agate-charts Documentation

Release 0.1.0 (alpha)

Christopher Groskopf

December 21, 2015

Contents

CHAPTER 1

About

agate-charts adds exploratory charting support to agate.

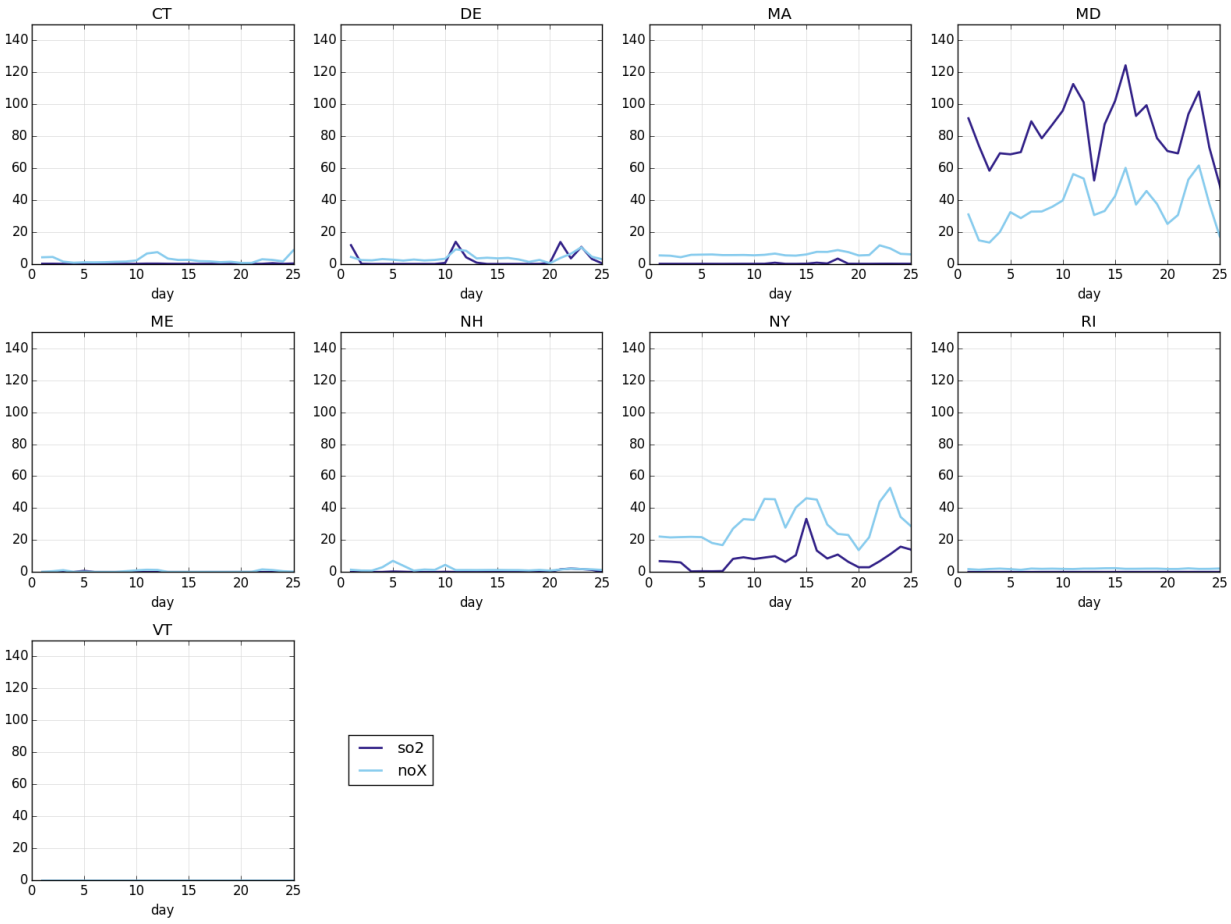
Important links:

- agate: http://agate.rtfd.org
- Documentation: http://agate-charts.rtfd.org
- Repository: https://github.com/onyxfish/agate-charts
- Issues: https://github.com/onyxfish/agate-charts/issues

CHAPTER 2

Why agate-charts?

- A clean, readable API.
- Automatically infers good defaults from your data.
- Far simpler than using matplotlib directly.
- Code, chart, code, chart. No context switching.
- It makes this with one line of code:



CHAPTER 3

Table of contents

3.1 Installation

3.1.1 Users

If you only want to use agate-charts, install it this way:

```
pip install agate-charts
```

3.1.2 Developers

If you are a developer that also wants to hack on agate-charts, install it this way:

```
git clone git://github.com/onyxfish/agate-charts.git
cd agate-charts
mkvirtualenv agate-charts
pip install -r requirements.txt
python setup.py develop
tox
```

3.1.3 Supported platforms

agate-charts supports the following versions of Python:

- Python 2.6 (tests pass, but some dependencies claim not to support it)
- Python 2.7
- Python 3.2
- Python 3.3
- Latest PyPy

It works anywhere matplotlib works.

3.2 Tutorial

3.2.1 About agate-charts

agate-charts is an extension for the agate data analysis library that adds support for quickly exploring data using charts. It does not create polished or publication-ready graphics. If you haven't used agate before, please read the agate tutorial before reading this.

In this tutorial we will use agate-charts to explore a time-series dataset from the EPA documenting US greenhouse gas emissions for the month of June 2015.

3.2.2 Installing agate-charts

```
pip install agate-charts
```

3.2.3 Getting setup

Let's start by creating a clean workspace:

```
mkdir agate_charts_tutorial
cd agate_charts_tutorial
```

Now let's download the data:

```
curl -L -O https://raw.githubusercontent.com/onyxfish/agate-charts/master/examples/epa-emissions-201
```

You will now have a file named `epa-emissions-20150910.csv` in your `agate_charts_tutorial` directory.

Note: agate-charts plays nicely with ipython, Jupyter notebooks and derivative projects like Atom's hydrogen plugin. If you prefer to go through this tutorial in any of those environments all the examples will work the same. You may need to add `%matplotlib inline` to the top of your scripts as you would in an ipython notebook.

3.2.4 Importing out dependencies

Our only dependencies for this tutorial will be agate and agate-charts. When we import `agatecharts` we call its *`patch()`* function, which attaches the *`TableCharts`* methods to `Table` and the *`TableSetCharts`* methods to `TableSet`.

```
import agate
import agatecharts

agatecharts.patch()
```

3.2.5 Loading the data

Now let's load the dataset into an `Table`. We'll use an `TypeTester` so that we don't have to specify every column, but we'll force the :code:' Date' column to be a date since it is in a known format.

```
tester = agate.TypeTester(force={
    ' Date': agate.Date('%Y-%m-%d')
})

emissions = agate.Table.from_csv('examples/epa-emissions-20150910.csv', tester)
```

Now let's compute a few derived columns in order to make our charting easier. The first column will be the numerical day of the month. The latter three correct for an issue where the EPA has included empty columns instead of numerical zeroes.

```
emissions = emissions.compute([
    (agate.Formula(agate.Number(), lambda r: r[' Date'].day), 'day'),
    (agate.Formula(agate.Number(), lambda r: r[' SO2 (tons)'] or 0), 'so2'),
    (agate.Formula(agate.Number(), lambda r: r[' NOx (tons)'] or 0), 'nox'),
    (agate.Formula(agate.Number(), lambda r: r[' CO2 (short tons)'] or 0), 'co2')
])
```

Of course, for analysis purposes you should always be extremely cautious in assuming that blank fields are equivalent to zero. For the purposes of this tutorial, we will assume this is a valid transformation.

3.2.6 Your first chart

The emissions dataset includes data for several states. We'll look at the states individually later on, but to start out let's aggregate some totals:

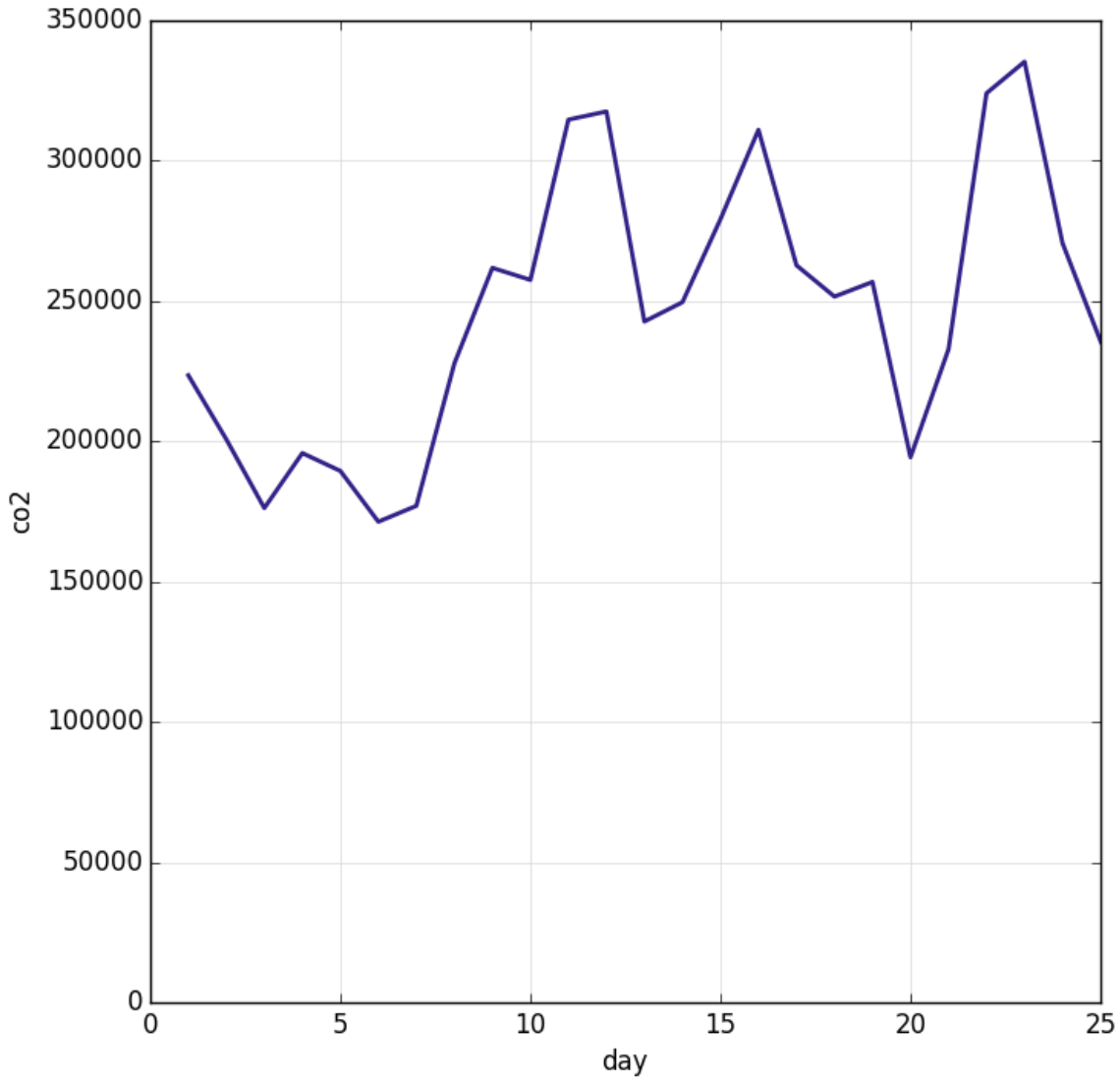
```
days = emissions.group_by('day', key_type=agate.Number())
day_totals = days.aggregate([
    ('so2', agate.Sum(), 'so2'),
    ('co2', agate.Sum(), 'co2'),
    ('nox', agate.Sum(), 'nox')
])
```

The `day_totals` table now contains total counts of each type of emission. Note that we don't know if this data is comprehensive so we shouldn't assume these are national totals. (In fact, I know that they aren't for reasons that will become obvious shortly.)

Now let's render a line chart of the total `co2`:

Notice that `line_chart` is a method on the `Table`. Remember that when we imported `agatecharts` with called *`patch()`* which added *`TableCharts`* methods such as *`TableCharts.line_chart()`* to `Table` and the *`TableSetCharts`* methods to `TableSet`.

If all goes well, you should see a window popup containing this image:



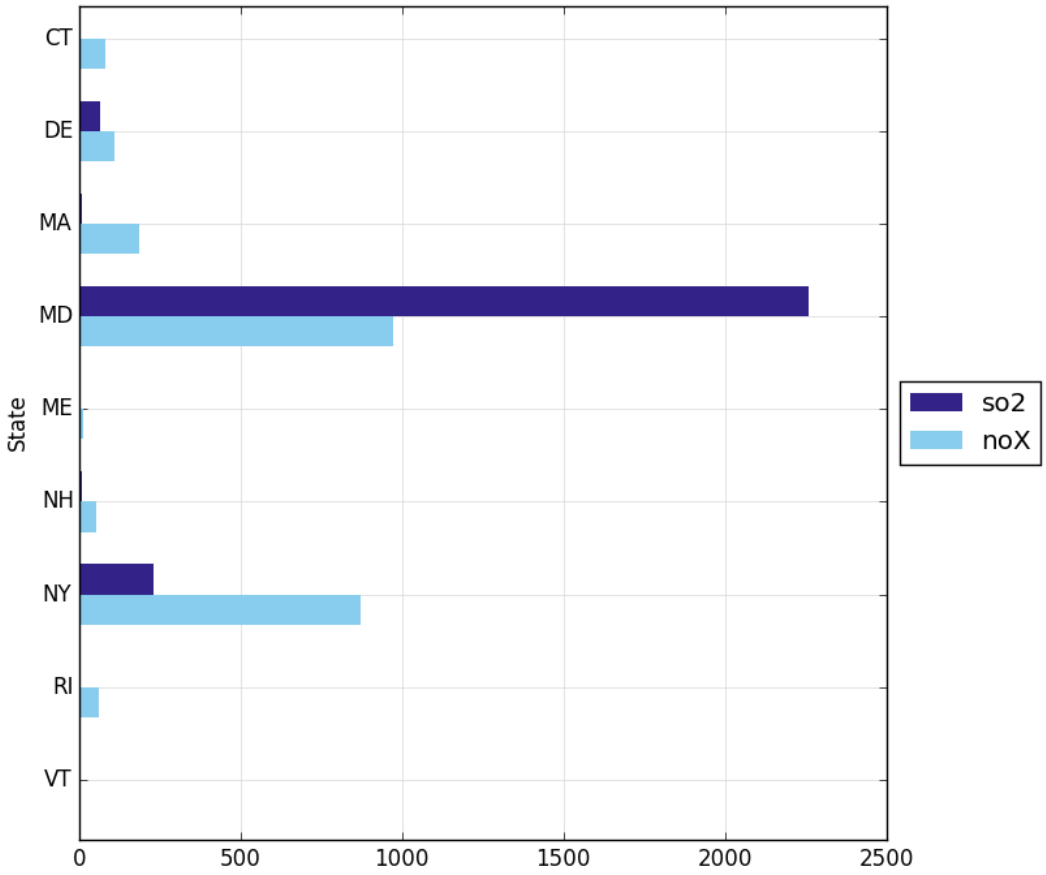
You can also choose to render the image directly to disk, by passing the `filename` argument:

Warning: agate-charts uses matplotlib to render charts. Matplotlib is a notoriously complicated and finicky piece of software. agate-charts attempts to abstract away all the messiest bits, but you may still have issues with charts not rendering on your particular platform. If the script hangs, or you don't see any output, try specifying a rendering backend *before* importing agate-charts. This shouldn't be an issue if you're rendering to files.

3.2.7 Rendering multiple series

You may also want to render charts that compare to series of data. For instance, in this dataset the sulfur dioxide (`so2`) and nitrogen oxide (`nox`) amounts are on similar scales. Let's roll the data up by state and compare them with a bar chart:

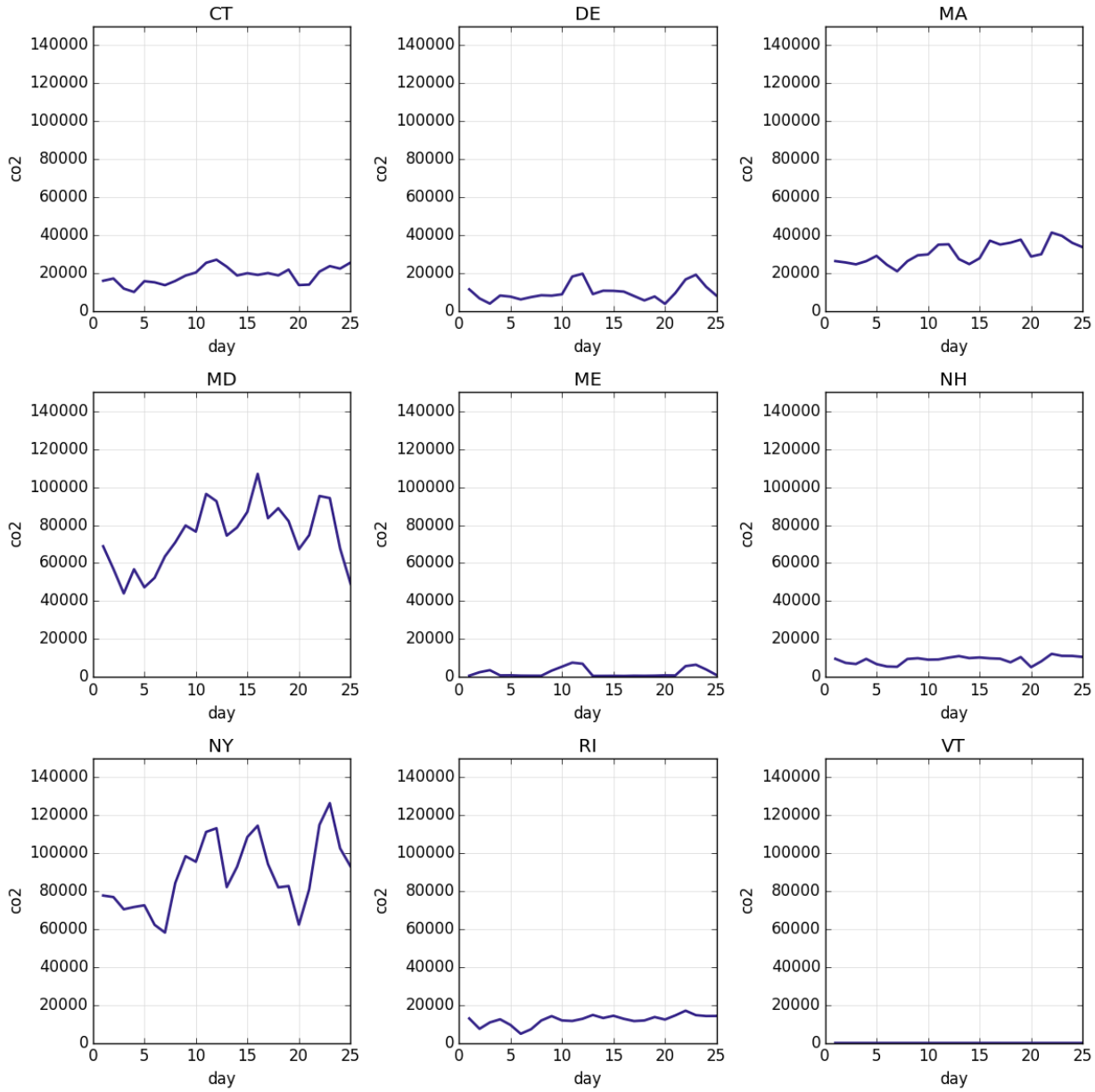
```
states = emissions.group_by('State')
state_totals = states.aggregate([
    ('so2', agate.Sum(), 'so2'),
    ('co2', agate.Sum(), 'co2'),
    ('noX', agate.Sum(), 'noX')
])

state_totals.bar_chart('State', ['so2', 'noX'])
```



3.2.8 Small multiples

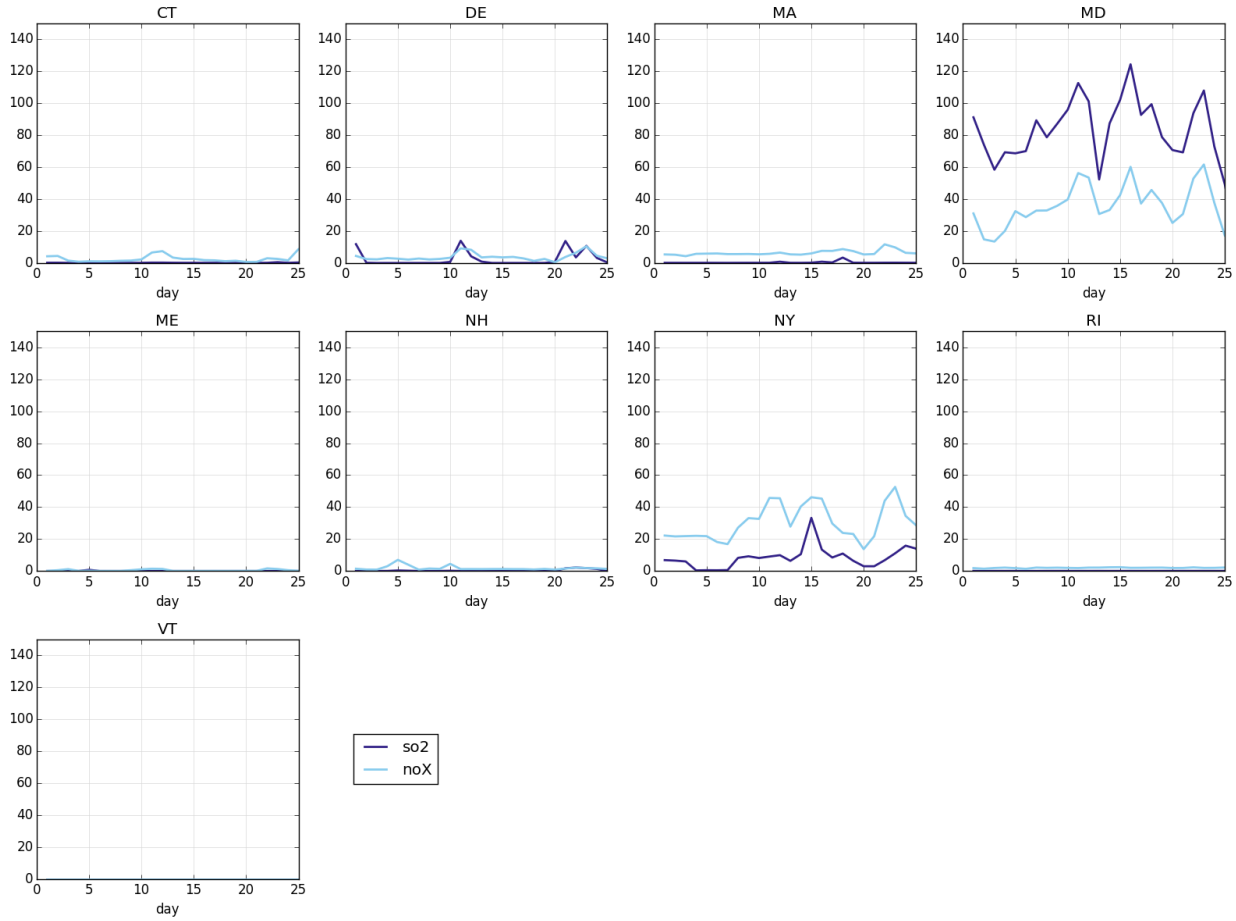
agate-charts most powerful feature comes when these same methods are applied to instances of agate's `TableSet`. In this case, agate-charts will automatically create small multiples of the chart for each table in the set. For example, here is a let's create a line chart of the `co2` output for each state:

```
states.line_chart('day', 'co2')
```



Of course, you can also combine small multiples and multiple time series:

```
states.line_chart('day', ['so2', 'noX'])
```



3.2.9 Where to go next

agate-charts is designed for making quick exploratory charts that you don't put a lot of thought into. From here you might take your data into Illustrator, D3 or some other tool for creating a polished presentation.

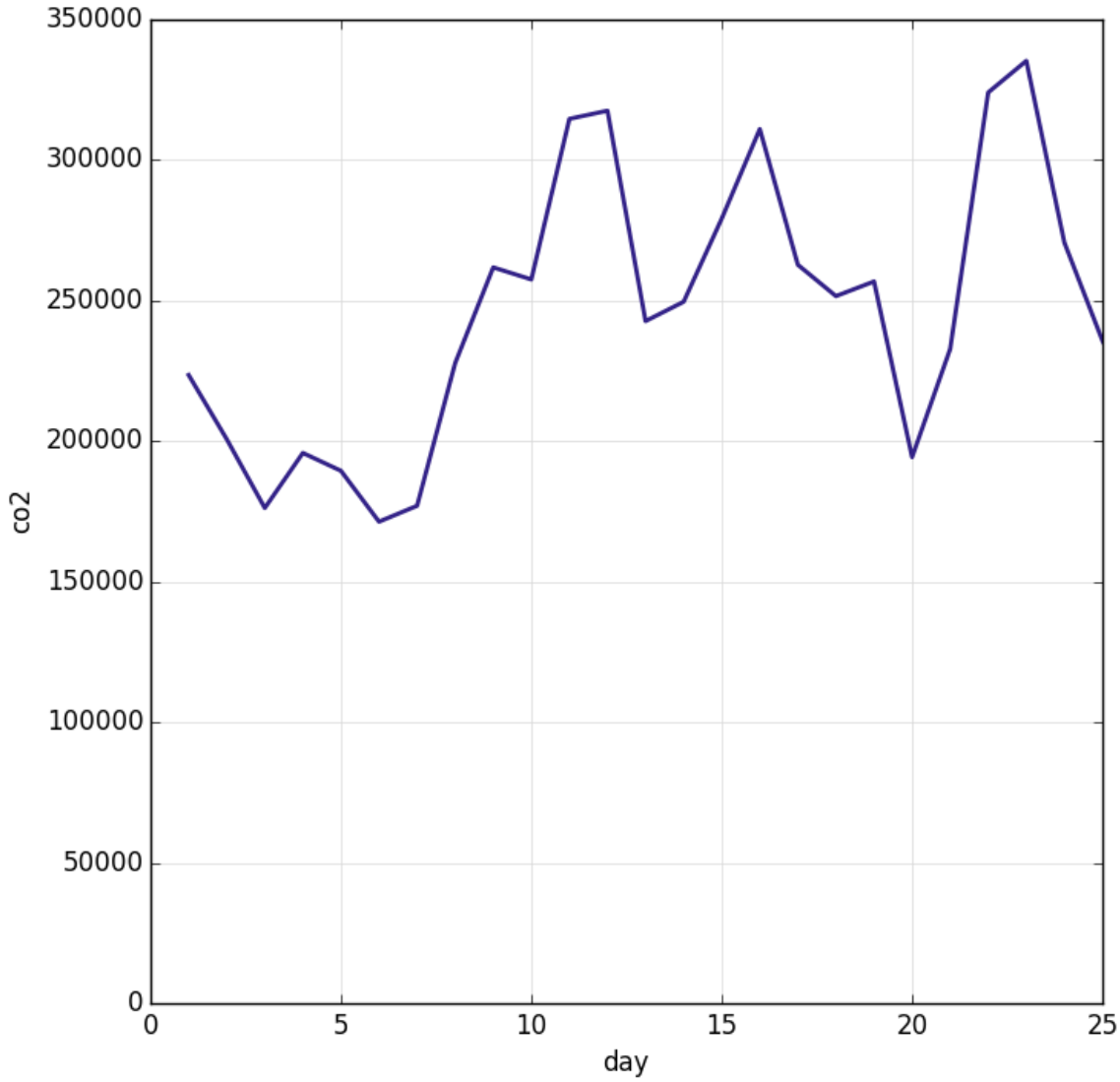
If you enjoy using agate-charts you should also check out proof, a library for building data processing pipelines that are repeatable and self-documenting. If you're rendering many charts it can save you tons of time by skipping ones you've already done.

3.3 Sample charts

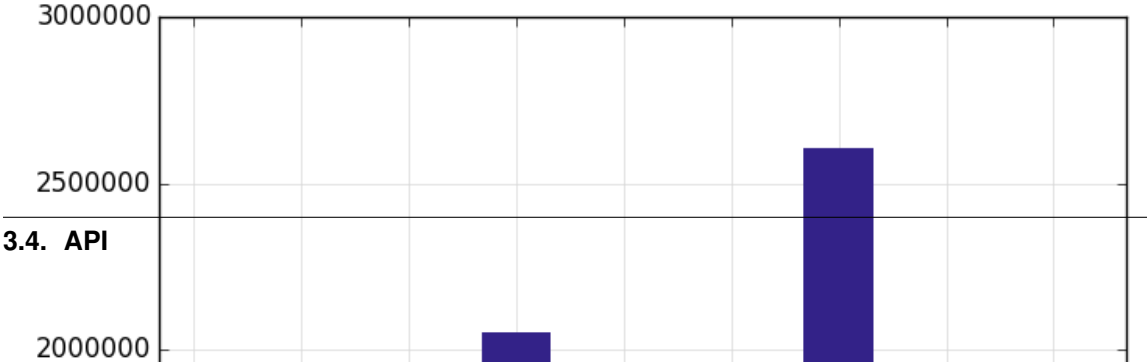
All of the charts on this page were produced by agate-charts with a single line of code each.

3.3.1 Single series

Lines



Columns



agatecharts.table.**DEFAULT_SIZE = (8, 8)**
Default rendered chart size in inches

agatecharts.table.**DEFAULT_DPI = 72**
Default rendered chart dpi

class agatecharts.table.**TableCharts**

bar_chart(*label_column_name*, *value_column_names*, *filename=None*, *size=(8, 8)*, *dpi=72*)
Plots a bar chart.

See *TableCharts._plot()* for an explanation of keyword arguments.

Parameters

- **label_column_name** – The name of a column in the source to be used for the vertical axis labels. Must refer to a column containing `Text`, `Number` or `Date` data.
- **value_column_names** – One or more column names in the source, each of which will used to define the horizontal width of a bar. Must refer to a column containing `Number` data.

column_chart(*label_column_name*, *value_column_names*, *filename=None*, *size=(8, 8)*, *dpi=72*)
Plots a column chart.

See *TableCharts._plot()* for an explanation of keyword arguments.

Parameters

- **label_column_name** – The name of a column in the source to be used for the horizontal axis labels. Must refer to a column containing `Text`, `Number` or `Date` data.
- **value_column_names** – One or more column names in the source, each of which will used to define the vertical height of a bar. Must refer to a column containing `Number` data.

line_chart(*x_column_name*, *y_column_names*, *filename=None*, *size=(8, 8)*, *dpi=72*)
Plots a line chart.

See *TableCharts._plot()* for an explanation of keyword arguments.

Parameters

- **x_column_name** – The name of a column in the source to be used for the horizontal axis. May refer to a column containing `Number`, `Date` or `DateTime` data.
- **y_column_names** – A sequence of column names in the source, each of which will be used for the vertical axis. Must refer to a column with `Number` data.

scatter_chart(*x_column_name*, *y_column_name*, *filename=None*, *size=(8, 8)*, *dpi=72*)
Plots a scatter plot.

See *TableCharts._plot()* for an explanation of keyword arguments.

Parameters

- **x_column_name** – Column containing X values for the points to plot. Must refer to a column containg `Number` data.
- **y_column_name** – Column containing Y values for the points to plot. Must refer to a column containg `Number` data.

_plot(*chart*, *filename=None*, *size=(8, 8)*, *dpi=72*)
Execute a plot of this `Table`.

This method should not be called directly by the user.

Parameters

- **chart** – An chart class to render.
- **filename** – A filename to render to. If not specified will render to screen in "interactive mode".
- **size** – A (width, height) tuple in inches defining the size of the canvas to render to.
- **dpi** – A number defining the pixels-per-inch to render.

agatecharts.tableset.**DEFAULT_MULTIPLE_SIZE = (4, 4)**
Default small multiple chart size in inches

class agatecharts.tableset.**TableSetCharts**

bar_chart(*label_column_name*, *value_column_names*, *filename=None*, *size=None*, *dpi=72*)
See *TableCharts.bar_chart()*.

column_chart(*label_column_name*, *value_column_names*, *filename=None*, *size=None*, *dpi=72*)
See *TableCharts.column_chart()*.

line_chart(*x_column_name*, *y_column_names*, *filename=None*, *size=None*, *dpi=72*)
See *TableCharts.line_chart()*.

scatter_chart(*x_column_name*, *y_column_name*, *filename=None*, *size=None*, *dpi=72*)
See *TableCharts.scatter_chart()*.

_plot(*chart*, *filename=None*, *size=None*, *dpi=72*)
See *TableCharts._plot()*.

CHAPTER 4

Authors

The following individuals have contributed code to agate-charts:

- Christopher Groskopf

CHAPTER 5

License

The MIT License

Copyright (c) 2015 Christopher Groskopf and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 6

Changelog

6.1 0.1.0

- Use agate 0.10.0 extension API.
- TableSet.bar_chart
- TableSet.scatter_chart
- TableSet.column_chart
- TableSet.line_chart
- Table.bar_chart
- Table.scatter_chart
- Table.column_chart
- Table.line_chart

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

a

Symbols

A

B

C

D

L

P

S

T