Como fazer o controle básico de versão com o Git e o Github

As primeiras coisas que se precisa fazer são:
1. Verifique se o Git está instalado na máquina (aqui está um guia para verificar em Mac e outro para verificar em Windows),
2. Criar uma conta do Github, caso não tiver uma e, em seguida,
3. Configurar a autenticação do Github na máquina.

Depois de configurar o Git e o Github, a primeira coisa a fazer é criar um repositório remoto para um novo projeto (veja aqui um excelente guia sobre como fazer isso). Você então precisa clonar uma cópia local deste repositório para o computador, desta forma.
Assim que tivermos nosso repositório local, podemos começar a trabalhar em nosso novo projeto em R. Digamos que criamos um novo arquivo R script! E começamos a trabalhar nele.
A qualquer momento (quanto mais frequente, melhor), podemos fazer um **commit**. Para fazer isso, abra o terminal e navegue até o repositório local para este projeto.
Em seguida, verificamos quais arquivos o Git está **acompanhando** (*tracking*) (ou não - *untracking*) digitando:

```
git status

On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working
directory)

     [31mmodified:   2015-11-11-understanding-object-orientated-
programming-in-python.ipynb [m
     [31mmodified:   2016-06-01-web-scraping-in-python.ipynb [m
     [31mmodified:   2016-10-04-reproducible-research-in-
python.ipynb [m

Untracked files:
  (use "git add <file>..." to include in what will be committed)

     [31m.DS_Store [m
     [31m.ipynb_checkpoints/ [m
     [31mUntitled.ipynb [m
no changes added to commit (use "git add" and/or "git commit -a")
```

Isso nos informa que temos vários arquivos que já estamos pedindo ao Git para rastrear e que tem algumas alterações não confirmadas (as "Mudanças não testadas para confirmação" - ‘Changes not staged for commit’), bem como uma que não estamos pedindo ao Git para acompanhar (os 'arquivos não rastreados' - ‘Untracked files’). Os arquivos que o Git está rastreando são aqueles que já foram confirmados e que o Git verifica se alguma alteração foi feita desde o último **commit**.

Vamos confirmar nossas alterações mais recentes no arquivo "2016-10-04-reproducible-research-in-python.ipynb".

```
!git add '2016-10-04-reproducible-research-in-python.ipynb'
```

O Git agora colocou esse arquivo em uma fila para ser confirmado, mas precisa que nós confirmemos explicitamente o arquivo com uma mensagem.

```
!git commit -m "Completed the final section of the blog post"
```

```
[master 8f544a0] Completed the final section of the blog post
 1 file changed, 5 insertions(+), 6 deletions(-)
```

O que este comando significa é que nós confirmamos (comitted) nossas alterações no repositório local. O que precisamos agora é levar (push) nossas alterações para o repositório remoto. Praticamente toda vez que eu carrego para o repositório remoto, eu estou leevando para o **repo** de origem na *ramificação* **branch master**, o que significa que eu uso o comando abaixo:

```
!git push origin master
```

```
Counting objects: 8, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 9.43 KiB | 0 bytes/s, done.
Total 8 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local
objects. [K
To https://github.com/t-redactyl/Blog-posts.git
   83df6d4..8f544a0  master -> master
```

No entanto, se você tiver uma estrutura de **repo** mais complicada que contenha ramificações, será necessário adequar sua mensagem de carregar (push) para garantir que você se confirmou (comitted) no local correto (consulte esta postagem para obter mais detalhes).

Isso resume meu guia introdutório sobre como implementar uma pesquisa reproduzível em R - espero que se possa ver como algumas modificações simples seus fluxos de trabalho (e de seus colaboradores), pode garantir que que se está construindo projetos que sejam possíveis de revisar nos próximos anos!