



ADRF User Guide



1. ADRF User Onboarding Handbook	2
1.1 Obtaining ADRF Access and Account Set Up	2
1.2 Onboarding Modules and Security Training	2
1.3 Do's and Don'ts For Discussing Data Inside the ADRF	4
1.4 Accessing and Using the Workspace	4
1.5 Accessing Your Data	15
1.6 Storing Analytic Results	23
1.7 Sharing Information within the ADRF	24
1.8 Exporting Results	24
1.9 SVN Documentation	28
1.10 Technical Support	36
1.11 References	36

ADRF User Onboarding Handbook

Obtaining ADRF Access and Account Set Up

Onboarding Modules and Security Training

Do's and Don'ts For Discussing Data Inside the ADRF

Accessing and Using the Workspace

Accessing Your Data

Storing Analytic Results

Sharing Information within the ADRF

Exporting Results

SVN Documentation

Technical Support

References

Obtaining ADRF Access and Account Set Up

- [Requesting an Account](#)
- [Account Registration and Onboarding Tasks](#)
- [Obtaining ADRF Access](#)
- [More Information](#)

Requesting an Account

- [Agency-affiliated researcher.](#) If you are an agency-affiliated researcher, your agency will set up an ADRF account for you.
- [Individual part of a training program.](#) If you are part of a training program, Coleridge Initiative will create an account for you once you have been accepted into the program.

Account Registration and Onboarding Tasks

- You will receive an email invitation to activate your account. The email will come from <http://okta.com>, so please make sure that it doesn't get caught in your email spam filter. Follow the steps outlined in the email to set up your password and your multifactor authentication preferences.
- After activating your account, you will be logged in to the ADRF Applications page. Proceed to the Data Stewardship application by clicking on the icon.
- In the Data Stewardship application, you will notice a "Tasks" section with a number of items you will need to complete before you can gain access to the project space. Refer to the next section for details about the onboarding process.

Obtaining ADRF Access

- Agency-affiliated researcher. If you are an agency-affiliated researcher using an agency-sponsored account, you will be granted ADRF access once you complete your onboarding tasks and required data access agreements. If you are a self-paying agency-affiliated researcher, your ADRF access is conditional on receipt of payment. If your institution or Office of Sponsored Programs will be submitting payment on your behalf, please be aware of potential access delays. Whenever possible, the Coleridge Initiative advises paying with a personal credit card or institutional payment card and using the generated invoice to request reimbursement.
- Individual part of a training program. If you are part of a training program, you will be granted ADRF access once you complete your onboarding tasks and required data access agreements.

More Information

If you have any questions, please contact support@coleridgeinitiative.org.

Onboarding Modules and Security Training

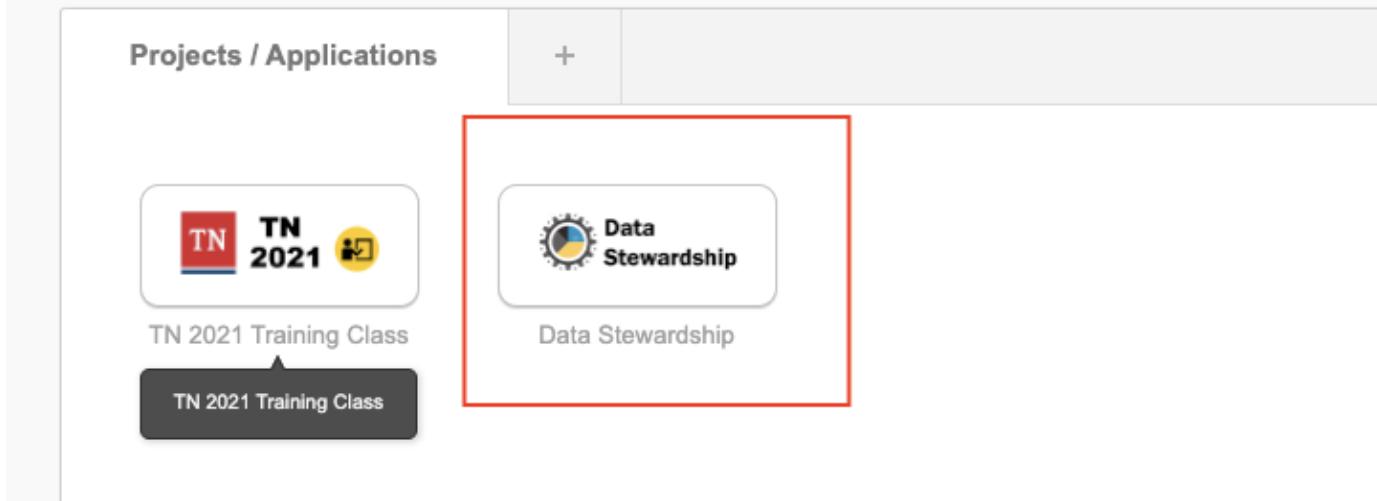
- [Data Stewardship App](#)
- [ADRF - Terms of Use](#)
- [Security Training Video](#)
- [Security Training Quiz](#)

Data Stewardship App

The Data Stewardship web-based application is positioned primarily as the management and monitoring console for project and data stewards. It provides detailed insight on project configurations, user activity, user onboarding status, and overall cost of a project on the ADRF. We focus on four primary pillars of information a Project/Data Steward most often focuses on:

- People – Who are the members of projects, how often do they use the ADRF, what exports have they requested and their status, estimated cost per person/project for current month and for the project since inception, and detailed usage metrics.
- Projects – Details of project start/end dates, abstract description, number of members onboarded and pending, and resources the project has access to (i.e. datasets, etc).
- Datasets – Description of the dataset, location on the ADRF (database or file system), size, name of the data steward(s), and the link to Enterprise Data Catalog (Informatica) describing the dataset and metadata.
- Agreements – What agreements are related to these projects, indication of each member's signing status, members pending signature, and term (dates) covered by the agreement(s).

As mentioned, the data stewardship application will track your ADRF usage. The app will also consolidate your ADRF Terms of Use, Security Training Quiz, and Security Training Video into one place. In order to complete ADRF onboarding, all three of the mentioned tasks are to be completed by the user (researcher). To access the Data Stewardship app, log in using your credentials at <https://adrft.okta.com> and click on the Data Stewardship icon. See picture below:



Once inside the Data Stewardship app, you have access to your personal workspace sessions statistics and the three tasks. See the example below:

Task	Assigned Date	Status
ADRFT - Terms of Use	June 8, 2021	Pending: 3
Security Training Quiz	June 17, 2021	Completed: 0
Security Training Video	June 17, 2021	Completed: 0

ADRFT - Terms of Use

The Terms of Use need to be completed before you are given access to the data and project space inside the ADRF. To complete ADRF Terms of Use, open the Data Stewardship app and click on ADRFT - Terms of Use. This will direct you to an DocuSign site to complete the signing of the agreement.

Security Training Video

The Security Training Video needs to be completed as well. To complete the training, open the Data Stewardship app and click on Security Training Video. This will direct you to the video; click Mark Complete when you have completed this training.

Security Training Quiz

The Security Training Quiz needs to be completed after the Security Training Video. To complete the training, open the Data Stewardship app and click on Security Training Quiz. This will direct you to the quiz, where you must answer five out of six questions correctly to pass.

Do's and Don'ts For Discussing Data Inside the ADRF

It is important to protect the confidential data that is inside the ADRF in communicating with your team-mates. The general rule is that you should never take any exact number out of the ADRF. This means you should never write down or share any number by text, screenshot, or share an image even with a team-mate. The rules have become more complicated now that everything is online, because even though your team-mates are “safe people”, and zoom conversations are password protected and encrypted, we’d rather err on the side of caution when sharing information over zoom.

This cheat sheet summarizes some of the rules that apply to discussing data **before it has been exported from the ADRF and passed the ADRF team’s disclosure review**. If you are unsure about a specific situation, please ask a Coleridge at support@coleridgeinitiative.org.

Exact Numbers

Do not describe a statistic in exact numbers. If you would like to communicate these values while not in person, you can have a private discussion via the projects drive inside the ADRF.

Example: If an average within a specific group was 5,000, you would need to convey this average on the projects drive.

Comparing Values

When comparing values, you are permitted to say that one value is more than, less than, or about the same as another. However, you cannot refer to the exact difference between the two numbers.

In practice, you can use pluses and minuses to convey differences between values for data that has not been exported from the ADRF.

Example: “The mean for Group A was roughly the same as the mean for Group B, but these values were both greater than that of Group C.”

Percentages/Proportions

Percentages and proportions also cannot be directly mentioned. Instead, you can refer to the percentage/proportion within 25%.

Example: If a proportion was 30%, you could say “The proportion is about 25%” or “The proportion is between 25% and 50.”

Accessing and Using the Workspace

- [Logging into and Logging out of the ADRF](#)
- [Virtual Desktop Environment](#)
 - What is a VDE?
 - Temporary Nature of the Environment
- [Modifying the Environment](#)
 - Establishing Personal Folders
 - The U: Drive and the P: Drive
 - Other Modifications
 - Windows Settings
- [Software in the ADRF](#)
 - JupyterLab
 - Notebooks
 - Accessing Stored Data from a Notebook
 - Python 3
 - R
 - Stata
 - DBeaver
 - Database Navigator
 - SQL Editor
 - Open a saved .sql File
 - LibreOffice
 - More...

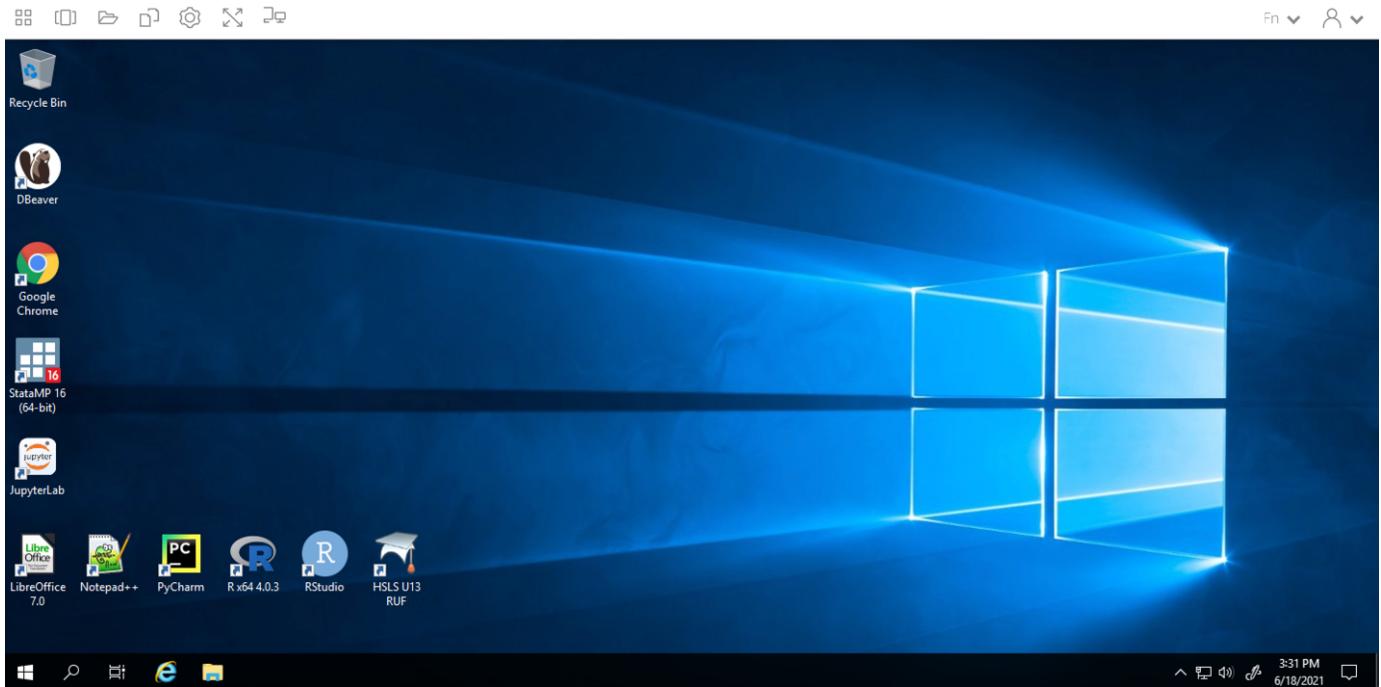
Logging into and Logging out of the ADRF

This video linked below runs through the necessary steps for logging into and logging out of the ADRF. If the video does not play, click [here](#).

Virtual Desktop Environment

What is a VDE?

Purpose, Contents, Capabilities
A virtual desktop environment (VDE) allows you to interact with a remote system as if it were your own personal computer. The majority of your standard desktop functions are available, but the programs, data, and permissions are all controlled by the remote administrator (Coleridge Initiative). Thus, you will be working in a familiar environment while accessing protected data, programs, and systems that would otherwise be difficult to distribute. The ADRF uses a standard Windows environment (Windows Server) and provides a variety of software packages to conduct your analysis. For more on Windows capabilities, see the section on Windows Settings.



Temporary Nature of the Environment

While the environment is similar to that on your home computer (for Windows users), there are a handful of key differences. The first is that the environment is temporary in nature. This means that if you are not using it for a prolonged period of time (default is four hours but can vary by project), running programs will stop running and the information stored in temporary locations will be deleted. You will receive an on-screen message before any sessions are terminated. For more on safe, non-temporary storage locations in the ADRF, see the section on Storing Analytic Results.

Given the temporary nature of the ADRF, it is crucial to make sure that your work is saved—and saved in an appropriate location. Once this is complete and you are finished working, make sure that you log out of the ADRF instead of closing the window. To do this, click the rightmost icon on the top taskbar to open up the dropdown menu and select End Session. You will be prompted to double-check that your work is saved prior to ending your session and confirm that you want to end your session.

Modifying the Environment

Establishing Personal Folders

Establishing your own personal folders is one of the simplest, yet most important, steps to take when setting up your environment. As we note in the section on Storing Analytic Results, the two possible places to store your analytic results or files are in either the U: drive or the P: drive.

An important folder to set up is your personal folder in the U: drive. Set this up by creating a folder with your Firstname.Lastname.

The U: Drive and the P: Drive

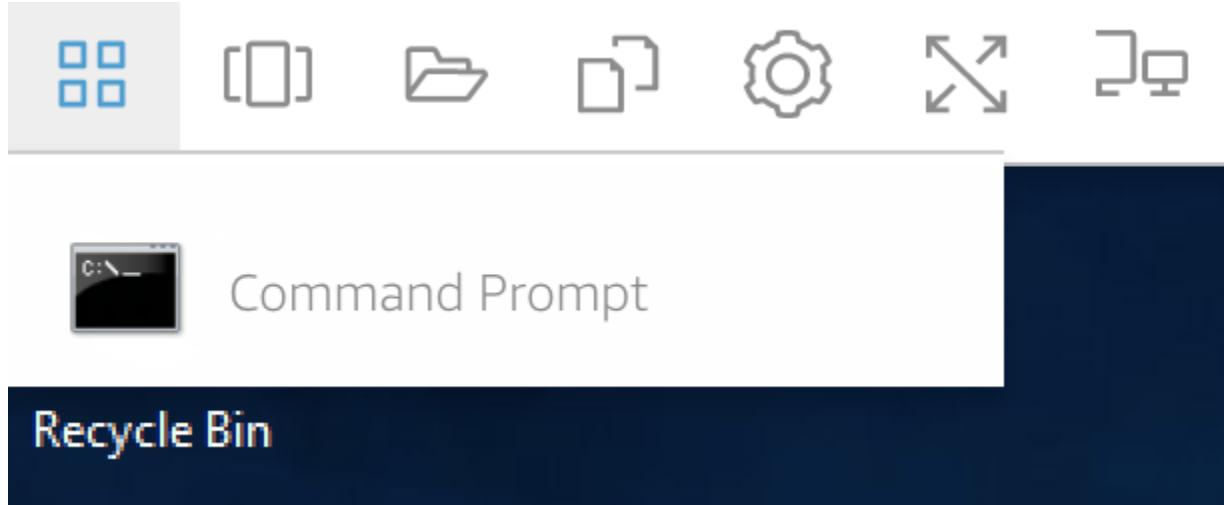
The U: drive is your user drive; it's where you will store any files you are working on. Only the user will have access to the U: drive. For example, if user A wants to share information with user B who is on the same project, user A will need to save files to a P: drive folder and not folders in their U: drive since user B will not be able to access user A's U: drive.

The P: drive is the project drive, which will be used to house project-specific folders. Thus, you and other collaborators on the same project will be able to save files to project drive folders.

Both the U: drive and P: drive have defined resource limitations of 150GB. When the workspace exceeds these limits, users will not be able to create new files or save data. The ADRF will not alert users when they approach 150GB used. Users can check their current usage by right clicking on the user folder and clicking on properties.

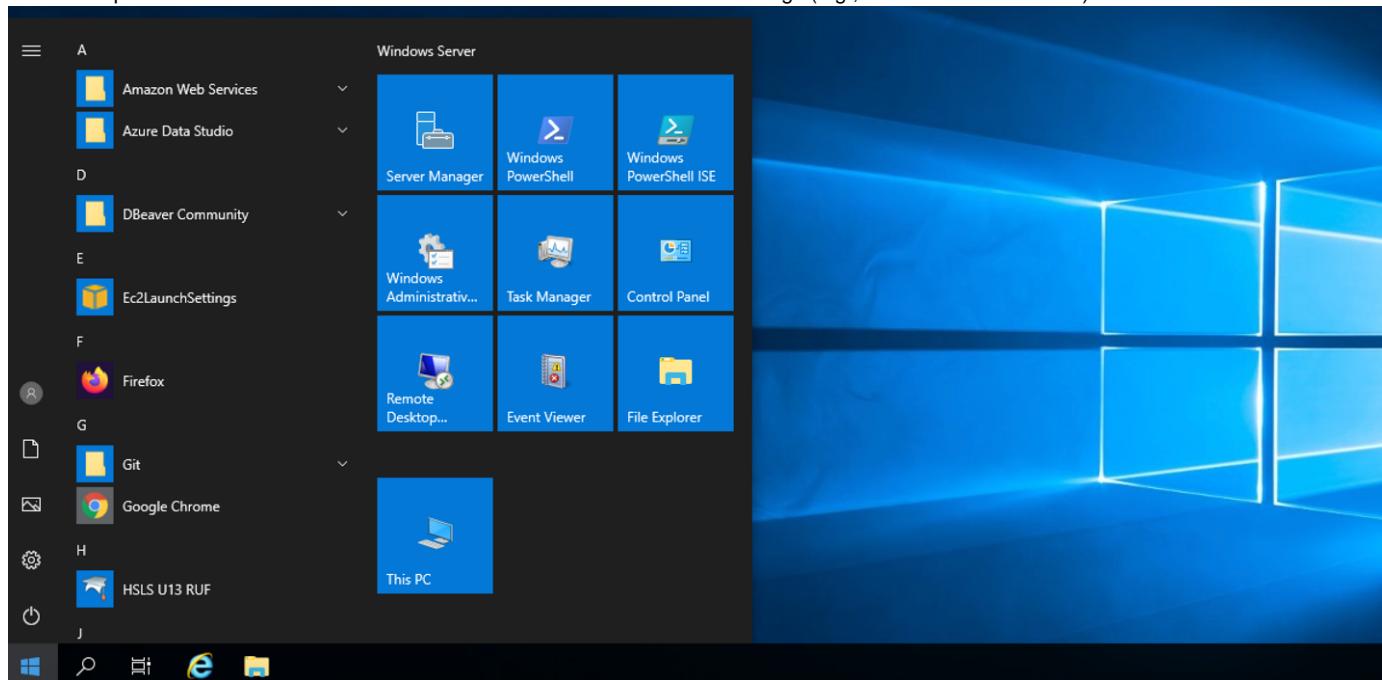
Other Modifications

The top taskbar contains shortcuts to the command prompt, multiple desktop windows, a temporary folder, settings, full-screen view, and toggling multiple monitors.



Windows Settings

Your desktop will look familiar if you are a Windows user. You will have icons for quick access to programs or browsers on your desktop. The windows icon on the bottom left side of the screen will open up a menu of programs, folders, and other tools, much as you would see on your own desktop. You will have access to PowerShell and several customization settings (e.g., remove bottom taskbar).

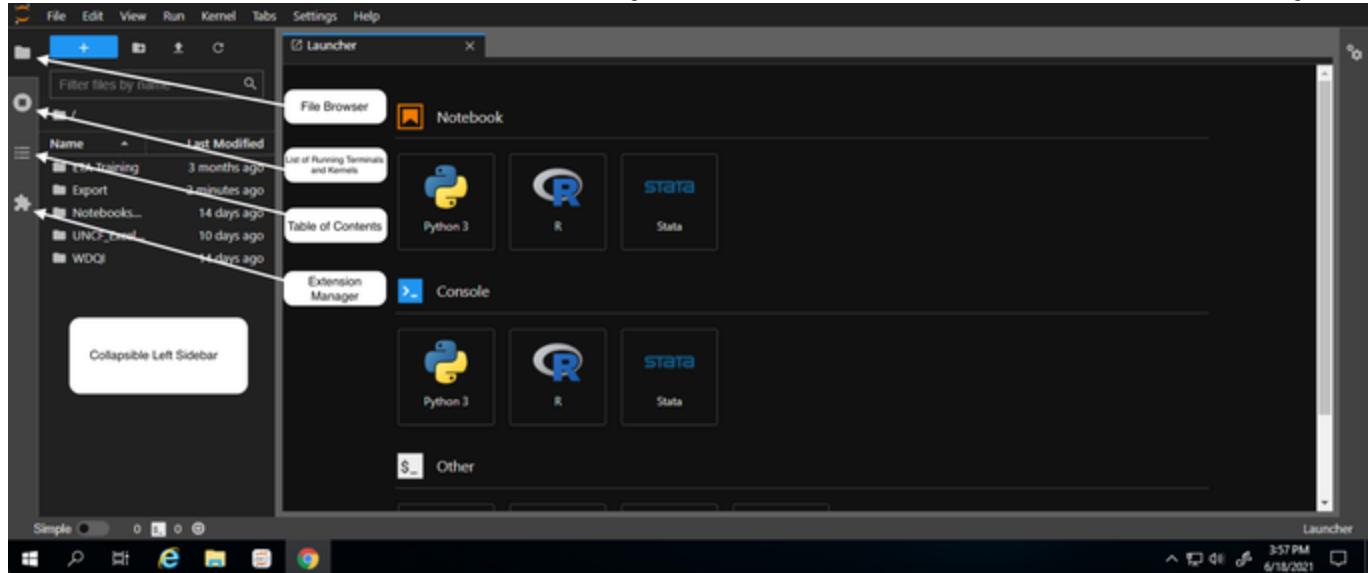


Software in the ADRF

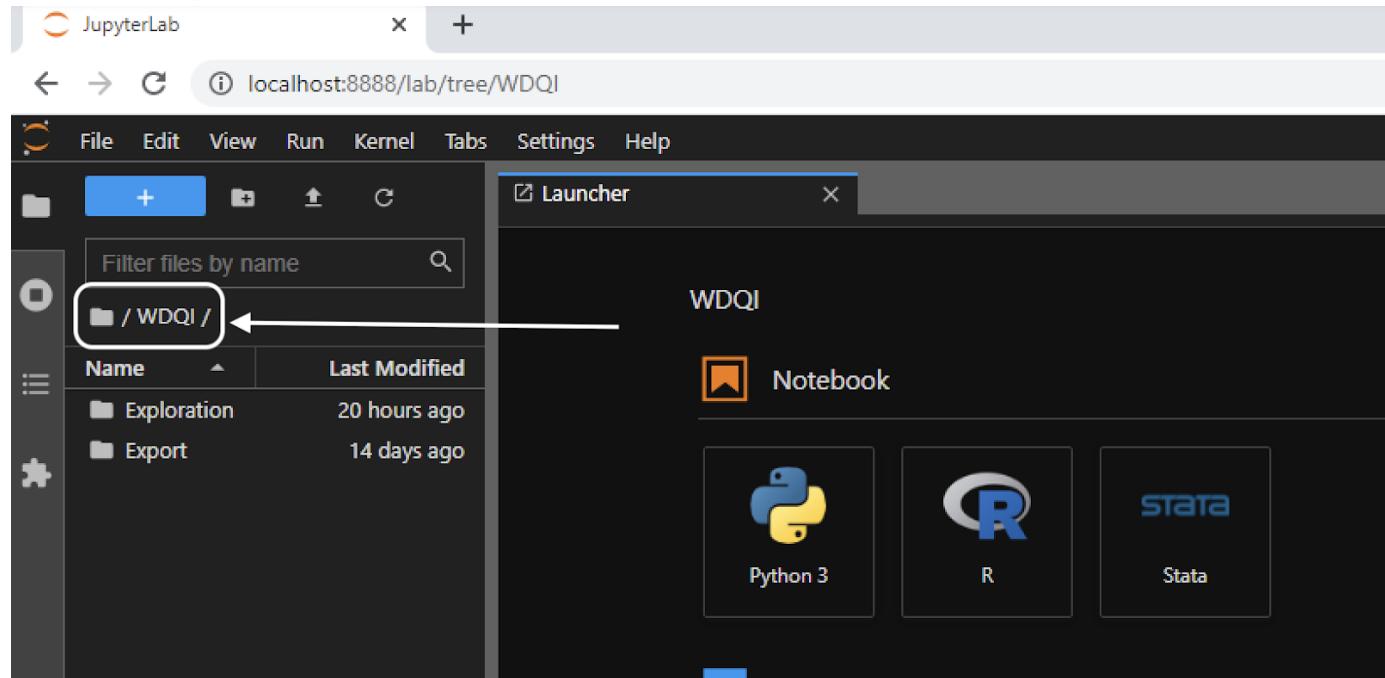
JupyterLab

JupyterLab provides flexible building blocks for interactive, exploratory computing. While JupyterLab has many features found in traditional integrated development environments (IDEs), it remains focused on interactive, exploratory computing. For more on JupyterLab, see the interface documentation.

The JupyterLab interface on the ADRF consists of a main work area containing tabs of documents and activities, a collapsible left sidebar, and a menu bar. The left sidebar contains a file browser, the list of running terminals and kernels, the table of contents, and the extension manager.



When using Jupyter Notebooks, make sure that all your work is saved to your U: drive and the correct directory within the U: drive. You can "find the active directory by reading the path displayed in the file browser. By default, JupyterLab opens with your U: drive as the base directory. Below, the folder icon in the white box is my user folder (not displayed, but titled Firstname.Lastname; you will have already set up your folder) and subfolder WDQI.



Notebooks

Jupyter Notebooks are documents that combine live runnable code with narrative text (Markdown), equations (LaTeX), images, interactive visualizations, and other rich output. You can create a notebook by clicking the blue + button in the file browser and then selecting a kernel (R, Python3, Stata) in the Launcher tab. For more information on getting started with Jupyter Notebooks, see JupyterLab Notebook documentation.

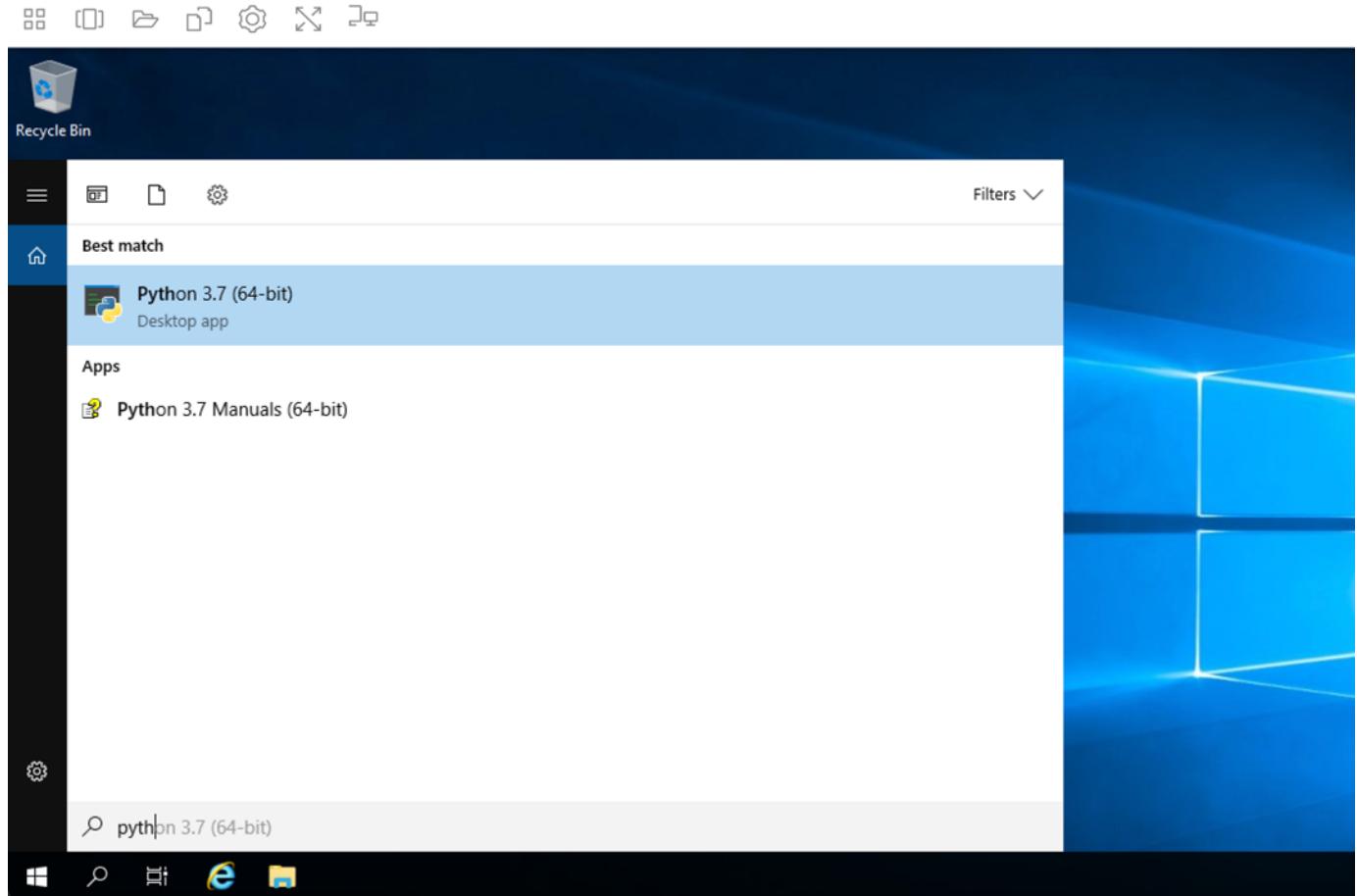
Accessing Stored Data from a Notebook

A common question is how to access stored data while writing to and using a Jupyter Notebook. Data in the ADRF are stored in a database using Microsoft SQL Server. For more information on how to access stored data in the ADRF based on choice of program (Python, R, Stata), see the section on Accessing Your Data.

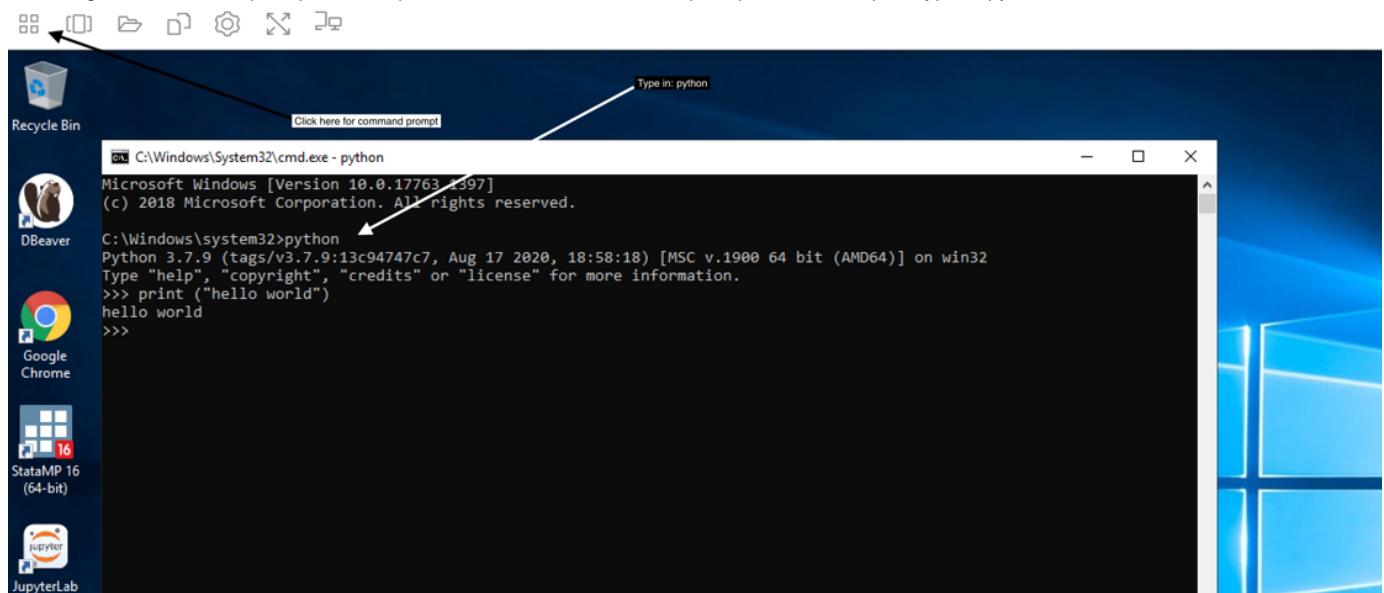
Python 3

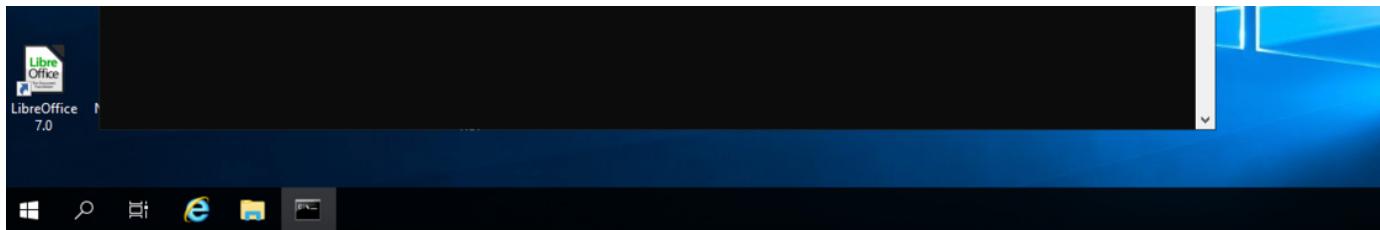
Python is a general-purpose programming language. You can access Python in a multitude of ways:

1. Through the start menu (windows icon). Type in Python. A desktop app called Python 3.7 (64-bit) will populate a window where you can begin programming.

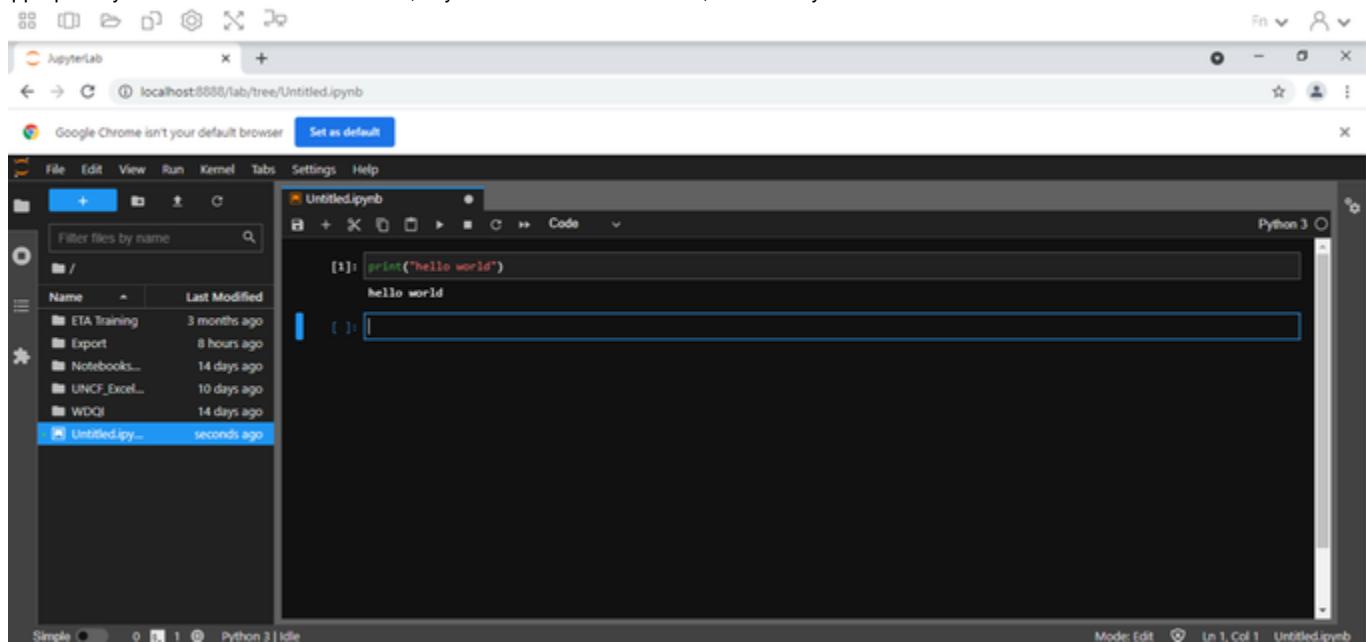


2. Through the command prompt in the top taskbar. Once the command prompt window is open, type in python.





3. Through JupyterLab. This is the recommended way to access Python since it has packages installed and available, and an execution environment for testing and running code (as well as a place to write and save code). Open JupyterLab and make sure your directory is set appropriately in the file browser. Once there, in your new Launcher window, click the Python 3 icon.



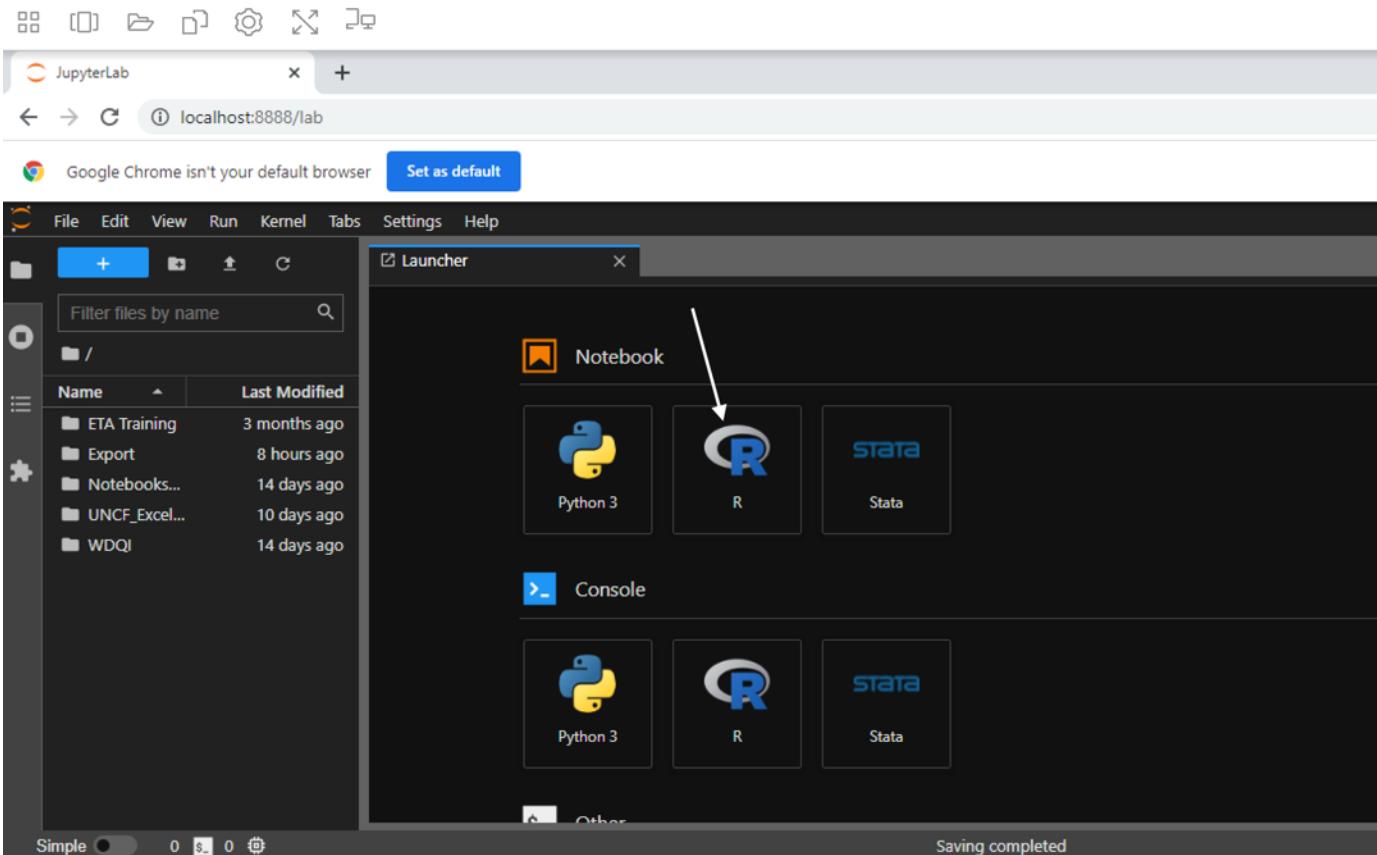
4. Through Pycharm.



R

R is a general-purpose programming language. You may access R in one of three ways:

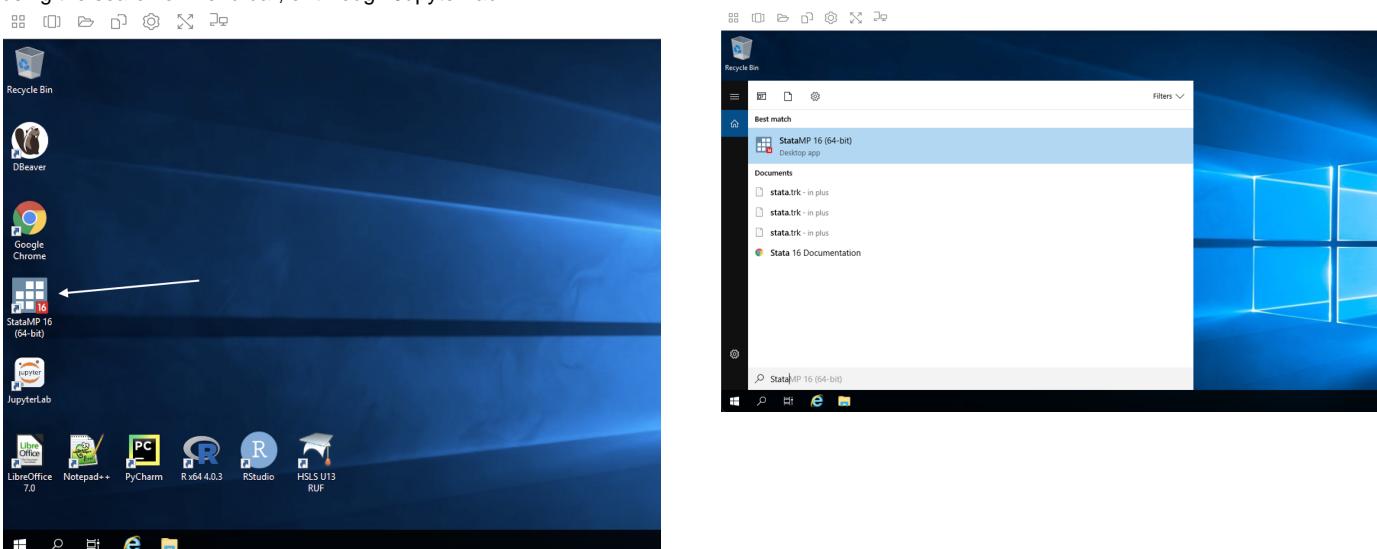
1. Through RStudio. This is an integrated development environment (IDE) for R. You can run R code, display variables, debug R code, do inline visualizations, and more. Open RStudio through the desktop shortcut, or type RStudio in the start menu.
2. Through JupyterLab. Open JupyterLab and make sure your directory is set appropriately in the file browser. Once there, in your new Launcher window, click the R icon.



3. Through the R GUI (graphical user interface). Type R in the search bar and click to open the RGui.

Stata

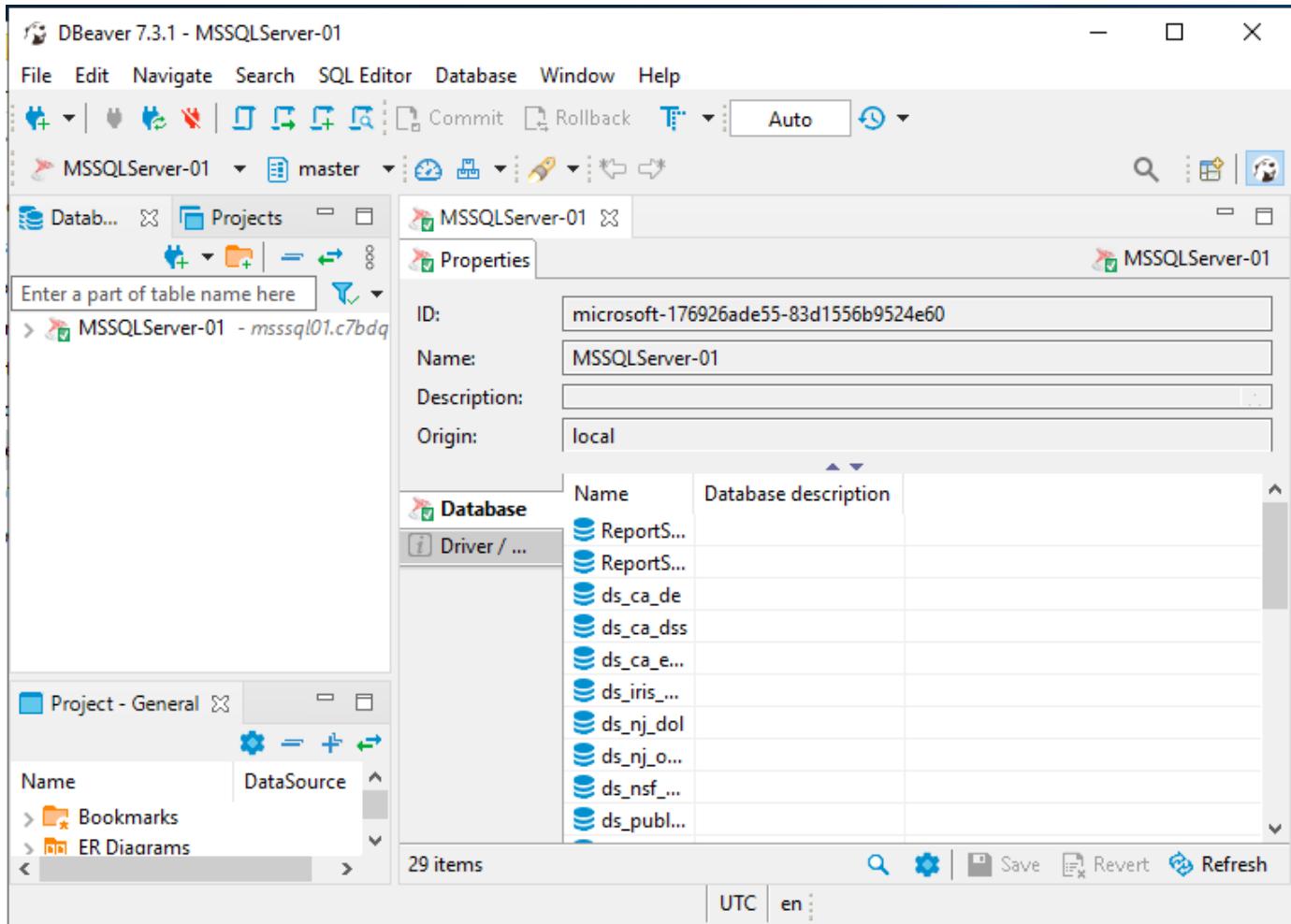
Stata is a general-purpose statistical software package. Stata can be accessed through the desktop shortcut StataMP 16 or by searching for it using the search or menu bar, or through JupyterLab.



DBeaver

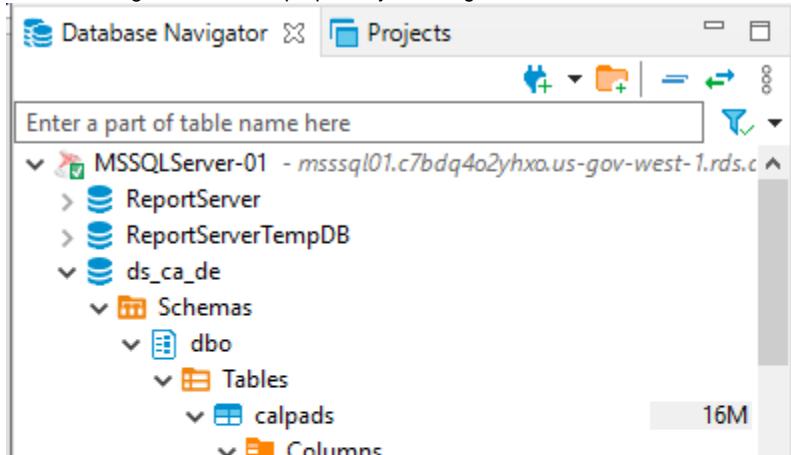
DBeaver is a universal tool for querying, editing, and managing data stored in SQL databases. The ADRF stores data using Microsoft SQL Server. DBeaver can be accessed through the desktop shortcut DBeaver or by looking it up using the search bar.

Once open, you will notice that the current settings of DBeaver automatically open your connection to MSSQLServer-01, as shown in the screenshot below.



Database Navigator

On the left side of DBeaver, a pane labeled Database Navigator allows you to easily explore what is in the server to which you are connected—on the ADRF it is MSSQLServer-01. By clicking the arrow, all the items within each server, Database, Schema, etc., are shown. When exploring these data and writing SQL queries, it is frequently useful to have the navigator expanded to see more easily what columns are in each table and their datatype; the datatype can be seen in the screenshot to the right in parentheses next to each column name (e.g., clientId(char64) is a text column of length 64—for our purposes you can ignore the char... and varchar... and simply treat it as text).



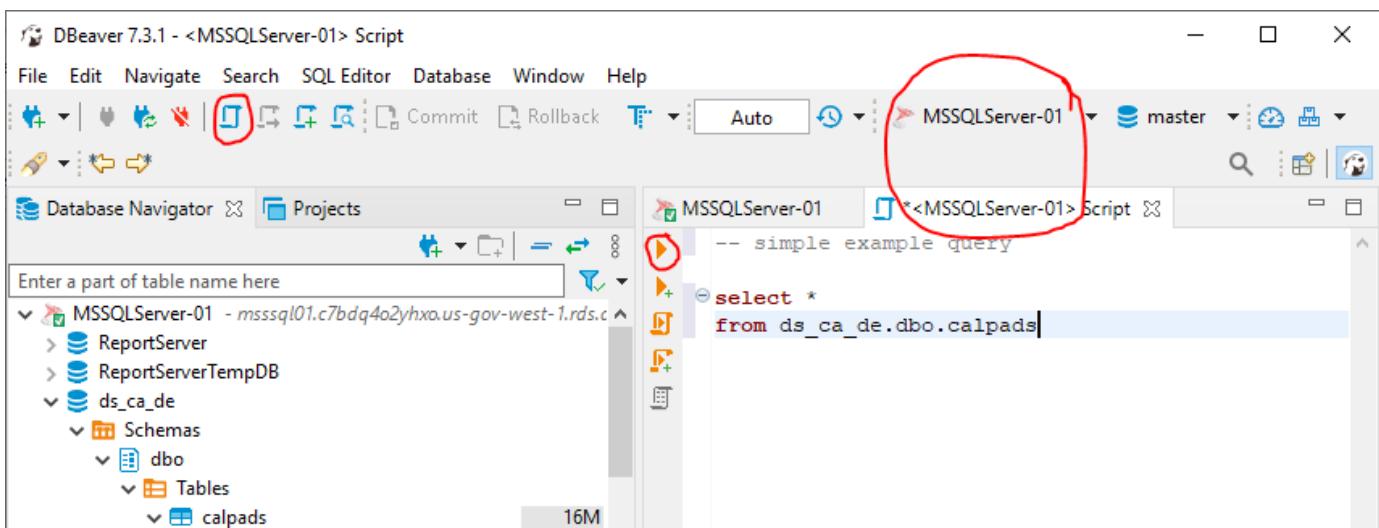
```
ABC clientid (char(64))
ABC race_ethn (varchar(19))
ABC schl_type (varchar(41))
⌚ sch_start_dt (date)
⌚ sch_exit_dt (date)
ABC completion (varchar(30))
ABC uccsu_req (char(1))
```

SQL Editor

The SQL Editor is where you can write your own queries to analyze the data. A new script can be opened by clicking on the blue almost-square (looks a bit like an unrolled scroll) on DBeaver's tool bar:

The location of this script button is circled in the red in the upper left of the screenshot below.

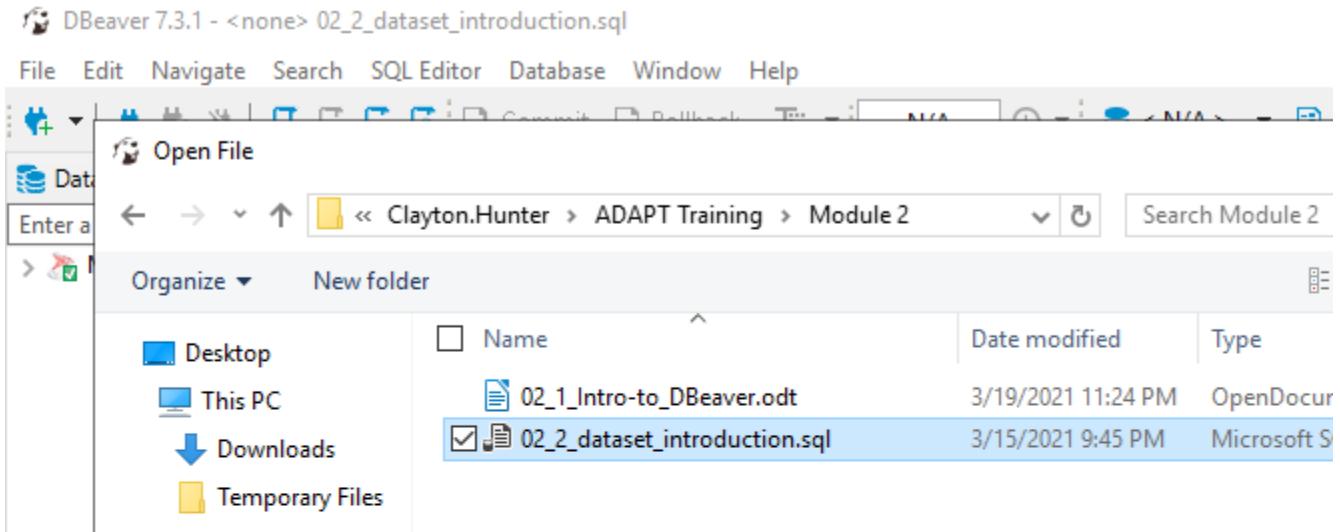
Note: If you use the SQL Editor button to open a new window, it will default to being connected to the server; however, if you open a saved "le, you will need to select the Server to tell it to connect—that is shown in section 3, and the connection is circled in the screenshot below.

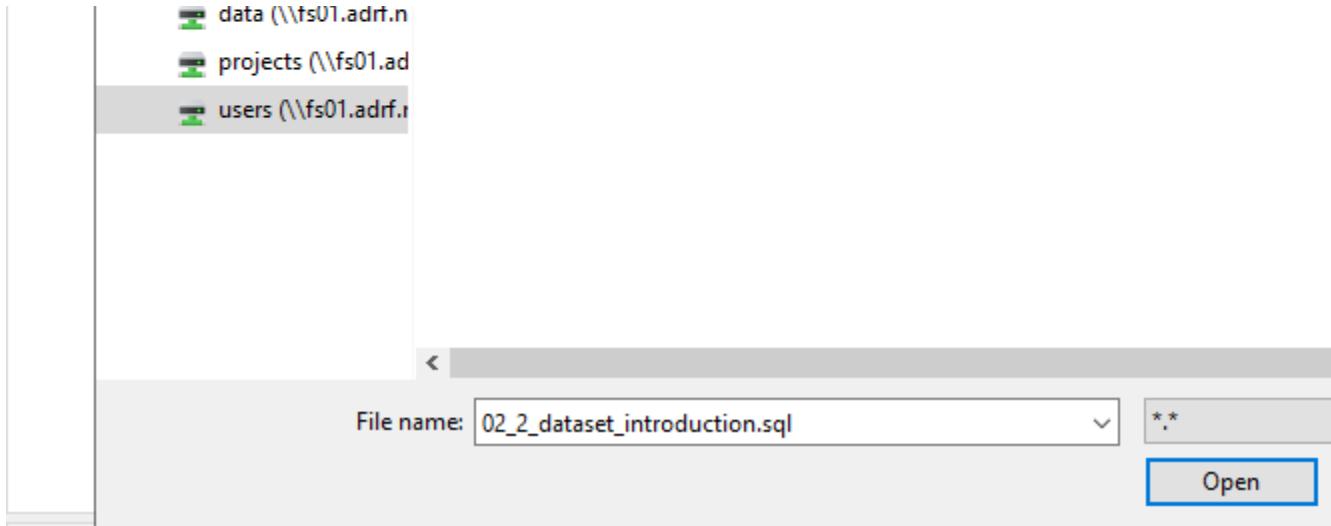


Once you have a SQL Editor window open, you can write a query and run it. One option to run a query is to use the keyboard to hit **ctrl+enter**, and another option is to use the orange triangle circled in red in the middle of the screenshot above.

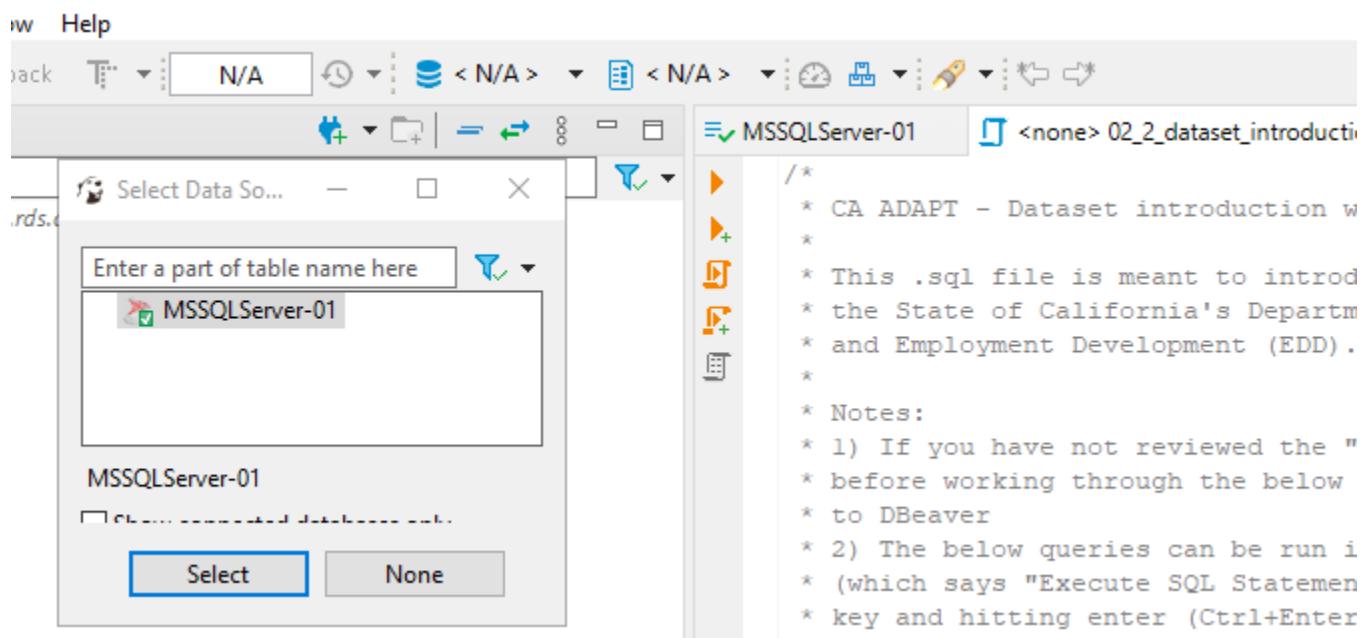
Open a saved .sql File

You do not have to create a new script every time! You can open a .sql file either by simply dragging and dropping it from the file explorer, or by going to **File Open File** and navigating to a .sql file, as shown in the screenshot below:

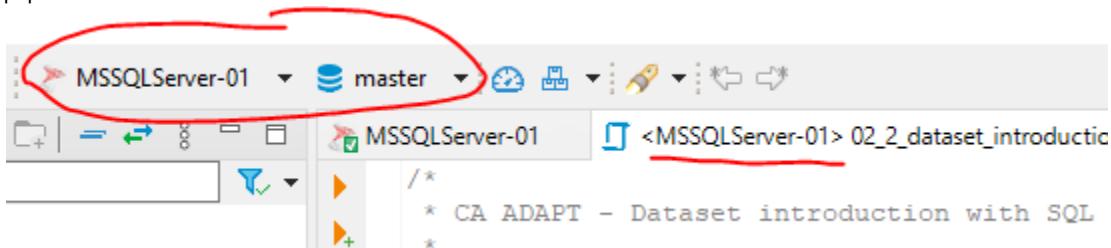




When you open a .sql file, it will not automatically connect to the server. You can set the connection by clicking the activate datasource button, selecting MSSQLServer-01, and clicking select.



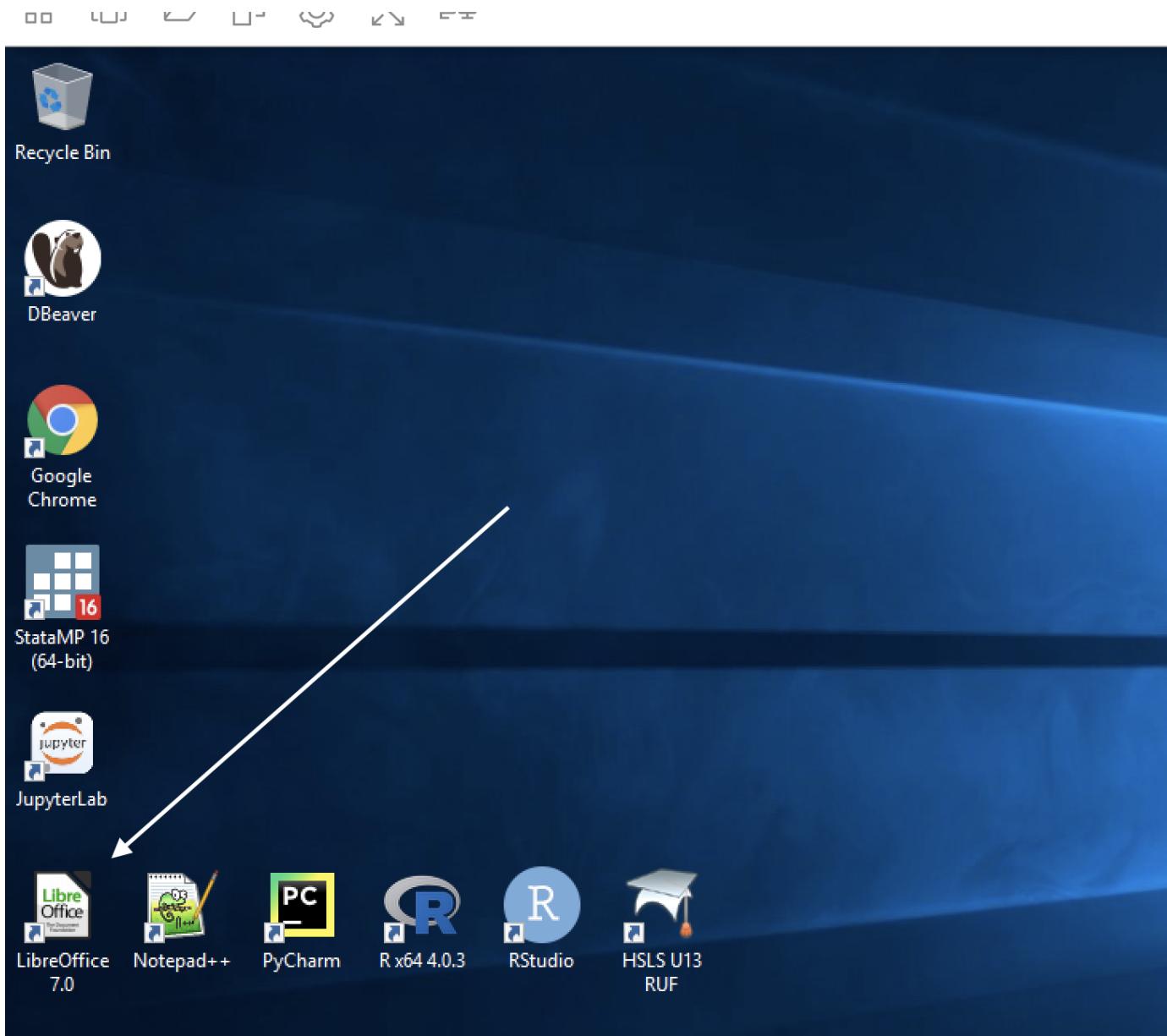
Once you have done so, the top of your SQL Editor window should name the server connection inside the angle brackets to the left of the filename (<MSSQLServer-01>, underlined in red below), and the Active datasource and active catalog/schema (circled in red below) will be populated:



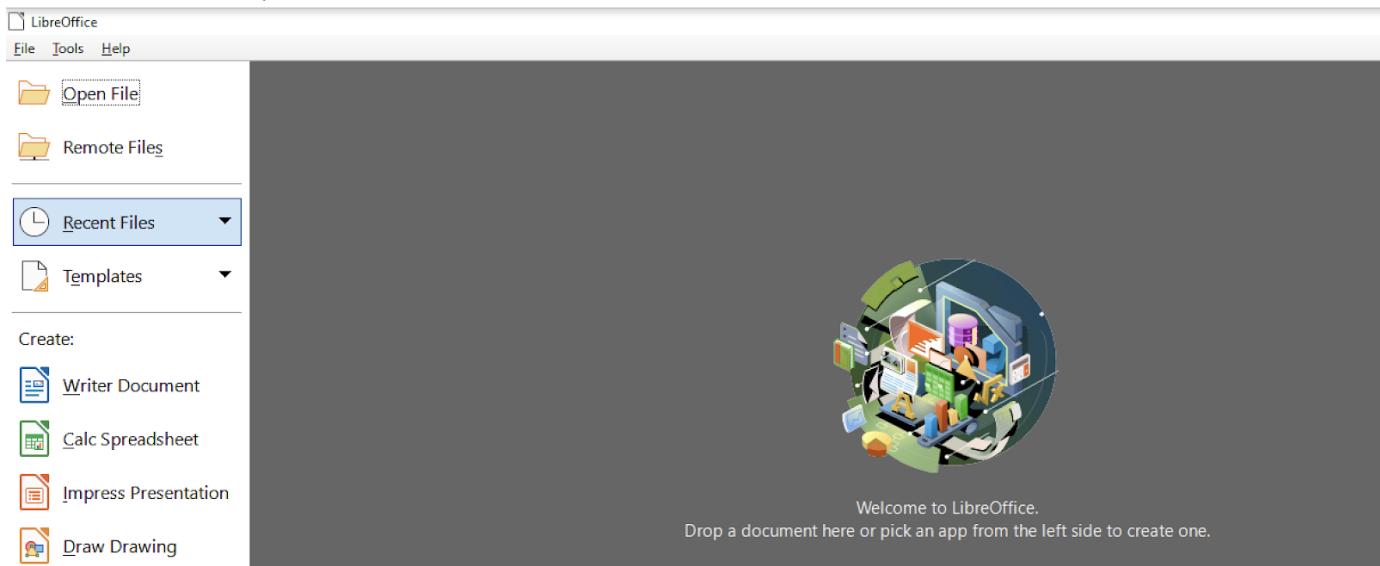
LibreOffice

LibreOffice is an office productivity suite. LibreOffice comes equipped with six different programs: a word processor program (Writer), a spreadsheet program (Calc), a presentation program (Impress), a graphics editor program (Draw), a math equation program (Math), and a database management program akin to Microsoft Access (Base). LibreOffice may be accessed through the desktop shortcut DBeaver or by looking it up using the search bar. Once you've opened up LibreOffice, you can open any of those six programs, using the left sidebar. For more information on LibreOffice, visit the [LibreOffice website](#).





Once you click on the icon, you'll see a page with a left sidebar that has a variety of document types under Create. Select the one suited to your needs and double click to open it.





More...

The ADRF provides a number of additional programs such as a simple text editor (Notepad++), PyCharm (an IDE for Python users), and several web browsers. Please note that web browsers are limited only to approved websites.

Accessing Your Data

- [Locate your Data in a Database](#)
 - [Python3](#)
 - [R](#)
 - [Stata](#)
 - [External Data and Code](#)

Locate your Data in a Database

The ADRF stores data using the Microsoft SQL Server. The simplest way to locate and get a quick overview of your data in a database is to use DBeaver. Once DBeaver is open, click MSSQLServer-01 (dropdown). The icons displayed all represent a database hosted in the ADRF. Choose the database corresponding to your project (contingent on your data agreements). If all proper permissions and data agreements are in place, clicking a database will open a menu containing Schemas, Database triggers, and Administration. Click on Schemas to access a list of schemas. Click on dbo, which is a default schema name. Once you've clicked on dbo, open Tables to display a list of data tables, or Views to display a list of views. Double clicking any table name will open the table or view properties. Use the tabs to explore data properties and the data contained in the table itself. This is a great way to ensure that you are using the correct table and that the data you are using are appropriate for your particular analysis. For more on using DBeaver to locate your data, see the section on [Software in the ADRF: DBeaver](#).

If you are uncertain about what database to use or project you are working on, contact your Data Steward.

Python3

In Python, you can establish a database connection and execute queries in a straightforward manner. If you are familiar with pandas, the output can be parsed into a pandas dataframe; otherwise, it will come as a list of results ([pandas](#) is highly recommended).

The required code will be the same regardless of how you access Python (Jupyter Notebook or the command prompt).

Begin by importing the requisite libraries. You will need to import pyodbc and pandas.

```
Untitled.ipynb
```

```
Begin by importing pyodbc, pandas
```

```
[1]: import pyodbc
      import pandas as pd
```

Now establish a connection to the SQL server database. Do this by writing a connection string and using `pyodbc.connect()`; assign this to connection.

Next, using pandas, we can create a dataframe using query results. Define a dataframe: `df = pandas.read_sql("query", connection)`.

The screenshot shows a Jupyter Notebook interface with a single code cell containing Python code. The code imports pyodbc and pandas, establishes a connection to an SQL Server database using a connection string, and executes a query to select distinct claim_types from a table. The notebook is titled 'connection_example.ipynb'.

```
Begin by importing pyodbc, pandas

import pyodbc
import pandas as pd

Now establish a connection to the SQL server database. Do this by writing a connection string and using pyodbc.connect(); assign this to connection.

connection_string = "Driver={SQL Server}; Server=mssql01.c7bdq4o2yhxo.us-gov-west-1.rds.amazonaws.com; Database=tr_dol_eta; Trusted_Connection=True"

connection=pyodbc.connect(connection_string)

query="SELECT DISTINCT(claim_type) FROM dbo.il_des_promis_1pct"

Next, if we are using pandas, we can create a dataframe using query results. Define a dataframe df=pandas.read_sql("query", connection).

df=pd.read_sql(query, connection)

df
```

Note here that your query will include the location of the server as well as the table you'd like to select. In this example, I am selecting distinct claim_types from the table il_des_promis_1pct in schema dbo and database tr_dol_eta.

Without pandas, you will still need to import pyodbc, write a connection string, and establish a connection. To get the data, you will need to create a cursor object that is then able to execute your query. `Cursor.execute("query")` will execute your query and `cursor.fetchall()` will return all records as a list. For more on cursor, see the documentation.

Note: As is always the case with programming languages, there are multiple methods for accessing data. This list should not be considered exhaustive, but rather instructive for researchers who are new to the ADRF or SQL server and, in particular, are looking for the location of the server.

R

In R, you can establish a database connection and execute queries much like in Python.

Begin by importing the requisite library ([ODBC](#)).

Next, establish a connection to the SQL server database using the `dbConnect` method from DBI (Database Interface) and assign the connection to a variable (e.g., `con`), much as in Python.

After that, establish your SQL query in string format and assign it to a variable (e.g., `query`).

Finally, access the data using `dbGetQuery(con, query)` and store the output as a dataframe with a name of your choosing. You may now reference this dataframe for your data analysis.

The screenshot shows an RStudio interface with a code editor containing R code. The code imports the odbc library, connects to the database using dbConnect, and stores the output as a connection object. It then defines a SQL query and stores it in a variable named `query`.

```
# Database interaction import

[1]: library(odbc)

# Connect to the database

[2]: con <- DBI::dbConnect(odbc::odbc(),
                           Driver = "SQL Server",
                           Server = "mssql01.c7bdq4o2yhxo.us-gov-west-1.rds.amazonaws.com",
                           Database = "tr_dol_eta",
                           Trusted_Connection = "True")

# Store SQL query to a variable and execute query

[3]: query <- "
```

```

SELECT DISTINCT(claim_type)
FROM tr_dol_eta.dbo.il_des_promis_1pct;
"

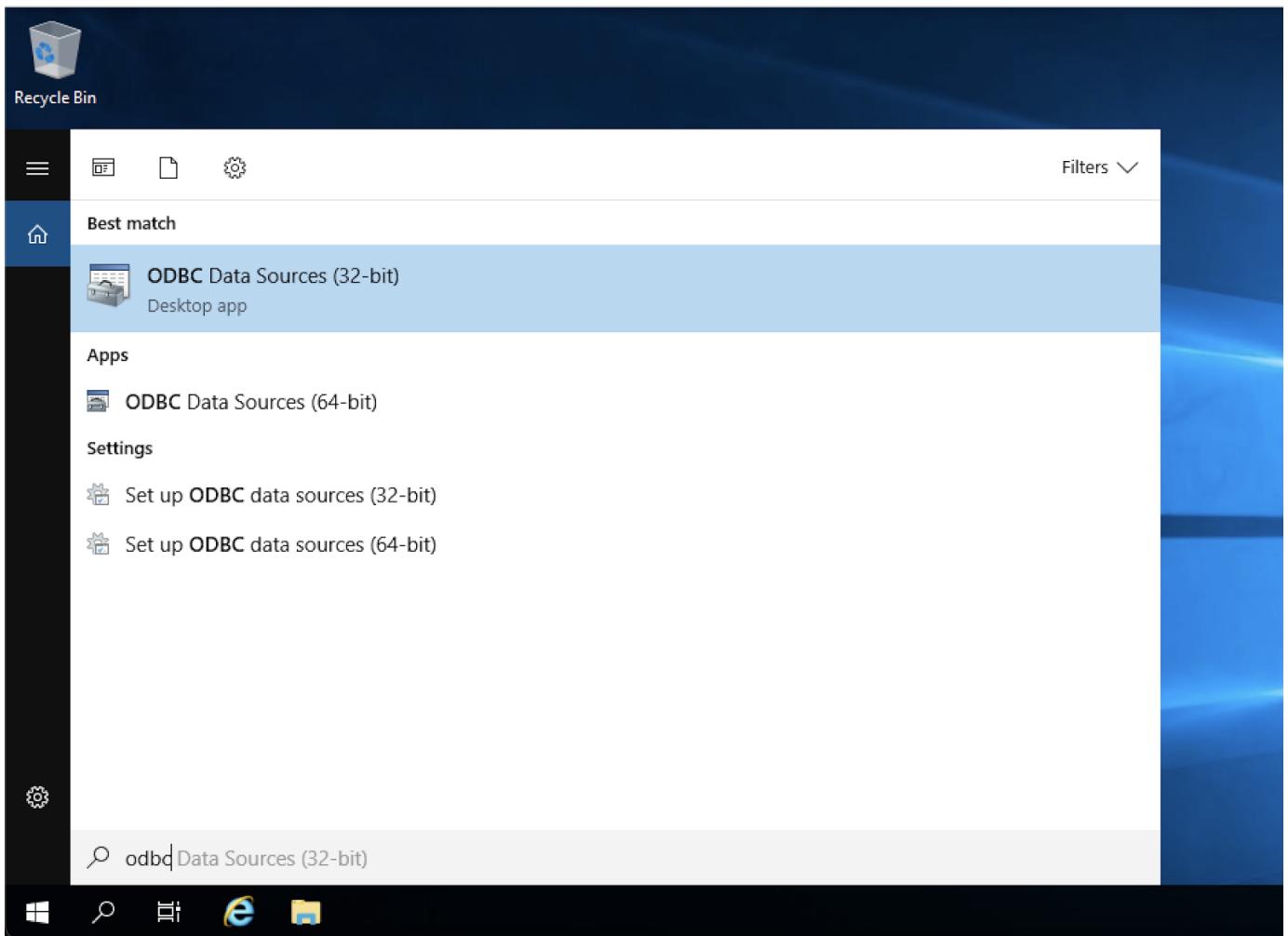
# Execute query
df_claim_types <- dbGetQuery(con,query)

```

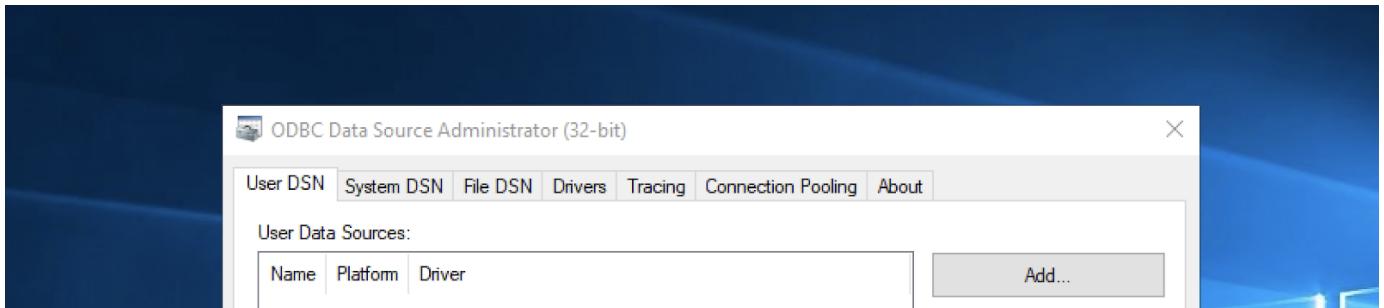
Stata

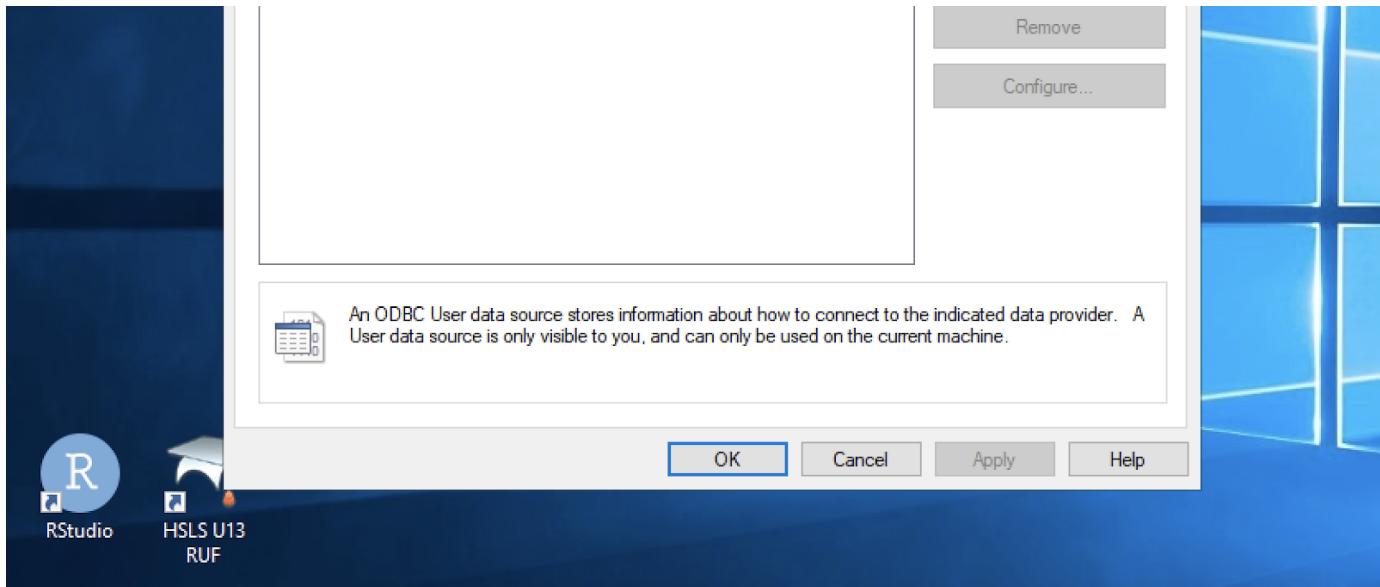
Establishing a connection to a database using Stata is also done through ODBC, but unlike in R and Python, you will need to create the connection itself through the ODBC Data Sources desktop app. Open up the app, hit Add, select SQL Server as the driver, add a name, add a description, and add the server location (which is the same as in the above two examples). Note that this will need to be done every time you log in to the ADRF. For more information on setting up an ODBC data source name in Windows, see [here](#).

1. Open up ODBC Data Sources

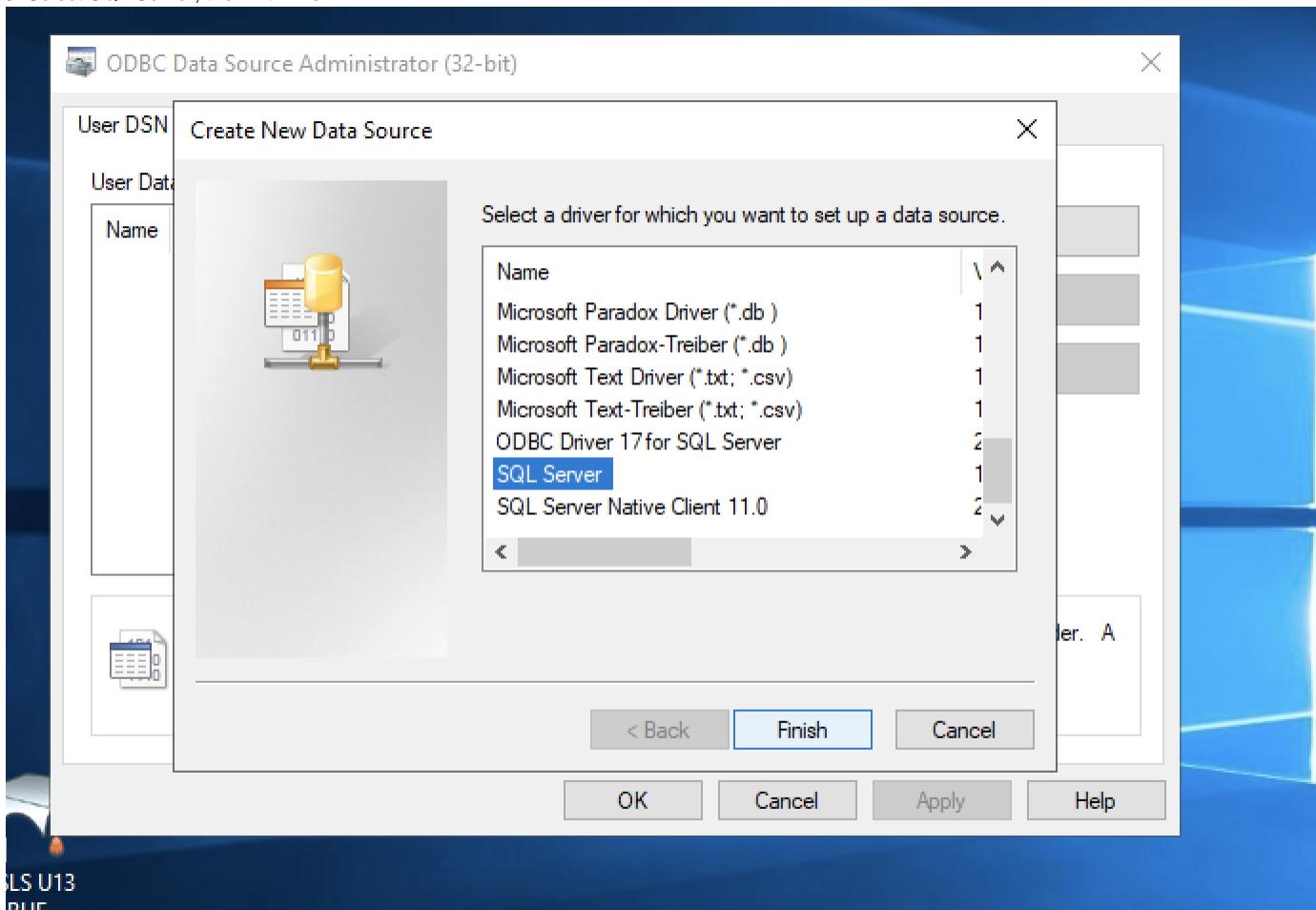


2. Click Add.

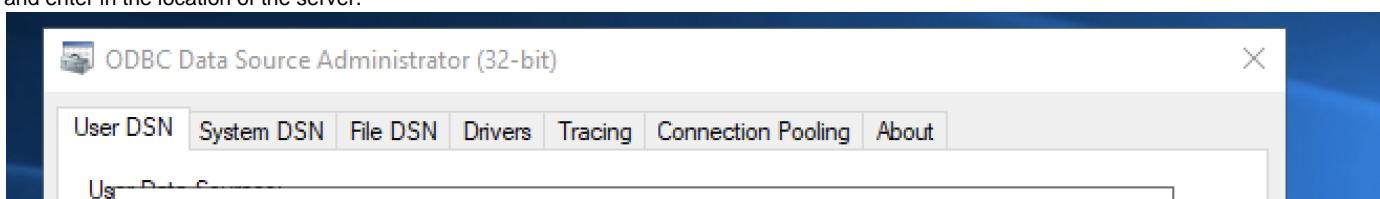


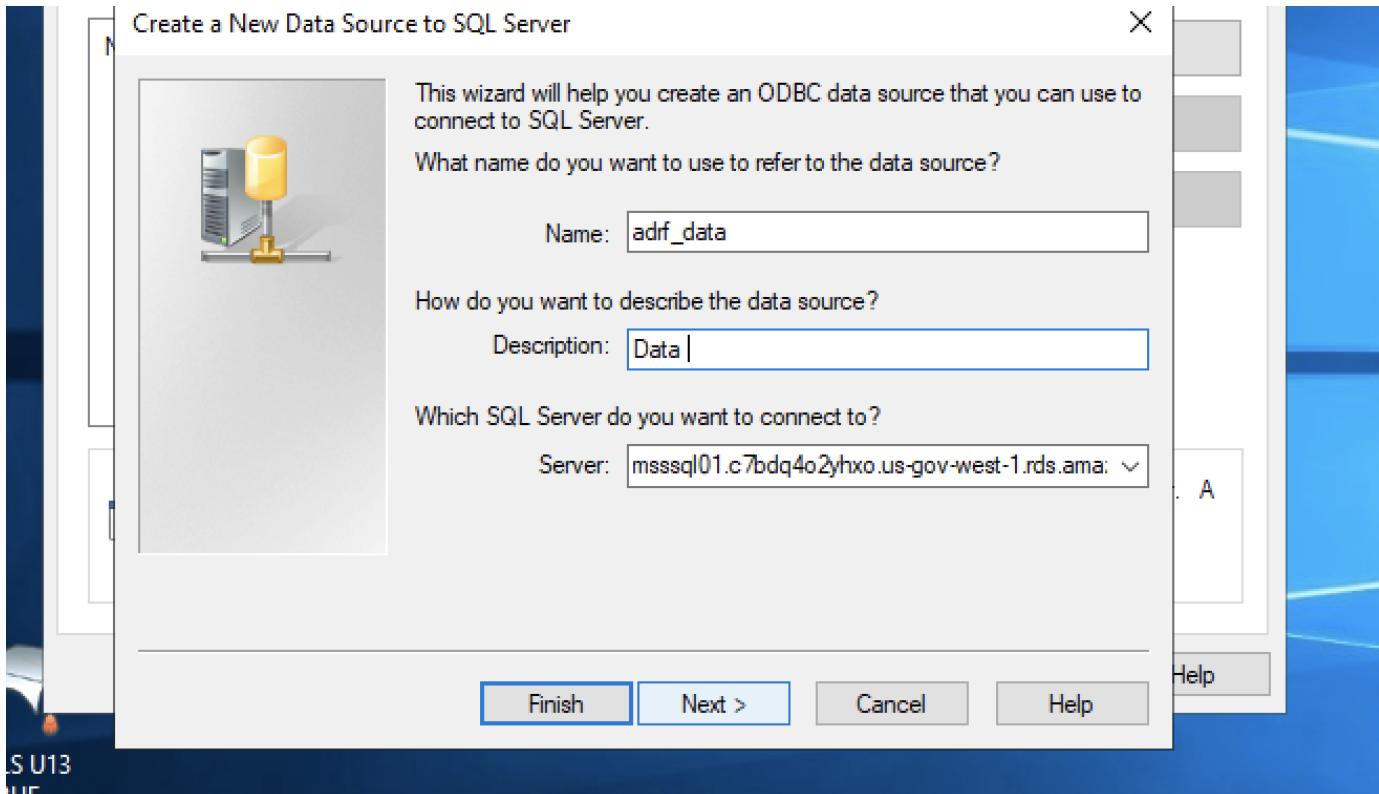


3. Select SQL Server, then hit Finish.

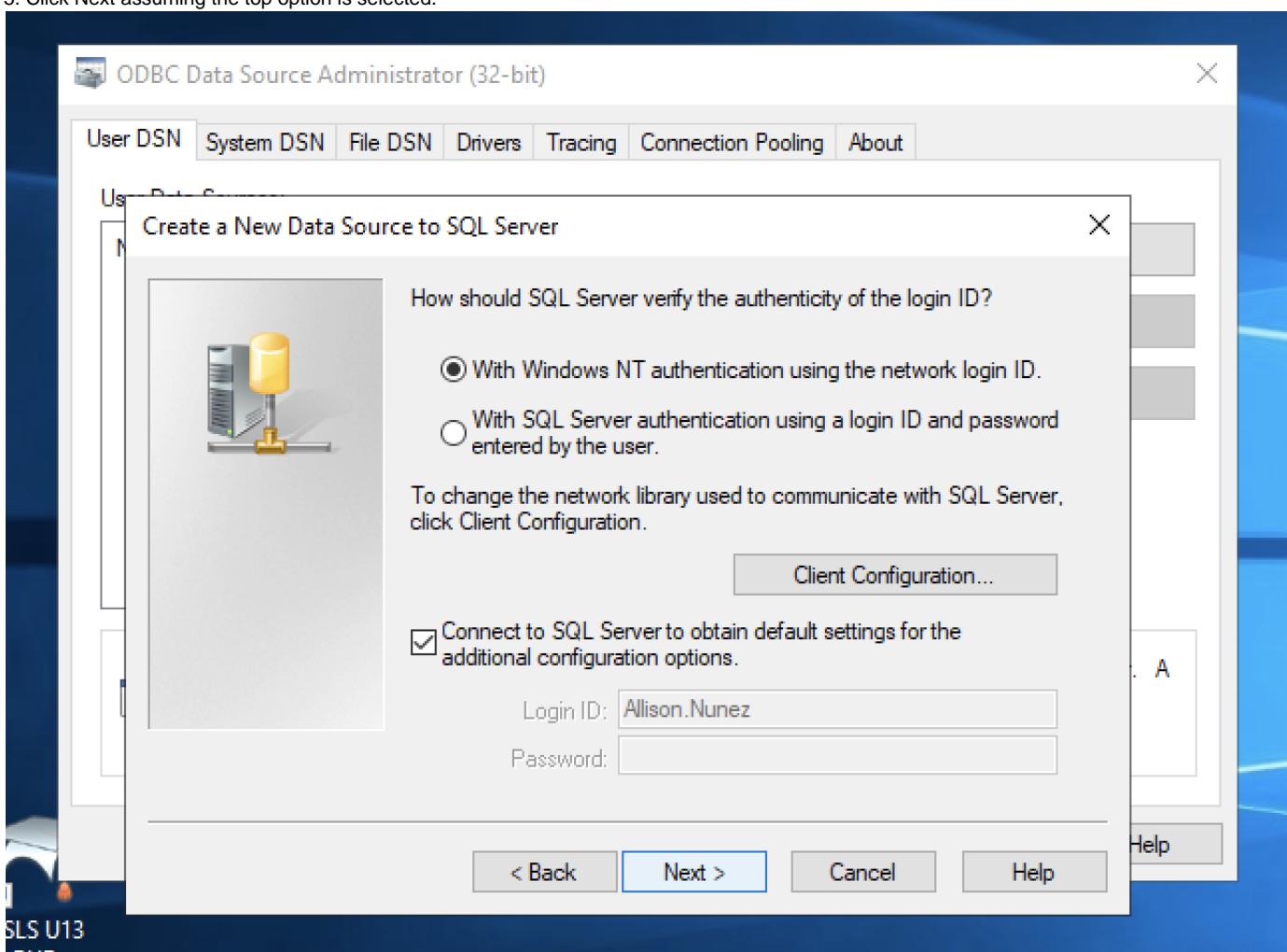


4. Enter the name you want to use to refer to the data source in Stata. In this example, I called the data source `adrf_data`. I describe it as Data and enter in the location of the server.

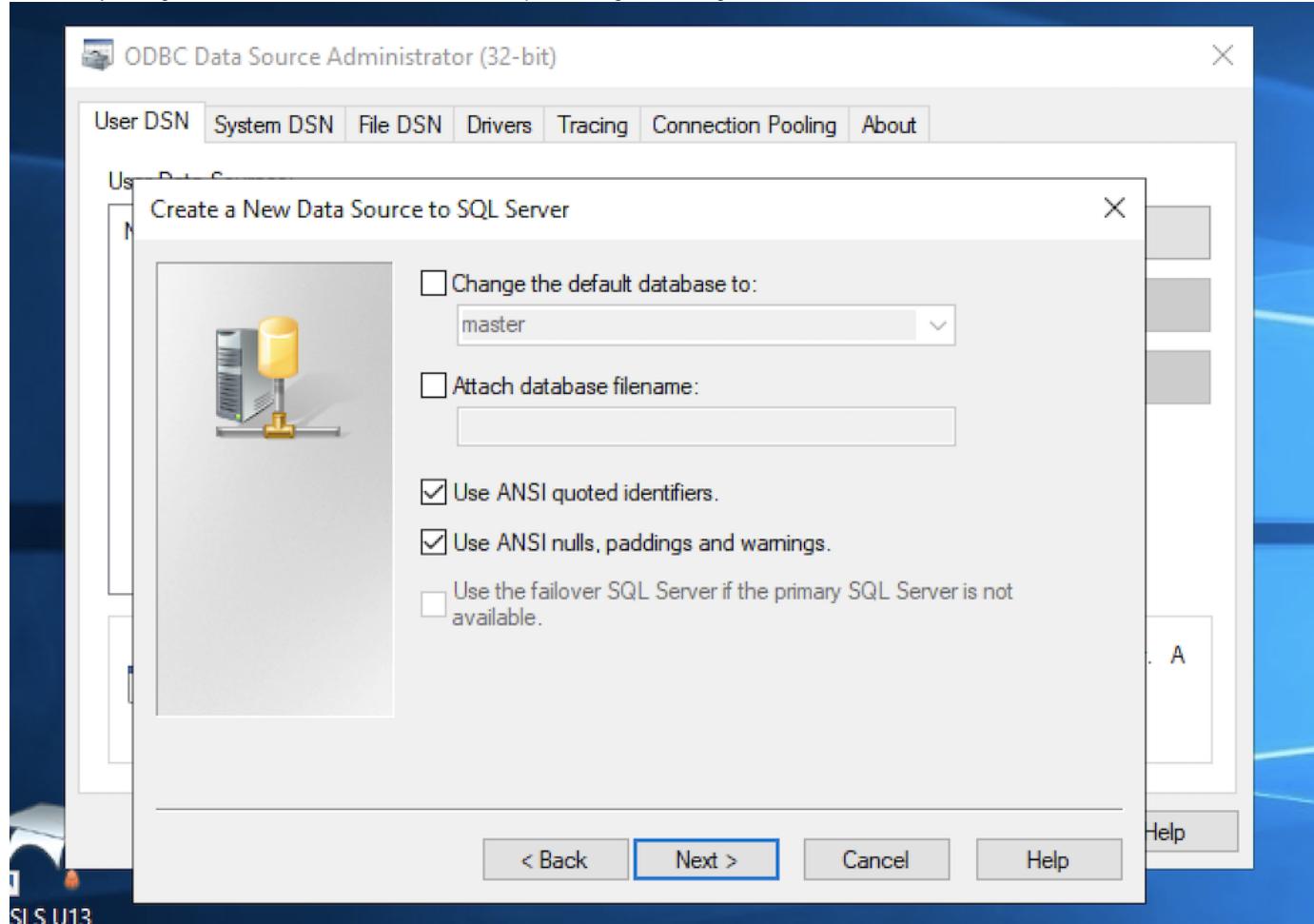




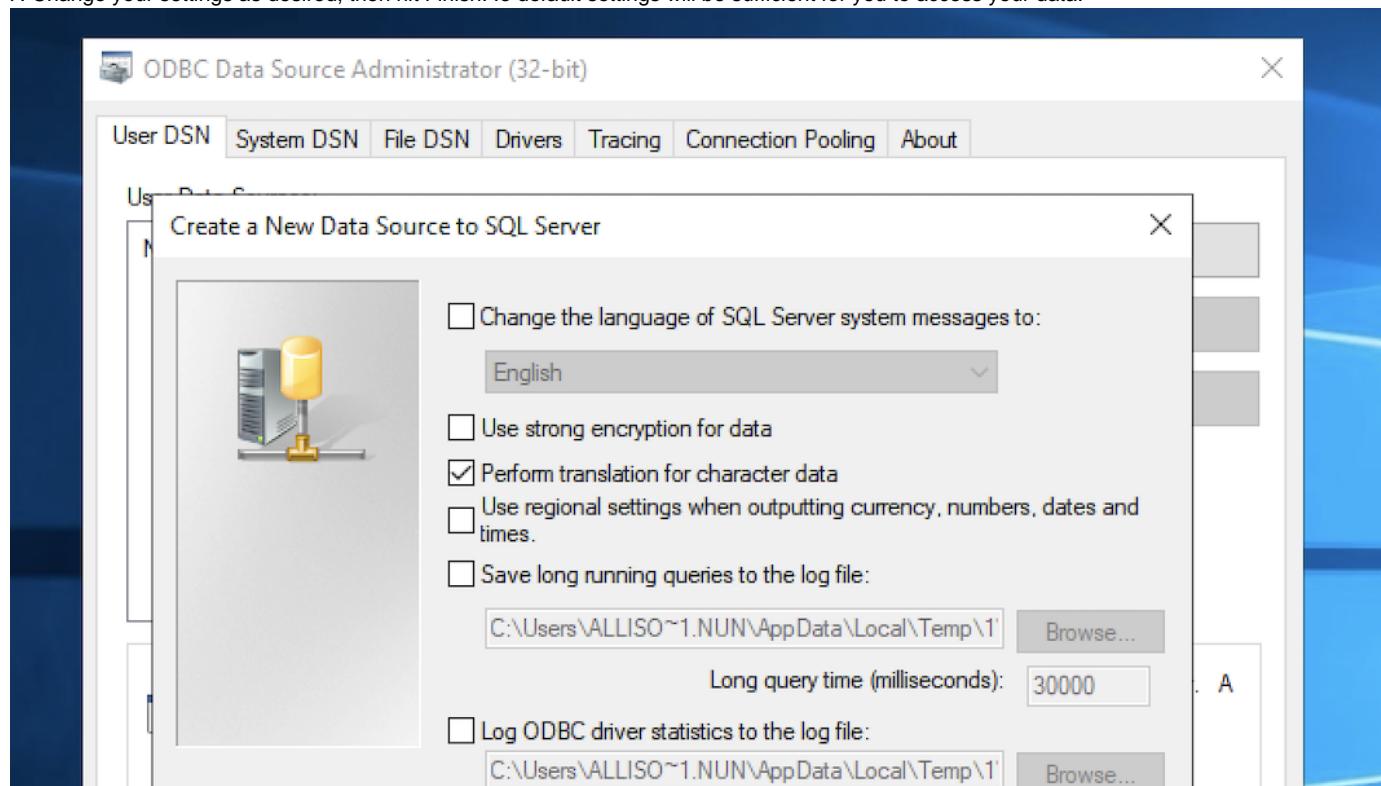
5. Click Next assuming the top option is selected.

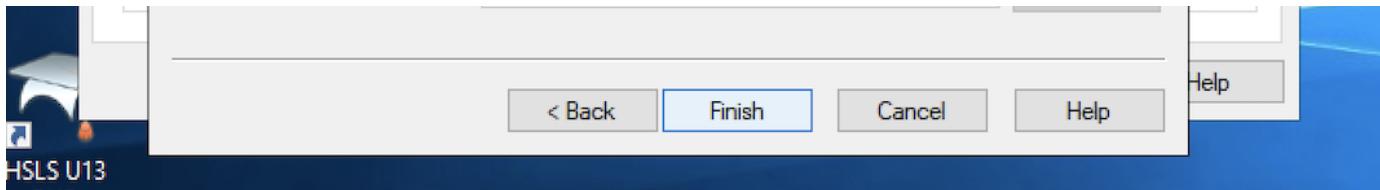


6. You may change the default database, but in this example nothing was changed. Hit Next to continue.

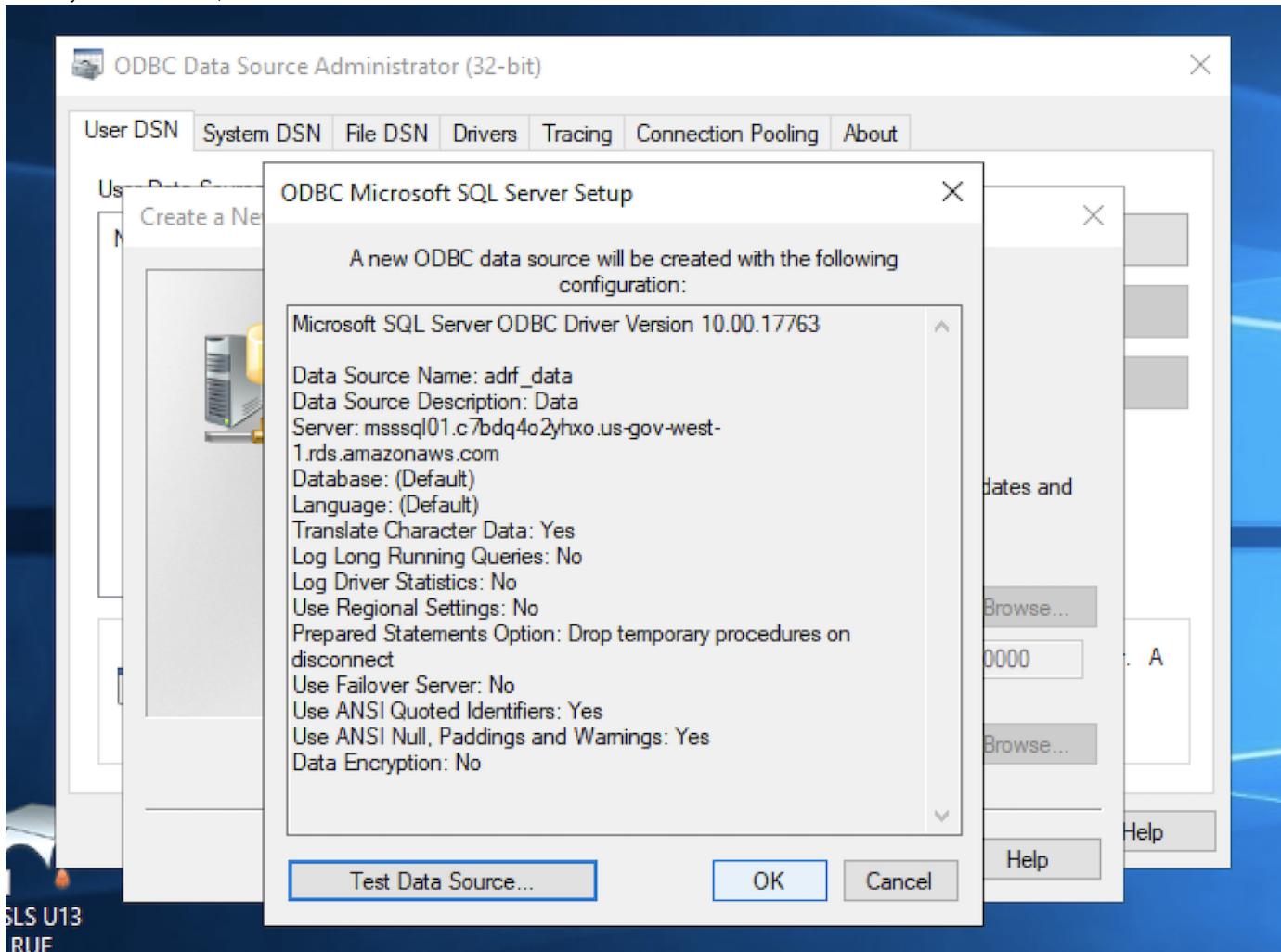


7. Change your settings as desired, then hit Finish. The default settings will be sufficient for you to access your data.

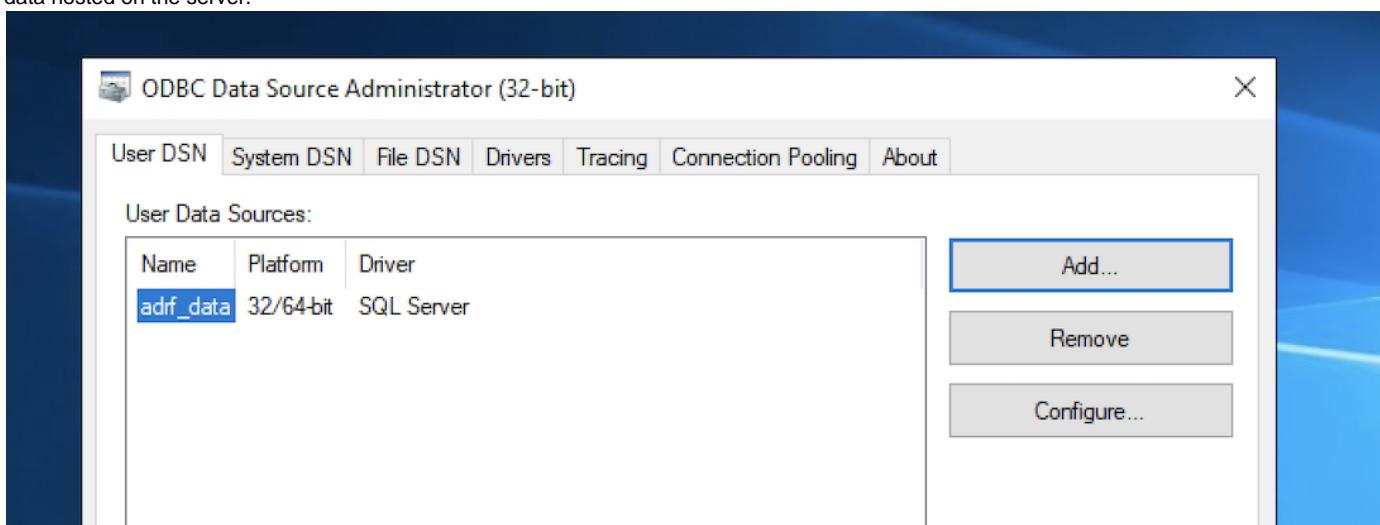


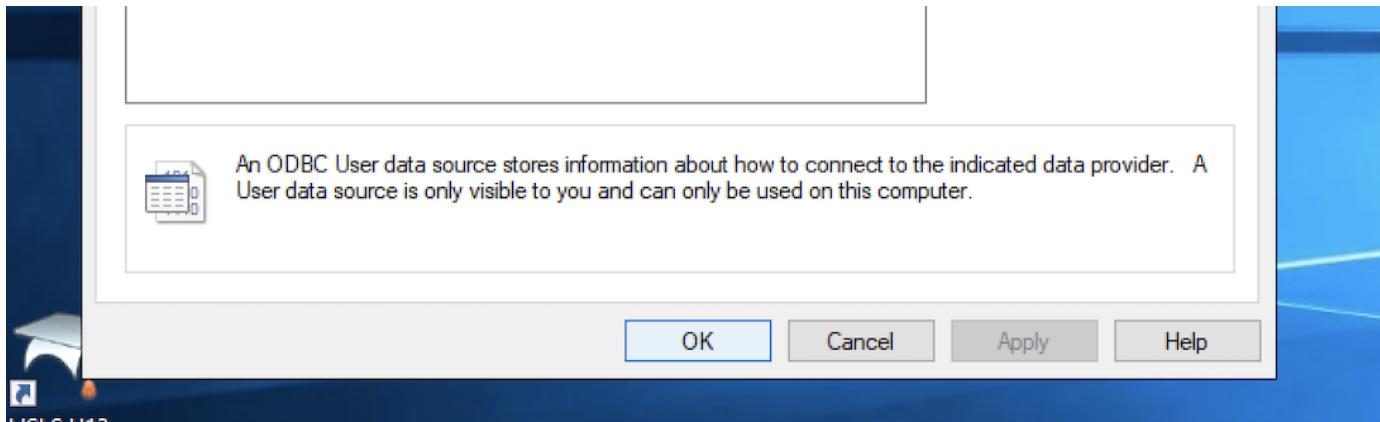


8. Test your data source, then hit OK to continue.



9. Click OK to finish the setup. Note that this process will happen every time you open a new session in the ADRF where you'd like to access data hosted on the server.





Next, open Stata and begin by checking the status of the connection. Do this by running `odbc list` in line. This will display the data source name that you created.

```
. clear  
  
. odbc list  
  
Data Source Name          Driver  
-----  
adrf_data                SQL Server  
MSSQL01                  SQL Server  
-----  
.
```

Command

You can now import data from that external source using Stata commands that will load the resultant data table into memory. See the example below on how to access an example dataset:

The SQL query selects distinct year observations from the table `il_qcew_2018q3_present` in the schema `dbo` from database `ds_el_des`.

```
rap  File  Edit  View  Language  Project  Tools  
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  
Untitled*  X  
1  clear  
2  /*Load data from a database in SQLServer(now called adrf_data) where dsn refers to data source  
3  name*/  
4  odbc_load, exec("SELECT DISTINCT(year) FROM ds_il_des.dbo.il_qcew_2018q3_present") dsn("adrf_data")
```

Note: This will load the data into memory. The dataset is relatively small, with only three observations corresponding to three distinct years in the source dataset.

The screenshot shows the Stata interface. On the left, the command window displays a license message and a command to load data from an ODBC connection. On the right, a variable properties panel is open for a variable named 'year'. The properties panel includes sections for Notes, Data (with fields like Frame, Filename, Label, Notes, Variables, Observations, Size, and Memory), and Properties.

```

979-696-4601 (fax)

Stata license: 20-user 2-core network, expiring 1 Sep 2021
Serial number: 501609329125
Licensed to: Coleridge Initiative
Coleridge Initiative

Notes:
  1. Unicode is supported; see help unicode_advice.
  2. More than 2 billion observations are allowed; see help obs_advice.
  3. Maximum number of variables is set to 5,000; see help set_maxvar.

Checking for updates...
(contacting http://www.stata.com)
server reported server error
http://www.stata.com did not respond or is not a valid update site
unable to check for update; verify Internet settings are correct.

. clear

.
.
.
. odbc load, exec("SELECT DISTINCT(year) FROM ds_il_des.dbo.il_qcew_2018q3_present") dsn("adrf_data")
.

```

Name	Label
year	

Properties

Value label
Notes

Data

Frame	default
Filename	
Label	
Notes	
Variables	1
Observations	3
Size	12
Memory	64M
Sorted by	

For more information on using ODBC in Stata, see the [Stata manual](#).

External Data and Code

Please note that importing of external data and code is restricted to only Coleridge staff. Given the secure and protected environment provided by the ADRF, all code, data, and packages that are coming from outside of the ADRF must be carefully vetted to prevent leaks, disclosure, or unauthorized access. This means that there is no direct method for uploading data or code from your system to the ADRF. Please contact support@coleridgeinitiative.org for any questions or assistance on importing your own code, data, or packages.

Storing Analytic Results

- Eligible Locations
 - User Drive
 - Project Drive
 - SQL
- Ineligible Locations
- Storage Size Restrictions
- Best Practices

Eligible Locations

User Drive

The U: drive is your user drive; it's where you will store any files you are working on. Only the user will have access to the U: drive. For example, if user A wants to share information with user B who is on the same project, user A will need to save files to a P: drive folder and not folders in their U: drive since user B will not be able to access user A's U: drive.

Project Drive

The P: drive also allows permanent storage. This drive is accessible by anyone on the same project, but not across projects. This is the only drive outside of the user drive where saved files will not be erased after logging out of the ADRF.

SQL

Each project will have a project-specific database created. All members of the project will have read and write permissions for data and may also create their own objects (tables, etc.). The project databases are prefixed with pr-.

Ineligible Locations

The G: drive (data), the L: drive (Libs), and the desktop are not eligible for long-term file storing. You won't have permissions to write to either the G: drive or the L: drive. The desktop will function only as temporary storage—as soon as a user is logged out of the ADRF, your desktop will be cleared. Additionally, since Wi-Fi connectivity can be imperfect, desktop storage for any amount of time is not recommended.

Storage Size Restrictions

Storage size varies by project, but is capped at a predetermined amount. Additional storage costs may vary depending on the resource requirements. <https://aws.amazon.com/appstream2/pricing/>

Best Practices

To save storage space, try not to save raw data tables—in particular, don't save copies of or large subsets of data that are already available through standard sources. Instead, access data through the methods described in the prior sections here, as appropriate for your programming language or program.

Organize folders in a way that makes sense for your particular project. For example, you might have folders for a particular analysis or sub-projects. Dates on file names can be helpful for version control.

Keep tabs on how much storage you are using compared to the allocated amount of storage.

Sharing Information within the ADRF

- [ADRF Messenger](#)
- [Shared Folders](#)
- [Sharing Restrictions](#)

ADRF Messenger

The ADRF messenger is an internal collaboration tool and will be made available once testing is complete.

Shared Folders

Shared folders within a project are a great way to share information with other members on a team project. Remember that when working with teams you may not share the ADRF screen (even project folders) with other members on video platforms or otherwise, whether or not your team members are working on the same project.

Sharing Restrictions

Again, remember that when working with teams you may not share the ADRF screen with other members on video platforms or otherwise, whether or not your team members are working on the same project.

The information contained in the ADRF is restricted to reside only in the ADRF for all purposes unless it passes Export Review. This means that it cannot be shared or potentially shared with any unauthorized parties. Do not write down any numbers or figures or tables corresponding to data in the ADRF. Copying and pasting is restricted, but manually circumventing this is also not permitted by your data agreements.

Exporting Results

- [Export Guidelines](#)
 - [Export Review Guidelines](#)
 - [General Best Practices for a Successful Export](#)
 - [Timelines for Export Process](#)
- [Preparing Data for Export](#)
 - [Tables](#)
 - [Graphs](#)
 - [Model Output](#)
- [Submitting an Export Request](#)

Export Guidelines

Export Review Guidelines

To provide ADRF users with the ability to draw from sensitive data, results that are exported from the ADRF must meet rigorous standards meant to protect privacy and confidentiality. To ensure that those standards are met, the ADRF Export Review team reviews each request to ensure that it follows formal guidelines that are set by the respective agency providing the data in partnership with the Coleridge Initiative. Prior to moving data into the ADRF from the agency, the Export Review team suggests default guidelines to implement, based on standard statistical approaches in the U.S. government ^{1,2} as well as international standards ^{3,4}, and ⁵. The Data Steward from the agency supplying the data works with the

team to amend these default rules in line with the agency's requirements. If you are unsure about the review guidelines for the data you are using in the ADRF or if you have any questions relating to exports, please reach out to support@coleridgeinitiative.org before submitting an export request.

To learn more about limiting disclosure more generally, please refer to the [textbook](#) or view the [videos](#).

General Best Practices for a Successful Export

1. Currently, the review process is highly manual: Reviewers will read your code and view your output files, which may be time-consuming.
2. Each additional release adds disclosure risk and therefore limits subsequent releases; we ask that users limit the number of files they request to export to just the outputs necessary to produce a particular report or paper. If you are requesting an export of more than 10 files, there may be an additional charge.
3. The reviewers may ask you to make changes to your code or output to meet the requirements of guidelines that have been given by the providers of the data in the ADRF. Thus, we strongly encourage you to produce all output files—tables with rounded numbers, graphs with titles, and so forth—through code, rather than manually.
4. We ask that you only request review of final versions of output files, rather than in-progress versions. Any file containing intermediate output will be rejected.
5. Every code file should have a header describing the contents of the file, including a summary of the data manipulation that takes place in the file (e.g., regression, table or figure creation, etc.).
6. Documenting code by using comments throughout is helpful for disclosure reviews. The better the documentation, the faster the turnaround of export requests. If data files are aggregated, please provide documentation on the level of aggregation and for where in the code the aggregation takes place.
7. To help reviewers, who may not have seen your code before, we ask that users create meaningful variable names. For instance, if you are calculating outflows, it is better to name the variable “outflows” than to name it “var1.”

Timelines for Export Process

1. Coleridge reviewers have five business days to complete an export from the day you submit an export request. However, timelines may differ depending on your agency, so please refer to your specific agency's guidelines.
2. The review process can be delayed if the reviewer needs additional information or if the reviewer needs you to make changes to your code or output to meet the ADRF nondisclosure requirements.

Preparing Data for Export

Tables

1. Cell Sizes
 - a. Each agency has specific disclosure review guidelines, especially with respect to the minimum allowable cell sizes for tables. Refer to these guidelines when preparing export requests. If you are unsure of what guidelines are in place for the dataset with which you are working in the ADRF, please reach out to support@coleridgeinitiative.org.
 - b. For individual-level data, please report the number of observations from each cell. For individual-level data, the default rule is to suppress cells with fewer than 10 observations, unless otherwise directed by the guidelines of the agency that provided the data.
 - c. If your table includes row or column totals or is dependent on a preceding or subsequent table, reviewers will need to take into account complementary disclosure risks—that is, whether the tables' totals, or the separate tables when read together, might disclose information about individuals in the data in a way that a single, simpler table would not. Reviewers will work with you by offering guidance on implementing any necessary complementary suppression techniques.
 - d. If you are working with data about businesses, report the proportion of the cell count or value accounted for by the largest four businesses in a cell. For business data, the default rule applied by the Export Review team is to suppress counts or values where more than 80% is accounted for by the top four businesses in the cell.
2. Cell Values
 - a. Round all reported values to a sensible unit (e.g., do not report earnings of \$45,675—report \$46,000; do not report employment at 12,345—report 12,000).
3. Weighted Data
 - a. If weighted results are to be exported, you must report both weighted and unweighted counts.
4. Ratios
 - a. If ratios are reported, please report the number of valid cases for both the numerator and the denominator (e.g., number of men in state X and number of women in state X, in addition to the ratio of women in state X).
5. Percentiles
 - a. Do not report exact percentiles. Instead, for example, you may calculate a “fuzzy median,” by averaging the true 45th and 55th percentiles.
6. Percentages
 - a. For any reported percentages or proportions, the underlying counts of individuals contributing to the numerators and denominators must be provided for each statistic in the desired export.
7. Maxima and Minima
 - a. Suppress maximum and minimum values in general.
 - b. You may replace an exact maximum or minimum with a top-coded value.

Graphs

- Graphs are representations of tables. Thus, for each graph (which may have, e.g., a jpg, pdf, png, or tif extension), provide the source data of the underlying table of the graph following the guidelines for tables above.
- Because graphs and other figures take the most time to review, the number of generated graphs should be as low as possible. Please consider the possibility that you could export the underlying table instead, and generate the graph in another package.
- If a graph is produced from aggregated data or from tables that have been disclosure-proofed following the guidelines above (e.g., bar charts of magnitudes), provide the underlying tables.
- If a graph is produced directly from unit-record data but aggregated in the visualization (e.g., frequency histograms), provide the underlying tables.
- If a graph is produced directly from unit-record data and displays unit-record values (e.g., scatterplots, plots of residuals), the graph can be released only after you ensure that individuals cannot be re-identified and that values can only be estimated with a high level of uncertainty. Further processing to meet this requirement can include, but is not restricted to, cutting off the tails of a distribution, removing outliers, jittering the actual values, and removing or modifying axis values.
- If a graph is produced from the results of modeling or derivation and uses the unit-record data (e.g., regression curves), the graph can be released only if the values cannot be used to find original data values.
 - Graphs of this type are generally automatically cleared.
 - For precision/recall graphs, you will need to report the sample size used to generate your model(s).

Model Output

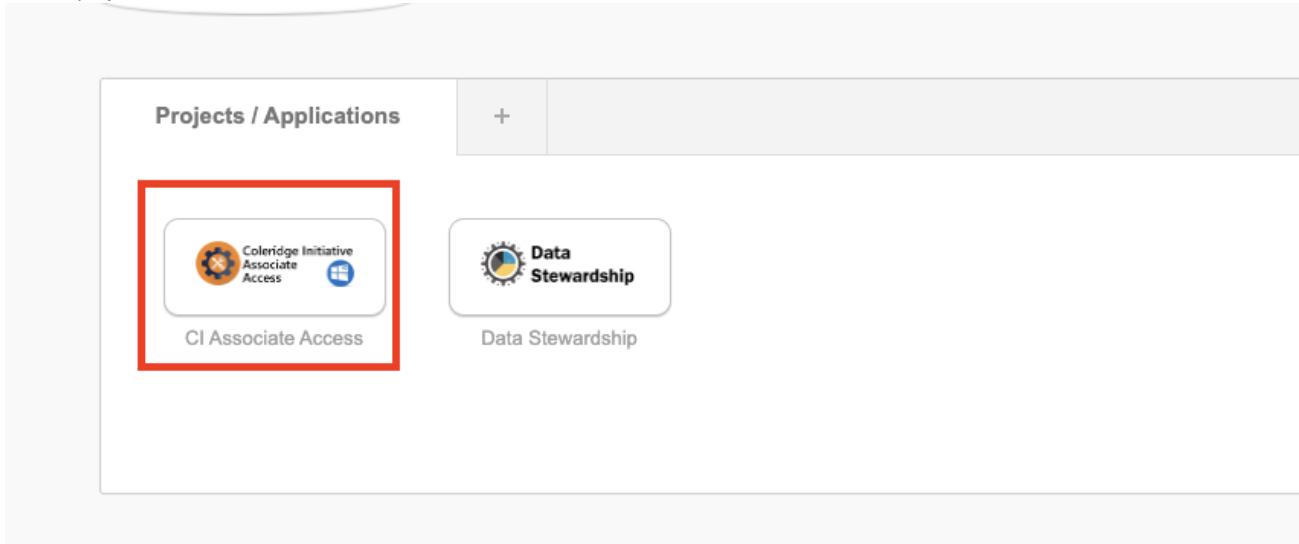
- Output from regression or machine-learning models generally does not pose a risk of disclosing personally identifiable information, as long as the models are not based on small samples. However, only request the release of key coefficients and suppress the coefficients of control variables.

Submitting an Export Request

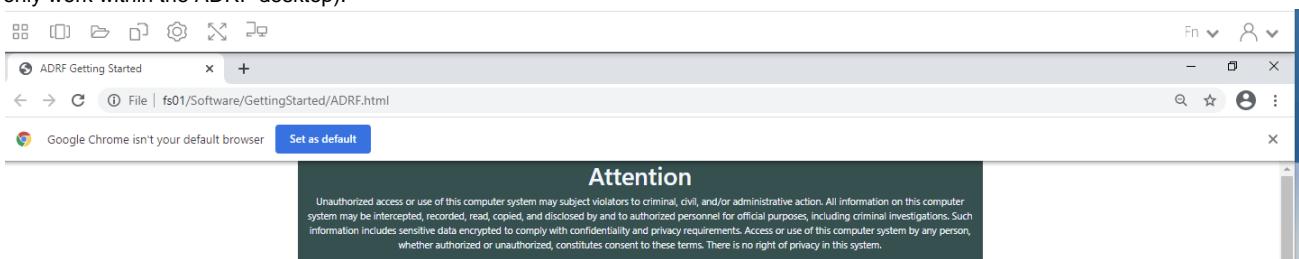
To request an export be reviewed, please watch the following video or follow the instructions below:

Export Video

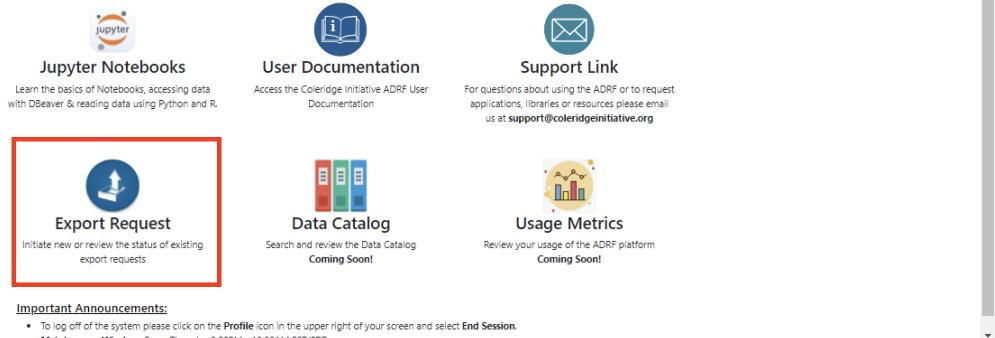
- Click here: <http://adrf.okta.com> (ADRF 3).
- Input your login credentials.
- Verify yourself with Okta (download Okta Verify on your smartphone or other device).
- Choose your project as seen in the photo below. For the purpose of this document, you are seeing the Coleridge Initiative Associate Access project.



- Select Desktop and login with the same credentials you had done previously.
- Upon entering the ADRF, a chrome page will appear as shown in the photo below. On this page, click Export Request in the bottom left corner. Or, from the ADRF desktop, open Google Chrome and navigate to export.adrf.net. (Note: export.adrf.net is an address that will only work within the ADRF desktop).



Please keep in mind that any removal of data is prohibited. This includes taking screenshots or writing down numbers on paper. All output has to go through disclosure review. Violations of this policy may result in penalties and revoking access from the system.



7. Click My Requests, or the top (person-shaped) icon, at the left side of the window as shown in the screenshot below.

The screenshot shows the "My Requests" page under the "EXPORT" section. The "My Requests" link in the sidebar is highlighted with a red box. The main table displays one request:

Date Request	Project Name	Primary Reviewer	Secondary Reviewer	Last Status	Last Status Date	Actions
2021/05/03 19:11:21	CI_Admin	Nathan Caplan		Rejected by Reviewer	2021/05/06 20:34:04	

At the bottom right of the table, there are pagination controls: Rows per page: 10, 1-1 of 1, and navigation arrows.

8. Click New Item as shown below.

The screenshot shows the "New Export Request" page. The "NEW ITEM" button in the top right corner is highlighted with a red box.

- You will be asked to select the project to which your export relates. If you do not see the correct project listed in the dropdown list, please reach out to our support team at support@coleridgeinitiative.org.
- After selecting a project, click Continue.

The screenshot shows the "New Export Request" form. The "Choose a project..." dropdown menu is open, showing the option "DOL ETA".

CANCEL

CONTINUE

9. Read through the entire page that loads. This page, titled "Create Export Request," will ask for you to comment on all supporting code files to explain the commands used to generate the files in the export request. The Export Review team will reject all requests containing intermediate output, and there should be no more than 10 separate files for export unless approval is given in advance. The Export Review team will typically release export requests within five business days. However, if the team has any clarifying questions, this could result in a longer review process. You need to document your output files in the text box provided. See the example below:

Create Export Request

Project Name
DOLETA

This form will provide the disclosure review team with enough information to evaluate the files in your export request. You must adequately comment on all supporting code files to explain the commands used to generate the files in the export request. We will reject all requests containing intermediate output, and we expect no more than 10 separate files for export unless approval is given in advance. We typically release export requests within five business days. However, if our team has any clarifying questions, we will reach out to you within two business days which may result in a longer review process. Short, well-documented requests will be prioritized.

If you have any questions regarding how to fill out this form, please contact support@coleridgeinitiative.org. Please do not include any direct references to specific data elements in these emails.

Document your output files

For each file requested for export, please enter the following information in a neatly bulleted list—

- File name
 - The source dataset(s) used to generate the output file. If a subset is used, describe the sample restrictions.
 - Program(s) that produced the file (e.g., R, Stata, Python, etc.)
 - File name containing underlying research sample counts for this file
 - File name(s) that contain supporting statistics required with the dataset
 - File name(s) containing the code used to create this file
 - Any additional comments to help the reviewers to understand the file or its context.

To your knowledge, do the outputs you are submitting with this form follow the guidelines above?

10. When you have read through and followed the page instructions, and are ready to proceed:

- a. Move the slider at the bottom of the page to indicate that you have followed the page's guidelines.
- b. At the bottom of the page, upload each of the files that you have prepared.
- c. Click Submit Request... to create the export request.



To your knowledge, do the outputs you are submitting with this form follow the guidelines above?

Files for Export Documentation

Files can be dragged and dropped here

Files for Export

Files can be dragged and dropped here



Are there any additional emails to send exported files to (text output, not required)?

CANCEL

SAVE DRAFT

SUBMIT REQUEST...

11. You can click My Requests at the left side of the window to view your current and previous export requests.

- a. To learn more about exporting results, please watch these [videos](#).

SVN Documentation

How to use SVN inside the ADRF

- [To see SVN's complete documentation, click here.](#)

Initial Setup

Commit / Update Steps

Creating a Branch

Using SVN Within R Studio

- [To see documentation, click here.](#)

Creating a New Project Based on SVN Repository

[To see SVN's complete documentation, click here.](#)

Initial Setup

First, users need to request that a repository be setup for their project by emailing support@coleridgeinitiative.org. Once users have a repository, follow the steps below to setup and use SVN.

Note: Users cannot create their own repository in SVN.

1. In your U: Drive, create a folder where you want to store your shared project files.
2. Open Google Chrome or Internet Explorer and type the URL into the browser to access SVN: <https://svn.adrf.net> or click on the Code Repository shortcut icon on the ADRF Getting Started page.

User Documentation
Access the Colridge Initiative ADRF User Documentation

Support Link
For questions about using the ADRF or to request applications, libraries or resources please email us at support@colridgeinitiative.org

Export Request
Initiate new or review the status of existing export requests

Data Catalog
Search and view the Data Catalog Coming Soon

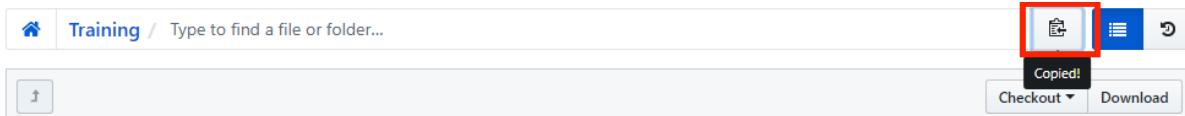
Frequently Asked Questions
Review the most common questions and answers for using the ADRF.

Code Repository
Access the ADRF Repositories [Here](#)

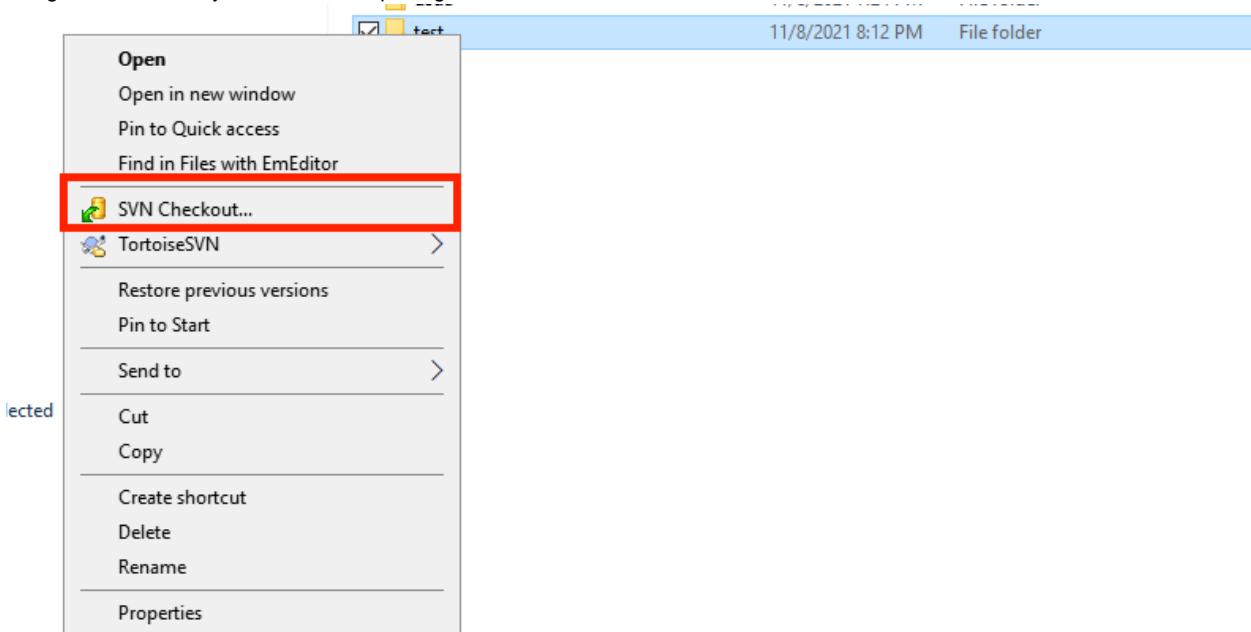
Important Announcements:

- To log off of the system please click on the **Profile** icon in the upper right of your screen and select **End Session**.
- Maintenance Window: Every Thursday 8:00PM - 12:00AM EST/EDT

3. Navigate to the repository you want to copy the URL for.
4. Copy the URL of the repository by clicking on the copy icon:

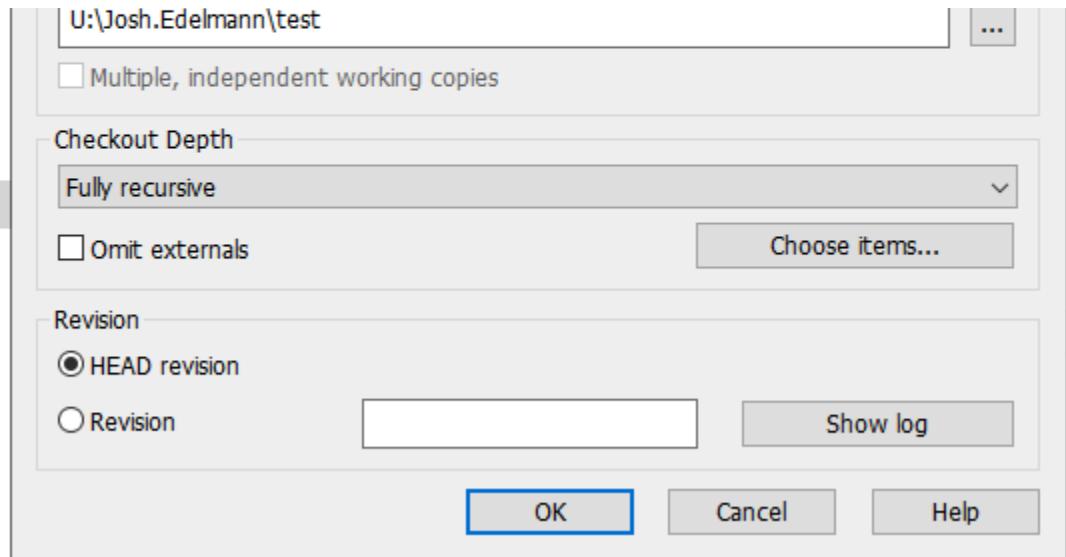


5. Navigate to the folder you created in step 1, right-click and choose "SVN checkout".



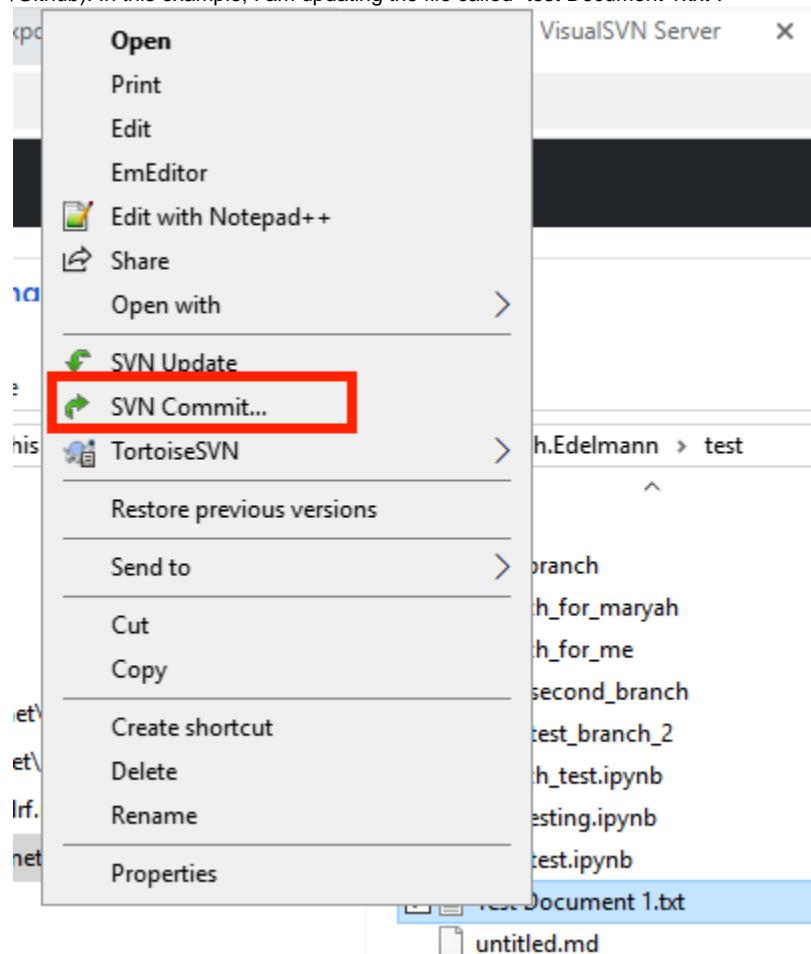
6. In the next window, enter the copied URL under "URL of repository" which in this example is "<https://svn.adrf.net/svn/Training/>". You can specify the checkout depth. Full recursive will checkout the entire tree, including all child folders and sub-folders. For details on other options, [check out the documentation](#). Then click "OK". This will pull the files from the shared repository into your U: drive. Note: your URL will be different from the example, this is only for show.





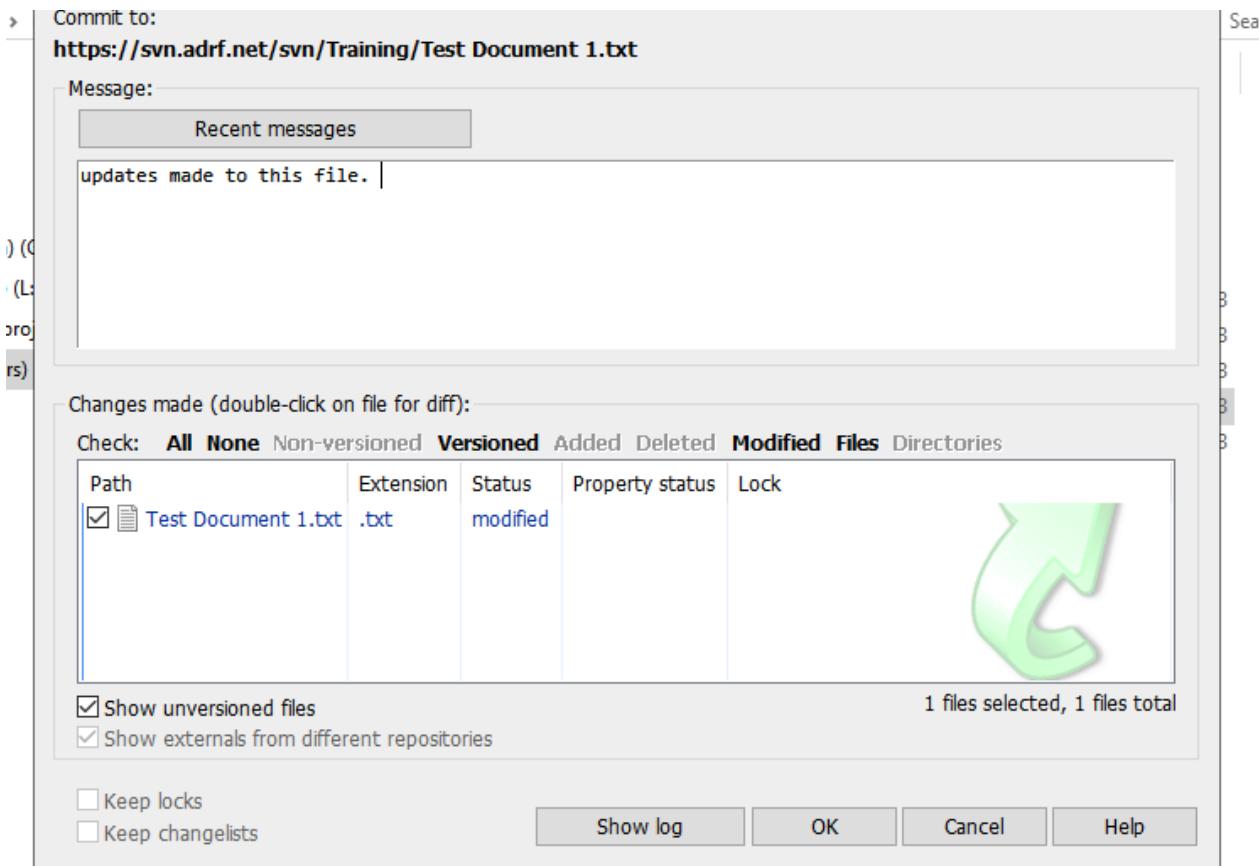
Commit / Update Steps

1. To make an update to the shared repository, once an edit is made in the file or folder, right-click and choose "SVN Commit" (like in Gitlab /Github). In this example, I am updating the file called "test Document 1.txt":

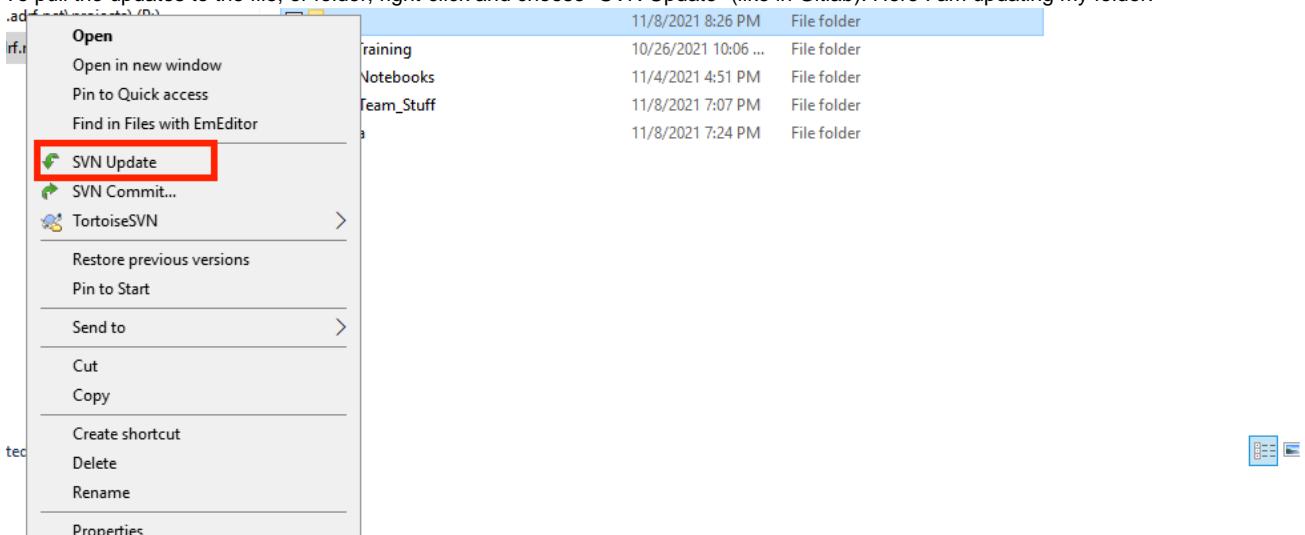


2. Then a box will appear where you can write a message, show the log, commit, or cancel. Here I am adding a message to commit with the file, but you do not need to add a message for the commit to work. Then, click OK.





3. To pull the updates to the file, or folder, right-click and choose "SVN Update" (like in Gitlab). Here I am updating my folder:

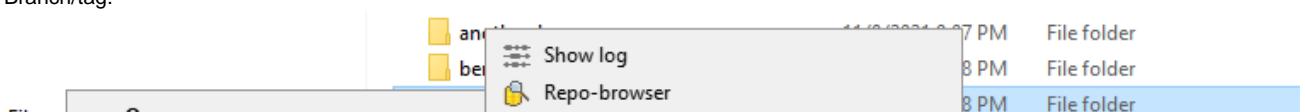


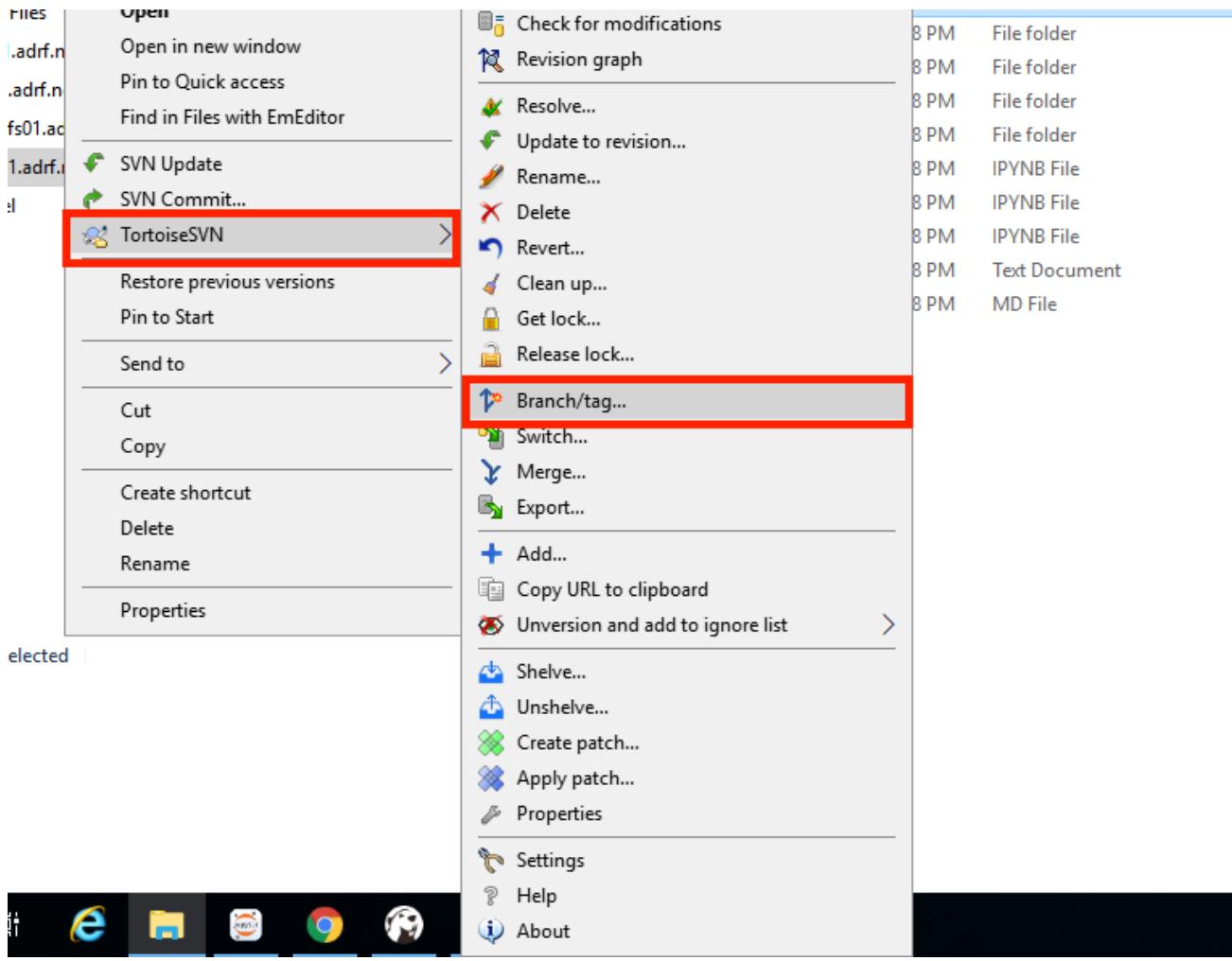
4. To delete files in the shared repository, first delete the file from your personal folder. Then commit the updated folder to the SVN shared repository.

Creating a Branch

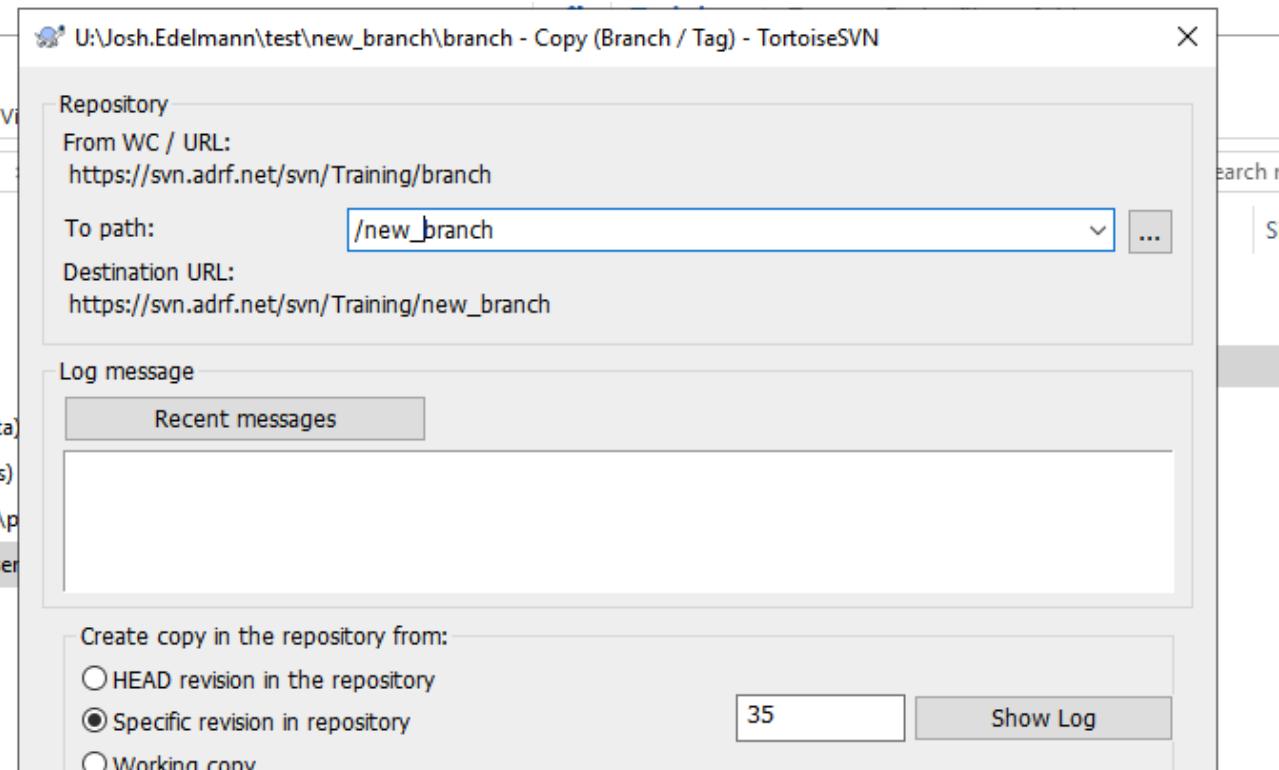
For more information on creating branches, we recommend looking at the section on
[Branching / Tagging in the SVN Daily Use Guide for Windows](#) available [here](#).

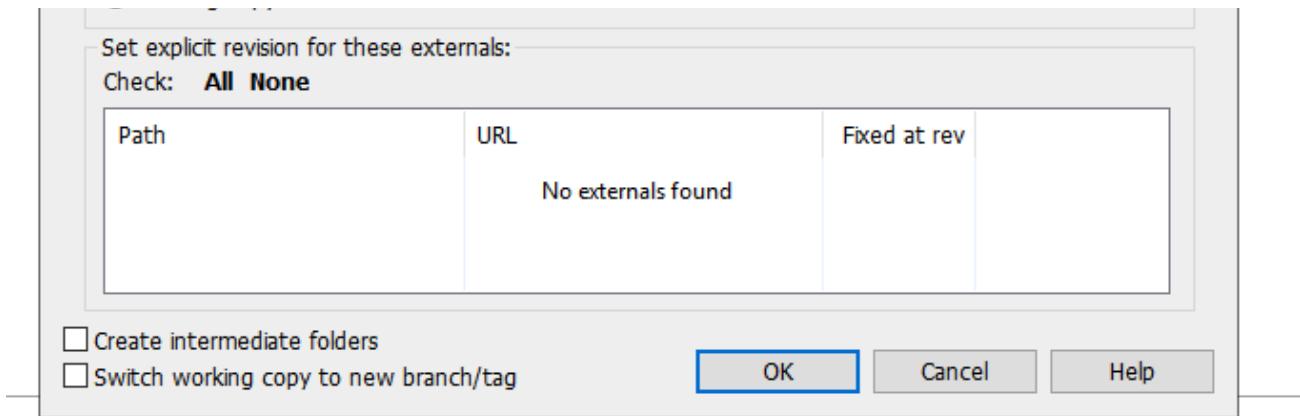
- To create a branch, select the folder on which you want to create a branch, then right click and select TortoiseSVN and then select Branch/tag:





2. A new screen will appear, you will need to rename the path to the branch in the repository. Then, select the option for "Create copy in the repository from:" section. After completing and reviewing the required information, click "OK". Below, I am naming my path "new_branch" and selecting "Specific revision in repository."





3. You can confirm a new branch was created by navigating to the SVN shared repository. Note: your repository will look different from the example below, this is for illustration purposes only.

Path	URL	Fixed at rev
No externals found		

Create intermediate folders Switch working copy to new branch/tag **OK** Cancel Help

Training / Type to find a file or folder... Checkout ▾ Download

File	Size	Revision	Author	Last Modified
another_br		r36	Josh.Edelmann	2 days ago
ben_branch		r13	Ben.Feder	a month ago
branch		r39	Josh.Edelmann	2 minutes ago
branch_for_maryah		r32	Josh.Edelmann	a month ago
branch_for_me		r23	Maryah.Garner	a month ago
josh_second_branch		r33	Josh.Edelmann	a month ago
josh_test_branch_2		r15	Josh.Edelmann	a month ago
new_branch		r37	Josh.Edelmann	2 days ago
branch_test.ipynb	2 KB	r11	Josh.Edelmann	a month ago
For_testing.ipynb	4 KB	r6	Ekaterina.Levitskaya	2 months ago
josh_test.ipynb	2 KB	r10	Josh.Edelmann	a month ago
rAdding.R	1 KB	r40	Josh.Edelmann	2 minutes ago
Test Document 1.txt	1 KB	r38	Josh.Edelmann	10 minutes ago
untitled.md	1 KB	r26	Josh.Edelmann	a month ago

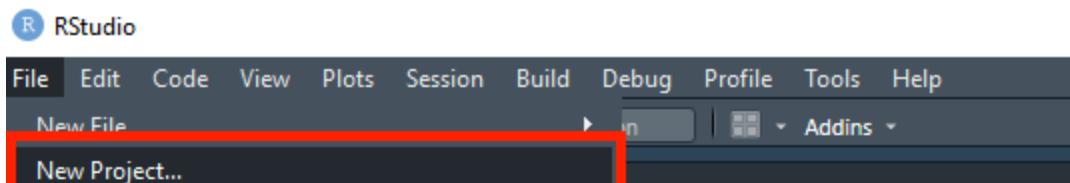
Powered by VisualSVN Server | Report a problem

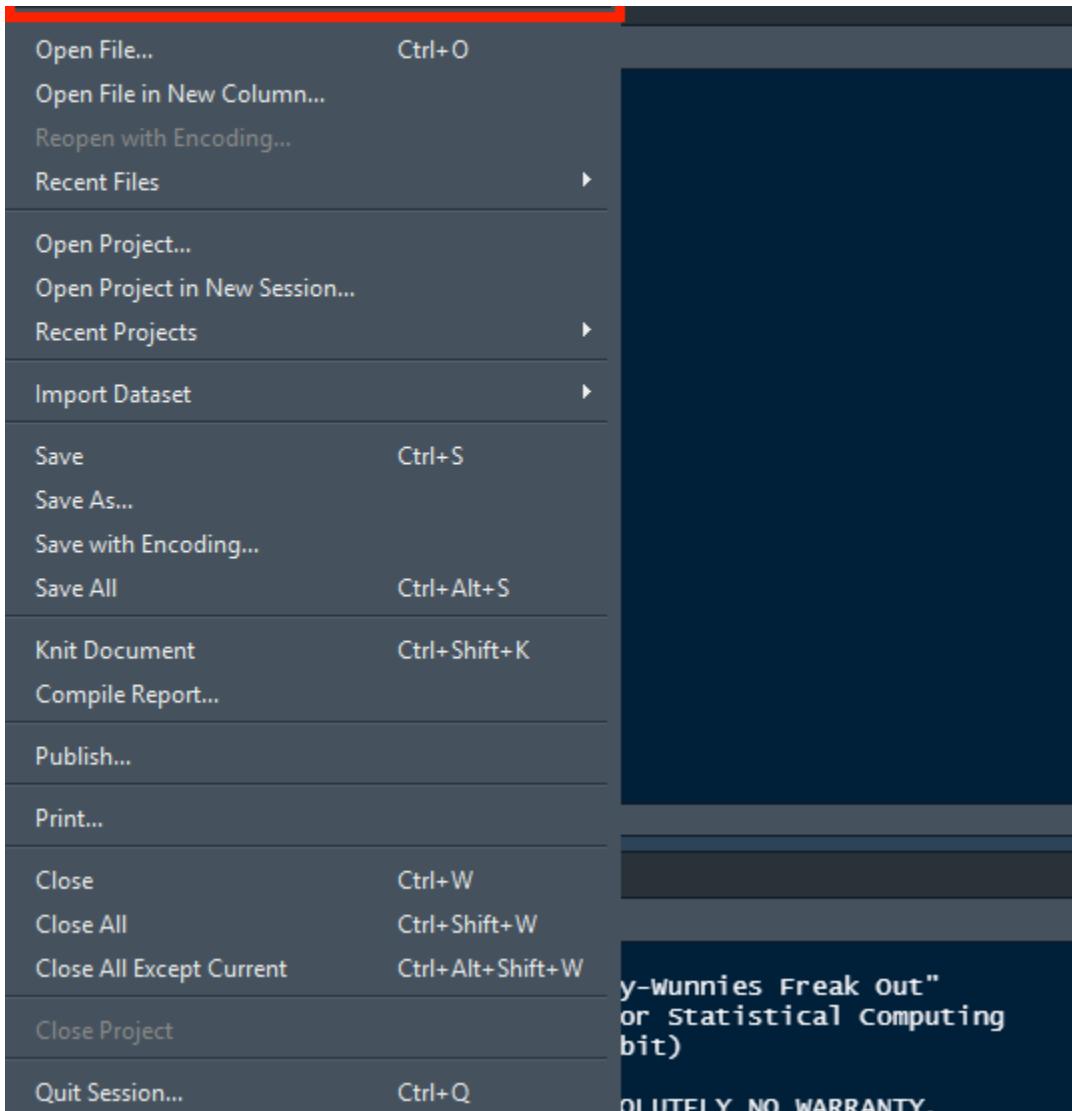
Using SVN Within R Studio

To see documentation, [click here](#).

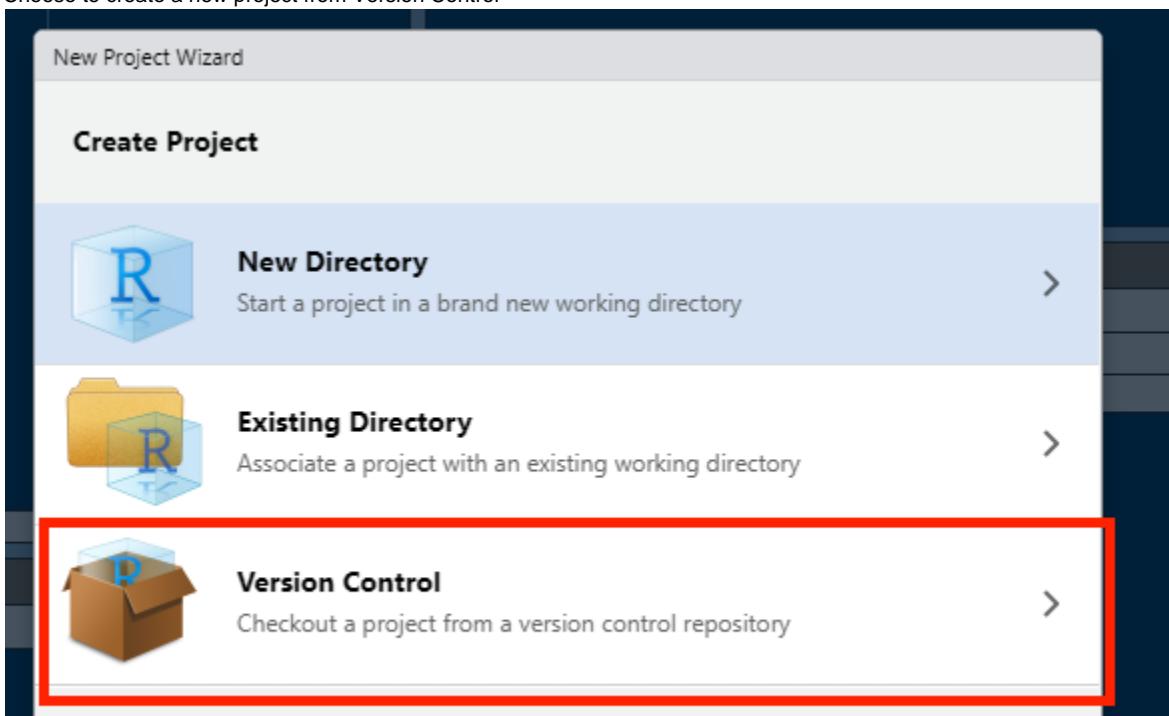
Creating a New Project Based on SVN Repository

- To commit R files to the shared repository, you first need to create a new project. This is where you will store your .R files to commit to the shared repository. To do so, execute the New Project command (from the Project menu)

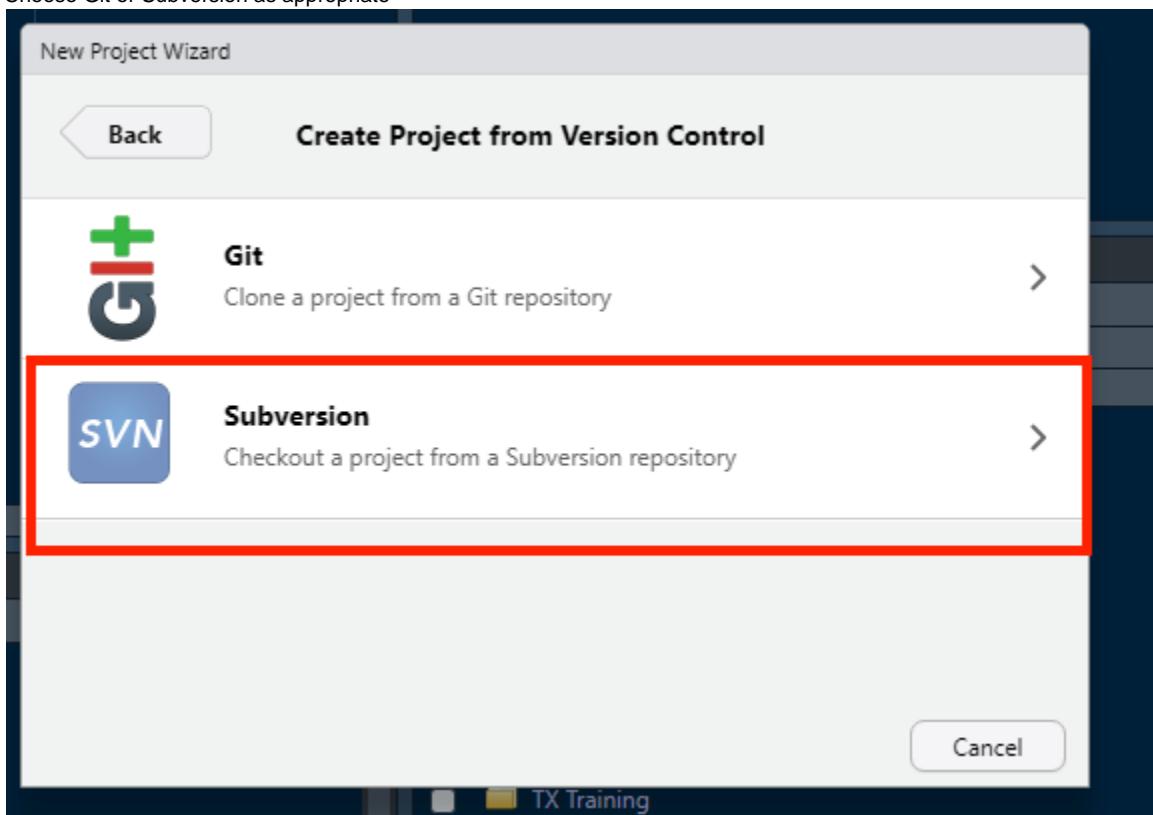




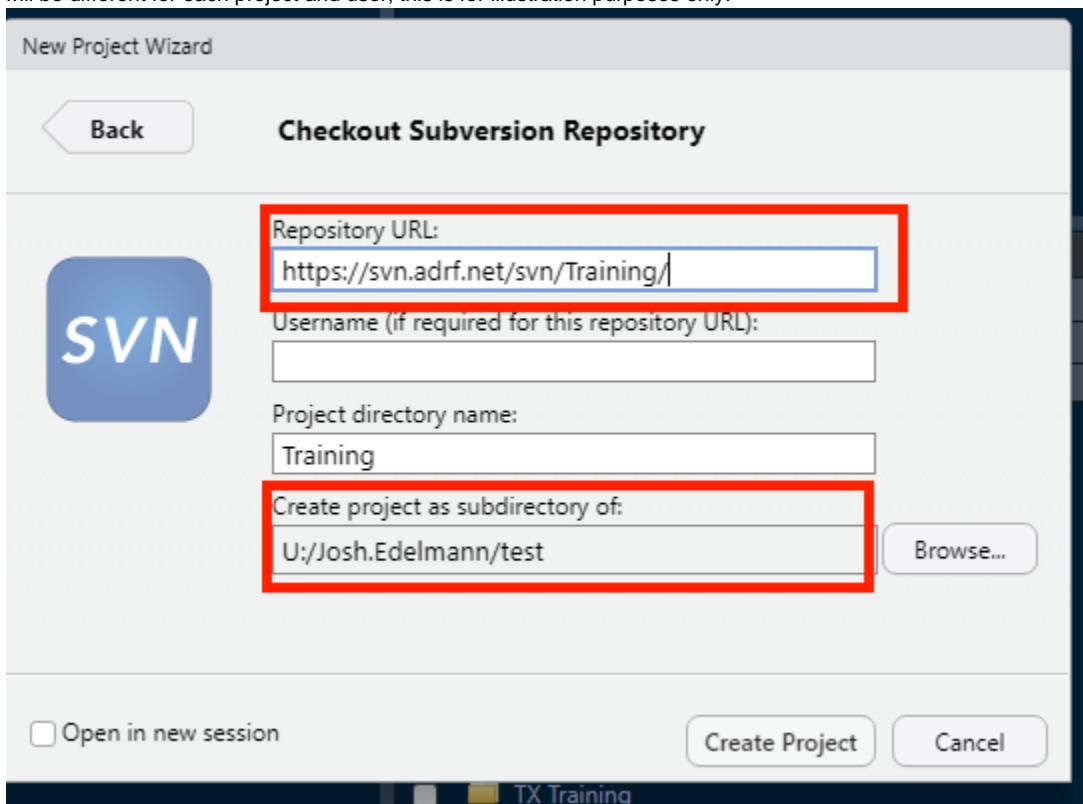
- Choose to create a new project from Version Control



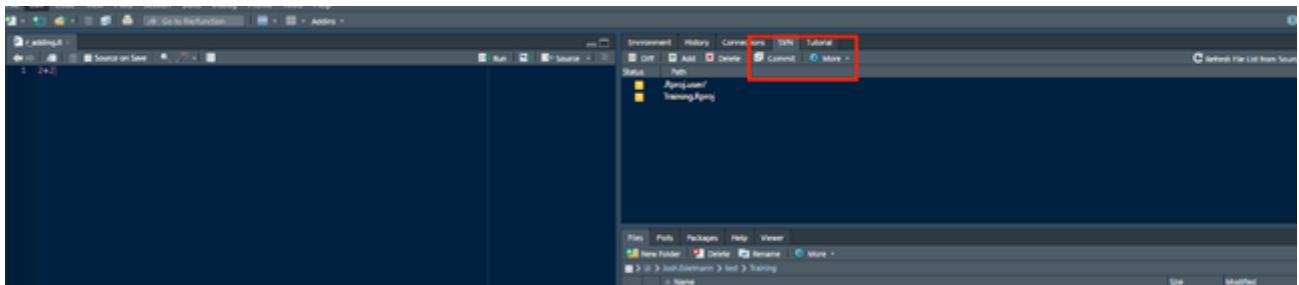
3. Choose Git or Subversion as appropriate



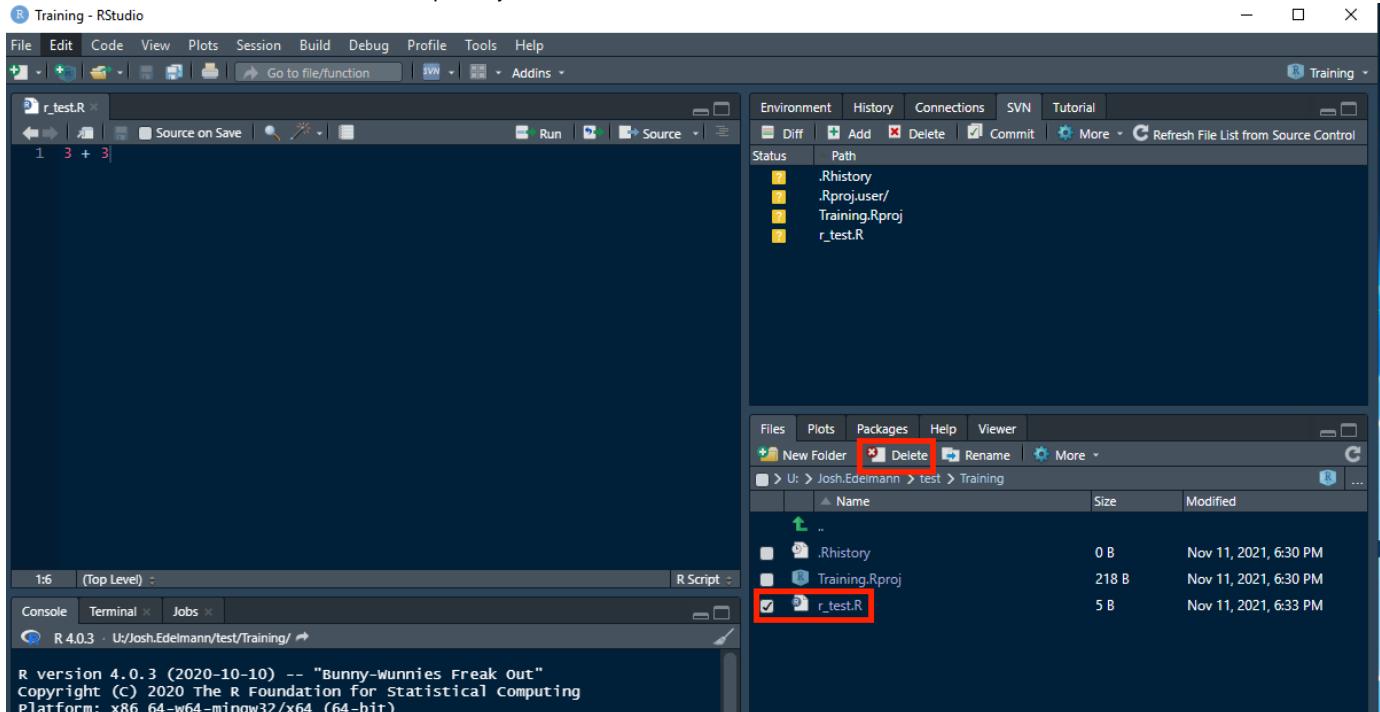
4. Provide the repository URL (and other appropriate options) and then click Create Project. Note: the repository URL and the subdirectory will be different for each project and user, this is for illustration purposes only.



5. Once you create the project, you can commit and update the repository from within R Studio by selecting the "SVN" tab and clicking "Commit" or the "More" menu.



6. To delete a file from a shared repository, first check the box of the file you wish to delete. Then click the “Delete” icon. You will then want to commit this action to the shared repository.



If you have any questions, please contact support@coleridgeinitiative.org.

Technical Support

For ADRF technical support, please email support@coleridgeinitiative.org

References

1. Confidential Information Protection and Statistical Efficiency Act of 2002. (Washington, DC: U.S. GPO, 2002).
2. Federal Committee on Statistical Methodology. “Report on Statistical Disclosure Limitation Methodology,” 22 (Second Version, 2005). <http://nces.ed.gov/fcsm/pdf/spwp22.pdf>.
3. “How to Use Microdata Properly: Self-Study Material for the Users of Eurostat Microdata Sets.” (2018). <https://ec.europa.eu/eurostat/web/microdata/overview/self-study-material-for-microdata-users>.
4. Research Data Centre of the German Federal Employment Agency at the Institute for Employment Research. “Remote Data Access and On-Site Use at the FDZ of the BA at the IAB.” (2020, December 8). http://doku.iab.de/fdz/access/Vorgaben_DAFE_EN.PDF
5. Welpton, Richard. Handbook on Statistical Disclosure Control for Outputs. (figshare, 2019). <https://doi.org/10.6084/m9.figshare.9958520.v1>.