# Feedback and Contribution

We welcome any input, feedback, bug reports, and contributions via [Altair's

GitHub Repository](http://github.com/altair-viz/altair/). In particular, we

welcome companion efforts from other visualization libraries to render the

Vega-Lite specifications output by Altair. We see this portion of the effort

as much bigger than Altair itself: the Vega and Vega-Lite specifications are

perhaps the best existing candidates for a principled \*lingua franca\* of data

visualization.

We are also seeking contributions of additional Jupyter notebook-based examples

in our separate GitHub repository: https://github.com/altair-viz/altair\_notebooks.

The altair users mailing list can be found at

https://groups.google.com/forum/#!forum/altair-viz. If you are working on

Altair, you can talk to other developers in the `#altair` channel of the [Vega

slack](https://bit.ly/join-vega-slack).

## How To Contribute Code to Alatir

### Setting Up Your Environment

Install the latest version of Altair locally using

```

$ pip install git+https://github.com/altair-viz/altair/

```

Next step is to fork the repository on GitHub and clone the fork to you local

machine. For more details on forking see the [GitHub

Documentation](https://help.github.com/en/articles/fork-a-repo).

```

$ git clone https://github.com/YOUR-USERNAME/altair.git

```

You can have a single clone of the repository that points to both your fork and

the main package repository. These pointers to GitHub are called "remotes" .

On your local clone you should run:

```

$ git remote add upstream https://github.com/altair-viz/altair.git

$ git checkout master

$ git pull upstream master

```

And then you'll have all the updates in the master branch of your local fork.

Note that git will complain if you've committed changes to your local master

branch that are not on upstream (this is one reason it's good practice to \*\*never\*\*

work directly on your master branch).

### Creating a Branch

Once your local environment is up-to-date, you can create a new git branch which will

contain your contribution:

```

$ git checkout -b <branch-name>

```

With this branch checked-out, make the desired changes to the package.

Note that Altair code uses the [black](https://black.readthedocs.io/)

code formatter, which you can apply to your modifications by installing

and running black on the local directory:

```

$ pip install black

$ black .

```

When you are happy with your changes, you can commit them to your branch by runing

```

$ git add <modified-file>

$ git commit -m "Some descriptive message about your change"

$ git push origin <branch-name>

```

Finally you will need to submit a pull request (PR) on GitHub asking to merge

your example branch into altair master. For details on creating a PR see GitHub

documentation [Creating a pull

request](https://help.github.com/en/articles/creating-a-pull-request). You can

add more details about your example in the PR such as motivation for the

example or why you thought it would be a good addition. You will get feed back

in the PR discussion if anything needs to be changed. To make changes continue

to push commits made in your local example branch to origin and they will be

automatically shown in the PR.

Hopefully your PR will be answered in a timely manner and your contribution will

help others in the future.

### Testing your Changes

When you submit a pull request, Altair's continuous integration test suite will

run a number of tests to validate the correctness of your code. It can be helpful

when debugging to run those tests locally; to do this first install Altair's

development requirements:

```

$ pip install -r requirements\_dev.txt

```

and then run the test suite with:

```

$ make test

```

## Adding Examples

We are always interested in new examples contributed from the community. These

could be everything from simple one-panel scatter and line plots, to more

complicated layered or stacked plots, to more advanced interactive features.

Before submitting a new example check the [Altair Example

Gallery](https://altair-viz.github.io/gallery/index.html) to make sure that

your idea has not already been implemented.

Once you have an example you would like to add there are a few guide lines to follow.

Every example should:

- be saved as a stand alone script in the `altair/examples/` directory.

- have a descriptive docstring, which will eventually be extracted for the

documentation website.

- contain a category tag.

- define a chart variable with the main chart object (This will be used both in

the unit tests to confirm that the example executes properly, and also

eventually used to display the visualization on the documentation website).

- not make any external calls to download data within the script (i.e. don't

use urllib). You can define your data directly within the example file,

generate your data using pandas and numpy, or you can use data

available in the `vega\_datasets` package.

The easiest way to get started would be to adapt examples from the [Vega-Lite

example gallery](https://vega.github.io/vega-lite/examples/) which are missing

in the altair gallery. Or you can feel free to be creative and build your own

visualizations.

Often it is convenient to draft an example outside of the main repository, such

as [Google Colab](https://colab.research.google.com/), to avoid difficulties

when working with git. Once you have an example you would like to add, follow the

same contribution procedure outlined above.

Some additional notes:

- all examples should be in their own file in the `altair/examples` directory, and

the format and style of new contributions should generally match that of existing examples.

- The file docstring will be rendered into HTML via

[reStructuredText](http://docutils.sourceforge.net/rst.html), so use that

format for any hyperlinks or text styling. In particular, be sure you include

a title in the docstring underlined with `---`, and be sure that the size of

the underline exactly matches the size of the title text.

- If your example fits into a chart type but involves significant configuration

it should be in the `case studies` category. If your example doesn't fit well

into any category then it can be included in the `other charts` category.

- For consistency all data used for a visualization should be assigned to the

variable `source`. Then `source` is passed to the `alt.Chart` object. See

other examples for guidance.

- Example code should not require downloading external datasets. We suggest

using the `vega\_datasets` package if possible.

If you are using the `vega\_datasets` package there are multiple ways to refer

to a data source. If the dataset you would like to use is included in local

installation (`vega\_datasets.local\_data.list\_datasets()`) then the data can

be referenced directly, such as `source = data.iris()`. If the data is not

included then it should be referenced by URL, such as `source =

data.movies.url`. This is to ensure that Altair's automated test suite does

not depend on availability of external HTTP resources.

Note that examples shown on the [altair website](https://altair-viz.github.io/)

are only updated when a new version is released so your new example might not show

up there for a while.