The `data_algebra` query system

John Mount
Win-Vector LLC
http://www.win-vector.com/

Outline

- What is `data_algebra`?
- Example
- Where we are
- Links

What is `data_algebra`?

- `data_algebra` is a package for building up complex data manipulation queries
- `data_algebra` queries are first class citizens in the Strachey sense (can be: passed as an argument, returned from a function, modified, assigned to a variable, printed, inspected, and traversed as a data structure)
- The operators are essentially those of the Codd relational algebra (select rows/columns, join, unionall, extend, project, and window functions).
- Composition is left to right using method chaining.
- Queries can be realized in `SQL` (targeting `PostgeSQL` and `Spark`) or in `Pandas` (hoping to look into `modin`, `RAPIDS`, and others).

Example

```
d = pandas.DataFrame({
    'g': ['a', 'b', 'b', 'c', 'c', 'c'],
    'x': [1, 4, 5, 7, 8, 9],
    'v': [10, 40, 50, 70, 80, 90],
})
```

```
table_description = describe_table(d)

ops = table_description. \
    extend({
        'row_number': '_row_number()',
        'shift_v': 'v.shift()',
        'cumsum_v': 'v.cumsum()',
    },
    order_by=['x'],
    partition_by=['g']). \
    extend({
        'ngroup': '_ngroup()',
        'size': '_size()',
        'max_v': 'v.max()',
        'min_v': 'v.min()',
        'sum_v': 'v.sum()',
        'mean_v': 'v.mean()',
    },
    partition_by=['g'])

res1 = ops.transform(d)
```

res

g	x	v	row_number	shift_v	cumsum_v	ngroup	size	max_v	min_v	sum_v	mean_v
a	1	10	1	NaN	10	0	1	10	10	10	10
b	4	40	1	NaN	40	1	2	50	40	90	45
b	5	50	2	40.0	90	1	2	50	40	90	45
c	7	70	1	NaN	70	2	3	90	70	240	80
c	8	80	2	70.0	150	2	3	90	70	240	80
c	9	90	3	80.0	240	2	3	90	70	240	80

https://github.com/WinVector/data_algebra/blob/master/Examples/WindowFunctions/WindowFunctions.md

Where we are

- Code works well for our internal projects.
  - We use `data_algebra` and its `R` siblings `rquery`/`rqdatatable` (based on `SQL` and `data.table`) for our internal projects.
  - Looking to expand to `Google BigQuery`, `modin`, and `RAPIDS`.
- Working to expand on the observation that `data_algebra` forms a really nifty category over table schemas.

Links

- Github: https://github.com/WinVector/data_algebra
- Introduction: http://www.win-vector.com/blog/2019/08/introducing-data_algebra/
- Category theory note: https://github.com/WinVector/data_algebra/blob/master/Examples/Arrow/Arrow.md

Thank You