#import library:

Import pandas library:

A screenshot of a cell phone

Description automatically generated

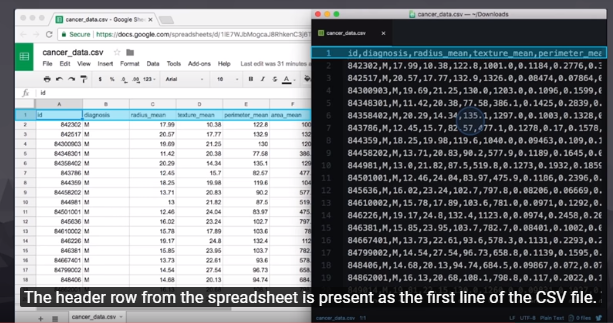
Where pandas is a library that is used to handle a data frame

#read file:

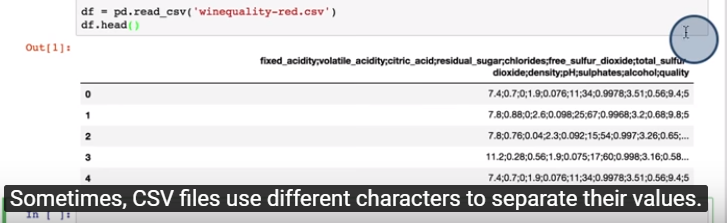
CSV: comma separated values. Csv file is a text file. It is not containing any function statements. It just contains values/ raw data.

Unlike xlsx file which is format of excel that contains all your functions!

You can see the csv file by a text editor like sublime for example, first row on spread cheat will be the first line on text editor. And so on. Each cell separated by a comma to the next cell and so on



-import csv file by pandas in python as dataframe:

­­­

#explore the data:

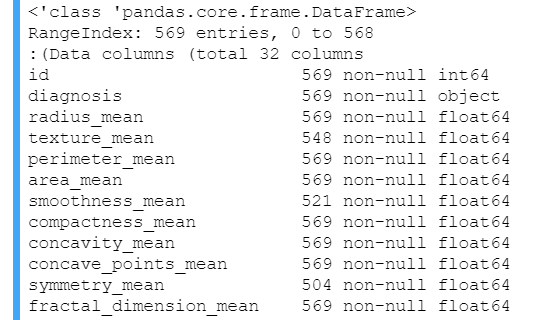
Now the dataframe is there, let explore it by pandas commands in order to find the issues of data and cleaning it:

df.info()

it displays a summary of the dataframe, including the type of each column, total number of non-null values of each column.

It is an important command to identify the most common issues of dataframe, missing values and wrong datatype.

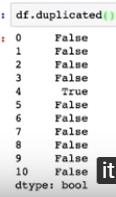
Note, string is called as object in pandas



df.duplicated()

duplicated data find if any rows has duplicated

We can find the duplicated data by df.duplicated()



df.duplicated().sum()

give the total number of diuplicated in the data frame.

A screenshot of a cell phone

Description automatically generated

df.isnull().sum()

give the number of missing/null value in each column of our data frame:

A screenshot of a cell phone

Description automatically generated

df.descirbe()

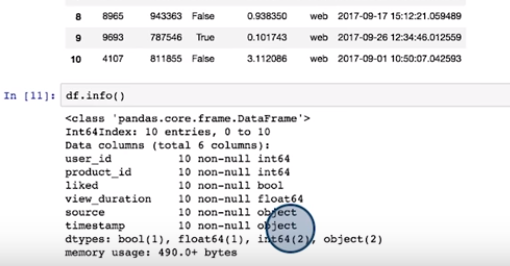
Give the descriptive statistics of each column such as mean, min, max, etc.

# fixing the data:

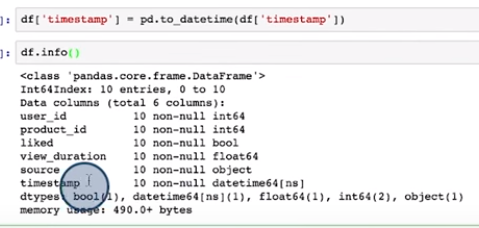
After we found the issues, we have now deal with it:

-Convert datatype:

Most commn wrong data type is of date time column is came always string/object and we need to convert it to datatime type:



To convert it to date time will use pd.to\_datetime(df.column)



To convert string/object to int or float will use df.column.astype(int)

A screenshot of a cell phone

Description automatically generated

-drop duplicated

A close up of a logo

Description automatically generated

-drop na

drop according any column has na

A screenshot of a cell phone

Description automatically generated

drop according one/specific column

A screenshot of a cell phone

Description automatically generated

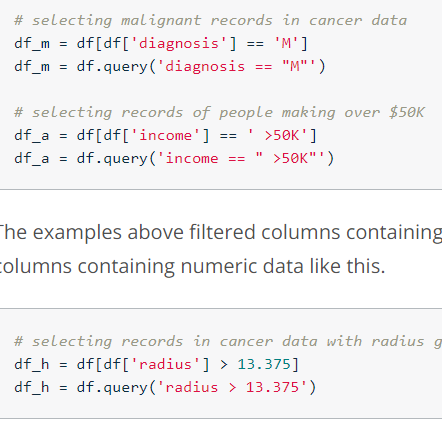
-drop some column:

New\_df=df.drop(column=[‘column1’,’column2’])

#filer/mask/ query the data frame to subset the specific contents:

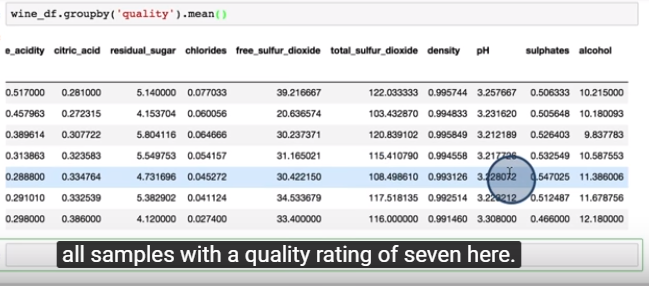


Note: it can also make filtering by using a function called a query:

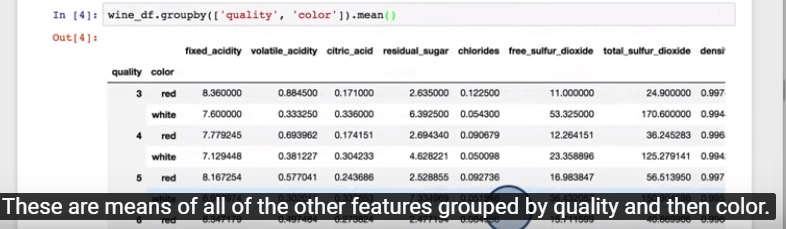


#grouping:

Group the dataset acording spesefic column. Where it used to rearrange the datadet according spesfic column to be as the index of data. started from lowest value by defult:



Also, it can grouping by muliple colmn. Group the data by first column then grouping each group by second column



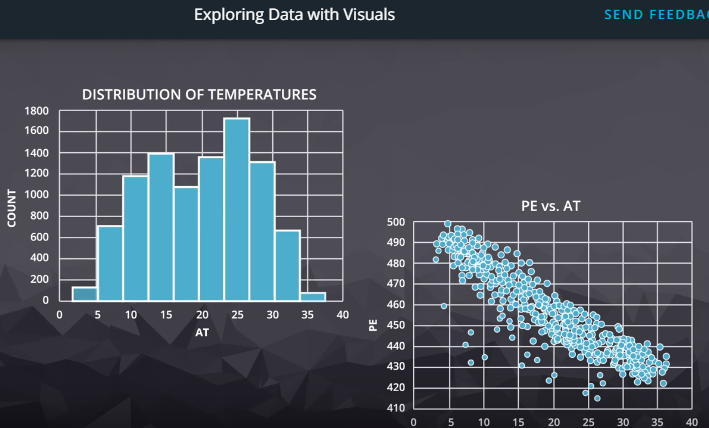
#sorting()

rearrange the dataframe according to a spesfic column. By default, it will sort the values from lowest to the highest value, we will use the ascending attribute to sort the values from highest to lowest.

A screenshot of a cell phone

Description automatically generated

#plotting:

Plotting is useful to find the pattern of the data that are be useful to make analysis. Histogram and scatterplot can use to determine which variables will be used in analysis. Where for example it can use histogram to view the distibuation of tempretures. And use scatterplot to find the relationship/correlation between temperature and power output. Where showing here the correlation is strong so we can use these fetures for analsysis

Plotting on pandas:

Pandas plot is wrapper/based on/according on matplotlib library/functions. That is why you see them used interchangeably! However, you need to use matplotlib for more customization if you need.

-histogram.

#by using general plotting function in pandas, plot function:

df.plot(kind=’hist’)

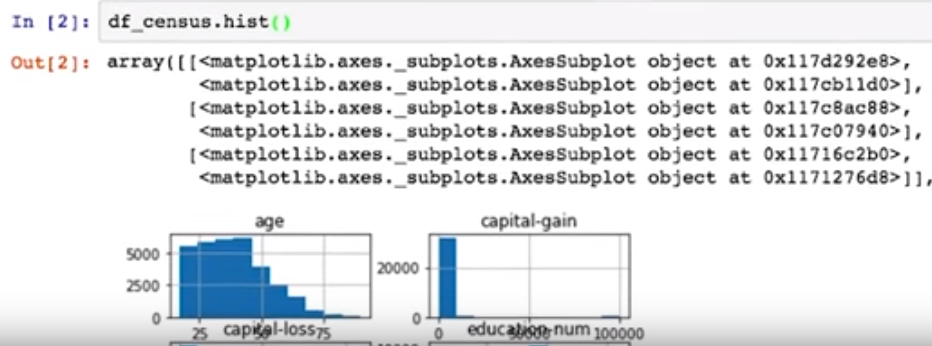
df.plot.hist()

# by using direct hist function:

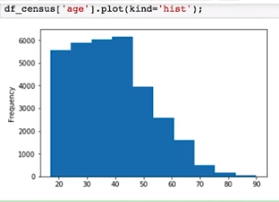
df.hist()

df.hist(figsize=(8,8));

here we can increase size of plot by figsize attrbuite. And semicolon at the end is useful if you want to remove the texting that appear before the plot

Histogram can do also for specific column also :

-barchart:

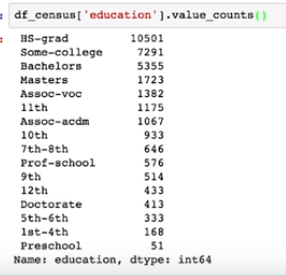
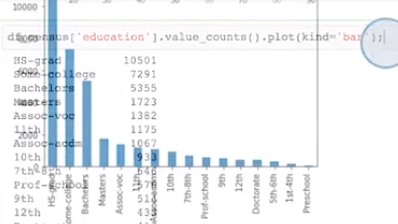
barchart need to plot the frequency/ number of each unique value in the column. To get this we need first to use value\_counts() function then plotting its result

-Df.column.count() or Df[‘column’]. count()

is used to count the total number of content/index/ values/ number of non null value in the column in the column: where example of same value you get from info() but for specific column

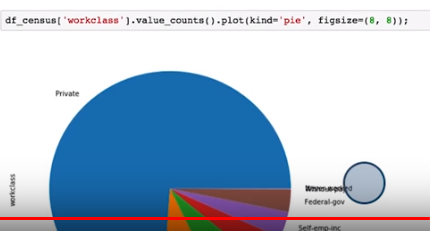
-Df.column.value\_counts() or Df[‘column’].value\_counts()

Is used to count number of each unique index/ value in the column

-piechart:

Similar of bar chart, we need to use value\_counts() to count the frequency first:

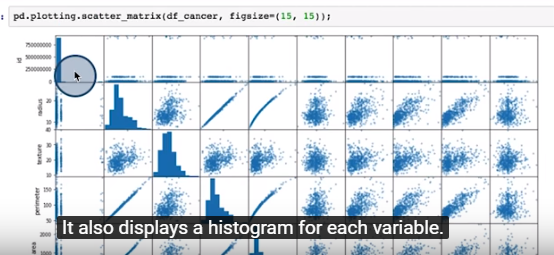


-scatter plot:

It use to find the correlation between features.

-For quick visualize the correlationship for all numerical variables/ features of data frame with scatter plot called by

scatter\_matrix(df) or pd.plotting.scatter\_matrix(df)



-for single scatter plot comparing between two variable/feature, you need to specify these two features one as y and ather as x axsis. Using :

Df.plot.scatter(x=’column1’,y=’column2’) or df.plot(kind=’scatter’, x=’column1’, y=’column2’)

-box plot:

To find the 3Q median. For specific column using

df.column.plot.box()

or df[‘column’].plot.box()

or df[‘column’].plot(kind=’box’)

