What is the best data file format for your project?

Schwinn

Disclaimer

This work expresses my views and not necessarily those of the SEC, the Commissioners, or other members of the staff.

Who Cares?

- You should!
- Your users will!

OK, So Which One Is the Best?



What Matters?

I measure:

- Reading Speed
- Writing Speed
- File Size
- Accuracy

Other considerations:

- Interoperability with other software
- Officially mandated use (SAS transfer format, XPT)
- Will others be connected to the data source at the same time?

System Characteristics

- CPU Cores
- RAM
- Storage

Data Characteristics

Size (KB, MB, GB, TB)

What does it contain?

- Does it contain exotic data types like geometric data (e.g. shape files)
- Long strings
- Special characters
- Special Attributes and R-specific data

Format Classes

- Flat Files (CSV, XLSX, JSON, etc.)
- Compressed Files (RDS, Parquet, QS, etc.)
- Database on Disk Files (DuckDB, RSQL)

Flat Files

Format Name	Multi-threaded	Long Strings	Complex Vectors	On disk row access	R Attributes	Nested Lists
ARFF (ASCII)	✗	✗	✗	✗	✗	✗
CSV (Arrow)	✓	✗	✗	✓	✗	✗
CSV (Base R)	✗	✗	✗	✓	✗	✗
CSV (Fwrite)	✓	✓	✗	✓	✗	✗
CSV (readr)	✓	✗	✗	✓	✗	✗
CSV (Vroom)	✓	✗	✗	✓	✗	✗
DBF (dBase)	✗	✗	✗	✗	✗	✗
JSON (rjson)	✗	✗	✗	✗	✗	✗
ODS (OpenDocument)	✗	✗	✗	✗	✗	✗
SAV (SPSS)	✗	✗	✗	✗	✗	✗
XLSX (Open.Xlsx)	✗	✗	✗	✗	✗	✗
XLSX (writexl)	✗	✗	✗	✗	✗	✗

Compressed Files

Format Name	Multi-threaded	Long Strings	Complex Vectors	On disk row access	R Attributes	Nested Lists
DTA (Stata)	✗	✗	✗	✗	✗	✗
FEATHER (Arrow)	✓	✓	✗	✗	✗	✗
FEATHER (Feather)	✓	✓	✗	✗	✗	✗
FST (R, Python)	✓	✓	✗	✓	✗	✗
H5 (HD5DB)	✗	✗	✗	✗	✗	✗
PARQUET (Arrow)	✓	✓	✗	✓	✗	✗
QData (R, Python)	✓	✓	✓	✗	✓	✓
QS (R)	✓	✓	✓	✗	✓	✓
QS2 (R)	✓	✓	✓	✗	✓	✓
RDS (Base R)	✗	✓	✓	✗	✓	✓
RDS (readr)	✗	✓	✓	✗	✓	✓
XPT (SAS)	✗	✗	✗	✗	✗	✗

Database on Disk Files

Format Name	Multi-threaded	Long Strings	Complex Vectors	On disk row access	R Attributes	Nested Lists
DUCKDB	✓	✓	✗	✓	✗	✗
RSQLITE	✓	✓	✗	✓	✗	✗

Demonstration

The benchmarking tool consists of a single 22KB app.R file. Using it, you can benchmark these file formats with any data, on any shiny compatible system, to find the optimal format for you.

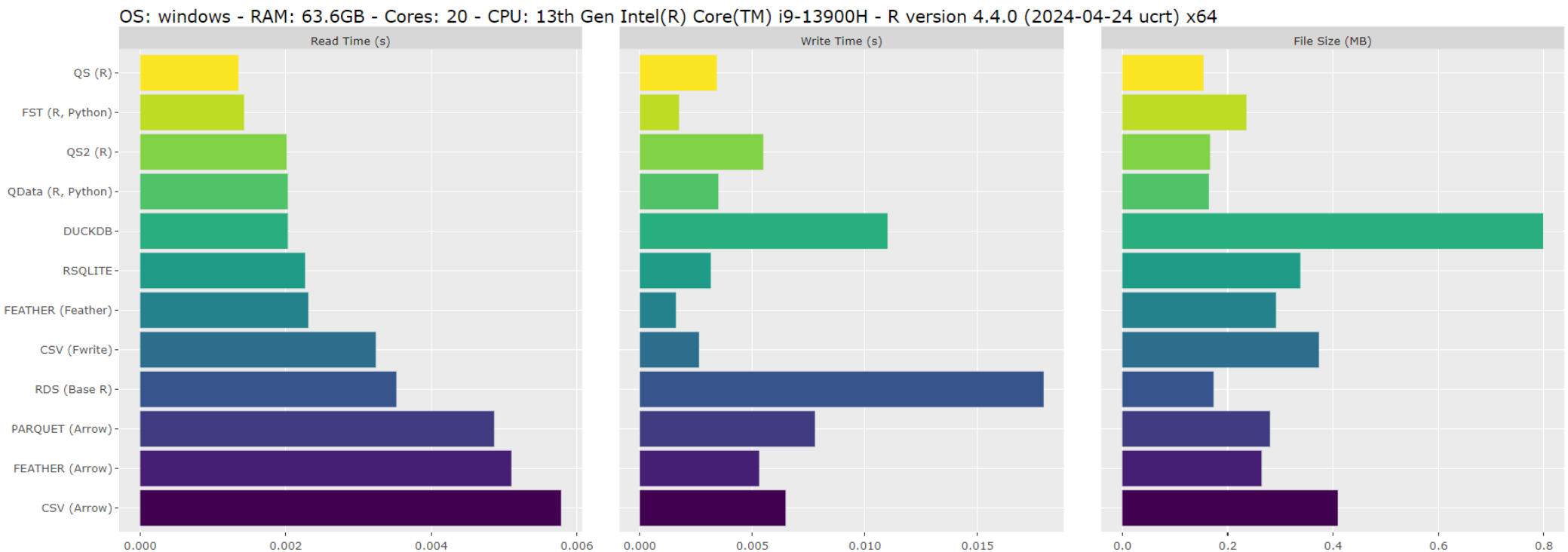
Findings: Small Files

Show 10 entries Search:

format_name	Read Time (s)	Write Time (s)	File Size (MB)
QS (R)	0.0013	0.0034	0.1538
FST (R, Python)	0.0014	0.0017	0.2354
QS2 (R)	0.0020	0.0055	0.1664
QData (R, Python)	0.0020	0.0035	0.1642
DUCKDB	0.0020	0.0110	0.7987
RSQLITE	0.0023	0.0032	0.3378
FEATHER (Feather)	0.0023	0.0016	0.2916
CSV (Fwrite)	0.0032	0.0026	0.3731
RDS (Base R)	0.0035	0.0179	0.1731
PARQUET (Arrow)	0.0049	0.0078	0.2802

Showing 1 to 10 of 12 entries

Findings: Small Files



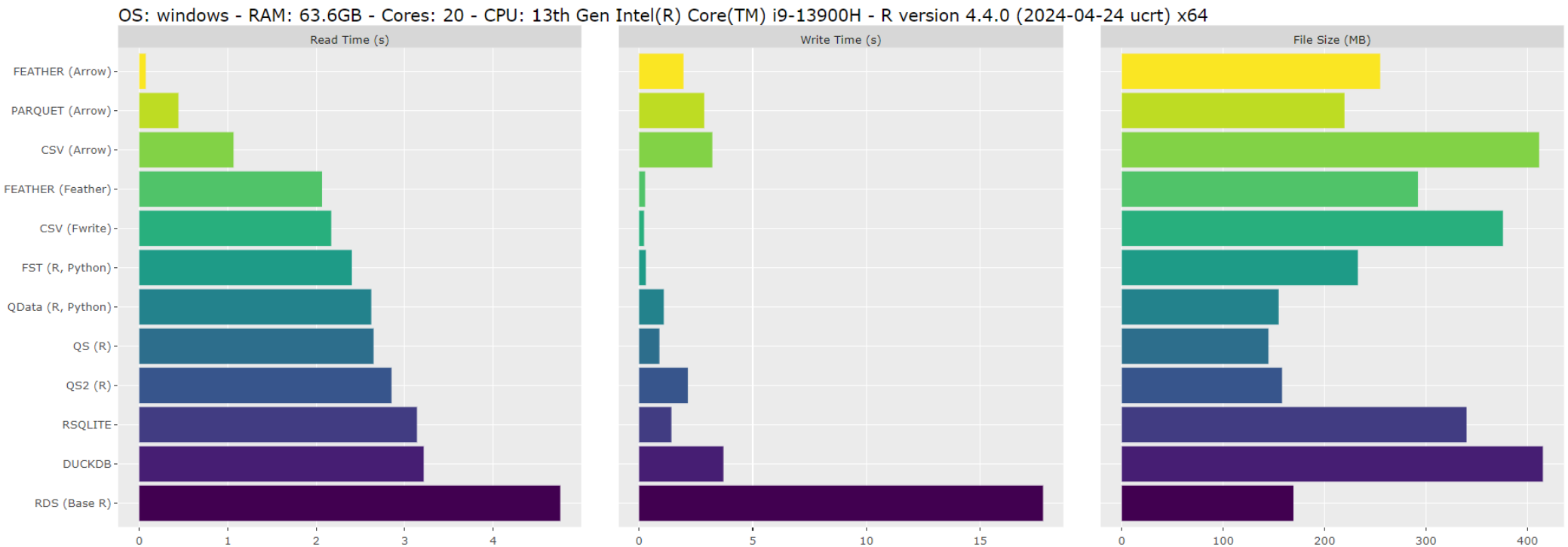
Findings: Large Files

Show 10 entries Search:

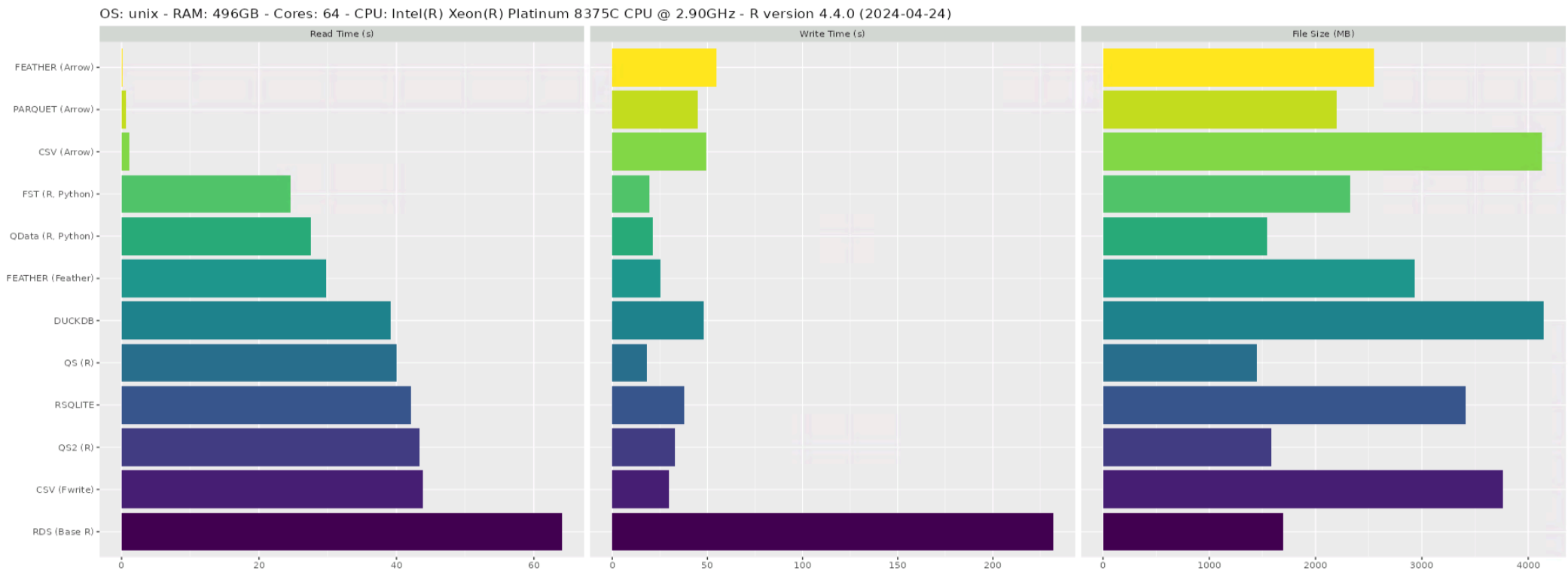
format_name	Read Time (s)	Write Time (s)	File Size (MB)
FEATHER (Arrow)	0.07	1.95	254.79
PARQUET (Arrow)	0.44	2.87	219.63
CSV (Arrow)	1.07	3.22	411.59
FEATHER (Feather)	2.07	0.26	292.00
CSV (Fwrite)	2.17	0.22	375.89
FST (R, Python)	2.40	0.30	232.80
QData (R, Python)	2.62	1.09	154.71
QS (R)	2.65	0.89	144.53
QS2 (R)	2.85	2.14	158.00
RSQLITE	3.14	1.42	340.00

Showing 1 to 10 of 12 entries Previous 1 2 Next

Findings: Large Files



Findings: Super Large Files



Findings: Super Large Files

Show 10 entries Search:

format_name	Read Time (s)	Write Time (s)	File Size (MB)
RSQLITE	50.52	13.00	3,410.00
CSV (Arrow)	34.53	47.08	4,125.89
FEATHER (Arrow)	4.75	34.46	2,547.87
PARQUET (Arrow)	8.63	43.45	2,196.62
RDS (Base R)	66.81	199.98	1,692.51
DUCKDB	42.59	41.50	4,145.82
FEATHER (Feather)	74.85	5.01	2,930.00
FST (R, Python)	54.57	3.13	2,328.01
CSV (Fwrite)	55.22	4.08	3,768.90
QData (R, Python)	46.99	10.42	1,546.86

Showing 1 to 10 of 12 entries

Results: Super Large Files

```
> file.info("~/Desktop/tests/test_arrow_parquet.parquet")$size
[1] 2196759451
> start_time = Sys.time()
> test_df = arrow::read_parquet("~/Desktop/tests/test_arrow_parquet.parquet")
> Sys.time() - start_time
Time difference of 0.594887 secs
> nrow(test_df)
[1] 10000000
```

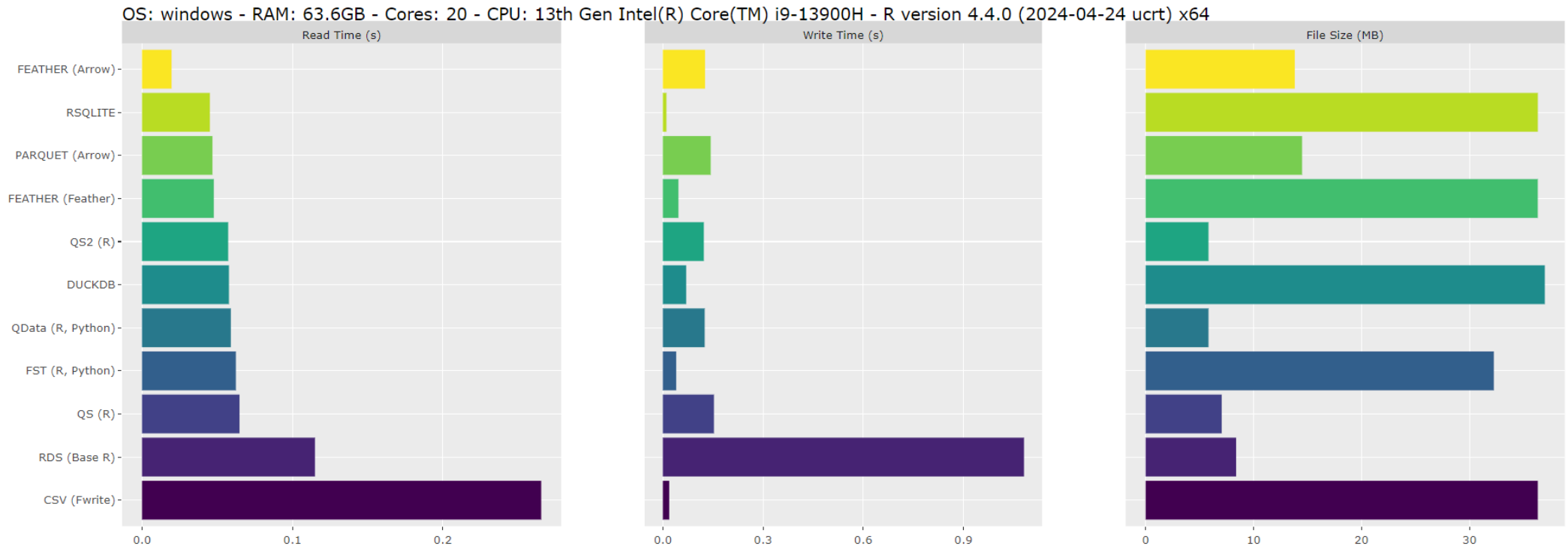
Findings: Long Strings

Show 10 entries Search:

format_name	Read Time (s)	Write Time (s)	File Size (MB)
RSQLITE	0.05	0.01	36.33
FEATHER (Arrow)	0.02	0.13	13.81
PARQUET (Arrow)	0.05	0.14	14.49
RDS (Base R)	0.12	1.08	8.38
DUCKDB	0.06	0.07	36.97
FEATHER (Feather)	0.05	0.05	36.32
FST (R, Python)	0.06	0.04	32.25
CSV (Fwrite)	0.27	0.02	36.32
QData (R, Python)	0.06	0.13	5.82
QS (R)	0.06	0.15	7.05

Showing 1 to 10 of 11 entries Previous 1 2 Next

Findings: Long Strings



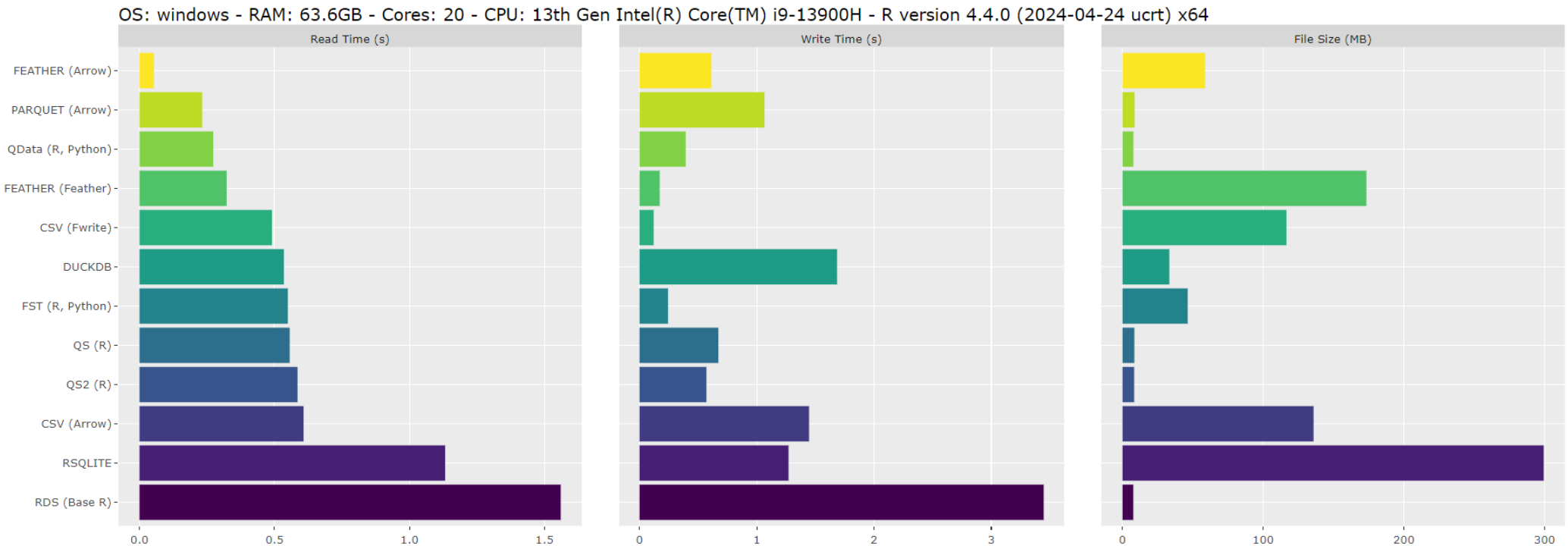
Findings: Loan Data

Show 10 entries Search:

format_name	Read Time (s)	Write Time (s)	File Size (MB)
FEATHER (Arrow)	0.05	0.61	58.65
PARQUET (Arrow)	0.23	1.07	8.57
QData (R, Python)	0.27	0.39	7.75
FEATHER (Feather)	0.32	0.17	173.37
CSV (Fwrite)	0.49	0.12	116.56
DUCKDB	0.53	1.68	33.30
FST (R, Python)	0.55	0.24	46.33
QS (R)	0.56	0.67	8.42
QS2 (R)	0.59	0.57	8.28
CSV (Arrow)	0.61	1.45	135.84

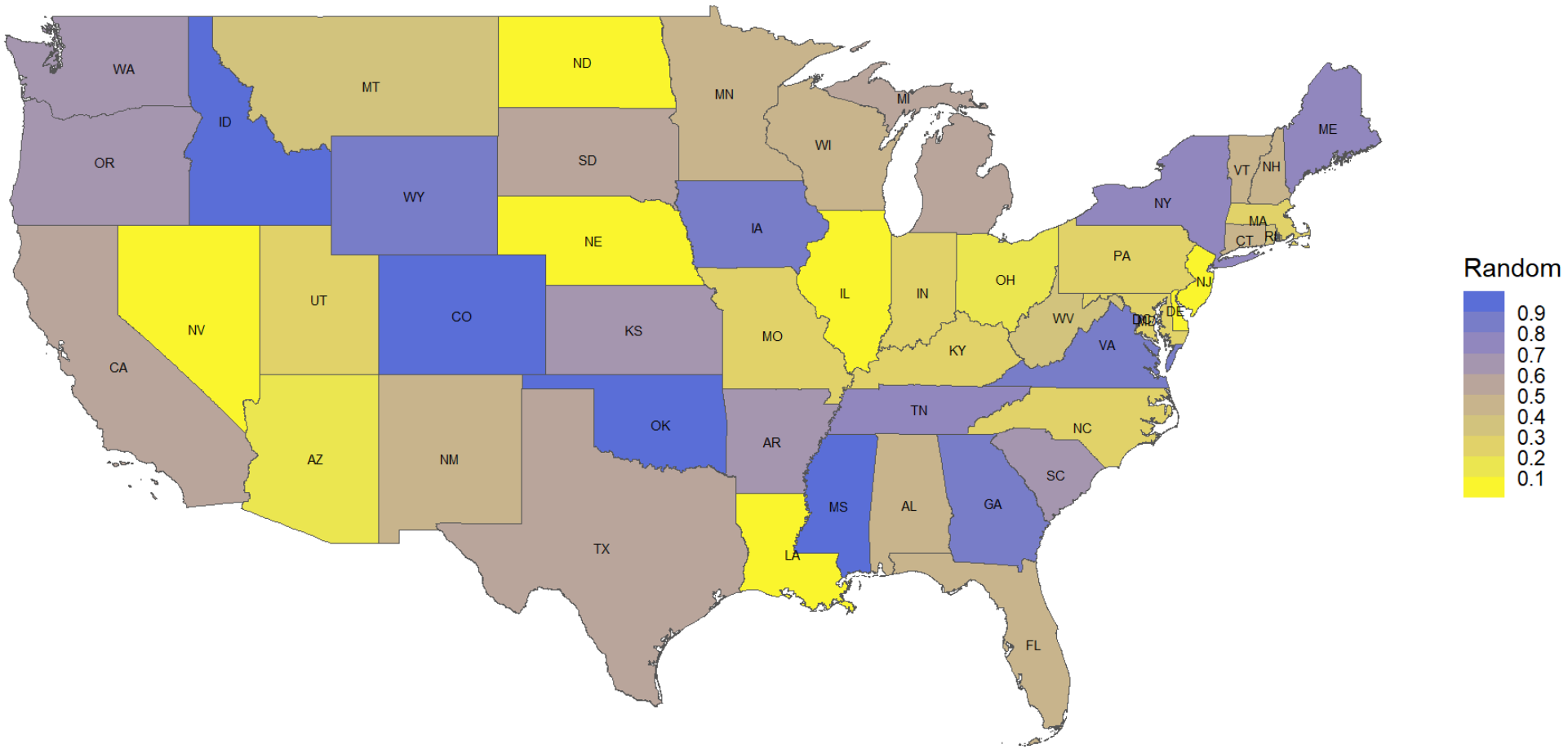
Showing 1 to 10 of 12 entries

Findings: Loan Data



Findings: Geometry Data

Random Uniform [0,1] by State



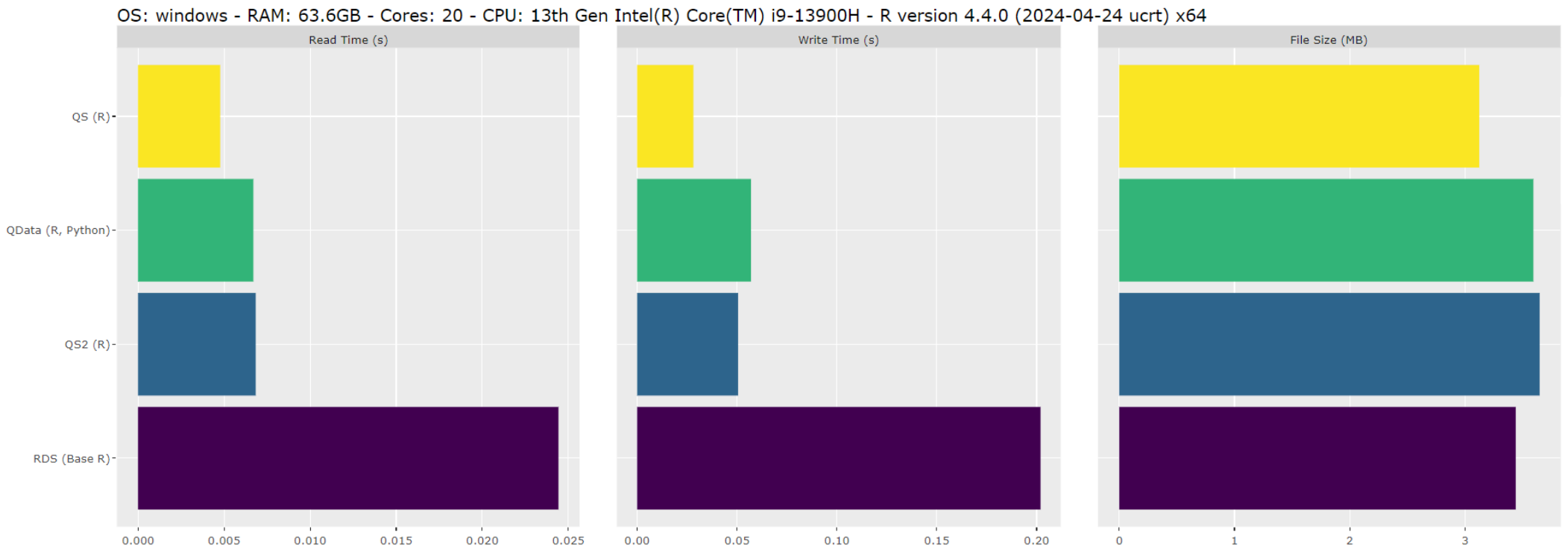
Findings: Geometry Data

Show 10 entries Search:

format_name	Read Time (s)	Write Time (s)	File Size (MB)
RDS (Base R)	0.02	0.20	3.44
QData (R, Python)	0.01	0.06	3.59
QS (R)	0.00	0.03	3.12
QS2 (R)	0.01	0.05	3.65

Showing 1 to 4 of 4 entries Previous 1 Next

Findings: Geometry Data



Summary

- What format is best depends on your data and your system.
- QS is a worthy replacement for the RDS format.
- Arrow is super fast on powerful machines!
- Don’t take my word for it.
- Feel free to reach out with questions or feedback to schwinnr@sec.gov

  Thank you so much for your attention!