xeus-sql

Mariana Meireles

Dec 29, 2020

GETTING STARTED

`xeus-sql` is a Jupyter kernel for general SQL implementations based on the native implementation of the Jupyter protocol xeus and SOCI, a database access library for C++.

ONE

LICENSING

We use a shared copyright model that enables all contributors to maintain the copyright on their contributions.

This software is licensed under the BSD-3-Clause license. See the LICENSE file for details.

1.1 Installation

1.1.1 With Conda or Mamba

`xeus-sql` has been packaged for the conda package manager.

To ensure that the installation works, it is preferable to install `xeus-sql` in a fresh conda/mamba environment. It is also needed to use a miniconda installation because with the full anaconda you may have a conflict with the `zeromq` library which is already installed in the anaconda distribution.

The safest usage is to create an environment named `xeus-sql` with your miniconda installation

```
conda create -n xeus-sql
conda activate xeus-sql # Or `source activate xeus-sql` for conda < 4.6
```

```
mamba create -n xeus-sql
mamba activate xeus-sql
```

Then you can install in this freshly created environment `xeus-sql` and its dependencies

```
conda install xeus-sql notebook -c conda-forge
```

```
mamba install xeus-sql notebook -c conda-forge
```

or, if you prefer to use JupyterLab

```
conda install xeus-sql jupyterlab -c conda-forge
```

```
mamba install xeus-sql jupyterlab -c conda-forge
```

Conda forge offers packaged versions for MySQL, PostgreSQL and SQLite and you can download them with: *soci-mysql*, *soci-postresql* or *soci-sqlite*.

`xeus-sql` includes `soci-core` only. Which consists on the SOCI package with no DB extension attached.

1.1.2 From Source

You can install `xeus-sql` from source with cmake. This requires that you have all the dependencies installed in the same prefix.

```
conda install cmake nlohmann_json xtl cppzmq xeus mysql sqlite postgresql cpp-
↪tabulate=1.3 xvega xvega-bindings xproperty jupyterlab -c conda-forge
```

```
mamba install cmake nlohmann_json xtl cppzmq xeus mysql sqlite postgresql cpp-
↪tabulate=1.3 xvega xvega-bindings xproperty jupyterlab -c conda-forge
```

```
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=/path/to/prefix ..
make install
```

On Windows platforms, from the source directory:

```
mkdir build
cd build
cmake -G "NMake Makefiles" -DCMAKE_INSTALL_PREFIX=/path/to/prefix ..
nmake
nmake install
```

1.2 SQL magics

Magics that allow you to operate on the database.

1.2.1 LOAD

%LOAD database_type name_of_database

To see how to use this command in depth, please refer to the specific page of the database.

1.3 XVega magics

Magics that allow you to create graph visualizations using XVega an implementation of vega-light to C++.

1.3.1 X_FIELD

%X_FIELD name_of_column
   Represents the X axis of the graph. The name of the axis should be the same as the name of the SQLite column
   (or result of SQLite query).

1.3.1.1 TYPE

%TYPE type_of_field
 Sub-attribute of **X_FIELD**.

> Bellow there's list of the types supported by *xeus-sqlite*. If you want to learn more about types please refer to vega lite type official documentation.

- QUANTITATIVE
- NOMINAL
- ORDINAL
- TEMPORAL

1.3.1.2 BIN

%BIN type_of_field
 Sub-attribute of **X_FIELD**.

 Binning discretizes numeric values into a set of bins. If bin is true, default binning parameters are used.

 To customize binning parameters, you can set bin to a bin definition object, which can have the following properties:

 If you want to learn more about bin please refer to vega lite bin official documentation.

ANCHOR

%ANCHOR bin_position
 Sub-attribute of **BIN**.

 A value in the binned domain at which to anchor the bins, shifting the bin boundaries if necessary to ensure that a boundary aligns with the anchor value.

BASE

%BASE number
 Sub-attribute of **BIN**.

 The number base to use for automatic bin determination (default is base 10).

BINNED

%BASE boolean
 Sub-attribute of **BIN**.

MAXBINS

`%MAXBINS number`
> Sub-attribute of **BIN**.

> Maximum number of bins.

MINSTEP

`%MINSTEP number`
> Sub-attribute of **BIN**.

> A minimum allowable step size (particularly useful for integer values).

NICE

`%NICE bool`
> Sub-attribute of **BIN**.

> If true, attempts to make the bin boundaries use human-friendly boundaries, such as multiples of ten.

STEP

`%STEP number`
> Sub-attribute of **BIN**.

> An exact step size to use between bins.

1.3.1.3 AGGREGATE

`%AGGREGATE type_of_aggregation`
> Sub-attribute of **X_FIELD**.

> The aggregate property of a field definition can be used to compute aggregate summary statistics (e.g., median, min, max) over groups of data.

> Bellow there's list of the aggregations supported by *xeus-sqlite*. If you want to learn more about aggregations please refer to vega lite aggregate official documentation.

> - COUNT
> - VALID
> - MISSING
> - DISTINCT
> - SUM
> - PRODUCT
> - MEAN
> - AVERAGE
> - VARIANCE
> - VARIANCEP

- STDEV

- STEDEVP

- STEDERR

- MEDIAN

- Q1

- Q3

- CI0

- CI1

- MIN

- MAX

- ARGMIN

- ARGMAX

1.3.1.4 TIME_UNIT

%TIME_UNIT time
    Sub-attribute of **X_FIELD**.

Time unit is used to discretize time.

> Bellow there's list of the time units supported by *xeus-sqlite*. If you want to learn more about time units please refer to vega lite time unit official documentation.

- YEAR

- QUARTER

- MONTH

- DAY

- DATE

- HOURS

- MINUTES

- SECONDS

- MILISECONDS

1.3.2 Y_FIELD

%Y_FIELD name_of_column
    Represents the Y axis of the graph. The name of the axis should be the same as the name of the SQLite column (or result of SQLite query).

1.3.2.1 TYPE

`%TYPE type_of_field`
Sub-attribute of **Y_FIELD**.

Bellow there's list of the types supported by *xeus-sqlite*. If you want to learn more about types please refer to vega lite type official documentation.

- QUANTITATIVE

- NOMINAL

- ORDINAL

- TEMPORAL

1.3.2.2 BIN

`%BIN type_of_field`
Sub-attribute of **Y_FIELD**.

Binning discretizes numeric values into a set of bins. If bin is true, default binning parameters are used.

To customize binning parameters, you can set bin to a bin definition object, which can have the following properties:

If you want to learn more about bin please refer to vega lite bin official documentation.

ANCHOR

`%ANCHOR bin_position`
Sub-attribute of **BIN**.

A value in the binned domain at which to anchor the bins, shifting the bin boundaries if necessary to ensure that a boundary aligns with the anchor value.

BASE

`%BASE number`
Sub-attribute of **BIN**.

The number base to use for automatic bin determination (default is base 10).

BINNED

`%BASE boolean`
Sub-attribute of **BIN**.

MAXBINS

%MAXBINS number
> Sub-attribute of **BIN**.

> Maximum number of bins.

MINSTEP

%MINSTEP number
> Sub-attribute of **BIN**.

> A minimum allowable step size (particularly useful for integer values).

NICE

%NICE bool
> Sub-attribute of **BIN**.

> If true, attempts to make the bin boundaries use human-friendly boundaries, such as multiples of ten.

STEP

%STEP number
> Sub-attribute of **BIN**.

> An exact step size to use between bins.

1.3.2.3 AGGREGATE

%AGGREGATE type_of_aggregation
> Sub-attribute of **Y_FIELD**.

> The aggregate property of a field definition can be used to compute aggregate summary statistics (e.g., median, min, max) over groups of data.

> Bellow there's list of the aggregations supported by *xeus-sqlite*. If you want to learn more about aggregations please refer to vega lite aggregate official documentation.

> - COUNT
> - VALID
> - MISSING
> - DISTINCT
> - SUM
> - PRODUCT
> - MEAN
> - AVERAGE
> - VARIANCE
> - VARIANCEP

- STDEV

- STEDEVP

- STEDERR

- MEDIAN

- Q1

- Q3

- CI0

- CI1

- MIN

- MAX

- ARGMIN

- ARGMAX

1.3.2.4 TIME_UNIT

%TIME_UNIT time
      Sub-attribute of **Y_FIELD**.

      Time unit is used to discretize time.

            Bellow there's list of the time units supported by *xeus-sqlite*. If you want to learn more about time
            units please refer to vega lite time unit official documentation.

- YEAR

- QUARTER

- MONTH

- DAY

- DATE

- HOURS

- MINUTES

- SECONDS

- MILISECONDS

1.3.3 WIDTH

%WIDTH number
      Width of the graph in pixels.

1.3.4 HEIGHT

%HEIGHT number
    Height of the graph in pixels.

1.3.5 MARK

%MARK mark
    Marcs can be one of the following:

- ARC

- AREA

- BAR

- CIRCLE

- LINE

- POINT

- RECT

- RULE

- SQUARE

- TICK

- TRAIL

1.3.5.1 COLOR

%COLOR color
    Sub-attribute of **MARK**.

    Sets the color of a mark. The color can be one of the valid CSS color string.

1.3.6 GRID

%HEIGHT boolean
    Enable or disable grid view on graph.