

Natality and Conflict Violence Reduction: A regression discontinuity design in Colombia - APPENDIX 1

Project by Manal Amin, Benoit Hayman and Marco Antonio Soto Novoa

In this appendix, we disclose the process of obtaining the data, cleaning the data and forming in panel data form.

0 - Obtaining the data

The data was obtained through the official statistical the office, the [National Administrative Department of Statistics](#).

The data is easily accessible in the page about birth.

Datos de nacimientos en Colombia

El DANE consolida, valida, procesa y difunde la información de nacimientos, a partir de los certificados diligenciados en medio físico o digital, por médicos, o por personal autorizado. Esta información se constituye en fuente básica para el cálculo de indicadores como la tasa bruta de natalidad y tasas de fecundidad.

A continuación, se reportan las características principales que identifican el hecho: edad, sexo, departamento y municipio de ocurrencia del hecho y de residencia de la madre, y que son registradas en el aplicativo RUAF-ND del Ministerio de Salud y Protección Social.



Conozca los resultados de nacimientos en Colombia a partir de cada año de referencia. Recuerde que debido a la actualización que realizó el DANE junto con el Ministerio de Salud y Protección Social, desde 2008 se implementaron de manera gradual en las instituciones de salud, la certificación de los nacimientos vía web, a través del Registro Único de Afiliados.

Para conocer los datos de nacimientos de periodos de referencia anteriores al 2008:

[Ingrese aquí](#)

Ingrese a la información

[2008](#)
[2009](#)
[2010](#)
[2011](#)
[2012](#)
[2013](#)
[2014](#)
[2015](#)

✓ 0s completed at 17:44



- **Cuadro 3.** Nacimientos por área y sexo, según departamento y municipio de residencia de la madre.
- **Cuadro 4.** Nacimientos por tipo de parto según departamento de ocurrencia y sitio del parto.
- **Cuadro 5.** Nacimientos por persona que atendió el parto según departamento, municipio de ocurrencia y sitio del parto.
- **Cuadro 6.** Nacimientos por peso al nacer, según departamento y área de residencia de la madre.
- **Cuadro 6a.** Nacimientos por peso al nacer, según departamento, municipio y área de residencia de la madre.
- **Cuadro 7.** Nacimientos por grupo de edad de la madre, según departamento y municipio de residencia de la madre.
- **Cuadro 7a.** Nacimientos por grupo de edad de la madre, según departamento de residencia de la madre.
- **Cuadro 8.** Nacimientos por tiempo de gestación, según departamento, municipio y área de residencia de la madre.
- **Cuadro 9.** Nacimientos por número de hijos nacidos vivos, según departamento y municipio de residencia de la madre.
- **Cuadro 10.** Nacimientos por tipo de parto, según departamento de residencia de la madre y multiplicidad del embarazo.
- **Cuadro 11.** Nacimientos por área y sexo, según departamento de residencia de la madre y pertinencia étnica del nacido vivo.
- **Cuadro 12.** Nacimientos por sitio de parto, según departamento, municipio de ocurrencia y régimen de seguridad social de la madre.

Cifras actualizadas a 30 de diciembre de 2010

This a very convinient way to structure the different datasets. We will focus on "Cuadro 3" which is "Births by area and sex, according to department and municipality of residence of the mother."

The functions we will use to quickly and automatically import the data are the following:

```
from bs4 import BeautifulSoup
import urllib.request

def get_links(page_to_scrape, criteria):
    parser = 'html.parser' # or 'lxml' (preferred) or 'html5lib', if installed
    list_links = []
    resp = urllib.request.urlopen(page_to_scrape)
    soup = BeautifulSoup(resp, parser, from_encoding=resp.info().get_param('charset'))
    for link in soup.find_all('a', href=True):
        if criteria in link['href']:
            new_link = "https://www.dane.gov.co"+link['href'] #predefined form for link
            list_links.append(new_link)
    return list_links

def links_year(first_year, last_year, criteria):
    year = first_year
    list_all_years = []
    while year <= last_year:
        page = "https://www.dane.gov.co/index.php/estadisticas-por-tema/salud/nacimient
        temp_list= get_links(page, criteria)
        list_all_years.extend(temp_list)
        year = year + 1
    return list_all_years
```

The first function

```
def links_year(first_year,last_year,criteria)
```

goes through pages of the statistical office for each year in a given range and applies the first function to each page. Again, the fact that we do not set a specific range allows us to quickly change the range of our study.

The links for the data are now obtained by simply running the second function:

```
criteria = "Cuadro3"  
the_good_list = links_year(2008,2017,criteria)  
the_good_list #we see we get lots of links are premiliminary and other are definit
```

```
['https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_08/  
'https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_09/  
'https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_10/  
'https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_11/  
'https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_12/  
'https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_13/  
'https://www.dane.gov.co/files/investigaciones/poblacion/2016/30-junio-2016/  
'https://www.dane.gov.co/files/investigaciones/poblacion/2017/30-junio-2017/  
'https://www.dane.gov.co/files/investigaciones/poblacion/2017/22-diciembre-2  
'https://www.dane.gov.co/files/investigaciones/poblacion/2018/21-diciembre-2  
'https://www.dane.gov.co/files/investigaciones/poblacion/2018/21-diciembre-2  
'https://www.dane.gov.co/files/investigaciones/poblacion/2018/21-diciembre-2  
'https://www.dane.gov.co/files/investigaciones/poblacion/2018/28-septiembre-  
'https://www.dane.gov.co/files/investigaciones/poblacion/2018/28-septiembre-  
'https://www.dane.gov.co/files/investigaciones/poblacion/2018/29-junio-2018/  
'https://www.dane.gov.co/files/investigaciones/poblacion/2018/29-junio-2018/  
'https://www.dane.gov.co/files/investigaciones/poblacion/2018/28-marzo-2018/  
'https://www.dane.gov.co/files/investigaciones/poblacion/2017/22-diciembre-2  
'https://www.dane.gov.co/files/investigaciones/poblacion/2017/28-septiembre-  
'https://www.dane.gov.co/files/investigaciones/poblacion/2017/30-junio-2017/
```

To obtained only the relevant links, we simply subset the list:

```
final_list = the_good_list[0:10]
```

the correct dataset (Cuadro 12)

```
import subprocess

def download_data(first_year, list_of_links):
    year = first_year
    for link_data in list_of_links:
        #print(link_data)
        file_name = "dev_data"+str(year)+".xls"
        #print(file_name)
        subprocess.call(["wget", "-O", file_name, link_data])
        year = year + 1

download_data(2008, final_list)
```

https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_08/Cu
https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_09/Cu
https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_10/Cu
https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_11/Cu
https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_12/Cu
https://www.dane.gov.co/files/investigaciones/poblacion/nacimientos/nac_13/Cu
<https://www.dane.gov.co/files/investigaciones/poblacion/2016/30-junio-2016/na>
<https://www.dane.gov.co/files/investigaciones/poblacion/2017/30-junio-2017/na>
<https://www.dane.gov.co/files/investigaciones/poblacion/2017/22-diciembre-201>
<https://www.dane.gov.co/files/investigaciones/poblacion/2018/21-diciembre-201>

We have now obtained our raw datasets, which can be found in this [github repository](#).

1 - Cleaning the data

05045 APARTADO	2'801	1'475	1'326	1'239	1'156	102	74
05051 ARBOLETES	494	270	224	116	104	54	34
05055 ARGELIA	148	71	77	24	26	0	0
05059 ARMENIA	56	27	29	4	5	8	10
05079 BARBOSA	629	344	285	144	142	45	39
05086 BELMIRA	91	47	44	20	14	5	5
05088 BELLO	5'597	2'909	2'688	2'819	2'583	21	14
05091 BETANIA	144	78	66	32	32	1	0
05093 BETULIA	262	143	119	38	44	17	6
05101 CIUDAD BOLÍVAR	505	274	231	165	149	22	23
05107 BRICEÑO	187	88	99	29	32	12	8
05113 BURITICA	123	66	57	8	7	12	11
05120 CÁCERES	674	381	293	162	113	143	111
05125 CAICEDO	155	76	79	16	9	6	13
05129 CALDAS	899	456	443	328	320	51	63
05134 CAMPAMENTO	145	80	65	11	16	6	3
05138 CAÑASGORDAS	190	95	95	33	31	9	18
05142 CARACOLÍ	57	30	27	23	13	1	3
05145 CARAMANTA	62	32	30	9	11	8	5

1.0 - Problems with the data

When importing the data into the python environment, we see that we are in trouble. For example, by running

```
pd.read_excel("dev_data2009.xls", sheet_name="17")
```

we get the following

E:	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	CUADRO 3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NACIMIENTOS POR ÁREA Y SEXO, SEGÚN DEPARTAMENT...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	AÑO 2009	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	Volver al índice	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	Caldas	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	Departamento y municipio de residencia de la m...	Total	Total	NaN	Cabecera municipal	NaN	Centro poblado	NaN	Rural disperso	NaN	Sin información	NaN
10	NaN	NaN	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres
11	Total	11483	5963	5520	4156	3826	575	578	1231	1108	1	8
12	17001 MANIZALES	4148	2150	1998	1989	1841	81	76	80	81	0	0
13	17013 AGUADAS	271	149	122	62	50	18	18	69	54	0	0
14	17042 ANSERMA	418	219	199	122	104	20	28	77	67	0	0
15	17050 ARANZAZU	143	68	75	43	49	1	1	24	25	0	0
16	17088 BELALCÁZAR	165	94	71	39	27	15	17	40	27	0	0
17	17174 CHINCHINÁ	781	407	374	331	298	26	20	50	56	0	0
18	17272 FILADELFIA	112	56	56	18	15	13	14	25	27	0	0
19	17380 LA DORADA	1327	700	627	618	560	46	35	36	29	0	3
20	17388 LA MERCED	61	28	33	5	8	7	13	16	11	0	1
21	17433 MANZANARES	248	136	112	64	58	4	9	68	45	0	0
22	17442 MARMATO	131	72	59	7	5	41	23	24	31	0	0

```
df2.head(30) #We see that only the first sheet got imported and without the xls f
```

	Unnamed: 0	Unnamed: 1
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	CUADRO 3	NaN
5	NACIMIENTOS POR ÁREA Y SEXO, SEGÚN DEPARTAMENT...	NaN
6	AÑO 2013	NaN
7	NaN	NaN
8	INDICE	NaN
9	Cod Depto	Departamento

```
Unnamed: 0      Cod Depto  
Unnamed: 1      Departamento  
Name: 9, dtype: object
```

```
data_code = df2[df2.index>8]  
data_code.reset_index(inplace=True, drop=True) #this drops the old indexing  
data_code.columns = ["Code","Department"] #we give proper names to our columns  
data_code.drop(index=0, inplace=True) #we drop the first row (that is now correctl  
data_code.reset_index(inplace=True, drop=True) #we index correctly once again  
data_code = data_code.dropna() #remove missing data  
data_code #the column code will be the list that will allows us to speed up the pr
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4174: SettingWith
```

23	73	Tolima
24	76	Valle del Cauca
25	81	Arauca
26	85	Casanare
