pandas groupby Cheat Sheet

by Daryabi via cheatography.com/181709/cs/37835/

Pandas objects

Series	a one-dimensional labeled array that can hold any data type
DataFrame	a two-dimensional labeled data structure with columns of potentially different types
Index	a sequence of axis labels that can be used to identify rows or columns in a DataFrame
MultiIndex	a hierarchical index object that allows for more than one index level in a DataFrame.
Timestamp	a specific moment in time, represented in Pandas as a datetime object.
Period	a specific interval of time, represented in Pandas as a period object.
DatetimeIndex	an index of datetime objects, used to index Pandas objects like Series and DataFrame.
Timedelta	a duration of time, represented in Pandas as a timedelta object.
Categorical	a data type used to represent categorical variables,
Sparse	a data structure used to represent sparse data efficiently,
Interval	a data type used to represent intervals
DatetimeTZ	a datetime object with a timezone.

Pandas functions called on pd

pd.read_csv()	used to read data from a CSV file and create a DataFrame.
pd.DataFrame()	a constructor function that is used to create a new DataFrame from data in memory.
pd.Series():	used to create a new Series object.
pd.concat()	used to concatenate two or more DataFrames or Series
pd.merge():	used to merge two DataFrames based on a common column.
pd.groupby()	used to group data in a DataFrame based on one or more columns.
pd.pivot_table():	used to create a pivot table from a DataFrame.
pd.to_datetime():	used to convert a column of strings to datetime objects.

Pandas functions called on pd (cont)

pd.to_numeric()	used to convert a column of strings to numeric objects.
pd.cut()	This function is used to bin data into discrete intervals.
pd.qcut():	This function is used to bin data into quantiles.
pd.date_range()	used to create a range of dates or DatetimeIndex.
pd.timedelta()	used to create a timedelta object representing a duration of time.

String vectorization

str.replace():	used to replace a pattern in a string with another pattern
str.extract()	used to extract a pattern from a string and return the first match.
str.split()	used to split a string into a list of strings based on a delimiter.
str.join()	used to concatenate a list of strings with a separator string
str.startswith()	used to check if a string starts with a particular substring
str.endswith()	used to check if a string ends with a particular substring
str.lower(), str.upper():	used to convert the case of a string to lowercase or uppercase, respectively
str.strip():	used to remove leading and trailing whitespace from a string
str.contains()	used to check if a pattern exists in a string

Date Vectorization

pd.to_datetime():	used to convert a column of strings or Unix timestamps to a datetime object.
dt.date	used to extract the date component of a datetime object or DatetimeIndex.

Date Vectorization (cont)

dt.time	extract the time component of a datetime object or DatetimeIndex.
dt.hour, dt.minute, dt.second	used to extract the hour, minute, and second components of a datetime object or DatetimeIndex.
dt.dayofweek:	used to extract the day of the week as an integer,
dt.day_name()	used to extract the name of the day of the week
dt.month_name()	used to extract the name of the month
dt.tz_localize() and dt.tz_convert():	used to set or convert the timezone of a datetime object or DatetimeIndex.
dt.is_month_start and dt.is_month_end	used to check if a datetime object or DatetimeIndex is at the start or end of a month, respectively.

pandas groupby

Consider it as DataFrame; you can use all DataFrame attributes and functions on the group object.

Pandas pivot_table

data	The DataFrame or Series to be used for the pivot table.
values	The column to aggregate. If not specified, all numerical columns will be used.
index	The column(s) to use as row labels.
columns	The column(s) to use as column labels.
aggfunc	The function to use for aggregating the values. Defaults to 'mean'.
margins	Add all row/columns (e.g. 'All') label and compute grand total for that row/column.
margins_name	The label for the row/column that contains the grand total. Defaults to 'All'.
dropna	Whether or not to exclude rows with missing values. Defaults to True.
sort	Whether or not to sort the rows by the values of the index.

Pandas multiIndex

```
# Create a MultiIndex from two separate indexes
index1 = pd.Ind ex( ['A', 'B', 'C'], name=' Ind -
ex1')
index2 = pd.Ind ex( ['X', 'Y', 'Z'], name=' Ind -
ex2')
multi_ index = pd.Mul tiI nde x.f rom _pr odu ct( -
[in dex1, index2], names= ['I ndex1', 'Index2'])
# Create a MultiIndex from a list of tuples
tuples = [('A', 'X'), ('A', 'Y'), ('B', 'X'),
('B', 'Y')]
multi_ index = pd.Mul tiI nde x.f rom _tu ple s(t -
uples, names= ['I ndex1', 'Index2'])
# Select data from a single level of the index
df.loc ['A'] # selects all rows where Index1 = 'A'
df.loc [('A', 'X')] # selects a single row where
Index1 = 'A' and Index2 = 'X'
# Select data from multiple levels of the index
# selects rows where Index1 is 'A' or 'B' and
Index2 is 'X', and returns the 'Column1' values
df.loc [(['A', 'B'], 'X'), 'Column1']
# Stack a DataFrame to move a level of the
MultiIndex to the columns
df.stack()
# Unstack a DataFrame to move a level of the
columns to the index
df.uns tack()
```