Make Literate Slides!

Oliver M. Haynold

Sr. Director Pricing and Market Structure

CME Group

Open Source Quantitative Finance Conference (osQF)

Chicago, Spring 2025

All opinions are mine alone and do not constitute investment advice

Literate Slides

DOI 10.5281/zenodo.15186270

The Man Wants Slides

Common requirements

- Driven by business data from large and odd enterprise databases
- Repeated production, *e. g.*, each month or quarter
- Same content for different domains,

  e. g., sections repeating for business lines, products
- Different editions of same product, *e. g.*, one deck for each business line, meeting
- Rapid
- Accurate

Wouldn’t it be nice?

- Automatic production from a formal, declarative description
- Consistent math
- Consistent design
- Rich math & graphics

Can we cobble together a domain-specific language for data-driven business analysis slides?

DOI 10.5281/zenodo.15186270

Domain-specific Languages and Their Processors

Domain	Language and processor
Abstracting access to odd business databases	dbplyr
Data analysis	R, tidyverse, *etc.*
Control flow in and around slides	R.rsp
Rich graphics	ggplot2
Rich tables	knitr::kable, kableExtra
Typesetting	(Quarto), LaTeX, beamer

DOI 10.5281/zenodo.15186270

Abstracting Access to Odd Business Databases: dbplyr

```
# This accesses an enormous query/view from an enterprise database
# tradeRevenueMonthly is defined in an in-house helper package
productsVolume <- tradeRevenueMonthly |>
  filter(
      Original_Process_YearMonth >= "202501" & # first quarter 2025
      Original_Process_YearMonth <= "202503" &
      # US futures and options
      Product_Exchange %in% c("CME", "CBT", "NYMBS", "COMBS") &
      Future_Option_Indicator == "F" # futures only
          ) |>
  group_by(Product_Line, Original_Process_YearMonth) |>
  summarize(Volume_Sides = sum(Volume_Sides, na.rm = TRUE)) |>
  arrange(Product_Line, Original_Process_YearMonth) |>
  collect() |>
  mutate(Original_Process_YearMonth=
    as.yearmon(Original_Process_YearMonth, format="%Y%m"))
```


DOI 10.5281/zenodo.15186270

Control Flow In and Around Slides: R.rsp

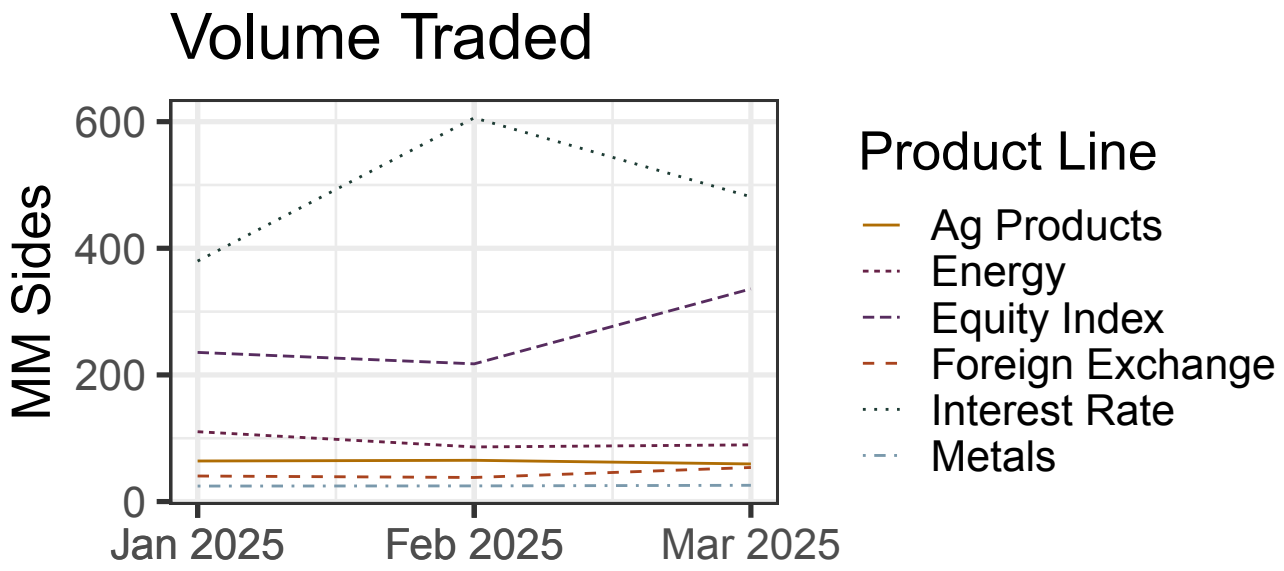
- Mix R arbitrarily with any language(s) of your choice (LaTeX, HTML, Markdown, …)
- Use all R language constructs and all target language constructs you like
- The spaghetti code monster is coming for you
  `<%@include file="file"%>` may help…
- Error messages tend to defy interpretation…
- (Doesn't play nice with %>% operator, use |>)

```
\begin{itemize}
<% for (i in 1:3) { %>
  \item The square of <%= i %> is <%= i*i %>.
<% } %>
\end{itemize}
```


DOI 10.5281/zenodo.15186270

Rich Graphics: ggplot2

```
ggplot(data=productsVolume, aes(Original_Process_YearMonth,
  Volume_Sides/1E6, colour=Product_Line, linetype=Product_Line)) +
   geom_line() +
   labs(x="",y="MM Sides", title="Volume Traded",
        colour="Product Line", linetype="Product Line") +
   scale_colour_product_line() + # CME house-style product colors
   theme_bw(20)
```



Read Leland Wilkinson: *The Grammar of Graphics*, 2nd ed. (Springer, 2005), $249?

DOI 10.5281/zenodo.15186270

Rich Tables: knitr::kable and kableExtra

```
productsVolume |>
  rename(`Product Line` = Product_Line) |>
  mutate(Volume_Sides = Volume_Sides / 1E6) |>
  # ...less than elegant call to kableExtra:cell_spec
  # for formatting omitted...
  spread(Original_Process_YearMonth,Volume_Sides) |>
  kable("latex", booktabs=T, escape=F, linesep="") |>
  row_spec(0, bold = T)
```

Sides traded, MM

Product Line	Jan 2025	Feb 2025	Mar 2025
Ag Products	64.1	65.2	59.4
Energy	110.3	86.2	89.5
Equity Index	235.6	217.5	335.9
Foreign Exchange	40.3	38.0	53.8
Interest Rate	379.8	606.0	481.3
Metals	24.5	24.7	25.7

DOI 10.5281/zenodo.15186270

What Works, What Doesn't?

Lots of fun

R & LaTeX

- Rock-solid, still going strong with roots in the 1970s for a reason

ggplot2

- From declarative grammar to beautiful output

dbplyr

- Really speedy business query writing (Use an editor with autocomplete!)

R.rsp

- Mix R with *any* language

Less fun

kableExtra

- Docs: "Only have to mess with raw HTML/LaTeX in the last 10% cases where kableExtra cannot solve the problem" (optimistic!)
- Need to turn `escape` off to inject LaTeX formatting, but then it's off

R.rsp

- Free mixing of languages and loops risks reading like PHP
- Error messages not helpful
- No editor support
- (Before R 4.1: %>% clash)

DOI 10.5281/zenodo.15186270

What Do We Need?—Call to Code

Grammar of tables

- Vocabulary for assembling common tables
  - Cell structure, *e. g.*, merged cells, concatenation, sparklines, subtables
  - Content structure, *e. g.*, hierarchical subtotals and automatic markup
  - Formatting, *e. g.*, vary number formatting, background coloring, foreground coloring, line width and style, font, based on calculation
- Operator notation to assemble things

Readable literate-programming controls

- knitr/Quarto's chunks not flexible enough
- R.rsp gets hard to read/debug
- What about knitr with limited loops/conditionals for blocks of chunks?
- Need good editor support

Formatting abstraction

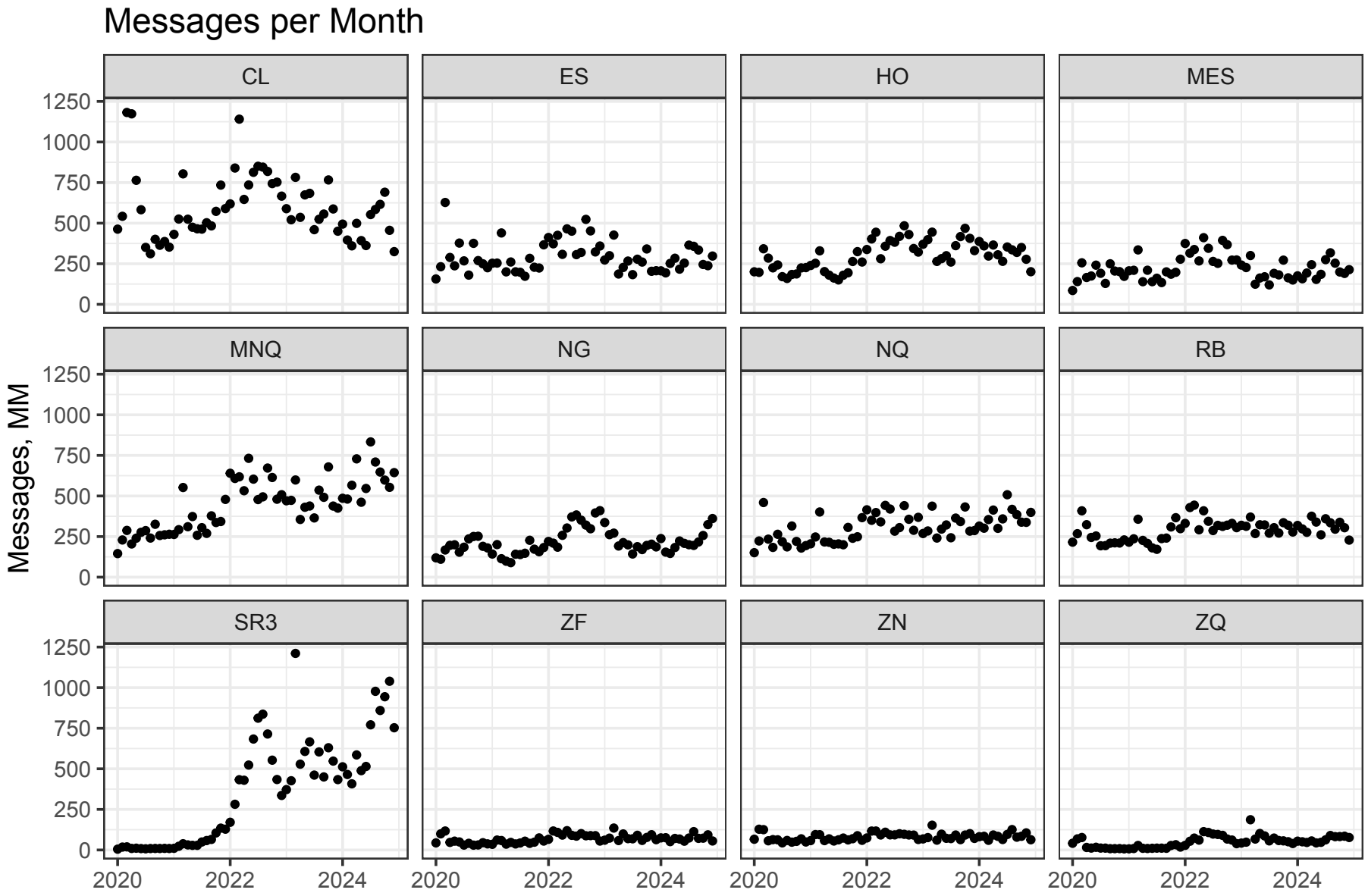
- Remove need for mixing HTML, LaTeX manually into kableExtra
- Remove need for mixing HTML, LaTeX manually into Quarto

DOI 10.5281/zenodo.15186270

Example: Globex Futures Order Flow

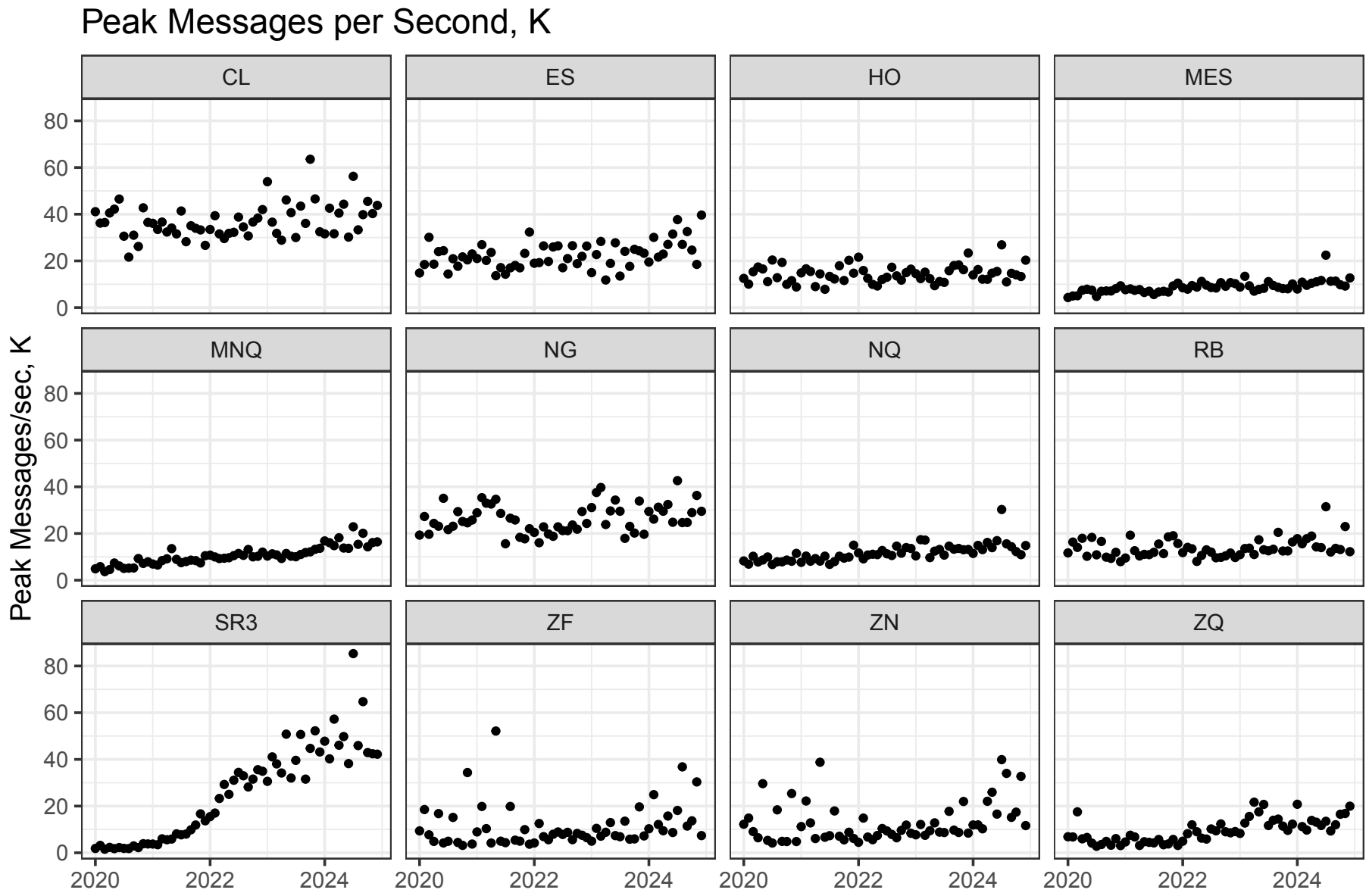
DOI 10.5281/zenodo.15186270

Monthly Message Volumes



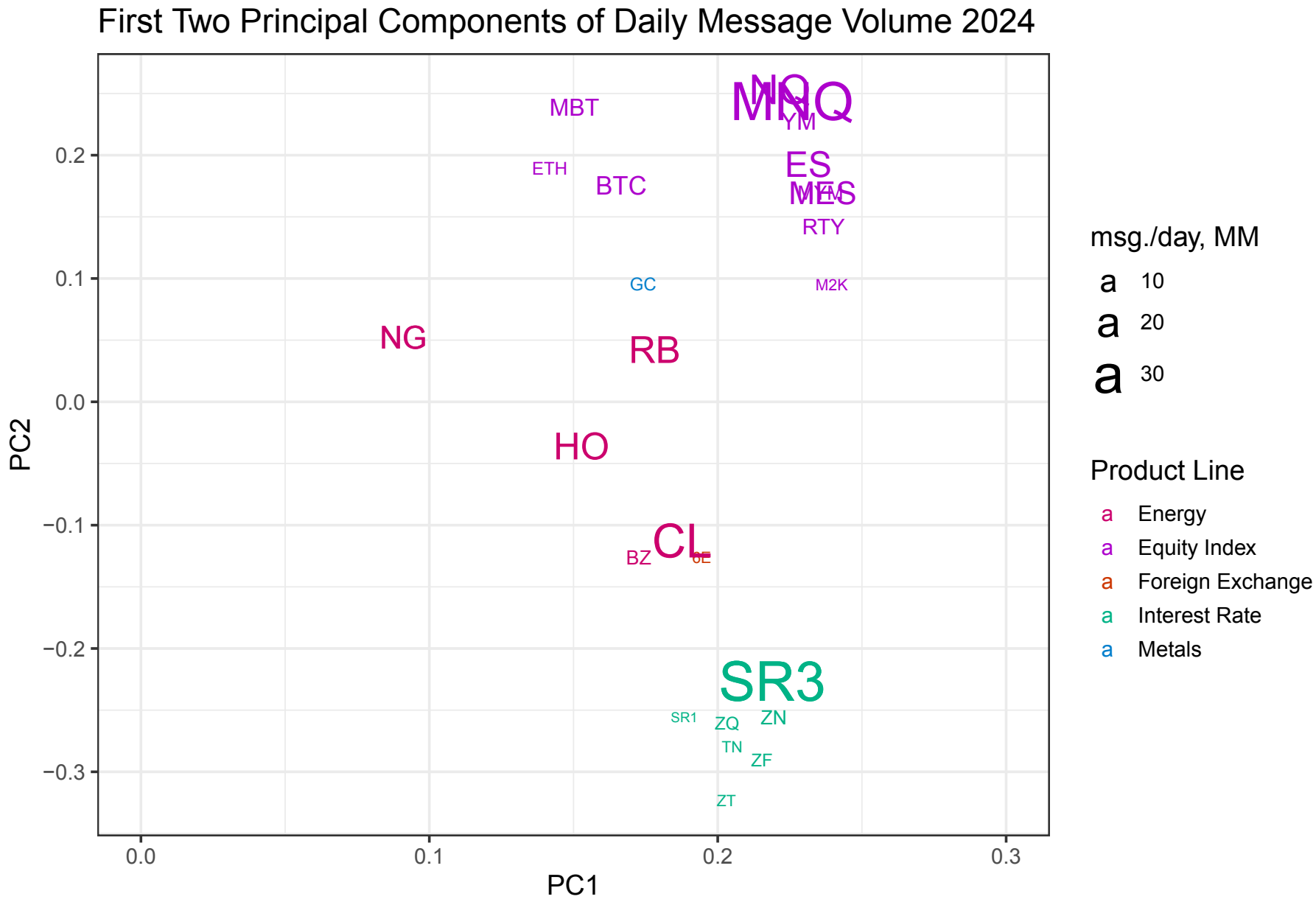
DOI 10.5281/zenodo.15186270

Peak Messages per Second



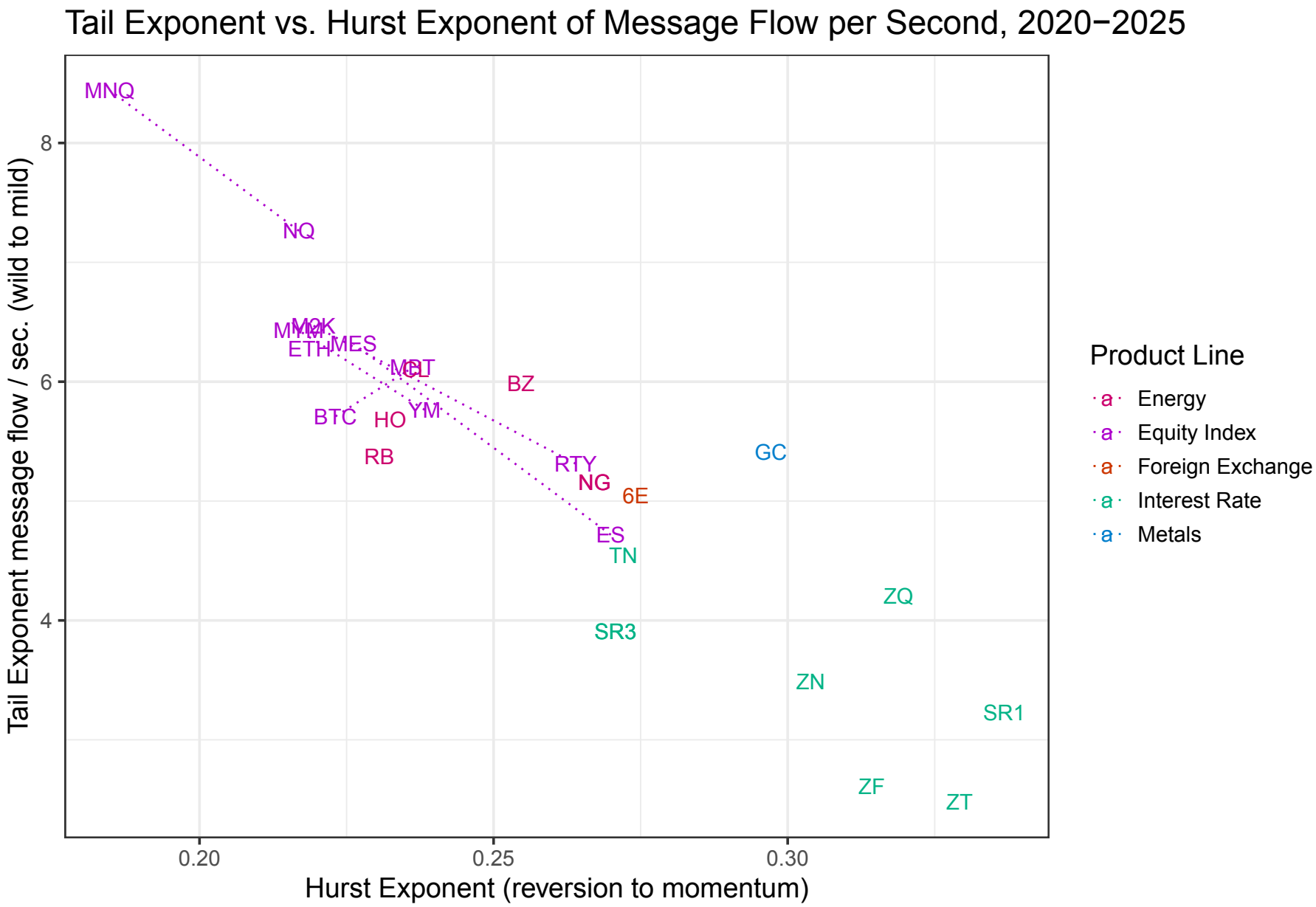
DOI 10.5281/zenodo.15186270

Messages per Day, Principal Components



DOI 10.5281/zenodo.15186270

Fractal Behavior of Message Flow



This is from going through 274,191,273,315 messages in our database…

DOI 10.5281/zenodo.15186270