Introduction to usage of VwGHat - the package for mining VwGH jurisdiction.

Martin Kober and Georg Zajko

February 15, 2013

1 Introduction

The Rechtsinformationssystem of the Austrian Federal Chancellary (`http://ris.bka.gv.at`) provides an online archive of all Austrian laws and supreme court jurisdiction. The purpose of this package is to make the rulings of the Austrian supreme administrative court usable for datamining in R and provide some basic mining functions.

2 Usage

2.1 Creation of the Database

The first step in the mining process is to create a database of the rulings.
The recommended and tested database for this package is SQLite, as it is available on most systems and requires little to none configuration. However, any database that supports basic functions of the DBI package should work. The only thing that the user needs to do is open a connection to the database of choice, everything else is handled by the various package functions.
To open a connection to an SQLite database (or create a new one if it does not exist), simply type:

```
R> library("RSQLite")
R> con <- dbConnect(dbDriver("SQLite"), dbname="risdb")
```

The next step is to fill the database:

```
R> build_RIS_db(con)
```

Beware that the database contains approximately 70.000 rulings so this will take several hours. It is possible to download only a subset of the rulings using the `to=()` and `from=()` arguments.

2.2 Mining the database

The package provides several functions for extracting interesting information from the database. They generally require only the database connection object as a parameter and create a new table in the database with the mining results.
The following functions are defined:

```
R> compute_result(con)
R> compute_senate_size(con)
R> compute_senate_number(con)
R> compute_duration_of_proc(con)
R> compute_affected_norms(con)
```

The documentation of the respective functions provide further detail on how (and in which table) the results are stored in the database.

2.3 Extracting the results

The easiest way to export the data into an R dataframe is to export a single table:

```
R> df <- dbReadTable(con, "<table_name>") # substitute table name
```

SQL has to be used for more complex queries. The function `dbGetQuery()` executes SQL and returns the result as a dataframe.

3 Defining new mining functions

The package provides an easy way to write new mining functions with minimal exposure to database specifics through the function `compute_from_db()`:

```
R> compute_from_db(con, func, table_name, fields_in, fields_out, field_types="TEXT",
+                  table_setup=setup_foreign_table, include_foreign_key=TRUE, limit=0)
```

All included mining functions use this function; `compute_senate_size()` for example, is defined as follows:

```
R> compute_senate_size <- function(con, limit=0)
+         compute_from_db(con, senate_size, table_name="senate_size",
+                  fields_in="Betreff", fields_out="size", field_types="INTEGER", limit=limit)
```

Here, `senate_size()` is the actual worker function that determines senate size. After creating the table ”senate_size”, `compute_from_db()` calls the function with a single-row dataframe with the required fields from the first record in the db, ”Betreff” in this case (and a record id). `senate_size()` returns a dataframe (with one row, but more than one row is possible) with the indicated fields, ”size” in this case. `compute_from_db()` then writes this data to the db and calls `senate_size()` with the next row in the db.

4 Database structure

The main table ”Rechtsinformationssystem” contains all the information from RIS. Column names are according to the sections in RIS:

- Norm
- Gerichtstyp
- Geschaeftszahl
- Entscheidungsdatum
- Veroeffentlichungsdatum
- Index
- Spruch
- Dokumentnummer

- Schlagworte
- Sammlungsnummer
- Beachte
- Gerichtsentscheidung
- RisID
- SP_Nr