

install and import

installing pandas

```
pip install pandas
```

pip install pandas

```
import pandas as pd
```

Reading and describing

pd -> pandas

df-> dataframe

to read a file into a dataframe

```
df= pd.read_csv('filename')
```

look at the first 5 lines

```
df.head()
```

to describe df

```
df.describe()
```

```
df.info()
```

to print all the column names

```
telecom_data.columns
```

to get the dimension of df

```
df.shape
```

Sorting and filtering

Rows and columns

to delete a row - [axis=0 means rows]

```
new_df = df.drop([2,3],axis = 0)
```

to delete a column- [axis=1 means columns]

```
new_df = df.drop(['col1', 'col2'],axis = 1)
```

Df manipulation

create or edit a new column

```
df['new_column'] = 5 #this creates a new column with all values = 5
```

create a new column

```
df['new_column'] = [list of values] #this creates a new column with list of values
```

NOTE: df['new_column'] = [list of values] throws an error if the no of items in list is not equal to no of rows

create or edit a new row

```
df.loc[in dex_of_row] = [list of items]
```

NOTE: df.loc[in dex_of_row] = [list of items] throws an error if the no of items in list is not equal to no of columns

Selection

```
df[col] Returns column with label col as Series
```

```
df[[col1, col2]] Returns multiple columns as a new DataFrame
```

	Country	Capital	Population
1	Belgium	Brussels	11190846
2	India	New Delhi	1303171035
3	Brazil	Brasilia	207847528

```
df.iloc([0], [0]) --> 'Belgium' | s.iloc[0] | Selection by position (0th position on row and column)
```

```
df.loc([0], ['Country']) --> 'Belgium'
```

```
df.ix[2] -->
Country    Brazil
Capital    Brasilia
Population 207847528
```

```
df.ix[1, 'Capital'] --> 'New Delhi'
```

```
df.iloc[0,:] | select First row
```

Data Cleaning

```
df.set_index('column_name') Change the index with a new column name
```

```
df.rename(columns={'old_col_name1','new_col_name2','new_col_name3'}) Rename the columns
```

```
pd.isnull() Checks for null values. Returns Boolean Array
```

```
pd.notnull() Opposite of pd.isnull()
```

```
df.dropna() Drop all rows that contain null values
```

```
df.dropna(axis=1) Drop all columns that contain null values
```

```
df.dropna(axis=1, threshold=n) Drop all rows have less than n non null
```

```
df.fillna(x) Replace all null values with x
```

JOIN/COMBINE

sort

sorting can be done column wise - default is ascending

```
df.sort_values(by='Total day charge')
```

```
df.sort_values(col1)
```

Sort values by col1 in ascending order (use ascending=False for descending sort)

```
df.sort_values([col1, col2], ascending=[True, False])
```

Sort values by col1 in ascending order then col2 in descending order

Filtering

```
df[condition] #eg: df[df['col1'] > 5]
```

```
df[df['col'] > 0.5]
```

Rows where the column col is greater than 0.5

```
df[(df[col] > 0.5) & (df[col] < 0.7)]
```

Rows where 0.7 > col > 0.5

Inplace

NOTE

`df.merge(df2)` gives you a copy of df merged with df2. you may save it to a new variable. ex `df3=df.merge(df2)`

if you want to merge df2 to df right away use `inplace.df.merge(df2, inplace=True)`

```
df1.append(df2)
```

```
pd.concat([df1, df2], axis=1)
```

```
df1.join(df2, on='col1, how='inner')
```

left = takes the index of left df

right =takes the index of left

outer = union of both keys

inner = intersection of both keys



By [sanjeev95](https://sanjeev95.cheatography.com/sanjeev95/)
[cheatography.com/sanjeev95/](https://sanjeev95.cheatography.com/sanjeev95/)

Published 20th January, 2020.
Last updated 22nd January, 2020.
Page 1 of 2.

Sponsored by [Readable.com](https://readable.com)
Measure your website readability!
<https://readable.com>