BASIC OPERATORS IN PYTHON

ARITHMETIC OPERATORS	DESCRIPTION	SYNTAX
+	Adds two operands	x+y
-	Subtracts two operands	x-y
*	Multiplies two operands	x*y
/	Division(float): divides the first operand by second	x/y
//	Division(floor): divides the first operand by second	x//y
**	Exponential of first operand raised to second	x**y
%	Modulus: returns the remainder when the first operand is divided by the second	x%y
RELATIONAL OPERATORS	**DESCRIPTION**	**SYNTAX**
>	Greater than: True if left operand is greater than the right	x > y
<	Less than: True if left operand is less than the right	x < y
==	Equal to: True if both operands are equal	x == y
!=	Not equal to - True if operands are not equal	x != y
>=	Greater than or equal to: True if left operand is greater than or equal to the right	x >= y
<=	Less than or equal to: True if left operand is less than or equal to the right	x <= y
LOGICAL OPERATORS	**DESCRIPTION**	**SYNTAX**
and	Logical AND: True if both the operands are true	x and y
or	Logical OR: True if either of the operands is true	x or y
not	Logical NOT: True if operand is false	not x
BITWISE OPERATORS	**DESCRIPTION**	**SYNTAX**
&	Bitwise AND	x & y
|	Bitwise OR	x | y
~	Bitwise NOT	~x
^	Bitwise XOR	x ^ y
>>	Bitwise right shift (prints bitwise right shift operation)	x>>
<<	Bitwise left shift (prints bitwise right shift operation)	x<<
ASSIGNMENT OPERATORS	**DESCRIPTION**	**SYNTAX**
=	Assign value of right side of expression to left side operand	x = y + z
+=	Add AND: Add right side operand with left side operand and then assign to left operand	a=a+b
-=	Subtract AND: Subtract right operand from left operand and then assign to left operand	a=a-b
*=	Multiply AND: Multiply right operand with left operand and then assign to left operand	a=a*b
/=	Divide AND: Divide left operand with right operand and then assign to left operand	a=a/b
%=	Modulus AND: Takes modulus using left and right operands and assign result to left operand	a=a%b
//=	Divide(floor) AND: Divide left operand with right operand and then assign the value(floor) to left operand	a=a//b
**=	Exponent AND: Calculate exponent(raise power) value using operands and assign value to left operand	a=a**b
&=	Performs Bitwise AND on operands and assign value to left operand	a=a&b
|=	Performs Bitwise OR on operands and assign value to left operand	a=a|b
^=	Performs Bitwise XOR on operands and assign value to left operand	a=a^b
>>=	Performs Bitwise right shift on operands and assign value to left operand	a=a>>b
<<=	Performs Bitwise left shift on operands and assign value to left operand	a= a << b

DATA TYPES

EXAMPLE	DATA TYPE
x = "Hello World"	str
x = 20	int
x = 20.5	float
x = 1j	complex
x = ["apple", "banana", "cherry"]	list
x = ("apple", "banana", "cherry")	tuple
x = range(6)	range
x = {"name" : "John", "age" : 36}	dict
x = {"apple", "banana", "cherry"}	set
x = frozenset({"apple", "banana", "cherry"})	frozenset
x = True	bool
x = b"Hello"	bytes
x = bytearray(5)	bytearray

STRING OPERATIONS

FUNCTION	DESCRIPTION
mystring[:N]	Extract N number of characters from start of string.
mystring[-N:]	Extract N number of characters from end of string
mystring[X:Y]	Extract characters from middle of string, starting from X position and ends with Y
str.split(sep=' ')	Split Strings
str.replace(old_substring, new_substring)	Replace a part of text with different substring
str.lower()	Convert characters to lowercase
str.upper()	Convert characters to uppercase
str.contains('pattern', case=False)	Check if pattern matches (Pandas Function)
str.extract(regular_expression)	Return matched values (Pandas Function)
str.count('sub_string')	Count occurrence of pattern in string
str.find( )	Return position of sub-string or pattern
str.isalnum()	Check whether string consists of only alphanumeric characters
str.islower()	Check whether characters are all lower case
str.isupper()	Check whether characters are all upper case
str.isnumeric()	Check whether string consists of only numeric characters
str.isspace()	Check whether string consists of only whitespace characters
len( )	Calculate length of string
cat( )	Concatenate Strings (Pandas Function)
separator.join(str)	Concatenate Strings

BUILT-IN FUNCTIONS

FUNCTION	DESCRIPTION
abs(x)	Returns absolute value of a number
ascii()	Returns ASCII value of a string
bin()	Returns binary representation of an integer
bool()	Converts value to boolean (True / False)
complex()	Used to convert numbers or string into a complex number
dict()	A constructor which creates a dictionary
dir()	Returns a list of names in the current local space
float()	Returns a floating point number from a number or string
format()	Returns a formatted representation of a given value
frozenset()	Returns an immutable frozenset object
globals()	Returns a dictionary containing the variables defined in the global namespace
help()	Used to get help related to the object passed during the call
input()	Used to get input from the user
int()	Returns the integer value of a number
iter()	Returns an iterator object
list()	Creates a list
locals()	Returns a dictionary containing the variables defined in the local namespace
map()	Returns a list of results after applying a given function to each item of an iterable
max()	Returns a maximum value in the sequence
min()	Returns a minimum value in the sequence

next()	Used to fetch the next item in the sequence
open()	Opens a file and returns a corresponding file object
pow()	Returns power of a number
print()	Prints the given object
range()	Returns an immutable sequence of numbers
round()	Round-off the digits of a number
reversed()	Returns the reversed iterator of the given sequence
set()	Returns unique values in a given sequence
slice()	To get slice of elements from a collection of elements
sorted()	To sort the elements
str()	Returns a string
sum()	To get the sum of numbers in an iterable
tuple()	Creates a tuple
type()	Returns the type of a specified object
zip()	Returns a zip object, which maps a similar index of multiple containers

LIST OPERATIONS

FUNCTION	DESCRIPTION
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list

TUPLE OPERATIONS

FUNCTION	DESCRIPTION
all()	Returns true if all elements are true or if tuple is empty
any()	return true if any element of the tuple is true. if tuple is empty, return false
len()	Returns length of the tuple or size of the tuple
enumerate()	Returns enumerate object of tuple
max()	return maximum element of given tuple
min()	return minimum element of given tuple
sum()	Sums up the numbers in the tuple
sorted()	input elements in the tuple and return a new sorted list
tuple()	Convert an iterable to a tuple.

SET OPERATIONS

FUNCTION	DESCRIPTION
add()	Adds an element to a set
remove()	Removes an element from a set. If the element is not present in the set, raise a KeyError
clear()	Removes all elements form a set
copy()	Returns a shallow copy of a set
pop()	Removes and returns an arbitrary set element. Raise KeyError if the set is empty
update()	Updates a set with the union of itself and others
union()	Returns the union of sets in a new set
difference()	Returns the difference of two or more sets as a new set
difference_update()	Removes all elements of another set from this set
discard()	Removes an element from a set if it is a member. (Do nothing if the element is not in the set)
intersection()	Returns the intersection of two sets as a new set
intersection_update()	Updates the set with the intersection of itself and another
isdisjoint()	Returns True if two sets have a null intersection
issubset()	Returns True if another set contains this set
issuperset()	Returns True if this set contains another set
symmetric_difference()	Returns the symmetric difference of two sets as a new set
symmetric_difference_update()	Updates a set with the symmetric difference of itself and another

DICTIONARY OPERATIONS

FUNCTION	DESCRIPTION
copy()	They copy() method returns a shallow copy of the dictionary.
clear()	The clear() method removes all items from the dictionary.
pop()	Removes and returns an element from a dictionary having the given key.
popitem()	Removes the arbitrary key-value pair from the dictionary and returns it as tuple.
get()	It is a conventional method to access a value for a key.
dictionary_name.values()	returns a list of all the values available in a given dictionary.
str()	Produces a printable string representation of a dictionary.
update()	Adds dictionary dict2's key-values pairs to dict
setdefault()	Set dict[key]=default if key is not already in dict
keys()	Returns list of dictionary dict's keys
items()	Returns a list of dictionary's (key, value) tuple pairs
has_key()	Returns true if key in dictionary dict, false otherwise
fromkeys()	Create a new dictionary with keys from seq and values set to value.
type()	Returns the type of the passed variable.
cmp()	Compares elements of both dict.

CONDITIONAL STATEMENTS

STATEMENT	SYNTAX	EXAMPLE
if statement	if <expression>:	if x < y:
	<body; body is executed if expression is true>	print('yes')
if.. else statement	if <expression>:	if x < y:
	<body; body is executed if expression is true>	print('yes')
	else:	else:
	<body; body is executed if expression in 'if' is false>	print('no')
Nested if-else statement	if <expression>:	if time == 'morning':
	<body; body is executed if expression is true>	print('Good morning')
	elif <expression>:	elif name == 'afternoon':
	<body; body is executed if 'elif' expression is true>	print('Good afternoon')
	elif <expression>:	elif name == 'evening':
	<body; body is executed if 'elif' expression is true>	print('Good evening')
	....	...
	else:	else:
	<body; body is executed if expression in 'if' is false>	print("Good day")
For loop	for i in sequence:	for val in range(11):
	<statement(s) of for >	print(val)
While loop	while <expression>:	while value < 10:
	<statement(s) of while; is executed till expression is true >	print(value)
List Comprehension	[expression for item in list if conditional]	[x : x**2 for i in range(100) if x%2 ==0]

BASIC OPERATORS

LOGIC FUNCTIONS	DESCRIPTION	SYNTAX
1. Truth value testing		
all	All array elements along given axis are True	all(a[, axis, out, keepdims])
any	Any array elements along given axis are True	any(a[, axis, out, keepdims])
2. Array contents		
isfinite	Test element-wise finiteness	isfinite(x,l[, out, where, casting, order])
isinf	Test element-wise for positive or negative infinity	isinf(x,l[, out, where, casting, order])
isnan	Test element-wise for NaN and return result as a boolean array	isnan(x,l[, out, where, casting, order])
isneginf	Test element-wise for negative infinity	isneginf(x[,out])
isposinf	Test element-wise for positive infinity	isposinf(x[,out])
3. Array type testing		
iscomplex	Return True if input element is complex	iscomplex(x)
iscomplexobj	Check for a complex number array	iscomplexobj(x)
isfortran	Return True if the array is fortran contiguous	isfortran(a)
isreal	Return True if input element is real	isreal(x)
isrealobj	Return True if x is a not complex	isrealobj(x)
isscaler	Return True if the type of num is scalar type	isscaler(num)

4. Comparison		
allclose	Return True if two array are element wise equal	allclose(a1, a2)
isclose	Return a boolean array if two array are element wise equal	isclose(a1, a2)
array_equal	True if two arrays have same shape and elements	array_equal(a1, a2)
array_equiv	Return True if input array are shape consistent and all elements equal	array_equiv(a1, a2)
greater	Return True if (x1>x2)	greater(x1, x2)
greater_equal	Return True if (x1>=x2)	greater_equal(x1, x2)
less	Return True if (x1<x2)	less(x1, x2)
less_equal	Return True if (x1=<x2)	less_equal(x1, x2)
equal	Return True if (x1==x2)	equal(x1, x2)
not_equal	Return True if (x1!=x2)	not_equal(x1, x2)

BUILT-IN FUNCTIONS IN NUMPY

FUNCTION	DESCRIPTION
1. Basic function	
ndim()	Return number of axes or rank of the array
shape()	Return tuple containing dimensionality of dataframe
size()	Total number of elements
2. Trigonometric function	
sin()	Trigonometric sine
cos()	Trigonometric cosine
tan()	Trigonometric tangent
arcsin()	Inverse sine
arccos()	Inverse cosine
arctan()	Inverse tangent
hypot()	Returns hypotenuse of right angled triangle
degree()	Convert angles from radians to degrees
radianns()	Convert angles from degrees to radians
3. Rounding	
around()	Evenly round to the given number of decimal
round()	Round an array to the given number of decimals
floor()	Return floor of input
ceil()	Return the ceiling of the input
truncate()	Return truncate value of the input before and after some index
4. Sums, products, differences	
prod()	Return product of array elements
sum()	Return sum of array elements
nanprod()	Return product of array elements over a given axis treating a not a number as ones
nansum()	Returns sum of array elements over a given axis treating not a number as zero
diff()	Calculate nth discrete difference along given axis
gradient	Return gradient of an n- dimensional array
cross	Returns cross product of two vectors
5. Exponents and logarithms	
exp()	Calculate exponential of all elements
log()	Natural logarithm element- wise
6. Arithmetic operations	
add()	Add arguments
reciprocal()	Return reciprocal of arguments
negative()	Numerical negative
multiply()	Multiply arguments
divide()	Divide arguments element wise
power()	First array element raised to powers from second array
subtract()	Subtract arguments element wise

BUILT IN FUNCTIONS IN PANDAS

FUNCTION	DESCRIPTION
add()	Returns addition of dataframe and other, element-wise (binary operator add)
sub()	Returns subtraction of dataframe and other, element-wise (binary operator sub)
mul()	Returns multiplication of dataframe and other, element-wise (binary operator mul)
div()	Returns floating division of dataframe and other, element-wise (binary operator truediv)
unique()	Method extracts the unique values in the dataframe
nunique()	Returns count of the unique values in the dataframe
value_counts()	Method counts the number of times each unique value occurs within the Series

columns()	Returns the column labels of the dataframe
axes()	Returns a list representing the axes of the dataframe
isnull()	Method creates a Boolean Series for extracting rows with null values
notnull()	Method creates a Boolean Series for extracting rows with non-null values
between()	Method extracts rows where a column value falls in between a predefined range
isin()	Method extracts rows from a dataframe where a column value exists in a predefined collection
dtypes()	Returns a Series with the data type of each column. The result's index is the original dataframe's columns
astype()	Method converts the data types in a Series
values()	Returns a Numpy representation of the dataframe i.e. only the values in the DataFrame will be returned, the axes labels will be removed
sort_values()- Set1, Set2	Method sorts a data frame in Ascending or Descending order of passed Column
sort_index()	Method sorts the values in a dataframe based on their index positions or labels instead of their values but sometimes a data frame is made out of two or more data frames and hence later index can be changed using this method
loc[]	Method retrieves rows based on index label
iloc[]	Method retrieves rows based on index position
ix[]	Method retrieves dataframe rows based on either index label or index position. This method combines the best features of the .loc[] and .iloc[] methods
rename()	Method is called on a dataframe to change the names of the index labels or column names
columns()	Method is an alternative attribute to change the coloumn name
drop()	Method is used to delete rows or columns from a dataframe
pop()	Method is used to delete rows or columns from a dataframe
sample()	Method pulls out a random sample of rows or columns from a dataframe
nsmallest()	Method pulls out the rows with the smallest values in a column
nlargest()	Method pulls out the rows with the largest values in a column
shape()	Returns a tuple representing the dimensionality of the DataFrame
ndim()	Returns an 'int' representing the number of axes / array dimensions. Returns 1 if Series, otherwise returns 2 if DataFrame
dropna()	Method allows the user to analyze and drop Rows/Columns with Null values in different ways
fillna()	Method manages and let the user replace NaN values with some value of their own
rank()	Values in a Series can be ranked in order with this method
query()	Method is an alternate string-based syntax for extracting a subset from a DataFrame
copy()	Method creates an independent copy of a pandas object
duplicated()	Method creates a Boolean Series and uses it to extract rows that have duplicate values
drop_duplicates()	Method is an alternative option to identifying duplicate rows and removing them through filtering
set_index()	Method sets the DataFrame index (row labels) using one or more existing columns
reset_index()	Method resets index of a Data Frame. This method sets a list of integer ranging from 0 to length of data as index
where()	Method is used to check a Data Frame for one or more condition and return the result accordingly. By default, the rows not satisfying the condition are filled with NaN value
merge	Merge dataframe or series by join
concat()	Concatenate pandas objects along a particular axis
get_dummies()	Convert categorical variables into dummy variables
isna()	Detect missing values for an array
isnull()	Detect missing values for an array
count()	Number of non-NA observations
mean()	Mean of values
median ()	Arithmetic median of values
max()	Maximum number
min()	Minimum number

READING FILES FROM SOURCES

STATISTIC	DESCRIPTION
read_csv	Load delimited data from a file, URL, or file-like object; use comma as default delimiter
read_excel	Read tabular data from an Excel XLS or XLSX file
read_hdf	Read HDF5 files written by pandas
read_html	Read all tables found in the given HTML document
read_json	Read data from a JSON (JavaScript Object Notation) string representation
read_sql	Read the results of a SQL query (using SQLAlchemy) as a pandas DataFrame

SUMMARY STATISTICS

STATISTIC	DESCRIPTION	SYNTAX
Mean	The sum of a collection of numbers divided by the count of numbers in the collection.	statistics.mean()
Median	The middle most value in the list of numbers	statistics.median()
Mode	The value in the list of numbers which has the highest frequency	statistics.mode()

Variance	Measure of dispersion of a set of values from the mean.	statistics.variance()
Standard Deviation	Measure of the amount of variation or dispersion of a set of values from the mean. (square of variance.	statistics.stdev()
Skewness	Skewness is a measure of the symmetry in a distribution.	scipy.stats.skew()
Kurtosis	Measure of peakedness (or flatness)	scipy.stats.kurtosis()
Correlation	Finds correlation of numeric features	corr()

VISUALIZATION

Matplotlib **import matplotlib.pyplot as plt**

CHART	DESCRIPTION
acorr	Plot the autocorrelation of x
annotate	Annotate the point xy with text text
arrow	Add an arrow to the axes
axes	Add an axes to the current figure and make it the current axes
axhline	Add a horizontal line across the axis
axhspan	Add a horizontal span (rectangle) across the axis
axis	Convenience method to get or set some axis properties
axvline	Add a vertical line across the axes
axvspan	Add a vertical span (rectangle) across the axes.
bar	Make a bar plot.
barbs	Plot a 2D field of barbs.
barh	Make a horizontal bar plot.
box	Turn the axes box on or off on the current axes.
boxplot	Make a box and whisker plot.
broken_barh	Plot a horizontal sequence of rectangles.
clf	Clear the current figure.
clim	Set the color limits of the current image.
close	Close a figure window.
cohere	Plot the coherence between x and y.
colorbar	Add a colorbar to a plot.
eventplot	Plot identical parallel lines at the given positions.
figlegend	Place a legend on the figure.
fignum_exists	Return whether the figure with the given id exists.
figtext	Add text to figure.
figure	Create a new figure.
gca	Get the current Axes instance on the current figure matching the given keyword args, or create one.
gcf	Get the current figure.
grid	Configure the grid lines.
hexbin	Make a hexagonal binning plot.
hist	Plot a histogram.
hist2d	Make a 2D histogram plot.
legend	Place a legend on the axes.
locator_params	Control behavior of major tick locators.
loglog	Make a plot with log scaling on both the x and y axis.
magnitude_spectrum	Plot the magnitude spectrum.
margins	Set or retrieve auto scaling margins.
matshow	Display an array as a matrix in a new figure window.
minorticks_off	Remove minor ticks from the axes.
minorticks_on	Display minor ticks on the axes.
pause	Pause for interval seconds.
pcolor	Create a pseudocolor plot with a non-regular rectangular grid.
pcolormesh	Create a pseudocolor plot with a non-regular rectangular grid.
phase_spectrum	Plot the phase spectrum.
pie	Plot a pie chart.
plot	Plot y versus x as lines and/or markers.
plot_date	Plot data that contains dates.
savefig	Save the current figure.
sca	Set the current Axes instance to ax.
scatter	A scatter plot of y vs x with varying marker size and/or color.
show	Display a figure.
stackplot	Draw a stacked area plot.
stem	Create a stem plot.
step	Make a step plot.
subplot	Add a subplot to the current figure.

subplots	Create a figure and a set of subplots.
subplots_adjust	Tune the subplot layout.
suptitle	Add a centered title to the figure.
text	Add text to the axes.
thetagrids	Get or set the theta grid lines on the current polar plot.
tick_params	Change the appearance of ticks, tick labels, and gridlines.
ticklabel_format	Change the ScalarFormatter used by default for linear axes.
tight_layout	Automatically adjust subplot parameters to give specified padding.
title	Set a title for the axes.
twinx	Make and return a second axes that shares the x-axis.
twiny	Make and return a second axes that shares the y-axis.
violinplot	Make a violin plot.
vlines	Plot vertical lines.
xcorr	Plot the cross correlation between x and y.
xlabel	Set the label for the x-axis.
xlim	Get or set the x limits of the current axes.
xscale	Set the x-axis scale.
xticks	Get or set the current tick locations and labels of the x-axis.
ylabel	Set the label for the y-axis.
ylim	Get or set the y-limits of the current axes.
yscale	Set the y-axis scale.
yticks	Get or set the current tick locations and labels of the y-axis.

Seaborn **import seaborn as sns**

Chart	DESCRIPTION
catplot()	Figure-level interface for drawing categorical plots onto a FacetGrid.
stripplot()	Draw a scatterplot where one variable is categorical.
swarmplot()	Draw a categorical scatterplot with non-overlapping points.
boxplot()	Draw a box plot to show distributions with respect to categories.
violinplot()	Draw a combination of boxplot and kernel density estimate.
boxenplot()	Draw an enhanced box plot for larger datasets.
pointplot()	Show point estimates and confidence intervals using scatter plot glyphs.
barplot()	Show point estimates and confidence intervals as rectangular bars.
countplot()	Show the counts of observations in each categorical bin using bars.
relplot()	Figure-level interface for drawing relational plots onto a FacetGrid.
scatterplot()	Draw a scatter plot with possibility of several semantic groupings.
lineplot()	Draw a line plot with possibility of several semantic groupings.
jointplot()	Draw a plot of two variables with bivariate and univariate graphs.
pairplot()	Plot pairwise relationships in a dataset.
distplot()	Flexibly plot a univariate distribution of observations.
kdeplot()	Fit and plot a univariate or bivariate kernel density estimate.
rugplot()	Plot datapoints in an array as sticks on an axis.
lmplot()	Plot data and regression model fits across a FacetGrid.
regplot()	Plot data and a linear regression model fit.
residplot()	Plot the residuals of a linear regression.
heatmap()	Plot rectangular data as a color-encoded matrix.
clustermap()	Plot a matrix dataset as a hierarchically-clustered heatmap.

EXPLORATORY DATA ANALYSIS

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_	Rename the columns using rename function
df.rename(index={'old_name': 'new_na	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, ...], ascendin	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_	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_	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_obje	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