Tidy Finance and Accessing Financial Data

R Consortium Webinar

Christoph Scheuch

2024-03-06

What is Tidy Finance?

A **transparent**, **open-source** approach to research in financial economics, featuring **multiple programming languages**

tidy-finance.org offers tools to:

- Learn about empirical applications using tidy principles
- Learn to work with financial data in a tidy manner
- Teach students the importance of reproducible research
- Contribute to reproducible finance research via our blog

Why Tidy?

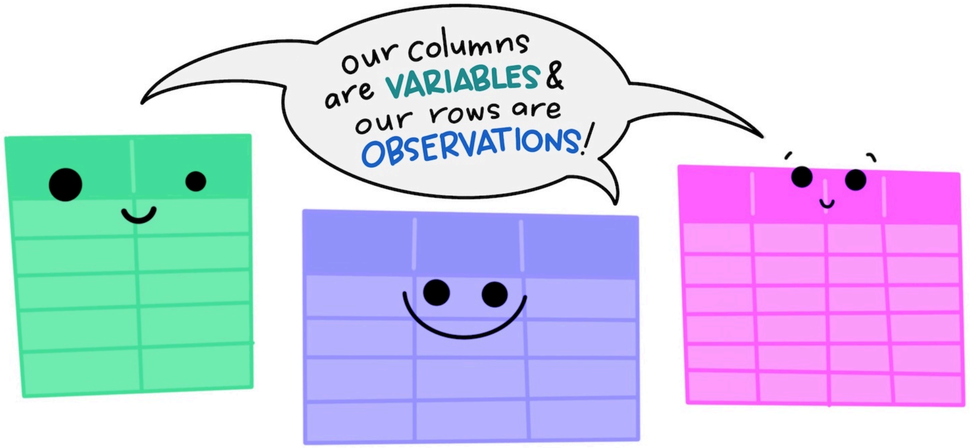
Code should not just be correct, but also follow principles:

1. Design so that code is **easy to read** for humans
2. **Compose simple functions** to solve complex problems
3. **Embrace functional programming** for reproducible results
4. **Reuse data structures** across applications

Focus of this talk: **tidy data**

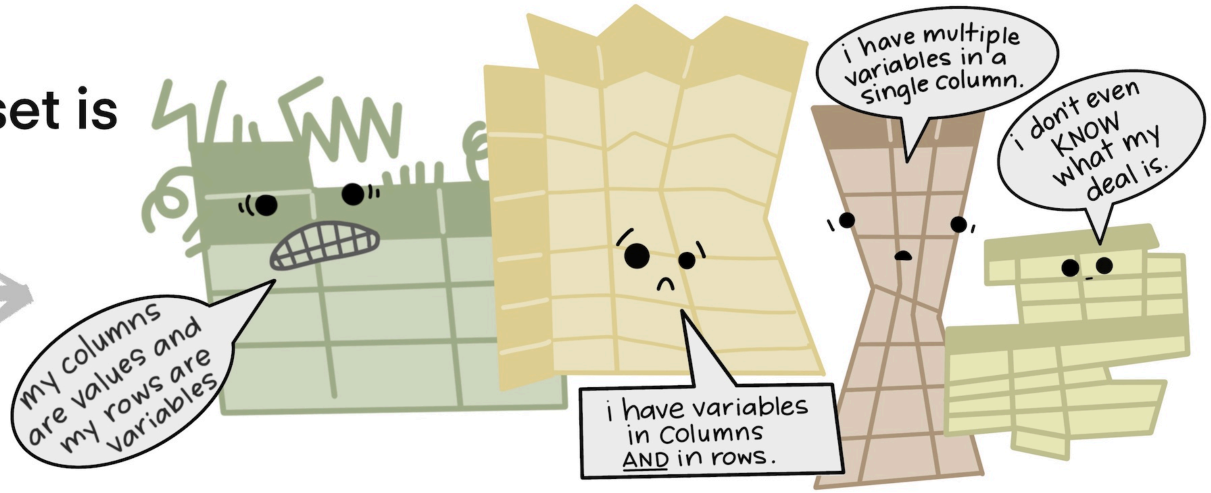
Recap: what is *tidy data*?

The standard structure of tidy data means that "tidy datasets are all alike…"



"…but every messy dataset is messy in its own way."

—HADLEY WICKHAM


my columns are values and my rows are variables
i have variables in columns AND in rows.
i have multiple variables in a single column.
i don't even KNOW what my deal is.

llustrations from the Openscapes blog Tidy Data for reproducibility, efficiency, and collaboration by Julia Lowndes and Allison Horst

Example chunks with *tidy code*

R Python

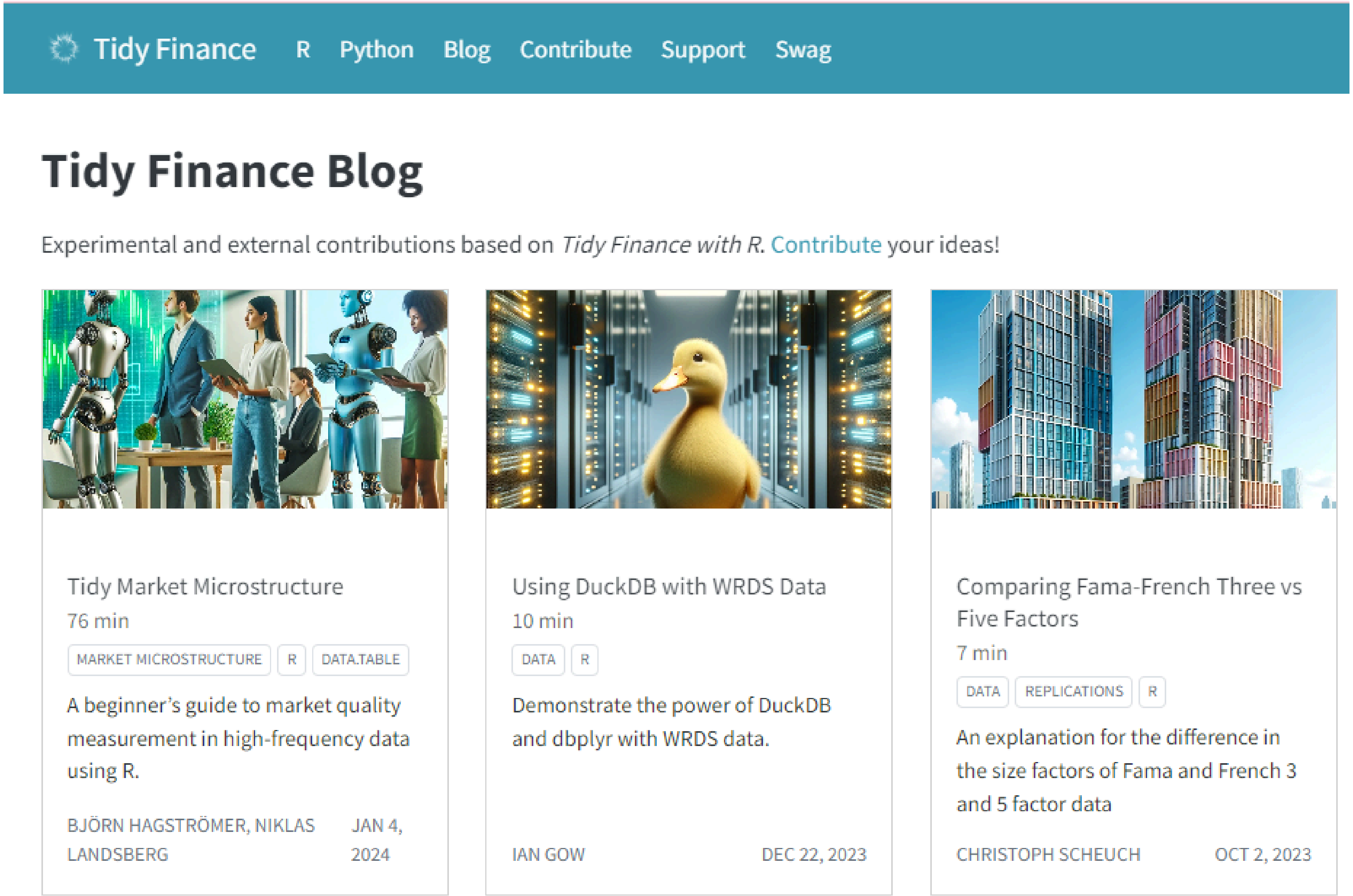
```r
# Load packages
library(tidyverse)
library(tidyquant)

# Download symbols of DOW index
symbols <- tq_index(x = "DOW") |>
  filter(company != "US DOLLAR")

# Download prices of DOW index constituents
prices <- tq_get(x = symbols, get = "stock.prices",
                 from = "2000-01-01", to = "2022-12-31")

# Calculate returns
returns <- prices |>
  group_by(symbol) |>
  mutate(ret = adjusted / lag(adjusted) - 1) |>
  select(symbol, date, ret) |>
  drop_na(ret)
```

Welcoming contributions on our blog



Tidy Finance Blog

Experimental and external contributions based on *Tidy Finance with R*. Contribute your ideas!

Tidy Market Microstructure

76 min

MARKET MICROSTRUCTURE · R · DATA.TABLE

A beginner's guide to market quality measurement in high-frequency data using R.

BJÖRN HAGSTRÖMER, NIKLAS LANDSBERG — JAN 4, 2024

Using DuckDB with WRDS Data

10 min

DATA · R

Demonstrate the power of DuckDB and dbplyr with WRDS data.

IAN GOW — DEC 22, 2023

Comparing Fama-French Three vs Five Factors

7 min

DATA · REPLICATIONS · R

An explanation for the difference in the size factors of Fama and French 3 and 5 factor data

CHRISTOPH SCHEUCH — OCT 2, 2023

Maintainers of tidy-finance.org

Christoph Scheuch

Head of Artificial Intelligence at wikifolio.com

Stefan Voigt

Assistant Professor of Finance at University of Copenhagen

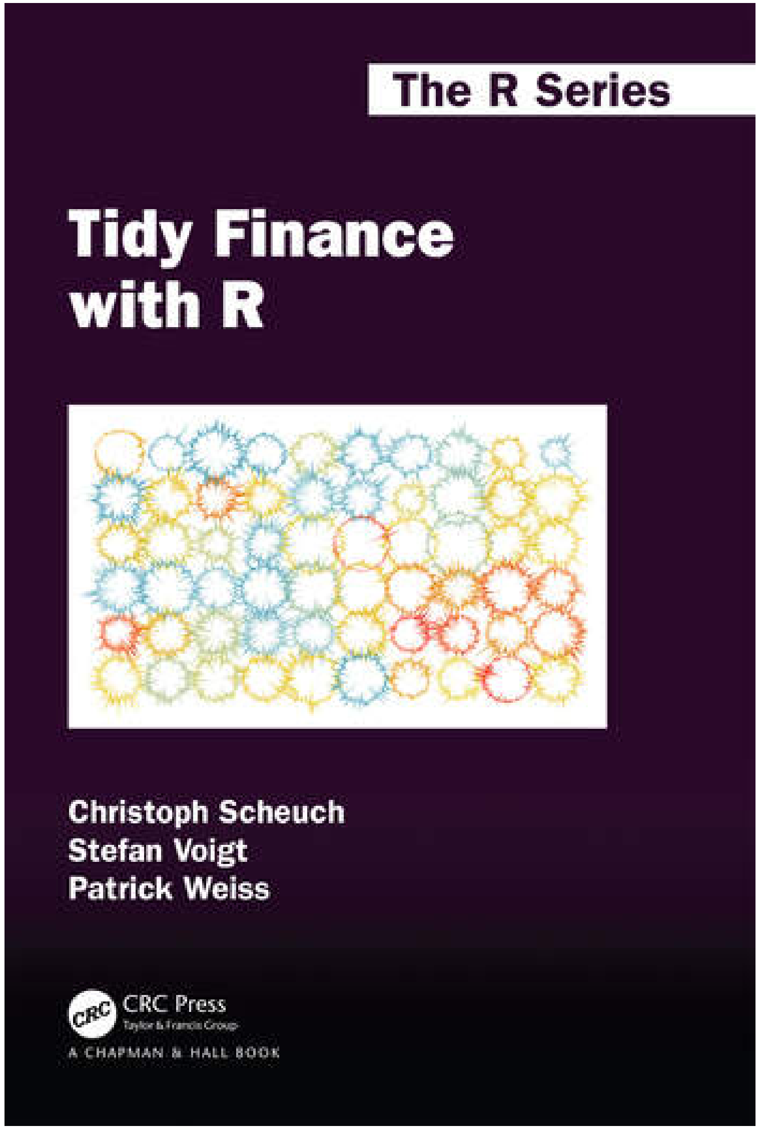
Patrick Weiss

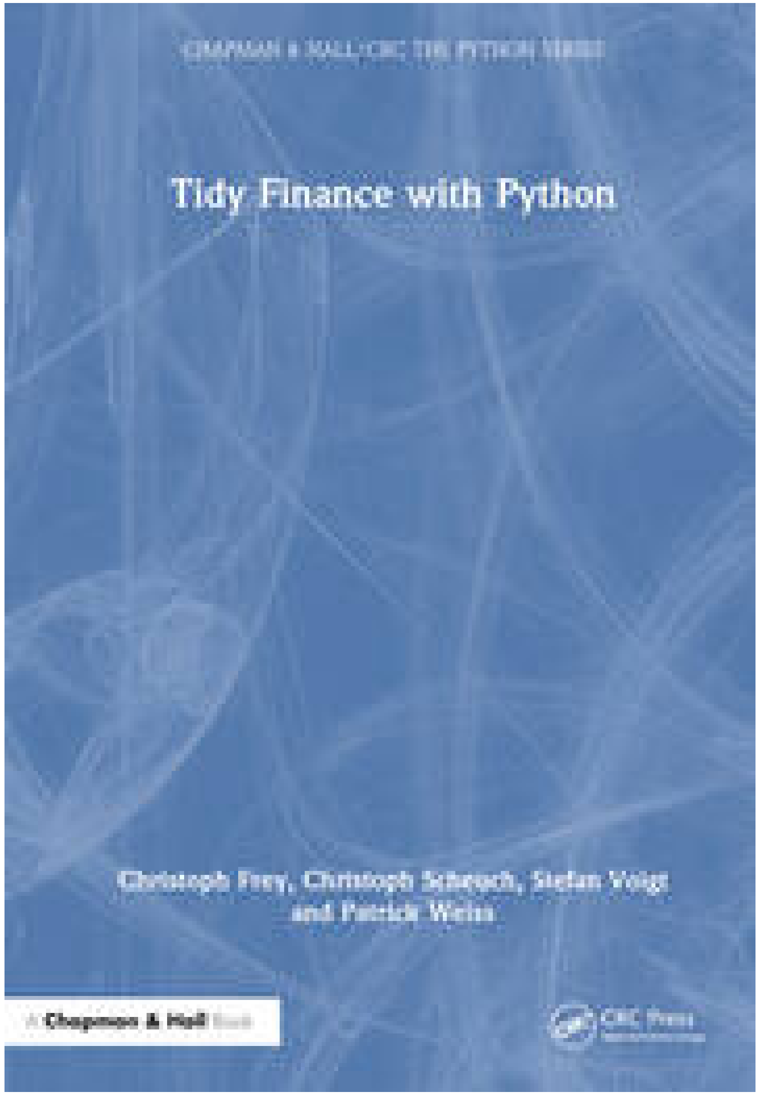
Assistant Professor of Finance at Reykjavik University

Christoph Frey

Quantitative Researcher at Pinechip Capital

We also wrote books




CHAPMAN & HALL/CRC THE PYTHON SERIES
Tidy Finance with Python
Christoph Frey, Christoph Scheuch, Stefan Voigt
and Patrick Weiss
A Chapman & Hall Book
CRC Press

Accessing & Managing Financial Data

Importance of organizing data efficiently

- **Challenge:** ensure consistency across various data sources
- **Solution:**
  - Use R to import, prepare & store data
  - Use SQLite to organize data in a database
- **R Packages:**
  - Manipulation: `tidyverse`
  - Import: `tidyquant`, `frenchdata`, `readxl`
  - Storage: `RSQLite`

Fama-French factors & portfolios

Most popular data for asset pricing tests since Fama and French (1993)

```r
library(frenchdata)

factors_ff3_monthly_raw <- download_french_data("Fama/French 3 Factors")
factors_ff3_monthly <- factors_ff3_monthly_raw$subsets$data[[1]] |>
  mutate(
    month = floor_date(ymd(str_c(date, "01")), "month"),
    across(c(RF, `Mkt-RF`, SMB, HML), ~as.numeric(.) / 100),
    .keep = "none"
  ) |>
  rename_with(str_to_lower) |>
  rename(mkt_excess = `mkt-rf`) |>
  select(month, everything())

print(factors_ff3_monthly, n = 5)
```

```
# A tibble: 1,170 × 5
  month      mkt_excess     smb     hml     rf
  <date>          <dbl>   <dbl>   <dbl>  <dbl>
1 1926-07-01     0.0296 -0.0256 -0.0243 0.0022
2 1926-08-01     0.0264 -0.0117  0.0382 0.0025
3 1926-09-01     0.0036 -0.014   0.0013 0.0023
4 1926-10-01    -0.0324 -0.0009  0.007  0.0032
5 1926-11-01     0.0253 -0.001  -0.0051 0.0031
# ℹ 1,165 more rows
```

q-Factors

Alternative to Fama-French data by Hou, Xue, and Zhang (2014)

```r
library(readr)

factors_q_monthly_link <-
  "https://global-q.org/uploads/1/2/2/6/122679606/q5_factors_monthly_2022.csv"

factors_q_monthly <- read_csv(factors_q_monthly_link) |>
  mutate(month = ymd(str_c(year, month, "01", sep = "-"))) |>
  select(-R_F, -R_MKT, -year) |>
  rename_with(~ str_remove(., "R_")) |>
  rename_with(~ str_to_lower(.)) |>
  mutate(across(-month, ~ . / 100))

print(factors_q_monthly, n = 5)
```

```
# A tibble: 672 × 5
  month            me       ia     roe       eg
  <date>        <dbl>    <dbl>   <dbl>    <dbl>
1 1967-01-01  0.0683  -0.0297  0.0192  -0.0218
2 1967-02-01  0.0165  -0.00227 0.0354   0.0222
3 1967-03-01  0.0200  -0.0178  0.0184  -0.0104
4 1967-04-01 -0.00690 -0.0288  0.0106  -0.0173
5 1967-05-01  0.0285   0.0252  0.00692  0.00158
# ℹ 667 more rows
```

Macroeconomic predictors

Collection of variables for equity premium prediction (Welch & Goyal, 2008)

```r
library(readxl)

download.file(
  url = "https://docs.google.com/spreadsheets/d/1g4LOaRj4TvwJr9RIaA_nwrXXWTOy46bP/expo
  destfile = "macro_predictors.xlsx",
  mode = "wb"
)

macro_predictors <- read_xlsx("macro_predictors.xlsx", sheet = "Monthly") |>
  mutate(
    # Several cleaning steps & variable transformations...
  )
```

```
# A tibble: 1,152 × 15
  month        rp_div    dp    dy    ep     de     svar    bm   ntis    tbl
  <date>        <dbl> <dbl> <dbl> <dbl>  <dbl>    <dbl> <dbl>  <dbl>  <dbl>
1 1926-12-01 -0.0220  -2.97 -2.96 -2.39 -0.586 0.000465 0.441 0.0509 0.0307
2 1927-01-01  0.0422  -2.94 -2.96 -2.37 -0.568 0.000470 0.444 0.0508 0.0323
3 1927-02-01  0.00363 -2.98 -2.93 -2.43 -0.549 0.000287 0.429 0.0517 0.0329
4 1927-03-01  0.0142  -2.98 -2.97 -2.45 -0.531 0.000924 0.470 0.0464 0.032
5 1927-04-01  0.0459  -2.98 -2.97 -2.47 -0.513 0.000603 0.457 0.0505 0.0339
# ℹ 1,147 more rows
# ℹ 5 more variables: lty <dbl>, ltr <dbl>, tms <dbl>, dfy <dbl>, infl <dbl>
```

Other macroeconomic data

10K data sets available via Federal Reserve Economic Data (FRED) database

```r
library(tidyquant)

# Example: consumer price index (CPI)
cpi_monthly <- tq_get("CPIAUCNS", get = "economic.data") |>
  mutate(
    month = floor_date(date, "month"),
    cpi = price / price[month == max(month)],
    .keep = "none"
  )
print(cpi_monthly, n = 5)
```

```
# A tibble: 121 × 2
  month        cpi
  <date>     <dbl>
1 2014-01-01 0.758
2 2014-02-01 0.761
3 2014-03-01 0.766
4 2014-04-01 0.769
5 2014-05-01 0.771
# ℹ 116 more rows
```

Use SQLite database for storage

```r
library(RSQLite)
library(dbplyr)

# Create database
tidy_finance <- dbConnect(
  SQLite(), "tidy_finance_r.sqlite", extended_types = TRUE
)

# Write data to database
dbWriteTable(
  conn = tidy_finance,
  name = "factors_ff3_monthly",
  value = factors_ff3_monthly,
  overwrite = TRUE
)

# Load data from database
factors_ff3_monthly <- tbl(tidy_finance, "factors_ff3_monthly") |>
  collect()
```

Why SQLite?

Pros:

- Lightweight, self-contained, serverless database engine
- Great for education purposes or prototyping

Cons:

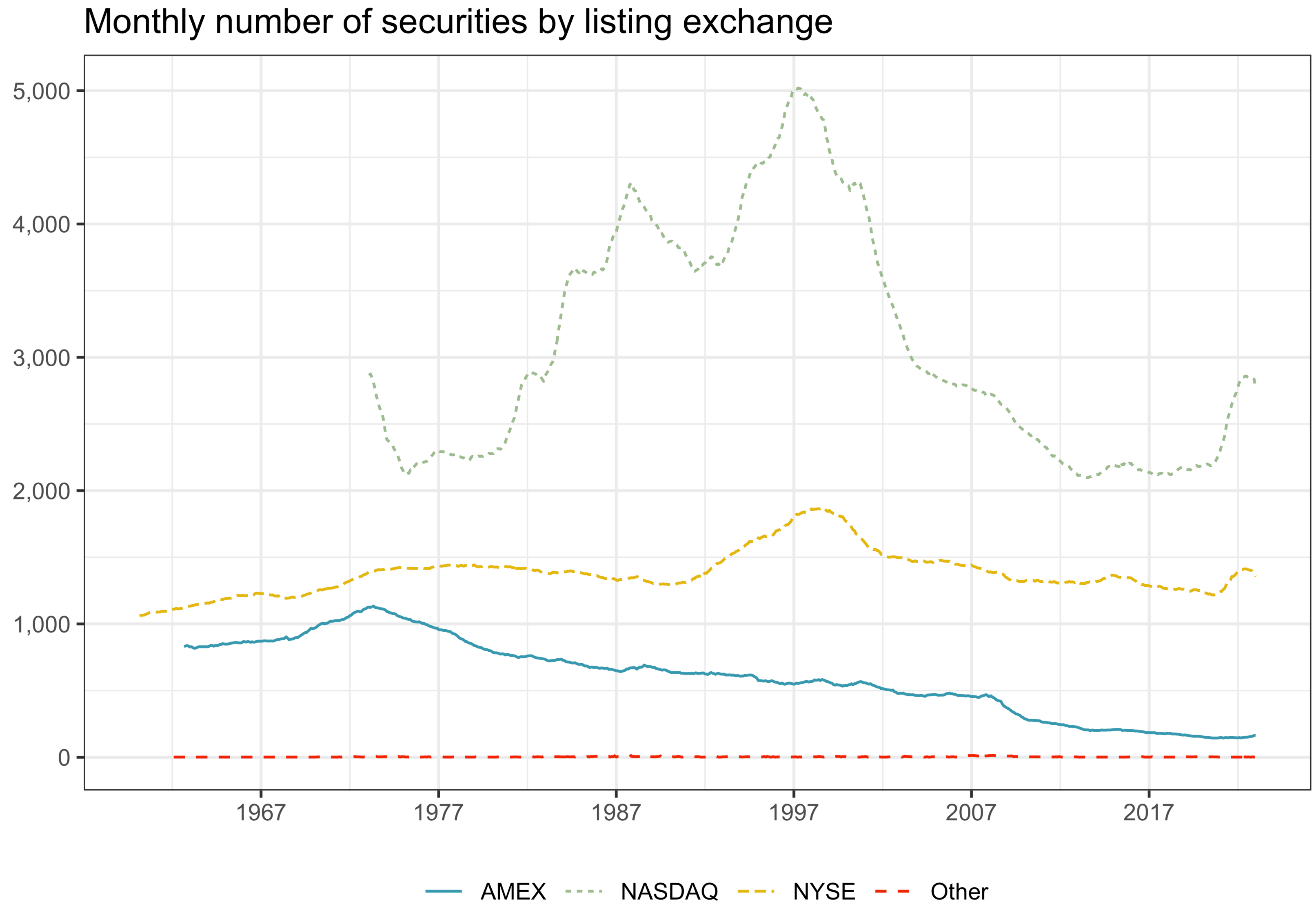
- Limitations with respect to very large data & concurrency
- Transfer to other languages cumbersome (e.g. Python)

WRDS & Other Data Providers

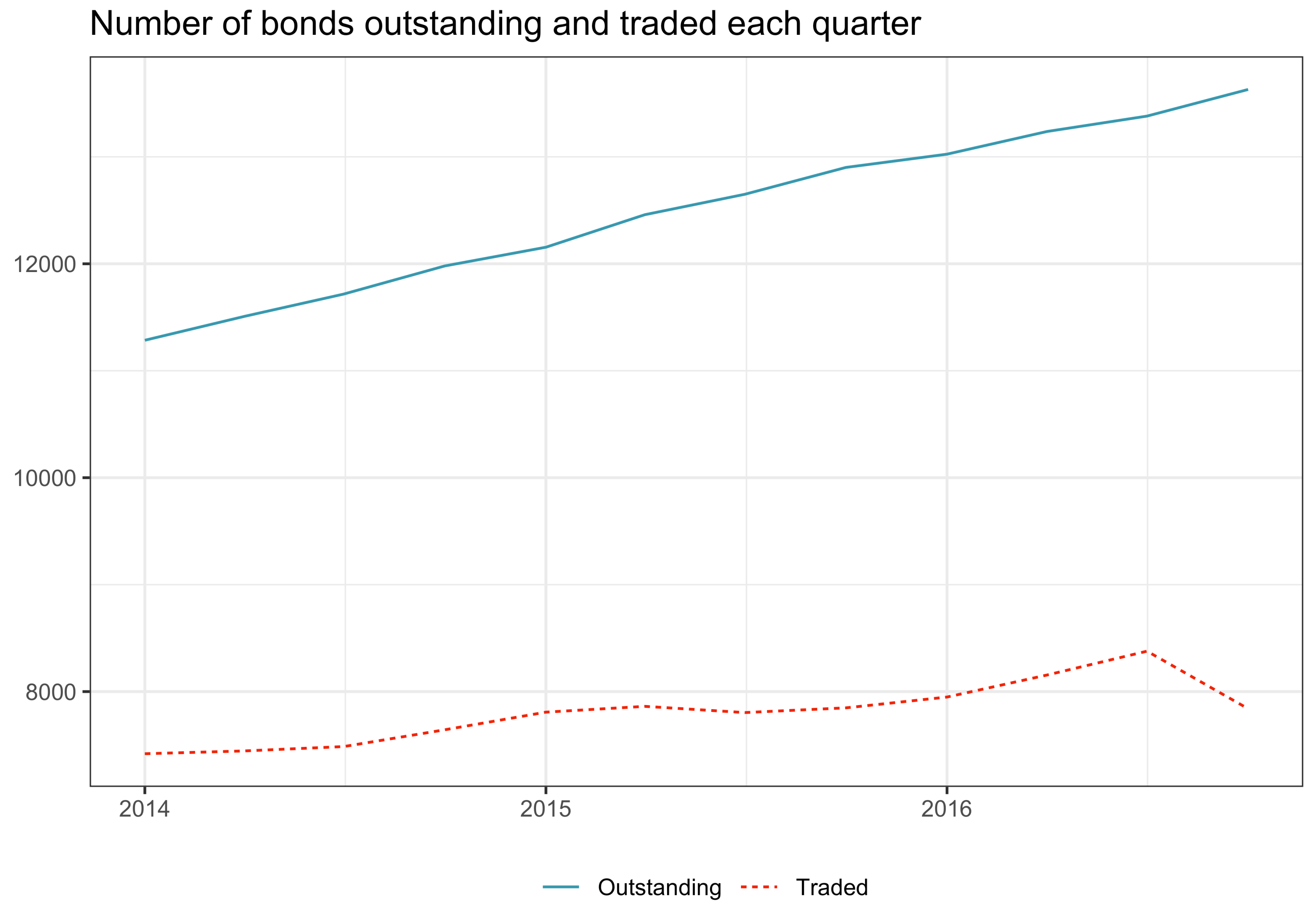
Wharton Research Data Services (WRDS)

- Popular provider of financial & economic data
- Focus on academic audience & research applications
- Access via `RPostgres` package
- Main data used in Tidy Finance
  - **CRSP:** historical monthly & daily returns for US stocks
  - **Compustat:** historical accounting data for US companies
  - **Mergent FISD:** characteristics of US corporate bonds
  - **TRACE:** detailed US corporate bond transactions

Glimpse at historical stock data



Glimpse at historical bond data



Other data providers

Large ecosystem of alternative data providers

- Extensive list of R packages on tidy-finance.org
- Examples: `fredr`, `ecb`, `Rblpapi`, `Quandl`, `edgarWebR`, etc.

Are we missing an important package?

- please reach out via **contact@tidy-finance.org**

Wrap-up

Tidy approach to teaching & research

`tidyfinance` R package to access financial data in a tidy way:

```r
install.packages("tidyfinance")

tidyfinance::download_data(
  type = "wrds_crsp_monthly",
  start_date = "1960-01-01", end_date = "2020-12-31"
)
```

- Detailed open source material at **tidy-finance.org**
- Get in touch for **teaching materials** & to **contribute to blog**
- Follow me for news linkedin.com/in/christophscheuch