Reproducibilidad cuando los datos son confidenciales

Lars Vilhuber

2025-06-01

Siga-me

labordynamicsinstitute.github.io/reproducibility-confidential/presentation/es (PDF)

Reproducibilidad cuando los datos son confidenciales

Las revistas requieren que **compartas tu código y datos** en un paquete de replicación al final de tu proyecto de investigación.

Seguir algunas mejores prácticas desde el día 1 no solo puede **ayudarte a preparar** este paquete más tarde, sino también hacerte **investigadores más productivos**.

Seguir algunas mejores prácticas antes de liberar un paquete puede **evitar revisiones costosas**.

Aparte

Al escribir

> Seguir algunas mejores prácticas antes de liberar un paquete puede **evitar revisiones costosas**.

mi IA de codificación sugirió que agregue

> “y retracciones embarazosas”…

¿Qué es un paquete de replicación?

- Política de Disponibilidad de Datos y Código de AEA
- Estándar de Disponibilidad de Datos y Código 

- Repositorio de Datos y Código de AEA

Ejemplo de depósito

Data and Code for: "Indirect Savings from Public Procurement Centralization"

Principal Investigator(s): Clarissa Lotti, Lear; Arieda Muço, Central European University; Giancarlo Spagnolo, Site - Stockholm School of Economics; Tommaso Valletti, Imperial College London

Version: V1

AMERICAN ECONOMIC ASSOCIATION

Name	File Type	Size	Last Modified
code			06/18/2024 01:15:PM
data			06/18/2024 01:16:PM
output			06/18/2024 01:14:PM
CITATION.CFF	text/plain	862 bytes	06/18/2024 09:14:AM
LICENSE.txt	text/plain	1.2 KB	06/18/2024 09:14:AM
README.md	text/x-web-markdown	6 KB	06/18/2024 09:14:AM
main.sh	application/x-sh	2.4 KB	06/18/2024 09:14:AM

DOWNLOAD THIS FOLDER

Usage Metrics

Overall Project Metrics

14	3	3
Views	Downloads	Publications

Folder/File-Level Metrics

0	0
Views	Downloads

Download Detailed Metrics

Política de AEA

AMERICAN ECONOMIC ASSOCIATION

Membership About AEA Log In

Journals Annual Meeting Careers Resources EconLit Committees Ethics/Ombuds

Home › Journals › AEA Data and Code Policies and Guidance › Data and Code Availability Policy

Journals

American Economic Review

AER: Insights

AEJ: Applied Economics

AEJ: Economic Policy

AEJ: Macroeconomics

AEJ: Microeconomics

Journal of Economic Literature

Journal of Economic Perspectives

Data and Code Availability Policy

It is the policy of the American Economic Association to publish papers only if the data and code used in the analysis are clearly and precisely documented and access to the data and code is nonexclusive to the authors.

Authors of accepted papers that contain empirical work, simulations, or experimental work must provide, prior to acceptance, information about the data, programs, and other details of the computations sufficient to permit replication, as well as information about access to data and programs.

The Editor should be notified at the time of submission if access to the data used in a paper is restricted or limited or if, for some other reason, the requirements above cannot be met.

If data or programs cannot be published in an openly accessible trusted data repository, authors must commit to preserving data and code for a period of no less than five years following publication of the manuscript and to providing reasonable assistance to requests for clarification and replication.

Objetivo

- [ ] Proporcionar orientación sobre la estructura de paquetes de replicación cuando los datos son confidenciales
- [ ] Proporcionar orientación sobre documentación
- [ ] Mantenerlo simple

El paquete de replicación final

Archivos

Contenido de un paquete

```
README.md
README.pdf
code/
  fsrdc/
    01-prepare-data.R
    02-analyze-data.R
    03-create-disclosable-data.R
  public/
    04-create-tables.do
    05-create-figures.do
    06-create-intext.do
data/
  public/
    dist_cepii.dta
    usa_00010.dta
run.sh
```

Archivos

Todo el código, ya sea usado en RDC o no

```
README.md
README.pdf
code/
  fsrdc/
    01-prepare-data.R
    02-analyze-data.R
    03-create-disclosable-data.R
  public/
    04-create-tables.do
    05-create-figures.do
    06-create-intext.do
data/
  public/
    dist_cepii.dta
    usa_00010.dta
run.sh
```

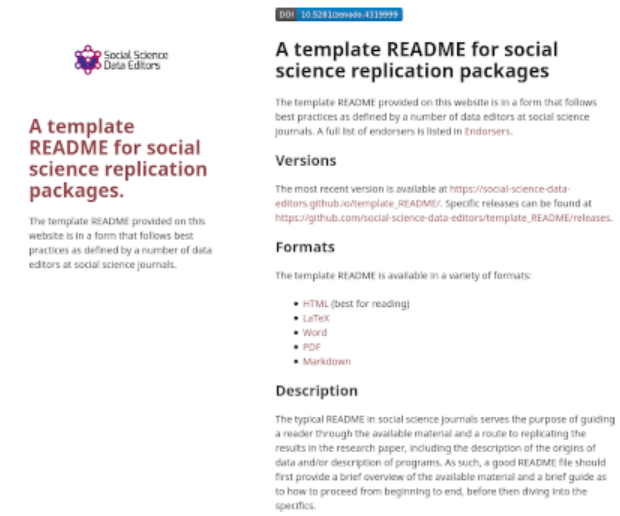
Archivos

Todos los datos públicos, ya sean usados en RDC o no

```
README.md
README.pdf
code/
  fsrdc/
    01-prepare-data.R
    02-analyze-data.R
    03-create-disclosable-data.R
  public/
    04-create-tables.do
    05-create-figures.do
    06-create-intext.do
data/
  public/
    dist_cepii.dta
    usa_00010.dta
run.sh
```

Contenido

Descripción completa según la (plantilla) README



README

El README

Tres partes del README

- Disponibilidad de datos (y citas)
- Requisitos del computador
- Descripción del procesamiento

Comience con la última parte

Eso es fácil: has estado manteniendo instrucciones claras desde el principio, ¿verdad?

- Ejecutar "`main.do`" o `run.sh`
- Describir qué partes podrían omitirse
- Describir qué hacen las diversas partes
- Describir qué partes usan datos confidenciales

¡Has estado haciendo eso **desde el día 1!**

Requisitos del computador

En la mayoría de entornos confidenciales, como FSRDC/IRE, esta parte está fuera de tu control. ¡Pero descríbela de todos modos!

Requisitos del computador

- Descripción aproximada de computadoras/nodos utilizados
  - tamaño de memoria (¡pero interesado en el uso real, no en el máximo de lo que tiene el sistema!)
  - ¡tiempo de cómputo! ¿Cuánto tiempo toma una ejecución limpia, de arriba a abajo?
  - número de nodos: ¿algún procesamiento paralelo?
- Software
  - Versión del software (Stata 17, nivel de actualización)
  - ¡Todos los paquetes! Idealmente, versión del paquete

Requisitos del computador (FSRDC)

- ¿Utilizaste PBS? Claro que sí.

¡Incluye los archivos `qsub`! (O si utilizaste `qstata` o similar, describe eso).

```
...
run.sh
qsub-complete.sh
```

Disponibilidad de datos

- Esto es fácil: ¡son los datos que solicitaste tener incluidos en tu proyecto FSRDC!
- ¡Así que tenías esta información desde el **Día -90** del proyecto!

Disponibilidad de datos redux

Para describir la disponibilidad de datos, divídela en dos:

- cómo OBTUVISTE acceso a los datos (eso es antiguo)
- cómo pueden OTROS obtener acceso a los mismos datos (¡eso podría ser diferente!)
- Los dos no siempre son lo mismo, pero ambos son relevantes.

Ejemplos

Los ejemplos incluyen

- esta excelente descripción de un artículo de Teresa Fort (ReStud):

Ejemplos

Los ejemplos incluyen

- esta excelente descripción de un artículo de Teresa Fort (ReStud):

> 1. Todos los resultados en el artículo utilizan microdatos confidenciales de la Oficina del Censo de EE.UU. Para obtener acceso a los microdatos del Censo, sigue las instrucciones aquí sobre cómo escribir una propuesta para acceso a los datos a través de un Centro Federal de Datos de Investigación Estadística: https://www.census.gov/ces/rdcresearch/howtoapply.html.
>
> •

Ejemplos

Los ejemplos incluyen

- esta excelente descripción de un artículo de Teresa Fort (ReStud):

> 2. Debes solicitar los siguientes conjuntos de datos en tu propuesta:
>
> - Base de Datos Longitudinal de Negocios (LBD), 2002 y 2007
> - Base de Datos de Comercio Exterior – Importación (IMP), 2002 y 2007
> - Encuesta Anual de Manufactura (ASM), incluyendo el Suplemento de Uso de Redes de Computadoras (CNUS), 1999
> - […]
> - Encuesta Anual de Insumos Mágicos (ASMI), 2002 y 2007

Ejemplos

Los ejemplos incluyen

- esta excelente descripción de un artículo de Teresa Fort (ReStud):

> 3. Referencia
>
> - “Tecnología y Fragmentación de la Producción: Abastecimiento Doméstico versus Extranjero” por Teresa Fort, número de proyecto br1179 en la propuesta. Esto te dará acceso a los programas y conjuntos de datos de entrada requeridos para reproducir los resultados. Solicitar una búsqueda de archivos con el DOI del artículo (“10.1093/restud/rdw057”) debería generar los mismos resultados.
> -

Ejemplos

Los ejemplos incluyen

- esta excelente descripción de un artículo de Teresa Fort (ReStud):

> NOTA: Los archivos relacionados con el proyecto están disponibles por 10 años a partir de 2015.

Ejemplos

Los ejemplos incluyen

- esta descripción por Fadlon y Nielsen sobre datos daneses

> La información utilizada en el análisis combina varios registros administrativos daneses (como se describe en el artículo). El uso de datos está sujeto al Reglamento General de Protección de Datos (GDPR) de la Unión Europea según las nuevas regulaciones danesas de mayo de 2018. Los datos se almacenan físicamente en computadoras en Statistics Denmark y, debido a consideraciones de seguridad, los datos no pueden transferirse a computadoras fuera de Statistics Denmark.

Ejemplos

Los ejemplos incluyen

- esta descripción por Fadlon y Nielsen sobre datos daneses

> Los investigadores interesados en obtener acceso a los datos de registro empleados en este artículo deben presentar una solicitud escrita para obtener la aprobación de Statistics Denmark. La solicitud debe incluir una descripción detallada del proyecto propuesto, su propósito y su contribución social, así como una descripción de los conjuntos de datos requeridos, variables y población de análisis.

Ejemplos

Los ejemplos incluyen

- esta descripción por Fadlon y Nielsen sobre datos daneses

> Las solicitudes pueden ser presentadas por investigadores que estén afiliados con instituciones danesas aceptadas por Statistics Denmark, o por investigadores fuera de Dinamarca que colaboren con investigadores afiliados con estas instituciones.

(Ejemplo tomado de Fadlon y Nielsen, AEJ:Applied 2021).

Ejemplos

También otorga permiso a los archivos de tu proyecto:

> Otorgo a cualquier investigador con el permiso de proyecto aprobado por el Censo apropiado el uso de mis archivos de investigación exactos siempre que esos archivos estuvieran entre los que solicitaron cuando se obtuvo la aprobación (un requisito de la Oficina del Censo). Estos archivos se pueden encontrar buscando el DOI de [este archivo/ este artículo] entre las copias de seguridad/archivos hechos en [mes del archivo].

No olvides citar los datos

> Bureau of the Census. (año de publicación). American Community Survey-Master Address File Crosswalk YYYY-YYZZ [Archivo de Datos]. Federal Statistical Research Data Center [distribuidor].
>
> Graf, Tobias; Grießemer, Stephan; Köhler, Markus; Lehnert, Claudia; Moczall, Andreas; Oertel, Martina; Schmucker, Alexandra; Schneider, Andreas; Seth, Stefan; Thomsen, Ulrich; vom Berge, Philipp (2023): "Versión débilmente anónima de la Muestra de Biografías Integradas del Mercado Laboral (SIAB) – Versión 7521 v1". Centro de Datos de Investigación de la Agencia Federal de Empleo (BA) en el Instituto de Investigación del Empleo (IAB). https:/doi.org/10.5164/IAB.SIAB7521.de.en.v1

- Más ejemplos en Zotero para FSRDC (posiblemente no el más actual).
- Idealmente, cada centro de datos de investigación tendría "páginas de destino" para los datos (el ejemplo del IAB sí las tiene)

Tres partes del README: cronología

- Disponibilidad de datos (y citas):	**Inicio del proyecto, editar al final**
- Requisitos del computador:	Mitad del proyecto
- Descripción del procesamiento:	Mitad del proyecto

con el final realmente solo una última lectura/edición.

Entornos en Stata

TL;DR

- Crear entornos virtuales en Stata es factible
- Hacerlo estabiliza el código y lo hace más transportable

Rutas de búsqueda en Stata

En Stata, típicamente no hablamos de entornos, pero la misma estructura básica aplica: Stata busca a lo largo de un orden establecido para sus comandos.

Rutas de búsqueda en Stata

Algunos comandos están incorporados en el ejecutable (el software que se abre cuando haces clic en el icono de Stata), pero la mayoría de otros comandos internos, y todos los comandos externos, se encuentran en una ruta de búsqueda.

Los directorios `sysdir`

El conjunto predeterminado de directorios que se pueden buscar, desde un Stata recién instalado, se puede consultar con el comando `sysdir`, y se verá algo así:

```
1  sysdir
```

```
1     STATA:  C:\Program Files\Stata18\
2      BASE:  C:\Program Files\Stata18\ado\base\
3      SITE:  C:\Program Files\Stata18\ado\site\
4      PLUS:  C:\Users\lv39\ado\plus\
5  PERSONAL:  C:\Users\lv39\ado\personal\
6  OLDPLACE:  c:\ado\
```

El orden de búsqueda `adopath`

Las rutas de búsqueda donde Stata busca comandos se consulta por `adopath`, y se ve similar, pero ahora tiene un orden asignado a cada entrada:

```
adopath
```

```
[1]   (BASE)       "C:\Program Files\Stata18\ado\base/"
[2]   (SITE)       "C:\Program Files\Stata18\ado\site/"
[3]                "."
[4]   (PERSONAL)   "C:\Users\lv39\ado\personal/"
[5]   (PLUS)       "C:\Users\lv39\ado\plus/"
[6]   (OLDPLACE)   "c:\ado/"
```

La ruta en funcionamiento

Para buscar un comando, Stata buscará en el primer directorio, luego en el segundo, y así sucesivamente, hasta que lo encuentre. Si no lo encuentra, devolverá un error.

```
1  which reghdfe
```

```
command reghdfe not found as either built-in or ado-file
r(111);
```

¿Dónde se instalan los paquetes?

Cuando instalamos un paquete (`net install`, `ssc install`)[1], solo una de las rutas (`sysdir`) es relevante: `PLUS`.

```
1    [1]   (BASE)       "C:\Program Files\Stata1
2    [2]   (SITE)       "C:\Program Files\Stata1
3    [3]                "."
4    [4]   (PERSONAL)   "C:\Users\lv39\ado\perso
5    [5]   (PLUS)       "C:\Users\lv39\ado\plus/
6    [6]   (OLDPLACE)   "c:\ado/"
```

Instalando paquetes

```
1  ssc install reghdfe
2  which reghdfe
```

```
1  . ssc install reghdfe
2  checking reghdfe consistency and verifying not already install
3  installing into C:\Users\lv39\ado\plus\...
4  installation complete.
5
6  . which reghdfe
7  C:\Users\lv39\ado\plus\r\reghdfe.ado
8  *! version 6.12.3 08aug2023
```

Usando entornos en Stata

Pero el directorio `(PLUS)` puede ser manipulado

```
* Establecer el directorio raíz
global rootdir : pwd
* Definir una ubicación donde mantendremos
global adodir "$rootdir/ado"
* asegurarse de que exista, si no crearlo.
cap mkdir "$adodir"
* Ahora simplifiquemos el adopath
* - eliminar las rutas OLDPLACE y PERSONAL
* - ¡NUNCA ELIMINAR LAS RUTAS DEL SISTEMA -
adopath - OLDPLACE
adopath - PERSONAL
* modificar la ruta PLUS para que apunte a
sysdir set PLUS "$adodir"
adopath ++ PLUS
* verificar la ruta
adopath
```

Usando entornos en Stata

```
* Establecer el directorio raíz
global rootdir : pwd
* Definir una ubicación donde mantendrer
global adodir "$rootdir/ado"
* asegurarse de que exista, si no crear
cap mkdir "$adodir"
* Ahora simplifiquemos el adopath
* - eliminar las rutas OLDPLACE y PERSON
* - ¡NUNCA ELIMINAR LAS RUTAS DEL SISTEN
adopath - OLDPLACE
adopath - PERSONAL
* modificar la ruta PLUS para que apunte
sysdir set PLUS "$adodir"
adopath ++ PLUS
* verificar la ruta
adopath
```

```
. adopath
    [1]   (PLUS)       "C:\Users\lv39\Documents/PROJECT123/ado/"
    [2]   (BASE)       "C:\Program Files\Stata18\ado\base/"
    [3]   (SITE)       "C:\Program Files\Stata18\ado\site/"
    [4]                "."
```

Usando entornos en Stata

Vamos a verificar nuevamente dónde está el paquete `reghdfe`:

```
1  which reghdfe
```

```
1  . which reghdfe
2  command reghdfe not found as either built-i
3  r(111);
```

Usando entornos en Stata

Así que ya no se encuentra. ¿Por qué? Porque hemos eliminado la ubicación anterior (la antigua ruta `PLUS`) de la secuencia de búsqueda. Es como si no existiera.

Anteriormente:

```
. which reghdfe
C:\Users\lv39\ado\plus\r\reghdfe.ado
*! version 6.12.3 08aug2023
```

```
. adopath
  [1]  (PLUS)      "C:\Users\lv39\Documents/PROJECT
  [2]  (BASE)      "C:\Program Files\Stata18\ado\ba
  [3]  (SITE)      "C:\Program Files\Stata18\ado\si
  [4]              "."
```

Instalando paquetes cuando un entorno está activo

Cuando ahora instalamos `reghdfe` nuevamente:

```
. ssc install reghdfe
checking reghdfe consistency and verifying not already installed...
installing into C:\Users\lv39\Documents\PROJECT123\ado\plus\...
installation complete.

. which reghdfe
C:\Users\lv39\Documents\PROJECT123\ado\plus\r\reghdfe.ado
*! version 6.12.3 08aug2023
```

Ahora lo vemos en el directorio **específico del proyecto**, que podemos distribuir con todo el proyecto.

Instalando versiones precisas de paquetes de Stata

Imaginemos que necesitamos una versión anterior de `reghdfe`.

- En general, **no** es posible en Stata instalar una versión anterior de un paquete de manera directa.
- *Puedes* tener éxito con el archivo Wayback Machine de SSC.

Repositorios de paquetes

La mayoría de repositorios de paquetes tienen versiones:

- R: CRAN, Bioconductor
- Python: PyPI
- Julia: registro de paquetes Julia predeterminado “General”.

Stata no lo hace (a partir de 2024). **Pero** ve el sitio completo para un enfoque.

Conclusiones

De los desiderata anteriores de *entornos*:

- ✅ **Aislado**: Instalar un paquete nuevo o actualizado para un proyecto no romperá tus otros proyectos, y viceversa.
- ✅ **Portable**: Transportar fácilmente tus proyectos de una computadora a otra, *incluso a través de diferentes plataformas*.
- ❌ **Reproducible**: Registra las versiones exactas de paquetes de las que dependes, y asegura que esas versiones exactas sean las que se instalen donde quiera que vayas.

Conclusiones

- ☑ tu código funciona sin problema, después de toda la depuración.
- ☑ tu código funciona sin intervención manual, y con poco esfuerzo
- ☑ realmente produce todas las salidas
- ☑ tu código genera un archivo de registro que puedes inspeccionar, y que podrías compartir con otros.
- ☑ ❓ funcionará en la computadora de otra persona

Secretos en el código

¿Qué son los secretos?

- Claves API
- Credenciales de inicio de sesión para acceso a datos
- Rutas de archivos (¡FSRDC!)
- Nombres de variables (¡IRS!)

Práctica estándar

Almacenar secretos en variables de entorno o archivos que no se publican.

Algunos servicios se toman esto en serio

Escaneo de secretos de Github

Dónde almacenar secretos

- **variables de entorno**
- archivos “dot-env” (Python), archivos “Renviron” (R)
- o algún otro archivo claramente identificado en el proyecto o directorio de inicio

Variables de entorno

Escritas interactivamente (aquí para Linux y Mac)

```
1 MYSECRET="dfad89ald"
2 CONFDATALOC="/path/to/irs/files"
```

(esto **no** se recomienda)

Almacenar estos en archivos

La misma sintaxis se usa para el contenido de archivos “dot-env” o “Renviron”, y de hecho archivos de inicio de `bash` o `zsh` (`.bash_profile`, `.zshrc`)

Usar en R

Editar archivos `.Renviron` (¡nota el punto!):

```r
# Editar Renviron global (personal)
usethis::edit_r_environ()
# También puedes considerar crear configuraciones específicas del proyecto:
usethis::edit_r_environ(scope = "project")
```

Usar las variables definidas en `.Renviron`:

```r
mysecret <- Sys.getenv('MYSECRET')
```

Usar en Python

Cargar variables de entorno regulares:

```
1 import os
2 mysecret = os.getenv("MYSECRET")  # cargará variables de entorno
```

Cargar con `dotenv`

```
1 from dotenv import load_dotenv
2 load_dotenv()  # tomar variables de entorno del proyecto .env.
3 mysecret = os.getenv("MYSECRET")  # cargará variables de entorno
```

Usar en Stata

Sí, esto también funciona en Stata

```
// cargar desde entorno
global mysecret : env MYSECRET
display "$mysecret"  // en realidad no hagas esto en el código
```

y a través de (¿qué más?) un paquete escrito por usuario para cargar desde archivos:

```
net install doenv, from(https://github.com/vikjam/doenv/raw/master/)
doenv using ".env"
global mysecret "`r(MYSECRET)'"
display "$mysecret"
```

Solución más simple

```
1  //============ parámetros no confidenciales =========
2  include "config.do"
3
4  //============ parámetros confidenciales =============
5  capture confirm file "$code/confidential/confparms.do"
6  if _rc == 0 {
7      // el archivo existe
8      include "$code/confidential/confparms.do"
9  } else {
10     di in red "No se encontraron parámetros confidenciales"
11 }
12 //============ fin parámetros confidenciales =========
```

¿Código confidencial?

¿Qué es código confidencial, dices?

- En Estados Unidos, algunas **variables en bases de datos del IRS** se consideran super-ultra-secretas. Así que no puedes nombrar esa-variable-que-llenaste-en-tu-Formulario-1040 en tu código de análisis de los mismos datos. (A menudo se les refiere en jerga como “variables Título 26”).

¿Qué es código confidencial, dices?

- Tu código contiene la **semilla aleatoria que usaste para anonimizar** los identificadores sensibles. Esto podría permitir hacer ingeniería inversa de la anonimización, y no es buena idea publicar.

¿Qué es código confidencial, dices?

- Usaste una **tabla de búsqueda codificada directamente** en tu código Stata para anonimizar los identificadores sensibles (`replace anoncounty=1 if county="Tompkins, NY"`).

Una **muy mala idea**, pero sí, probablemente quieres ocultar eso.

¿Qué es código confidencial, dices?

- Tu especialista en TI u oficial de divulgación piensa que publicar la **ruta exacta** a tu copia de los datos confidenciales del Censo 2010, ej., “/data/census/2010”, es un riesgo de seguridad y se niega a dejar pasar ese código.

¿Qué es código confidencial, dices?

- Has cumplido con las reglas de divulgación, pero por alguna razón, el tamaño mínimo preciso de celda es un parámetro confidencial.

¿Qué es código confidencial, dices?

Así que ya sea razonable o no, **esto es un problema**. ¿Cómo haces eso, sin arruinar el código, o gastar horas redactando tu código?

Ejemplo

- Esto servirá como ejemplo. Nada de esto es específico a Stata, y las soluciones para R, Python, Julia, Matlab, etc. son todas bastante similares.
- Asume que las variables `q2f` y `q3e` se consideran confidenciales por alguna regla, y que el tamaño mínimo de celda `10` también es confidencial.

```
set seed 12345
use q2f q3e county using "/data/economic/cmf2012/extract.dta", clear
gen logprofit = log(q2f)
by county: collapse (count)  n=q3e (mean) logprofit
drop if n<10
graph twoway n logprofit
```

Ejemplo

Solo una línea que no contiene información “confidencial”.

```
set seed 12345
use q2f q3e county using "/data/economic/cmf2012/extract.dta", clear
gen logprofit = log(q2f)
by county: collapse (count)  n=q3e (mean) logprofit
drop if n<10
graph twoway n logprofit
```

No hagas esto

Un mal ejemplo, porque literalmente hace más trabajo para ti y para futuros replicadores, es redactar manualmente la información confidencial con texto que no es código legítimo:

```
set seed NNNNN
use <vars removidas> county using "<ruta removida>", clear
gen logprofit = log(XXXX)
by county: collapse (count)  n=XXXX (mean) logprofit
drop if n<XXXX
graph twoway n logprofit
```

El programa redactado arriba ya no funcionará, y será muy tedioso des-redactar si un replicador posterior obtiene acceso legítimo a los datos confidenciales.

Mejor

Simplemente reemplazar los datos confidenciales con reemplazo que son marcadores de posición válidos en el lenguaje de programación de tu elección ya es mejor. Aquí está la versión confidencial del archivo:

```
//============ parámetros confidenciales =============
global confseed    12345
global confpath    "/data/economic/cmf2012"
global confprofit  q2f
global confemploy  q3e
global confmincell 10
//============ fin parámetros confidenciales =========
set seed $confseed
use $confprofit county using "${confpath}/extract.dta", clear
gen logprofit = log($confprofit)
by county: collapse (count)  n=$confemploy (mean) logprofit
drop if n<$confmincell
graph twoway n logprofit
```

Mejor

y este podría ser el archivo liberado, parte del paquete de replicación:

```
1  //============ parámetros confidenciales =============
2  global confseed    XXXX    // un número
3  global confpath    "XXXX"  // una ruta que te será comunicada
4  global confprofit  XXX     // Nombre de variable para ganancia T26
5  global confemploy  XXX     // Nombre de variable para empleo T26
6  global confmincell XXX     // un número
7  //============ fin parámetros confidenciales =========
8  set seed $confseed
9  use $confprofit county using "${confpath}/extract.dta", clear
10 gen logprofit = log($confprofit)
11 by county: collapse (count)  n=$confemploy (mean) logprofit
12 drop if n<$confmincell
13 graph twoway n logprofit
```

Aunque el código no funcionará tal como está, es fácil des-redactar, independientemente de cuántas veces referencien los valores confidenciales, ej., `q2f`, en cualquier lugar del código.

Mejor

- Archivo principal
- Procesamiento condicional
- Archivo separado para parámetros confidenciales que puede simplemente ser excluido de la solicitud de divulgación

Mejor

Archivo principal `main.do`:

```
//============ parámetros confidenciales =============
capture confirm file "$code/confidential/confparms.do"
if _rc == 0 {
    // el archivo existe
    include "$code/confidential/confparms.do""
} else {
    di in red "No se encontraron parámetros confidenciales"
}
//============ fin parámetros confidenciales =========

//============ parámetros no confidenciales =========
global safepath "$rootdir/releasable"
cap mkdir "$safepath"

//============ fin parámetros ======================
```

Mejor

Archivo principal `main.do` (continuado)

```
// ::::  Procesar solo si los datos confidenciales están presentes

capture confirm  file "${confpath}/extract.dta"
if _rc == 0 {
   set seed $confseed
   use $confprofit county using "${confpath}/extract.dta", clear
   gen logprofit = log($confprofit)
   by county: collapse (count)  n=$confemploy (mean) logprofit
   drop if n<$confmincell
   save "${safepath}/figure1.dta", replace
} else { di in red "Omitiendo procesamiento de datos confidenciales" }

//============ en este punto, los datos son liberables ======
// ::::  Procesar siempre

use "${safepath}/figure1.dta", clear
graph twoway n logprofit
graph export "${safepath}/figure1.pdf", replace
```

Mejor

Archivo auxiliar `$code/confidential/confparms.do"` (no liberado)

```
1 //============ parámetros confidenciales =============
2 global confseed    12345
3 global confpath    "/data/economic/cmf2012"
4 global confprofit  q2f
5 global confemploy  q3e
6 global confmincell 10
7 //============ fin parámetros confidenciales =========
```

Mejor

Archivo auxiliar `$code/include/confparms_template.do` (esto se libera)

```
1 //============ parámetros confidenciales =============
2 // Copia este archivo a $code/confidential/confparms.do y edita
3 global confseed    XXXX    // un número
4 global confpath    "XXXX"  // una ruta que te será comunicada
5 global confprofit  XXX     // Nombre de variable para ganancia T26
6 global confemploy  XXX     // Nombre de variable para empleo T26
7 global confmincell XXX     // un número
8 //============ fin parámetros confidenciales =========
```

Mejor paquete de replicación

Así, el paquete de replicación tendría:

```
...
code/main.do
README.md
include/confparms_template.do
releasable/figure1.dta
releasable/figure1.pdf
```

Evitando datos confidenciales en tu código

El problema

A menudo vemos código que “arregla” problemas en los datos codificando directamente un mapeo:

```
# ... 1000 líneas de código arriba...
# Mala práctica
data$name[data$name == "Joe Biden"] <- "Joseph Robinette Biden Jr."
data$county[data$county == "Tompins, NY"] <- "Tompkins County, NY"
# ... 500 líneas de código abajo ...
```

¿Por qué es esto un problema?

La información en las columnas `name` o `county` podría ser confidencial.

¡Al codificar esta información como parte de tus programas, has hecho el **código** confidencial!

- Ahora podrías tener que redactar el código antes de liberarlo

Una solución

Como antes, podrías mover este código a un archivo separado:

```
# ... 1000 líneas de código arriba...
# Mejor práctica
source("confidential/mappings.R")
# ... 500 líneas de código abajo ...
```

Mejor solución

Si te das cuenta de que el mapeo es en realidad **datos**, entonces tratarlo como cualquier otro dato (mucho del cual también podría ser confidencial) es tanto

- *más robusto* y
- *más manejable*

mientras es *seguro*.

Mejor solución

```
1
2  if (!file.exists("data/confidential/names_mapping.csv")) {
3      names_confidential %>%
4          left_join(read_csv("data/confidential/names_mapping.csv"), by = "na
5          # reemplazar name con name_alt si este último no es NA
6          mutate(name = if_else(!is.na(name_alt), name_alt, name)) %>%
7          # eliminar la columna name_alt
8          select(-name_alt) -> names_clean
9  }
```

Nota

- ¡Todavía podrías querer des-identificar los datos antes de liberarlos!
- El código, sin embargo, ahora está **libre de información confidencial**.

Ejemplo de tutorial

- Ve código R de muestra en este repositorio de Github para un ejemplo donde tratamos los nombres de presidentes como datos confidenciales.

Resumiendo todo

Resumiendo

- El paquete de replicación público contiene código inteligible, omite detalles confidenciales (pero proporciona código de plantilla), tiene declaraciones detalladas de procedencia de datos
- El paquete de replicación confidencial contiene todo lo mismo, más el código confidencial, está archivado en el FSRDC

Cosas para recordar

- Usar código para guardar figuras y tablas (`estout`, `graph export`, `regsave`)
- Crear archivos de registro para cada ejecución (`stata -b do file.do` no es lo suficientemente granular) enlace

Cosas para recordar

¡Ejecutarlo todo de nuevo, de arriba a abajo!

Cosas para recordar

- Al hacer una solicitud de revisión de divulgación, recuerda solicitar el **código**
- Al producir estadísticas, *considera las reglas de divulgación* - menos cambios, más rápido el resultado (en teoría), pero en particular menos sorpresas
- No pienses "*nadie leerá jamás este código*" - ¡es muy probable que alguien lo haga!

Fin

¡Ahora esperas a que aparezcan los replicadores!

Apéndice

Mantenerse al día con la procedencia

- Licencias
- Simplificación para la reproducibilidad

Licencias

¿De dónde viene el archivo?

- ¿Cómo podemos describir esto más tarde a alguien?
  - Apuntar y hacer clic es largo de describir
  - ¿Cuáles son los derechos que tenemos?

¿Qué es una licencia?

> Una licencia es un permiso o autorización oficial para hacer, usar o poseer algo (así como el documento de ese permiso o autorización).[2] [3]

Ejemplos

- Licencias Creative Commons, utilizadas para productos artísticos y datos
- Licencias de código abierto (BSD, GPL, MIT, etc.), utilizadas para software (código)

Licencia aplicada a los datos Geodist

- CEPII GeoDist está bajo una “licencia Etalab 2.0”

¿Podemos republicar el archivo?

Descarga a través de código

Lo más fácil:

Stata

```
1 use "$URL" , clear
```

¿Por qué no?

- ¿estará allí en dos meses? ¿en 6 años?
- ¿qué pasa si la conexión a internet está caída?

Fácil:

Stata

```
global URL "https://www.cepii.fr/distance/dist_cepii.dta"
copy "$URL" (outputfile), replace
```

R

```
download.file(url="$URL",destfile="(outputfile)")
```

Llegaremos a métodos aún mejores un poco más tarde

Creando un README

- README de plantilla
  - Citar tanto el conjunto de datos como el documento de trabajo
  - Agregar URL de datos y tiempo de acceso (¿puedes pensar en una forma de automatizar esto?)
  - Agregar un enlace a la licencia (también: descargar y almacenar la licencia)

Enlace

Paso 1: Stata, R [4]

Enlaces

Orientación

Se puede encontrar orientación adicional en el sitio web de los Editores de Datos de Ciencias Sociales (URLs sujetas a cambio):

- https://social-science-data-editors.github.io/guidance/DCAS_Restricted_data.html#us-census-bureau-and-fsrdc
- https://social-science-data-editors.github.io/guidance/Requested_information_hosting.html#generated-repositories

Recursos de entrenamiento adicionales

- “Tutorial del Día 1”: Presentado el 12 de septiembre de 2024 en la conferencia FSRDC (pre-programa), sujeto a cambios: https://larsvilhuber.github.io/day1-tutorial/
- Orientación de propósito general sobre “auto verificación” de tu paquete de reproducibilidad: https://larsvilhuber.github.io/self-checking-reproducibility/

Ejemplos de paquetes de replicación

- https://doi.org/10.3886/E154241V2 no solo código, sino que enfrenta el problema de que los datos del IRS no pueden tener variables reveladas. Su solución no es la misma que en este tutorial.
- https://doi.org/10.3886/E162581V1

- Fuente de esta presentación: labordynamicsinstitute/reproducibility-confidential
- Con licencia bajo 

Footnotes

1.

Referencia de `net install`. Estrictamente hablando, la ubicación donde se instalan los paquetes ado se puede cambiar a través del comando `net set ado`, pero esto rara vez se hace en la práctica, y no lo haremos aquí.

2. Cambridge Dictionary

3. Wikipedia

4. 🔒 Tag: stage1