RStudio と Git でバージョン管理

Shouhei TAKEUCHI

May 9, 2016

Contents

このドキュメントの目的

Windows 環境で、RStudio と Git を使って、R のコードのバージョン管理（変更履歴を残す）および共有ができるようになることを目的とする。[1] また、RStudio と Git の連携において、テキストファイル以外を使うためには、工夫が必要なので、そちらについても軽く触れておく。[2]

最終更新時の情報

- このファイルの最終更新日時：2016-05-09 17:44:06
- R のバージョン：3.2.4 Revised
- RStudio のバージョン：00.99.893
- Git のバージョン：2.7.1

Git とは？

コードを書いたファイル（テキストファイル）をバージョン管理する際に、変更のたびにファイル名を変えて履歴を管理すると、ファイル数がふくれあがっていく。Git は、そういう問題を解決するために、ファイルの「変更履歴を記録」しておくための仕組みとなる。

サルでもわかる Git 入門[3]の入門編までを読んでみると使い方の簡単な説明があり、すごく理解の助けになる。最終的には、発展編の branch まで理解しておきたい。

Git 関連用語

Git を扱う上で、理解しておくべき言葉のまとめを先にしておく。

- レポジトリ：ファイルやフォルダの状態を記録する場所
- ローカルレポジトリ：自分の手元の PC 上にあるレポジトリ
- リモートレポジトリ：Github や Bitbucket などのサーバ上にあるレポジトリ
- コミット（commit）：ファイルやフォルダの追加・変更を記録する操作
- プッシュ（push）：ローカルレポジトリの変更履歴をアップロードすること
- プル（pull）：リモートレポジトリから最新の変更履歴をダウンロードし、ローカルレポジトリに取り込むこと
- クローン（clone）：リモートレポジトリをローカルに複製すること
- ブランチ（branch）：履歴の流れを分岐して記録するもの[4]
- マージ（merge）：異なるブランチの変更を統合すること

[1] Mac は手元にないので、試せない。

[2] この資料では、R と RStudio のインストールが終わっていることを前提とする。

[3] http://www.backlog.jp/git-guide/

[4] RStudio と Git の連携では、最後にすべてマージすることが多くなると思う。

Git のインストール

Git 本家[5]から最新版（Git-2.8.2-64-bit.exe：64bit システムの場合）をダウンロードし、インストールする。オプションはよく読んで設定する。

Git のセットアップ

残念ながら、Git はインストールしてすぐ簡単に使えるものではない。最低限必要な設定などを以下にまとめておく。

Git の設定

最低限やることは以下の通り。

1. すべてのプログラム->Git->Git Bash より、以下のコマンドを用いて Git に作業者の情報を登録する。当然、メールアドレスと、名前は自分のものに変更する。名前は、半角英数のみに限定しておいた方が、余計なトラブルを起こさない。[6]

```
git config --global user.email “Git や GitHub、Bitbucket で使うメールアドレス”
git config --global user.name “Git で使う名前”
```

ここの設定を怠ると、誰がコードに変更を加えたのかがわからなくなるので、十分に注意して設定する。特に GitHub や Bitbucket などリモート環境での共有・管理を考えている場合は、登録したメールアドレスを使っておくと楽である。（さらにそもそも GitHub や Bitbucket に登録するユーザ名・メールアドレスは統一しておくと便利。）下記のコマンドで設定の確認ができるので、試しておくと良い。

```
git config --list
```

RStudio で Git を有効にする

RStudio 側でも Git を有効にする必要がある。[7]

1. RStudio で、「Tools」->「Global Options…」->「Git/SVN」を選択
2. Enable version control interface for RStudio projects にチェックを入れる。
3. Git executable: に「C:/Program Files/Git/bin/git.exe」を追加（Browse…から git.exe を探せば良い）

[5] https://git-scm.com/

[6] 個人的には、トラブルは可能な限り避けたいので、ニックネームを使って名前にもスペースを入れないようにしている。

[7] 勝手に選んでくれている時もあるが、一応確認しておく。

プロジェクトで Git を有効にする

RStudio で Git を有効にしても、勝手に全てのプロジェクトで Git が有効になるわけではない。プロジェクトごとに Git を有効にする必要がある。[8]

Git が有効な新規プロジェクトの作成

RStudio と Git の連携では、基本的にリモートレポジトリを使うことが想定されている。なので、事前に GitHub か Bitbucket にレポジトリを作成しておくとよい。

1. 「New project」->「Version Control」->「Git」を選択
2. 「Repository URL:」にリモートレポジトリの URL を貼り付ける。[9] 問題がなければ、自動で、「Project directory name:」も入力される。
3. 「Create project as subdirectory of:」に、プロジェクトのフォルダを作成するフォルダを指定する。[10]

既存のプロジェクトで Git を有効にする

1. 「Tools」->「Project Options…」->「Git/SVN」を選択
2. 「Version control system:」で「Git」を選択

この場合、リモートの指定は、Git タブの More から、Shell を開いて、以下のように行う。

```
git remote add origin “リモートレポジトリの URL”
```

この作業は、最初は複雑に思えるので、できる限り、新規にプロジェクトを作るように考えるとよい。

Git での管理

実際にコードをバージョン管理する方法に移る。

コードのバージョン管理

1. Git でバージョン管理するプロジェクトで、ファイルを作成（いつも通り）する。
2. ある程度意味をもったまとまりを変更したら、Git タブで、チェックボックスをクリックして選択し（最初の 1 回は、アイコンが A（Add）、それ以降は M（Modified）に変わる。）、Commit をクリックする。
3. 「Review Changes ウィンドウ」が立ち上がるので、右上の「Commit message」に変更についてコメントをして、チェックが付いていることを確認して「Commit」ボタンを押す。

コメントは、「1 行目：概要」、「2 行目：空行」、「3 行目：詳細な記述」が推奨されている。RStudio でのみ Git を使っていると、2 行目以降のコメントは読めないが、GitHub などリモートに Push すると見ることが出来る。

[8] もちろん、有効にしなくても普通に RStudio は使える。

[9] 例: https://github.com/takeshou/rstudiogit.git

[10] Mac だと自動では入らない上、うまく行かない場合もある。コレについては、手元に Mac がないので、調べられないが、どうにかしたいので、知っている人はコメントなどで教えていただけると助かります。

バージョンを戻す方法

コミット前に以前の状態に戻すのであれば、Git ウィンドウでファイルを選択し、Revert ボタンを押せば良い。

すでにコミットした状態であれば、下記のような手順を踏む。

1.「Review Changes ウィンドウ」左上にある「History」でコードの履歴を参照する。
2. 戻したいバージョンを選択し、右の中頃にある View file をクリック。
3. 名前を付けて保存でも、上書き保存でもする。

管理したくないファイルをコミットしてしまった場合

RStudio では Git のすべての機能がクリックで使えるわけではない。いくつかの作業は、Git タブ->More->Shell を使うか、Git Bash を使って Git の作法に従って行うか、バッドノウハウ的に行う必要がある。たとえば、Git で管理したくないファイルをコミットしてしまった場合は、以下の 2 通りが考えられる。

1. git commit –amend を使う。
2. 先に別フォルダにコピーしておいて、RStudio の Files タブで選択して、Delete ボタンを使う。

Git の作法に従うのは、正しい方法を知っていれば、何の問題もない。2 つめのバッドノウハウ的な作法は、注意が必要となる。RStudio の File タブにある Delete は、Git の管理からも削除するようになっているが、Windows のフォルダから、普通のファイルのように削除した場合、Git の管理の記録は残ったままになる。結果的に、Git には管理すべきファイルとして記録されたまま、ファイルが存在しないという状態になってしまい、厄介になことが起こってしまう。

リモートで管理する

共有・共同開発・バックアップなどいろんな用途があるが、リモートで管理する場合は、先に GitHub[11]や Bitbucket[12]の登録が必要となる。登録にはメールアドレスが必要だが、大学のアドレスを使うと、Bitbucket はアカデミックライセンスとなって便利だし、GitHub は大学のアドレスを登録した後、申請すればアカデミックライセンスになってプライベートレポジトリ（非公開）が使えるようになる。申請が面倒な場合は、Bitbucket だと最初からプライベートレポジトリが使えるので、そちらを登録すると良い。ただし、世の中的には、GitHub が標準的なものとして扱われているので、注意すること。

[11] https://github.com/
[12] https://bitbucket.org/