

Walkthrough of Setting Up a Replication Example

Lars Vilhuber and David Wasser

May 17, 2019

Walking Through An Example

- We will work through an example together
- Let's use the good example of a reproducible paper:
10.1257/pol.20140215
- Do each step on your computer
- Keep the Wiki open in another tab

Outline

- 1 Access article and download materials
- 2 Create and populate repo
- 3 Download template
- 4 Download template
- 5 The config file
- 6 Running main Stata file
- 7 Committing and pushing to repo

Outline

- 1 Access article and download materials
- 2 Create and populate repo
- 3 Download template
- 4 Download template
- 5 The config file
- 6 Running main Stata file
- 7 Committing and pushing to repo

Access article and download materials

- All of the (pre-publication) papers that need to be replicated are listed in the "Replication_List" sheet on Google Drive
- Follow the URL in the second column to the AEA page
- Click on **Data Set** (under Additional Materials)
- Fill out Entry Questionnaire

Access article and download materials

- All of the (pre-publication) papers that need to be replicated are listed in the "Replication_List" sheet on Google Drive
- Follow the URL in the second column to the AEA page
- Click on **Data Set** (under Additional Materials)
- Fill out Entry Questionnaire
- Now you do it

Outline

- 1 Access article and download materials
- 2 Create and populate repo**
- 3 Download template
- 4 Download template
- 5 The config file
- 6 Running main Stata file
- 7 Committing and pushing to repo

Create repo

- Repositories for AEA Verification are on <https://bitbucket.org/aeaverification/>
- Follow the instructions in the Wiki for creating a new repo
- **Today only:** Add your netid to the end of the repo name
- Repository name should be "[journal]-[doi]", where
 - ▶ "[journal]" may be "aej-applied", "aer", "aej-macro" etc.
 - ▶ "[doi]" will be something like "10.1257-pol.20140215"
 - ▶ (replace all spaces and "/" with "-")
- All together: "aej-policy-10.1257-pol.20140215-NETID"

Create repo

- Repositories for AEA Verification are on <https://bitbucket.org/aeaverification/>
- Follow the instructions in the Wiki for creating a new repo
- **Today only:** Add your netid to the end of the repo name
- Repository name should be "[journal]-[doi]", where
 - ▶ "[journal]" may be "aej-applied", "aer", "aej-macro" etc.
 - ▶ "[doi]" will be something like "10.1257-pol.20140215"
 - ▶ (replace all spaces and "/" with "-")
- All together: "aej-policy-10.1257-pol.20140215-NETID"
- Now you do it

Populate repo

- Follow the instructions in the Wiki for cloning a repo
- Save the downloaded materials from the paper in your repo
- Now we will use git to add, commit, and push these files to Bitbucket

Populate repo

- Follow the instructions in the Wiki for cloning a repo
- Save the downloaded materials from the paper in your repo
- Now we will use git to add, commit, and push these files to Bitbucket
- Now you do it
- In the terminal, navigate to the repo. Then:

Populate repo

- Follow the instructions in the Wiki for cloning a repo
- Save the downloaded materials from the paper in your repo
- Now we will use git to add, commit, and push these files to Bitbucket
- Now you do it
- In the terminal, navigate to the repo. Then:
 - ① Use `git add` to add the appropriate files (careful!)

Populate repo

- Follow the instructions in the Wiki for cloning a repo
- Save the downloaded materials from the paper in your repo
- Now we will use git to add, commit, and push these files to Bitbucket
- Now you do it
- In the terminal, navigate to the repo. Then:
 - ① Use `git add` to add the appropriate files (careful!)
 - ② Commit: `git commit -m "Write your commit message here"`








Populate repo

- Follow the instructions in the Wiki for cloning a repo
- Save the downloaded materials from the paper in your repo
- Now we will use git to add, commit, and push these files to Bitbucket
- Now you do it
- In the terminal, navigate to the repo. Then:
 - ① Use `git add` to add the appropriate files (careful!)
 - ② Commit: `git commit -m "Write your commit message here"`
 - ③ Push: `git push origin master`

Populate repo

- Follow the instructions in the Wiki for cloning a repo
- Save the downloaded materials from the paper in your repo
- Now we will use git to add, commit, and push these files to Bitbucket
- Now you do it
- In the terminal, navigate to the repo. Then:
 - ① Use `git add` to add the appropriate files (careful!)
 - ② Commit: `git commit -m "Write your commit message here"`
 - ③ Push: `git push origin master`
- Check Bitbucket—our files should be there now

Template looks like this

Name	Type
	Text Document
 mk_tex_table	Shell Script
 README	MD File
 REPLICATION	MD File
 SRC	MD File
 template-config	R File
 template-config	DO File








Outline

- 1 Access article and download materials
- 2 Create and populate repo
- 3 Download template
- 4 Download template
- 5 The config file
- 6 Running main Stata file
- 7 Committing and pushing to repo

Download template

- Follow the instructions in the Wiki to download the template

Template looks like this

Name	Type
	Text Document
 mk_tex_table	Shell Script
 README	MD File
 REPLICATION	MD File
 SRC	MD File
 template-config	R File
 template-config	DO File

Outline

- 1 Access article and download materials
- 2 Create and populate repo
- 3 Download template
- 4 Download template**
- 5 The config file
- 6 Running main Stata file
- 7 Committing and pushing to repo

Download template

- Save only the necessary files in the repo!
- Be sure to re-name `template-config.do` to `config.do`

Download template

- Save only the necessary files in the repo!
- Be sure to re-name `template-config.do` to `config.do`
- Now you do it

Download template

- Save only the necessary files in the repo!
- Be sure to re-name template-config.do to config.do
- Now you do it
- For this example, you should only save: REPLICATION.md, config.do, and .gitignore

Outline

- 1 Access article and download materials
- 2 Create and populate repo
- 3 Download template
- 4 Download template
- 5 The config file**
- 6 Running main Stata file
- 7 Committing and pushing to repo

The config file

- Again: make sure you re-name this!
- Add a global basepath and other paths as necessary (the paper's README should help here)

```
/* paths */  
global basepath "/path/to/project"      // change this for your specific system  
global inputdata "$basepath/inputdata"  // this is where you would read data acquired elsewhere  
global outputdata "$basepath/outputdata" // this is where you would write the data you create in this project  
global results "$basepath/tables"       // All tables for inclusion in your paper go here  
global programs "$basepath/programs"    // All programs (which you might "include") are to be found here  
global adobase "$basepath/programs/ado" // Ado packages used by the project are to be found here
```

The config file

- Again: make sure you re-name this!
- Add a global basepath and other paths as necessary (the paper's README should help here)

```
/* paths */  
global basepath "/path/to/project"      // change this for your specific system  
global inputdata "$basepath/inputdata"  // this is where you would read data acquired elsewhere  
global outputdata "$basepath/outputdata" // this is where you would write the data you create in this project  
global results "$basepath/tables"       // All tables for inclusion in your paper go here  
global programs "$basepath/programs"    // All programs (which you might "include") are to be found here  
global adobase "$basepath/programs/ado" // Ado packages used by the project are to be found here
```

- Now you do it

Outline

- 1 Access article and download materials
- 2 Create and populate repo
- 3 Download template
- 4 Download template
- 5 The config file
- 6 Running main Stata file**
- 7 Committing and pushing to repo

Running main Stata file

- Let's go back and re-read the README carefully

DO FILE

The do-file can recreate all tables in the main paper and in the online appendix. The only instruction to run the do-file is to change the file directory in **line 51**. To output the tables you will have to create a folder called “Output” in the directory from which you are calling the datafile.

Please note that in order to see Table 1, you need to stop the do file on line 128 (when the table will output in Stata itself).

Running main Stata file

- Let's go back and re-read the README carefully

DO FILE

The do-file can recreate all tables in the main paper and in the online appendix. The only instruction to run the do-file is to change the file directory in **line 51**. To output the tables you will have to create a folder called “Output” in the directory from which you are calling the datafile.

Please note that in order to see Table 1, you need to stop the do file on line 128 (when the table will output in Stata itself).

- What should we do?

Running main Stata file

- Let's go back and re-read the README carefully

DO FILE

The do-file can recreate all tables in the main paper and in the online appendix. The only instruction to run the do-file is to change the file directory in **line 51**. To output the tables you will have to create a folder called "Output" in the directory from which you are calling the datafile.

Please note that in order to see Table 1, you need to stop the do file on line 128 (when the table will output in Stata itself).

- What should we do?
 - 1 Create an "Output" directory in the repo and add it to config.do
 - 2 Add `include config.do` to the top of the code

Running main Stata file

- Let's go back and re-read the README carefully

DO FILE

The do-file can recreate all tables in the main paper and in the online appendix. The only instruction to run the do-file is to change the file directory in **line 51**. To output the tables you will have to create a folder called "Output" in the directory from which you are calling the datafile.

Please note that in order to see Table 1, you need to stop the do file on line 128 (when the table will output in Stata itself).

- What should we do?
 - 1 Create an "Output" directory in the repo and add it to config.do
 - 2 Add `include config.do` to the top of the code
- What about the note on Table 1?

Outline

- 1 Access article and download materials
- 2 Create and populate repo
- 3 Download template
- 4 Download template
- 5 The config file
- 6 Running main Stata file
- 7 Committing and pushing to repo

Committing and pushing to repo

- Before running the code, we should `git commit` again.
- After running the code, we will write our report and then `git commit` and `git push` again.
- Always remember: commit frequently, push (at least) daily