Cheatography

pandas and openpyxl Cheat Sheet

by Reszi via cheatography.com/199671/cs/42205/

Read/Write and Inspection

Read/Write	
`df = pd.rea d_e xce l(' fil e.x lsx')`	Read file (*CSV, Excel, JSON, HTML, SQL*)
`df.to_csv('file.csv')`	Write file
Inspection	
`df.head()/.tail()`	View first/last rows
`df.shape`	Gives dimensions
`df.dtypes`	Shows types in each column (*int, str etc*)
`df.info`	Lists: (*range of index, list of all columns, no. non null, data types, memory usage*)
`df.describe`	Lists: (*count, mean, std deviation, min, 25%, 50%, 75%, max*)

Filter

Functionality	
`df[df['column]` vs `df['column']`	1st filters dataframe, 2nd creates list of booleans
`df[df[x] > y & df[df[a] > v`	& combines filters
`df[df.apply(lamba row: row['c ol1'] * 2 > row['c ol2'], axis=0)`	Filter by function. Axis=0 for columns, 1 for rows.
`df.reset_index(drop=True)`	Reset indexs for filter
Rows and Columns	
`df[['column1', 'colum n2']]`	Filter by columns
`df.iloc[1:5]`	Filter by row (2nd - 6th)
`df[df['column'] > value`	Filters rows based on boolean
`df[df['column'].isin([value1, value2])`	Filter rows based on list of values
`df.query('age > @min_age')`	Filter rows based on query string, @ points to variable 'and' to combine
`df.loc[df['column'] > value, 'colum n_n ame']`	Filters rows and columns
Dropping and replacing	
`df.drop(columns=['column1', 'colum  n2'])' or '{{lang-python}}df.drop(['unnecessary_column'], axis=1, inplac e=True)`	Drop columns
`df.drop_duplicates(subset=['column'])`	Drop duplicate rows from specific columns
`df.dropna()` vs `.notna()`	Removes rows with missing values/non-missing values
`df.mask/.where(df['column'] > value)`	Replaces rows not meeting/meeting condition with NaN

Filter (cont)

String

`df[df['column'].str.contains('substring')`	Fitler where string column contains substring
`df[df['column'].str.match(r'abc$')]`	`^abc`: Start with 'abc'; `abc$`: End with 'abc'; `a|b`: 'a' or 'b'; `abc+`: 'abc' followed by 1+ 'c's; `abc*`: 'ab' followed by 0+ 'c's; `abc?`: 'ab' followed by 0 or 1 'c'; `[abc]`: Any one of 'a', 'b', or 'c'

Data Manipulation

Editing data

`df['new_column'] = new_values`	Assign/create new values for column
`df['column'] *= value`	Multiply (/=, -= , +=) each entry in a column by a value.
`df.fillna(value or method)`	Fill Nan values
`df['column'].replace(to_replace, value)`	Replace values
`df['column'].rolling(window=7).mean()`	Rolling aggregate
`df.update(other_df)`	Updates values from other df
`df['column'].astype(dtype)`	Converts data type
`np.where(df['column']> value, 'Value =True', 'Value =Fa lse')`	Create array based on new conditions
`df['column'].apply(lambda/function name))`	Apply functions to selection
`df.rename(columns={'old_name': 'new_n ame'})`	Rename columns

Combining data

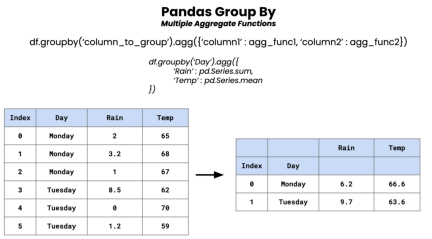
`pd.merge(df1,df2, on='common column', how='i nner')` or `pd.merge(df1, df2, left_i nde x=True, right_ ind ex= True, how='o - uter')`	Merges two dfs based on a common column. 'inner' requires both df to have all columns filled vs 'outer, 'left' vs 'right' to filter for columns of either df
`pd.concat([df1, df2])`	Concatenate (combines all values)
`df['column'].map(mapping_dict)`	Create a dictionary from a list to map keys and values to each other

Cheatography

pandas and openpyxl Cheat Sheet

by Reszi via cheatography.com/199671/cs/42205/

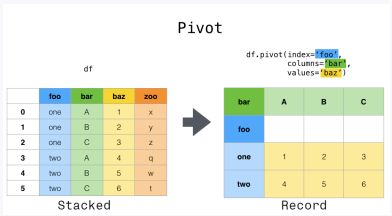
Group by




```
grouped = df.gro upb y([ 'St ore', 'Produ ct' ]).a gg (To tal _Sa les =(' Sal es' ,'sum) Averag e_C ost =(
' Cost', 'mean'), Sales_ Cou nt= ('S ales', 'count'))
filtered_grouped = groupe d[g rou ped ['T ota l_S ales'] > 500]
```

Aggregations: `sum`; `mean`; `median`; `min`; `max`; `count`; `size`; `std` (standard deviation); `var` (variance); `first`; `last`; `prod` or `p roduct`; `nunique` (number of unique values)

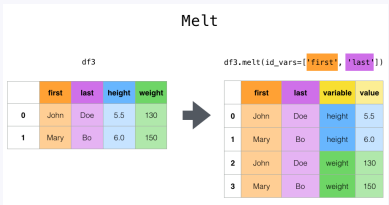
Pivot




```
pivoted_df = df.piv ot( ind ex= 'Date', column s=' Var iable', values ='V alue')
pivot_table_df = df.piv ot_ tab le( ind ex= 'Date', column s=' Var iable', values ='V alue', aggfun c=' su
m')
```

values (optional): Columns whose data will be aggregated.
index: Columns used as index.
columns (optional): Columns to pivot into new DataFrame's columns.
aggfunc: Aggregation function for values.
pivot_table: aggregates duplicates

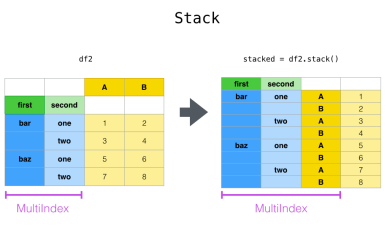
Melt




```
melted_df = pd.mel t(df, id_var s=[ 'id _co lumn'], var_na me= 'va ria ble _name', value_ nam e=' val ue_
name')
```

id_vars (optional): Columns to keep unchanged. Otherwise default melted
value_vars (optional): Columns to melt.
var_name (optional): Melted column name. Default is 'variable'.
value_name (optional): Melted values column name. Default is 'value'.
col_level (optional): Multi-index level

Stack/unstack




```
stacked_df = df.sta ck( lev el=-1, dropna =True)
```

Level: The level(s) of the column labels you want to stack. The default is the last level.

Dropna: Whether to drop rows in the resulting DataFrame with missing values. Default is True.

By **Reszi**
cheatography.com/reszi/

Published 28th January, 2024.
Last updated 29th January, 2024.

Cheatography

pandas and openpyxl Cheat Sheet

by Reszi via cheatography.com/199671/cs/42205/

Time Operations

`pd.to_datetime(df['column'])`	Converts a column to datetime format
`pd.Timestamp.now()`	Returns the current date and time
`df['datetime_column'].dt.date/.time`	Extracts the date/time from a datetime column
`datetime.datetime.strptime(date_string, format)`	Useful for strings with words and dates, or including hours/minutes, or multiple dates types in one column (requires try except function)

Open pyxl

`openpyxl.load_workbook(filename)` `workbo ok.s av e(f ile name)`	Read/Write
`workbook[sheetname] or workbo ok.a ctive`	Sheet selection
Cell Formats	
`cell.font = Font(s ize=12)`	Font
`cell.number_format = '0.00%'`	Cell formats

`Alignment(horizontal="center");`
`Border (le ft= Sid e(b ord er_ sty le= " thi n", color= " 000 000 ");`
`Patter nFi ll( " sol id", fgColo r="D DDD DD")`
	More cell options

`Comment('Text', 'Author')`	Cell Comments, can change comment.widt-h/height too.
`sheet.unmerge/merge_cells('A1:D1')`	Merge/Unmerge
`cell.hyperlink = 'http: //w ww.e xa mpl e.com'`	Hyperlinks
`ColorScaleRule(start_type="min", start_ col or= " FFF FFF ", end_ty pe= " max ", end_co lor ="FF 000 0")` `ws.con dit ion al_ for mat tin g.a dd( " A1: A9", rule)`	Conditional Formatting
`DataValidation(type="list", formul a1= '"It em1 ,It em2 ,It em3 "', showDr opD own =True)` `ws.add _da ta_ val ida tio n(dv)`	Data validation
Charts	

`chart = BarCha rt(),`
`chart.s tyle = 13,`
`sheet.a dd _ch art (chart)`
	Create, style, then add chart to sheet

`Image('path/to/image'), sheet.a dd _im age (img, 'A1')`	Add images, better for mathplot or other libraries
`cell.value = '=SUM( A1: A10)'`	Write formulas