Package 'fcall'

May 8, 2026

Title Parse Farm Credit Administration Call Report Data into Tidy Data Frames

Version 0.1.6

Description Parses financial condition and performance data (Call Reports) for institutions in the United States Farm Credit System. Contains functions for downloading files from the Farm Credit Administration (FCA) Call Report archive website and reading the files into tidy data frame format. The archive website can be found at <https://www.fca.gov/bank-oversight/call-report-data-for-download>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 4.1)

Imports cli, dplyr, glue, purrr, rlang, stringr, tibble, tidyr, utils, waldo

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

URL https://ketchbrookanalytics.github.io/fcall/, https://github.com/ketchbrookanalytics/fcall

BugReports https://github.com/ketchbrookanalytics/fcall/issues

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Michael Thomas [aut, cre], Ivan Millanes [aut], Ketchbrook Analytics [cph, fnd]

Maintainer Michael Thomas <mthomas@ketchbrookanalytics.com>

Repository CRAN

Date/Publication 2025-12-10 19:40:02 UTC

Contents

`compare_files_content` *Compare content of a specific file between two folders*

Description

`compare_files_content()` reads the content of a specified file from two folders and compares the content using the `waldo::compare` function. It identifies any differences in the content and returns the comparison results.

Usage

```
compare_files_content(filename, dir1, dir2)
```

Arguments

`filename`	A character string specifying the name of the file to compare.
`dir1`	A character string specifying the path to the first folder.
`dir2`	A character string specifying the path to the second folder.

Details

compare_files_content() reads the content of the specified file from both folders using readLines(), and compares the content using the waldo::compare() function.

Value

A list containing information about differences in the content of the specified file.

---

compare_metadata *Compare FCA Call Report metadata files between two folders*

Description

compare_metadata() compares the content of the metadata files (files that start with "D_") between two specified folders containing FCA Call Report data (from two different quarters).

Usage

```
compare_metadata(dir1, dir2)
```

Arguments

dir1	(String) The path to a folder containing FCA Call Report .TXT files for a single quarter
dir2	(String) The path to a folder containing FCA Call Report .TXT files for a (different) single quarter

Details

compare_metadata() lists metadata files in each folder, identifies shared metadata files, and then compares (a) the number of files, (b) file names, (c) file order, and (d) file content (using the waldo::compare() function).

Value

A list containing information about differences in file names, file order, and content differences between the metadata files in dir1 and dir2

Examples

```
# Download data from September 2025
path_1 <- tempfile("fcadata1")
dir.create(path_1)

download_data(
  year = 2025,
  month = 9,
```

```
  dest = path_1
)

# Download data from September 2011
path_2 <- tempfile("fcadata2")
dir.create(path_2)

download_data(
  year = 2011,
  month = 9,
  dest = path_2
)

compare_metadata(path_1, path_2)
```

`download_data` *Download FCA Call Report Data and Unzip*

Description

Download FCA Call Report Data and Unzip

Usage

```
download_data(year, month, dest, files = NULL, quiet = FALSE)
```

Arguments

`year`	(Integer) The year of the Call Report (e.g., `2022`)
`month`	(String) The month of the Call Report (e.g., `"March"`); you may also supply an integer (e.g., `3`) representing the month numerically
`dest`	(String) The path to the directory where the data will be downloaded (and unzipped) into
`files`	(Optional) Character vector, representing the names of the files in the .zip archive to be downloaded; default is `NULL`, meaning all files will be downloaded
`quiet`	(Optional) Logical. Controls whether download progress messages are displayed in the console. Defaults to `TRUE`.

Details

FCA publishes Call Report data quarterly. These .zip files are typically named "`YYYY`March.zip", "`YYYY`June.zip", "`YYYY`September.zip" and "`YYYY`December.zip" (where `YYYY` represents the 4-digit year). Therefore, valid values to the `month` argument should be limited to `c(3, 6, 9, 12)`, unless there is an anomaly in FCA's reporting/publishing. Check https://www.fca.gov/bank-oversight/call-report-data-for-download to ensure the data is available for the quarter you are interested in. Ketchbrook Analytics downloads these files and stores them in a public AWS S3 bucket, which is the location that `download_data()` retrieves them from.

Value

Console message informing the user where the data was successfully downloaded (and unzipped) into

Examples

```
path_1 <- tempfile("fcadata1")
dir.create(path_1)

download_data(
  year = 2025,
  month = "September",   # using the name of the month
  dest = path_1
)

list.files(path_1)

path_2 <- tempfile("fcadata2")
dir.create(path_2)

download_data(
  year = 2025,
  month = 9,   # using the month number (to refer to September)
  dest = path_2,
  # only download the following files
  files = c(
    "D_INST.TXT",
    "INST_Q202509_G20251112.TXT"
  )
)

list.files(path_2)
```

---

file_metadata *Descriptions for data files*

Description

Descriptions for data files

Usage

```
file_metadata
```

Format

`file_metadata`**:**
A data frame with 36 rows and 2 columns:

file_prefix Data file root name

description Short description of data file

Source

Metadata files headers

`get_codes_dict` *Retrieve dictionary of lookup codes for a specified dataset name*

Description

`get_codes_dict()` searches for an internal .rda file in the specified package and retrieves the codes dictionary based on the provided data name and naming convention. The naming convention is assumed to include the data name followed by a double underscore "__".

Usage

```
get_codes_dict(data_name)
```

Arguments

`data_name`	A character string specifying the data name to retrieve the codes dictionary for.

Details

`get_codes_dict()` uses the provided data name to construct the expected naming convention and searches for an internal .rda file in the specified package. If found, it attempts to retrieve the codes dictionary using `get` and returns it; otherwise, it returns NULL.

Value

A list with the codes dictionary (`codes_dict`) and the associated variable name (`codes_varname`) if found, otherwise each element will be NULL.

Examples

```
rcb_dict <- get_codes_dict("RCB")

# Access codes dictionary
rcb_dict$codes_dict

# Access the name of the variable that stores the codes
rcb_dict$codes_varname
```

process_data *Process FCA Call Report data in a specified folder*

Description

process_data() reads the downloaded (and unzipped) .TXT files into tidy data frames, applying the schema from the "D_" files to the corresponding raw comma-separated data files, as well as storing the metadata from the "D_" files

Usage

```
process_data(dir)
```

Arguments

dir	(String) The path to a folder containing FCA Call Report .TXT files for a single quarter

Details

process_data() assumes that metadata and data files share a common root name (characters until the first underscore occurrence).

Value

A list containing processed data and metadata.

Examples

```
path <- tempfile("fcadata")
dir.create(path)

download_data(
  year = 2025,
  month = "September",
  dest = path
)

processed_data <- process_data(path)

# Access "RCB" data
processed_data$data$RCB

# Access "RCB" metadata
processed_data$metadata$RCB
```

process_data_file *Process a data file using metadata and codes dictionary*

Description

process_data_file() reads a data file, applies the provided metadata and codes dictionary, and organizes the data into a tidy format. The column names are determined based on the metadata scenario (e.g., "single", "single_multiple", "single_multiple_single").

Usage

```
process_data_file(file, metadata, dict = NULL)
```

Arguments

file	(String) The path to the data file
metadata	A list containing the scenario and variable information obtained from the metadata file using process_metadata_file.
dict	(Optional) A data frame containing codes dictionary information

Details

process_data_file() processes the data file according to the metadata scenario. It handles cases where variables have multiple occurrences and organizes the data into a tidy format with appropriate column names. The function relies on the read_data_file function for the actual data reading.

Value

A tibble containing the processed data in a tidy format

Examples

```
path <- tempfile("fcadata")
dir.create(path)

download_data(
  year = 2025,
  month = "September",
  dest = path
)

process_data_file(
  file = file.path(path, "RCB_Q202509_G20251112.TXT"),
  metadata = process_metadata_file(file.path(path, "D_RCB.TXT")),
  dict = RCB__INV_CODE
)
```

process_metadata_file *Process metadata file to extract variable information*

Description

process_metadata_file() reads a metadata file and extracts information about the column names, column types, decimal positions, and variable definitions.

Usage

```
process_metadata_file(file)
```

Arguments

file	(String) The path to the metadata file.

Details

process_metadata_file() processes metadata files following specific rules to handle encoding, remove unnecessary information, and extract variable details. It detects the scenario based on the occurrence of double asterisks in variable names.

Value

A list containing the scenario (e.g., "single", "single_multiple", "single_multiple_single") and a tibble with variable information.

Examples

```
path <- tempfile("fcadata")
dir.create(path)

download_data(
  year = 2025,
  month = "September",
  dest = path
)

process_metadata_file(file.path(path, "D_RC1.TXT"))
```

RCB2__AssetCodeRCB2 *Dictionary for AssetCodeRCB2 variable of RCB2 file*

Description

Dictionary for AssetCodeRCB2 variable of RCB2 file

Usage

```
RCB2__AssetCodeRCB2
```

Format

`RCB2__AssetCodeRCB2`:
A data frame with 17 rows and 2 columns:

code Integer code

value Character description

Source

D_RCB2.TXT

RCB3__DebtMaturityCode *Dictionary for DebtMaturityCode variable of RCB3 file*

Description

Dictionary for DebtMaturityCode variable of RCB3 file

Usage

```
RCB3__DebtMaturityCode
```

Format

`RCB3__DebtMaturityCode`:
A data frame with 10 rows and 2 columns:

code Integer code

value Character description

Source

D_RCB3.TXT

RCB__INV_CODE *Dictionary for INV_CODE variable of RCB file*

Description

Dictionary for INV_CODE variable of RCB file

Usage

```
RCB__INV_CODE
```

Format

RCB__INV_CODE:

A data frame with 35 rows and 2 columns:

code Integer code

value Character description

Source

D_RCB.TXT

RCF1__LOANSTATUS *Dictionary for LOANSTATUS variable of RCF1 file*

Description

Dictionary for LOANSTATUS variable of RCF1 file

Usage

```
RCF1__LOANSTATUS
```

Format

RCF1__LOANSTATUS:

A data frame with 13 rows and 2 columns:

code Integer code

value Character description

Source

D_RCF1.TXT

RCF__LOANSTATUS *Dictionary for LOANSTATUS variable of RCF file*

Description

Dictionary for LOANSTATUS variable of RCF file

Usage

```
RCF__LOANSTATUS
```

Format

`RCF__LOANSTATUS`**:**

A data frame with 6 rows and 2 columns:

code Integer code

value Character description

Source

D_RCF.TXT

RCI2B__DerivCode *Dictionary for DerivCode variable of RCI2B file*

Description

Dictionary for DerivCode variable of RCI2B file

Usage

```
RCI2B__DerivCode
```

Format

`RCI2B__DerivCode`**:**

A data frame with 23 rows and 2 columns:

code Integer code

value Character description

Source

D_RCI2B_2018.TXT

RCI2C__ExposureCode *Dictionary for ExposureCode variable of RCI2C file*

Description

Dictionary for ExposureCode variable of RCI2C file

Usage

```
RCI2C__ExposureCode
```

Format

`RCI2C__ExposureCode`**:**

A data frame with 12 rows and 2 columns:

code Integer code

value Character description

Source

D_RCI2C_2018.TXT

RCI2D__DerivRMCode *Dictionary for DerivRMCode variable of RCI2D file*

Description

Dictionary for DerivRMCode variable of RCI2D file

Usage

```
RCI2D__DerivRMCode
```

Format

`RCI2D__DerivRMCode`**:**

A data frame with 11 rows and 2 columns:

code Integer code

value Character description

Source

D_RCI2D_2018.TXT

RCO__ASSET_CODE *Dictionary for ASSET_CODE variable of RCO file*

Description

Dictionary for ASSET_CODE variable of RCO file

Usage

```
RCO__ASSET_CODE
```

Format

`RCO__ASSET_CODE`**:**

A data frame with 12 rows and 2 columns:

code Integer code

value Character description

Source

D_RCO.TXT

RCR3__RegCapCode *Dictionary for RegCapCode variable of RCR3 file*

Description

Dictionary for RegCapCode variable of RCR3 file

Usage

```
RCR3__RegCapCode
```

Format

`RCR3__RegCapCode`**:**

A data frame with 15 rows and 2 columns:

code Integer code

value Character description

Source

D_RCR3.TXT

RCR7__RegCapCode *Dictionary for RegCapCode variable of RCR7 file*

Description

Dictionary for RegCapCode variable of RCR7 file

Usage

```
RCR7__RegCapCode
```

Format

RCR7__RegCapCode**:**

A data frame with 29 rows and 2 columns:

code Integer code

value Character description

Source

D_RCR7.TXT

read_data_file *Read a data file based on metadata and codes dictionary*

Description

`read_data_file()` reads a data file and processes it based on the provided metadata and codes dictionary. The processing depends on the metadata scenario, which includes cases like `"single"`, `"single_multiple"`, and `"single_multiple_single"`. For certain scenarios, the function utilizes `read.csv` to infer column types without explicit specification.

Usage

```
read_data_file(file, metadata, dict)
```

Arguments

`file`	A character string specifying the path to the data file.
`metadata`	A list containing the scenario and variable information obtained from the metadata file using `process_metadata_file`.
`dict`	A data frame containing codes dictionary information.

Details

`read_data_file()` reads the data file and applies necessary processing based on the metadata scenario. For scenarios like `"single"` and `"single_multiple"`, it uses `read.csv` for convenient type inference. For `"single_multiple_single"`, it reads the file line by line, collapses every `(N_CODES + 2)` lines, and then reads the collapsed lines using `read.table`.

Value

A tibble containing the processed data.

RID__CAP_CODE *Dictionary for RegCapCode variable of RID file*

Description

Dictionary for RegCapCode variable of RID file

Usage

```
RID__CAP_CODE
```

Format

`RID__CAP_CODE`**:**

A data frame with 12 rows and 2 columns:

code Integer code

value Character description

Source

D_RID.TXT

RIE1__ACLCode *Dictionary for ACLCode variable of RIE1 file*

Description

Dictionary for ACLCode variable of RIE1 file

Usage

```
RIE1__ACLCode
```

Format

`RIE1__ACLCode`**:**

A data frame with 7 rows and 2 columns:

code Integer code

value Character description

Source

D_RIE1.TXT

Index