



data science websites

David Nicholson

why should I care?

blogs and other websites are a good way to:

- share your data science projects
- interact with a research community
- "build your brand"

high level overview

1. **follow instructions for set up**
2. **write your posts**
3. **follow instructions to update after you write post**
4. tweet out your blog posts and hope someone reads them
5. repeat steps 2 through 4 as needed
6. ...
7. become a famous data scientist (sexiest job, 2014-present)
8. publish a book and / or start a podcast

instructions for set up

1. make sure you have **git** installed for **version control**, and so you can take advantage of **Github**
2. make a **virtual environment** for the **software libraries** you'll use to blog
 - a. **pelican**: a **static** website generator written in Python
 - b. **jupyter** to create notebooks
 - c. **ghp-import**: a **script** that lets you use **Github Pages** to host your site by doing all the work for you
 - d. some other tools that make it easy to use pelican and jupyter (Markdown, etc.)
3. use **pelican-quickstart** command to quickly start a project (your blog)

instructions for set up (to follow blindly)

1. make sure you have **git** installed (using **conda**)

```
$ conda install git
```

instructions for set up (to follow blindly)

2. make an **environment** (using **conda**) ...

```
$ conda create -n pelican-blog python=3.6 jupyter notebook
```

```
$ source activate pelican-blog
```

instructions for set up (to follow blindly)

2. ... and install the **software libraries** you'll use (with **pip**)

```
(pelican-blog) $ pip install pelican Markdown ghp-import
```

instructions for set up (to follow blindly)

2. ... and install the **software libraries** you'll use (with **pip**)

```
(pelican-blog) $ mkdir your-blog
```

```
(pelican-blog) $ cd your-blog
```

```
(pelican-blog) $ git init
```


instructions for set up (to follow blindly)

2. ... and install the **software libraries** you'll use (with **pip**)

```
(pelican-blog) $ git submodule add  
https://github.com/danielfrg/pelican-ipynb
```

```
(pelican-blog) $ git submodule add  
https://github.com/getpelican/pelican-plugins
```

instructions for set up (to follow blindly)

3. use `pelican-quickstart` command to quickly start a project (your blog)

```
(pelican-blog) $ pelican-quickstart
```

Pelican will ask you a series questions about your site. You can accept all the defaults by just hitting enter--everything in this tutorial should work with those defaults.

write your posts

using **Markdown**, e.g.:

Title: Red, A Network Analysis Library

Date: 2018-06-27

Category: test

Introduction

I just released the first version of a [network analysis library, Red](github/YourUserName/red.git).

How-To

Here's an example script in Python:

```
```Python
```

```
import red
```

```
twitter_data = red.datasets.twitter
```

```
red = RedCreador(twitter_data)
```

# instructions to update after you write your post

1. use pelican to convert your Markdown post into **html** that browsers know how to read

```
(pelican-blog) $ pelican content -o output -s pelicanconf.py
```

2. use ghp-import to publish your Pelican site on **Github**, as either a **Project Page** or a **User Page**

```
(pelican-blog) $ ghp-import output
```

```
(pelican-blog) $ git push origin gh-pages
```

See notes on how to do this for a User Page on the next two slides

# instructions to update after you write your post

2. use `ghp-import` to publish your Pelican site on **Github**, as either a **Project Page** or a **User Page**

for a User Page, you need to "*push*" the content from the output directory to the master branch of a repository named `Username.github.io`. (So you should make one, replacing Username with your actual user name on Github,)

Github automagically converts a repository with such a name into a website.

Step-by-step instructions for this on the command line are on the next slide.

**Note that this means you need a separate repository for your source code!**

For example we have the source code for the group's blog in:

<https://github.com/Data-Science-for-Scientists-ATL/DataSci4ScienceATL-blog-source>

And the "output" gets pushed to a different repository:

<https://github.com/Data-Science-for-Scientists-ATL/Data-Science-for-Scientists-ATL.github.io>

# instructions to update after you write your post

2. use `ghp-import` to publish your Pelican site on **Github**, as either a **Project Page** or a **User Page**

so once you have set up the separate repository for your User Page, you'd push to it like so:

```
(pelican-blog) $ pelican content -o output -s pelicanconf.py
(pelican-blog) $ ghp-import output
(pelican-blog) $ git push git@github.com:elemoine/elemoine.github.io.git
gh-pages:master
```

if pushing fails, e.g. because you already have content in the repository, you can *force* it with the `-f` flag, although be warned that this will write over everything:

```
(pelican-blog) $ git push -f
git@github.com:elemoine/elemoine.github.io.git gh-pages:master
```

## useful links

get Anaconda here, with the conda command-line tool for managing packages and installing libraries (including many scientific libraries):

<https://www.anaconda.com/download/>

pelican themes: <http://pelicanthemes.com/>

how to use themes:

<https://github.com/getpelican/pelican-themes#using-themes>

## more reading

get Anaconda here, with the conda command-line tool for managing packages and installing libraries (including many scientific libraries):

<https://www.anaconda.com/download/>

good example of why you would use **virtual environments**:

<https://www.dabapps.com/blog/introduction-to-pip-and-virtualenv-python/>

all about **Markdown**: <https://daringfireball.net/projects/markdown/>

using ghp-import to publish to Github (from Pelican docs):

<http://docs.getpelican.com/en/stable/tips.html#publishing-to-github>