

Present Your Data Science Projects with Jupyter Notebook Slides!



Tahsin Mayeesha

Follow



Feb 12, 2018 · 5 min read

The art of going from code to communication.

Making presentation slides for technical lectures, product live-demo's and sharing analysis is one of those boring tasks that has high return compared to investment. A good presentation slide can make sharing the output of complex analysis easy and used for concise communication to customers and shareholders. Making good slides is more of an art than craft, however using good tools can go a long way.

Trouble With Common Presentation Tools

Microsoft Powerpoint, Google Slides, Prezi are some of the commonly used tools for making presentation. But when it comes to making technical presentations with code those tools often fall short. Taking snapshots of code and pasting the images to the

slides is ineffective because fonts often get blurred. It's also hard to customize the slides easily when using power-point or google slides. Images often have to be resized and overall look and feel of the slides ends up looking more corporate than technical.

Benefits of Jupyter Slides

Given all of these issues finding a good presentation tool that lets us effectively deliver code, visualization, model diagram and outputs of the executed code is very important for presenting machine learning projects.

For my personal projects, I've found slides work better for communicating with general audience compared to sharing jupyter notebooks directly because many people don't want to directly look at a wall of code at once unless they are also technical.

I mostly work in python and I'm a big fan of Jupyter Notebooks. So I was super happy to learn that we can directly convert our notebooks to HTML slideshow with reveal.js HTML presentation framework using nbconvert commands.

A output slide will look like this :

Jupyter Slides Demo



Implementation

Make the notebook

First step is to create the notebook that will be used for generating the slides. Add your resources, notes, visualizations, format code as necessary. Fix the relative path, url, blockquote, diagrams etc . This is the notebook that was used here.

Jupyter Slides Demo

```
In [23]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('seaborn-white')
%matplotlib inline
```

```
In [28]: # dataset resource : http://blog.revolutionanalytics.com/2017/05/the
datasauras = pd.read_csv('DatasaurusDozen.tsv', sep='\t')
```

```
In [29]: dino = datasauras[datasauras['dataset']=='dino']
```

```
In [30]: datasauras.head()
```

```
Out[30]:
```

	dataset	x	y
0	dino	55.3846	97.1795
1	dino	51.5385	96.0256

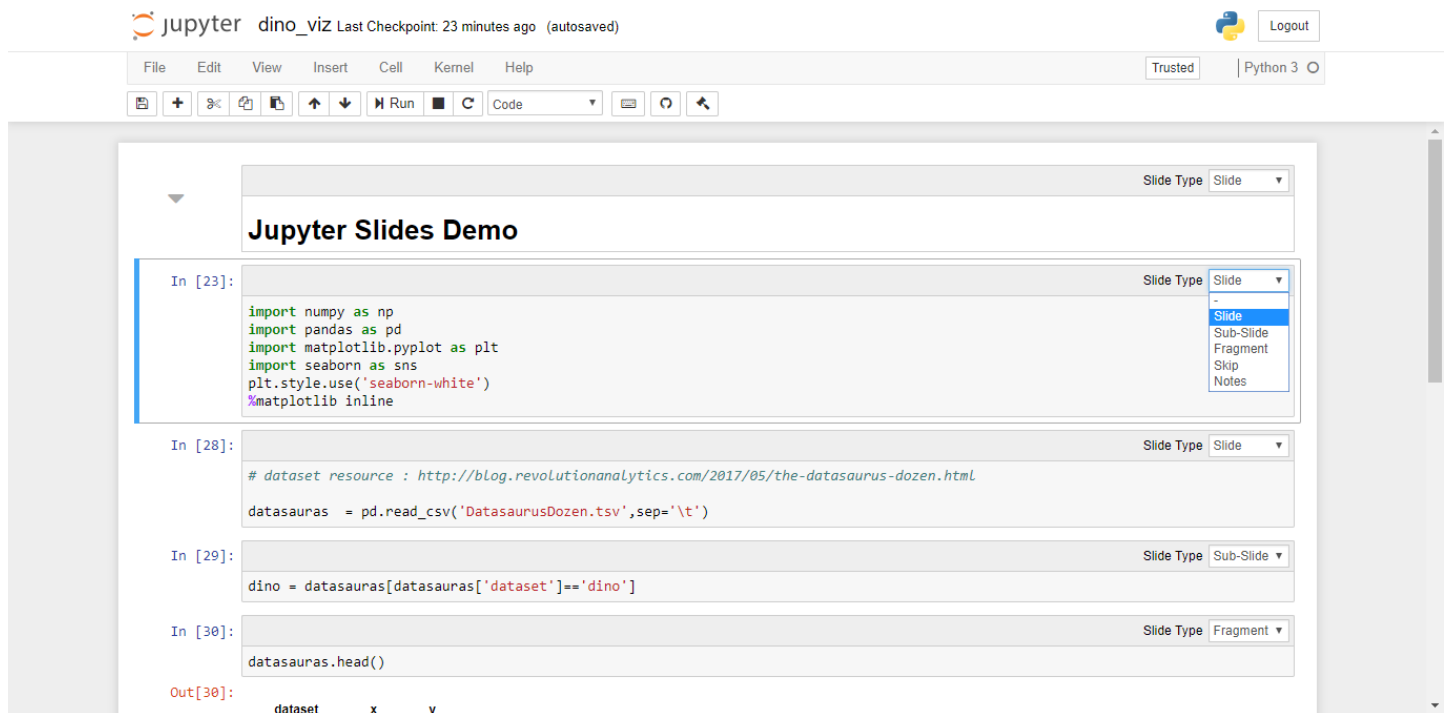
dino_viz.ipynb hosted with ❤ by GitHub

[view raw](#)

Format the cells

Then we have to format the cells for the slideshow. For this go to View → Cell Toolbar → Slideshow. This creates a light gray bar on top of each cell to customize the slide type. Remember that in Jupyter Notebook cells are the units containing code and markdown. There are 5 options for each cell. Slide, Sub-slide, fragment, skip and notes.

- **Slide** : Presents each cell in a different slide. Good for containing heading, sub-heading, image and text notes. Use right arrow to navigate.
- **Sub-slide** : A continuation from the previous slide. Use down arrow to go to a sub-slide from a slide.
- **Fragment** : Fragment is interesting because it presents multiple cells in one slide one by one.
- **Skip/Notes** : Skip skips the marked cells. Cells that are unused for the presentation can also be marked as the notes.



Serve Locally and Iterate

After that go to the directory of the notebook and type out the nbconvert command to convert the notebook to the slides which will open the slides in your browser.

To serve the slides in default format type and insert your notebook name instead of *.ipynb

```
jupyter nbconvert *.ipynb --to slides --post serve
```

The above notebook was generated by this command. Notice the extra configuration I've used with SlidesExporter .

```
jupyter nbconvert dino_viz.ipynb --to slides --post serve
--SlidesExporter.reveal_theme=serif
--SlidesExporter.reveal_scroll=True
--SlidesExporter.reveal_transition=none
```

Brief overview of configuration used :

- `SlidesExporter.reveal_theme` : sets the theme to serif. There's also the default and sky theme along with many others.

- `SlidesExporter.reveal_scroll` : sets the scrolling option to True. For big images or long cells scrolling options are helpful. It's also helpful for visualizing dataframes.
- `SlidesExporter.reveal_transition` : Sets the transition to None. I don't like to use any transition effect because adding them creates a sort of jerkiness to the screen which I believe to be unsuitable for code. The options are : none, fade, slide, convex, concave and zoom.

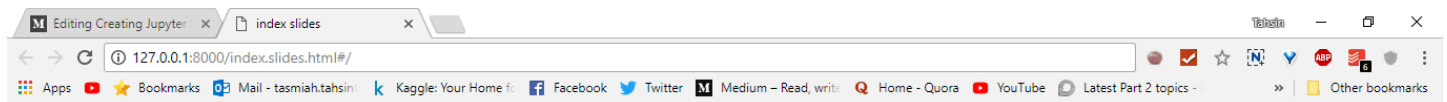
A full list of the configuration options can be found here :

Configuration options - nbconvert 5.3.2.dev0 documentation

Configuration options may be set in a file, `~/.jupyter/jupyter_nbconvert_config.py` , or at the command line when...

nbconvert.readthedocs.io

The slides will be served at port 8000. However don't just stop there. Experiment with multiple layouts and fragments(specially) to see how it works best for your own project. A hardcore machine learning based project might do well in sticking to the default theme while a data journalism based project may need to try all the transition effects. Adding custom .css files may also help.



Jupyter Slides Demo



Hosting on Github Pages

Dheepak Krishnamurthy has written an incredibly good post on it which I've used with minimal changes. I'll just add how I did it and add some more details.

- First clone reveal.js repo to your directory containing the notebook to add it as a submodule.

```
git submodule add https://github.com/hakimel/reveal.js.git reveal.js
```

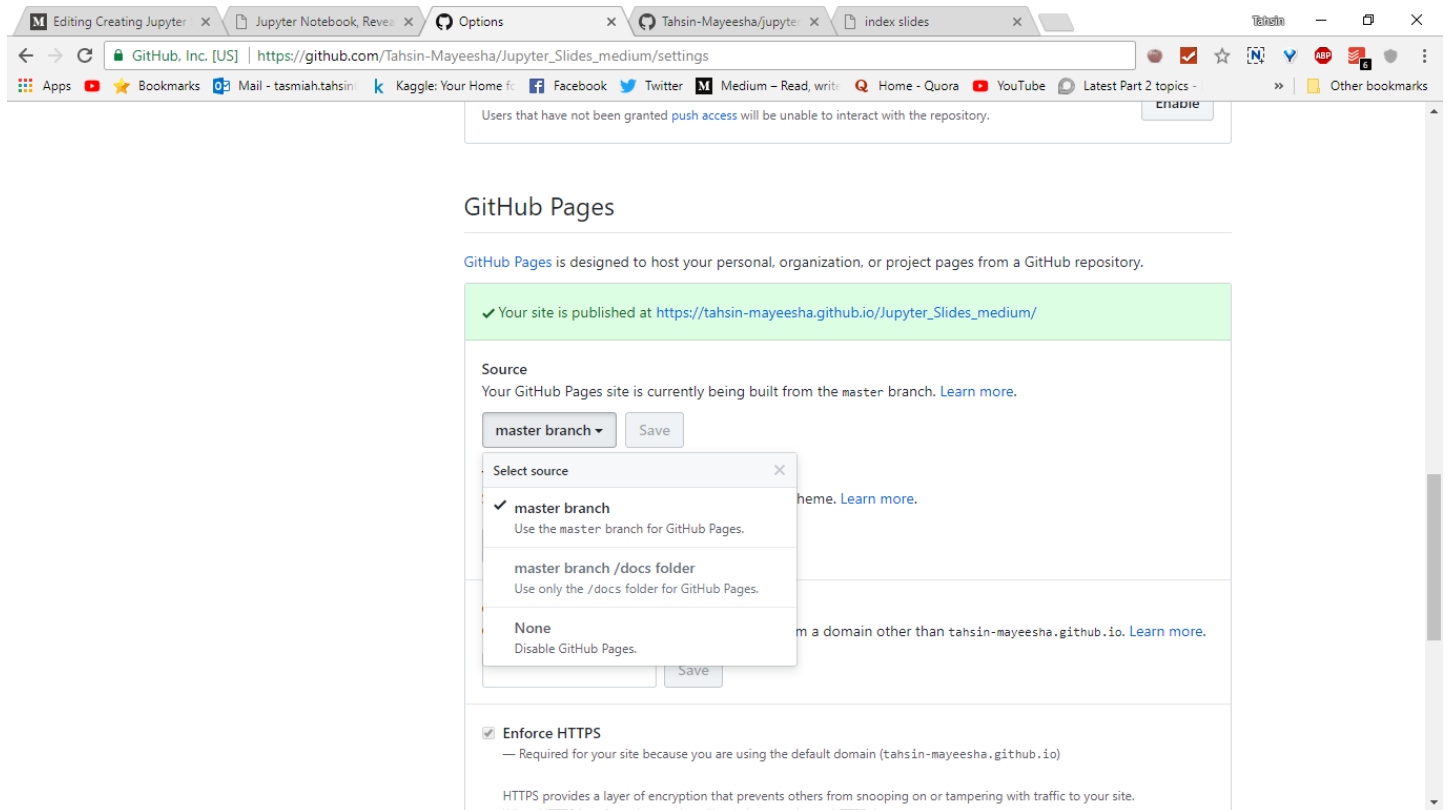
- Rename your notebook to `index.ipynb` because the `nbconvert` command creates a html file with the notebook name and we want to create the `index.html` file for our site to host on github pages.
- Run the script in the terminal to generate `index.slides.html` file.

```
jupyter nbconvert --to slides index.ipynb --reveal-prefix=reveal.js
```

- Since I've added my configuration parameters I've used the following :

```
jupyter nbconvert --to slides index.ipynb --reveal-prefix=reveal.js  
--SlidesExporter.reveal_theme=serif  
--SlidesExporter.reveal_scroll=True  
--SlidesExporter.reveal_transition=none
```

- Push them to github in a new repo and make that repository a github pages site. Easiest way to do it is to just go to settings → Github pages section → select master as source → save.



- Your site will be available in `username.github.io/reponame/`. In this case the slides are available [here](https://tahsin-mayeesha.github.io/jupyter_slides_medium/). You can just click the “your site is published here” link in settings though.
- If you have forked reveal.js instead of cloning consult the [post](#) by Krisnamurthy and follow the procedure.

NOTE : I'm on Windows so the terminal commands can be a little different depending on your OS. Please consult stackoverflow or other resources in that case because I'd be unable to help.

And thanks to Matthew Speck for helping me to first get started with jupyter slides with his amazing [article](#).

References :

- <https://medium.com/@mjspeck/presenting-code-using-jupyter-notebook-slides-a8a3c3b59d67>
- <https://nbconvert.readthedocs.io/en/latest/customizing.html>
- <https://www.youtube.com/watch?v=EOpcxy0RA1A&t=539s>
- <http://blog.kdheepak.com/jupyter-notebook-revealjs-and-github-pages.html>

Thanks to TDS Editors.

[Github](#)

[Data Science](#)

[Machine Learning](#)

[Jupyter Notebook](#)

[Data Visualization](#)

[About](#)

[Write](#)

[Help](#)

[Legal](#)

Get the Medium app

