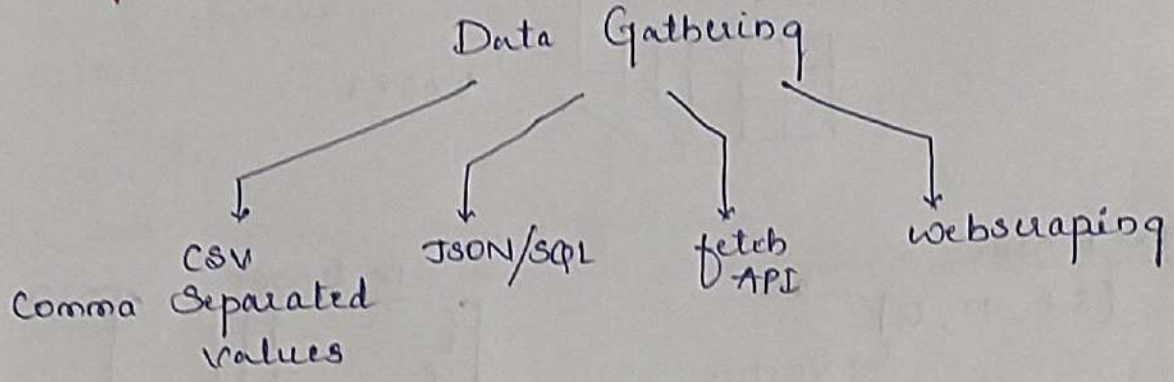
* Gathering data :→



* Working with CSV

1. Importing pandas
   import panda as pd
2. Opening a local csv file → shift tab = Sep(,)
   df = pd.read_csv('aug-train.csv')
3. Opening a csv file from an url.
   import requests
   from io import StringIO
   url = " "
   headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:66.0) Gecko/20100101 Firefox/66.0"}
   req = requests.get(url, headers = headers)
   data = StringIO(req.text)
   pd.read_csv(data)

4. Sep parameter
   pd.read_csv('movie_titles_metadata.tsv', sep='\t')

- provide names to rows & columns as per requirement
  sep='\t', names=['sno', 'name', 'release_year', 'rating', 'votes', 'genres'])

↳ uses the first column as the dataframe index instead of default numeric indices

5. Index_col parameter

```
pd.read_csv('aug_train.csv', index_col='enrollee_id')
```

6. Header parameter

```
pd.read_csv('test.csv', header=1)
```

```
    unnamed    unnamed    unnamed
0   enrollee id  city     gender
1   2934       mumbai     female
```

```
   0   enrollee id   city     gender
0  1   2934          pune     female
1  2   2836          mumbai   female
```

7. Use_col parameters

↳ you can specify which columns you need

```
pd.read_csv('aug_train.csv', usecols=['enrollee_id',
                                      'gender', 'education_level'])
```

8. Squeeze parameters

↳ It displays the required columns

```
pd.read_csv('aug_train.csv', usecols=['gender'],
            squeeze=True)
```

O/p = Gender

female
Male
Male

9. Skiprows / nrows parameter

```
pd.read_csv('aug_train.csv', skiprows=[0,1])
```

" " , nrows=[100])

↳ displays 100 rows

10. Encoding parameter

```
pd.read_csv('zomato.csv', encoding='latin-1')
```

11. Skip bad lines

```
pd.read_csv('BX-Books.csv', sep=';', encoding=
            "latin-1", error_bad_lines=False)
```

↓

Skip the lines which are bad.

12. dtypes parameter

```
pd.read_csv('aug_train.csv', dtype={'target':int})
```

↓

If the values are in float u can change it to integer.

13. Handling Dates

```
pd.read_csv('IPL matches 2008-2020.csv',
            parse_dates=['date'])
```

14. Converters

```
def rename(name):
    if name == "Royal challengers Bangalore":
        return "RCB"
    else
        return name

o/p = rename("Royal challengers Bangalore")
'RCB'

pd.read_csv('IPL Matches 2008-2020.csv',
            converters={'team1':rename})
```

↓

Helps to ~~change~~ convert the names / data

15. na-values parameter
↳ missing values

```
pd.read_csv('aug-train.csv', na_values=['Male'])
```

16. Loading a huge dataset in chunks

```
pd.read_csv('aug_train.csv', chunksize=5000)
```

↳ divide rows

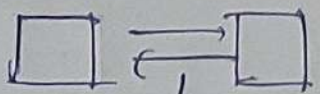
* JSON/SQL :→

```
import pandas as pd
pd.read_json('train.json')
pd.read_json('https://api.exchangerate-api.com/v4/latest
/INR')
```

* SQL :

```
!pip install mysql.connector
import mysql.connector
conn=my-sql.connector.connect(host='localhost', usu='root',
password='', database='world')
pd.read_sql_query("SELECT * FROM city", conn) WHERE
CountryCode Like 'USA'", conn)
```

* Fetching data from an API :→



↳ data pipelines

```
import pandas as pd
import requests
response = requests.get('https://.....')
response.json()['results']
pd.DataFrame(response.json()['results']).head(2)
[['id', 'title', 'overview', 'release_date', 'vote_count']]
```

```
df.head()
```

* Fetching data using Web Scraping :→

AmbitionBox

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
requests.get('https://  .  .  .  .  .  ').text
Soup = BeautifulSoup(Webpage, 'lxml')
print(Soup.prettify())
Company = soup.find_all('div', class_ = 'company-content
                                        - wrapper')
len(company)
Of = 30
name = []
for i in company:
    print(i.find('h2'))

name = []
ratings = []
reviews = []

for i in company:
    name.append(i.find('h2').text.strip())
    rating.append(i.find('p', class = 'rating').
                                    text.strip())
    reviews.append(i.find('a', class_ = 'review-
                                    count').text.stri
                                    -p()
```