Read txt file

```
import pandas as pd
data = pd.read_csv('abc.txt', delimiter = '\t')
```

Create dataframe using excel file

read_excel()

```
import pandas as pd
df = pd.~~excel~~read_excel(r'path', sheet2')
```

O/P

df

Priya IS A GIRL.

Skip the one unnecessary header or add yours.

- CSV file you can skip it using the following command. It is skipping null value

```
df = pd.read_excel(r'C:\Users\path\filename', skiprows=2)
```

call

df ⟶ show output

OR

```
df = pd.read_excel(r'path\filename with extension'
                   header = 3)
```

~~It is also skipping header data and reading is~~ store shows on which rows your header locates

⊙ If you don't have header in your file and you want to add some column header

```
data = pd.read_excel(r'path\filename with extension',
                     header = None, name = ['ps','sj','py'])
```

call
df

	ps	sj	py
0	name	Degree	NaN
1	Priyg	Bcn	"
2	Riyg	B Com	"
3	Shivangi	B Sc	"

→ Means None

Read CSV Continues...

⊙ Read a limited number of rows from the top of the file

```
df = pd.read_excel(r'path\filename with extension',
                   nrows = 3)
```

⊙ If you want to replace your column value with NaN
~~Replace all unwanted value with NaN~~
by specific to column.

```
df = pd.data.read_excel(r'path\B.filename with extension',
     ~~na_values = ['n.a', 'not available']~~)
```

```
na_values = {'columnName': ['na', 'not available'],
             'columnName': ['n.a', 'not available'],
             'columnName': ['n.a', 'not available']})
```

⊙ Replace all unwanted value with NaN.

```
df = pd.read_excel(r'path\foldername with extension',
                   na_values ['n.a', 'not available']
```

Write CSV

⊙ If you want to write your data frame to a csv file.

df.to_csv(r'path\createnewcsvfilename, index=False)

⊙ Write only specific columns

df.to_csv(r'path\newfilename(create), index=False, columns=['columnname', 'columnname'])

⊙ Write Without Header

df.to_csv(r'& path \ ~~columnname~~ newfile', index=False, header=False)

⊙ To append

df.to_csv(r'path\createfilename', mode='a', index=False, header=False)

Using Converters

Step1

```
def convert(co):
    if co == 'nurse':
        return 'senior nurse'
    else:
        return co
```

Step2

```
newdata = pd.read_excel(r'path\ ~~fol~~ excelfilename, converters={'selectcolumn': convert})
```

Call

newdata

Write two data frames in a single excel file.

Using write ExcelWriter(r'path\excelfilename.xlsx') as wr:

dataframe.to_excel(wr, sheet_name='Sheet1')
dataframe.to_excel(wr, sheet_name='Sheet2')

2 different

Basics

Shape

df.shape → provide the shape of the Dataframe.

head()

df.head() → provides the first 5 rows of the Dataframe
df.head(3) → provides first 3

tail()

df.tail() → provides the last 5 rows of the Dataframe
df.tail() → provides " last 2 rows " " "

Columns

df.columns

To print a specific column data

df.name or df['name'] → where name is the column name in df dataframe.

To get some columns

df['id', 'name', 'age']

Index

df.index → to view the index assigned to the dataframe.

df.set_index('columnname') → To change the index to some other column value.

df.reset_index() → Reset index to previous stage.