bangalore	18-04-2008	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	0
1	2	2008	Chandigarh	19-04-2008	Chennai Super Kings	Kings XI Punjab	Chennai Super Kings	bat	normal	0
2	3	2008	Delhi	19-04-2008	Rajasthan Royals	Delhi Daredevils	Rajasthan Royals	bat	normal	0
3	4	2008	Mumbai	20-04-2008	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	normal	0

In [92]:

```
# dataset1_name.join(d2,on = col2,how = inner)
```

In [93]:

```
### stastical concepts:
```

```
data.mean()
```

Out[93]:

```
id          348.500000
season      2012.965517
dl_applied    0.027299
win_by_runs  13.472701
win_by_wickets 3.349138
dtype: float64
```

In [94]:



```
data.corr()
```

Out[94]:

	id	season	dl_applied	win_by_runs	win_by_wickets
id	1.000000	0.995002	0.025212	-0.026325	-0.017205
season	0.995002	1.000000	0.024882	-0.028647	-0.011903
dl_applied	0.025212	0.024882	1.000000	-0.018307	-0.011985
win_by_runs	-0.026325	-0.028647	-0.018307	1.000000	-0.561075
win_by_wickets	-0.017205	-0.011903	-0.011985	-0.561075	1.000000

In [95]:



```
data.count()
```

Out[95]:

```
id          696
season      696
city        689
date        696
team1       696
team2       696
toss_winner 696
toss_decision 696
result      696
dl_applied  696
winner      693
win_by_runs 696
win_by_wickets 696
player_of_match 693
venue       696
umpire1     695
umpire2     695
umpire3      60
dtype: int64
```


In [96]:



```
data.max()
```

Out[96]:

```
id                696
season            2018
date              31-05-2008
team1             Sunrisers Hyderabad
team2             Sunrisers Hyderabad
toss_winner       Sunrisers Hyderabad
toss_decision     field
result            tie
dl_applied        1
win_by_runs       146
win_by_wickets    10
venue             Wankhede Stadium
dtype: object
```

In [104]:



```
data["id"].argmax()
```

C:\Users\Niharikaa\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning:
The current behaviour of 'Series.argmax' is deprecated, use 'idxmax' instead.
The behavior of 'argmax' will be corrected to return the positional maximum in the future. For now, use 'series.values.argmax' or 'np.argmax(np.array(values))' to get the position of the maximum row.
"""Entry point for launching an IPython kernel.

Out[104]:

```
695
```

In [105]:



```
data["id"].argmin()
```

C:\Users\Niharikaa\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning:
The current behaviour of 'Series.argmin' is deprecated, use 'idxmin' instead.
The behavior of 'argmin' will be corrected to return the positional minimum in the future. For now, use 'series.values.argmin' or 'np.argmin(np.array(values))' to get the position of the minimum row.
"""Entry point for launching an IPython kernel.

Out[105]:

```
0
```

In [106]:

```
data.median()
```

Out[106]:

```
id          348.5
season      2013.0
dl_applied   0.0
win_by_runs  0.0
win_by_wickets 3.0
dtype: float64
```

In [111]:

```
a = data.groupby("season")
```

In [110]:

```
a.count()
```

Out[110]:

	id	city	date	team1	team2	toss_winner	toss_decision	result	dl_applied	winner	\
season											
2008	58	58	58	58	58	58	58	58	58	58	
2009	57	57	57	57	57	57	57	57	57	57	
2010	60	60	60	60	60	60	60	60	60	60	
2011	73	73	73	73	73	73	73	73	73	72	
2012	74	74	74	74	74	74	74	74	74	74	
2013	76	76	76	76	76	76	76	76	76	76	
2014	60	53	60	60	60	60	60	60	60	60	
2015	59	59	59	59	59	59	59	59	59	57	
2016	60	60	60	60	60	60	60	60	60	60	
2017	59	59	59	59	59	59	59	59	59	59	
2018	60	60	60	60	60	60	60	60	60	60	

In [112]:

```
a = data.groupby("city")
```

In [114]:



```
a.count()
```

Out[114]:

	id	season	date	team1	team2	toss_winner	toss_decision	result	dl_applied
city									
Abu Dhabi	7	7	7	7	7	7	7	7	7
Ahmedabad	12	12	12	12	12	12	12	12	12
Bangalore	66	66	66	66	66	66	66	66	66
Bengaluru	7	7	7	7	7	7	7	7	7
Bloemfontein	2	2	2	2	2	2	2	2	2
Cape Town	7	7	7	7	7	7	7	7	7
Centurion	12	12	12	12	12	12	12	12	12
Chandigarh	46	46	46	46	46	46	46	46	46
Chennai	49	49	49	49	49	49	49	49	49
Cuttack	7	7	7	7	7	7	7	7	7
Delhi	67	67	67	67	67	67	67	67	67
Dharamsala	9	9	9	9	9	9	9	9	9
Durban	15	15	15	15	15	15	15	15	15
East London	3	3	3	3	3	3	3	3	3
Hyderabad	56	56	56	56	56	56	56	56	56
Indore	9	9	9	9	9	9	9	9	9
Jaipur	40	40	40	40	40	40	40	40	40
Johannesburg	8	8	8	8	8	8	8	8	8
Kanpur	4	4	4	4	4	4	4	4	4
Kimberley	3	3	3	3	3	3	3	3	3
Kochi	5	5	5	5	5	5	5	5	5
Kolkata	70	70	70	70	70	70	70	70	70
Mohali	3	3	3	3	3	3	3	3	3
Mumbai	94	94	94	94	94	94	94	94	94
Nagpur	3	3	3	3	3	3	3	3	3
Port Elizabeth	7	7	7	7	7	7	7	7	7
Pune	38	38	38	38	38	38	38	38	38
Raipur	6	6	6	6	6	6	6	6	6
Rajkot	10	10	10	10	10	10	10	10	10
Ranchi	7	7	7	7	7	7	7	7	7
Sharjah	6	6	6	6	6	6	6	6	6
Visakhapatnam	11	11	11	11	11	11	11	11	11



In []:

