EDA Cheat Sheet

Exploratory Data Analysis

Importing Data

Function	Description
pd.read_csv(file_name)	Read from a csv file
pd.read_csv(file_name, sep='\t')	Read from a csv file separated by tabs
pd.read_excel(file_name)	Read from excel file
pd.read_table(file_name)	Read from a delimited text file
pd.read_sql(sql_query, connection_object)	Read from a database
pd.read_json("string, url or file")	Read from a json string, url or a file
pd.read_html(URL)	Read from a url or a file

Data Exploration

Function	Description
df.info()	Provides information like datatype, shape of the dataset and memory usage
df.describe()	Provides information like count, mean, min, max, standard deviation and quantiles
df.shape	Returns the shape of the dataset
df.head()	Prints top 5 rows of the dataset
df.tail()	Prints last 5 rows of the dataset
df.column_name.value_counts()	Returns count of the unique classes in a column
df.count()	Returns total number of observations in each column
df.column_name.unique()	Returns unique classes in the column

Filter data

Function	Description
df.loc[condition]	Returns the rows based on one condition
df[(condition) & (condition)]	Returns the rows based on two conditions (& operator)
df[(condition) | (condition)]	Returns the rows based on two conditions (| operator)
df.loc[(condition) & (condition)]	Returns the rows based on two conditions (& operator) using loc
df.loc[(condition) | (condition)]	Returns the rows based on two conditions (| operator) using loc

Renaming Columns and Indices

Function	Description
df.columns = ['Column 1', 'Column 2', ...]	Rename the columns by passing a list
df.rename(columns={'old_name': 'new_name'})	Rename the columns using rename function
df.rename(index={'old_name': 'new_name'})	Rename the indices using rename function
df.set_index("Column_name")	Set the column as indices

Statistical Functions

Function	Description
df.mean()	Finds the mean of every column
df.median()	Finds the median of every column
df.column_name.mode()	Finds the mode of a column
df.corr()	Creates a correlation table
df.max()	Finds the max value from a column
df.min()	Finds the min value from a column
df.std()	Finds the standard deviation of each column
df.cov()	Creates a covariance matrix

Sort and Group By

Function	Description
df.sort_values(col, ascending)	Sorts the dataframe on the basis of a column
df.sort_values([col1, col2, ...], ascending)	Sorts the dataframe on the basis of multiple columns
df.groupby(column_name)	Groups a dataframe by the column name
df.groupby([column_1, column_2, ...])	Groups a dataframe by multiple column names
df.groupby(column_1)[column_2].mean()	Finds the mean of the column from the group
df.groupby(column_1).agg(np.mean())	Finds the mean of all the columns from the group
df.apply(function, axis)	Applies a function on all the columns (axis=1) or rows (axis=0) of a dataframe

Append, Concat, Join, Merge

Function	Description
df1.append(df2)	Appends a dataframe df2 to df1
pd.concat([df1, df2], axis)	Concates multiple dataframes based on axis value
df1.join(df2,on=col1,how='inner')	Joins a dataframe df2 with df1 on some column
pd.merge(left, right, on, how)	Merge two columns on a column

Null Value Analysis and Data Cleaning

Function	Description
df.isnull()	Returns True where the value is null
df.isnull().sum()	Returns the count of null values in each column
df.isnull().sum().sum()	Returns the count of all the null values from a dataframe
df.notnull()	Returns True where the value is not null
df.dropna(axis, thresh)	Drops the columns (axis=1) or rows (axis=0) having null values based on threshold
df.fillna(value)	Fills the cells having null values with the passed value
df.replace('old_value', 'new_value')	Replace a value by a new value
df.replace([old_1, old_2], [new_1, new_2])	Replace multiple values with multiple new values
df.column_name.astype('data_type')	Change the data type of the column

Selecting rows and columns

Function	Description
df.column_name	Select the column using. Note: a column having white spaces cannot be selected by this method
df["column_name"]	Select a column
df[["column_name_1", "column_name_2", ...]]	Select multiple columns
df.iloc[: , :]	Pass the row and column start and end indices to extract selected rows and columns
df.iloc[index_position]	Pass the index position to extract rows
df.loc[index_value]	Pass the index value to extract rows

Write Data

Function	Description
df.to_csv(file_name)	Write the data from df to a csv file
df.to_excel(file_name)	Write the data from df to an excel file
df.to_html(file_name)	Write the data from df to a html file
df.to_sql(table_name, connection_object)	Write the data from df to a table in a database
df.to_json(file_name)	Write the data from df to a json file

Duplicates

Function	Description
df.duplicated(keep='first')	Find the first occuring duplicates.
df.drop_duplicates(keep, inplace)	Drop the duplicate rows