

*New Challenges for Statistical Software - The Use of R in Official Statistics*  
The 4th International Conference

# Integrating R and Java for Enhancing Interactivity of Algorithmic Data Analysis Software Solutions

**Titus Felix FURTUNĂ**

**Claudiu VINȚE**

Bucharest University of Economic  
Studies

Bucharest, 7-8 April 2016



# Agenda



- Introduction
- Objectives
- Premises
- Libraries, tools and techniques for supporting Java within R
- R code integration within JavaFX framework
- Case study upon hierarchical cluster analysis
- Conclusions and further research

# Introduction

Software solutions for statistical data processing require various levels of interoperability and integration



Competence of an applications residing on different platforms, different operating systems (OS), written in different programming languages to interchange data seamlessly, as components of a cohesive distributed solution

Heterogeneous applications, platforms, data feeds, data formats



R is multiplatform, import/export capabilities from and towards multiple database systems  
Microsoft Excel, Microsoft Access, Oracle, MySQL, SQLite



# Objectives



Exploit both R and Java strengths



Interoperability at code level



Integration of JavaFX code within R applications



Integration of R code within JavaFX applications

A series of dark blue wavy lines on the left side of the slide, extending from the top arrow down to the bottom.



# Premises

- Java powerful graphical user interface (GUI) frameworks (Java Foundation Classes, JavaFX)
- R code provides a more efficient way of processing data for statistical purposes
- GUI is considered a crucial part of programming for any software development, being currently included in the area of Human Computer Interaction (HCI)
- The aim of a GUI is to offer to user a facile and friendly manner of interaction with a given application, improving thus efficiency, usability, and lowering the learning curve




# Libraries, tools and techniques for supporting Java within R

- **RGtk2** is delivered as an R package for programming GUI-s in R
- **tcltk2** package, developed by Philippe Grosjean, is built upon Tcl/Tk library
- **wxWidgets** library, written in C++, contains graphical components developed in native code for various platforms, preserving the platform specific look-and-feel
- **rJava** is package that allows for running Java within R applications via JNI
- **JRI** is a Java-R interface which facilitates running R code within Java applications
- **Rengine** is a Java interpreter for R
- **Rserve** is a TCP/IP server which provides the means for using R



# JavaFX advantages as GUI builder framework

- 
- The event-listener interaction mechanism is better designed, more consistently, and oriented on properties
  - Graphical components structure is less heavy; layout, shape, and controls belong now to a unified structure based on *Node* class; the control sets is more divers comparing to Swing
  - JavaFX employs widely the concepts of property; a property is an object that can be monitored through event-listener mechanism
  - JavaFX applications look-and-feel can be customized using CSS files
  - JavaFX allows for applying special effects to any *Node* type component
  - JavaFX offers direct support for animation and geometrical transformations



# Integrating JavaFX within R code

- JavaFX applications can naturally integrate any Java API
- JavaFX applications are implemented as self-contained applications, having their own children within Java Runtime and JavaFX Runtime environments
- Using rJava package there can be instantiated Java objects in R, and have access to their fields and methods
- A standard JavaFX application is a `javafx.application.Application` object type. JavaFX Runtime creates the Application object
- Then it calls orderly the methods *init()* and *start(Stage)* of Application class. Later on, *stop()* method is executed when the application ends. An JavaFX application ends either when the static method *exit()*, or when all the application windows are closed





# The main functions provided by rJava package

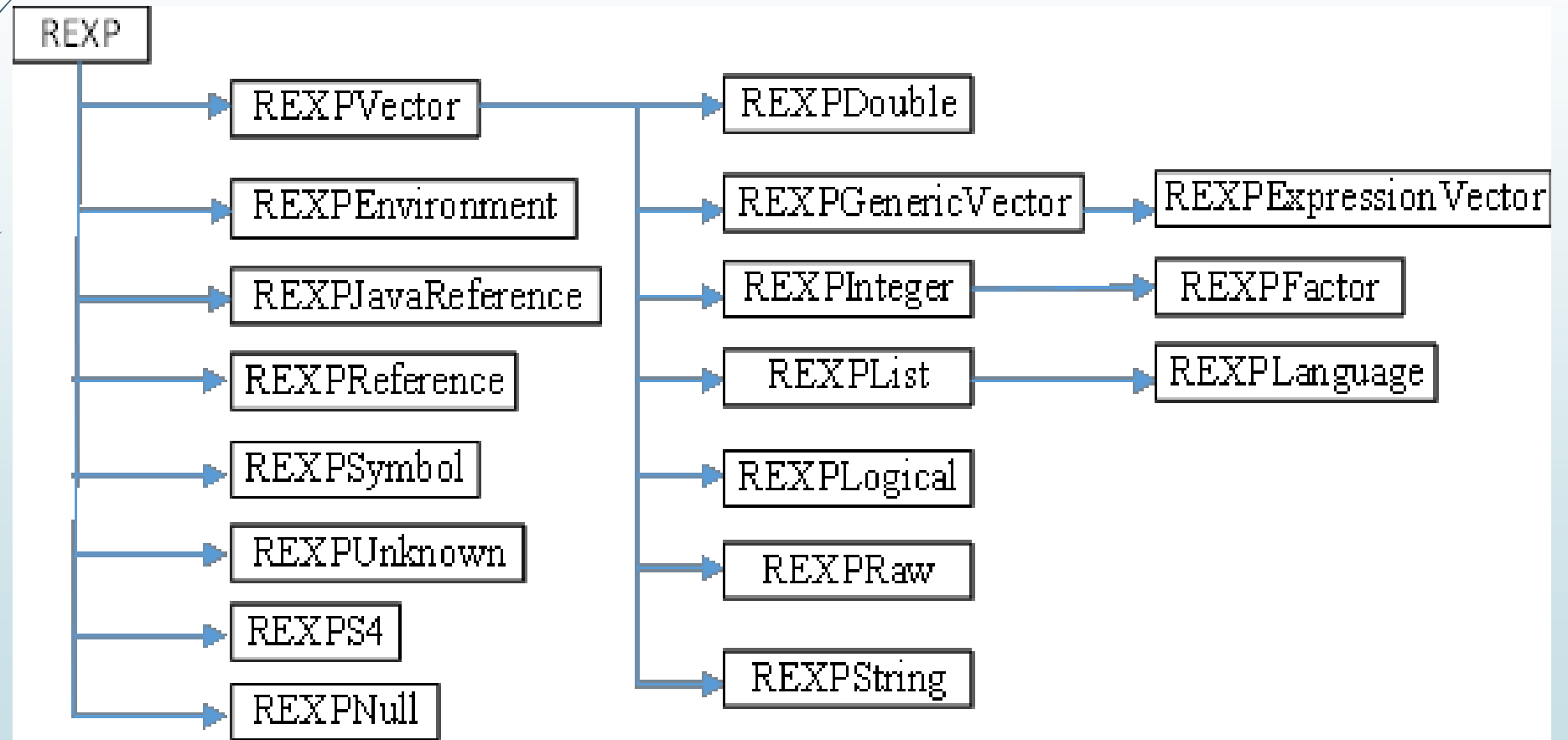
- `.jinit()` – for initializing JVM;
- `.jaddClassPath(path)` – provides an additional path to directories hosting JAR files along with the default one;
- `.jclassPath()` – lists the directories from the current path;
- `.jnew(class, ..., check=TRUE, silent=!check)` – instantiates a Java object;
- `.jcall(obj, returnSig = "V", method, ...)` – for calling a method belonging to a Java class, object instantiation respectively.



# Integrating R code within an JavaFX application

- **JRI** library implements a parsing and interpreting engine for R, and then returns the results to Java
- The main class of the specification is **JRIEngine**, through which a **REngine** object can be instantiated and then used for parsing and executing R code
- A Java application can create only one instance of **REngine**, which is thread-safe
- The link between R and Java objects is provided through **REXP** class and its subclasses
- Through **REXP** class is modeled any type of R object, at the most generic level

## REXP class hierarchy






## REXP methods for creating **DataFrame** and **Matrix** data structures

- ***public static REXP createDoubleMatrix(double[ ][ ] matrix);*** Creates an R *Matrix* object from the specified Java multidimensional array
- ***public static REXP createDataFrame(RList l) throws REXPMismatchException;*** Creates an R *DataFrame* object from the supplied Java JRI *RList* object
- In JavaFX data table visualization is achieved through *TableView* class
- As part of our research, we propose an implementation of a graphical user interface that implies an extension of JavaFX *TableView* class, in order to handle the tabular perspective of R objects like *data.frame* and *matrix*

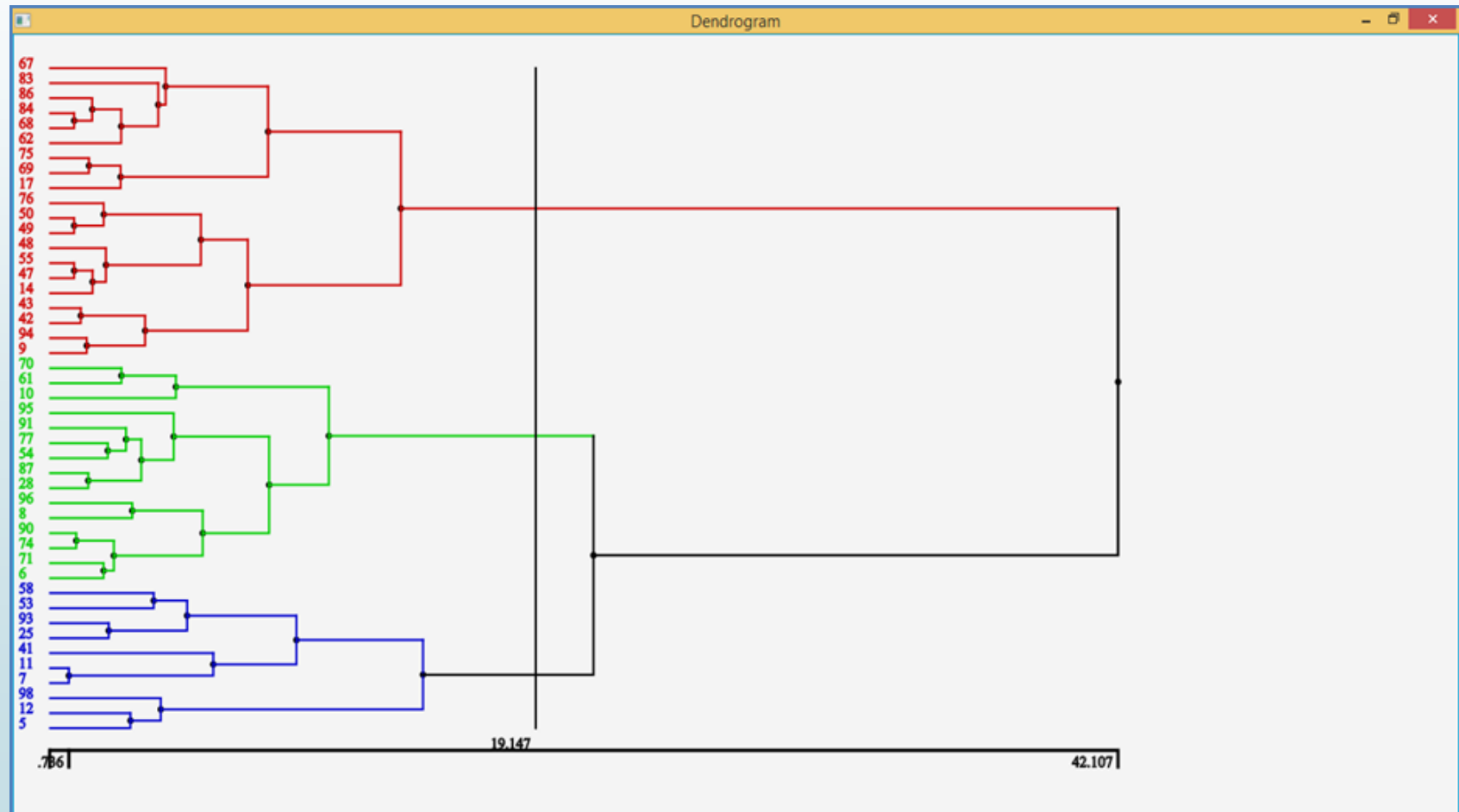


# Case study upon hierarchical cluster analysis

The data input consist of instances (buildings in Bucharest) and variables:

- year of construction (numeric value),
  - number of floors (numeric),
  - historical monument (categorical value),
  - structural frame type (categorical),
  - risk assigned to the masonry type (categorical),
  - risk assigned to the type of structural frame (categorical),
  - number of apartments in the building (numerical),
  - number of people in the building during daytime (numerical),
  - number of people in the building during nighttime (numerical),
  - number of people working in the building (numerical),
  - number of people visiting the building (numerical).
- 

R script outputs the dendrogram created in JavaFX.



# JavaFX GUI for hierarchical clustering

**Hierarchical clustering**

**Open Data**

**Input**

- ☒ Read instances names
- ☐ Read variables names
- ☐ N degrees of freedom

**Variables**

**Distances**

- ☒ Euclidean
- ☐ Minkowski
- ☐ Manhattan
- ☐ Canberra

**Categorical Variables**

**Methods**

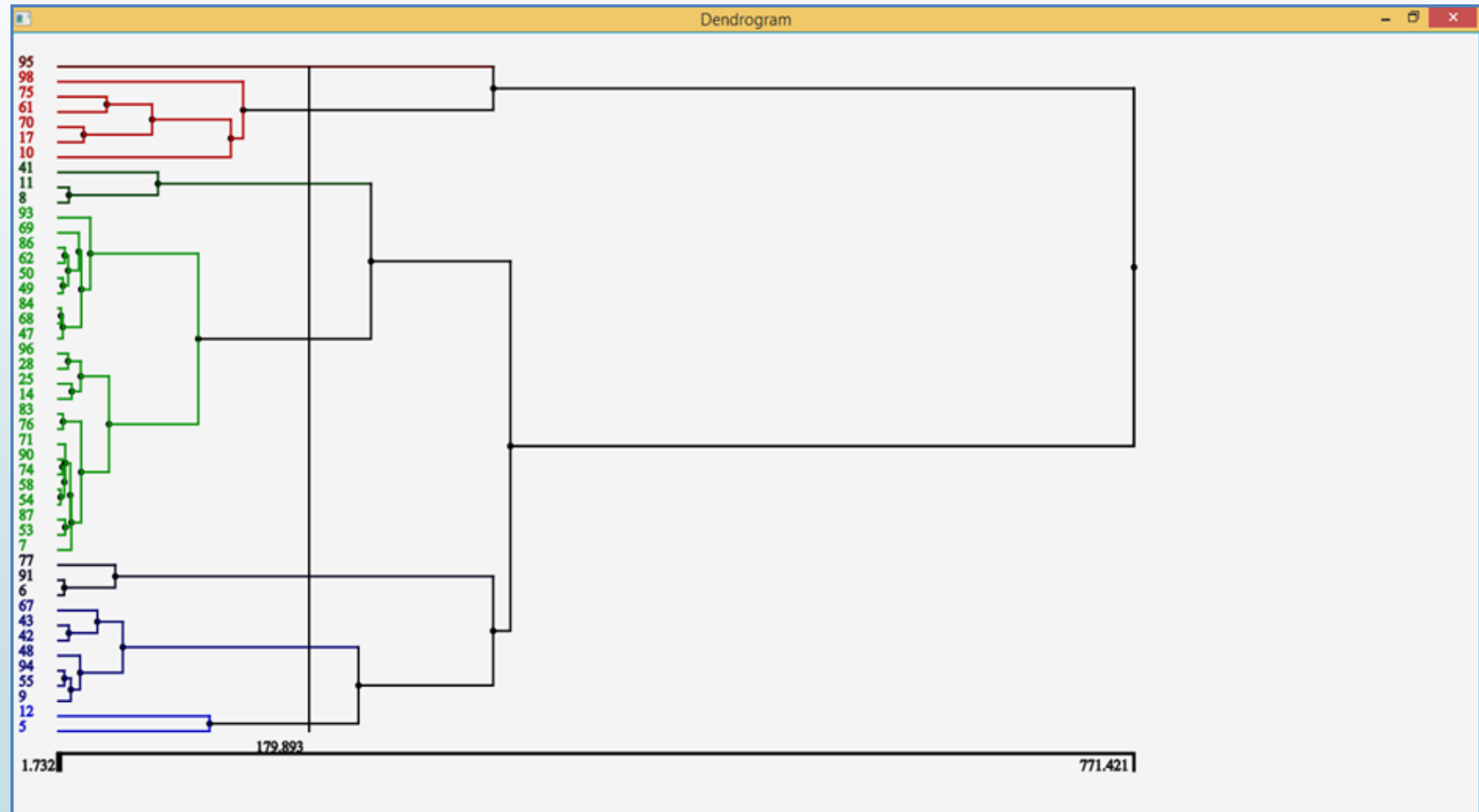
- ☒ Complete Linkage
- ☐ Single Linkage
- ☐ Average Linkage
- ☐ Ward

**Distance Matrix**

**Hierarchy**

**Hierarchical tree plot**

The corresponding dendrogram is then generated within the JavaFX application







# Conclusions and further research

- Treat both aspects of integration and interoperability that refer to integrating Java code into R applications, and bringing R processing sequences into Java driven software solutions
- Integrating R developing environment with Java programming language for interoperability at a source code level
- The goal is to create an API wrapped around rJava and JRI, that would readily allow for combining the intensive data processing capabilities of R programming language with the flexibility offered by Java programming language, in order to develop interactive GUI-s based on JavaFX as presentation platform
- Our research will continue to explore for graphical tools developed through R and Java integration for supporting pattern recognition and cluster analysis



Thank you!