LexisNexis Risk Solutions **André Duarte**
HPCC/ECL coding challenge Report 16 June 2017

Problem Statement

The goal of this report is to show my results after working through a data file using ECL. I will first talk about the contents of the available data. Then, I will mention how I used ECL code to extract meaningful relationships and information from said data.

Exploratory Data Analysis

The dataset I used for this assignment is from data.rio, a website that collects open data from Rio de Janeiro, in Brazil. In particular, I worked with this csv file that contains information about 1,484 public municipal schools in Rio de Janeiro. The csv file contains 20 columns. These include a unique school identifier number, the name of the school, its address (street, neighborhood, number, complement, and zip code), IDEB 1 and 2 scores (Basic Education Development Index), latitude and longitude, whether the school is a "Ginásio Carioca", whether it has accessibility options, which grades ("séries") and shifts ("turnos") are available, whether the school is a "Ginásio Olímpico", the name of its director, the telephone number, the INEP (National Institute of Studies and Educational Research) identifier, and finally the name of the pedagogical coordinator. It is important to know that not all rows (schools) are complete, and some data may be missing.

The file is loaded (sprayed) into HPCC using the ECL Watch interface and UTF-8 encoding. I won't go into detail about how to do that in this report.

It is important to know that this is my first foray with ECL. I tried to keep best practices with my code, but acknowledge that it may likely be improved. My ECL code is annexed to this document and available upon request.

Results

Number of schools by neighborhood

One of the first questions we may have when obtaining this data is how the schools are distributed by neighborhood ("bairro"). For instance, which neighborhood has the most schools? How many is that? What about the neighborhood with the least number of schools? By creating a sorted cross-table grouped by this metric ("bairro"), we can see that Santa Cruz has the most schools (78), followed by Campo Grande (75), and Bangu (63). On the other hand, there are 76 neighborhoods with only one school. However, it must be noted that these results are not exactly correct. First, the grouping is done in such way that the neighborhood is case-sensitive. This means that, although 75 schools are in the "Campo Grande" neighborhood, we can also see one school in "CAMPO GRANDE" that was not grouped together. Second, some neighborhoods are written in more detail. For example, we can find in our data several subcategories for "Campo Grande", such as "Campo Grande/Centro", "Campo Grande/Monteiro", "Campo Grande/Benjamin Dumont", "Campo Grande/Cachamorra", "Campo Grande/Dumont", "Campo Grande/Loteamento Joari", and "Campo Grande/Posse". Ideally, these should all be grouped together, but I could not find a way to do it in ECL in my short time learning this language.

Neighborhoods with most schools that work full-time

In Brazil, schools usually work part-time, meaning that students in general either have classes in the morning and have the afternoon free ("primeiro turno") or the other way around ("segundo turno"). Some schools have a "terceiro turno" for older students that have lessons at night. Finally, some schools are now beginning to offer full-time schooling, which is called "turno integral".

With ECL, I created a derived dataset of schools that contains the keyword "Integral" in the "turnos_atendidos" column. Just like above, we can group them by neighborhood to see which one has the most schools that work full-time (it's the same three as earlier, Santa Cruz, Campo Grande, and Bangu). What we can do next is check the percentage of schools by neighborhood that work full-time. We obtain this by joining the two datasets above (an inner join is useful here to keep only those neighborhood that actually have this type of school). We get some interesting results. For example, all six "Ilha do Governador" schools offer full-time schooling, and both schools from the "Cidade de Deus" slum do as well. However, only around 25% of the schools in the top-three neighborhoods (by count of schools) do so.

IDEB ratings

This index is used by the federal government of Brazil to assess the quality of the public education system throughout the country. This score is calculated every few years, and the last one was in 2013. Grades range from 0 to 10, higher meaning a better quality of education provided by the school. Here, we have two such ratings: "IDEB 1o segmento" (IDEB 1) that corresponds to grades 1 through 5, and "IDEB 2o segmento" (IDEB 2) that corresponds to grades 6 through 9. The averages for the country in these two ratings were 5.2 and 4.2 respectively.

In Rio de Janeiro, schools in "Urca" averaged a mean IDEB 1 rating of 6.7, while "Todos os Santos" and "Praia da Bandeira" both achieved a 6.6 average and are the highest in the city. "Urca" also leads the ranking for the IDEB 2 scores, with a 6.05, followed by "Campo Grande/Monteiro" with a 5.3 (note: as mentioned above, this should have been grouped with the "Campo Grande" neighborhood and therefore this result may not be significant) and "Todos os Santos" with a 4.8 average. In general, the average IDEB 1 score for public schools in Rio de Janeiro is 5.47 (standard deviation: 0.73), and that of IDEB 2 is 4.45 (standard deviation: 0.62). Both these scores are above the average for the country, which is positive for the city. However, many schools do not possess this information, so the result is incomplete.

Directors and Pedagogical Coordinators

One question we may want to verify is whether a person is affiliated with more than one school. A quick ECL query shows that this is not the case. However, 537 schools do not list a pedagogical coordinator (36% of total schools), and 83 do not specify a director (6% of total schools). These result are a nice example of how dealing with public data may sometimes be difficult, as the quality of the data may not always be of the utmost quality. Indeed, I would expect all schools to have a director.

Accesibility, Ginásio Carioca, Ginásio Olímpico

Simple groupings show that 951/1,484 schools do not provide accessibility options for disabled students, while little more than half of that – 494/1,484 – do (39 do not provide such information). Only 10 schools are considered "Ginásio Carioca" (39 also have this information missing).

Finally, only two of the schools are considered ”Ginásio Olímpico” (GEO JUAN ANTONIO SAMARANCH and GEO DOUTOR SÓCRATES).

Conclusion

In this report, we have seen how the data from public municipal schools in Rio de Janeiro can be aggregated to obtain global statistics and more detailed information about each neighborhood. In particular, we have seen that a few neighborhoods have tens of schools, while others have only one or two. We have shown that ”turno integral” is becoming a popular choice of schooling system, with some neighborhoods having all schools offering this schedule. In addition, we were able to show that ”Urca” has the highest rated schools according to the IDEB national score, and the the average across the city is higher than the national average. Finally, we have seen a few examples of how this particular dataset contains missing and badly organized entries (for example neighborhoods that contain subregions, or are written with a different case), and how these affect our results.

With more time and a more extensive knowledge of ECL, I would have liked to to some string processing in order to group the neighborhoods better (i.e., ”Campo Grande” and ”Campo Grande/XXX” should both be in the same category). I tried at least converting every neighborhood to lowercase, but the results still showed up in separate groups (even though the strings were all lowercase), so my implementation was not adequate.

I would also have liked to perform more data cleaning. Many rows (schools) are mostly empty or fully empty apart from school name and identifier number and mess up the results to some extent. I could not find a way to perform this subsetting in ECL from my short time with it.

ECL code

```
IMPORT Std;

/*
Define layout of the dataset.
This is based from the escolas__.csv file downloaded from https://goo.gl/UwvpGb.
*/
LayoutEscolas := RECORD
 INTEGER designacao;
 STRING nome;
 STRING logradouro;
 STRING bairro;
 INTEGER numero;
 STRING complemento;
 STRING cep;
 REAL ideb1;
 REAL ideb2;
 REAL lat;
 REAL lng;
 STRING ginasio_carioca;
 STRING acessibilidade;
 STRING series_atendidas;
 STRING turnos_atendidos;
 STRING ginasio_olimpico;
 STRING diretor;
 STRING telefone;
 INTEGER inep;
 STRING coordenador;
END;

/*
Load dataset from file.
Make sure file has been properly sprayed to thor target.
*/
SetEscolas := DATASET('~thor::escolas_csv', LayoutEscolas,
 CSV(heading(2), separator(','), quote('')));

//OUTPUT(Escolas_List, named('all_results'));

/*
Total COUNT of schools by neighborhood.
*/
AggPorBairro_Escolas := RECORD
 SetEscolas.bairro;
 //bairro := Std.Str.ToLowerCase(SetEscolas.bairro); lowercase the "bairro".
 //But results do not group together...
 num_escolas := COUNT(GROUP);
END;
```

```
TblPorBairro_Escolas := TABLE(SetEscolas, AggPorBairro_Escolas, bairro);

OUTPUT(SORT(TblPorBairro_Escolas, num_escolas), named('por_bairro_asc'));
OUTPUT(SORT(TblPorBairro_Escolas, -num_escolas), named('por_bairro_desc'));

/*
Subset by schools that list their schedule as 'integral'.
*/
SetIntegral := SetEscolas(Std.Str.contains(turnos_atendidos, 'Integral', true));

//OUTPUT(SetIntegral, named('turno_integral'));

/*
Total COUNT of schools with 'integral' schedules by neighborhood.
*/
AggPorBairro_Integral := RECORD
 SetIntegral.bairro;
 num_integral := COUNT(GROUP);
END;

TblPorBairro_Integral := TABLE(SetIntegral, AggPorBairro_Integral, bairro);

OUTPUT(SORT(TblPorBairro_Integral, -num_integral), named('por_bairro_integral'));

/*
Define new record structure to show the percentage of schools that offer
 'integral' schedule by neighborhood.
*/
AggPctPorBairro := RECORD
 string bairro;
 integer num_escolas;
 integer num_integral;
 real pct_integral;
END;

/*
Define transform function to calculate the percentage of schools that offer
 'integral' schedule by neighborhood.
*/
AggPctPorBairro toPct(TblPorBairro_Escolas L, TblPorBairro_Integral R) := TRANSFORM
 self := L;
 self := R;
 self.pct_integral := (R.num_integral / L.num_escolas) * 100;
END;

/*
Perform JOIN operation with the two elements defined above.
*/
```

```
TblPctPorBairro := JOIN(TblPorBairro_Escolas, TblPorBairro_Integral,
                        left.bairro = right.bairro, toPct(left, right));

OUTPUT(SORT(TblPctPorBairro, -pct_integral, -num_escolas),
       named('pct_bairro_integral_integral'));
OUTPUT(SORT(TblPctPorBairro, -num_escolas, -pct_integral),
       named('pct_bairro_integral_escolas'));

/*
Find out if there are any persons who are pedagogical coordinators
 at more than one school.
*/
AggPorCoordenador_Escolas := RECORD
 SetEscolas.coordenador;
 num_escolas := COUNT(GROUP);
END;

TblPorCoordenador_Escolas := TABLE(SetEscolas, AggPorCoordenador_Escolas, coordenador);

OUTPUT(SORT(TblPorCoordenador_Escolas, -num_escolas), named('por_coordenador'));

/*
Find out if there are any persons who are directors
 at more than one school.
*/
AggPorDiretor_Escolas := RECORD
 SetEscolas.diretor;
 num_escolas := COUNT(GROUP);
END;

TblPorDiretor_Escolas := TABLE(SetEscolas, AggPorDiretor_Escolas, diretor);

OUTPUT(SORT(TblPorDiretor_Escolas, -num_escolas), named('por_diretor'));

/*
Count of schools with and without accessibility.
*/
AggPorAcessibilidade_Total := RECORD
 SetEscolas.acessibilidade;
 num_escolas := COUNT(GROUP);
END;

TblPorAcessibilidadeTotal := TABLE(SetEscolas, AggPorAcessibilidade_Total, acessibilidade);

OUTPUT(SORT(TblPorAcessibilidadeTotal, -num_escolas), named('por_acessibilidade_total'));

/*
Count of schools with and without accessibility by neighborhood.
*/
```

```
AggPorAcessibilidade_Escolas := RECORD
 SetEscolas.bairro;
 SetEscolas.acessibilidade;
 num_escolas := COUNT(GROUP);
END;

TblPorAcessibilidade := TABLE(SetEscolas, AggPorAcessibilidade_Escolas, bairro,
                              acessibilidade);

OUTPUT(SORT(TblPorAcessibilidade, -acessibilidade, -num_escolas),
       named('por_acessibilidade_bairro'));

/*
Count of schools with and without carioca gym.
*/
AggPorGinasioCarioca_Escolas := RECORD
 SetEscolas.ginasio_carioca;
 num_escolas := COUNT(GROUP);
END;

TblPorGinasioCarioca_Escolas := TABLE(SetEscolas, AggPorGinasioCarioca_Escolas,
                                      ginasio_carioca);

OUTPUT(TblPorGinasioCarioca_Escolas, named('por_ginasio_carioca'));
OUTPUT(SetEscolas(ginasio_carioca = 'S'), named('has_ginasio_carioca'));

/*
Count of schools with and without olympic gym.
*/
AggPorGinasioOlimpico_Escolas := RECORD
 SetEscolas.ginasio_olimpico;
 num_escolas := COUNT(GROUP);
END;

TblPorGinasioOlimpico_Escolas := TABLE(SetEscolas, AggPorGinasioOlimpico_Escolas,
                                       ginasio_olimpico);

OUTPUT(TblPorGinasioOlimpico_Escolas, named('por_ginasio_olimpico'));
OUTPUT(SetEscolas(ginasio_olimpico = 'S'), named('has_ginasio_olimpico'));

/*
Average IDEB scores by neighborhood.
*/
AggPorIDEB_Escolas := RECORD
 SetEscolas.bairro;
 media_ideb1 := AVE(GROUP, SetEscolas.ideb1);
 media_ideb2 := AVE(GROUP, SetEscolas.ideb2);
END;
```

```
TblPorIDEB_Escolas := TABLE(SetEscolas, AggPorIDEB_Escolas, bairro);

OUTPUT(SORT(TblPorIDEB_Escolas, -media_ideb1, -media_ideb2), named('por_ideb1'));
OUTPUT(SORT(TblPorIDEB_Escolas, -media_ideb2, -media_ideb1), named('por_ideb2'));

/*
Subset by schools that have IDEB 1 score and get the average score.
Subset by schools that have IDEB 2 score and get the average score.
Show summary of statistics for these two metrics.
*/
SetIdeb1 := SetEscolas(ideb1 > 0);
SetIdeb2 := SetEscolas(ideb2 > 0);

AggIdeb_Escolas := RECORD
 media_ideb1 := AVE(SetIdeb1, SetIdeb1.ideb1);
 stdev_ideb1 := SQRT(VARIANCE(SetIdeb1, SetIdeb1.ideb1));
 media_ideb2 := AVE(SetIdeb2, SetIdeb2.ideb2);
 stdev_ideb2 := SQRT(VARIANCE(SetIdeb2, SetIdeb2.ideb2));
END;

TblIdeb_Escolas := TABLE(SetEscolas, AggIdeb_Escolas);

OUTPUT(CHOOSEN(TblIdeb_Escolas, 1), named('ideb_stats'));
```