



初心者セッション1

- RとデータのInput/Output-

Tokyo.R #76

@kotatyamtema

自己紹介



TwitterID : @kotatyamtema

大学での専門: 行動生態学

立案~捕獲~実験~分析までワンオペ

今まで : データ分析会社で客先ばっち分析屋

現在 : 外資系アパレル会社でデータ分析

R歴 : もうすぐ10年?

近況 : 黒魔術的SQLしか書いてない…

絶賛転職活動中！！

求む！使えるデータのある会社！

目的

Rとは何か

Rの環境設定

分析に必要なデータの入出力が
できるようにする

目次

1. Rとは何か
2. Rを使う環境設定とpackageによる機能拡張
3. 外部データのinput方法の紹介
4. 結果のoutput方法の紹介

MENU

1. Rとは何か
2. Rを使うための環境設定
 - RとRStudioのインストール
3. Rの機能拡張
 - Packageの取り込み
 - Packageの呼び出し
4. 外部データのInput方法
 - table形式
 - その他
5. 結果のoutput方法
 - dataframe形式

MENU

1. Rとは何か
2. Rを使うための環境設定
 - RとRStudioのインストール
3. Rの機能拡張
 - Packageの取り込み
 - Packageの呼び出し
4. 外部データのInput方法
 - table形式
 - その他
5. 結果のoutput方法
 - dataframe形式

Rとは何か

統計学のために特化したプログラム 及び開発環境

- データの読み込み
- 前処理
- 基礎集計
- 統計解析
- 可視化

これらが簡単にできるように設計

MENU

1. Rとは何か
2. Rを使うための環境設定
 - RとRStudioのインストール
3. Rの機能拡張
 - Packageの取り込み
 - Packageの呼び出し
4. 外部データのInput方法
 - table形式
 - その他
5. 結果のoutput方法
 - dataframe形式

Rを使うための環境設定

R本体のインストール

CRAN(Comprehensive R Archive Network)

- ・ CRANとは：

公式の本体及びパッケージ配信用Webサイト

Mac環境の場合はhomebrewでもインストール可

Rを使うための環境設定

おすすめIDE(統合開発環境)はRStudio

RStudio

Mac環境の場合はhomebrew caskでもインストール可

RStudioとは：

その名もRStudio社が提供するRに特化した
開発プラットフォーム

RmarkdownやShiny、DWHとの連携がスムーズ
コーディング中にサジェストや補完機能が充実

MENU

1. Rとは何か
2. Rを使うための環境設定
 - RとRStudioのインストール
3. Rの機能拡張
 - Packageの取り込み
 - Packageの呼び出し
4. 外部データのInput方法
 - table形式
 - その他
5. 結果のoutput方法
 - dataframe形式

Rの機能拡張

“package”を追加することで簡単に機能拡張ができる

- 世界中で開発されていて便利な手法がすぐ使える
- デモ実行のためのサンプルデータも使える

R公式のpackage集

- CRAN
 - <https://cran.ism.ac.jp/>

githubに公開されている個人作成package

- こちらは自己責任で

packageの追加

packageの追加方法

1. RStudioのPackagesタブを使う
2. install.package関数を自分で書く

今回の範囲

- ・オンラインでCRANからインストール
 - * オフラインでインストールする場合は下記参照
<http://datasciesotist.hatenablog.jp/entry/2015/03/24/004539>
- ・ miniCRANパッケージが必要なので最初はオンラインで

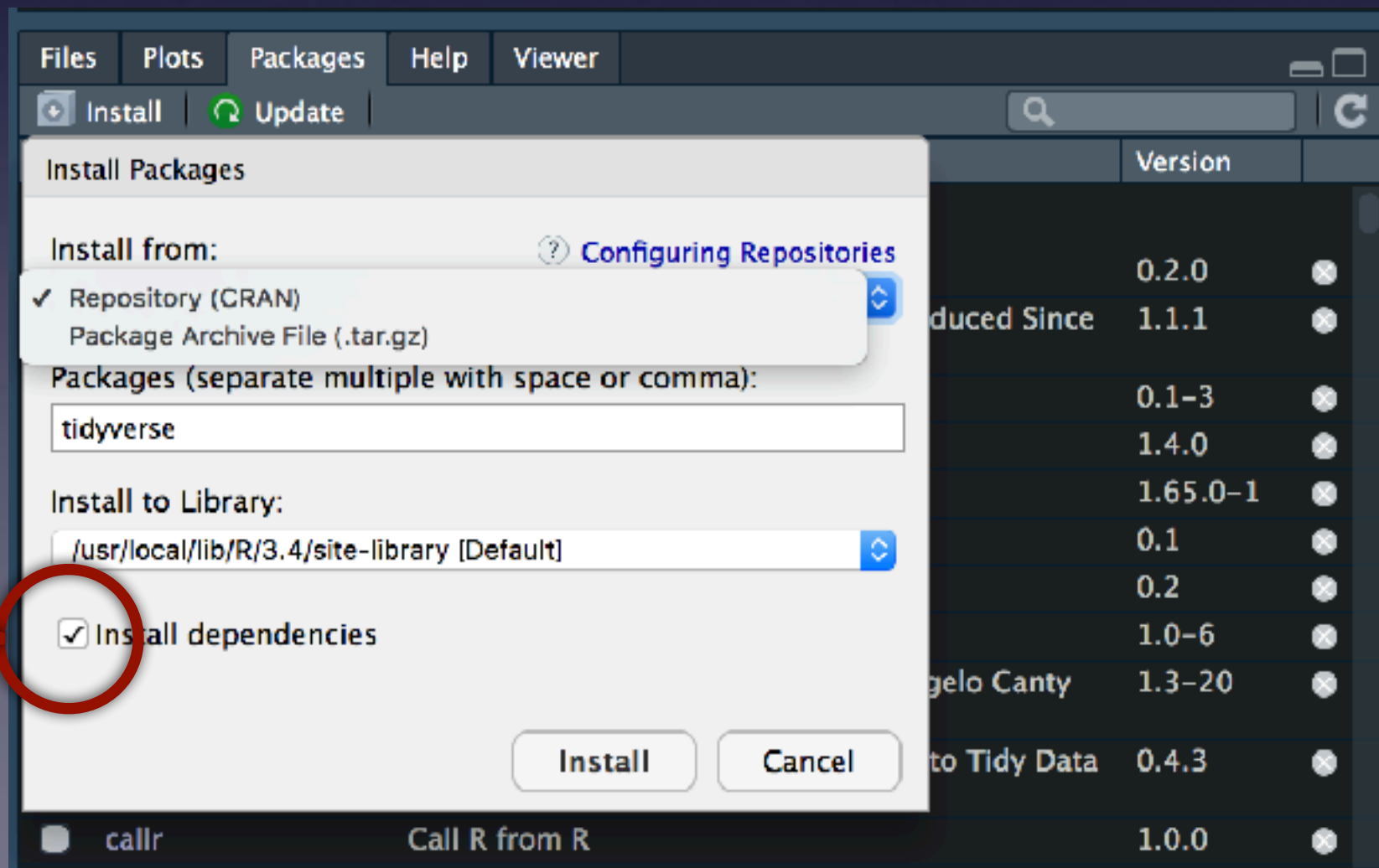
注意点

- ・ 依存packageと一緒にインストールすること

packageの追加

packageの追加方法

1. RStudioのPackagesタブを使う
 - Repository(CRAN)にチェック
 - インストールしたいpackage名を入力
 - installを押す



チェックを
外さないこと

packageの追加

packageの追加方法

2. install.package関数を自分で書く

- 複数インストールする時はc()で囲む
- 依存packageのうちSuggestsまでインストールする時は
“dependencise = TRUE”を追加

Example

```
install.pakcages(c("readr", "data.table"),  
                  dependencies = TRUE)
```

参照:

<https://stats.biopapyrus.jp/r/basic/package.html>

packageの呼び出し

使う時には “library()” or “require()” で呼び出す

```
> library(dplyr) or require(dplyr)
```

何が違うか？

- packageがインストールされていなかった場合の挙動が違う
 - library(): packageないとエラー出して終了
 - require(): packageがあると “TRUE”、ないと “FALSE” を返す

参照:

<https://stackoverflow.com/questions/5595512/>


```
install_func <- function(Pack) {  
  if(require(Pack)){  
    print(paste0(Pack, "is loaded correctly"))  
  } else {  
    print(paste0("trying to install", Pack))  
    install.packages(Pack)  
    if(require(Pack)){  
      print(paste0(Pack, "installed and loaded"))  
    } else {  
      stop(paste0("could not install", Pack))  
    }  
  }  
}
```

MENU

1. Rとは何か
2. Rを使うための環境設定
 - RとRStudioのインストール
3. Rの機能拡張
 - Packageの取り込み
 - Packageの呼び出し
4. 外部データのInput方法
 - table形式
 - その他
5. 結果のoutput方法
 - dataframe形式

外部データのinput方法

table形式の場合、方法は3つ

- baseか、packageの"data.table"/"readr"

1.base

```
utils::read.table(file_path)
```

2.data.table

```
data.table::fread(file_path)
```

3.readr

```
readr::readr_delim(file_path)
```

外部データのinput方法

良いところ悪いところ

1. utils::read.table

良い

- ・すぐ使える
- ・ほぼ全てのfile形式に対してwrapperがある

悪い

- ・file形式によってwrapperを指定するか引数をつける必要がある
- ・データ量が増えると読み込みが遅い
- ・カラム名が勝手に変わることがある

外部データのinput方法

良いところ悪いところ

2.data.table::fread

良い

- ・読み込みが早い(大きいファイルほど顕著)
- ・自動的にfile形式に合わせて読み込む

悪い

- ・"data.table = FALSE"という指定が必要
- ・ERRORが出た時に融通が効かない

外部データのinput方法

良いところ悪いところ

3.readr::readr_delim

良い

- ・読み込みが早い
- ・ほぼ全てのfile形式に対するwrapperがある
- ・カラム名をほぼそのまま読み込む

悪い

- ・file形式によってwrapperを指定するか引数をつける必要がある

外部データのinput方法

その他の形式

- 癖が強いのでオススメしません…

Excelデータ

```
readxl::read_excel("file_path")
```

SPSSデータ

```
haven::read_spss("file_path")
```

SASデータ

```
haven::read_sas("file_path")
```

MENU

1. Rとは何か
2. Rを使うための環境設定
 - RとRStudioのインストール
3. Rの機能拡張
 - Packageの取り込み
 - Packageの呼び出し
4. 外部データのInput方法
 - table形式
 - その他
5. 結果のoutput方法
 - dataframe形式

結果のoutput方法

table形式の場合

```
utils::write.table(data, "file_path",  
                    row.names = FALSE)
```

注意点

指定しないとカラム名と一緒に列名も出力する

列名の出力をしない指定が"row.names = FALSE"

```
readr::write_csv(data, "file_path")
```

最後に

Rはpackageをうまく使えばやりたいことのほとんどは
できるようになっています

packageの有無や使い方に困ったら…

1. help()を使う
2. CRANの公式pdfを読む
3. r-wakalangに質問する
4. google先生のお世話になる

ENJOY!