# Introduction to `rattle` in R

**Xiaogang Su, Ph.D.**
Department of Mathematical Sciences
University of Texas at El Paso (UTEP)
xsu@utep.edu

## Contents



## 1 Install `rattle`

In this topic, we introduce the R GUI facility, package {`rattle`} for data analysis and modeling. `Rattle` (Williams, 2009) is free and open source software, which is built on top of the R statistical

software package (R Development Core Team, 2012). As free software the source code of Rattle and R is available to everyone, without limitation. Everyone is permitted, and indeed encouraged, to read the source code to learn, understand verify, and extend it. Both are supported by a worldwide network of some of the world's leading statisticians and implements all of the key algorithms for data mining.

To install the package, simply type

```
install.packages("rattle")
```

Once installed, the function `rattleInfo()` provides version information for `rattle` and dependencies and will also check for available updates and generate the command that can be cut-and-pasted to update the appropriate packages.

```
install.packages(rattleInfo())
```

In order to use the graphical user interface (GUI) feature, you also need to have GTK installed, which is a highly usable feature-rich toolkit for creating graphical user interfaces.

```
install.packages("RGtk2")
```

The following options are set in order to obtain output formatted for publication. We can see that `width=` is set to 58 to limit the line width for publication. The two options `scipen=` and `digits=` affect how numbers are presented:

```
options(width=58, scipen=5, digits=4, continue=" ")
```

Now we are ready to start with `rattle` by typing

```
library(rattle)
rattle()
```

The command `rattle()` will start the GUI for `Rattle`. Many functions and commands can also take arguments, which we indicate by trailing the argument with an equals sign. The `rattle()` command, for example, can accept the command argument `csvfile=`.

## 2 Framework and Workflow of `Rattle`

`Rattle` follows the Cross Industry Process for Data Mining (CRISP-DM, 1996) framework for delivering data mining projects. CRISP-DM identifies six steps within a typical data mining project: Problem Understanding; Data Understanding; Data Preparation; Modeling; Evaluation; Deployment (PDDMED).

The typical workflow for a data mining project in the context of `Rattle` can be summarized as:

1. Load a Dataset;
2. Select variables and entities for exploring and mining;
3. Explore the data to understand how it is distributed or spread.
4. Transform the data to suit our data mining purposes.

5. Build Models.

6. Evaluate the models on other datasets.

7. Export the models for deployment.

For a brief introduction on how to start with Rattle, check out the following two YouTube videos (Checked on 08/29/2016):

- *Rattle for Data Mining* at `http://www.youtube.com/watch?v=OBilaZZpvGs`
- Rattle – Data Mining in R at `https://www.youtube.com/watch?v=ARGfOHPVERc`

# 3 Load a Dataset into Rattle

Rattle offers direct input of an Excel Spreadsheet, CSV, Rdata, etc. This can be seen from the main screen.

To load a data set of other types into Ratthle, you can always read them first into R using the `library(foreign)` and other libraries such as `sas7bdat` for SAS datasets. Once an R dataset in available, you can load it into Rattle by selecting the "R Dataset" option for Source in Rattle main manu screen. The **log** tab in rattle provides all the R codes used in the GUI analysis. These facilitate the necessary interaction between rattle and R programming.

## 3.1 Load SPSS Datasets

As an example, we read an SPSS dataset, **cancer.sav**, into R first.

```
library(foreign)
cancer <- read.spss(file="http://calcnet.mth.cmich.edu/
    org/spss/V16_materials/DataSets_v16/Cancer.sav", to.data.frame=T)
```

Once the datasets that we wish to use with Rattle has been constructed or loaded into the same R session that is running Rattle, they are ready to be loaded.

## 3.2 Load SAS Datasets

The following offers a quick example of reading an SAS dataset.

```
install.packages("sas7bdat")
library(sas7bdat)
hotel <- read.sas7bdat(file="http://bus.utk.edu/stat/stat579/hotel.sas7bdat")
```

See Figure 1 for how to load the `hotel` data into Rattle. After making the selection, click on the Execute button. You will see a screen with info on variables in the dataset.

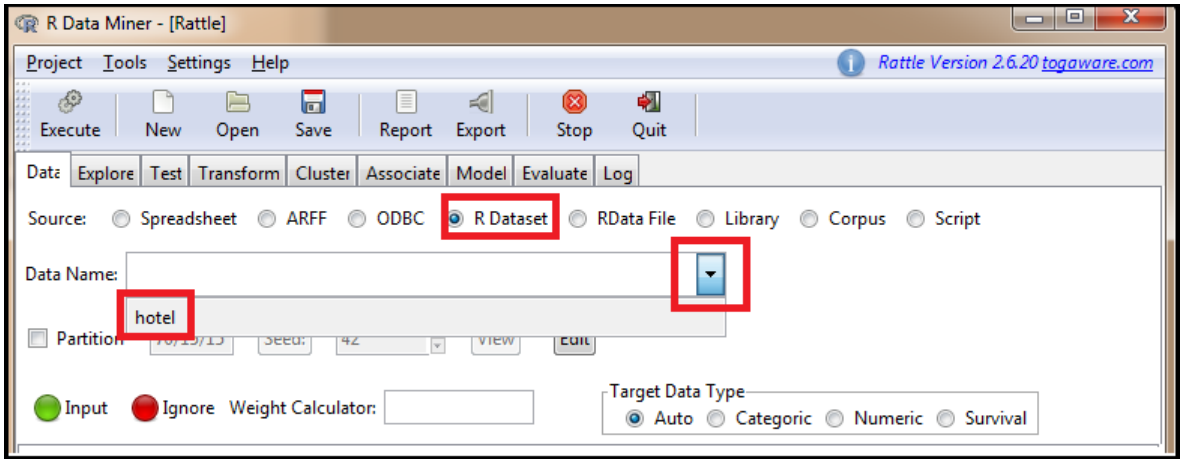


Figure 1: Loading an already defined R data frame as a dataset for use in Rattle.

### 3.3 R Datasets Available in Packages

Almost every R package provides a sample dataset that is used to illustrate the functionality of the package. We can explore the wealth of datasets that are available to us through the packages that are contained in our installed R library. The following command shows a long list of all datasets available in your installed packages.

```
da <- data(package=.packages(all.available=TRUE))
names(da)
sort(paste(da$results[, "Item"], " : ",
    da$results[, "Package"], " : ",
    da$results[, "Title"], sep=""))
```

To access a datset, you can load its associated library first then the datasets in that library would become available. Otherwise, `data()` needs to be run before the dataset can be accessed.

## 4 A Classification Example

Consider the `car` evaluation data from UCI data repository:

`http://archive.ics.uci.edu/ml/datasets/Car+Evaluation`

The goal is to built a predictive model to evaluate cars according to the following concept structure:

- `evaluation` – car acceptability (the target variable)
- PRICE overall price
  - `buying` – buying price
  - `mainte` – price of the maintenance
- TECH technical characteristics
  - COMFORT comfort

    - `doors` – number of doors
    - `persons` – capacity in terms of persons to carry
    - `lug.boot` – the size of luggage boot
  - `safety` – estimated safety of the car

Because of known underlying concept structure, this database may be particularly useful for testing constructive induction and structure discovery methods. The data file is in CSV format. So we first bring in the data using `read.csv`.

```
car <- read.csv(header=F,
    file="http://archive.ics.uci.edu//ml//machine-learning-databases//car//car.data",
    col.names=c("buying", "maint", "doors", "persons", "lug.boot",
        "safety", "evaluation"))
dim(car)   # 1728    7
head(car)
```

Via illustration on this data set, we will walk through some simple data exploration and analysis steps in class.

## References


CRISP-DM (1996), Cross Industry Process — Data Mining. `http://www.crisp-dm.org/`.

R Development Core Team (2012). R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna. `http://www.R-project.org/`.

Williams, G. J. (2009), Rattle: A data mining GUI for R, *The R Journal*, **1**(2): 45–55. `http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf`.

Williams, G. J. (2011). *Data Mining with Rattle and R.* Springer UseR! Series. New York, NY: Springer.

# APPENDIX

## A The Cancer Data

The data set contains part of the data for a study of oral condition of cancer patients conducted at the Mid-Michigan Medical Center. The oral conditions of the patients were measured and recorded at the initial stage, at the end of the second week, at the end of the fourth week, and at the end of the sixth week. The variables age, initial weight and initial cancer stage of the patients were recorded. Patients were divided into two groups at random: One group received a placebo and the other group received aloe juice treatment.

Sample size, $n = 25$ patients with neck cancer. The treatment is Aloe Juice. The variables in the data set are:

- `ID` – Patient ID
- `TRT` – treatment group: 0 = placebo; 1 = aloe juice
- `AGE` – patient's age in years
- `WEIGHTIN` – patient's weight at the initial stage
- `STAGE` – initial cancer stage, coded 1 through 4
- `TOTALCIN` – oral condition at the initial stage
- `TOTALCW2` – oral condition at the end of week 2
- `TOTALCW4` – oral condition at the end of week 4
- `TOTALCW6` – oral condition at the end of week 6

NOTE: The variables `TOTALCIN`, `TOTALCW2`, `TOTALCW4`, and `TOTALCW6` are the dependent variables, constituting repeated measures over time. The variables `AGE`, `WEIGHTIN` and `STAGE` are considered covariates in this study. The variable `TRT` is a between-subject factor in the study.