rwrds

April 3, 2019

Type Package

Title An R package for downloading and cleaning WRDS data.

Version 0.1.0

Description An R package for downloading and cleaning data from WRDS.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr,
dbplyr,
DBI,
RPostgres,
sqldf

RoxygenNote 6.1.1

R topics documented:

---

compustat   *Download Compustat*

Description

compustat downloads the annual or quarterly Compustat table.

Usage

```
compustat(wrds, begin_year, end_year, frequency, vars = "default")
```

Arguments

wrds	WRDS connection object from wrds_connect function.
begin_year	Numeric.
end_year	Numeric.
frequency	Character. Input either 'annual' or 'quarterly'.
vars	Optional character. Defaults to a recommended subset of variables. Choose 'all' if you want to download all variables. Otherwise, provide a list of variables.

Details

Downloads the annual or quarterly Compustat table. The table address is either compa.funda or compa.fundq. By default, the function downloads a subset of the variables. You can specify your own variable list by providing a string vector to the vars function. You can download all the variables by setting vars equal to 'all'. By default, mergeos on industry information from the names table and removes duplicate gvkey fiscal quarter or fiscal year observations. The de-duping logic is to keep the observation with the fewest NA values. See function documentation for compustat_annual and compustat_quarterly.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
compa_df = compustat(wrds = wrds, begin_year = 2010, end_year = 2012, frequency = 'annual', vars = c('gvkey','fy
```

---

compustat_annual   *Download annual Compustat*

Description

compustat_annual downloads the Compustat table with annual observations.

Usage

```
compustat_annual(wrds, begin_year, end_year, subset = TRUE)
```

Arguments

wrds	WRDS connection object from wrds_connect function.
begin_year	Numeric.
end_year	Numeric.
subset	Optional Boolean. Download recommended subset of variables? Defaults to TRUE.

Details

Downloads the annual observation Compustat table that is updated at an annual frequency. The table address is compa.funda. By default, the function only downloads a subset of the variables. You can download all the variables by setting the subset argument to FALSE. By default, merges on industry information from the names table and removes duplicate gvkey fiscal year observations. The de-duping logic is to keep the observation with the fewest NA values.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
compa_df = compustat_annual(wrds = wrds, begin_year = 2010, end_year = 2012)
```

---

compustat_append_crsp_links

Append CRSP identifiers onto a Compustat data frame

Description

compustat_append_crsp_links appends on CRSP permno and permco identifiers onto Compustat gvkey identifiers.

Usage

```
compustat_append_crsp_links(wrds, comp_df)
```

Arguments

wrds	WRDS connection object from wrds_connect function.
comp_df	Compustat data.frame. Requires the gkvey and datadate fields to be present.

Details

Appends on CRSP permno and permcos based on the standard Compsutat-CRSP linking table. The linking table address is crspa.ccmxpf_linktable. Requires that the Compustat data.frame be physically located in memory (not a lazy dplyr table reference). Requires that the gvkey and datadate fields are present in the Compustat data.frame.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
compa_annual = compustat_annual(wrds = wrds, begin_year = 2010, end_year = 2011)
comp_linked = compustat_append_crsp_links(wrds = wrds, comp_df = compa_annual)
```

---

compustat_crsp_annual   *Download annual Compustat with CRSP annual returns*

Description

compustat_annual downloads the annual Compustat table and merges on annual CRSP returns.

Usage

```
compustat_crsp_annual(wrds, begin_year, end_year, subset = TRUE)
```

Arguments

wrds	WRDS connection object from wrds_connect function.
begin_year	Numeric.
end_year	Numeric.
subset	Optional Boolean. Download recommended subset of variables? Defaults to TRUE.

Details

Downloads the annual Compustat file using the compustat_annual function. Appends on CRSP links and returns using the functions compustat_append_crsp_links and crsp_annual, respectively. The annual return merge is naive and is based on a fiscal year to calendar year match.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
comp_crsp_df = compustat_crsp_annual(wrds = wrds, begin_year = 2010, end_year = 2012)
```

---

compustat_crsp_linking_table
                    Download Compustat-CRSP linking table

Description

compustat_crsp_linking_table downloads the Compustat-CRSP linking table.

Usage

```
compustat_crsp_linking_table(wrds, dl = TRUE)
```

Arguments

wrds	WRDS connection object from wrds_connect function.
dl	Optional Boolean. Download the data? Defaults to TRUE. FALSE outputs a lazy dplyr table reference.

Details

Downloads the standard Compustat-CRSP linking table. The table address is crspa.ccmxpf_linktable. By default, the function downloads the data. A lazy `dplyr` table reference can be returned by setting the `dl` argument to `FALSE`.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
linking_tbl = compustat_crsp_linking_table(wrds = wrds, dl = FALSE)
```

---

compustat_names   *Download Compustat names table*

Description

`compustat_names` downloads the Compustat names table.

Usage

```
compustat_names(wrds, subset = TRUE, dl = TRUE)
```

Arguments

wrds	WRDS connection object from `wrds_connect` function.
subset	Optional Boolean. Download recommended subset of variables? Defaults to `TRUE`.
dl	Optional Boolean. Download the data? Defaults to `TRUE`. `FALSE` outputs a lazy `dplyr` table reference.

Details

The names table contains general gvkey-level information, such as NAICS industry. The table address is compa.names. By default, the function only downloads a subset of the variables. You can download all the variables by setting the `subset` argument to `FALSE`.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
names_df = compustat_names(wrds = wrds, subset = TRUE, dl = TRUE)
```

---

crsp *Download CRSP*

Description

crsp downloads the annual, monthly, or daily CRSP table.

Usage

```
crsp(wrds, begin_year, end_year, frequency, dl = FALSE)
```

Arguments

wrds	WRDS connection object from wrds_connect function.
begin_year	Numeric.
end_year	Numeric.
frequency	Character. Input either 'annual', 'monthly', or 'daily'.
dl	Optional Boolean. Download the data? Defaults to TRUE. FALSE outputs a lazy dplyr table reference.

Details

Downloads the annual, monthly, or daily CRSP Compustat table. The schema is crspa and the table is either the aggregated msf, msf, or dsf. Removes observations with weakly negative prices and shares outstanding. The function merges on identifying information from the CRSP header table. Please see either crsp_annual, crsp_monthly, or crsp_daily documentation for more details.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
crsp_df = crsp(wrds = wrds, begin_year = 2010, end_year = 2012, frequency = 'annual')
```

---

crsp_annual *Download CRSP annual returns and end-of-year prices*

Description

crsp_annual downloads CRSP annual returns and end-of-year prices.

Usage

```
crsp_annual(wrds, begin_year, end_year, dl = TRUE)
```

Arguments

wrds	WRDS connection object from wrds_connect function.
begin_year	Numeric.
end_year	Numeric.
dl	Optional Boolean. Download the data? Defaults to TRUE. FALSE outputs a lazy dplyr table reference.

Details

Constructs annual returns from the monthly stock file, located at crspa.msf. Only keeps permnos with a full year of data with non-missing values for returns and strictly positive values for shares outstanding and price. The function merges on identifying information from the CRSP header table.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
crspa_df = crsp_annual(wrds = wrds, begin_year = 2010, end_year = 2012)
```

---

crsp_daily   *Download CRSP daily return and price data*

Description

crsp_daily downloads CRSP daily returns and end-of-month prices.

Usage

```
crsp_daily(wrds, begin_year, end_year, dl = TRUE)
```

Arguments

wrds	WRDS connection object from wrds_connect function.
begin_year	Numeric.
end_year	Numeric.
dl	Optional Boolean. Download the data? Defaults to TRUE. FALSE outputs a lazy dplyr table reference.
subset	Optional Boolean. Download recommended subset of variables? Defaults to TRUE.

Details

Downloads data from table crspa.dsf. Removes observations with weakly negative prices and shares outstanding. The function merges on identifying information from the CRSP header table.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
crspd_df = crsp_daily(wrds = wrds, begin_year = 2010, end_year = 2010)
```

---

crsp_header   *Download CRSP header table*

Description

`crsp_header` downloads the CRSP header table.

Usage

```
crsp_header(wrds, subset = TRUE, dl = TRUE)
```

Arguments

wrds   WRDS connection object from `wrds_connect` function.

subset   Optional Boolean. Download recommended subset of variables? Defaults to
                TRUE.

dl   Optional Boolean. Download the data? Defaults to TRUE. FALSE outputs a lazy
                `dplyr` table reference.

Details

The header table contains permno-level information, such as SIC industry. The table address is
crspa.msfhdr. By default, the function only downloads a subset of the variables. You can download
all the variables by setting the subset argument to FALSE.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
crsp_names = crsp_header(wrds = wrds, subset = TRUE, dl = FALSE)
```

---

crsp_monthly   *Download CRSP monthly return and price data*

Description

`crsp_monthly` downloads CRSP monthly returns and end-of-month prices.

Usage

```
crsp_monthly(wrds, begin_year, end_year, dl = TRUE)
```

Arguments

wrds   WRDS connection object from `wrds_connect` function.

begin_year   Numeric.

end_year   Numeric.

dl   Optional Boolean. Download the data? Defaults to TRUE. FALSE outputs a lazy
                `dplyr` table reference.

Details

Downloads data from table crspa.msf. Removes observations with weakly negative prices and shares outstanding. The function merges on identifying information from the CRSP header table.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
crspm_df = crsp_monthly(wrds = wrds, begin_year = 2010, end_year = 2012)
```

---

hello	*Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

---

mergent_corporates	*Download Mergent corporate bonds*

Description

mergent_corporates downloads corporate bond issues from the Mergent issues table

Usage

```
mergent_corporates(wrds, clean = TRUE, vanilla = TRUE, subset = TRUE,
  dl = TRUE)
```

Arguments

wrds	WRDS connection object from wrds_connect function.
clean	Optional Boolean. Clean the data to remove data errors and missings? Defaults to TRUE.
vanilla	Optional Boolean. Restrict data to common vanilla bonds? Defaults to TRUE.
subset	Optional Boolean. Clean the data for errors and missing fields? Defaults to TRUE.
dl	Optional Boolean. Download the data? Defaults to TRUE. FALSE outputs a lazy dplyr table reference.

Details

Downloads issue-level bond data from the table fisd.fisd_mergedissue for corporate bonds. Corpo-
rates are defined by being in utility, financial, or industrial Mergent industry groups, having bond
types in CDEB, CMTN, CZ, CMTZ, UCID, or CP, and having security levels in SEN, SENS, SUB,
or SS. This function calls the `mergent_issues` function to download the data You can download all
the variables by setting the `subset` argument to `FALSE`. By default, this function also cleans the data
and restricts to vanilla corporates, which removes: floating rates notes, convertible bonds, foreign
domiciled issuers, foreign currency notes, private placements, perpetuals, preferred securities, retail
notes, slobs, exchangeables, unit deals, pay-in-kinds, defeased bonds, and bonds with non-strictly
positive offering amounts and missing coupon types, issue ids, maturity, and offering date. By de-
fault, also merges on initial credit ratings for the issue and makes simple variable transformations.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
mergent_corps = mergent_corporates(wrds, clean = TRUE, vanilla = TRUE, subset = TRUE, dl = FALSE)
```

---

mergent_issues   *Download Mergent bond issues table*

Description

`mergent_issues` downloads the Mergent bond issues table.

Usage

```
mergent_issues(wrds, clean = TRUE, vanilla = TRUE, subset = TRUE,
  dl = TRUE)
```

Arguments

wrds	WRDS connection object from `wrds_connect` function.
clean	Optional Boolean. Clean the data to remove data errors and missings? Defaults to `TRUE`.
vanilla	Optional Boolean. Restrict data to common vanilla bonds? Defaults to `TRUE`.
subset	Optional Boolean. Clean the data for errors and missing fields? Defaults to `TRUE`.
dl	Optional Boolean. Download the data? Defaults to `TRUE`. `FALSE` outputs a lazy `dplyr` table reference.

Details

Downloads issue-level bond data from the table fisd.fisd_mergedissue. Appends on information
from the callable, agentid, and exchangeable tables. By default, the function only downloads a
subset of the variables. You can download all the variables by setting the `subset` argument to
`FALSE`. By default, this function also cleans the data and restricts to vanilla bonds, which removes:
floating rates notes, convertible bonds, foreign domiciled issuers, foreign currency notes, private
placements, perpetuals, preferred securities, retail notes, slobs, exchangeables, unit deals, pay-in-
kinds, defeased bonds, and bonds with non-strictly positive offering amounts and missing coupon
types, issue ids, maturity, and offering date. By default, also merges on initial credit ratings for the
issue and makes simple variable transformations.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
mergent_issues_df = mergent_issues(wrds, clean = TRUE, vanilla = TRUE, subset = TRUE, dl = FALSE)
```

---

wrds_connect	*Connect to the WRDS cloud*

Description

wrds_connect creates a remote connection to the WRDS cloud.

Usage

```
wrds_connect(username, password)
```

Arguments

username	Character. WRDS username.
password	Character. WRDS password.

Details

Note: WRDS cloud runs PostgreSQL.

Examples

```
wrds = wrds_connect(username = Sys.getenv("WRDS_NAME"), password = Sys.getenv("WRDS_PASS"))
```

---

wrds_schema_list	*List available schemas in WRDS*

Description

wrds_schema_list downloads a list of the available schemas in the WRDS cloud.

Usage

```
wrds_schema_list(wrds)
```

Arguments

wrds	WRDS connection object from wrds_connect function.

Details

Schemas represent different data vendors in WRDS. Schemas can be thought of as separate databases that contain their own set of tables.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
schema_list = wrds_schema_list(wrds = wrds)
```

---

wrds_table_list   *List the tables in a WRDS schema.*

Description

`wrds_table_list` lists the tables in a WRDS schema

Usage

```
wrds_table_list(wrds, schema)
```

Arguments

wrds	WRDS connection object from `wrds_connect` function.
schema	Character. WRDS schema from `wrds_schema_list`.

Details

Each WRDS schema contains its own set of tables. Tables are queried directly in the package.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
table_list = wrds_table_list(wrds = wrds, schema = "compa")
```

---

wrds_variable_list   *List the variables in a WRDS table*

Description

`wrds_variable_list` lists the columns (variables) in a WRDS table.

Usage

```
wrds_variable_list(wrds, schema, table)
```

Arguments

wrds	WRDS connection object from `wrds_connect` function.
schema	Character. WRDS schema from `wrds_schema_list`.
table	Character. WRDS table from `wrds_table_list`.

Details

NA.

Examples

```
wrds = wrds_connect(username = "testing", password = "123456")
variable_list = wrds_variable_list(wrds = wrds, schema = "compa", table = "funda")
```

Index