2016.11.26.SAT

データ解析のためのR GUI

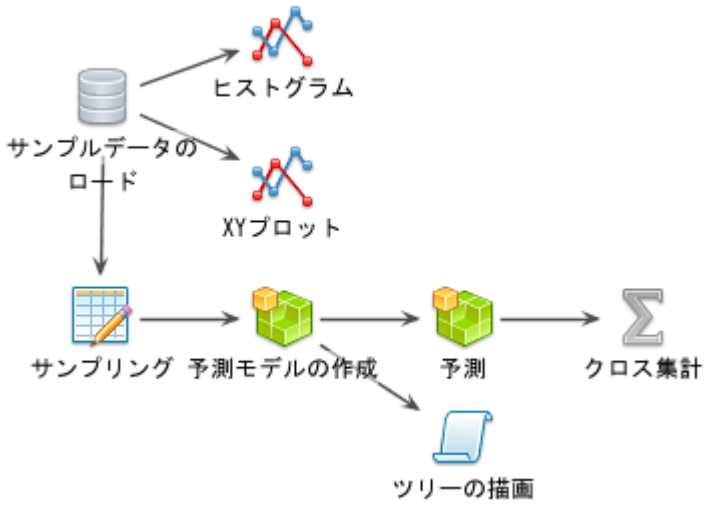
R AnalyticFlowの歩みとこれから

株式会社ef-prime
鈴木 了太

ef-prime, inc.

R AnalyticFlowとは

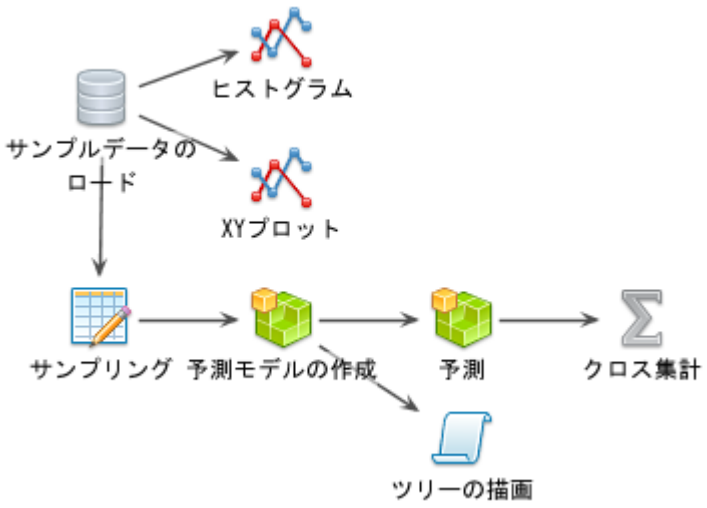
- データ解析のためのR GUI
  - 分析プロセスをワークフローで表現
  - オープンソース
  - Javaで開発、マルチOS対応
    - Windows / Mac / Linux



ef-prime, inc.

R AnalyticFlowの特徴

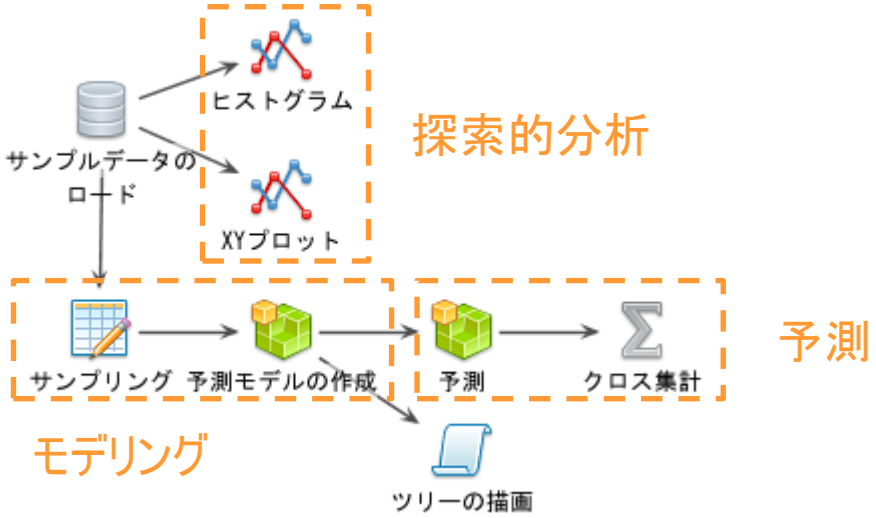
分析フローを作成し、対応するRスクリプトを生成・実行




```r
# 1. データの読み込み
data(iris)

# 2. 探索的分析
plot(iris[, 1:4], col =
as.integer(iris$Species) + 1)
boxplot(Petal.Length ~ Species, data =
iris, col = 3, main = "Petal.Length")

# 3. モデリング
library(rpart)
rp <- rpart(Species ~ ., iris)

# 4. モデルの確認
plot(rp, margin = 0.1, branch = 0.3)
text(rp, fancy = T, all = T, use.n = T)

# 5. 予測および評価
pred <- predict(rp, type = "class")
xtabs(~pred + iris$Species)
```

分析フローの利点

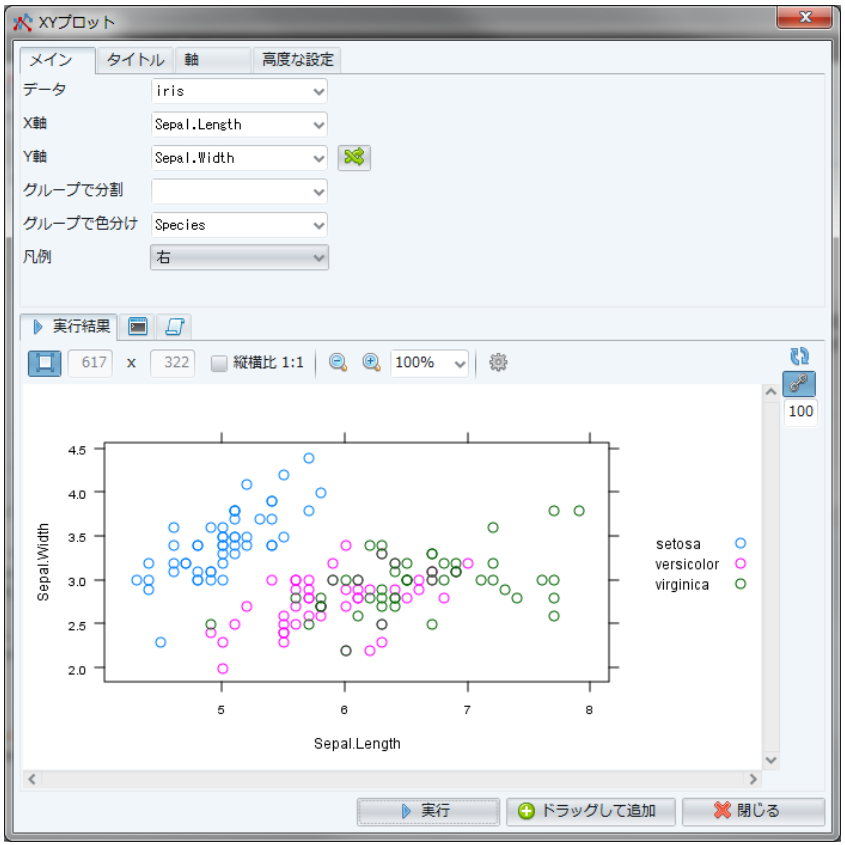
探索的分析からモデリング・予測に至る作業の流れを
視覚的に整理して記述することができる

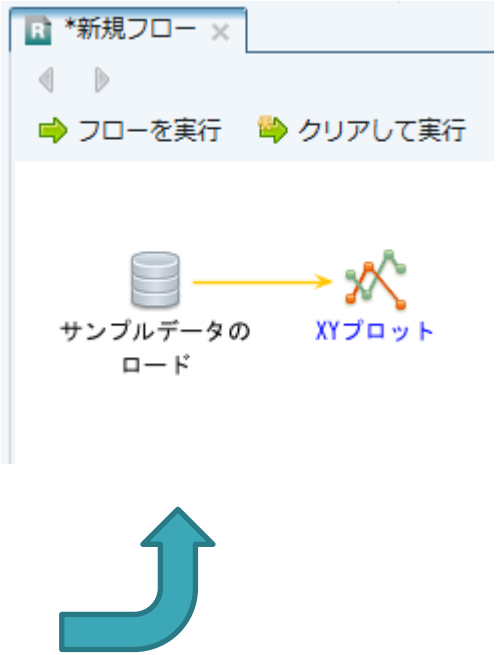


充実した分析GUI

簡単なマウス操作でRスクリプトを記述せずに分析が可能

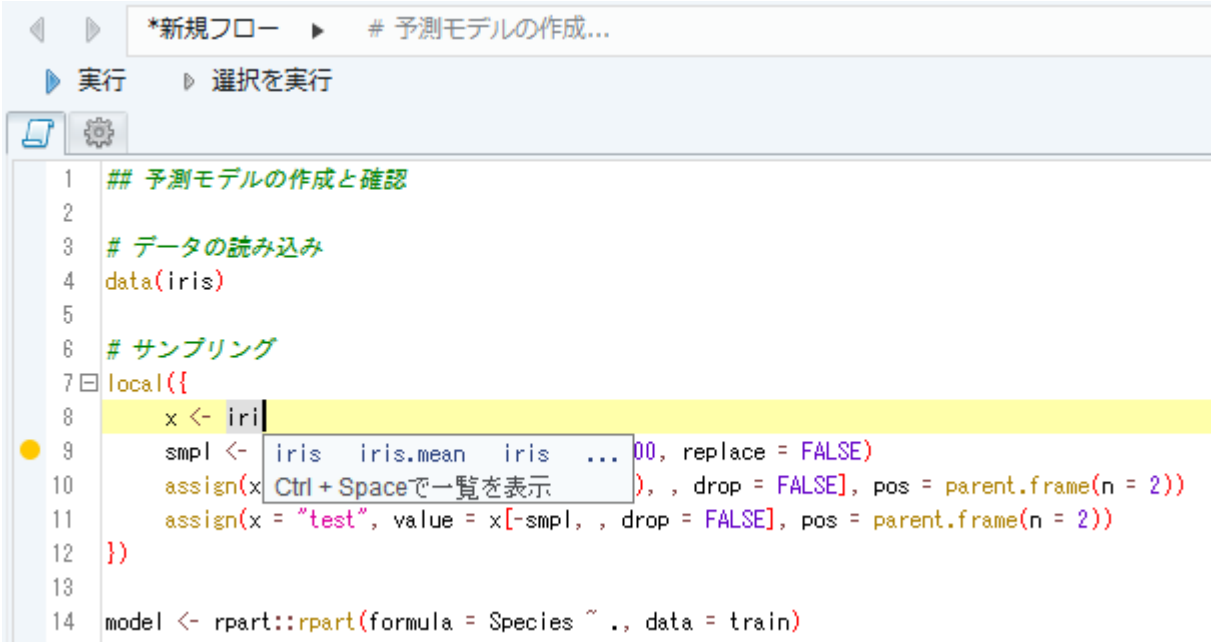
処理をドラッグしてフローに追加





Rプログラミングのサポート

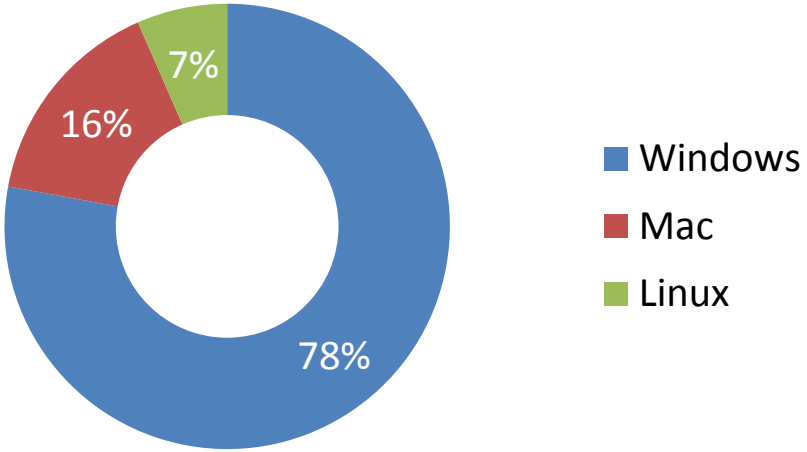
任意のRスクリプトを記述してフローに含めることが可能
ハイライト表示やコード補完、ブレークポイントの作成など



多言語・マルチOS

表示言語として日本語と英語を選択可能
世界80ヶ国でダウンロード（2016年11月時点、直近6ヶ月間）
Windows / Mac / Linux をサポート

1. Japan
2. United States
3. India
4. Taiwan
5. United Kingdom
6. China
7. Germany
8. Italy
9. Spain
10. Australia



これまでの歩み

2007年の開発開始から、約9年間の歩みを振り返ります。

時期	バージョン	詳細
2007年夏ごろ	プロトタイプ	構想・開発着手
2007年12月	0.1.0	プレビュー版公開（Windowsのみ）
2008年3月	0.3.0	ユーザビリティの向上、ファイルエクスプローラ
2009年12月	1.0.0	オブジェクトブラウザ、ボックス、キャッシュ、コードエディタ
2012年12月	2.0.0	タブインターフェースへの統合、デバッグ補助、自動バックアップ、64bit対応
2015年12月	3.0.0	分析機能の完全GUI化、新インターフェース、プロジェクト管理、カスタムUI作成機能

開発の動機

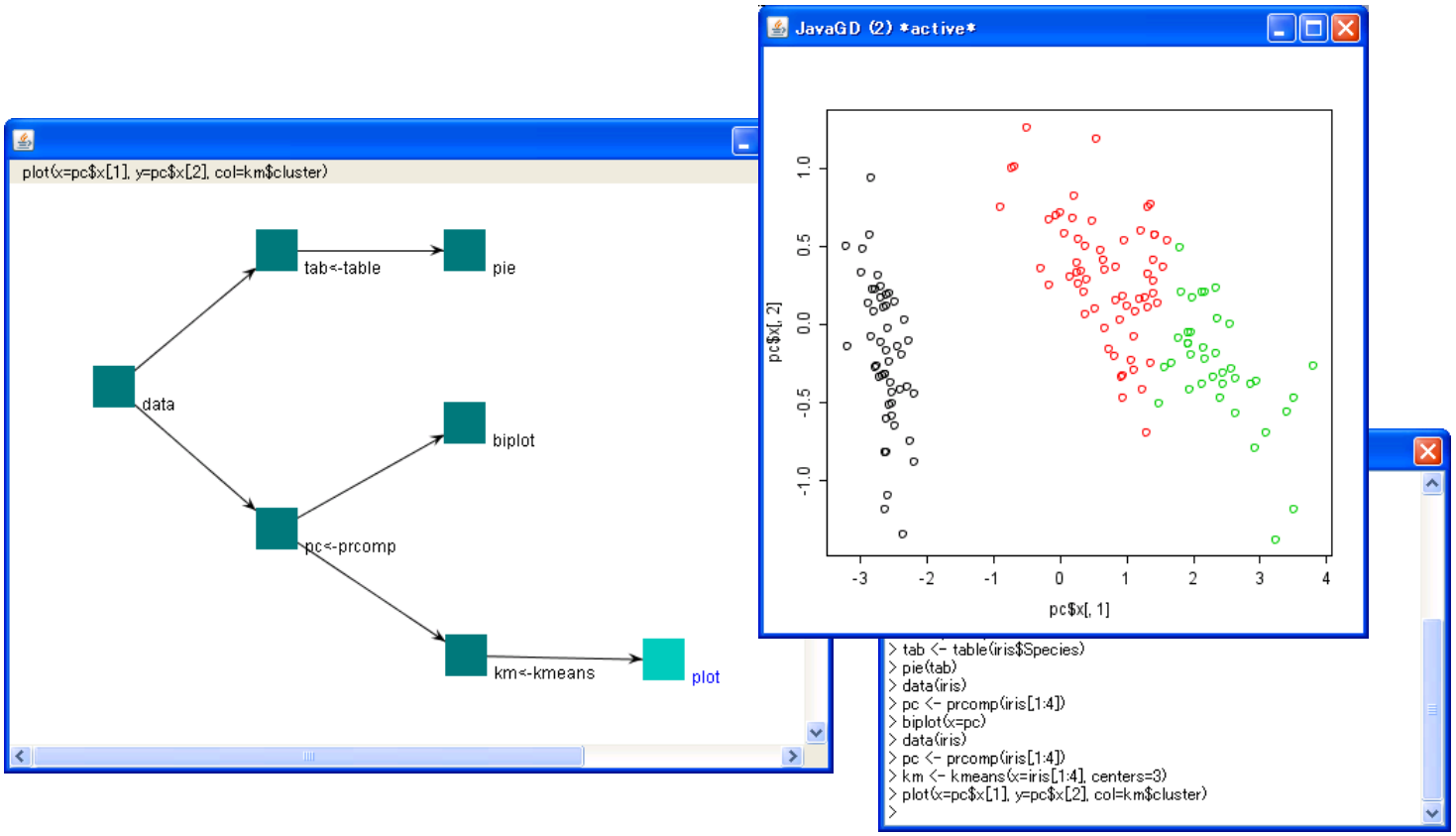
- 背景
  - 2006年3月、株式会社ef-prime創業
    - マーケティングにおける予測分析など
  - 具体的なアウトプットを短期間で作成し、多くの人と共有
    - Rのほか商用ツールや自社開発ツールを併用

- 分析ツールに求められること
  - 高い機能性と使いやすいインターフェース
  - 分析の経過や過程を共有できる
  - 誰でも簡単に結果を再現できる

Rの機能性にビジュアルプログラミングの要素を加えてみよう！

プロトタイプ

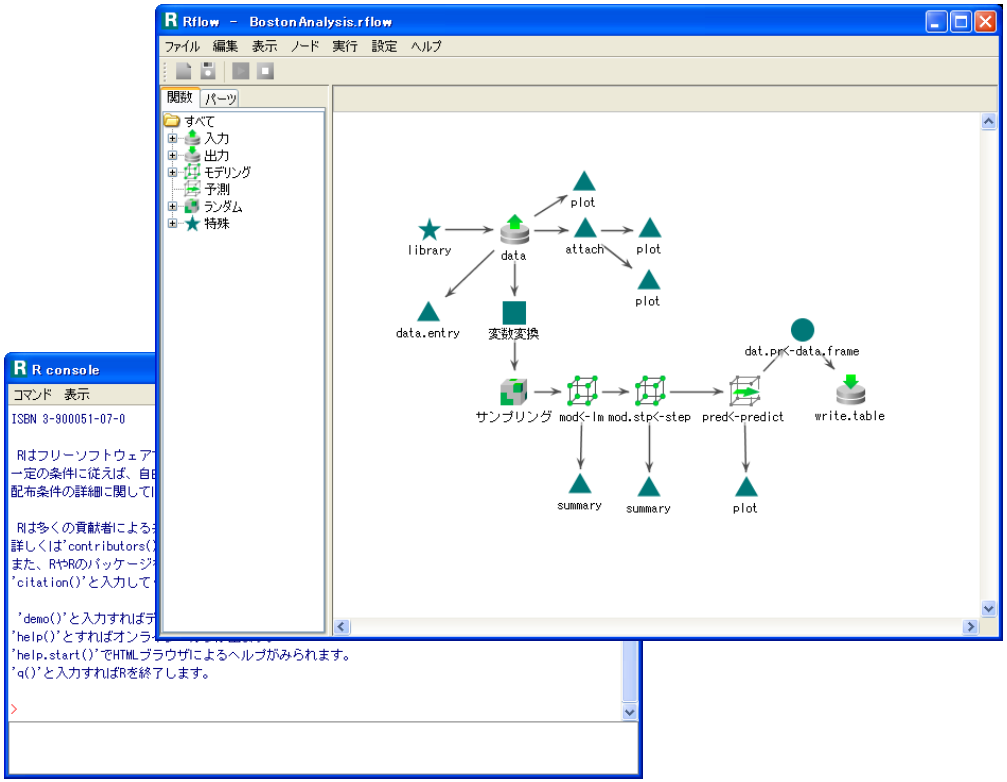
Rでビジュアルプログラミングを実現するツールとして
2007年夏ごろ開発に着手。以下は9月時点でのスクリーンショット

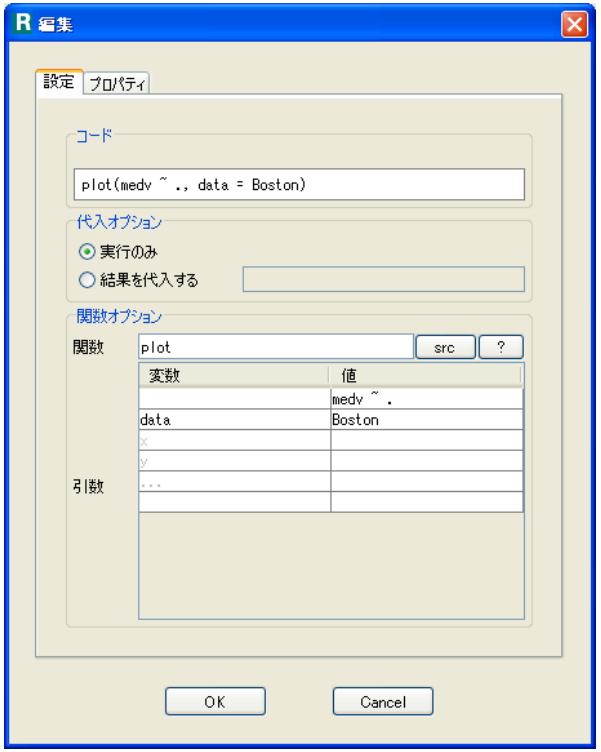


ef-prime, inc.

プレビュー版（0.1.0）

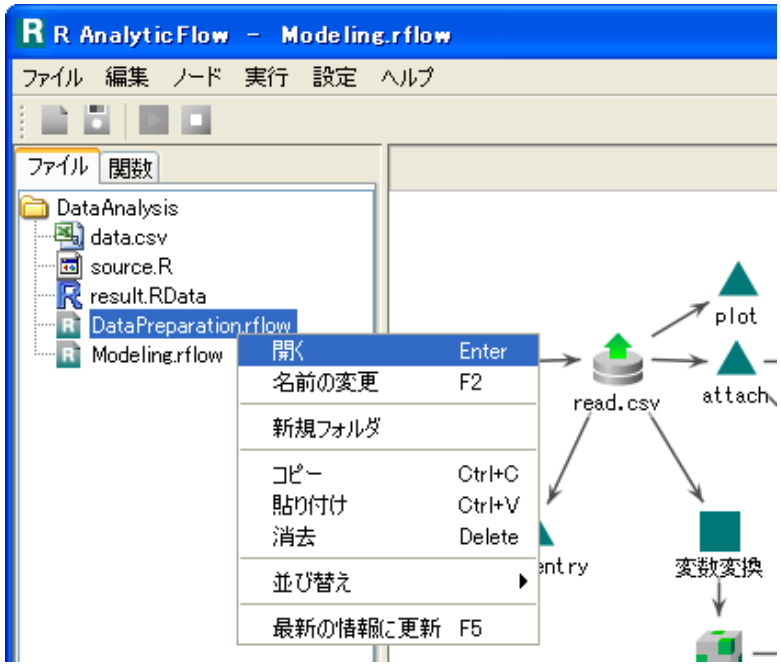
2007年のRユーザー会にて発表、Windows版のみ。
コンソールとフロー画面で構成され、関数名と引数を選んで記述できる
「一行ノード」とスクリプトを直接記述する「自由記述ノード」があった

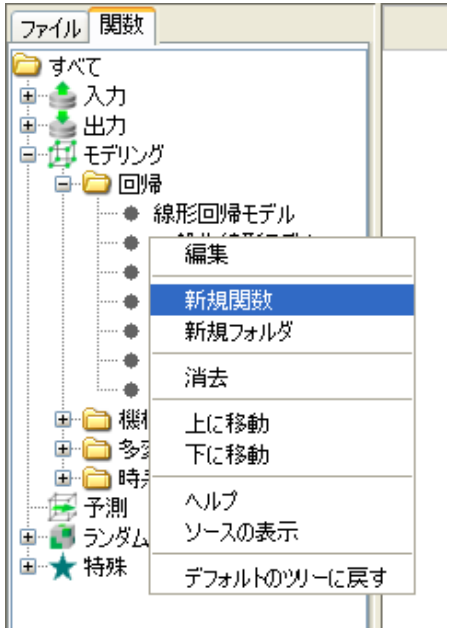




プレビュー版（0.3.0）

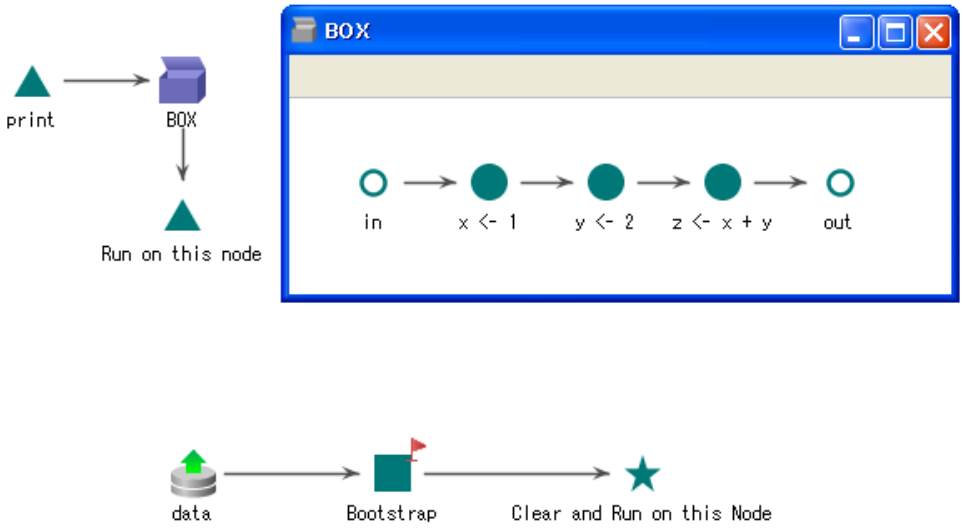
2008年3月公開。Linuxに対応し、日本語と英語が選択可能に。
ファイルエクスプローラが実装されるなど、より実用的になった。
よく使う関数を登録する機能が追加された

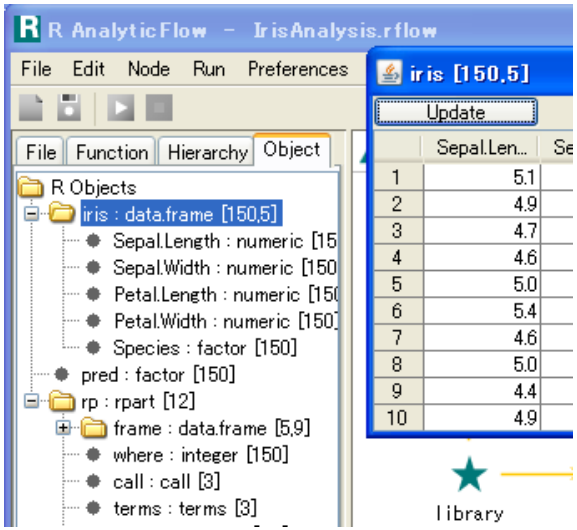




1.0.0

2009年12月公開。Macにも対応。
フローの一部をまとめるボックス、実行結果を保存するキャッシュ、
オブジェクトブラウザなどが実装された

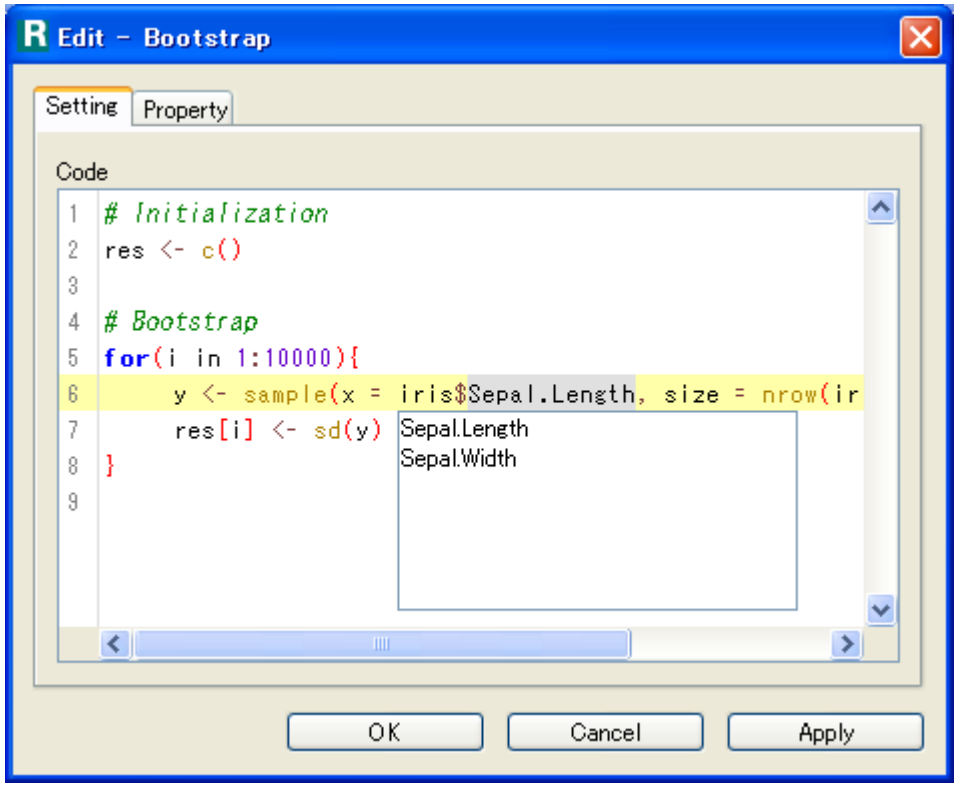




1.0.0

当時はまだ決定版といえるRエディタがなく、ESSやテキストエディタの代わりとしてR AnalyticFlowが使われるケースもあったようです。

（RStudioの一般リリースは約1年後の2011年2月）



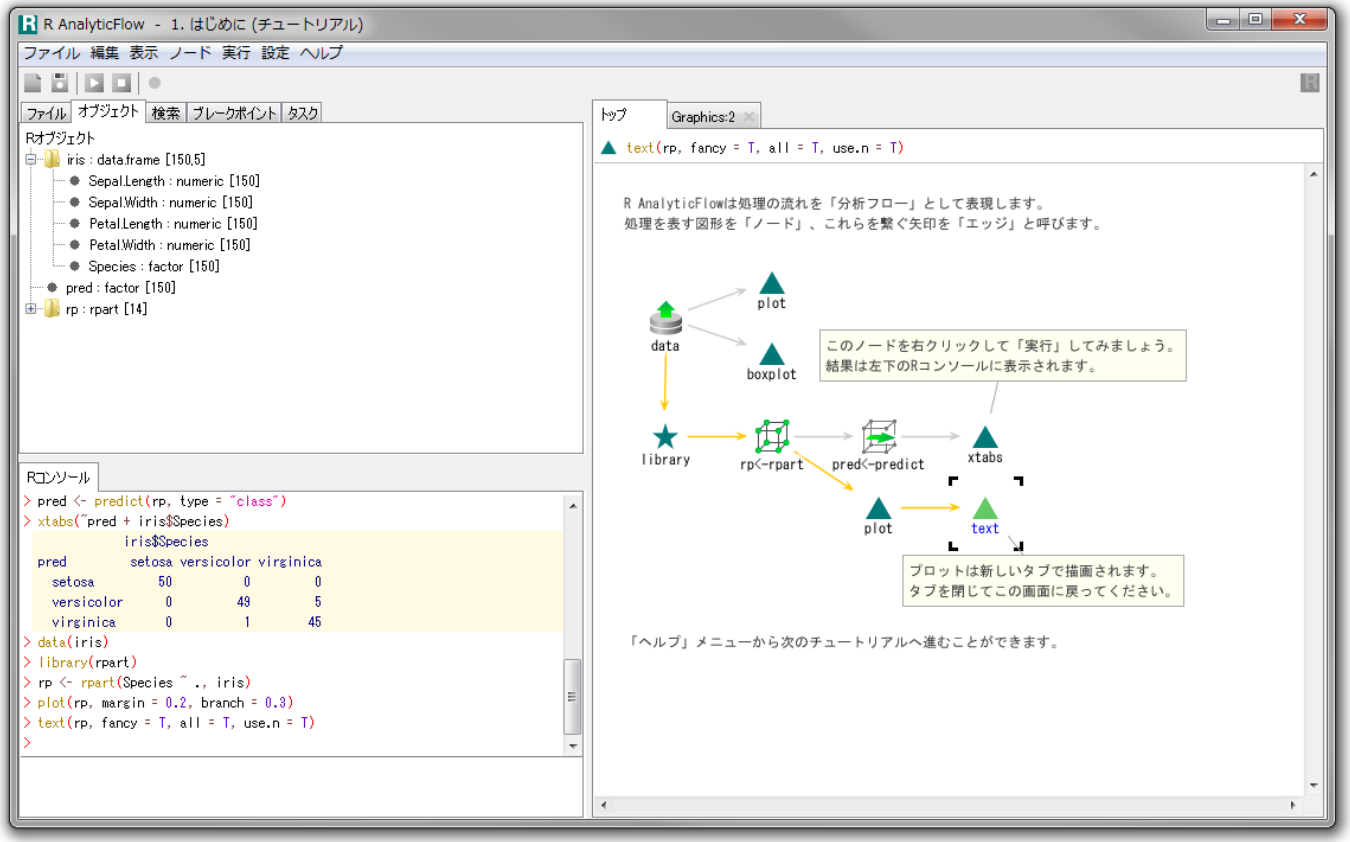
0.x～1.x系列のまとめ

- 特徴
  - Rスクリプトを含む「ノード」を繋いで実行する、というシンプルな発想が出発点
  - 分析プロセスであるRスクリプトだけではなく、入出力にあたるファイルやオブジェクトも一元管理
  - フローをまとめるボックス、不要な再実行を省略するキャッシュなどでより実用的に

- 次バージョンへの展開
  - 表示すべき情報が増え、マルチウィンドウ形式のインターフェースでは表示が煩雑に
  - ほとんどが「自由記述ノード」になってしまい、一行ノードの意味が希薄に

2.0.0

2012年12月公開。マルチウィンドウからタブインターフェースに移行。
64bit対応やデバッグ機能、自動バックアップ機能などでより実用的に



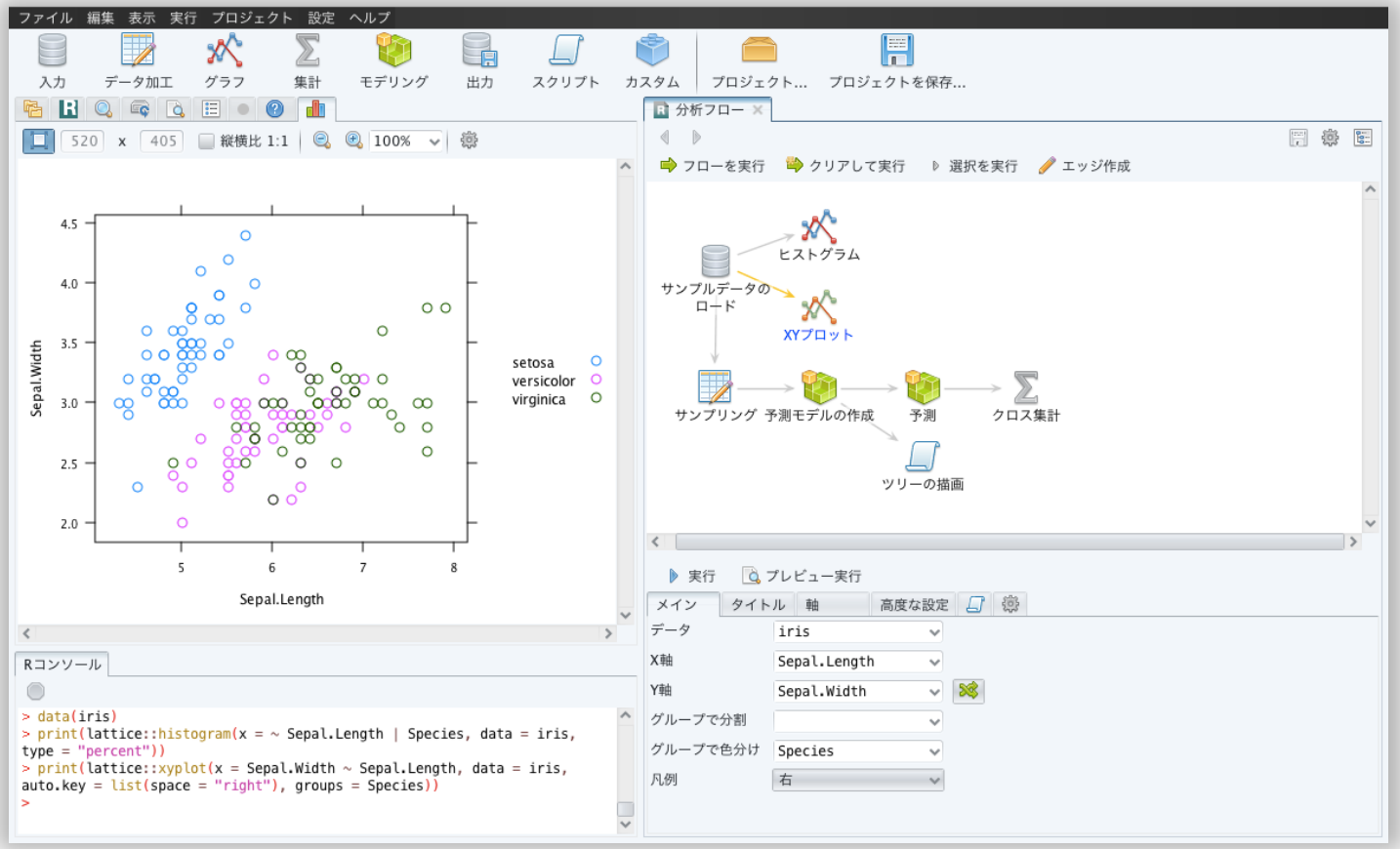
2.x系列のまとめ

- 特徴
  - タブインターフェースに統合したことでより使いやすく
  - 一行ノードを廃止し、すべてのノードがスクリプト
    - 引数やヘルプの表示はエディタの補完機能などで実施
    - 1つの関数呼び出しのみのスクリプトであれば
      対応する機能を表すアイコンでわかりやすく表示

- 次バージョンへの展開
  - OSによってUIが異なり、タブ表示が使いにくいケースがある
  - 複雑なスクリプトが記述されることが増え、共有が困難に
  - 「Rがわかる分析担当者」と「結果を利用するユーザー」
    を想定した設計になっている
    - もっとユーザーフレンドリーにできないだろうか？

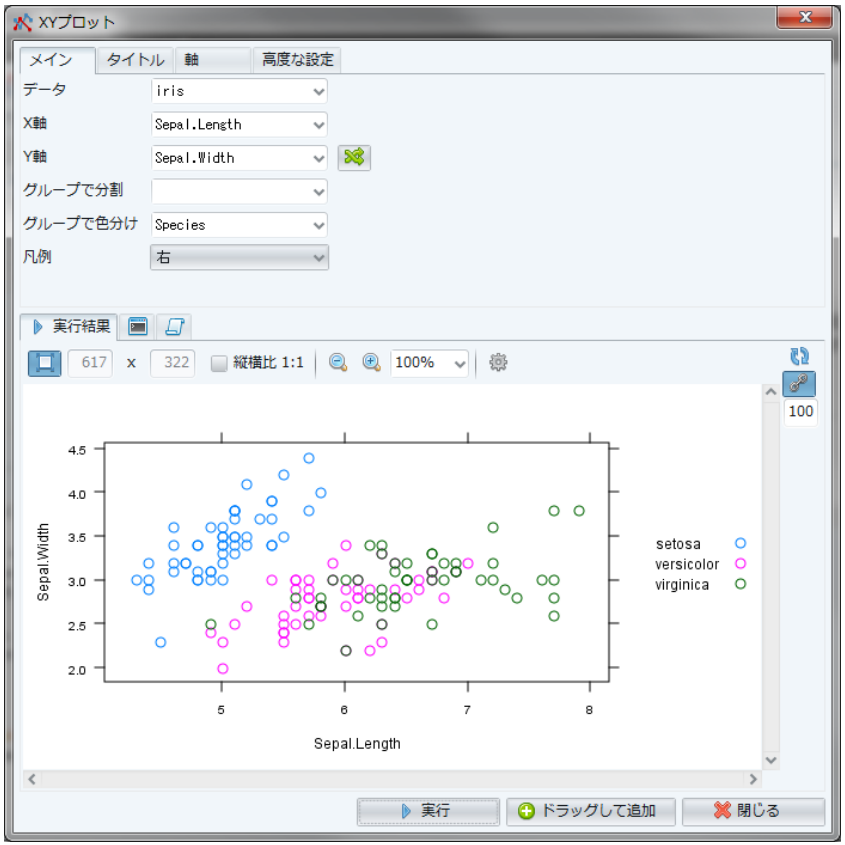
3.0.0

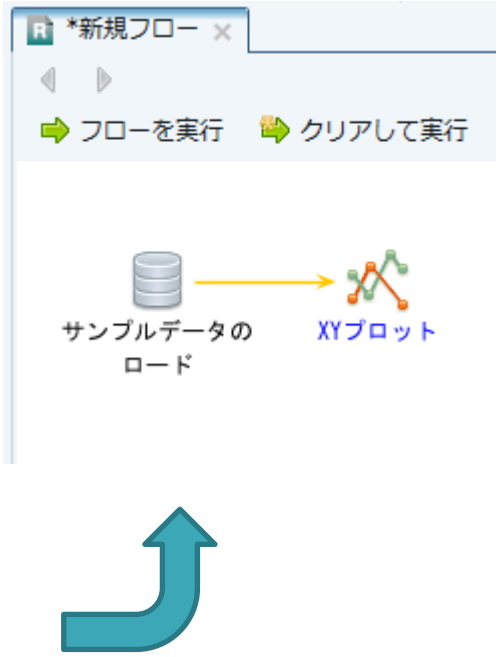
2015年12月公開。グラフィカルな分析インターフェースを実装し、
Rスクリプトを記述することなく分析が可能に。



3.0.0

メニューから分析モジュールを選んで設定を入力。
結果のプレビューが表示され、クリックで実行。ドラッグしてフローに追加。





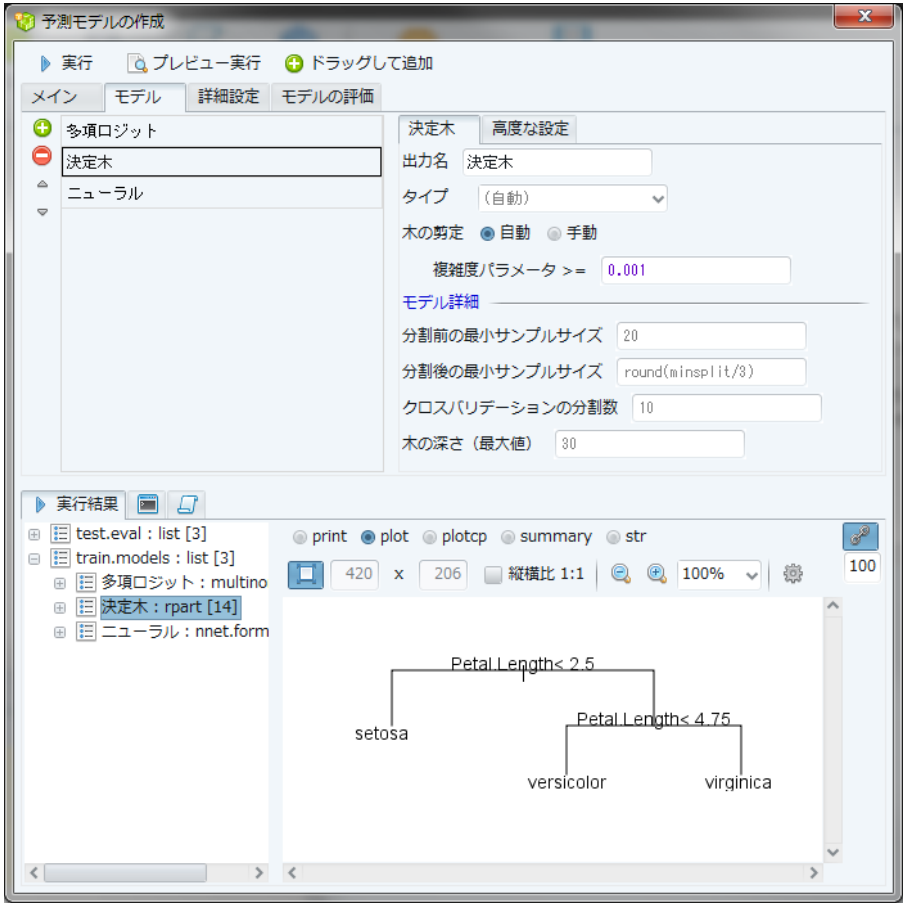
3.0.x系列のまとめ

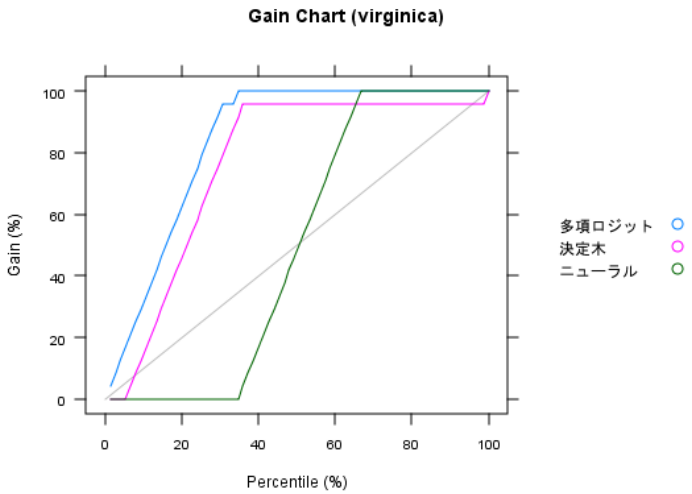
- 特徴
  - 多くの処理をGUIのみで実現することで、
    Rの知識を問わず分析が可能に
  - プレビューにより結果が見えるインターフェース、
    プロジェクト管理機能によりデータとプロセスを統合

- 次バージョンへの展開
  - 実際のデータ分析で必要とされる機能を強化
    - 予測分析、多変量解析、仮説検定など
  - 安定性と利便性の向上

3.1.0

近日公開予定。安定性の向上に加え、分析機能を大幅に強化。
予測モデルの選択などより実践的な応用がGUIのみで実現可能に。



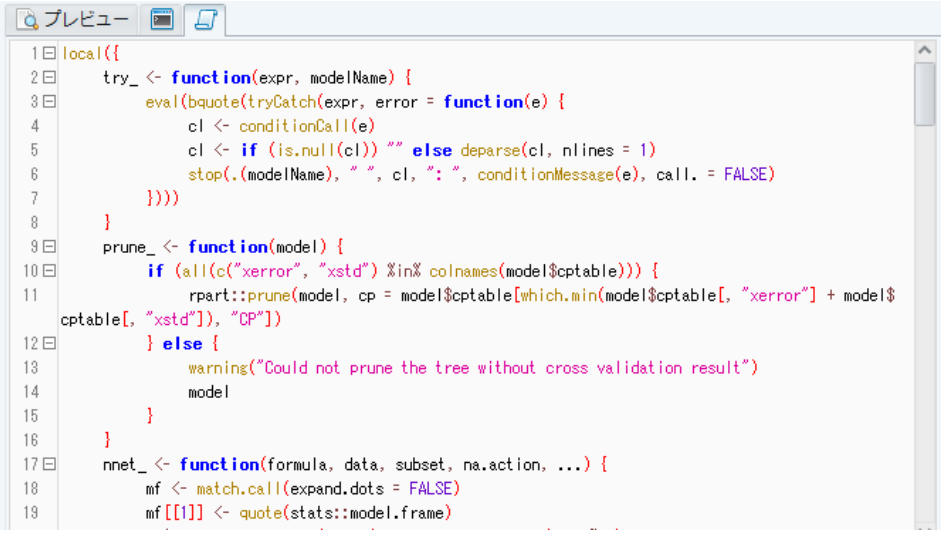


3.1.0の特徴（予定）

- 分析機能の追加
  - データ型の設定、欠損値の処理
  - 予測分析（複数モデルの作成、モデル選択）
  - 仮説検定（t検定、Wilcoxon検定、比率の検定）
  - 多変量解析（主成分、クラスター）
- 安定性の向上
  - Rプロセスを分離し、道連れのクラッシュを防止
- 利便性の向上
  - プロジェクトのインポート・エクスポート
  - 外部Rスクリプトファイルをフローに組み込み

3.2に向けての構想（一例）

- パッケージの導入
  - （3.1まで）外部パッケージに依存しない設計
    - rJavaなどRとの通信に必要なパッケージはあるが、生成されるコードは標準のRで実行可能
  - コードの煩雑化が問題化
    - 実用的でかつ読みやすいコードを生成するには専用関数の利用、ひいてはパッケージの導入が効果的



3.2に向けての構想（一例）

- パッケージで実現できること
  - 独自関数のパッケージ化
    - R AnalyticFlowのインターフェースに合わせた関数群
    - より読みやすいコードを生成し、RScript等からも利用可能に
  - 外部パッケージの利用
    - データ読み込みにはreadr、ランダムフォレストにrangerなど
      高速で有用な外部パッケージを利用しやすい仕組みを構築

- 課題：ポータビリティの担保
  - 生成したスクリプトを外部で実行する際に
    パッケージ不足によるエラーを防ぐための仕組みが必要

ef-prime, inc.

ご清聴ありがとうございました

http://r.analyticflow.com

@ef-prime_jp　R AnalyticFlow

ef-prime, inc.