R AnalyticFlow 3
Interactive Data Analysis GUI for R

Ryota Suzuki, Tatsuhiro Nagai
Ef-prime, Inc.
R AnalyticFlow is:

- Data Analysis GUI for R
  - Desktop application written in Java
  - Works on Windows / Mac / Linux
  - Open source (LGPL)
We are:

 Ef-prime, Inc.
  – Data Analysis Consulting Company
  – Tokyo, Japan
  – Working on this project since 2007

 Tatsuhiro Nagai
  – Main Programmer

 Ryota Suzuki
  – Project Manager, UI/UX Design
  – Author of package pvclust
Quick Tour
Projects Dialog

[Diagram of the Projects Dialog window with options for Quick Start, New Project, and Bookmarks.]
View Data
Read Text File

![Image of Read Text File interface]

**Input File (file):** "C:/Users/user/Desktop/iris.txt"

**Output Object:** iris

**Separator (sep):** "\t" (Tab)

**Header (header):** TRUE

**Encoding (fileEncoding):** "ISO8859-1"

**Data Preview:***

<table>
<thead>
<tr>
<th>Sepal.Length</th>
<th>Sepal.Width</th>
<th>Petal.Length</th>
<th>Petal.Width</th>
<th>Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>3.5</td>
<td>1.4</td>
<td>0.2</td>
<td>setosa</td>
</tr>
<tr>
<td>4.9</td>
<td>3.0</td>
<td>1.4</td>
<td>0.2</td>
<td>setosa</td>
</tr>
<tr>
<td>4.7</td>
<td>3.2</td>
<td>1.3</td>
<td>0.2</td>
<td>setosa</td>
</tr>
<tr>
<td>4.6</td>
<td>3.1</td>
<td>1.5</td>
<td>0.2</td>
<td>setosa</td>
</tr>
<tr>
<td>5.0</td>
<td>3.6</td>
<td>1.4</td>
<td>0.2</td>
<td>setosa</td>
</tr>
<tr>
<td>5.4</td>
<td>3.9</td>
<td>1.7</td>
<td>0.4</td>
<td>setosa</td>
</tr>
</tbody>
</table>

**Run**
Run

```r
> iris <- read.table(file = "C:/Users/user/Desktop/iris.txt", header = TRUE,
sep = "\t", fileEncoding = "ISO8859-1", quote = "\"", stringsAsFactors =
FALSE, comment.char = "", na.strings = "")
```
Add to Workflow

![Image of workflow editor]

- **Run**
- **Drag to Add**
- **Close**

**File Format**
- Input File (file): "C:/Users/user/Desktop/iris.txt"
- Output Object: iris
- File Format: Auto Detect
  - Separator (sep): "\t" (Tab)
  - Header (header): TRUE
  - Encoding (fileEncoding): "ISO8859-1"
Select Rows

Select Rows window with criteria set to select rows where Sepal.Length is greater than 5.
Add to Workflow

- Run
- Drag to Add
- Close

**New Flow**

- Run Flow
- Clear and Run
- Run Selection
- Draw Edge

**Read Text File**

- Select Rows

**Main**

- Input: iris
- Output: iris
- Type: Include
- And/Or: And
- Conditions: Sepal.Length > (greater than) 5
Adding Directly

- Read Text File
- Select Rows
- Add/Edit Columns
Modeling

![Image of a software interface for building a predictive model with linear regression and sepal length as the output variable. The model coefficients include intercept, petal length, species versicolor, species virginica, sepal width, petal width, with their respective estimates, standard errors, t-values, and p-values. The output code indicates significance levels with asterisks.]
Prediction

![Image of a software interface for prediction with data tables and model parameters.](image-url)
Plot

The image shows a scatter plot with the following settings:

- **Data**: iris
- **X**: pred
- **Y**: Sepal.Length
- **Conditioning**: Species
- **Grouping**: Species
- **Legends**: Right

The plot is divided into three sections based on the species: **setosa**, **versicolor**, and **virginica**. Each species is represented by different colored markers.

The x-axis represents the predicted values (pred), and the y-axis represents the sepal length (Sepal.Length). The plot visually compares the sepal lengths of different species.
Analysis Workflow

1. iris <- read.table(file = "C:/Users/user/Desktop/iris.txt", header = TRUE, sep = "\", fileEncoding = "ISO8859-1", quote = "\", stringsAsFactors = FALSE, comment.char = "", na.strings = "")
2. iris <- subset(x = iris, subset = Sepal.Length > 5)
3. iris <- local({
   x <- iris
   y <- iris
   y["dummy"] <- with(x, Species == "setosa")
   y
})
4. iris.lm <- step(lm(formula = Sepal.Length ~ 1, data = iris), direction = "forward",
   scope = terms(Sepal.Length ~ ., data = iris))
5. iris$pred <- function(model, data, predType, type, ...){
   (model = iris.lm, data = iris, predType = "prob")
}
6. print(lattice::xyplot(x = Sepal.Length ~ pred | Species, data = iris, auto.key = list( space = "right"), groups = Species))
Run through the Workflow

1. Read Text File
2. Select Rows
3. Add/Edit Columns
4. Build Model
5. Predict
6. X-Y Plot
Run through the Workflow

```r
if (predType == "prob") "raw" else "class"
```
Export to R Script

```r
iris <- read.table(file = "C:/Users/user/Desktop/iris.txt", header = TRUE, sep = "\t", fileEncoding = "ISO8859-1", quote = "\"", stringsAsFactors = FALSE, comment.char = ",", na.strings = ""

iris <- subset(x = iris, subset = Sepal.Length > 5)

iris <- local({

})

iris.lm <- step(lm(formula = Sepal.Length ~ 1, data = iris), direction = "forward", scope = terms(Sepal.Length ~ ., data = iris))

iris$pred <- (function(model, data, predType, type, ...) {
    ...
})(model = iris.lm, data = iris, predType = "prob")
```
Write R Script by Hand

```r
# Some Extra Work
alpha <- 0.05
cf <- as.data.frame(summary(lm(iris))), coefficients, check.names = FALSE)
cf$upper <- with(cf, Estimate + qnorm(1 - alpha/2) * `Std. Error'

cf$lower <- with(cf, Estimate + qnorm(alpha/2) * `Std. Error'

dotchart(cf$Estimate, xlab = extendrange(c(cf$upper, cf$lower)), labels = row.names(cf))
abline(v = 0, lty = 2, col = 8)
segments(x0 = cf$upper, x1 = cf$upper, y0 = 1:nrow(cf) - .1, y1 = 1:nrow(cf) + .1)
segments(x0 = cf$lower, x1 = cf$lower, y0 = 1:nrow(cf) - .1, y1 = 1:nrow(cf) + .1)
segments(x0 = cf$lower, x1 = cf$upper, y0 = 1:nrow(cf), y1 = 1:nrow(cf))
```
Insert R Script Files

```r
> source(file = "script.R", encoding = "windows-1252", echo = FALSE)
```
Write Custom UI
Write Custom UI

Select Rows → Add/Edit Columns → Build Model → Plot Coefficients

- Petal.Width
- Sepal.Width
- Species: Virginica
- Species: Versicolor
- Petal.Length
- (Intercept)
Benefits

- **Designed for Data Analysis**
  - Analyze interactively
  - Practical tools out of the box
  - More focus on data (not programming)

- **Easy to Use**
  - Best for R beginners
  - Great for working with non-statisticians
  - Help sharing your work visually with:
    - Your colleagues
    - Yourself several months later
Compatibility

- Simple Project System
  - Project folder contains:
    - Data, Workflows, etc.
    - Metadata (.afp/.raf hidden-folders)
  - Compatible with other systems
    - You can use the same folder as an RStudio project, or use version control system together (Git, Subversion, etc.)

- Switch between R Script
  - Export workflows as R scripts
  - Insert R scripts into a workflow
Try R AnalyticFlow

- Visit our website:

  http://r.analyticflow.com

  or  🕵️ R AnalyticFlow

  - Latest information including today’s slides are available at:

    http://r.analyticflow.com/news-tips/
Enjoy your analysis with…

http://r.analyticflow.com

Twitter @efprime

Today’s slide: http://r.analyticflow.com/news-tips/