stats_can Documentation

Ian Preston

Mar 29, 2022

CONTENTS

This library implements most of the functions defined by the Statistics Canada Web Data Service. It also has a number of helper functions that make it easy to read Statistics Canada tables or vectors into pandas dataframes.

CHAPTER

ONE

INSTALLATION

The package can either be installed with pip or conda:

```
conda install -c conda-forge stats_can
```

Or:

```
pip install stats-can
```

The code is also available on github,

CHAPTER

TWO

QUICKSTART

After installing the stats_can package, all of the core functionality is available by instantiating a StatsCan object:

```
from stats_can import StatsCan
sc = StatsCan()
```

Without any arguments the StatsCan object will look for a file in the current working directory named "stats_can.h5". If it doesn't exist it will create one when it is first asked to load a table. You can also pass in a path in order to specify the location of the file. This is useful if you or a team want persistent access to certain tables.

For example:

```
sc = StatsCan(data_folder="~/stats_can_data")
```

The most common use case for stats_can is simply to read in a table from Statistics Canada to a Pandas DataFrame. For example, table 271-000-22-01 is "Personnel engaged in research and development by performing sector and occupational category" to read in that table (downloading it first if it's the first time accessing it) run:

```
df = sc.table_to_df("271-000-22-01")
```

If there are only a couple specific series of interest you can also read them into a dataframe (whether they're in different source tables or not) as follows:

```
df = sc.vectors_to_df(["v74804", "v41692457"])
```

The above command takes an optional start_date argument which will return a dataframe beginning with a reference date no earlier than the provided start date. By default it will return all available history for the V#s provided.

You can check which tables you have stored locally by running

```
sc.downloaded_tables
```

Which will return a list of table numbers.

If a table is locally stored, it will not automatically update if Statistics Canada releases an update. To update locally stored tables run:

```
sc.update_tables()
```

You can optionally pass in a list of tables if you only want a subset of the locally stored tables to be updated.

Finally, if you want to delete any tables you've loaded you can run:

```
sc.delete_tables("271-000-22-01")
```

CHAPTER

THREE

STATSCAN CLASS DOCUMENTATION

Core functions outlined in the Quickstart along with some extra functionality are described here:

class stats_can.api_class.**StatsCan**(*data_folder=None*)

Load Statistics Canada data and metadata into python.

Parameters

- **data_folder** (*Path/str, default None*) –
- **location to save/search for locally stored Statistics Canada** (*The*) –
- **tables. Defaults to the current working directory** (*data*) –

delete_tables(*tables*)

Remove locally stored tables.

Parameters **tables** (*str or [str]*) – tables to delete

Returns

Return type [deleted tables]

property downloaded_tables

Check which tables you've downloaded.

Checks the file "stats_can.h5" in the instantiated data folder and lists all tables stored there.

Returns

Return type [table_ids]

static get_code_sets()

Get code sets.

Code sets provide additional metadata to describe variables and are grouped into scales, frequencies, symbols etc.

Returns **code_sets** – one dictionary for each group of information

Return type [dict]

static get_tables_for_vectors(*vectors*)

Find which table(s) a V# or list of V#s is from.

Parameters **vectors** (*str or [str]*) – V#(s) to look up tables for

Returns

- **dictionary of vector** (*table pairs plus an*)
- *"all_tables" key with a list of all tables*

- *containing the input V#s*

```
>>> StatsCan.get_tables_for_vectors("v39050")
{39050: '10100139', 'all_tables': ['10100139']}
>>> StatsCan..get_tables_for_vectors(["v39050", "v1074250274"])
{39050: '10100139', 1074250274: '16100011', 'all_tables': ['10100139',
↪'16100011']}
```

table_to_df (*table*)

Read a table to a dataframe.

Parameters **table** (*str*) – The ID of the table of interest, e.g "271-000-22"

Returns

- *pandas.DataFrame* – Dataframe of the requested table
- *If the table has been previously loaded to the file in self.data_folder*
- *it will retrieve that locally stored dataframe. If it's unavailable it will*
- *download it and then return the table. To update a locally stored table,*
- *call StatsCan.update_tables(), optionally passing just the table number of interest*

static tables_updated_on_date (*date*)

Get a list of tables that were updated on a given date.

Parameters **date** (*str or datetime.date*) – The date to check tables

Returns **changed_tables** – one dictionary for each table with its update date

Return type [dict]

static tables_updated_today ()

Get a list of tables that were updated today.

Returns **changed_tables** – one dictionary for each table with its update date

Return type [dict]

update_tables (*tables=None*)

Update locally stored tables.

Compares latest available reference period in locally stored tables to the latest available on Statistics Canada and updates any tables necessary

Parameters **tables** (*str or [str], default None*) – Optional subset of tables to check for updates, defaults to update all downloaded tables

Returns

Return type [str] list of tables that were updated, empty list if no updates made

static vector_metadata (*vectors*)

Get metadata on vectors.

Parameters **vectors** (*str or [str]*) – V#(s) to retrieve metadata

Returns **vector_metadata** – list of dictionaries with one dict for each vector

Return type [dict]

vectors_to_df (*vectors*, *start_date=None*)

Get a dataframe of V#s.

Parameters

- **vectors** (*str or [str]*) – the V#s to retrieve
- **start_date** (*datetime.date, optional*) – earliest reference period to return, defaults to all available history

Returns

- *pandas.DataFrame* – Dataframe indexed on reference date, with columns for each V# input
- *Note that any V#s in tables that are not currently locally stored will*
- *have their tables downloaded prior to returning the dataframe*

static vectors_to_df_remote(*vectors*, *periods=1*, *start_release_date=None*, *end_release_date=None*)

Retrieve V# data directly from Statistics Canada.

Parameters

- **vectors** (*str or [str]*) – V#(s) to retrieve data for
- **periods** (*int, default 1*) – Number of periods to retrieve data. Note that this will be ignored if start_release_date and end_release date are set
- **start_release_date** (*datetime.date, default None*) – earliest release date to retrieve data
- **end_release_date** (*datetime.date, default None*) – latest release date to retrieve data

Returns

- *pandas.DataFrame* – Dataframe indexed on reference (not release) date, with columns for each V# input
- *Note that start and end release date refer to the dates the data was released,*
- *not the reference period they cover. For example. October labour force survey*
- *data is released on the first or second Friday of November.*

static vectors_updated_today()

Get a list of all V#s that were updated today.

Returns changed_series – one dictionary for each vector with its update date

Return type [dict]

CHAPTER

FOUR

CONTRIBUTING

Contributions to this project are welcome. Fork the repository from github,

You'll need a python environment with poetry and nox installed. A good guide for setting up an environment and project (that I used for this library) is hypermodern python.

After making any changes you can run nox to make sure testing and linting went ok, and then you should be good to submit a PR.

I'd also welcome contributions to the docs, or anything else that would make this tool better for you or others.

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex

INDEX