

An Evaluation of Data Formats for Clinical Reporting in a Multilingual Environment

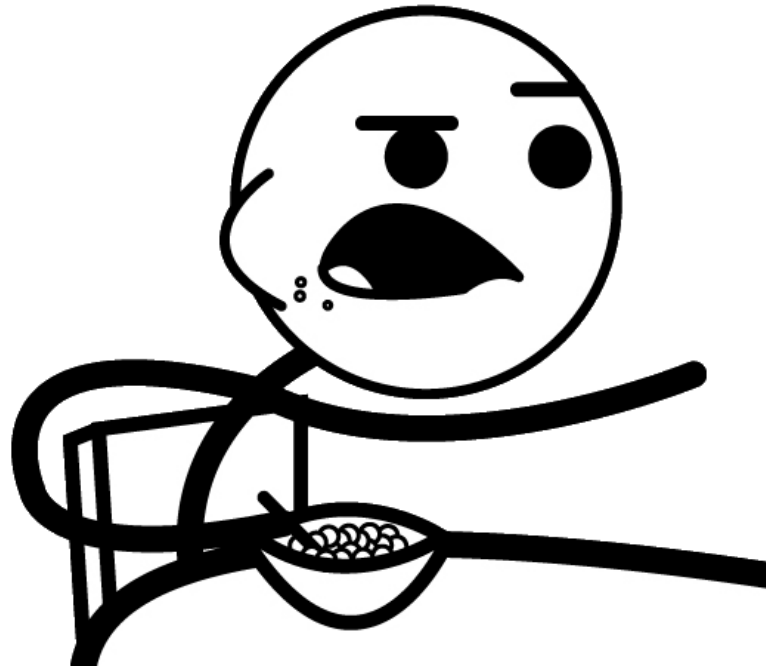
Craig Gower-Page
November 18th, 2025

Introduction

Problem Statement

How can clinical trial data be formatted and stored to ensure interoperability across programming languages without sacrificing features or performance?

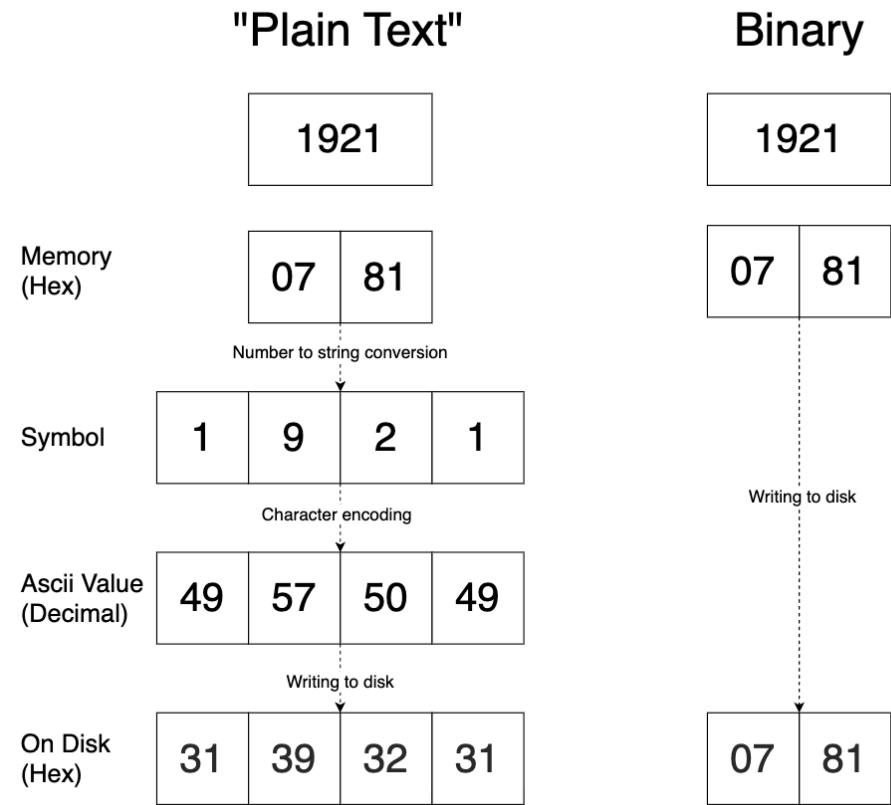




Just use a database....

Terminology and Key Concepts

Plain Text vs Binary



Column vs Row-based Storage

Dataset

A1	B1	C1
A2	B2	C2

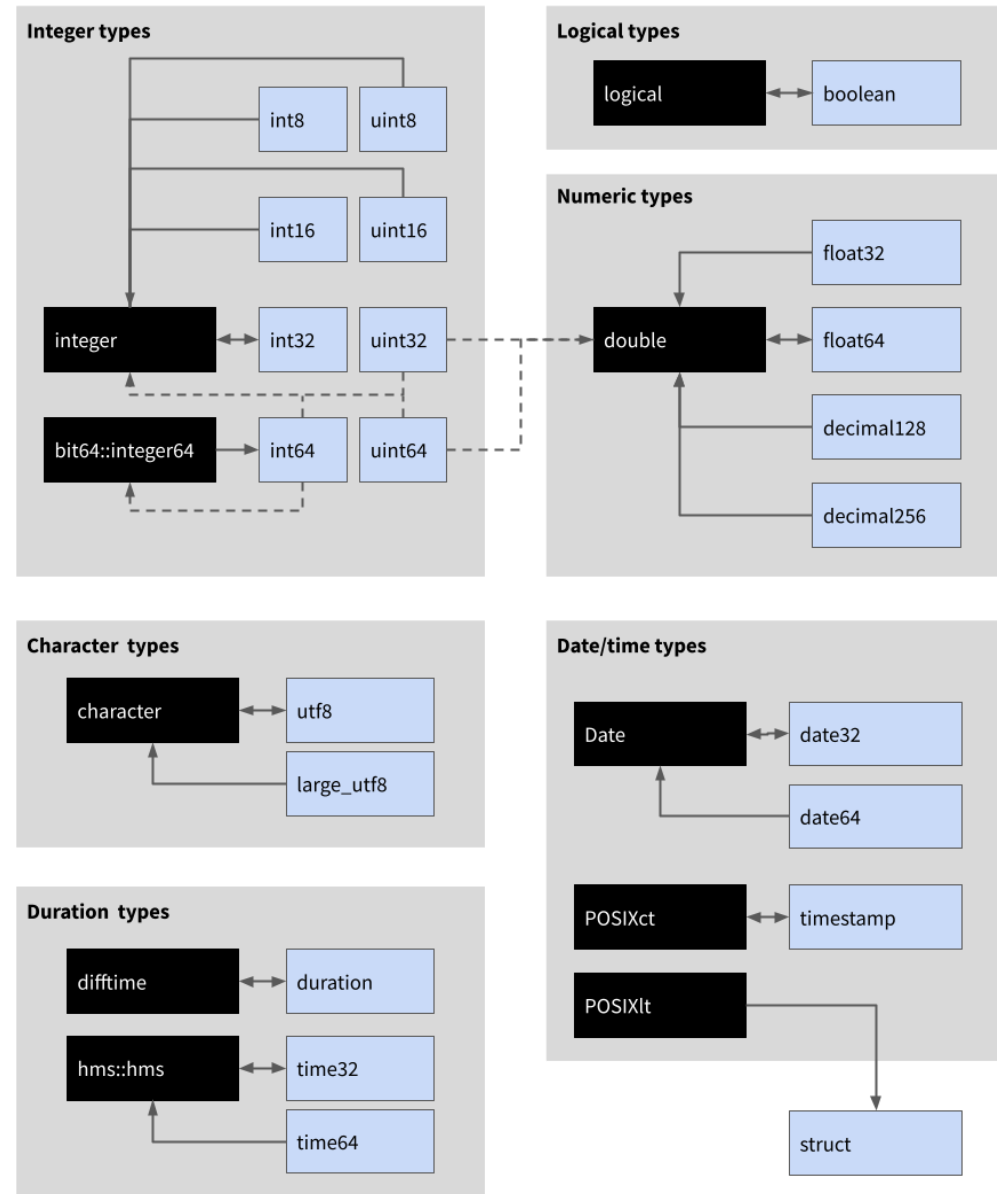
Column-Based Storage

A1	A2	B1	B2	C1	C2
----	----	----	----	----	----

Row-Based Storage

A1	B1	C1	A2	B2	C2
----	----	----	----	----	----

Data Types



Format Evaluation

Formats Considered

Plain Text

JSON
CSV

Binary

Apache Parquet
Feather (Arrow-IPC)
Apache Orc
Apache Avro
SAS XPT

“Database”

SQLite
DuckDB

Other

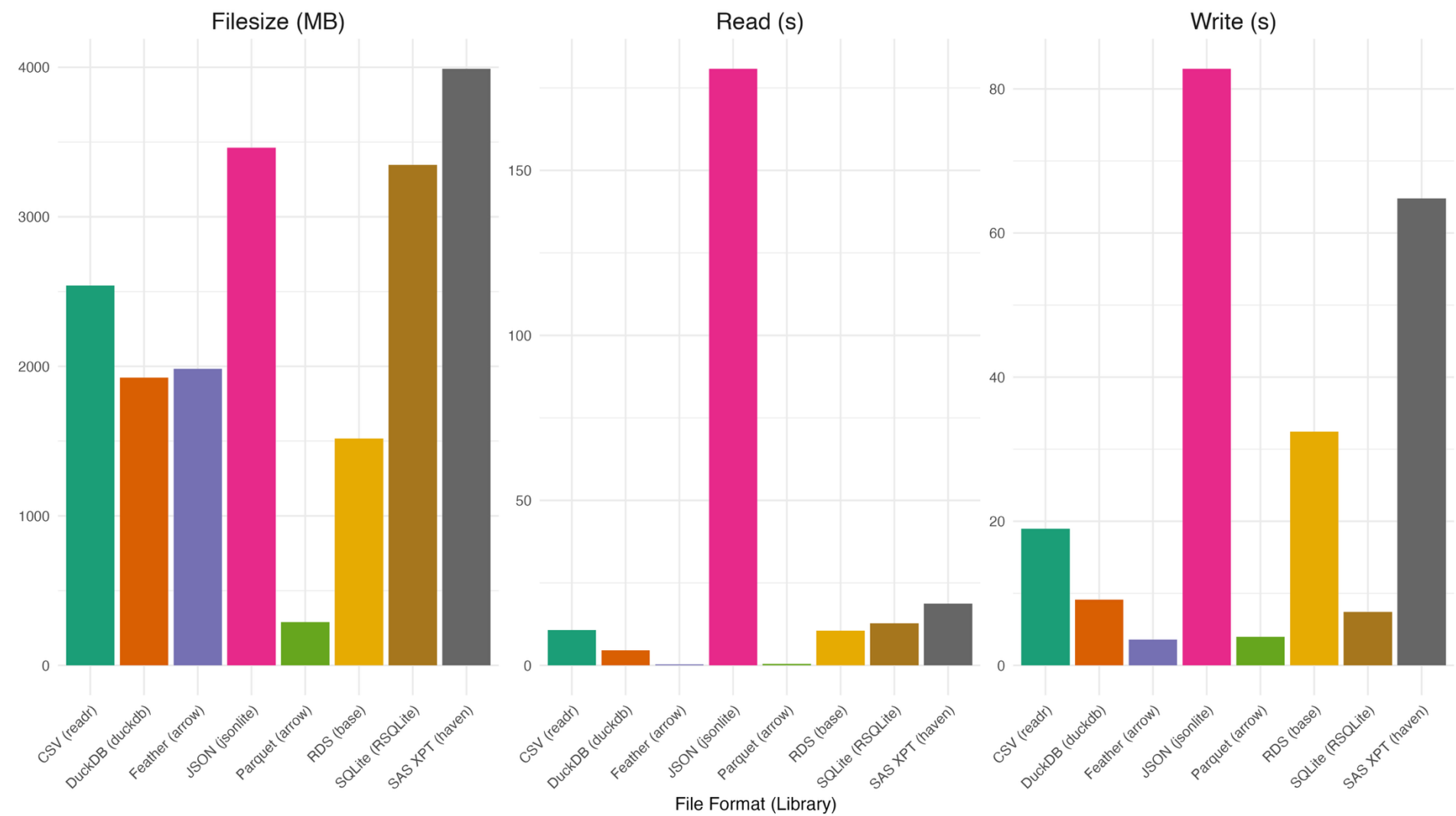
HDF5
Delta Lake
Apache Iceberg
Protocol Buffers

Evaluation Criteria

- 1 Library Availability
- 2 Data type coverage
- 3 I/O Performance
- 4 Integrations
- 5 File Sizes



Benchmarks



Format Spotlights – SAS XPT

1	Library Availability	
2	Data type coverage	
3	I/O Performance	
4	Integrations	
5	File Sizes	

```
haven::write_xpt(  
  ... ads1,  
  ... "data/ads1.xpt"  
)  
  
ads1 <- haven::read_xpt(  
  ... "data/ads1.xpt"  
)
```

Format Spotlights - JSON

1	Library Availability	
2	Data type coverage	
3	I/O Performance	
4	Integrations	
5	File Sizes	

```
jsonlite::write_json(  
  ... adsl,  
  ... "data/adsl.json"  
)
```

```
adsl <- jsonlite::read_json(  
  ... "data/adsl.json"  
)
```

Format Spotlights – Apache Parquet

1	Library Availability	
2	Data type coverage	
3	I/O Performance	
4	Integrations	
5	File Sizes	

```
arrow::write_parquet(  
  ... ads1,  
  ... "data/ads1.parquet"  
)
```

```
ads1 <- arrow::read_parquet(  
  ... "data/ads1.parquet"  
)
```

Format Spotlights - DuckDB

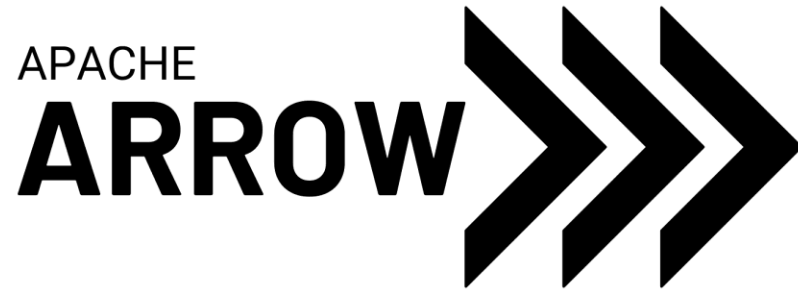
1	Library Availability	
2	Data type coverage	
3	I/O Performance	
4	Integrations	
5	File Sizes	

```
con <- DBI::dbConnect(  
  ... duckdb::duckdb(),  
  ... "data/ads1.duckdb"  
)  
DBI::dbWriteTable(  
  ... con, "ads1",  
  ... ads1, overwrite = TRUE  
)  
DBI::dbDisconnect(con)
```

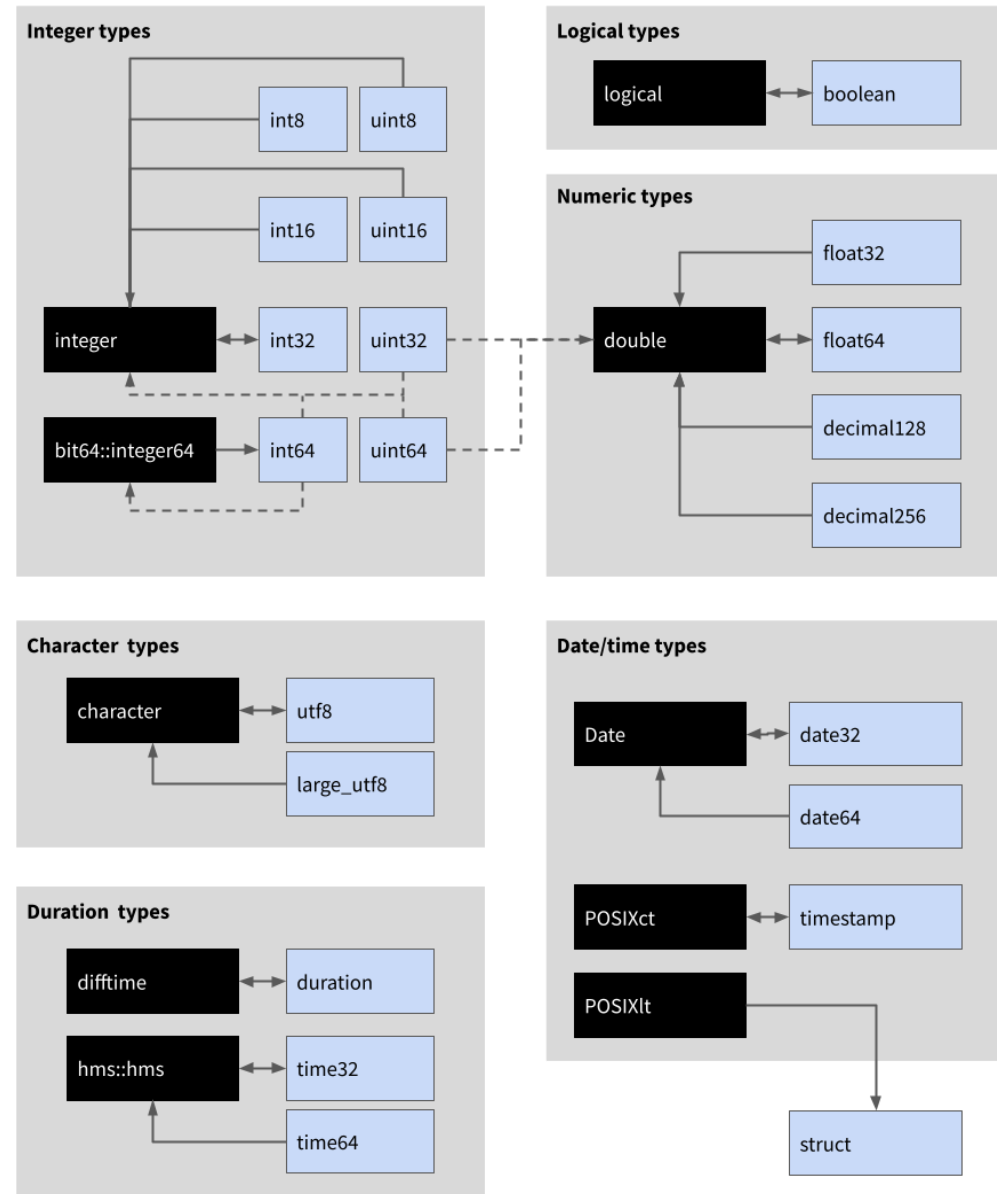
```
con <- DBI::dbConnect(  
  ... duckdb::duckdb(),  
  ... "data/ads1.duckdb"  
)  
ads1 <- DBI::dbReadTable(con, "ads1")  
DBI::dbDisconnect(con)
```

Apache Parquet

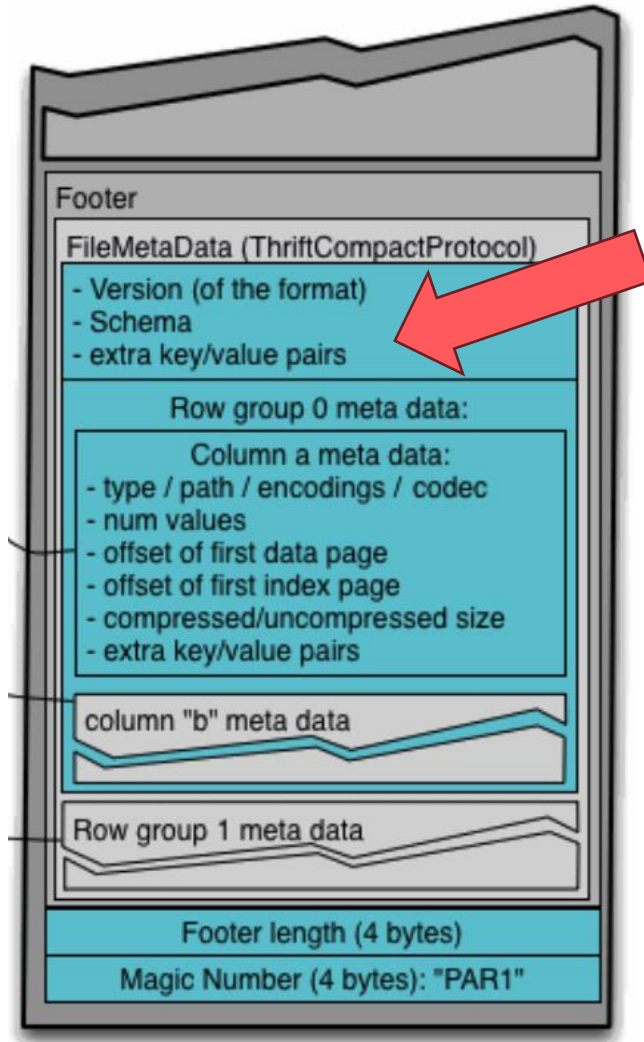
Apache Arrow vs Apache Parquet



Data Types



Metadata



```
library(arrow)
dat <- data.frame(
  x = c(1, 2, 3),
  y = c(4, 5, 6)
)
attr(dat$x, "label") <- "X Variable Label"
attr(dat$y, "label") <- "Y Variable Label"
arrow::write_parquet(dat, "data/labels.parquet")
```

```
> code/print_metadata data/labels.parquet
```

```
{b'r':
b'A\n3\n263425\n197888\n5\nUTF-8\n531\n1\n531\n2\n531\n2\n531\n1\n16\n1\n
262153\n16\nX\\040Variable\\040Label\n1026\n1\n262153\n5\nnames\n16\n1\n2
62153\n5\nlabel\n254\n254\n1026\n511\n16\n2\n262153\n10\nattributes\n2621
53\n7\ncolumns\n254\n531\n2\n531\n1\n16\n1\n262153\n16\nY\\040Variable\\0
40Label\n1026\n511\n16\n1\n262153\n5\nlabel\n254\n254\n1026\n511\n16\n2\n
262153\n10\nattributes\n262153\n7\ncolumns\n254\n1026\n511\n16\n2\n262153
\n1\nx\n262153\n1\ny\n254\n1026\n511\n16\n1\n262153\n7\ncolumns\n254\n'}
```

SAS Integration

```
83      libname pqlib PARQUET "/domino/datasets/local/test-flows";
NOTE: Libref PQLIB was successfully assigned as follows:
      Engine:          PARQUET
      Physical Name: /domino/datasets/local/test-flows
84      data work.example;
85          set pqlib.example;
86      run;
NOTE: There were 10000 observations read from the data set PQLIB.example.
NOTE: The data set WORK.EXAMPLE has 10000 observations and 85 variables.
NOTE: DATA statement used (Total process time):
      real time          54.61 seconds
      cpu time           6.11 seconds

system.time({
|... arrow::read_parquet("/domino/datasets/local/test-flows")
|})
#... user... system elapsed
#... 0.015... 0.011... 0.005
```

- Characters read with default length of 42KB
- Not supported in 9.4 or earlier
- Output data types based on informats
- No metadata support
- Critical bugs in earlier versions

Craig Gower-Page

craig.gower-page@novartis.com

Thank you

