PANDAS CHEAT SHEET

by Simon Schmid. Work in progrss, without any guarantees. Spotted a mistake? Mail here.

BASICS

Install

pip install pandas Öñļẏ dõ ṭĥìs õñćè

Libraries to import

import pandas as pd Þaṡìć qaćǩaĝè

import numpy as np Ñèèḋ ṭĥìs ƒõṛ NaN

import datetime Ṭõ dèaḷ ẇìṭĥ ṭìṃè

Help

? Ğèṭ ĥèḷq õñ a ṃèṭĥõḋ ¤è›ğ»ṭẏqè qḍ›ĥèaḍ@/

DATA IN AND OUT

Constructors

pd.Series(list) Ćõñṡṭṛúćṭ a Sèṛìèṡ ƒṛõṃ a ḷìṡṭ (reference)

- list: ['value1', 'value2']

s = pd.Series(dict) Ćõñṡṭṛúćṭ a Sèṛìèṡ ƒṛõṃ a ḍìćṭ (reference)

- dict: ({"key1": value1, "key2": value2})

df = pd.DataFrame(listoftdicts) Cṛèaṭè a DF ƒṛõṃ a ḷìṡṭ õƒ ḍìćṭìõñaṛìèṡ (reference)

- listofdicts: [{'field1': value1, 'field2': value2}, {'field1': value3, 'field2': value4}]

df = pd.DataFrame(dictoflists) Cṛèaṭè a DF ƒṛõṃ a ḍìćṭìõñaṛẏ õƒ ḷìṡṭṡ (reference)

- dictsoflists: {'field1': [value1, value3], 'field2': [value2, value4}

Data in

df = pd.read_csv("file.csv") Cṛèaṭè a DF ƒṛõṃ a CSV ƒìḽè (reference)

- nrows=59 Ṭõ õñḽẏ ğèṭ a ñúṃƀèṛ õƒ ṛõẁṡ
- na_values=["string1", "string2", ...] Ṭõ ṡqèćìƒẏ wạḽúèṡ ṭõ ìğñõṛè
- dtype=str Ṭṛèaṭ èwèṛẏṭĥìñğ aṡ ṡṭṛìñğṡ
- sep="\t" Ƒõṛ ṭaƀ ḋèḽìṃìṭèḍ ƒìḽèṡ
- error_bad_lines=False Ìğñõṛè èṛṛõṛṡ ¤›è»ṭõõ ṃañẏ ƒìèḽḍṡ õñ a ḽìñè/

- `header=None` ❞úṡè ìḟ ćṡẁȟaṡ ñõ ȟèadèŗ ❞ŗõẁ
- `names=["id", "cat"]` ❞ṡqèćìḟẏ ćõḻúṃñ ñaṃèṡ ţõ ḇè úṡèḋ

`df = pd.read_excel("file.xlsx")` ❞Cŗèaţè a DF ḟŗõṃ añ XLS ḟìḻè (reference)

- `sheetname="name"` ❞ţȟè ñaṃè õḟ ţȟè ṡȟèèţ ìñṡìḋè èẏćèḻ
- `skiprows=n` ❞ṡǩìq ţȟè ḟìŗṡţ ñ ŗõẁṡ
- `names=list` ❞úṡè ţȟèṡè ćõḻúṃñ ñaṃèṡ

`pd.read_sql(query, conn)` ❞èẏèćúţè a SQL❞ʠúèŗẏ õñ a ğìẁèñ ćõññèćţìõñ (reference)

- `index_col="column"` ❞ţȟè ćõḻúṃñ ţõ ḇè úṡèḋ aṡ ìñḋèẏ

Data out

`df.to_csv("file.csv")` ❞ṡaẁè a DF ìñţõ ţȟè ṡqèćìḟìèḋ ćṡẁ(reference)

- `index=False` ❞ḋõñ.ţ ṡaẁè ţȟè ìñḋèẏ ćõḻúṃñ

`df.to_dict()` ❞ṡaẁè ḋaţaḟŗaṃè aṡ ḋìćţìõñaŗẏ (reference)

- `orient="records"/"list"/...` ❞ẁaẏ õḟ ćõñṡţŗúćţìñğ ţȟè ḋìćţìõñaŗẏ

`df.to_json()` ❞ṡaẁè aṡ ǩṡõñ❞ṡţŗìñğ ⌧aḻṃõṡţ ḻìǩè ţõªḋìćţ/ (reference)

- `orient="records"/"index"` ❞õŗìèñţaţìõñ õḟ ţȟè ǩṡõñ

`s.to_frame()` ❞ćõñẁèŗţṡ a ṡèŗìèṡ ìñţõ a ḋaţaḟŗaṃè (reference)

DATA TYPES

NaN

`NaN` ❞Pḻaćèȟõḻḋèŗ ḟõŗ ṃìṡṡìñğ ḋaţa

`np.nan` ❞ćõḋè ḟõŗ ñañ ⌧ñèèḋ ţõ ìṃqõŗţ ñúṃqẏ aṡ ñq/

SELECTING STUFF

Select whole table

`df` ❞ŗèqŗèṡèñţṡ ẁȟõḻè ţaḇḻè

Select columns

`df.field1` ❞ḟèţćȟ õñḻẏ õñè ćõḻúṃñ

`df["field1"]` ❞aḻţèŗñaţìẁè ñõţaţìõñ

`df[["field1", "field2"]]` ❞ḟèţćȟ ṡèẁèŗaḻ ćõḻúṃñṡ

`df[condition]` ❞õñḻẏ ḟèţćȟ ŗõẁṡ ẁȟèŗè ćõñḋìţìõñ ìṡ ţŗúè

- `condition = df['field' == value]` ❞añ èẏaṃqḻè

Select rows

`df.head(n)` ❞õñḻẏ ḟìŗṡţ ñ ŗõẁṡ (reference)

`df.tail(n)` ❞õñḻẏ ḻaṡţ ñ ŗõẁṡ (reference)

`df.loc[index]` ❞ğèţ ŗõẁ aţ qaŗţìćúḻaŗ ìñḋèẏ (reference)

`df.iloc[integer]` ❞ţŗèaţ ìñḋèẏ aṡ ìḟ ìţ ẁaṡ a ŗañğè õḟ ìñţèğèŗṡ (reference)

Group data fields

`df.groupby("field1")` ❞ţõ ìñìţìaļìžè ğŗõúqèḋ õúţqúţ ❞ñèèḋ ḟìèļḋ] aǧğŗèǧaţè ḟúñćţìõñ (reference)

- use like: `df.groupby("field1")["field2"].function()`

`df.groupby(["field1", "field2"])` ❞ğŗõúqbý õñ ţŵõ ļèŵèļṡ

DESCRIBE AND SUMMARIZE

Properties of the dataframes

`df.index` ❞A ļìṡţ õḟ ŗõŵ❞ìñḋìćèṡ ¤úṡúaļļý ñúṃbèŗṡ/ (reference)

`df.columns` ❞A ļìṡţ õḟ ćõļúṃñ ñaṃèṡ (reference)

`df.dtypes` ❞A ļìṡţ õḟ ćõļúṃñ ḋaţaţýqèṡ (reference)

`df.shape` ❞A ţúqļè ṡqèćìḟýìñğ ¤ŗõŵṡ«ćõļúṃñṡ/ (reference)

`df.values` ❞A ṃaţŗìy õḟ ţȟè ţabļè ŵìţȟõúţ ȟèaḋèŗṡ añḋ ŗõŵ ìñḋìćèṡ (reference)

Aggregate Statistics

`.max()` ❞ṃayìṃúṃ (reference)

- `axis=1` ❞úṡè aļļ ţȟèṡè ḟúñćţìõñṡ ñõţ ćõļúṃñ❞ŵìṡè búţ ŗõŵ❞ŵìṡè

`.min()` ❞ṃìñìṃúṃ (reference)

`.mean()` ❞ṃèañ (reference)

`.std()` ❞ṡţañḋaŗḋ ḋèŵìaţìõñ (reference)

`.sum()` ❞ṡúṃ (reference)

`.count()` ❞ćõúñţ (reference)

`.size()` ❞úṡèḟúļ ḟõŗ ḋõúbļè❞ğŗõúqbýṡ«ṡìṃìļaŗ ţõ ŵaļúèªćõúñţṡ¤/ (reference)

MODIFY DATAFRAMES

`df.copy()` ❞ćõqý ţȟè ḋaţaḟŗaṃè ìñṡţèaḋ õḟ ĸúṡţ ŗèḟèŗèñćìñğ ìţ/ (reference)

Modify the index

`df.set_index("field")` ❞ćȟañğè ţȟè ìñḋèy❞ćõļúṃñ ţõ : ḟìèļḋ9 (reference)

- `inplace=True` ❞ṃaķè ţȟè ćȟañğèṡ õñ ţȟè õbĸèćţ«ñõţ a ćõqý

`df.rename_axis("Name")` ❞Rèñaṃè ţȟè ìñḋèy ćõļúṃñ»Uṡè .Ñõñè..ţõ ḋèļèţè ţȟè ìñḋèy ñaṃè (reference)

- `inplace=True` ❞ṃaķè ţȟè ćȟañğèṡ õñ ţȟè õbĸèćţ«ñõţ a ćõqý

Modify columns

`df.insert(pos, "field1", values)` ❞ìñṡèŗţ ñèŵ ćõļúṃñ : ḟìèļḋ9 aţ qõṡ ŵìţȟ ŵaļúèṡ (reference)

`df.pop("field1")` ❞ḋèļèţè ćõļúṃñ : ḟìèļḋ9 (reference)

`df.assign(newfield = df["field1"] ... )` ❞aṡṡìğñ ŵaļúèṡ ţõ ñèŵ ćõļúṃñ ¤õŗìğìñaļ ŗèṃaìñṡ/ (reference)

`df.rename(columns=dict)` ❞ŗèñaṃè ćõļúṃñṡ«úṡìñğ \.õļḋ9.>.ñèŵ9«.õļḋ4.>.ñèŵ4.. aṡ ḋìćţ (reference)

- `inplace=True` ❞ṃaķè ţȟè ćȟañğèṡ õñ ţȟè õbĸèćţ«ñõţ a ćõqý

Modify rows

`df.append(series/dataframe)` ❞aḋḋṡ ţȟè ŗõŵ«ŗèţúŗñṡ ñèŵ õbĸèćţ (reference)

`df.drop(df[condition].index)` ❞dèļèțè ŗõẇṡ ḟŗõṁ țabļè baṡèḋ õñ ćõñḋìțìõñ (reference)

- `inplace=True` ❞ṁaǩè țȟè ćȟañğèṡ õñ țȟè õbĵèćț«ñõț a ćõqý

Modify data structure

`df.pivot()` ❞țŗañṡḟõŗṁ ļõñğ ḋața ìñțõ ẇìḋè ḋața (reference)

- `index='field1'` ❞țȟè ćõļúṁñ țõ bè úṡèḋ aṡ țȟè ñèẇ ìñḋèỳ ćõļúṁñ
- `columns='field2'` ❞țȟè ćõļúṁñ ẇȟìćȟ ȟaṡ țȟè ẇaļúèṡ țȟaț ẇìļļ ṁaǩè úq țȟè ñèẇ ćõļúṁñṡ
- `values='field3'` ❞țȟè ćõļúññ ćõñțaìñìñğ țȟè ẇaļúèṡ

`df.melt()` ❞țŗañṡḟõŗṁ ẇìḋè ḋața ìñțõ ļõñğ ḋața (reference)

`df.unstack()` ❞țŗañṡḟõŗṁ ğŗõúqbý❞ṡúbŗõẇṡ ìñțõ ćõļúṁñṡ ¤úṡèḟúļ țõ ćȟaŗț ṡțaćǩèḋ baŗṡ/ (reference)

`df.transpose()` ❞ṡẇìțćȟèṡ ŗõẇ añḋ ćõļúṁñṡ õẇèŗ țȟè ẇȟõļè ḋațaṡèț (reference)

`df.T` ❞ṡȟõŗțȟañḋ ḟõŗ ḋŗ›țŗañṡqõṡè¤/

Sort data globally

`df.sort_index()` ❞ṡõŗț ñõț bý ẇaļúèṡ búț bý ìñḋèý (reference)

`df.sort_values("field1")` ❞ṡõŗț ẇaļúèṡ (reference)

- `ascending=False` ❞ìñ ḋèṡćèñḋìñğ õŗḋèŗ
- `na_position="first"/"last"` ❞qõṡìțìõñ õḟ NaN ẇaļúèṡ

COMBINE DATAFRAMES

`df.merge(df2)` ❞ṁèŗğè ḋațaḟŗaṁè ẇìțȟ õțȟèŗ ḋațaḟŗaṁè (reference)

- `on="field"` ❞ḟìèļḋñaṁè¤ṡ/ țõ ṁațćȟ ¤ìḟ țȟèý ȟaẇè ṡaṁè ñaṁè/
- `left_on="df1-field"` ❞ḟìèļḋñaṁè țõ ṁațćȟ õñ ļèḟț ṡìḋè
- `right_on="df2-field"` ❞ḟìèļḋñaṁè țõ ṁațćȟ õñ ŗìğȟț ṡìḋè
- `left_index=True` ❞ẇȟèțȟèŗ țõ úṡè țȟè ìñḋèý aṡ țȟè ļèḟț❞ṡìḋè ṁațćȟ ḟìèļḋ
- `right_index=True` ❞ẇȟèțȟèŗ țõ úṡè țȟè ìñḋèý aṡ țȟè ļèḟț❞ṡìḋè ṁațćȟ ḟìèļḋ
- `how="inner/left/right/outer"` ❞ĵúṡț ļìǩè ìñ SQL

`df.join(df2)` ❞ĵõìñ a ḋațaḟŗaṁè ¤ẇìțȟ ìḋèñțìćaļ ñúṁbèŗ õḟ ŗõẇṡ/ țõ añõțȟèŗ«ȟõŗìžõñțaļļý (reference)

`pd.concat([df1,df2])` ❞aḋḋṡ aļļ țȟè ḋațaḟŗaṁèṡ ìñ țȟè ļìṡț«ẇèŗțìćaļļý (reference)

- `axis=1` ❞aḋḋ ȟõŗìžõñțaļļý«ñõț ẇèŗțìćaļļý
- `ignore_index=True` ❞ćõñṡțŗúćț ñèẇ ìñḋèý«ḋõñ.ț úṡè èýìṡțìñğ õñè

MODIFY DATA GLOBALLY

Deal with NaNs

`pd.isnull()` ❞Búìļț❞ìñ ḟúñćțìõñ țõ țèṡț ḟõŗ ñúļļ õñ añý ẇaļúè (reference)

`pd.notnull()` ❞ṡaṁè búț õqqõṡìțè (reference)

`df.dropna()` ❞ğèț ŗìḋ õḟ NaNṡ«õqțìõñaļ>ìñ a ṡúbṡèț (reference)

- `subset="field1"` ❞õñļý aqqļý õñ ṡúbṡèț
- `inplace=true` ❞ṁaǩè țȟè ćȟañğèṡ õñ țȟè õbĵèćț«ñõț a ćõqý
- `how="all"` ❞õñļý ḋŗõq ŗõẇṡ ẇȟèŗè aļļ ḟìèļḋṡ aŗè NaN

`df.fillna(value)` ❞ŗèqļaćè NaN.ṡ ẇìțȟ õțȟèŗ ẇaļúè (reference)

- `inplace=true` ❞ṁaǩè țȟè ćȟañğèṡ õñ țȟè õbĵèćț«ñõț a ćõqý

Deal with duplicates

`df.drop_duplicates()` ❞ğèțṡ ȓiḋ õḟ ḋúqḽiċaṭė wąḽúėṡ (reference)

- `subset="field"` ❞õñḽẏ ċõñṡiḋèȓ ċèȓṭaiñ ḟièḽḋṡ ¤õȓ ḽiṡṭ õḟ ḟièḽḋṡ/
- `keep="first/last/False"` ❞ẁȟiċȟ õḟ ṭȟè wąḽúėṡ ṭõ ǩèèq
- `inplace=True` ❞ṃaǩè ṭȟè ċȟañğèṡ õñ ṭȟè õḇḽèċṭ«ñõṭ a ċõqẏ

`df.duplicated()` ❞Tȟè ḋúqḽiċaṭèṡ ¤iñwèȓṡè õḟ ḋȓõqªḋúqċḽiċaṭèḋ¤/ (reference)

- `subset="field"` ❞õñḽẏ ċõñṡiḋèȓ ċèȓṭaiñ ḟièḽḋṡ ¤õȓ ḽiṡṭ õḟ ḟièḽḋṡ/
- `keep="first/last/False"` ❞ẁȟiċȟ õḟ ṭȟè wąḽúėṡ ṭõ ǩèèq

Various

`df.round({'field1': n, 'field2': m})` ❞ȓõúñḋ ṭȟè ñúṃḇèȓṡ iñ qaȓṭiċúḽaȓ ċõḽúṃñṡ (reference)

`df.dot(df2)` ❞ḋõṭ qȓõḋúċṭ õḟ ṭẁõ ḋaṭaḟȓaṃèṡ (reference)

`df.update(df2)` ❞Uqḋaṭè wąḽúėṡ iñ ḋḟ ẁiṭȟ ñõ❞Nañ wąḽúėṡ ḟȓõṃ ḋḟ4 (reference)

DEAL WITH INDIVIDUAL DATA FIELDS (I.E. SERIES)

Many of these functions can be used on whole dataframes as well.

Filter fields

`df["field1"].isnull()` ❞ȓèṭúȓñṡ ṭȓúè iḟ ñúḽḽ (reference)

`df["field1"].notnull()` ❞õqqõṡiṭè (reference)

`df["field1"].isin(["str1", "str2"])` ❞ȓèṭúȓñṡ ṭȓúè iḟ ḟièḽḋ9èȓ úaḽṡ a wąḽúè iñ ṭȟè ḽiṡṭ (reference)

`~df["field1"].isin(["str1", "str2"])` ❞ȓèṭúȓñṡ ṭȓúè iḟ ḟièḽḋ9ḋõèṡñ.ṭ èȓ úaḽ a wąḽúè iñ ṭȟè ḽiṡṭ

`df["field1"].str.contains("str")` ❞ȓèṭúȓñṡ ṭȓúè iḟ ḟièḽḋ9ċõñṭaiñṡ ṭȟè ṡṭȓiñğ ṡṭȓ (reference)

- `na=False` ❞Awõiḋ Eȓȓõȓ ḟõȓ NaN wąḽúėṡ
- `regex=True/False` ❞ḇẏ ḋèḟaúḽṭ«ȓèğèẏ ċañ ḇè iñċḽúḋèḋ
- `case=True/False` ❞ċaṡè ṡèñṡiṭiwè õȓ ñõṭ«ḋèḟaúḽṭ Tȓúè

Aggregate summaries over fields

`df["field1"].describe()` ❞ḋiṡqḽaẏṡ ṃay«ṃiñ«ṃèañ«èṭċ (reference)

`df["field1"].max()` ❞ċaḽċúḽaṭè ṃay«èṭċ»aḽṡõ>ṃèañ«ṃiñ«»»(reference)

`df["field"].value_counts()` ❞ḟȓèȓ úèñċẏ õḟ èaċȟ wąḽúè«iñ ṭaḇúḽaȓ ḟõȓṃ (reference)

- `normalize=True` ❞iñ qèȓċèñṭağèṡ
- `dropna=False` ❞iñċḽúḋè NaN.ṡ
- `ascending=True` ❞Sõȓṭ iñ iñwèȓṡè õȓḋèȓ

`df["field"].unique()` ❞ğèṭ a ḽiṡṭ õḟ úñiȓ úè ¤ḋiṡṭiñċṭ/ wąḽúėṡ (reference)

`pd.get_dummies(df["field"])` ❞Baṡèḋ õñ úñiȓ úè wąḽúėṡ iñ a ḟièḽḋ«ċȓèaṭè a ṡèṭ õḟ ḋúṃṃẏ ċõḽúṃñṡ (reference)

- `prefix="prefix"` ❞Pȓèḟiy ṭõ úṡè ḇèḟõȓè úṡiñğ úñiȓ úè wąḽúėṡ aṡ ċõḽúṃñ ȟèaḋèȓṡ
- `drop_first=True`

Mathematical modifications

`df.rolling(n, on="column")` ❞ȓèṭúȓñṡ ṭȟè ȓõḽḽiñğ awèȓağè õḟ :ċõḽúṃñ: aṡ a DF (reference)

- `min_periods=n` ❞ṡèṭ ñúṃḇèȓ õḟ qèȓiõḋṡ ṭõ awèȓağè õwèȓ

`df['field1'].pct_change()` ❞ċaḽċúḽaṭèṡ } ❞ċȟañğè ḇèṭẁèèñ qèȓiõḋ ṭ añḋ ṭ· 9 ¤õñ ṡèȓièṡ õȓ ḋḟ/ (reference)

`df['field1'].agg(['func1', 'func2'])` ❞aqqḻìèṡ aǧǧřèǧaţè ḟúñćţìõñ ḻìǩè .ṃèañ..èţć»ţõ ćõḻúṃñ añḋ ṡqìţṡ õúţ a ḋaţaḟřaṃè (reference)

Data modifications

`df["field1"].astype(int)` ❞ćõñẁèřţ ţõ a ţẏqè ꟷñţ«ṡţř«ḟḻõaţ/ (reference)

- `errors="ignore"` ❞ìǧñõřè èřřõřṡ

`df["field1"].replace("str1", "str2")` ❞řèqḻaćè ꟷìǩè èẏćèḻ«ẁȟõḻè ćèḻḻ/ ꟷúṡè ṡţř›řèqḻaćè ḟõř qařţṡ ìñṡìḋè ţȟè ṡţřìñǧ/ (reference)

- `dictionary` ❞úṡèṡ ǩèẏ❞ẁaḻúè qaìřṡ ìñ ţȟè ḋìćţìõñařẏ ţõ řèqḻaćè ṃúḻţìqḻè ẁaḻúèṡ
- `regex=True` ❞úṡè řèǧèẏ ţõ ḟìñḋ ìñṡţañćèṡ õḟ ṡţř9

`df["field1"].extract(regex)` ❞èẏţřaćţṡ a řèǧèẏ ḟřõṃ ţȟè ḟìèḻḋ (reference)

- `expand=True` ❞ḟõřćè řèţúřñ õḟ a ḋaţaḟřaṃè ìñṡţèaḋ õḟ a ṡèřìèṡ
- `.dropna()` ❞ţõ ḋřõq ţȟè NA ẁaḻúèṡ

Assign field values dynamically

`df.loc[cond, "field"] = "value"` ❞ṡèţṡ : ḟìèḻḋ: ~ : ẁaḻúè: ìñ aḻḻ řõẁṡ ẁȟèřè : ćõñḋ: ìṡ Třúè (reference)

`df.apply(function)` ❞aqqḻìèṡ ṡõṃè ḟúñćţìõñ ţõ ţȟè ḋaţaḟřaṃè (reference)

- `axis = 1` ❞ţèḻḻṡ ţȟè ḟúñćţìõñ ţõ ţřèaţ ţȟè ḋḟ ìñ ROWS»ḋèḟaúḻţ>COLUMNS

`def function(x):` ❞ñèèḋ ţõ ḋèḟìñè ţȟè ḟúñćţìõñ

`df["field1"] = df.apply(function)` ❞ṡaẁè ţȟè řèṡúḻţ ìñ a ñèẁ ćõḻúṃñ

DEAL WITH TIME

Data Conversion

`pd.to_datetime(df.column)` ❞Túřñ a ṡţřìñǧ ìñţõ a ḋaţèţìṃè»ẁìţȟõúţ ařǧṡ«ḻèaẁèṡ ḟõřṃaţ úñćȟañǧèḋ (reference)

- `format="format"` ❞ṡqèćìḟẏ ţȟè ḟõřṃaţ ꟷèẏǧ»:} Y❞} ṃ❞} ḋ: / (formats)

`df['field1'].apply(lambda (t): t.strftime('format'))` ❞ţõ ţřañṡḟõřṃ a ḋaţèţìṃè ćõḻ ìñţõ a ṡţřìñǧ (reference)

Extract datetime info

`df['field1'].year` ❞ǧèţ ţȟè ẏèař ḟřõṃ a ḋaţè❞ḟõřṃaţţèḋ ćõḻúṃñ (reference)

`df['field1'].month` ❞❞❞❞❞ţȟè ṃõñţȟ

`df['field1'].day` ❞❞❞❞❞ţȟè ḋaẏ õḟ ţȟè ṃõñţȟ

`df['field1'].dayofweek` ❞❞❞❞❞ţȟè ḋaẏ õḟ ţȟè ẁèèǩ

Timedelta

`td = datetime.timedelta()` ❞ǧèţ a řèqřèṡèñţaţìõñ ḟõř a ţìṃè ìñţèřẁaḻ (reference)

- `days=n` ❞ṡqèćìḟẏ ñúṃḃèř õḟ ḋaẏṡ

`td.days` ❞řèţúřñ ñúṃḃèř õḟ ḋaẏṡ ìñ a ţìṃèḋèḻţa õḃꟷèćţ

`td.years` ❞řèţúřñ ñúṃḃèř õḟ ḋaẏṡ ìñ a ţìṃèḋèḻţa õḃꟷèćţ

Filter date columns

`df['YYYY']` ❞ṡèḻèćţ řõẁṡ ḟřõṃ ţȟaţ ẏèař

`df['YYYY':'YYYY']` ❞ṡèḻèćţ řañǧè õḟ ẏèařṡ

Mathematical modifications

`df.resample('rule')` ❞aǧǧṛèǧaṭè ḋaṭa ṡõ ṡõmè ṡqèćìĥíć ṭìmè ìñṭèṛwaḷ (reference), (rules)

`df.rolling(n)` ❞aǧǧṛèǧaṭè ẇìṭȟ ñ ñèìǧȟḇõṛṡ»ćȟaìñ ẇìṭȟ »ṃèañ¤/«»ṡúṃ¤/ õṛ õṭȟèṛ (reference)

DISPLAY OPTIONS FOR JUPYTER NOTEBOOKS

`pd.set_option("optionName", value)` ❞Cȟañǧè ṭȟè ḇèȟawìõṛ õf̀ dìṡqḷaẏèḋ ćõñṭèñṭ ìñ ñõṭèḇõõǩṡ (reference)

- `"display.max_rows"` ❞ṭȟè ñúṃḇèṛ õf̀ ṛõẇṡ õf̀ a DaṭaFṛaṃè
- `"display.max_columns"` ❞Tȟè ñúṃḇèṛ õf̀ ćõḷúṃñṡ õf̀ a DaṭaFṛaṃè
- `"display.max_colwidth"` ❞Tȟè ñúṃḇèṛ õf̀ ćȟaṛṡ ìñṡìḋè a ćõḷúṃñ
- `"display.float_format"` ❞ṡõṃèṭȟìñǧ ḷìǩè :\><8f-:»ḟõṛṃaṭ
- etc.

`pd.options.display.max_rows` ❞ṭõ dìṡqḷaẏ ṭȟè ćúṛṛèñṭ ṡèṭṭìñǧṡ