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

Reading CSV which is in our computer

In [2]:
```
df=pd.read_csv(r"C:\Users\USER\Downloads\aug_train.csv")
df.head()
```

Out[2]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experienc
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>2
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	1
2	11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	
3	33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>2

Reading CSV from URL - like directly from Github (LOADING FROM SERVER)

In [3]:
```
df1=pd.read_csv("https://media.githubusercontent.com/media/datablist/sample-csv-files/main/files/customers/cust
```

In [4]:
```
df1.head()
```

Out[4]:

	Index	Customer Id	First Name	Last Name	Company	City	Country	Phone 1	Phone 2	Emai
0	Index	Customer Id	First Name	Last Name	Company	City	Country	Phone 1	Phone 2	Emai
1	1	DD37Cf93aecA6Dc	Sheryl	Baxter	Rasmussen Group	East Leonard	Chile	229.077.5154	397.884.0519x718	zunigavanessa@smith.info
2	2	1Ef7b82A4CAAD10	Preston	Lozano	Vega-Gentry	East Jimmychester	Djibouti	5153435776	686-620-1820x944	vmata@colon.com
3	3	6F94879bDAfE5a6	Roy	Berry	Murillo-Perry	Isabelborough	Antigua and Barbuda	+1-539-402-0259	(496)978-3969x58947	beckycarr@hogan.com
4	4	5Cef8BFA16c5e3c	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonview	Dominican Republic	001-808-617-6467x12895	+1-813-324-8756	stanleyblackwell@benson.org

Reading CSV file but separated with tab (Tab Separated file)

In [5]:
```
df2=pd.read_csv(r"C:\Users\USER\Downloads\movie_titles_metadata.tsv", sep="\t")
```

In [6]:
```
df2.head(4)
```

Out[6]:

	m0	10 things i hate about you	1999	6.90	62847	['comedy' 'romance']
0	m1	1492: conquest of paradise	1992	6.2	10421.0	['adventure' 'biography' 'drama' 'history']
1	m2	15 minutes	2001	6.1	25854.0	['action' 'crime' 'drama' 'thriller']
2	m3	2001: a space odyssey	1968	8.4	163227.0	['adventure' 'mystery' 'sci-fi']
3	m4	48 hrs.	1982	6.9	22289.0	['action' 'comedy' 'crime' 'drama' 'thriller']

Notice something in tab separated values, look first row become columns because there is no column names, so pass list

In [7]:
```
df3= pd.read_csv(r"C:\Users\USER\Downloads\movie_titles_metadata.tsv", sep="\t", names=['symbol','movie','relea
```

In [8]:
```
df3.head()
```

Out[8]:

	symbol	movie	release_year	rating	profit	genere
0	m0	10 things i hate about you	1999	6.9	62847.0	['comedy' 'romance']
1	m1	1492: conquest of paradise	1992	6.2	10421.0	['adventure' 'biography' 'drama' 'history']
2	m2	15 minutes	2001	6.1	25854.0	['action' 'crime' 'drama' 'thriller']
3	m3	2001: a space odyssey	1968	8.4	163227.0	['adventure' 'mystery' 'sci-fi']
4	m4	48 hrs.	1982	6.9	22289.0	['action' 'comedy' 'crime' 'drama' 'thriller']

Now lets look - index_col Parameter

In [9]:
```
#lets see first df
df.head()
```

Out[9]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experienc
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>2
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	1
2	11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	
3	33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<
4	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>2

What we are noticing? we saw that, there is normal pandas default index too 0,1,2... and there is enrollment id as enrollee_id, which can act as primary key

In [10]:
```
df=pd.read_csv(r"C:\Users\USER\Downloads\aug_train.csv",index_col='enrollee_id')
df.head()
```

Out[10]:

enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience
8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20
29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	15
11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	5
33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<1
666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>20

Now, lets look Header Parameter

In [11]:
```
#First of all lets look the problem
df4=pd.read_csv(r"C:\Users\USER\Downloads\test.csv")
df4.head(2)
```

Out[11]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	
0	0	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	e
1	1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	

What we notice? - header is becoming our first row

In [13]:
```
df4=pd.read_csv(r"C:\Users\USER\Downloads\test.csv",header=1)
df4.head(2)
```

Out[13]:

	0	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experier
0	1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	
1	2	11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	

Now, sometime, in ML we may not need all column, we may only need few columns - we can decrease and can only import needed column if we want directly during import of data rather than dropping later in preprocessing steps.

In [14]:
```
#lets look df in this too
df.head()
```

Out[14]:

enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience
8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20
29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	15
11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	5
33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<1
666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>20

In [17]:
```
#suppose I only want city, gender and education level
df=pd.read_csv(r"C:\Users\USER\Downloads\aug_train.csv",usecols=['city','gender','education_level'])
df.head()
```

Out[17]:

	city	gender	education_level
0	city_103	Male	Graduate
1	city_40	Male	Graduate
2	city_21	NaN	Graduate
3	city_115	NaN	Graduate
4	city_162	Male	Masters

Skiprows parameter

In [18]:
```
df
```

Out[18]:

	city	gender	education_level
0	city_103	Male	Graduate
1	city_40	Male	Graduate
2	city_21	NaN	Graduate
3	city_115	NaN	Graduate
4	city_162	Male	Masters
...	...	...	...
19153	city_173	Male	Graduate
19154	city_103	Male	Graduate
19155	city_103	Male	Graduate
19156	city_65	Male	High School
19157	city_67	NaN	Primary School

19158 rows × 3 columns

In [19]:
```
#Now suppose I do not need 3 and 4th rows, look city_21 and 115, now i do not need that row
df=df=pd.read_csv(r"C:\Users\USER\Downloads\aug_train.csv",skiprows=[3,4])
```

In [20]:
```
df.head()
```

Out[20]:

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experienc
0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>2
1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	1
2	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>2
3	21651	city_176	0.764	NaN	Has relevent experience	Part time course	Graduate	STEM	1
4	28806	city_160	0.920	Male	Has relevent experience	no_enrollment	High School	NaN	

dtypes parameter

In [21]:
```
#it is useful when we want to convert datatypes of column. suppose:
df=pd.read_csv(r"C:\Users\USER\Downloads\aug_train.csv",index_col='enrollee_id')
df.head()
```

Out[21]:

enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience
8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20
29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	15
11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	5
33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<1
666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>20

In [22]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19158 entries, 8949 to 23834
Data columns (total 13 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   city                    19158 non-null  object 
 1   city_development_index  19158 non-null  float64
 2   gender                  14650 non-null  object 
 3   relevent_experience     19158 non-null  object 
 4   enrolled_university     18772 non-null  object 
 5   education_level         18698 non-null  object 
 6   major_discipline        16345 non-null  object 
 7   experience              19093 non-null  object 
 8   company_size            13220 non-null  object 
 9   company_type            13018 non-null  object 
 10  last_new_job            18735 non-null  object 
 11  training_hours          19158 non-null  int64  
 12  target                  19158 non-null  float64
dtypes: float64(2), int64(1), object(10)
memory usage: 2.0+ MB
```

In [23]:
```
#here target is float and suppose we want target as integer.
df=pd.read_csv(r"C:\Users\USER\Downloads\aug_train.csv",index_col='enrollee_id',dtype={'target':int})
df.head()
```

Out[23]:

enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience
8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate	STEM	>20
29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate	STEM	15
11561	city_21	0.624	NaN	No relevent experience	Full time course	Graduate	STEM	5
33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate	Business Degree	<1
666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters	STEM	>20

In [24]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19158 entries, 8949 to 23834
Data columns (total 13 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   city                    19158 non-null  object 
 1   city_development_index  19158 non-null  float64
 2   gender                  14650 non-null  object 
 3   relevent_experience     19158 non-null  object 
 4   enrolled_university     18772 non-null  object 
 5   education_level         18698 non-null  object 
 6   major_discipline        16345 non-null  object 
 7   experience              19093 non-null  object 
 8   company_size            13220 non-null  object 
 9   company_type            13018 non-null  object 
 10  last_new_job            18735 non-null  object 
 11  training_hours          19158 non-null  int64  
 12  target                  19158 non-null  int32  
dtypes: float64(1), int32(1), int64(1), object(10)
memory usage: 2.0+ MB
```

Handling dates

In [25]:
```
df6=pd.read_csv(r"C:\Users\USER\Downloads\IPL Matches 2008-2020.csv")
df6.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               816 non-null    int64
 1   city             803 non-null    object
 2   date             816 non-null    object
 3   player_of_match  812 non-null    object
 4   venue            816 non-null    object
 5   neutral_venue    816 non-null    int64
 6   team1            816 non-null    object
 7   team2            816 non-null    object
 8   toss_winner      816 non-null    object
 9   toss_decision    816 non-null    object
 10  winner           812 non-null    object
 11  result           812 non-null    object
 12  result_margin    799 non-null    float64
 13  eliminator       812 non-null    object
 14  method           19 non-null     object
 15  umpire1          816 non-null    object
 16  umpire2          816 non-null    object
dtypes: float64(1), int64(2), object(14)
memory usage: 108.5+ KB
```

In [26]:
```
#Its a cricket match data and I directly use info function to see datatypes, see the date, it is object - means
#Convert it in date
df6=pd.read_csv(r"C:\Users\USER\Downloads\IPL Matches 2008-2020.csv", parse_dates=['date'])
df6.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               816 non-null    int64
 1   city             803 non-null    object
 2   date             816 non-null    datetime64[ns]
 3   player_of_match  812 non-null    object
 4   venue            816 non-null    object
 5   neutral_venue    816 non-null    int64
 6   team1            816 non-null    object
 7   team2            816 non-null    object
 8   toss_winner      816 non-null    object
 9   toss_decision    816 non-null    object
 10  winner           812 non-null    object
 11  result           812 non-null    object
 12  result_margin    799 non-null    float64
 13  eliminator       812 non-null    object
 14  method           19 non-null     object
 15  umpire1          816 non-null    object
 16  umpire2          816 non-null    object
dtypes: datetime64[ns](1), float64(1), int64(2), object(13)
memory usage: 108.5+ KB
```

Converters

In [27]:
```
df6.head()
```

Out[27]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	field	Kolkata Knight Riders
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore
4	335986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	Kolkata Knight Riders	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders

In [29]:
```
#suppose in above dataframe instead of Kolkata Knight rider, I need KKR

def rename(a):
    if a=="Kolkata Knight Riders":
        return "KKR"
```

```
    else:
        return a
```

In [31]:
```
df6=pd.read_csv(r"C:\Users\USER\Downloads\IPL Matches 2008-2020.csv",converters={'team1':rename,'team2':rename}
```

In [32]:
```
df6.head()
```

Out[32]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winner
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	KKR	Royal Challengers Bangalore	field	Kolkata Knight Riders
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chennai Super Kings
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	Delhi Daredevils
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Royal Challengers Bangalore
4	335986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	KKR	Deccan Chargers	Deccan Chargers	bat	Kolkata Knight Riders

Loading a huge dataset (Need to work in chunk)

In [33]:
```
#This below is just an example, suppose we will have 19lakh dateset, where our laptop/Machine
#could not load and handle operation at a time so at that time we should divide data in chunk and need to work
```

In [36]:
```
df6
```

Out[36]:

	id	city	date	player_of_match	venue	neutral_venue	team1	team2	toss_winner	toss_decision	winn
0	335982	Bangalore	2008-04-18	BB McCullum	M Chinnaswamy Stadium	0	Royal Challengers Bangalore	KKR	Royal Challengers Bangalore	field	Kolk Kni Rid
1	335983	Chandigarh	2008-04-19	MEK Hussey	Punjab Cricket Association Stadium, Mohali	0	Kings XI Punjab	Chennai Super Kings	Chennai Super Kings	bat	Chen Su Kir
2	335984	Delhi	2008-04-19	MF Maharoof	Feroz Shah Kotla	0	Delhi Daredevils	Rajasthan Royals	Rajasthan Royals	bat	De Daredev
3	335985	Mumbai	2008-04-20	MV Boucher	Wankhede Stadium	0	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	bat	Ro Challeng Bangal
4	335986	Kolkata	2008-04-20	DJ Hussey	Eden Gardens	0	KKR	Deccan Chargers	Deccan Chargers	bat	Kolk Kni Rid
...	...	...	...	...	...	...	...	...	...	...	
811	1216547	Dubai	2020-09-28	AB de Villiers	Dubai International Cricket Stadium	0	Royal Challengers Bangalore	Mumbai Indians	Mumbai Indians	field	Ro Challeng Bangal
812	1237177	Dubai	2020-11-05	JJ Bumrah	Dubai International Cricket Stadium	0	Mumbai Indians	Delhi Capitals	Delhi Capitals	field	Mum India
813	1237178	Abu Dhabi	2020-11-06	KS Williamson	Sheikh Zayed Stadium	0	Royal Challengers Bangalore	Sunrisers Hyderabad	Sunrisers Hyderabad	field	Sunris Hyderab
814	1237180	Abu Dhabi	2020-11-08	MP Stoinis	Sheikh Zayed Stadium	0	Delhi Capitals	Sunrisers Hyderabad	Delhi Capitals	bat	De Capit
815	1237181	Dubai	2020-11-10	TA Boult	Dubai International Cricket Stadium	0	Delhi Capitals	Mumbai Indians	Delhi Capitals	bat	Mum India

816 rows × 17 columns

In [37]:
```
#Just an example lets divide df6 in chunks
dfs=pd.read_csv(r"C:\Users\USER\Downloads\IPL Matches 2008-2020.csv", chunksize=200)
```

In [38]:
```
for chunks in dfs:
    print(chunks.shape)
```

```
(200, 17)
(200, 17)
(200, 17)
(200, 17)
(16, 17)
```

In [39]:
```
#see it divided our data set 5 sets, 4 set of equal row and column and last is leftover data set.
```