Package 'drugs'

June 10, 2015

Type Package

Title Drug Prices on Online Market Places

Version 0.1

Date 2015-06-09

Maintainer Nick Janetos <njanetos@econ.upenn.edu>

Description Drug Prices on Online Market Places

License GPL-2

Depends RMySQL, data.table, Quandl

Imports DBI

LazyData TRUE

Suggests knitr

VignetteBuilder knitr

R topics documented:

---

connect.database	*Opens a connection to the SQL server, and optionally connects to the database.*

Description

Opens a connection to the SQL server, and optionally connects to the database.

Usage

```
connect.database(dbname = NULL)
```

Arguments

dbname   The name of the database. A complete list of names can be found using show.databases after running connect.database().

Examples

```
connect.database("drugs_agora")
```

---

construct.index	*Constructs a price index by binning price data.*

Description

Constructs a price index by binning price data.

Usage

```
construct.index(market = "agora", category = "2361707", units = NULL,
  additionalQuery = "", binsize = 1000, scale = 1)
```

Arguments

market	The name of the marketplace to draw data from.
category	The category of drug to use.
units	Units the drug is measured in.
additionalQuery	
	Any additional SQL constraints on the price data.
scale	Optional scaling factor.
dates	The dates which form the endpoints of the bins. If a single number, then the full date range is used, and evenly split according to the number.

Examples

```
construct.index(market = "agora",
                category = "2361707",
                units = "mg",
                additionalQuery = "amount > 50",
                dates = 20)
```

---

disconnect.database   *Disconnects from the SQL server.*

Description

Disconnects from the SQL server.

Usage

```
disconnect.database()
```

---

get.connection   *Get database connection from the package environment*

Description

Get database connection from the package environment

Usage

```
get.connection()
```

---

get.date.range   *Returns the date that scraping began and ended for this marketplace.*

Description

Returns the date that scraping began and ended for this marketplace.

Usage

```
get.date.range()
```

`get.query`	*Returns the results of a SQL query. By default, performs an inner join on listings and prices and returns the first 10 results.*

Description

Returns the results of a SQL query. By default, performs an inner join on listings and prices and returns the first 10 results.

Usage

```
get.query(query = "SELECT * FROM Listing L INNER JOIN Listing_prices P ON L.id = P.Listing_id LIMIT 1
```

Arguments

query   The SQL query to run.

Examples

```
select.database("drugs_agora")
get.query("SELECT * FROM Listing L
                   INNER JOIN Listing_prices P
                   ON L.id = P.Listing_id
                   WHERE category = '2361707'
                   AND denomination = 'USD'
                   AND units = 'mg'")
```

`get.selected.database`	*Returns the database currently connected to.*

Description

Returns the database currently connected to.

Usage

```
get.selected.database()
```

---

plot.index	*Plots price indices.*

Description

Plots price indices.

Usage

```
## S3 method for class 'index'
plot(market = "agora", category = "2361707", units = NULL,
  additionalQuery = "", binsize = 1000, dateRange = NULL, scale = 1,
  type = "l")
```

Arguments

market	The name of the marketplace to draw data from, if a vector, then plots multiple lines.
category	The category of drug to use.
units	Units the drug is measured in.
additionalQuery	
	Any additional SQL constraints on the price data.
dateRange	The dates which form the endpoints of the bins. If a single number, then the full date range is used, and evenly split according to the number.
scale	Optional scaling factor.

Examples

```
plot.index(market = c("agora", "evolution",
               category = "2361707",
               units = "mg",
               additionalQuery = "amount > 50")
```

---

select.database	*Connects to the database.*

Description

Connects to the database.

Usage

```
select.database(dbname)
```

Arguments

dbname   The name of the database. A complete list of names can be found using show.databases
                after running connect.database().

Examples

```
select.database("drugs_agora")
```

---

 show.databases   *Lists all available databases.*

Description

Lists all available databases.

Usage

```
show.databases()
```

Index