What is Raconteur?

• As of today, just a collection of ideas and some code.
What Should Racounter Be?

- A simple procedure to take your models, algorithms, and data and display them on the web.
What Will it Look Like?

The following example[1] demonstrates the flexibility of spower and related functions from Hmisc. We simulate a 2-arm (350 subjects/arm) 5-year follow-up study for which the control group's survival distribution is Weibull with 1-year survival of .98 and 3-year survival of .7. All subjects are followed at least one year, and patients enter the study with linearly increasing probability starting with zero. Assume (1) there is no chance of dropout for the first 8 months, then the probability increases linearly up to .15 at 5 years; (2) there is a linearly increasing chance of dropout up to .3 at 5 years; and (3) the treatment has no effect for the first 9 months, then it has a constant effect (hazard ratio of .72).


Characteristics of control and intervention groups with a lag in the treatment effect and with non-compliance in two directions.
What Does it Do?

- Allow users to enter values that interact with your model, algorithm, data, etc.
- Provide range checking and autocompletion of entered values.
- Update the display with graphics and/or structured text.
- Provide a well designed interface based on good design principles.
How Do I Use It?

● On your local computer, load the Racontour R package.
● Call new_racontuer() with your R script as the argument.
● Call raconteur_start()
● Work within your web browser to design your new Raconteur app.
How Do I Use it (part 2)?

• Within the browser, choose the parts of your R script to run the app.
• Choose the input parameters that are presented to the user.
• Add parameter types, ranges, default values, etc.
• Decide whether your app will update a graphic, text area, or both.
How Do I Use it (part 3)?

• Once you are satisfied with your app, you can deploy it to your organization's Raconteur server.
Challenges

• Robust web app frameworks like Ruby on Rails and Django for Python rely on Object Oriented design principles. Objects are inherent to the language

• Raconteur ought to leverage these frameworks as well, but R's object system needs more time to mature. R 2.12 is as step in the right direction with the introduction of reference classes.
Challenges (part2)

- Develop ralite: A light web server based on rApache to run within desktop R.
- Make Hadley Wickam's sinartra package work with rApache.
- Get more statisticians and developers involved!
Thanks

• Want to work on this project?
• Email me: jeffrey.horner@vanderbilt.edu
• Code can be found here:
• https://github.com/jeffreyhorner/raconteur