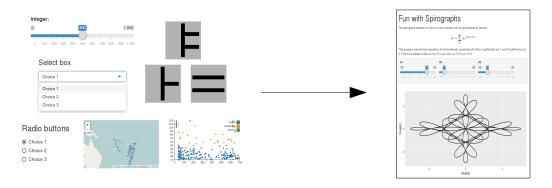


Developing with R on HUBzero



Derrick Kearney

HUBzero® Platform for Scientific Collaboration
Purdue University

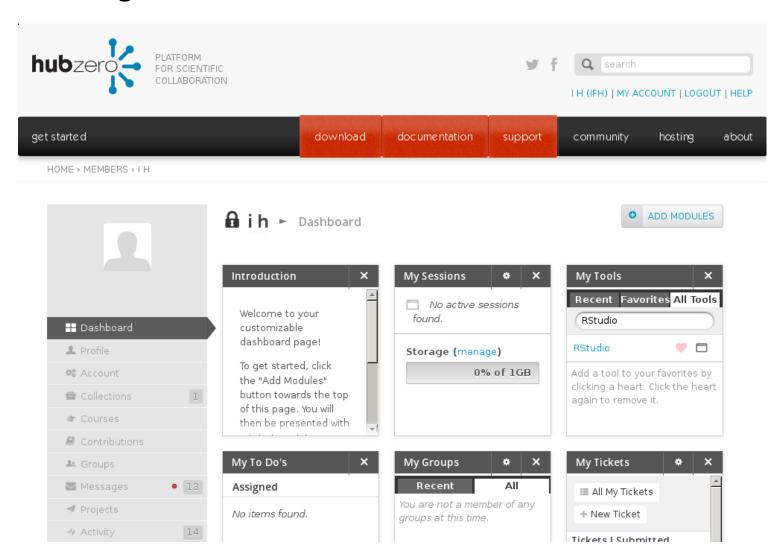
This work licensed under Creative Commons



See license online: by-nc-sa/3.0

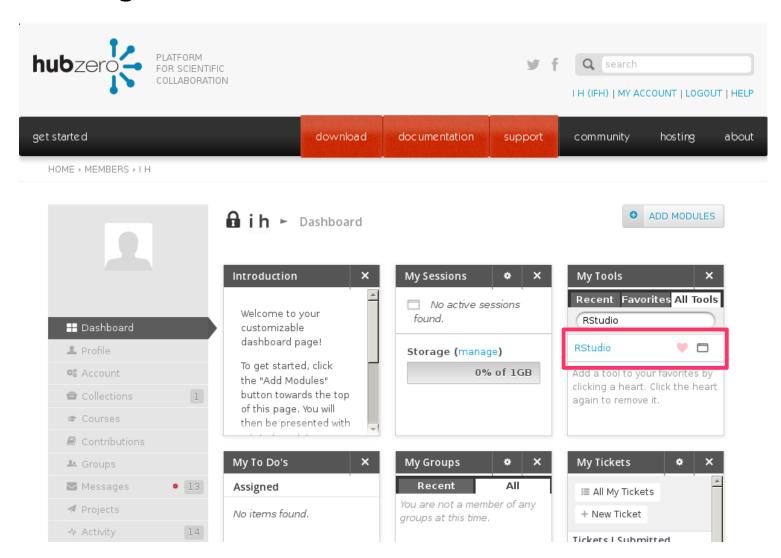
Launching RStudio From HUBzero





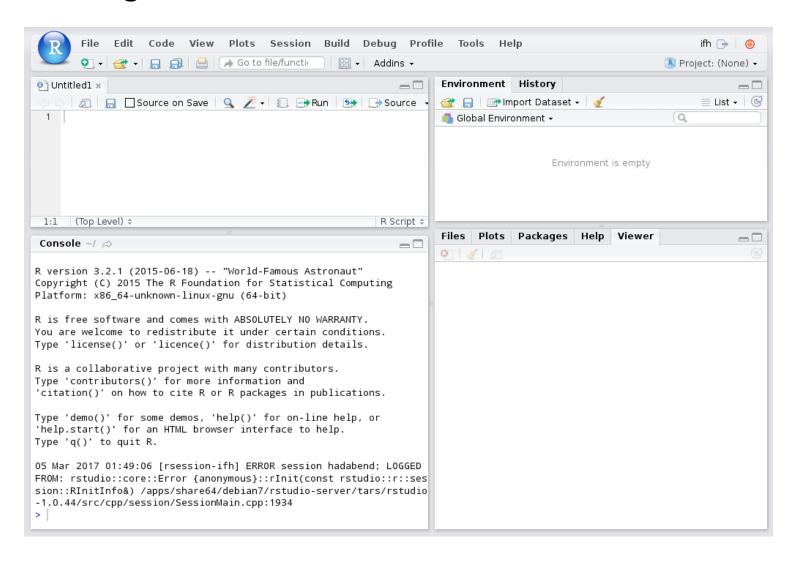
Launching RStudio From HUBzero





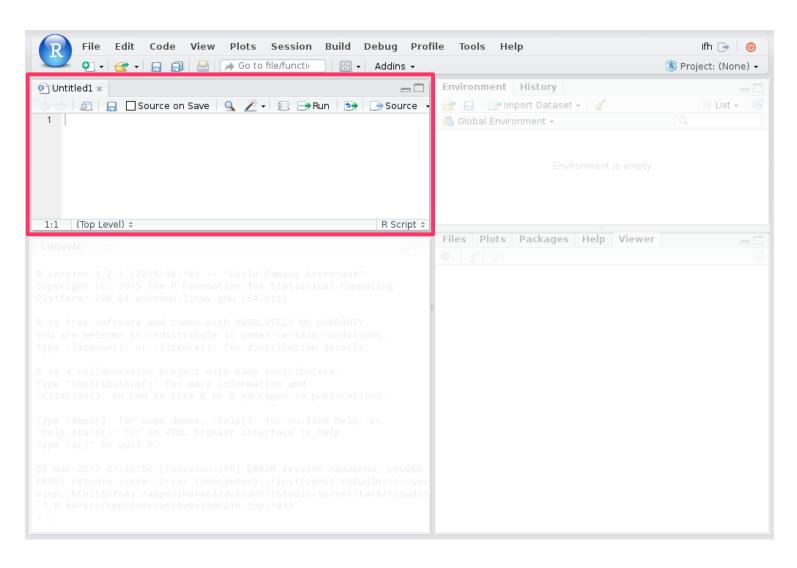
Launching RStudio From HUBzero





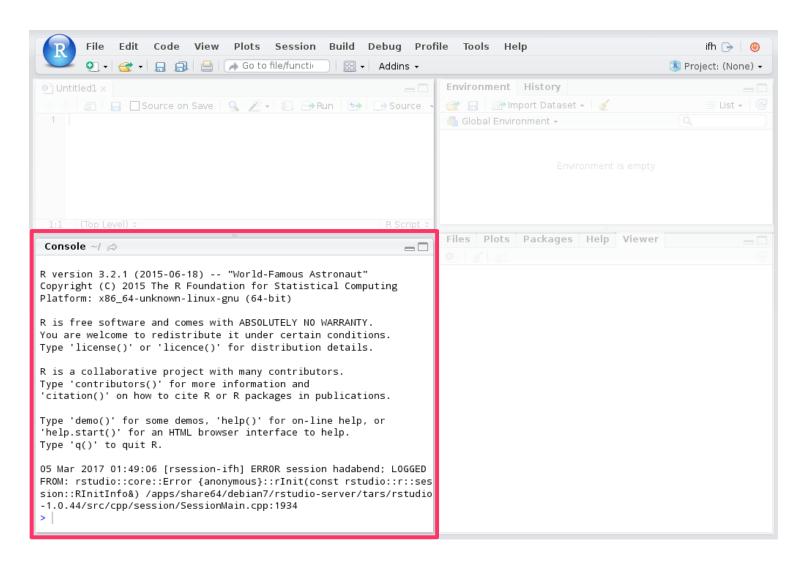
RStudio Interface Tour - Source Pane





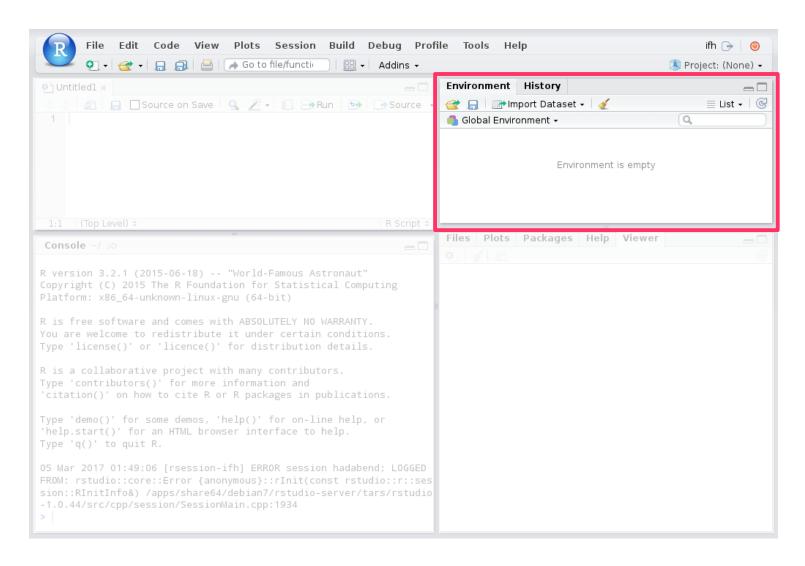
RStudio Interface Tour - Console Pane





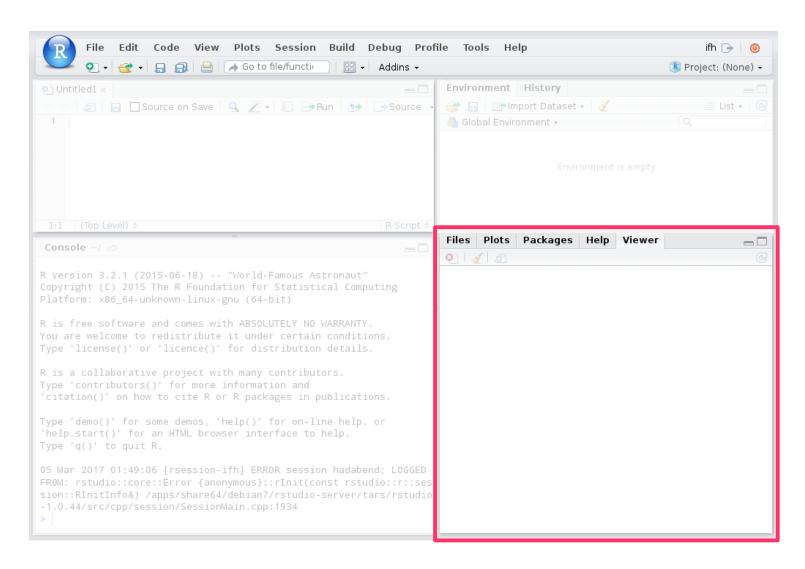


RStudio Interface Tour - Environment Pane



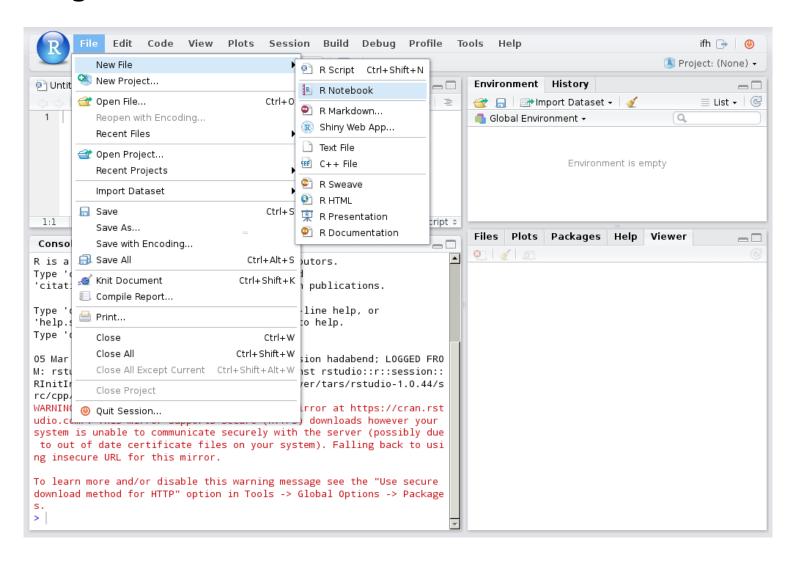
RStudio Interface Tour - Viewer Pane





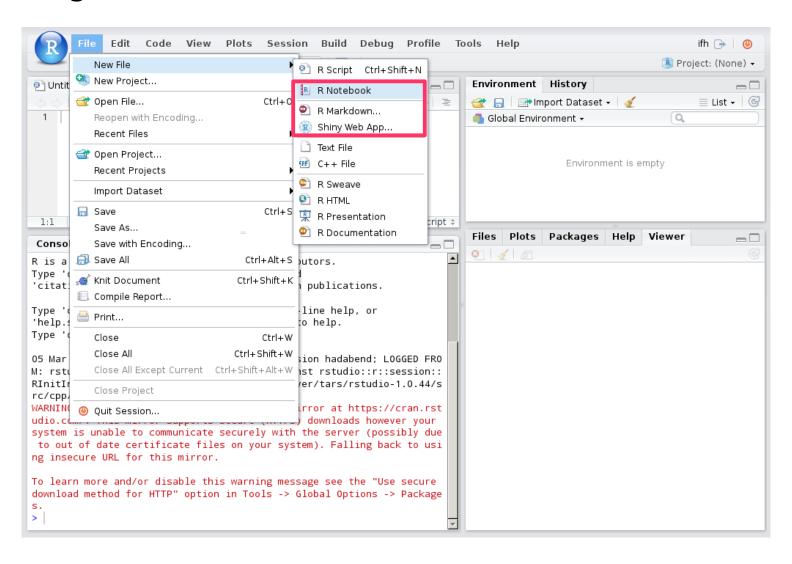
Writing R Code





Writing R Code



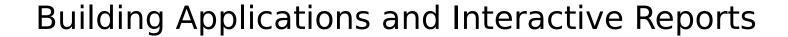


Building Applications and Interactive Reports











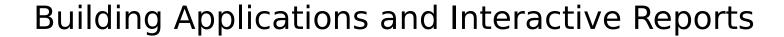






Radio buttons

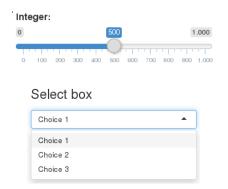
- Choice 1
- O Choice 2
- O Choice 3





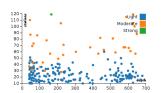


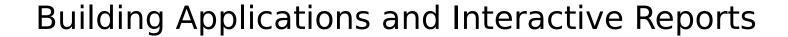








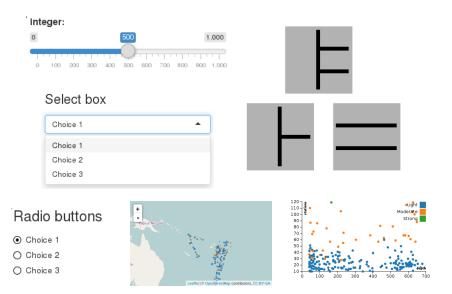










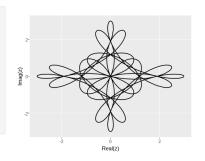


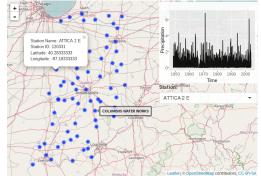
Shiny Example 1 - Spirograph





Spirograph





Spirograph Example

Indiana Precip Example

https://github.com/codedsk/hubzero-tool-spiro-shiny-1

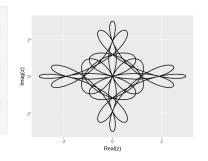
https://github.com/codedsk/rplay/mapping/indiana-precip-shiny

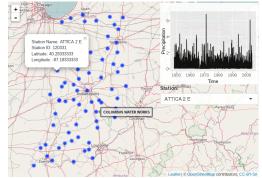






Spirograph





Spirograph Example

Indiana Precip Example

https://github.com/codedsk/hubzero-tool-spiro-shiny-1

https://github.com/codedsk/rplay/mapping/indiana-precip-shiny

Building Applications and Interactive Reports









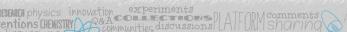


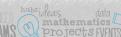
```
{r setup, include=FALSE}
10 knitr::opts_chunk$set(echo = TRUE)
12
14 The spirograph equation for three of more wheels can be generalized
16 \$z(t) = \sum_{k=1}^n a_k e^{i2\pi(n_kt+\theta_k)}
18 This program solves those equations for three wheels, assuming all
    of the $a$ coefficients are 1 and $\theta$ coefficients are 0. Find
    more details online at <a href="http://linuxgazette.net/133/luana.html">http://linuxgazette.net/133/luana.html</a>.
21 · ```{r echo=TRUE}
22 spiro <- function(n1,n2,n3) {
    t <- seq(0,1,length.out=1000)
    z \leftarrow \exp(1i^*2^*pi^*n1^*t) + \exp(1i^*2^*pi^*n2^*t) + \exp(1i^*2^*pi^*n3^*t)
    result <- data frame(x=Re(z),y=Im(z))
     return (result)
27 }
29 result <- spiro(13,-7,-3)
```

RStudio and Shiny are trademarks of RStudio, Inc.



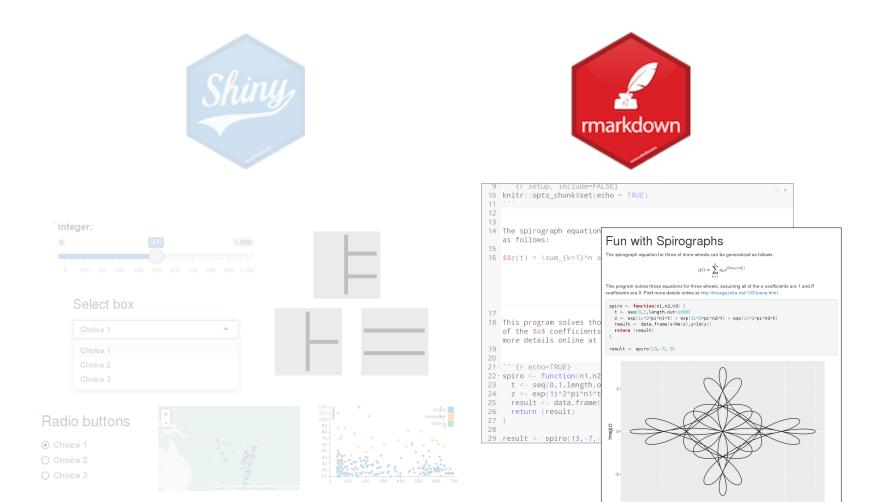
Choice 1





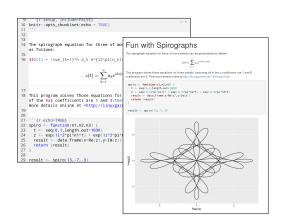
Building Applications and Interactive Reports





Rmd Example 1 – Markup



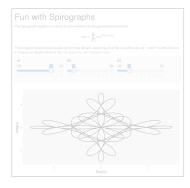


Markup Example





Notebook w/ HTML Widgets



RMarkdown w/ Shiny Widgets

https://github.com/codedsk/rplay

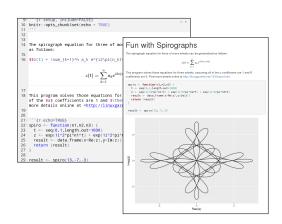






Rmd Example 2 - Notebook w/ HTML Widgets



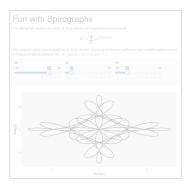


Markup Example





Notebook w/ HTML Widgets

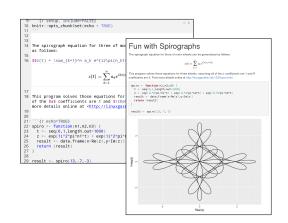


RMarkdown w/ Shiny Widgets

https://github.com/codedsk/rplay

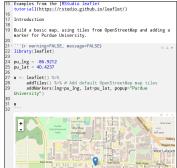
Rmd Example 3 - Rmd w/ Shiny Widgets



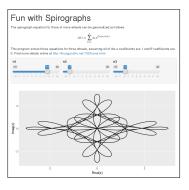


Markup Example





Notebook w/ HTML Widgets



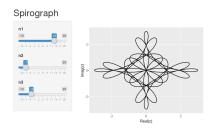
RMarkdown w/ Shiny Widgets

https://github.com/codedsk/rplay









Graphical User Interface



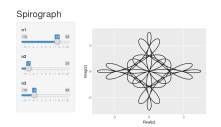
Makefile



Invoke Script







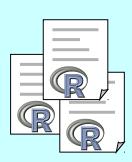




Graphical User Interface

Makefile

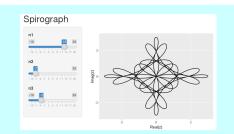
Invoke Script







Source Code



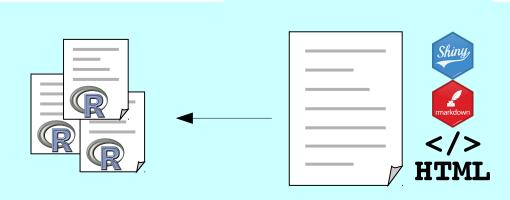
Graphical User Interface



Makefile



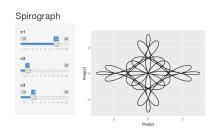
Invoke Script











Graphical User Interface



Makefile



Invoke Script



all:

```
@. letclenviron.sh; \
  use -e -r R-3.2.1; \
  use -e -r rstudio-server-1.0.44; \
  Rscript -e "rmarkdown::render('${RMDFILE}')"
```

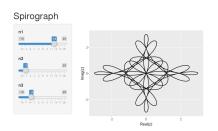
install:

install --mode 0444 -D \${HTMLFILE} \${WWWDIR}/\${HTMLFILE} In -s \${HTMLFILE} \${WWWDIR}/index.html









Graphical User Interface



Makefile



Invoke Script



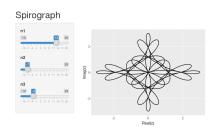
/usr/bin/invoke_app "\$@" -t spirormddoc \

- -u R-3.2.5 \
- -u rstudio-server-1.0.44 \
- -u wrwroxy-0.1 \
- -u wrapprun-0.1 \
- -c "wrwroxy --listenPort 8000 --forwardPort 8001 --stream-log --logfile /dev/null" \
- -C "wrapprun --host 0.0.0.0 --port 8001 @tool/www/index.html"









Graphical User Interface



Makefile



Invoke Script



all:

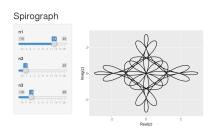
install:

install --mode 0444 -D \${RMDFILE} \${BINDIR}/\${RMDFILE}









Graphical User Interface



Makefile



Invoke Script

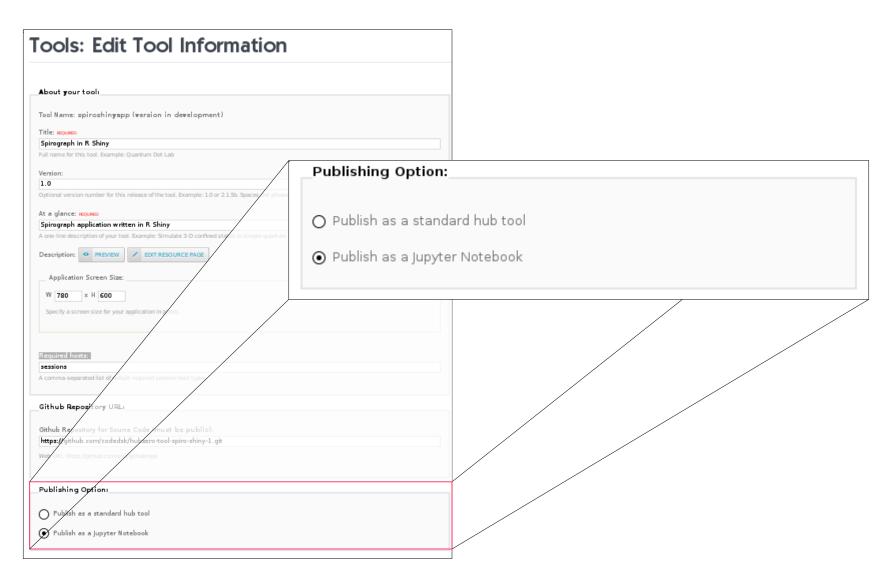


/usr/bin/invoke_app "\$@" -t spirormddoc \

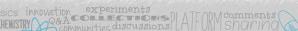
- -u R-3.2.5 \
- -u rstudio-server-1.0.44 \
- -u wrwroxy-0.1 \
- -u wrapprun-0.1 \
- -c "wrwroxy --listenPort 8000 --forwardPort 8001 --stream-log --logfile /dev/null" \
- -C "wrapprun --host 0.0.0.0 --port 8001 @tool/bin/spirograph.Rmd"

Contribtool: View as Notebook





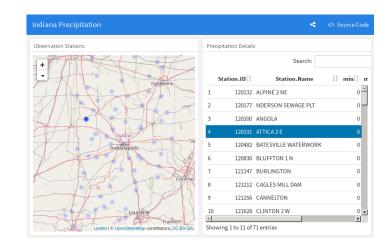








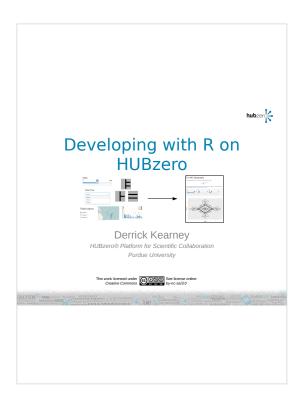








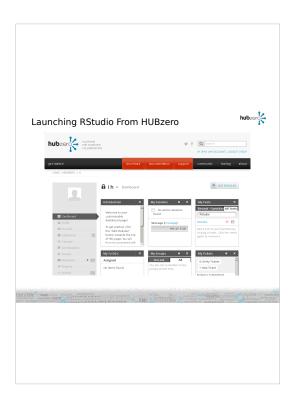




Literate computing has a long history with contributions to the idea reaching from its creator Donald Knuth, to Mathematica, and Matlab. More recently we've seen the introduction of Jupyter Notebooks revolutionize how people explore data with Python and other languages.

Rstudio Inc, a small company focused on the R programming language has built an ecosystem of tools that bring Literate Programming capabilities to the R programming language, while nicely incorporating the reporting and analysis tools that have been a halmark of the R programming language.

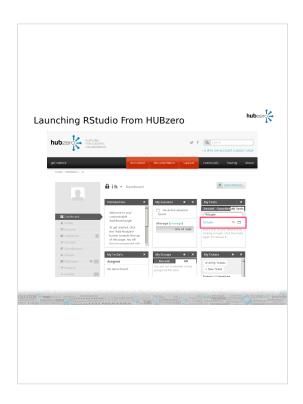
Today we'll get a brief introduction to a few of these tools and see how they can be deployed as resources on the HUB.



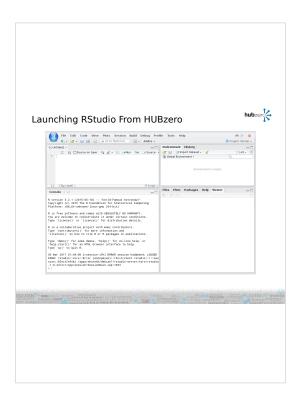
You can try out the Rstudio IDE from hubzero.org if it is not installed on your HUB.

First, navigate to the HUB webpage and login.

After you've logged in, navigate to the Dashboard page.



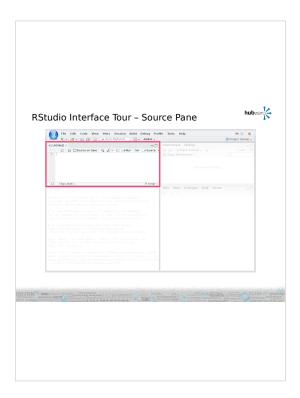
You can search for RStudio under the My Tools module
When you click the launch tool button,



You'll notice this doesn't launch the normal vnc viewer. Instead, the user is served a javascript based user interface.

This is the RStudio IDE.

There are 4 main parts to the IDE



The Code Pane is where users manipulate source code



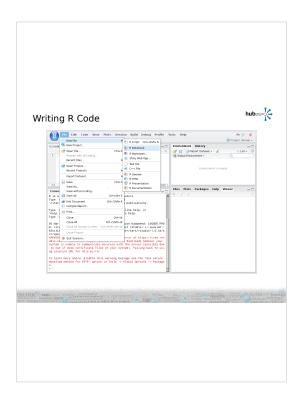
The console is where R commands are executed



The Environment / History pane shows variables and files that have been loaded into the ide environment.



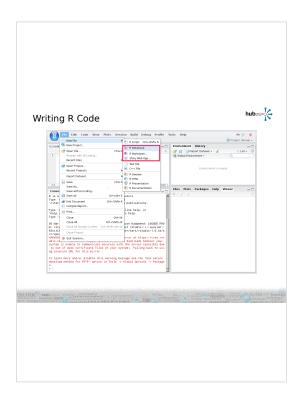
And the viewer pane shows output that can be visualized



To write code, use the File → New menu item to open a new file.

highlight today.

Notice there are a number of different types of files you can open. You would probably expect to be able to open an R file to write a regular program in R, but there are three other types of files I'd like to



Shiny Web Apps, RMarkdown, and the closely related R Notebooks



The R language has a history of allowing people to easily generate reports. People have been using Sweave to combine R code and plots with LaTeX for years.

RStudio, the company behind the graphical user interface, has been focusing on building ways for people to more easily disseminate their research results and analysis, particularly on the web.

Two packages they developed to make this happen include Shiny and RMarkdown.

Shiny is a toolkit that allows users to put together web apps.



It includes things like the normal input widgets you would expect, sliders, radio buttons, dropdown menus.



And output widgets like text boxes and static plots.

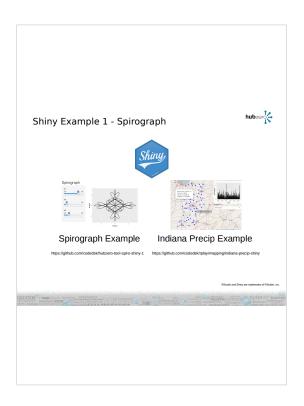
It also includes a number of interactive output widgets backed by JavaScript libraries like leaflet for mapping, Plotly for plots, and data tables.



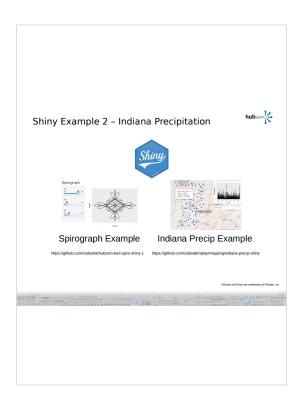
And it allows users to either fully describe their layout using a grid system or one of their predefined templates.

Shiny applications are backed by a shiny server, which produces the HTML, CSS, and JavaScript that the user renders in their web browser.

Let's check out a couple of examples.



https://github.com/codedsk/hubzero-tool-spiro-shiny-1



https://github.com/codedsk/rplay/mapping/indiana-precip-shiny



RMarkdown is a type of markup language, based on the markdown wiki syntax that holds a few additional features including being able to hold and run R code.

It allows us to quickly write text explanations rich with formatting, structure, and code chunks embedded within it. You can also embed the output of a code chunk, like a graph, table, or numeric value.



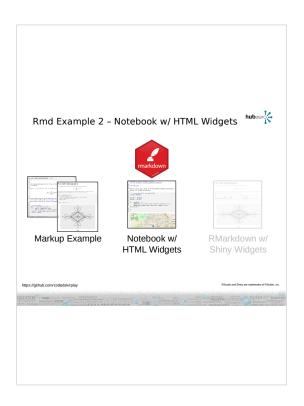
RMarkdown files can be converted to HTML files for distribution as a web page, or as a PDF file, Word document, LaTeX document and a handful of other formats.

Thats one of the strengths of the system.



Let's check out some examples

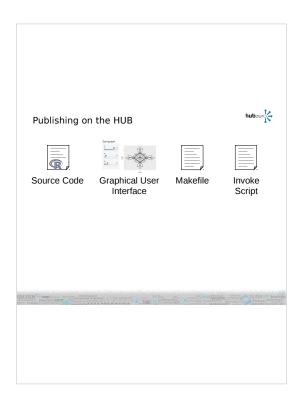
http://github.com/codedsk/rplay/rmarkdown/plain_markup.Rmd



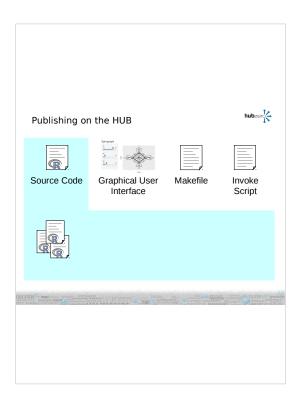
 $http://github.com/codedsk/rplay/mapping/notebook_htmlwidgets_example.Rmd$



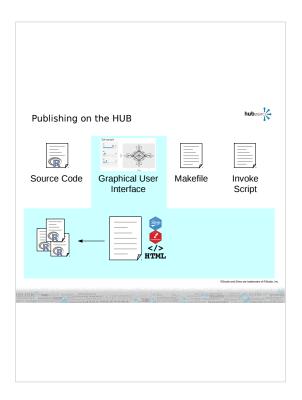
 $http://github.com/codedsk/hubzero-tool-spiro-rmd-html-rshiny-widgets/src/spirograph_with_shiny_widgets.Rmd.\\$



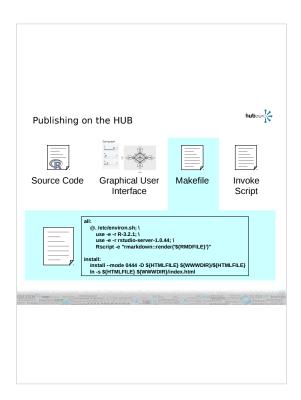
If you are familiar with the tool contribution process on the hub, you know we require 4 things to get your application published. Publishing Shiny and RMarkdown applications is no different.



We need to have the source code in a source code repository

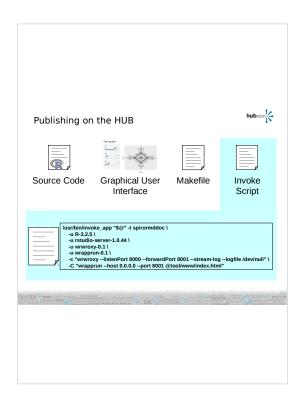


There needs to be a graphical user interface. Since these are web applications and web pages, they will have a HTML and JavaScript based user interface.

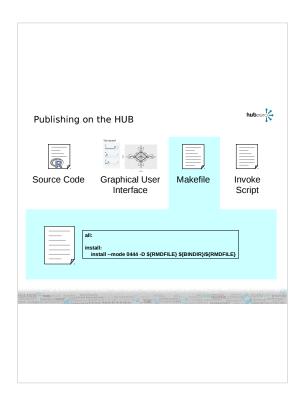


It's always a good idea to include a Makefile that describes how to compile and install the application.

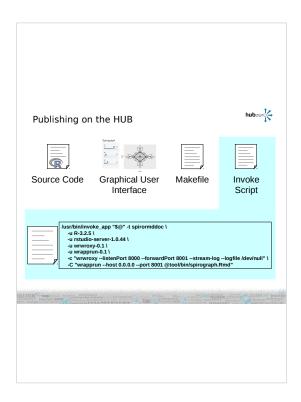
In the case where the application is a static web page, install files in the www directory.



Lastly, the invoke script tells us how to launch the application. For Shiny and RMarkdown applications, we generally launch a rewrite proxy in front of the application. Users can call wraprun to launch their Shiny and RMarkdown application



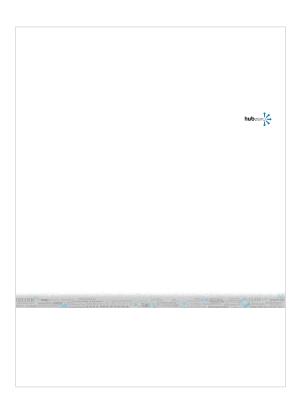
For Shiny and RMarkdown files can be installed in the bin directory or the www directory $% \left(1\right) =\left(1\right) +\left(1\right) +\left$



And the files can be handed to wraprun to launch the application.

Contribtool: V	iew as Notebook	hubzer
About your tools Ted Hance principlingsup (sention is development) This was Springsup in 1 Stary Let week to the late description of the late Marcan	Publishing Option:	
LO 1.0 per service section for the section of the test formulae shade in the section of the sec	Publish as a standard hub tool Publish as a Jupyter Notebook	
Chair to the Control of the Control		
Other in our for the course See of the course of the cour		
The BEEN CARCO SECURITY CONTINUES INTO SECURITY CONTINUES CONTINUE	THE PROPERTY AND ADDRESS OF THE PROPERTY ADDRESS OF TH	notes notes notes Address to technology

In the Contribtool interface, we added a switch that tells the middleware to launch the application as a Weber (or Jupyter Notebook) application and to stream the HTML, JavaScript, and CSS back to the user's web browser, instead of using the VNC based viewer.





https://github.com/codedsk/rplay/mapping/indiana_precip_dashboard_ 1.Rmd

Shiny apps require a server and provide great flexibility to control how widgets react to changes to each other.

The flexdashboard module allows users to layout a dashboard using html widgets.

The crosstalk module allows you to coordinate communication between the widgets so as you filter with say a map widget, an xy plot also sees the changes and reacts accordingly.