datafrik.co

Python Cheat Sheet

Cheat	structure	Syntax
Append	The append() method appends a DataFrame-like object at the end of the current DataFrame.	Syntax:df.append((table,variable,dataframe to be added))
Insert	Insert column into dataframe at specified location.	df.insert(location,column_name, value)
Drop	The drop() method removes the specified row or column	df.drop(row_index, column_name)
Read_csv()	This is used to import the file with which we want to work with.	import pandas as pd df=pd.read_csv('Filename')
Head()	Head returns the first rows, if no input is given it will always show above 5 rows. In contrast to see below rows, we can use df.tail().	df.head(number of rows to the displayed) df.tail(number of rows to the displayed)
Shape	Gives a total number of rows and then columns	syntax:df.shape()
size()	Returns the number of rows times number of columns in the data frame.	syntax:df.shape()
info()	Returns different information such as rows from RangeIndex, Data columns and then data type of each column. It also includes the information of non-null counts.	syntax:df.info()
isna()	This is used If one needs to get the total number of null values in a data.	Syntax:df.isna().sum()
describe()	This gives you count, mean, standard deviation, and also 5 number summary of the data.	syntax:df.describe()
value_counts()	We can get the value counts of each category using this function.	syntax:df.value_counts()
.columns	We use this to know the names of all the variables in a data frame.	Syntax:df.column
len()	This provides you with the length of the DataFrame. (total number of rows in the dataframe)	len(df)
iloc()	This function takes as a parameter the rows and column indices and gives you the subset of the dataframe accordingly.	df.iloc[10, 5] (this returns the value in the 10th row and column 5)
loc()	This function does almost the similar operation as .iloc() function. But here we can specify exactly which row index we want and also the name of the columns we want in our subset. ()	df.loc[[3, 10, 14, 23], ['nationality', 'weight_kg', "height_cm"]] ()
dtypes	This function helps us to know the data types of the variables before we dive into the analysis, visualization, or predictive modeling.	df."column_name".dtypes or df.dtypes
select_dtypes()	We use this to select the variables or columns of a certain data type using this function. For example, to select the columns with data types 'int64' only. Here is how to do that:	df.select_dtypes(include='int64') or use df.select_dtypes(exclude='int64') to exclude columns with 'int64' data type
insert()	As the name of the function suggests, it inserts a column in the specified position	df.insert(column_position, 'column_name', random_col hvyfvyugb65tfyu)
where()	This function helps you query a dataset based on a boolean condition.	df['column_name'].where(specified condition)
unique()	:It is used to find out the unique values of a categorical column.Useful where we have categorical variables.	df.column_name.unique()
nunique():	This returns all the unique values a variable contains.	syntax:df.nunique()
replace():	It replaces the values of a column.	df.replace('old_name', 'new_name')
rename()	It is used to rename the column/s.	df.rename(columns = {"old_name": "new_name", "old_name2": "new_name2"})
fillna()	This function replaces the null values with some other value of your choice Filling up the null values is part of your daily task if you are a data analyst or a data scientist.	df['column_name'].fillna(value to be filled, inplace=True)
groupby()	You can group the data as per a certain variable and find out useful information about those groups.	df.groupby("column1") ['column2'].sum()
count()	It provides you the number of data in the DataFrame in the specified direction. When the direction is 0, it provides the number of data in the columns:	df.count(0)