Google Sheets R API & Shiny!
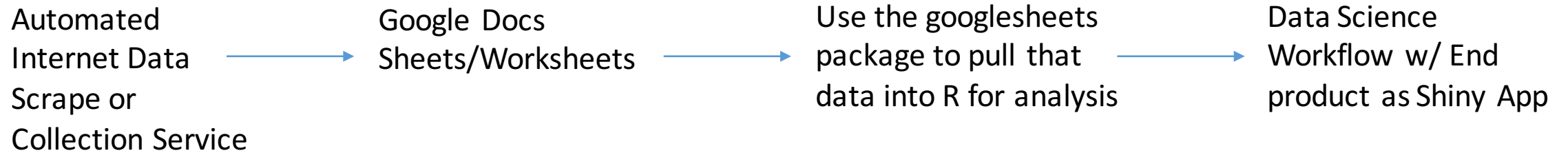
Access and manage Google Spreadsheets from R with `googlesheets`
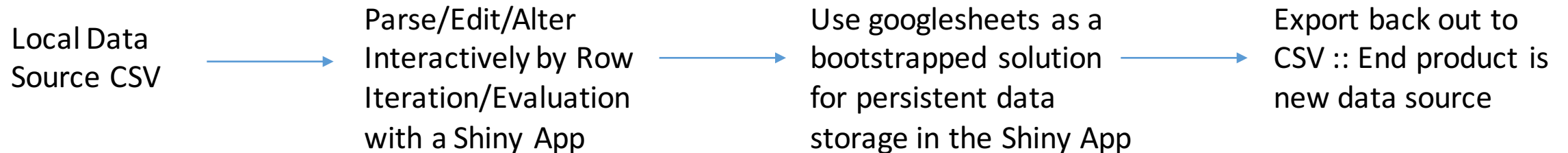
Googlesheets – So cool!

- Register a Sheet: gs_title() or gs_key()
- Inspect a Sheet: gs_ws_ls()
- Return a dataframe of sheets: gs_ls()
- Read all the data in one worksheet: gs_read(ws="…")
- Create a new sheet: gs_new()
- Add a new worksheet to an existing sheet: gs_ws_new()
- Edit cells: gs_edit_cells()

Github Repo: https://github.com/jennybc/googlesheets
Basic Usage Vignette: https://rawgit.com/jennybc/googlesheets/master/vignettes/basic-usage.html

What to do with googlesheets & Shiny?

Automated Internet Data Scrape or Collection Service → Google Docs Sheets/Worksheets → Use the googlesheets package to pull that data into R for analysis → Data Science Workflow w/ End product as Shiny App

Two Interesting Content Management Paradigms to Consider…

Local Data Source CSV → Parse/Edit/Alter Interactively by Row Iteration/Evaluation with a Shiny App → Use googlesheets as a bootstrapped solution for persistent data storage in the Shiny App → Export back out to CSV :: End product is new data source

Google Auth Tokens Overview

- Some googlesheets activities require Auth Tokens and some do not
  - Not Req: Reading from a sheet that is published to the web identified by key or URL
  - Auth Req: Any activity from a sheet identified by name/title, and all interactive actions
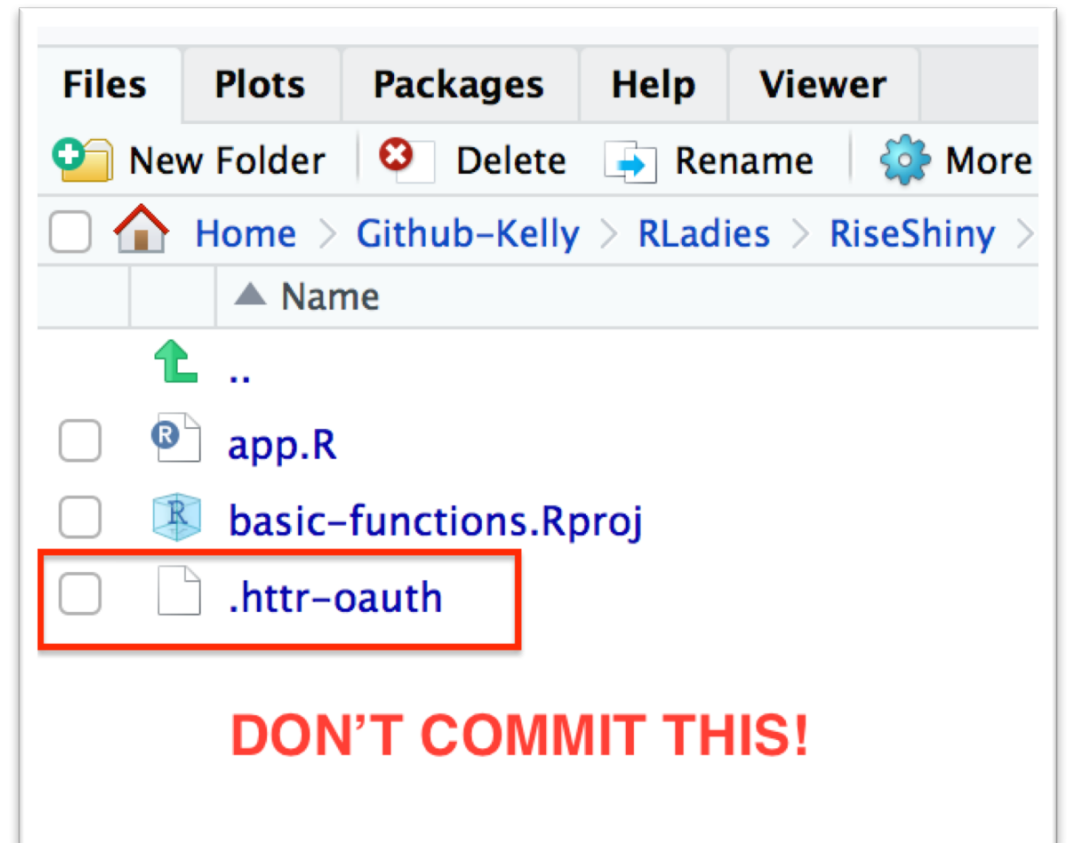- Check for a valid token in your Env: gs_user()

Google Oauth 2.0 Flow
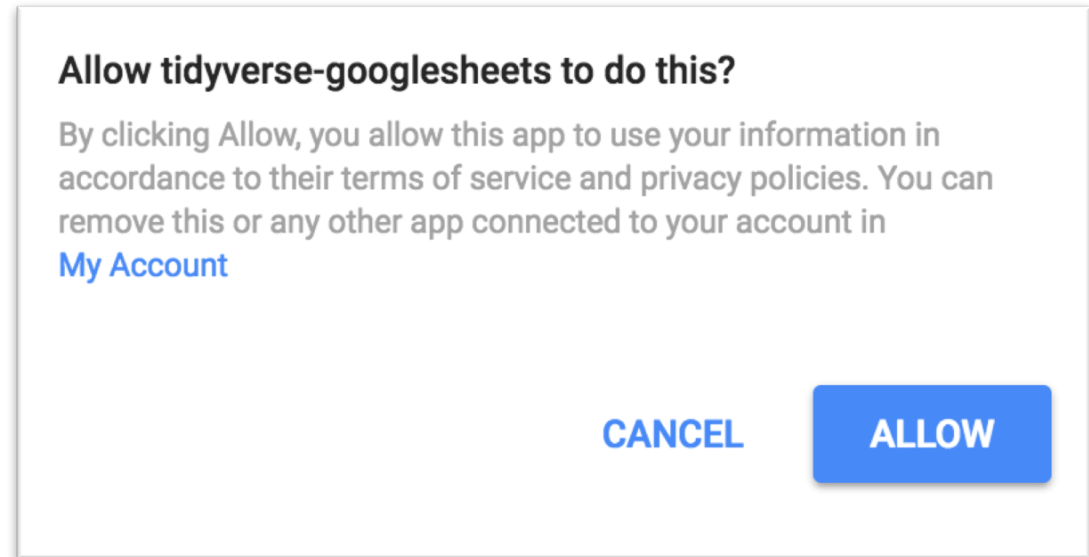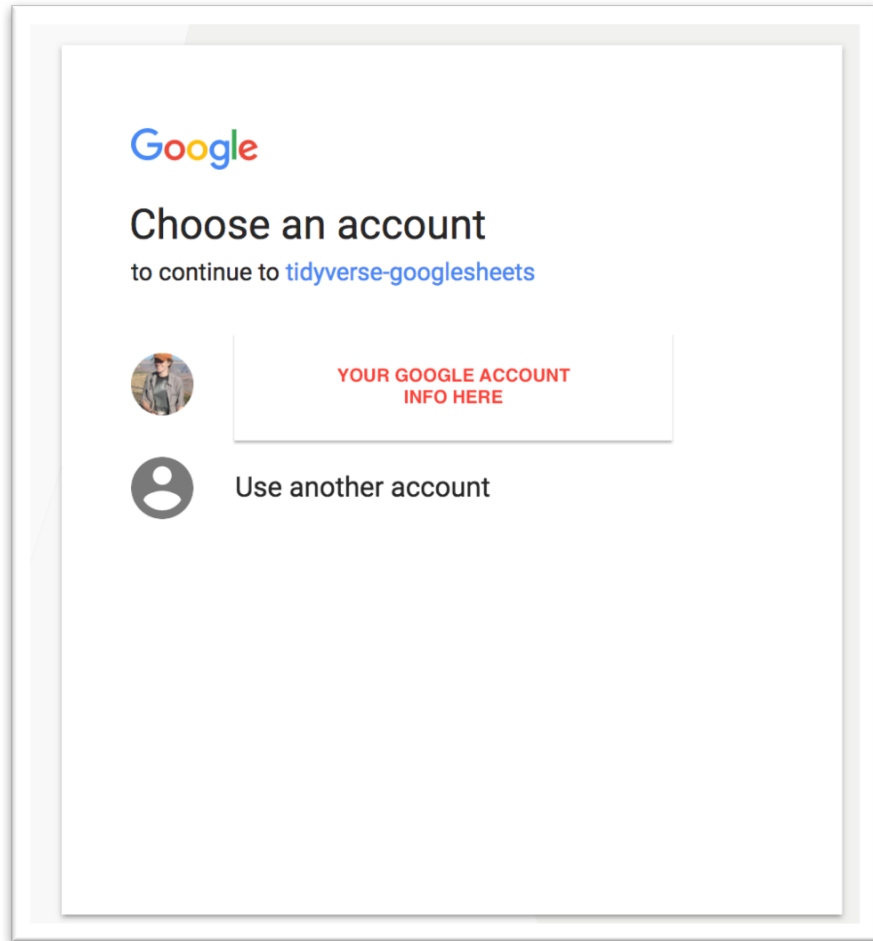
Tokens WARNING – DO NOT PUBLISH

Do not push your .httr-oauth config file to Github

Be sure to list .httr-oauth in your .gitignore file!

Workshop

1. Create a new Google Sheets document in your Docs account and populate it with some data.

   - Check out https://www.kaggle.com/datasets for some data inspiration

2. Build a Shiny App that will:

   - Take user input from a text field [your new Google sheet title]
   - Use the googlesheets package to register that sheet
   - Display the sheet as a Data Table in the main body of the app

3. Start from scratch, or use this template to get started:

   - Template on Github

Template on Github: https://github.com/rladies/meetup-presentations_dc/blob/master/RiseAndShiny/January2018/WorkshopChallengeTemplate/app.R

Extension Challenges – Choose your own Adventure

1. **"Fail Gracefully"** – Add a nice error handling solution for when the user asks for a sheet title that isn't available (or they spell the name wrong)

2. **"Search by Key"** - Add an additional optional input for getting a sheet by its Key instead of its name/title

3. **"Next Steps"** – Add a feature that will allow a user to remove columns from the data that they don't want to see
   - This could be done before the sheet is read in, after the sheet is read in, or possibly both before AND after