FRED (Federal Reserve Economic Data)

Info

Studente: Di Battista Mattia, 0304938;

Gruppo: DylanDog;

Specifica

L'obiettivo è definire un package in Python per interagire con le API messe a disposizione da FRED (sito fornitore di dati economici, organizzati in categorie).

Codice

L'applicazione è stata sviluppata in Jupyter, per poi esserne creato il relativo package (disponibile a www.pypi.com). Il codice scritto è dunque lo stesso, cambia solo l'organizzazione. Di seguito sono riportati i file, con i metodi implementati:

request.py

```
request_json(url);
#Effettua una richiesta di GET, sfruttando la libreria 'requests', e
ritorna il file .json.

get(url_base, url_key, type, id);
#Impacchetta l'input da passare a request_json(), come la chiave per l'API
e l'identificativo di ciò che si vuole scaricare (e.g. series, category).
```

database.py

```
connect_db(db_file);
#Ritorna l'istanza di connessione ad un db SQLite locale, prendendo come
parametro il nome del db (e.g. 'database.db').

create_db(db_file);
#Crea il db SQLite locale, con 4 tabelle: categories, seriess,
observations, seriess_downloaded.

insert_db(db_file, id, type, update);
#Permette il popolamento della tabelle contenute nel db. Il parametro
update consente di aggiornare le entries qualora queste fossero già
presenti.

get_db(db_file, type, id);
#Ritorna il dataframe corrispondente alla tabella che si vuole vedere (i.e.
categories, seriess, observations, seriess_downloaded).
```

```
download_insert_tree_category(db_file, id, type, update, array);
#Fornendo un id da cui partire, sono scaricate ricorsivamente tutte le
sottocategorie figlie. Il metodo è molto lento, soprattutto se fornito un
id 'alto', come lo 0. Per evitare di saturare il server FRED è stato
inserito un timer, di 0.5 secondi tra due chiamate.
```

display.py

```
convert_string_float(array_string);
#Operazione necessaria nel calcolo della media mobile.

display(x, y, title, id, x_label);
#Effettua il setting dei grafici da ritornare.

display_observation(id);
#Ritorna l'oggetto plt da visualizzare, basato sull'id della serie passata.

moving_average(input_array, day);
#Calcola la media mobile.

display_moving_average(id, day);
#Ritorna l'oggetto plt da visualizzare, basato sull'id della serie passata.
```

L'implementazione del case study richiesto (*Money, Banking, & Finance > Exchange Rates > Daily Rates*), è disponibile nel file *user.py*.

Installazione

Per installare il package:

```
pip install fred-matt-merman
```

Sono inoltre necessarie le librerie esterne:

- numpy;
- requests;
- pandas;
- matplotlib;