R-ICE
A Modular **R** GUI

Hutcha Sriplung, Edward McNeil
Apiradee Lim, Naratip Junsakul, et al.

Prince of Songkla University, THAILAND

R-ICE

- **I**ntegrated **C**omputing **E**nvironment for **R**
  - customizability
  - open environment
  - modularity
  - platform independence
- Created by tcltk package within **R** itself

R-ICE

- The project is sponsored by
  - Thailand Research Fund (TRF)
  - Thai National Health Foundation (TNHF)

Customizability

- Customizability means
  - **R-ICE** can be modified to speak any human language.
  - Users can select some of its components to fit their works and also create new GUIs to do many things more.

Open environment

- Open environment means
  - users can use it, share it, and modify the source code under the basic concepts of the GPL license.
  - It is open for developers to create their own modules and share with others.

Modularity

- Modularity means
  - it comprises a number of modules that may or may not be dependent on the others.
  - it is a plug and play environment.

Platform Independence

- Platform Independence means
  - **R-ICE** modules are, in fact, **R** packages that depend on the tcltk library in **R**.

R-ICE Modules

- **R-ICE** consists of four groups of modules or grains
  - global
  - main
  - associated
  - extended

Global Module

- At the moment there is only one **global module** called

  ice

  - collects additional functions, especially those for data management and summaries,
  - thus, it does not have its own GUI interface.

Main Module

- one **main module** called

  ice.main

  - is the GUI responsible for basic file and object management,
  - and setting some preferences in the **ICE** environment.

Associated Modules

- 6 **associated modules**

  ice.dataman, **ice.summary**, **ice.graph**, **ice.statis**, **ice.commands**, **ice.objects**

  - deal with basic data frame management, object summary, graphics, basic statistics, and other basic functions.

Extended Modules

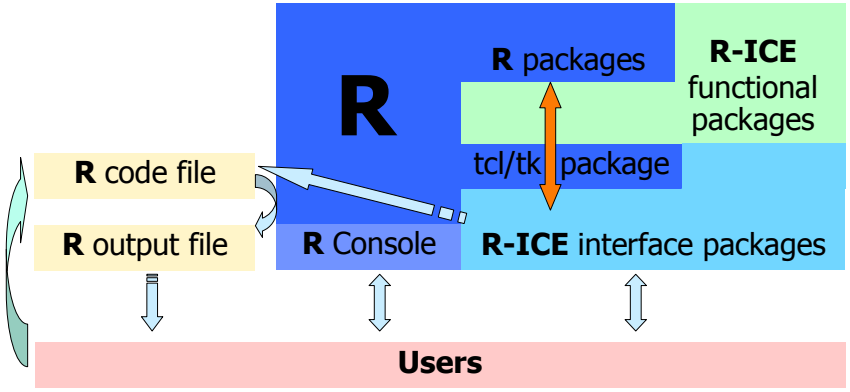
- two **extended modules**

  epid

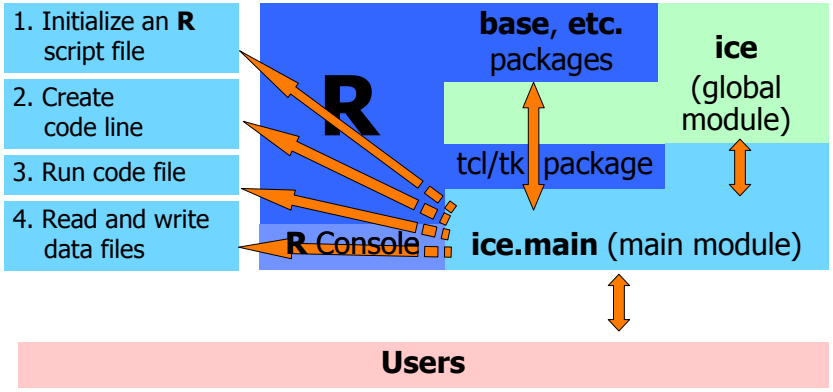
  ice.epid

  - are the plug for creativity.
  - Basically, this plug is designed for developers to encapsulate an existing **R** package with a GUI with the same fashion of menus and dialog boxes.

How **R-ICE** interface modules work



How **ice.main** works



How **ice.main** works

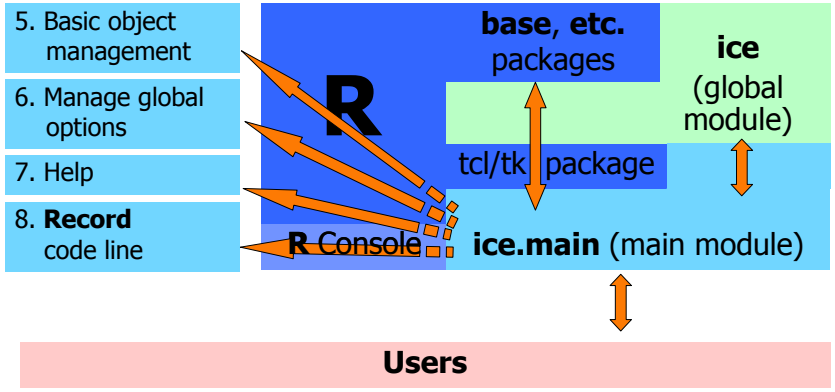
- Initialize or open an **R** script file for all other **R-ICE** modules.
- Create code line for any selection. (Every **R-ICE** interface modules do this job on their own and put the last line in .ec.last.com.line global variable.
- Record the last code line for all **R-ICE** modules.

How to save the last code line(s)

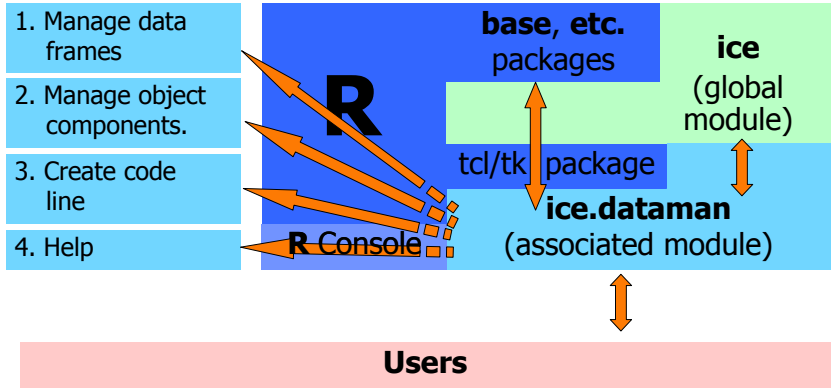
- The following **R** code does the work.

```
com.line <- "your code"
# Use ; to join the lines if there are more than
  1.
# Display the code line on R console.
print(parse(text=com.line)[[1]])
# Execute the code line
# and display the output on R console.
print(eval.parent(parse(text = com.line)[[1]]))
# Save the line in a global variable.
assign(".ec.last.com.line",com.line,
  env=.GlobalEnv)
```

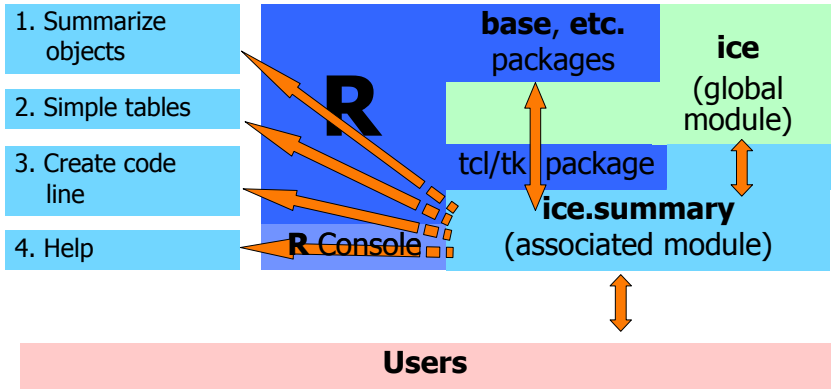
How **ice.main** works



How **ice.dataman** works



How **ice.summary** works



How other **R-ICE** modules work

- They work in the same way as the examples shown previous slides.
- They do a group of similar tasks in one or two drop down menu(s).
- They save the last code line in .ec.last.com.line global variable and let ice.main '**Record**' button to record to the same **R** code file.
- They have their own help or other specific options.

How to use **R-ICE**

- Visit **R-ICE** web site at http://www.r-ice.org.
- Download and install these packages.
  - **ice**
  - **ice.main**
  - **ice.dataman**
  - **ice.summary**
  - **ice.graph**
  - **ice.statis**
  - **ice.objects**
  - **ice.commands**

How to use **R-ICE**

- Create a small source file called 'start.ice' in home directory with these lines.

```
rm(list=ls())
library(ice)
# Call ice libraries
library(ice.main)
library(ice.dataman)
library(ice.summary)
library(ice.graph)
library(ice.statis)
library(ice.objects)
library(ice.commands)

# Open menus and windows
ice.main()
ice.dataman()
ice.summary()
ice.graph()
ice.statis()
ice.objects()
ice.commands()
```

How to use **R-ICE**

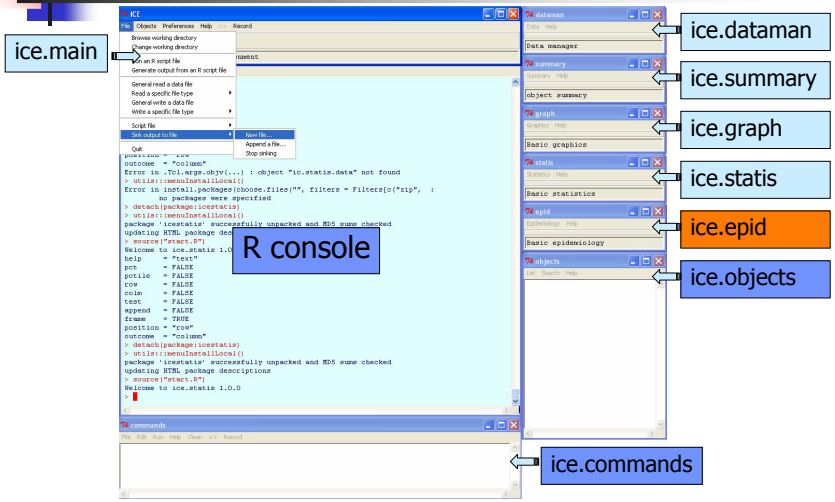
- Then on the **R** console, type at **R** prompt.

```
source("start.ice")
```

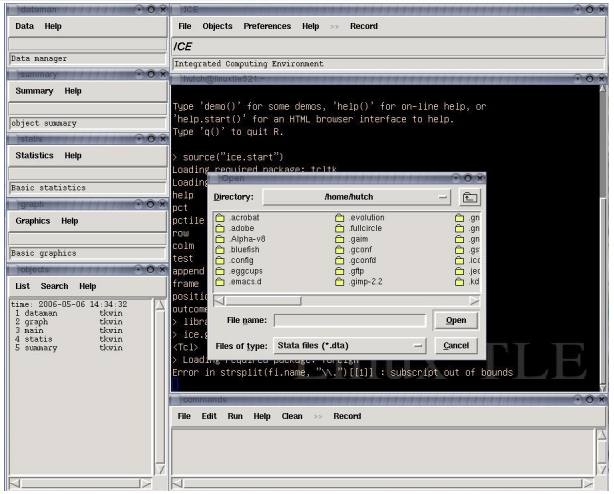
How to use **R-ICE**

- When windows and menus appear, you can move them to the place you feel comfortable using them especially when you are also working with other applications.
- See a few examples of window arrangement.

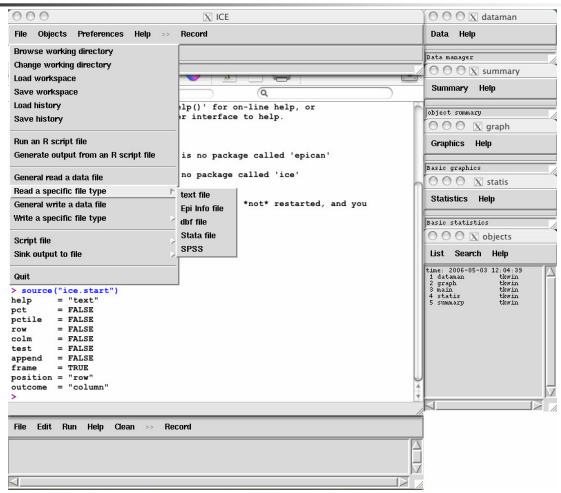
R-ICE on Windows



R-ICE on Linux



R-ICE on Mac OSX



R-ICE speaks your language

- In fact, you can customize **R-ICE** to speak any language that **tcl/tk** on your computer supports.
- This facility can be limited on some platforms. For example, **Thai** can be displayed correctly on **Windows** but not on **Linux** and **Mac OSX**.
- Developers who make a **R-ICE** module in any other language should also supply an **English** version.

R-ICE speaks your language

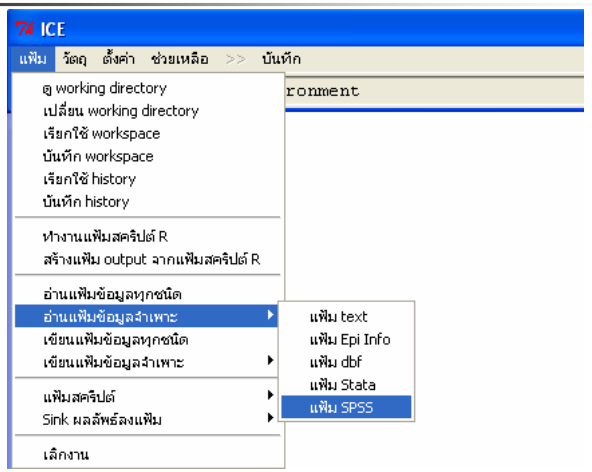
- English

```
OK.but <- tkbutton(dlg, text="    OK    ",
    command=onOK)
Cancel.but <- tkbutton(dlg, text="  Cancel  ",
    command=onCancel)
```

- Thai

```
OK.but <- tkbutton(dlg, text="   ตกลง   ",
  command=onOK)
Cancel.but <- tkbutton(dlg, text="  ยกเลิก  ",
  command=onCancel)
```

R-ICE speaks Thai



R-ICE is open to everyone

- **R-ICE** is open for everyone to join.
  - Customizing menus in any language.
  - Making menus for any existing **R** package.
  - Express your wishes.
  - Give suggestions.
  - And use the modules.
- The official **R-ICE** web site is http://www.r-ice.org

R-ICE Web site