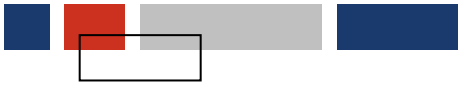
INSTITUTO NACIONAL DE ESTATÍSTICA
STATISTICS PORTUGAL

Use of R to increase the Automatic Coding Rate in Census2021

21-05-2019



Rui Alves
Almiro Moreira

About me

- Sociology degree since 1993
- Working at Statistics Portugal (Data Collection and Management Department) since 2013
- Using with R since 2015
- Statistics Portugal Data Science Team since 2019

Overview

- Create and expand existing dictionaries
- Monitor performance and validate results
- Some results
- Next steps

Challenges

- Economic Activity and Occupation are two of the most commonly collected variables in household surveys
- Approximately 100.000 responses / per year
- Census 2011: about 50% (five million) manually coded: 250.000 working hours
- Census 2021: collect all data with CAWI
- Text strings: very demanding computation

CREATE AND EXPAND EXISTING DICTIONARIES

Starting Point

- Previous experience (Census 2011)
- Available data (Household Surveys 2011-…)
- An old idea (2012)
- Paradigm shift
- Coders should be part of the process
- Reproduce manual coding process

Household Surveys
2011-2019

- LFS - Labour Force Survey
- HFCS - Household Finance and Consumption Survey
- SILC - Income and Living Conditions Survey
- AES - Adult Education Survey
- TSR - Travel Survey of Residents
- ICT - Information and Communication Technologies

Household Surveys
2011-2019

```
glimpse(df_household_survs)
```

```
## Observations: 919,309
## Variables: 2
## $ cpp_dsg  <chr> "pintor automoveis", "agente imobiliario", "operadora l…
## $ cpp_cod3 <chr> "713", "333", "522", "815", "812", "432", "751", "422",…
```

Household Surveys 2011-2019

```
# Household Survey global dataframe

df_household_survs %>%
      head() %>%
      knitr::kable()
```

cpp_dsg	cpp_cod3
pintor automoveis	713
agente imobiliario	333
operadora loja faz tudo repoem material atende clientres	522
costureira textil	815
industrial fundicao operador maquinas fundicao	812
fiel armazem	432

Frequency rules

Eligible for dictionary test:

- A string with more than one code
  - n ≥ 10 & ≥ 90% code consistency
- A string with same code
  - n ≥ 5 & ≥ 100% code consistency

```r
# my_id
df_household_survs$my_id <-
        str_c(df_household_survs$cpp_cod3,
              df_household_survs$cpp_dsg,
              sep = "_")

# count_my_id
count_my_id <- df_household_survs %>%
        count(my_id) %>%
        arrange(desc(n)) %>%
        rename(n_my_id = n)

# n_cpp_dsg
count_cpp_dsg <- df_household_survs %>%
        count(cpp_dsg) %>%
        arrange(desc(n)) %>%
        rename(n_cpp_dsg = n)

# joins n_cpp_dsg & n_my_id to df_household_survs
df_household_survs <- left_join(df_household_survs, count_cpp_dsg, by = "cpp_dsg") %>%
        left_join(count_my_id, by = "my_id")

# prob
df_household_survs <- df_household_survs %>%
        mutate(prob = n_my_id * 100 / n_cpp_dsg) %>%
        distinct(my_id, .keep_all = T) %>%
        arrange(cpp_dsg)
```

Frequency rules

```
# create dic rule n>=10 & prob <=90%

rule_10_90 <-
  df_household_survs %>%
  distinct(my_id, .keep_all = T) %>%
  select(cpp_dsg, cpp_cod3, my_id:prob) %>%
  filter(n_cpp_dsg >= 10, prob >= 89.5) %>%
  arrange(desc(n_cpp_dsg  ))
```

```
bind_rows(head(rule_10_90, n = 5),
          tail(rule_10_90, n = 5)) %>%
            knitr::kable()
```

cpp_dsg	cpp_cod3	my_id	n_cpp_dsg	n_my_id	prob
empregada domestica	911	911_empregada domestica	16112	16100	99.92552
empregada limpeza	911	911_empregada limpeza	15229	15129	99.34336
pedreiro	711	711_pedreiro	11732	11656	99.35220
cozinheira	512	512_cozinheira	8084	8078	99.92578
enfermeira	222	222_enfermeira	6065	6012	99.12613
vendedora numa loja	522	522_vendedora numa loja	10	10	100.00000
vigilancia criancas	531	531_vigilancia criancas	10	10	100.00000
vigilante portaria	541	541_vigilante portaria	10	9	90.00000
vigilante praia	541	541_vigilante praia	10	10	100.00000
vigilante recepcionista	541	541_vigilante recepcionista	10	9	90.00000

Frequency rules

The same word can be written in a multitude of variations:

- spelling errors
- (mis)use of abbreviation
- typo’s

String Distance

Levenshtein Distance

$$d_{lv}(s,t) = \begin{cases} 0 \text{ if } s = t = \varepsilon \\ \min\{ \\ \quad d_{lv}(s, t_{1:|t|-1}) + w_1, \\ \quad d_{lv}(s_{1:|s|-1}, t) + w_2, \\ \quad d_{lv}(s_{1:|s|-1}, t_{1:|t|-1}) + [1 - \delta(s_{|s|}, t_{|t|})] w_3 \\ \} \text{ otherwise.} \end{cases}$$

The **Optimal String Alignment distance** (method='osa') is like the Levenshtein distance but also allows **transposition of adjacent characters**.

Each substring may be edited only once.

Very demanding in terms of computation

- Substitution: ‘foo’ → ‘boo’
- Deletion: ‘foo’ → ‘oo’
- Insertion: ‘foo’ → ‘floo’
- **Transposition**: ‘foo’ → ‘ofo’

stringdist R package by M.P.J. van der Loo (2014)

Levenshtein Distance

$$d_{lv}(s,t) = \begin{cases} 0 \text{ if } s = t = \varepsilon \\ \min\{ \\ \quad d_{lv}(s, t_{1:|t|-1}) + w_1, \\ \quad d_{lv}(s_{1:|s|-1}, t) + w_2, \\ \quad d_{lv}(s_{1:|s|-1}, t_{1:|t|-1}) + [1 - \delta(s_{|s|}, t_{|t|})]w_3 \\ \} \text{ otherwise.} \end{cases}$$

Vladimir Levenshtein

- Stringdistance:
  - Soldado (*soldier*)
  - Soldador (*welder*)
- ^{d}lv = 1!

```
bind_rows(head(rule_10_90, n = 5),
          tail(rule_10_90, n = 5)) %>%
            knitr::kable()
```

cpp_dsg	cpp_cod3	my_id	n_cpp_dsg	n_my_id	prob
empregada domestica	911	911_empregada domestica	16112	16100	99.92552
empregada limpeza	911	911_empregada limpeza	15229	15129	99.34336
pedreiro	711	711_pedreiro	11732	11656	99.35220
cozinheira	512	512_cozinheira	8084	8078	99.92578
enfermeira	222	222_enfermeira	6065	6012	99.12613
vendedora numa loja	522	522_vendedora numa loja	10	10	100.00000
vigilancia criancas	531	531_vigilancia criancas	10	10	100.00000
vigilante portaria	541	541_vigilante portaria	10	9	90.00000
vigilante praia	541	541_vigilante praia	10	10	100.00000
vigilante recepcionista	541	541_vigilante recepcionista	10	9	90.00000

```
library(tidystringdist)

tidy_comb("enfermeira",
          df_household_survs$cpp_dsg[df_household_survs$cpp_cod3 == "222"]) %>%
        tidy_stringdist(method = "osa") %>%
        filter(osa <= 3) %>%
        arrange(osa) %>%
        distinct(V1) %>%
        knitr::kable()
```

enfermeira	enferemira	enfermeira	efermeiro	enfremeira
3nfermeira	enferimeira	3nfermeira	ehfermeiro	engermeira
efermeira	enfermaeira	efermeira	emfermeiro	infermeira
emfermeira	enfermaira	emfermeira	enefermeiro	nfermeira
emnfermeira	enfermeia	emnfermeira	enfemeiro	unfermeira
enefermeira	enfermeiar	enefermeira	enfemerira	e nfermeiro
enfeermeira	enfermeiira	enfeermeira	enfereima	efermeiro
enfeirmeira	enfermeir	enfeirmeira	enfereiro	ehfermeiro
enfemeira	enfermeiraq	enfemeira	enferemeiro	emfermeiro
enfemreira	enfermeiras	enfemreira	enferemiro	enefermeiro

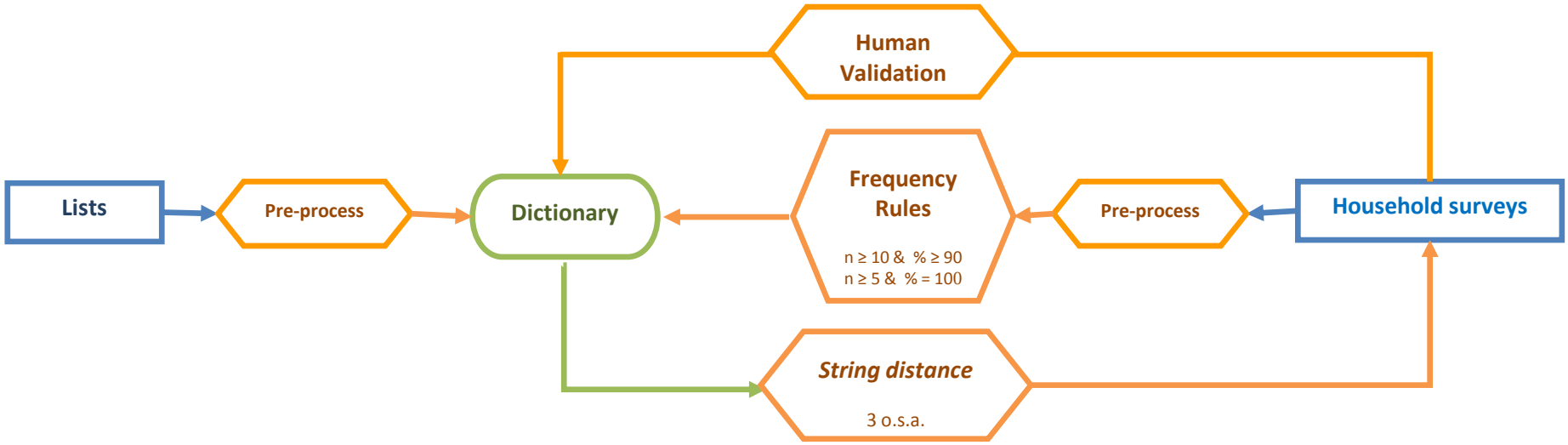
Let's see a wordcloud

```
# create a df: only cpp code "222

df_household_survs_cpp222 <- filter(df_household_survs, cpp_cod3 == "222")
```

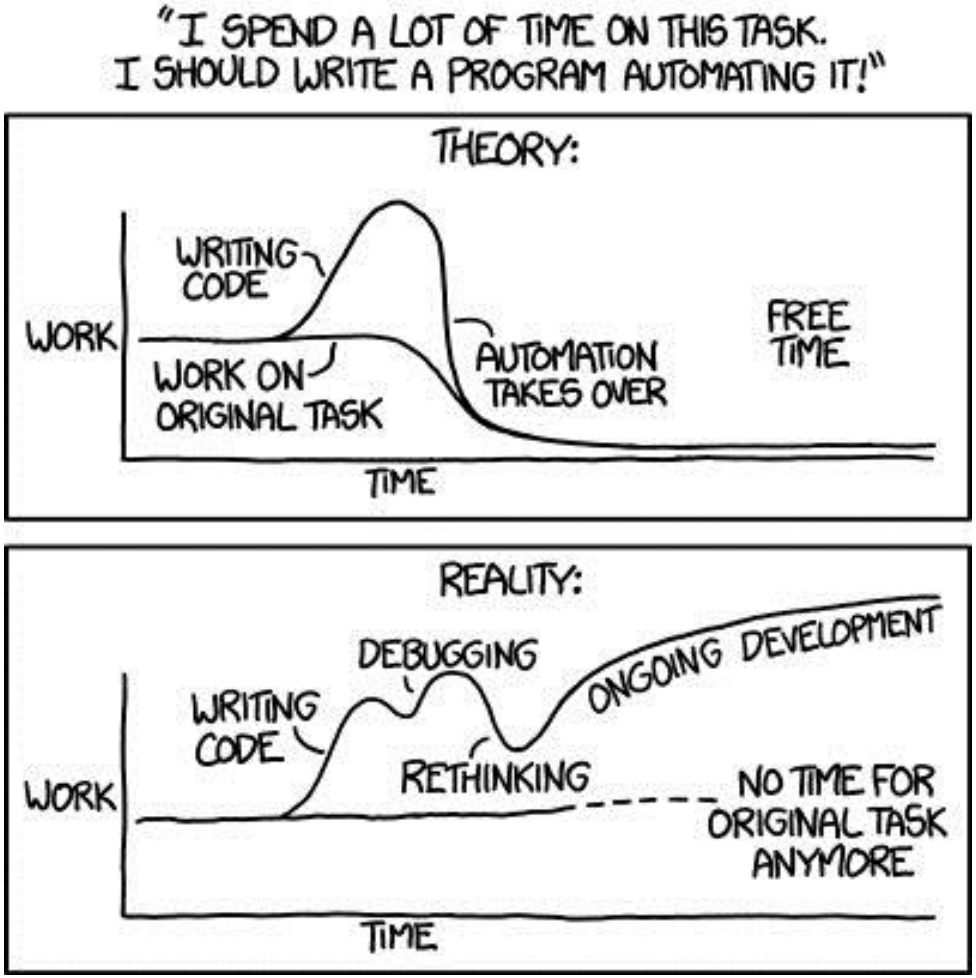
```
# results in a wordcloud

library(tidystringdist)

wc <- tidy_comb("enfermeira", df_household_survs_cpp222$cpp_dsg) %>%
        tidy_stringdist(method = "osa") %>%
        filter(osa <= 3)

wespal <- wesanderson::wes_palette(name = "Zissou1", 5, type = "continuous")
wordcloud::wordcloud(wc$V1, wc$osa, scale = c(2, .1), min.freq = 1, colors = wespal)
```

Enfermeira (*female nurse*)

Create and expand dictionaries



Motivation



XKCD

Ongoing development…

```
library(purrr)

# dataframe to add and / or to check
ok_to_add_check <- data.frame(
        cpp_dsg = c("1 escrituraria", "1 escriturario"),
        correct_cod3 = c("431", "431"))

# search dic for strings within 3osa , compare cod
  map2_df(list(ok_to_add_check$cpp_dsg),dic_cpp$cpp_dsg, tidy_comb) %>%
  tidy_stringdist(method = "osa") %>%
  filter(osa <= 3) %>%
  arrange(osa) %>%
  rename(cpp_dsg = V1) %>%
  left_join(dic_cpp) %>%
  left_join(ok_to_add_check, by = c("V2" = "cpp_dsg")) %>%
  filter(cpp_cod3 != correct_cod3) %>%
  head(n = 15) %>%
  knitr::kable(.)
```

More data to analyze…

cpp_dsg	V2	osa	cpp_cod3	cpp_fonte3	correct_cod3
1 escrituraria	1 escrituraria	0	411	coders	431
1 escriturario	1 escriturario	0	411	coders	431
1 escrituraria	1 escriturario	1	411	coders	431
1 escriturario	1 escrituraria	1	411	coders	431
1o escriturario	1 escriturario	1	411	coders	431
2 escriturario	1 escriturario	1	411	coders	431
3 escrituraria	1 escrituraria	1	411	coders	431
1escrituraria	1 escrituraria	1	411	coders	431
2 escrituraria	1 escrituraria	1	411	coders	431
3 escriturario	1 escriturario	1	411	coders	431

Algorithm

- Classify
  - Occupation and Economic Activity
  - Exact matching
  - 2 and 3 digits level
- Compare with manual coding (preliminary stage)
- Communicate results

Automation and Integration

- R Package: classify, plot and summary report
- RMarkdown: classify, plot extended report, output files
- Integrated in our management system: full automation

Packages

- tidyverse
- janitor
- stringi
- tidystringdist
- knitr
- qdap
- readxl
- rio
- wesanderson
- wordcloud

MONITOR PERFORMANCE AND VALIDATE RESULTS

Monitor performance and validate results

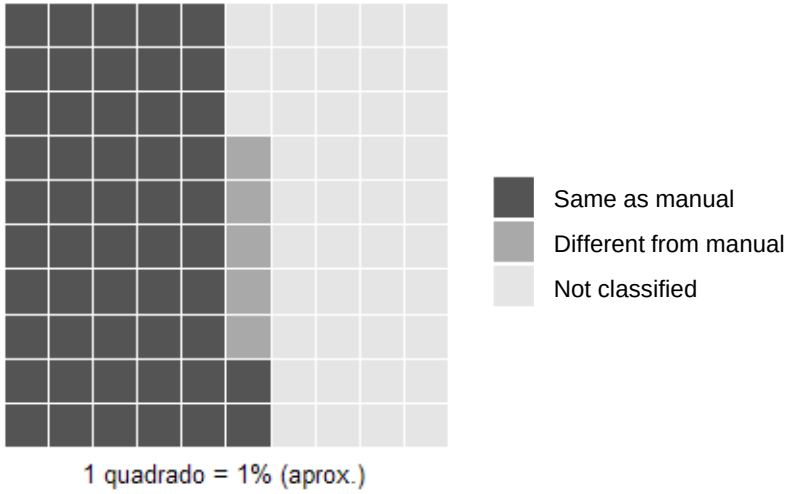
- Plot and report
- Rmarkdown report
- Validation:
  - AES automatic classification
  - Dictionaries

Report & Plot

Report

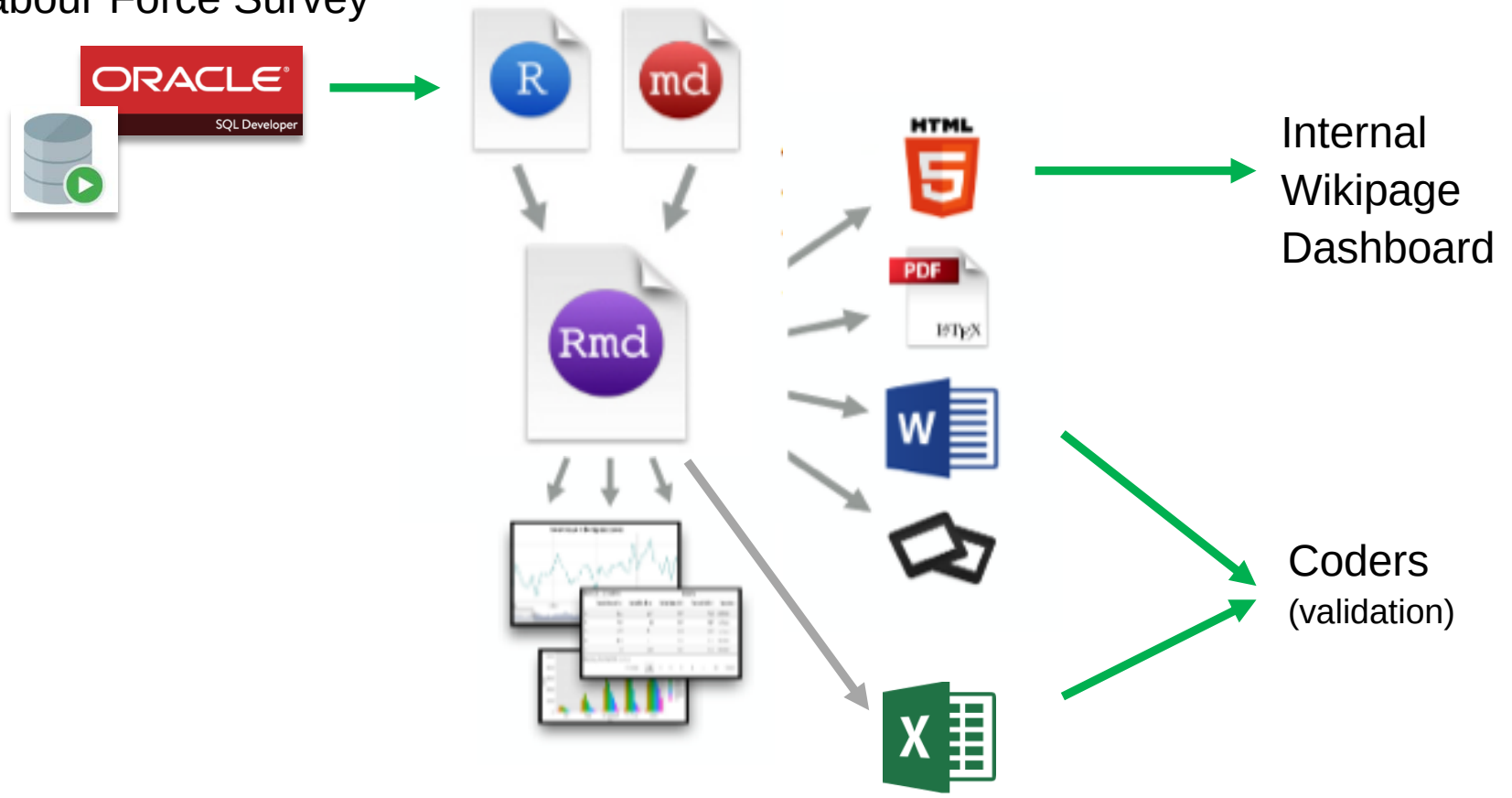
```
============================================================
  Summary Automatic Classification of Economic Activity
============================================================

  Total cases: 25607
  Classified cases: 14637 ( 57.16 % )
    *  13333 ( 52.07 % ) equal to manual classification
    *  1304 ( 5.09 % ) different from manual classification
  Not classified: 10970 ( 42.84 % )

------------------------------------------------------------------------------------------------------------
```

Plot

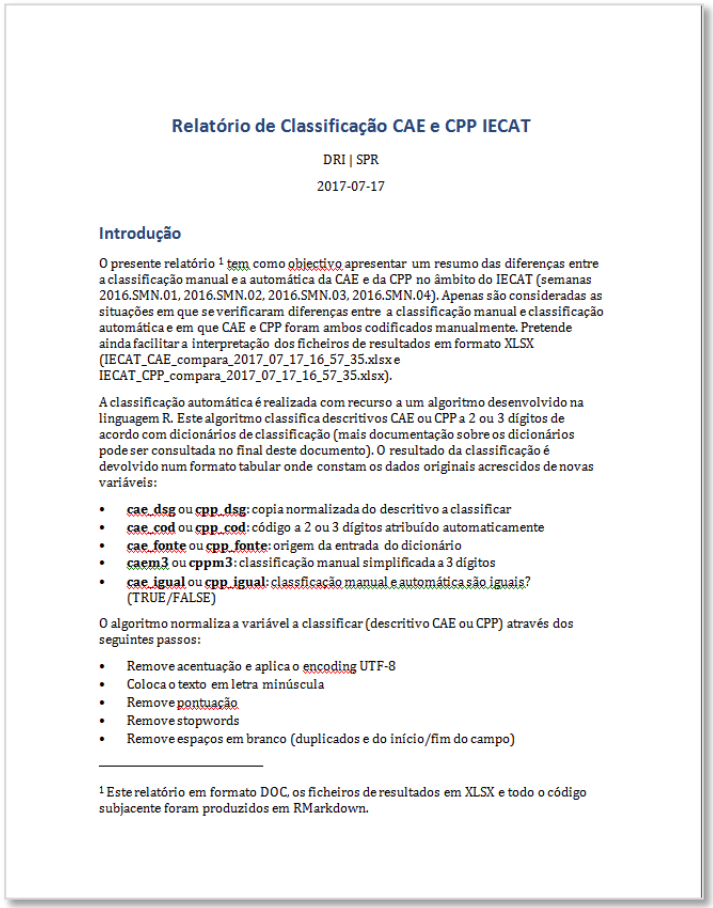


RMarkdown Report

Labour Force Survey



RMarkdown Report



Relatório de Classificação CAE e CPP IECAT

DRI | SPR

2017-07-17

Introdução

O presente relatório [1] tem como objectivo apresentar um resumo das diferenças entre a classificação manual e a automática da CAE e da CPP no âmbito do IECAT (semanas 2016.SMN.01, 2016.SMN.02, 2016.SMN.03, 2016.SMN.04). Apenas são consideradas as situações em que se verificaram diferenças entre a classificação manual e classificação automática e em que CAE e CPP foram ambos codificados manualmente. Pretende ainda facilitar a interpretação dos ficheiros de resultados em formato XLSX (IECAT_CAE_compara_2017_07_17_16_57_35.xlsx e IECAT_CPP_compara_2017_07_17_16_57_35.xlsx).

A classificação automática é realizada com recurso a um algoritmo desenvolvido na linguagem R. Este algoritmo classifica descritivos CAE ou CPP a 2 ou 3 dígitos de acordo com dicionários de classificação (mais documentação sobre os dicionários pode ser consultada no final deste documento). O resultado da classificação é devolvido num formato tabular onde constam os dados originais acrescidos de novas variáveis:

- **cae_dsg** ou **cpp_dsg**: copia normalizada do descritivo a classificar
- **cae_cod** ou **cpp_cod**: código a 2 ou 3 dígitos atribuído automaticamente
- **cae_fonte** ou **cpp_fonte**: origem da entrada do dicionário
- **caem3** ou **cppm3**: classificação manual simplificada a 3 dígitos
- **cae_igual** ou **cpp_igual**: classficação manual e automática são iguais? (TRUE/FALSE)

O algoritmo normaliza a variável a classificar (descritivo CAE ou CPP) através dos seguintes passos:

- Remove acentuação e aplica o encoding UTF-8
- Coloca o texto em letra minúscula
- Remove pontuação
- Remove stopwords
- Remove espaços em branco (duplicados e do início/fim do campo)

---

[1] Este relatório em formato DOC, os ficheiros de resultados em XLSX e todo o código subjacente foram produzidos em RMarkdown.

RMarkdown Report

Relatório de Classificação CAE

DRI | SPR

2017-07-17

Introdução

O presente relatório [1] tem como objectivo apresentar um
a classificação manual e a automática da CAE e da CPP no
2016.SMN.01, 2016.SMN.02, 2016.SMN.03, 2016.SMN.04
situações em que se verificaram diferenças entre a classi
automática e em que CAE e CPP foram ambos codificado
ainda facilitar a interpretação dos ficheiros de resultados
(IECAT_CAE_compara_2017_07_17_16_57_35.xlsx e
IECAT_CPP_compara_2017_07_17_16_57_35.xlsx).

A classificação automática é realizada com recurso a um
linguagem R. Este algoritmo classifica descritivos CAE ou
acordo com dicionários de classificação (mais document
pode ser consultada no final deste documento). O resulta
devolvido num formato tabular onde constam os dados c
variáveis:

- **cae_dsg** ou **cpp_dsg**: copia normalizada do descritiv
- **cae_cod** ou **cpp_cod**: código a 2 ou 3 dígitos atribuíc
- **cae_fonte** ou **cpp_fonte**: origem da entrada do dicio
- **caem3** ou **cppm3**: classificação manual simplificad
- **cae_igual** ou **cpp_igual**: classficação manual e autor
(TRUE/FALSE)

O algoritmo normaliza a variável a classificar (descritivo
seguintes passos:

- Remove acentuação e aplica o encoding UTF-8
- Coloca o texto em letra minúscula
- Remove pontuação
- Remove stopwords
- Remove espaços em branco (duplicados e do início/

[1] Este relatório em formato DOC, os ficheiros de resultad
subjacente foram produzidos em RMarkdown.

Análise dos resultados do classificador automático

Classificação CAE

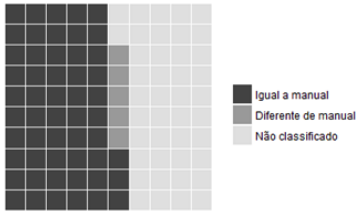
Os resultados detalhados da classificação automática podem ser consultados no ficheiro "IECAT_CAE_compara_2017_07_17_16_57_35.xlsx". De seguida apresenta-se um resumo.

Total de observações: 3751
Casos classificados 2166 (57.74%):

- 1979 (52.76%) igual a classificação manual
- 187 (4.99%) diferente da classificação manual

Sem classificação: 1585 (42.26%)

Gráfico "waffle" da classificação CAE



1 quadrado = 1% (aprox.)

Nas semanas de referência (2016.SMN.01, 2016.SMN.02, 2016.SMN.03, 2016.SMN.04) foram identificados 5 descritivos com as seguintes características: com classificação manual diferente da automática e com uma frequência igual ou superior a 4.

Na tabela em baixo podemos observar um resumo destes resultados.

INSTITUTO NACIONAL DE ESTATÍSTICA
STATISTICS PORTUGAL

uRosConf 2019

RMarkdown Report

Relatório de Classificação CAE

DRI | SPR

2017-07-17

Introdução

O presente relatório [1] tem como objectivo apresentar um
a classificação manual e a automática da CAE e da CPP no
2016.SMN.01, 2016.SMN.02, 2016.SMN.03, 2016.SMN.04
situações em que se verificaram diferenças entre a classi
automática e em que CAE e CPP foram ambos codificado
ainda facilitar a interpretação dos ficheiros de resultados
(IECAT_CAE_compara_2017_07_17_16_57_35.xlsx e
IECAT_CPP_compara_2017_07_17_16_57_35.xlsx).

A classificação automática é realizada com recurso a um
linguagem R. Este algoritmo classifica descritivos CAE ou
acordo com dicionários de classificação (mais document
pode ser consultada no final deste documento). O resulta
devolvido num formato tabular onde constam os dados c
variáveis:

- **cae_dsg** ou **cpp_dsg**: copia normalizada do descritiv
- **cae_cod** ou **cpp_cod**: código a 2 ou 3 dígitos atribuíd
- **cae_fonte** ou **cpp_fonte**: origem da entrada do dicio
- **caem3** ou **cppm3**: classificação manual simplificad
- **cae_igual** ou **cpp_igual**: classficação manual e autor
  (TRUE/FALSE)

O algoritmo normaliza a variável a classificar (descritivo
seguintes passos:

- Remove acentuação e aplica o encoding UTF-8
- Coloca o texto em letra minúscula
- Remove pontuação
- Remove stopwords
- Remove espaços em branco (duplicados e do início/

[1] Este relatório em formato DOC, os ficheiros de resultad
subjacente foram produzidos em RMarkdown.

Análise dos resultados do classificador automático

Classificação CAE

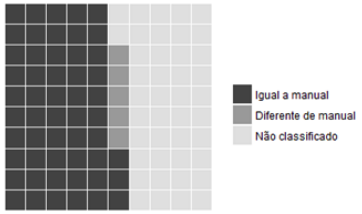
Os resultados detalhados da classificação automática podem ser consultados
ficheiro "IECAT_CAE_compara_2017_07_17_16_57_35.xlsx". De seguida apres
um resumo.

Total de observações: 3751
Casos classificados 2166 (57.74%):

- 1979 (52.76%) igual a classificação manual
- 187 (4.99%) diferente da classificação manual

Sem classificação: 1585 (42.26%)

Gráfico "waffle" da classificação CAE



1 quadrado = 1% (aprox.)

Nas semanas de referência (2016.SMN.01, 2016.SMN.02, 2016.SMN.03, 2016
foram identificados 5 descritivos com as seguintes características: com class
manual diferente da automática e com uma frequência igual ou superior a 4.

Na tabela em baixo podemos observar um resumo destes resultados.

Descritivos CAE com classificações manuais e automáticas distintas

cae_dsg	total	n_cods	n_reg
agropecuaria	14	1	1
agricultura subsistencia	5	1	2
clinica fisioterapia sem internamento	5	2	3
escola basica integrada	4	1	1
snack bar	4	1	2

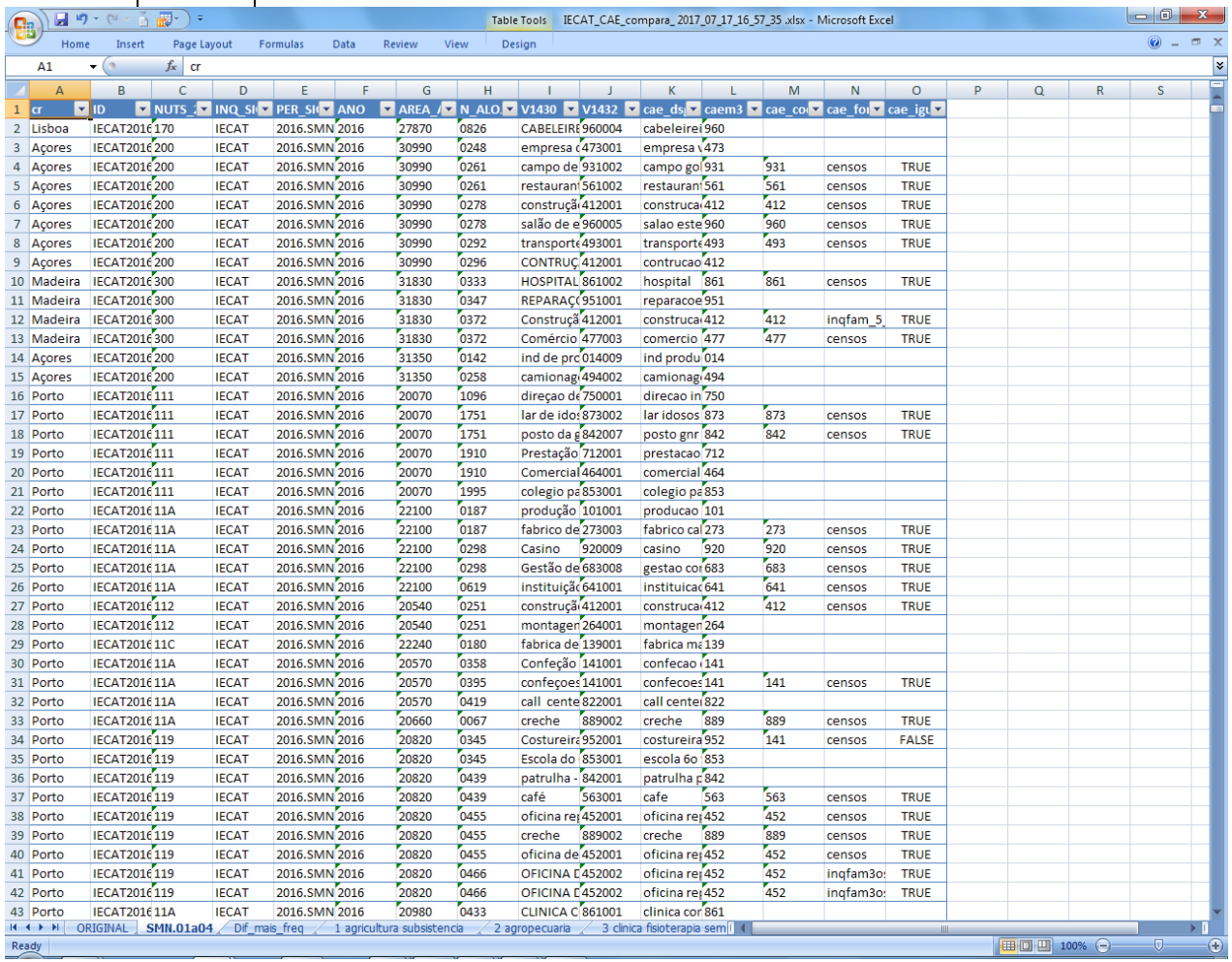
Legenda

- cae_dsg: descritivo normalizado
- total: número de vezes que o descritivo surge nas semanas em análise
- n_cods: número de diferentes códigos atribuídos manualmente ao mesmo descritivo
- n_reg: número de regiões que utilizaram código(s) diferente(s) do classificador automático

Esta tabela indica-nos, por exemplo, que o descritivo "agropecuaria" é utilizado 14 vezes nas semanas de referência, com 1 código(s) diferente(s) da classificação automática, sendo que esta situação ocorreu em 1 região(ões).

Outro exemplo, o descritivo "snack bar" é utilizado 4 vezes nas semanas de referência, com 1 código(s) diferente(s) da classificação automática, sendo que esta situação ocorreu em 2 região(ões).

RMarkdown Report

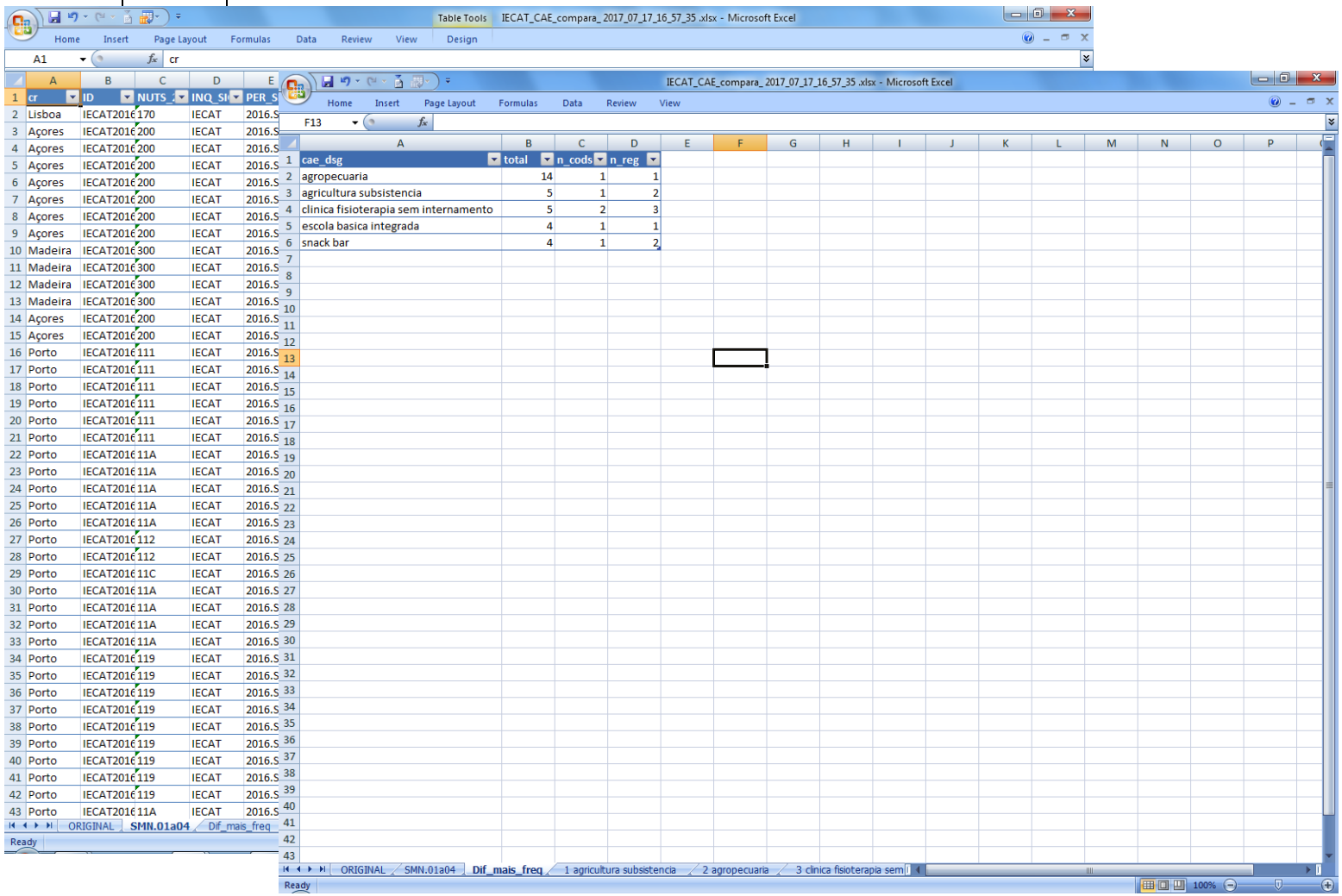


cr	ID	NUTS_2	INQ_SI	PER_SI	ANO	AREA_A	N_ALO	V1430	V1432	cae_ds	caem3	cae_co	cae_fo	cae_igu
Lisboa	IECAT2016	170	IECAT	2016.SMN	2016	27870	0826	CABELEIRE	960004	cabeleirei	960			
Açores	IECAT2016	200	IECAT	2016.SMN	2016	30990	0248	empresa c	473001	empresa v	473			
Açores	IECAT2016	200	IECAT	2016.SMN	2016	30990	0261	campo de	931002	campo go	931	931	censos	TRUE
Açores	IECAT2016	200	IECAT	2016.SMN	2016	30990	0261	restauran	561002	restauran	561	561	censos	TRUE
Açores	IECAT2016	200	IECAT	2016.SMN	2016	30990	0278	construçã	412001	construca	412	412	censos	TRUE
Açores	IECAT2016	200	IECAT	2016.SMN	2016	30990	0278	salão de e	960005	salao este	960	960	censos	TRUE
Açores	IECAT2016	200	IECAT	2016.SMN	2016	30990	0292	transporte	493001	transporte	493	493	censos	TRUE
Açores	IECAT2016	200	IECAT	2016.SMN	2016	30990	0296	CONTRUÇ	412001	contrucao	412			
Madeira	IECAT2016	300	IECAT	2016.SMN	2016	31830	0333	HOSPITAL	861002	hospital	861	861	censos	TRUE
Madeira	IECAT2016	300	IECAT	2016.SMN	2016	31830	0347	REPARAÇ	951001	reparacoe	951			
Madeira	IECAT2016	300	IECAT	2016.SMN	2016	31830	0372	Construçã	412001	construca	412	412	inqfam_5	TRUE
Madeira	IECAT2016	300	IECAT	2016.SMN	2016	31830	0372	Comércio	477003	comercio	477	477	censos	TRUE
Açores	IECAT2016	200	IECAT	2016.SMN	2016	31350	0142	ind de prc	014009	ind produ	014			
Açores	IECAT2016	200	IECAT	2016.SMN	2016	31350	0258	camionag	494002	camionag	494			
Porto	IECAT2016	111	IECAT	2016.SMN	2016	20070	1096	direçao de	750001	direcao in	750			
Porto	IECAT2016	111	IECAT	2016.SMN	2016	20070	1751	lar de idos	873002	lar idosos	873	873	censos	TRUE
Porto	IECAT2016	111	IECAT	2016.SMN	2016	20070	1751	posto da g	842007	posto gnr	842	842	censos	TRUE
Porto	IECAT2016	111	IECAT	2016.SMN	2016	20070	1910	Prestação	712001	prestacao	712			
Porto	IECAT2016	111	IECAT	2016.SMN	2016	20070	1910	Comercial	464001	comercial	464			
Porto	IECAT2016	111	IECAT	2016.SMN	2016	20070	1995	colegio pa	853001	colegio pa	853			
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	22100	0187	produção	101001	producao	101			
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	22100	0187	fabrico de	273003	fabrico ca	273	273	censos	TRUE
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	22100	0298	Casino	920009	casino	920	920	censos	TRUE
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	22100	0298	Gestão de	683008	gestao cor	683	683	censos	TRUE
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	22100	0619	instituiçã	641001	instituica	641	641	censos	TRUE
Porto	IECAT2016	112	IECAT	2016.SMN	2016	20540	0251	construçã	412001	construca	412	412	censos	TRUE
Porto	IECAT2016	112	IECAT	2016.SMN	2016	20540	0251	montagen	264001	montagen	264			
Porto	IECAT2016	11C	IECAT	2016.SMN	2016	22240	0180	fabrica de	139001	fabrica ma	139			
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	20570	0358	Confeção	141001	confecao	141			
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	20570	0395	confeçoes	141001	confecoes	141	141	censos	TRUE
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	20570	0419	call cente	822001	call cente	822			
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	20660	0067	creche	889002	creche	889	889	censos	TRUE
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0345	Costureira	952001	costureira	952	141	censos	FALSE
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0345	Escola do	853001	escola 6o	853			
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0439	patrulha -	842001	patrulha p	842			
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0439	café	563001	cafe	563	563	censos	TRUE
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0455	oficina re	452001	oficina re	452	452	censos	TRUE
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0455	creche	889002	creche	889	889	censos	TRUE
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0455	oficina de	452001	oficina re	452	452	censos	TRUE
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0466	OFICINA D	452002	oficina re	452	452	inqfam3os	TRUE
Porto	IECAT2016	119	IECAT	2016.SMN	2016	20820	0466	OFICINA D	452002	oficina re	452	452	inqfam3os	TRUE
Porto	IECAT2016	11A	IECAT	2016.SMN	2016	20980	0433	CLINICA C	861001	clinica cor	861			

INSTITUTO NACIONAL DE ESTATÍSTICA
STATISTICS PORTUGAL

uRosConf 2019

RMarkdown Report

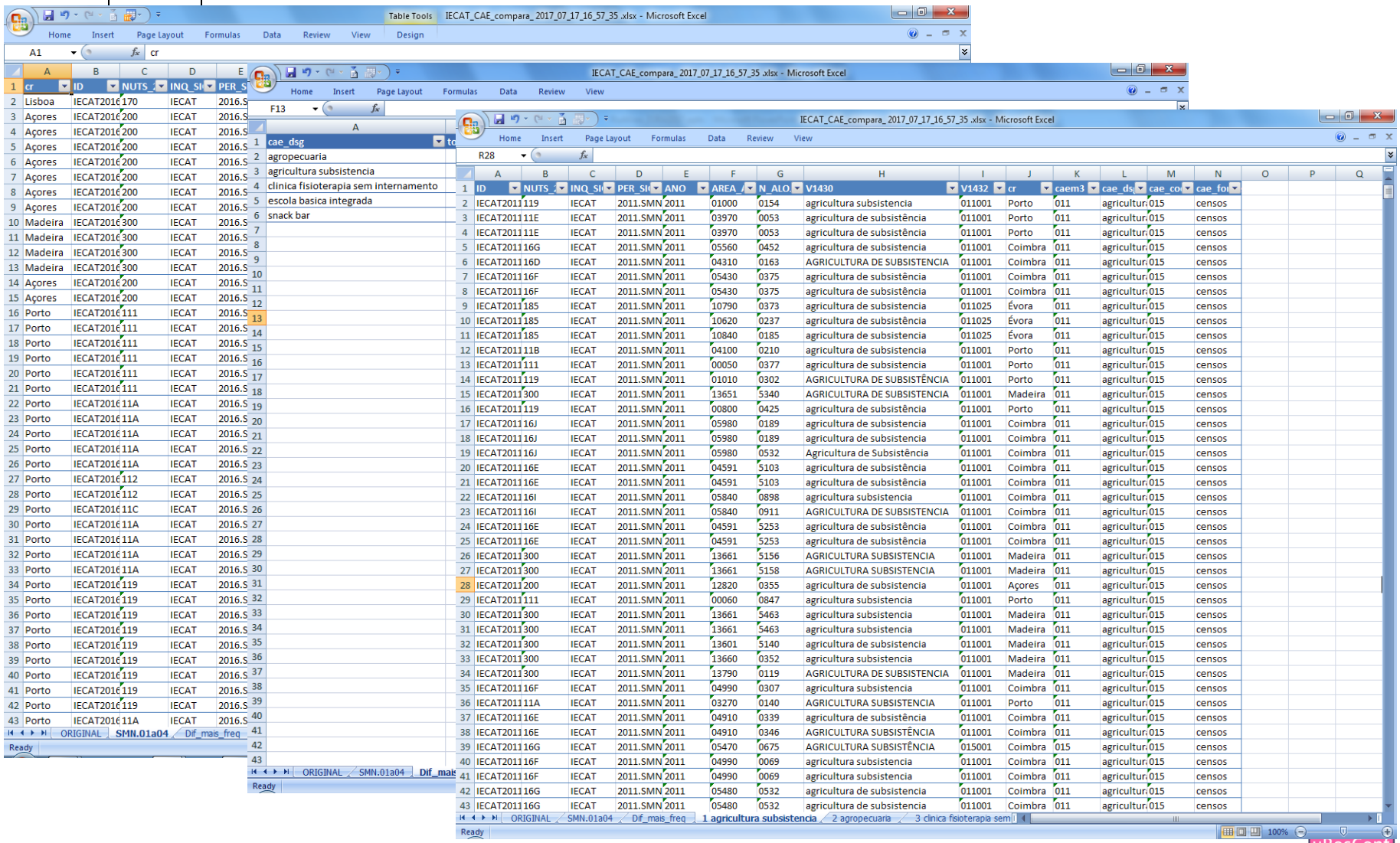


cae_dsg	total	n_cods	n_reg
agropecuaria	14	1	1
agricultura subsistencia	5	1	2
clinica fisioterapia sem internamento	5	2	3
escola basica integrada	4	1	1
snack bar	4	1	2

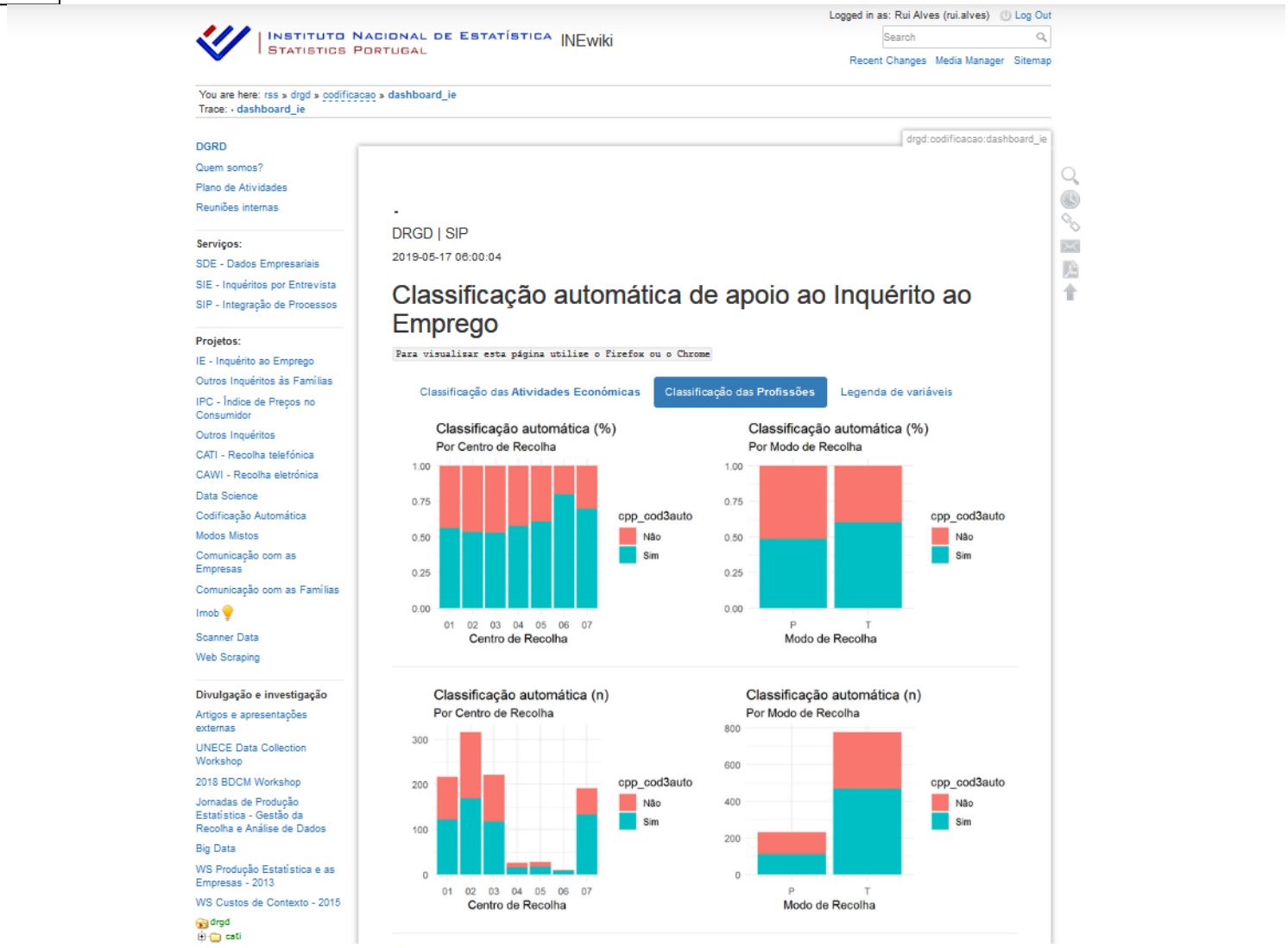
INSTITUTO NACIONAL DE ESTATÍSTICA
STATISTICS PORTUGAL

uRosConf 2019

RMarkdown Report



Wikipage dashboard



Monitor performance and validate results

- Adult Education Survey (2016):
  - Automatic classification thoroughly reviewed by the best team of coders
- Dictionaries
  - Occupation and Economic Activity Dictionaries reviewed by all the 7 regional teams of coders
- Data collection
  - Feedback from / to coders

Occupation
(3 digits)

```
dic_cpp %>%
  tabyl(cpp_fonte3) %>%
  knitr::kable()
```

cpp_fonte3	n	percent
3osa	16810	0.4048359
coders	1336	0.0321749
list	20213	0.4867905
rule	3164	0.0761987

Packages

- tidyverse
- formattable
- xda
- ROracle
- Openxlsx
- knitr
- waffle
- lubridate
- janitor
- png
- grid
- gridExtra
- RColorBrewer

SOME RESULTS

Information and Communication Technologies 2018 (CAWI)

```
left_join(sictu_cawi, dic_cpp, "cpp_dsg") %>%
        mutate(coda = !is.na(cpp_fonte3)) %>%
        tabyl(coda) %>% knitr::kable()
```

coda	n	percent
FALSE	402	0.3089931
TRUE	899	0.6910069

```
left_join(sictu_cawi, dic_cpp, "cpp_dsg") %>%
  tabyl(cpp_fonte3) %>%
  knitr::kable()
```

cpp_fonte3	n	percent	valid_percent
3osa	166	0.1275942	0.1846496
coders	110	0.0845503	0.1223582
list	166	0.1275942	0.1846496
rule	457	0.3512683	0.5083426
NA	402	0.3089931	NA

Limitations and shortcomings

- Ineffective with new strings
- Dictionary expansion depends on human validation
- Changes may occur with new interviewers and new coders
- Changes may occur when coders change their mind

Advantages and benefits

- Consistent
- Error management
- Speed
- Effective

NEXT STEPS

Next steps

- Every new [string + code] is added to dictionary
- spelling + stemming
- Different distance metrics for short and long strings
- n-grams

Next steps

- Special attention on specific cases
- Use auxiliary variables
- Provide an extensive and properly validated dataset for machine leaning
- uRos2020 8th international conference

rui.alves@ine.pt

THANK YOU FOR YOUR ATTENTION!