Cheatography

Pandas for Data Analysis Cheat Sheet

by email2automate via cheatography.com/184557/cs/38509/

Shared Attributes

```
.head(n)
.tail(n)
.index
.values
.shape
.axes
.info()
```

Common attributes shared between pd.Series and pd.DataFrame

Null Unique Values

Drop Null Values

`df.dro pna (how= any/all, inplace = T/F)`

`df.dro pna (subset = ['col_ name1', 'col_n ame2'], how = any/all, inplace = T/F)` - from specific columns

Fill Null Values

`df.fil lna (value, inplace = T/F)` - acts on entire data frame.

`df['co l_n ame '].f il lna (value = 'fill_ value', inplace = T/F)` - act on selected columns

Unique Values

`df['co l].u ni que()` - Returns unique values as a list(including null values)

`df.nun ique()` - No of unique Values.(by default doesn't count null)

`df.uni que (dropna = False)`

`df['co l'].nu nique()`

`df['co l'].un iqu e(d ropna = True/F alse)`

Panda Series

`pd.Ser ies.ha snans`

`pds.nu nique()` - No of unique Values.

`pdf.un iqu e(d ropna = True/F alse)` - Returns unique values in a list.

Dealing Columns

`df.columns` -retuns column names as a list

`df['ne w_c ol' ]=l ist /series`

`df.ins ert[loc = n, column= col_name, value= New_va lue /array]` -inserts column at desired position / location

Sort, Rank, Count

Sort

`df.sor t_i nde x(a sce nding = True/F alse)` - Sort Index

`df.sor t_v alu es( col _name, na_pos ition = " Fir st/ Las t", ascending = True/F alse)` - Sort based on values

`df.sor t_v alu es( ['c ol_ name, 'col2'], ascending = [True, False])` - Sort by Multiple Columns

Rank

`df['co l'].ra nk( asc ending = True/F alse)` - Ranks are assigned based on sorted values

counts

`df.val ue_ cou nts()` - counts exact rows

`df.val ue_ cou nts (no rmalise = True)`

Sort, Rank, Count (cont)

`pds.va lue _co unts()` - pd series

Data Type Conversions and Optimization

```
df['col'] = pd.to_ dat eti me( df[ 'col'])
```

or we can parse dates in import itself.

```
pd.rea d_c sv(....., parse_ dates = ['col1 ',' col2'])
df['col'] = df['co l'].as typ e(d type)
dtype = bool, category, int
```

Filtering Data

Multiple Conditional Filtering

`mask1 = df[col1'] == 'value'` - Returns True/False boolean series

```
mask2 = df[col2'] <= 'value'
mask3 = df[col3'] >= 'value'
df[(mask1 & mask2) | mask3]
```

Inclusion Check

`mask = df['co l'].is in( ['v al1 ',' val 2', 'va l3'])` - Check for inclusion using isin() method.

`mask1 = df[col1'] == 'val1'` - isin method is equal to three conditional checks.

```
mask1 = df[col1'] == 'val2'
mask1 = df[col1'] == 'val3'
```

For NULL values

`mask = df['co l'].is null()` - Returns True/False boolean series

`mask = df['co l'].no tnull()` - Returns True/False boolean series

Inclusion Check within a range

`mask = df['co l'].be twe en( val1, val2)` -Returns True/False boolean series. True for values within range.

Duplicate Values

`mask = df['co l'].du pli cat ed(keep = " Fir st/ Las t/F als e")` - Returns boolean series, True for Duplicates

`df.dro p_d upl ica tes()` -Deletes duplicate from df. if applied on df complete row should be identical.

`df.dro p_d upl ica tes (subset = ['col1 ',' col2'])` - Drops if the combination of col1 and col2 are identical.

Data Extraction

SET INDEX

RESET INDEX

LOC ACCESSOR

ILOC ACCESSOR

CODE