After downloading data from Ruslana and performing SQL script (described in file “Процесс выгрузки данных большого объема и их импорта в SQL.docx”) we’ve got:

1. Folder with data files (.csv) from Ruslana grouped by regions. Each file contains information on firms in the corresponding region including:
   1. long and short names of the company separately;
   2. BvD, OKPO, INN ids;
   3. region, city
   4. status of the firm (active, closed, in the process of reorganization), year of status changing (the only date in Ruslana helpful with determining which firms had been already closed by the time of research), foundation year
   5. organization form (“ООО”, “ЗАО”, “ИП” etc.)
2. Initial file from ICSID db in .csv format

**Task:** to match the name of the firm in ICSID table with long or short name of the firm in Ruslana db in order to get basic identificators: OKPO, INN, BvD number. We account for the region and city where it is possible.

**Part 1. Editing ICSID table**

1. Leave just the columns needed for the matching task: "reg\_idnew", "inn", "name", "address", "firm"
2. Delete all the symbols of new line and return of carriage that can affect operations with rows
3. Fill in the blank spaces with the info for the corresponding school
4. Assign the “row\_id” to each row. It makes looking for the corresponding row in initial table easier
5. Extract the city from the college address. Leave only the name without prefix “г.”, “село” etc.

I’ve written the specific code using “regular expressions” for that.

1. Edit the columns “firm” and “city”. Make new columns “ed\_firm” and “ed\_city” with the following changes:
   1. Change hyphen "-" and dash " - " to space " "
   2. Separate “firm” if there are parentheses. Make 2 new columns with info inside and outside of parentheses
   3. Remove all the punctuation
   4. Convert to lower case
   5. Remove multiple whitespaces and ones from start and ending of the string
   6. Standardize the symbol "№..."
   7. Transliterate if needed. Using specific R function “stri\_trans\_general” from “stringr” package
   8. Remove all rows with blank or zero ("0") "firm" cell (it means that there are no companies corresponding to the particular school)

**Part 2. Creating subsidiary data frames**

1. Create vector of regions from ICSID table
2. Final report – data frame consisting of the following data:
   1. Region
   2. Phase of matching process (so far 7 of them)
   3. Number of unique rows in ICSID table corresponding to the region
   4. Number and percentage of (unique rows) matches found during the corresponding phase
   5. Number of unique firm names in ICSID table
   6. Number and percentage of (unique firm) matches found during the corresponding phase
   7. Number of rows in Ruslana db corresponding to the region
   8. Total number of rows found after the final phase

**Part 3. Matching phases**

For every region id from “region” vector do the following

1. Take Ruslana data file corresponding to the region
2. Shrink Ruslana data according to the following:
   1. Delete all the firms that were founded after Decemver, 31, 2013
   2. Delete all the firms that have become inactive before January, 1, 2013
3. Edit Ruslana rows (the process is similar to ICSID case)
   1. Change hyphen "-" and dash " - " to space " "
   2. Remove all the punctuation
   3. Convert to lower case
   4. Remove multiple whitespaces and ones from start and ending of the string
   5. Standardize the symbol "№..."
   6. Extract the name of the city. Leave only the name without prefix “г.”, “село” etc.

For each phase the duplicated rows in the end are deleted

Phase 1

1. Finding perfect match with long firm name in Ruslana
2. Finding perfect match with short firm name in Ruslana

Phase 2

1. Adding form abbreviation to the long name of the company («ооо», «оао», «зао», «ао», «муп», «ип», «фгуп»)
2. Editing rows with "ип" - if #of words in the string with "ип" is more than 4 make them 4 (to leave only "ип иванов владимир викторович" instead of "ип иванов владимир викторович магазин ромашка")
3. Deleting initials from the name and adding "ип" to the long name of the company
4. Getting initials from the full name and adding "ип" to the long name of the company
5. Editing rows with "ип" - if #of words in the string with "ип" is more than 4 make them 4 in ICSID table (the same logics as in b)
6. Changing long business form to short business form. («общество с ограниченной ответственностью» to «ооо» etc.)

Phase 3

The same as in Phase 2 but for the short firm name

Phase 4. Working with ICSID database

1. Adding organization forms to firm names to ICSID table. (“ооо”, “зао”, “ао”, “оао”, “общество с ограниченной ответственностью” etc)

Phase 5

Matching names outside and inside parentheses using the same methods as in Phases 1-4

Phase 6

1. Leave only rows where cities of school and firms match. If there are no firms that match to school by city then leave all the firms found
2. Account for business form. Delete matches if business forms don't match for sure. Leave matches if there is no correspondence in form (the same logics as in correspondence of cities)

Phase 7. Fuzzy Matching

For firm names that don’t have matches after Phases 1-6 we use Fuzzy string matching algorithm. It is based on special metrics – string distance. There are several popular ones.

Comparison of different string distance algorithms can be found, for example, here <http://www.joyofdata.de/blog/comparison-of-string-distance-algorithms/>

1. We use generalized Levenshtein (edit) distance as it has showed better results than the others. It gives the minimal possibly weighted number of insertions, deletions and substitutions needed to transform one string into another (details here <https://stat.ethz.ch/R-manual/R-devel/library/utils/html/adist.html>).

We use chosen algorithm to find the best match for firm name from ICSID among long names and short names from Ruslana separately. “Best” here means less value of the distance.

Thus, in the end we have two different tables (for each region) with matches found using fuzzy string matching algorithm (one for long names, another one for short names).

1. Manual coding matches found in a). Adding column Match\_true. 3 possible values: 0- for clearly incorrect match, 1 – for clearly correct match, 2- if the correctness is not obvious
2. Create table with clearly true matches consolidating matching results for “long” and “short name” cases

After 7 phases we get 4 data frames for each region: 1st – for results from phases 1-6, 2nd – for results from phase 7, 3rd – for firms with no matches after all phases, 4th –report with statistics (described in Part 2.2 above)

**Results:**

Summary is in the folder “Final” (files with matched rows) and file “report” (statistics)

**Some Notes:**

1. For some regions (Moscow, Moscow region, Krasnodar region) R script needed more CPU then my computer had. I used computer “Burenka” with bigger CPU (under supervision of Boris Demeshev, Department of Applied Economics)
2. One of the possible approaches after Phase 7: do Fuzzy matching for the firm names that we didn’t match after 7 phases. But now take Ruslana row corresponding to the second minimum (second best answer). It looks like it works reasonably well, especially for the rows that were coded “2” at Phase 7 b). Corresponding part of coding can be found in the end of the R script.