# READING PDF FILES REBECCA DUONG

- Extract the tables and the data from Pdf
- Find the country with the highest % of Child Marriage by age 15 and 18 (using the data from the textbook by Jacqueline Kazil, Katherine Jarmul)
- Python Libraries that are availble for reading PDF files:
  - pdfminer
  - pdftotext
  - tabula-py
  - PyPDF2
  - pdftables
- The library I will use for the tutorial:
  - PyPDF2
  - pdftables
  - matplotlib
  - pandas

### **TABLE 9** | CHILD PROTECTION

	Child labour (%)*			Child marriage (%)		Birth registration .	Female genital mutitation/cutting (%)* 2002–2012*			wife be	ation of ating (%)	Viol	ent discipline	%)*
Countries -		2005-2012*		2005-	2012*	(%)+	preva	femce	attitudes	2005-	-2012*		2005-2012*	
and areas	total	mate	female	married by 15	married by 18	2005-2012* total	women*	glirish	support for the practice <sup>c</sup>	male	female	total	male	female
Afghanistan	10	11	10	15	40	37	-	-	-	-	90	74	75	74
Albania	12	14	9	0	10	99	-	-	-	36	30	75	78	71
Algeria	5 y	6 y	4 y	0	2	99	-	-	-	-	68	88	89	87
Andorra	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Angola	24 x	22 x	25 x	-	-	36 x	-	-	-	-	-	-	-	-
Antigua and Barbuda	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Argentina	7 y	8 y	5 y	-	-	99 y	-	-	-	-	-	-	-	-
Armenia	4	5	3	0	7	100	-	-	-	20	9	70	72	67
Australia	_	_	_	_	_	100 v	_	-	-	_	_	_	_	_
Austria	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Azerbaijan	7 y	8 y	5 y	1	12	94	_	_	-	58	49	75	79	71
Bahamas	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bahrain	5 x	6 x	3 x	_	_	_			_		_			
Bangladesh	13	18	8	29	65	31	_	_	_	_	33 y	_	_	_
Barbados	-	-	-	_	-	-			_		- 00 /			
Belarus	1	1	2	0	3	100 y		_		4	4	65 y	67 y	62 y
Belgium			-	v	-	100 y			_	-	4	00 7	0, 1	02 9
Belize	6	7	5	3	26	95		_	_	_	9	71	71	70
Benin	46	47	45	8	34	80	13	2 y	1	14	47	/ /	/ /	70
Bhutan	3	3	3	6	26	100	13	2 y		14	68	_	_	
Bolivia (Plurinational	9	3	3	0	20	100	-	-	-	-	00	-	-	-
State of)	26 y	28 y	24 y	3	22	76 y	-	-	-	-	16	-	-	-
Bosnia and Herzegovina	5	7	4	0	4	100	-	-	-	6	5	55	60	50
Botswana	9 y	11 y	7 y	-	-	72	-	-	-	-	-	-	-	-
Brazil	9 y	11 y	6 y	11	36	93 y	-	-	-	-	-	-	-	-
Brunei Darussalam	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bulgaria	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Burkina Faso	39	42	36	10	52	77	76	13	9	34	44	83	84	82
Burundi	26	26	27	3	20	75	-	-	-	44	73	-	-	-
Cabo Verde	3 x,y	4 x,y	3 x,y	3	18	91	-	-	-	16 y	17	-	-	-
Cambodia	36 y	36 y	36 y	2	18	62	-	-	-	22 y	46 y	-	-	-
Cameroon	42	43	40	13	38	61	1	1 y	7	39	47	93	93	93
Canada	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Central African Republic	29	27	30	29	68	61	24	1	11	80 y	80	92	92	92
Chad	26	25	28	29	68	16	44	18 y	38	-	62	84	85	84
Chile	3 x	3 x	2 x	-	_	100 y	-	-	-	_	-	_	-	_
China		-	-	-	-	-	-	-	-	-	-	-	-	-
Colombia	13 y	17 y	9 y	6	23	97		_	_		_	_	_	
Comoros	27 x	26 x	28 x	_	_	88 x	_	_		_	_	_	_	_
Congo	25	24	25 X	7	33	91	_	_	_	_	76	_	_	
Congo	25	24	25	,	33	91	_		-		70	_		

# USING PDFTABLES

```
from pdftables import get tables
import pprint
#The get tables function return each pages as its own table
#Each of those tables has a list ofrows with a contained list of columns
all_tables = get_tables(open('EN-FINAL Table 9.pdf', 'rb'))
#Tying to find the titles we can use for our columns
#Here we are looking at the first page's first 6 rows
print all tables[0][:6]
#NOTE: we can see that the titles are included in the first 3 list and they are messy
# Manually settting up our titles
#So we add all of our headers, including the country names, to one list
headers = ['Country', 'Child Labor 2005-2012 (%) total',
        'Child Labor 2005-2012 (%) male',
'Child Labor 2005-2012 (%) female',
'Child Marriage 2005-2012 (%) married by 15',
        'Child Marriage 2005-2012 (%) married by 18'.
        'Birth registration 2005-2012 (%)'.
        'Female Genital mutilation 2002-2012 (prevalence), women',
        'Female Genital mutilation 2002-2012 (prevalence), girls',
        'Female Genital mutilation 2002-2012 (support)',
        'Justification of wife beating 2005-2012' (%) male',
        'Justification of wife beating 2005-2012 (%) female',
        "Violent discipline 2005-2012 (%) total",
        Violent discipline 2005-2012 (%) male',
        Violent discipline 2005-2012 (%) female'l
```



[u'Afghanistan', u'10', u'11', u'10', u'15', u'40', u'37', u'\u2013', u\u2013', u\u2013', u\u2013', u\u2013', u'90', u'74', u'75', u [u'Albania', u'12', u'14', u'9', u'0', u'10', u'99', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'36', u'30', u'75', u'78', u'71'] [u'Algeria', u'5 y', u'6 y', u'4 y', u'0', u'2', u'99', u'\u2013', u\u2013', [u'Andorra', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'\u2013', [u'Angola', u'24 x', u'22 x', u'25 x', u'\u2013', u\u2013', u'\u2013', u'\u2013', u\u2013', u\u2 [u'Antigua and Barbuda', u'\u2013', u'\u2013', u\u2013', [u'Argentina', u'7 y', u'8 y', u'5 y', u\u2013', u\u2013 [u'Armenia', u'4', u'5', u'3', u'0', u'7', u'100', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'0', u'7', u'70', u'72', u'67'] [u'Australia', u'\u2013', u'\u201 [u'Austria', u'\u2013', u'\u2013' [u'Azerbaijan', u'7 y', u'8 y', u'5 y', u'1', u'12', u'94', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'49', u'75', u'79', u'71'] [u'Bahamas', u'\u2013', u'\u2013' [u'Bahrain', u'5 x', u'6 x', u'3 x', u'\u2013', u\u2013', u\u2013' [u'Bangladesh', u'13', u'18', u'8', u'29', u'65', u'31', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'\u2013' [u'Barbados', u'\u2013', u'\u2013 [u'Belarus', u'1', u'1', u'2', u'0', u'3', u'100 y', u'\u2013', u\u2013', u\u2013', u\u2013', u'4', u'4', u'65 y', u'67 y', u'62 y'] [u'Belgium', u'\u2013', u'\u2013' [u'Belize', u'6', u'7', u'5', u'3', u'26', u'95', u'\u2013', u'\u2 [u'Benin', u'46', u'47', u'45', u'8', u'34', u'80', u'13', u'2 γ', u'1', u'14', u'47', u'\u2013', u\u2013', u\u2013'] [u'Bhutan', u'3', u'3', u'3', u'6', u'26', u'100', u'\u2013', u\u2013', u\u2013', u\u2013', u\u2013', u\u2013' [u'Bolivia (Plurinational Bolivia (Plurinational State of)', u'26 y', u'28 y', u'24 y', u'3', u'22', u'76 y', u'\u2013', u'\u20 [u'Bosnia and Herzegovina', u'5', u'7', u'4', u'0', u'4', u'100', u'\u2013', u'\u2013', u'\u2013', u\u2013', u'6', u'5', u'55', u'60', [u'Botswana', u'9 y', u'11 y', u'7 y', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u\u2013', u\ [u'Brazil', u'9 y', u'11 y', u'6 y', u'11', u'36', u'93 y', u'\u2013', u\u2013', u\u2013', u\u2013', u\u2013', u\u2013', u\u2013', u\u2013', u\u2013',

[u'Burkina Faso', u'39', u'42', u'36', u'10', u'52', u'77', u'76', u'13', u'9', u'34', u'44', u'83', u'84', u'82']

[u'Cameroon', u'42', u'43', u'40', u'13', u'38', u'61', u'1', u'1 y', u'7', u'39', u'47', u'93', u'93', u'93']

[u'Burundi', u'26', u'26', u'27', u'3', u'20', u'75', u\u2013', u\u2013', u\u2013', u\u2013', u\u2013', u\u2013']

[u'Cabo Verde', u'3 x,y', u'4 x,y', u'3 x,y', u'3', u'18', u'91', u'\u2013', u'\u2013', u'\u2013', u'\u2013', u'16 y', u'17', u\u2013', u\u2013', u\u2013'] [u'Cambodia', u'36 y', u'36 y', u'36 y', u'21, u'18', u'62', u'\u2013', u'\u2013'

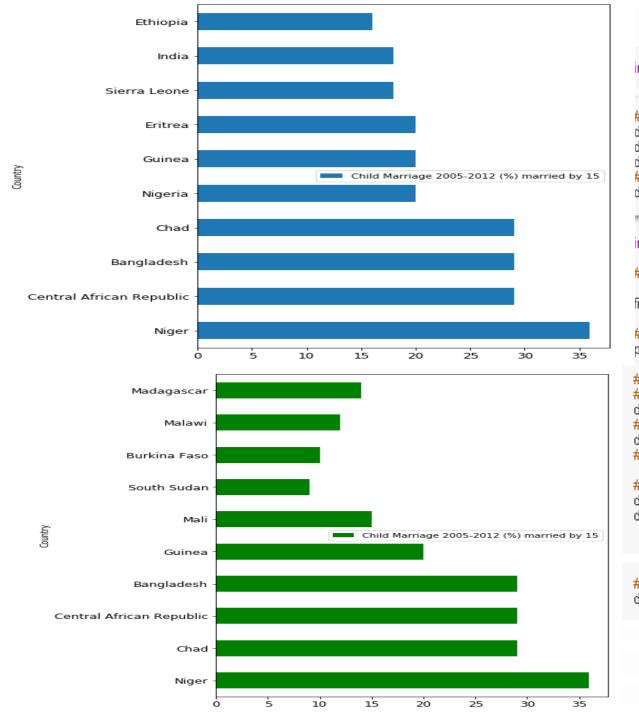
[u'Canada', u'\u2013', u'\u2013',

```
#extract the data values from the pdf
                                                                                                                                                                                                                                                                                                                                                                first_name = False
                                                                                                                                                                                                                                                                                                                                                                final data = []
                                                                                                                                                                                                                                                                                                                                                                for table in all tables:
                                                                                                                                                                                                                                                                                                                                                                             #isolates only the rows for each page we want; slice from the 5th index onward
                                                                                                                                                                                                                                                                                                                                                                             for row in table[5:]:
                                                                                                                                                                                                                                                                                                                                                                                         # if data row is missing index 0, it has no country name and is a blank row
                                                                                                                                                                                                                                                                                                                                                                                        #skip this row using continue
                                                                                                                                                                                                                                                                                                                                                                                         if row[0] == " or row[0][0].isdigit():
                                                                                                                                                                                                                                                                                                                                                                                                      continue
                                                                                                                                                                                                                                                                                                                                                                                        #if the data row is missing index 2, we know this is probably the first part of a country name
                                                                                                                                                                                                                                                                                                                                                                                         #We saves the first part of the name in a variable first name
                                                                                                                                                                                                                                                                                                                                                                                         elif row[2] == ":
                                                                                                                                                                                                                                                                                                                                                                                                     first name = row[0]
                                                                                                                                                                                                                                                                                                                                                                                                     continue
                                                                                                                                                                                                                                                                                                                                                                                        # Manipulate the country name entry in the row if it has a first name
                                                                                                                                                                                                                                                                                                                                                                                         #Put the second part of a country name back together with the first name
                                                                                                                                                                                                                                                                                                                                                                                         if first_name:
                                                                                                                                                                                                                                                                                                                                                                                                     row[0] = u{} {}.format(first_name, row[0])
                                                                                                                                                                                                                                                                                                                                                                                                     #set first_name back to False, so our nexxt iteration operates properly
                                                                                                                                                                                                                                                                                                                                                                                                    first_name = False
                                                                                                                                                                                                                                                                                                                                                                                         #Before cleaning up the messy data...
                                                                                                                                                                                                                                                                                                                                                                                        #Note there are in unicode
                                                                                                                                                                                                                                                                                                                                                                                         # print(row)
[u'Brunei Darussalam', u'\u2013', u'\u2013',
[uˈBulgaria', uˈ\u2013', u·\u2013', u·\u2013
```



```
clean row = []
# clean the data value in each row
# strip away the non-numeric letter next to the numeric data in each column such as 'x', 'y', 'V', or x,y'
#also change the unicode representation of "into an empty string
for col in range(0, len(row)):
 #Translate the value from unicode to ascii and do not need to transalte the non-ASCII chars
 # store the data values into the variable clean row
 # https://stackoverflow.com/questions/2365411/convert-unicode-to-ascii-without-errors-in-python/35536228
 clean_row.append(row[col].encode('ascii', 'ignore'))
 #Then clean up the data values
 # strip away the non-numeric letter next to the numeric value
 clean row[col] = clean row[col].strip('x')
 clean_row[col] = clean_row[col].strip('y')
 clean_row[col] = clean_row[col].strip(V)
 clean row[col] = clean row[col].strip('x,y')
 # convert the numeric data for each row to integer
 if clean_row[col].isdigit():
    clean row[col] = eval(clean row[col])
    # print(type(clean_row[col]))
 elif clean row[col] == ' ':
    clean_row[col] = None
# print(clean_row)
#append all the cleaned row into the list final data
final data.append(clean row)
#Exit the loop when you reach the last data row
if row[0] == 'Zimbabwe':
   break
```

# prints out a list of each row in the table pprint.pprint(final\_data)



```
import pandas as pd
```

```
# convert the list of dict into a dataframe and set the 'Country' column as the index

df = pd.DataFrame(final_data, columns = headers)

df = df.set_index('Country')

df = df.apply(pd.to_numeric)

# df.columns

df
```

import matplotlib.pyplot as plt

#Find the country with the highest % of child marriage in 2005-2012 of age 15 and 18

```
fig. axes = plt.subplots(nrows = 1, ncols = 2)
```

# adjust the space between the 2 plots; set it a little more farther apart plt.subplots adjust(wspace = 1)

#Sort the dataframe according to its % of child marriage by age 15 and 18; the highest % will be at the store these into a new variable  $df_1 = df_1 + df_2 = df_1 + df_2 = df_2 + df_3 = df_3 + df_4 = df_4 + df_4 + df_4 = df_4 + df_4$ 

#Looking at the data, it seems like Niger the highest % of Child Marriage in 2005-2012 for both age 1: df\_1.head(10).plot(ax = axes[0], kind = 'barh', y = ['Child Marriage 2005-2012 (%) married by 15'], font: df\_2.head(10).plot(color = 'g', ax = axes[1], kind = 'barh', y = ['Child Marriage 2005-2012 (%) married b

#Convert the dataframe to csv df.to\_csv('table\_9.csv')

## USING PYPDF2 LIBRARY

	(1)	*(4º) ruodal bli			(20) again	dhi8 nobutziger		2002-2012*		thealtiteut. itsed aliw	(%) Bu	ViolV	(%) enilqiseib tr	-10
- seint		*S105-2005			3015+	*2005-2012* (%)*	prevail	-	sobutitie odi tol hoqque	S005-20			*2105-2012*	
sean	later	alsm	alemat		87 yd bainem	latet	"namaw	shig.	practice	alem	alsmat	later A.T.	alem	olemol
neteinen	OT 2T	*L	6 0L	0 SL	0t	66 £E				98	30 06	9.L	9 <i>L</i>	17
ein	Λg	Λg	٧.	0	Z	66		-	-	-	89	88	68	78
6110	4 -	4.0	4.			v 00 f	-	-	-	-	-	-	-	-
elo	X #Z	× 22	x gz	-	-	× 96	-	-	-	-	-	-	-	-
ebuched bne eug	-	-	-	-	-	-	-	-	-	-	-	-	-	-
entine	A L	γ8	Λg	-	-	γ 66	-	-	-	-	-	-	-	-
eina	*	9	3	0	L	100	-	-	-	50	6	0.4	72	<i>L</i> 9
ellet	-	-	-	-	-	V 00 F	-	-	-	-	-	-	-	-
gitt	-	-	-	-	-	V 00 f	-	-	-	-	-	-	-	-
nelied	A L	γ8	Λg	L	ZL	16	-	-	-	89	69	94	64	1.4
sewe	-	-	-	-	-	-	-	-	-	-	-	-	-	-
nier	x g	x 9	×ε	-	-	-	-	-	-	-	-	-	-	-
ysapej8	13	81	8	6Z	99	31	-	-	-	-	33 y	-	-	-
sopeo	-	-	-	-	-	-	-	-	-	-	-	-	-	-
snu	L	L	2	0	3	V 001	-	-	-	7	7	۹ چو	Λ <i>L</i> 9	V 28
wnit	-	-	-	-	-	V 00 F	-	-	-	-	-	-	-	-
82	9	L	9	3	92	96	-	-	-	-	6	LL	LL.	04
uj	97	LV	97	8	34	08	13	5 A	L.	ÞΙ	LV	-	-	-
net	3	3	3	9	92	001	-	-	-	-	89	-	-	-
lenoitenihul (Plurinational	A 9Z	Y 82	74 7	3	ZZ	YaT	-	-	-	-	91	-	-	-
enivogassaH bne ein	9	L	7	0	7	001	-	-	-	9	9	99	09	09
euews	۸ 6	V II	ΛL	-	-	7.5	-	-	-	-	-	-	-	-
19	۸ 6	V II	Λg	II.	98	V 56	-	-	-	-	-	-	-	-
melessmed lan	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ejset	-	-	-	-	-	V 001	-	-	-	-	-	-	-	-
ose3 enti	33	45	98	OL	29	LL	94	13	6	34	77	83	18	85
ipun	92	97	ZZ	3	SO	94	-	-	-	77	73	-	-	-
o Verde	A'x g	A'x y	A'x g	8	81	16	-	-	-	۱6 ۸	41	-	-	-
eipoqu	A 98	36 y	¥ 98	Z	81	29	-	-	-	22 A	Á 97	-	-	-
neroon	45	43	09	εL	38	19	L	A L	L	68	LV	88	86	88
ере	-	-	-	-	-	V 00f	-	-	-	-	=	-	-	-
bilduqaR nephtA lett	62	LZ	30	6Z	89	L9	24	L	11	V 08	08	26	76	76
p	92	97	8Z	6Z	89	91	77	Y SF	38	-	79	78	98	48
a	×ε	×ε	x z	-	-	V 001	-	_	-	-	-	-	-	_
eu	-	-	-	-	-	=	-	-	-	-	-	-	-	-
ejquo	13 A	A 41	V 95	9	23	46		_	-		-			
90100	× LZ	74 78 ×	SS x		-	x 88	-	-	-	-	94	-	-	-

```
#A function for rotating the page
def PDFrotate(origFileName, newFileName, rotation):
   # creating a pdf File object of original pdf
   pdfFileObj = open(origFileName, 'rb')
   # creating a pdf Reader object
   pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
   # creating a pdf writer object for new pdf
   #rotated pages will be written to a new pdf
   pdfWriter = PyPDF2.PdfFileWriter()
   #iteriate through each page of original pdf
   for page in range(pdfReader.numPages):
     # creating rotated page object
     pageObj = pdfReader.getPage(page)
     # now we rotate the page by rotateClockwise() method
     pageObj.rotateClockwise(rotation)
     #adding rotated page object to pdf writer
     pdfWriter.addPage(pageÓbj)
   # open new pdf file object
   newFile = open(newFileName, wb)
   #writing rotated pages to new file using write() method
   pdfWriter.write(newFile)
   # closing the original pdf file object
   pdfFileObj.close()
   # closing the new pdf file object
   newFile.close()
```

			•••											
Wed Name	7	7	7	1		86	-	-	-	-	36	74	76	71
Ternes	23	21	26	12	32	TTy	117	19	47 6/9	-	-	99	99	99
Zambia	41 y	42 y	40 y		62	16	-	-	-	48	62	-	-	-
Zerkelsen	-	_	-		30	49	-	-	-	316	-	-	-	-

### DUMMARY NOCHTON

Sub-Saharan Africa	27	37	26	13	39	44	40	127	32	-	84	-	-	-
Southern Africa	27	28	211	10	38	38	66	10	21	38	14			_
West and Central Africa	26	26	27	16	41	67	32	16	21	-	56	80	81	90
Middle East and North Africa		11	7	3	18	87	-		-	-	-	20	80	
South Asia	12	13	12	18	48	39	-	-	-	60	62	-	-	-
East Asia and Pacific	E	8	8	2	16 ***	-	-	-	-	-	31 ***	-	-	-
Latin America and Caribbean	11	13		7	29	82	-	-	-	-	-	-	-	-
CHI/CH			4	1	10	88	-	-	-	-	20	-	-	-
Least developed countries	23	28	32	10	48	28	-	-	-	-	62	-	-	-
World	75 ***	16 ***	16 ***	11 ***	36 ***	65**	-	-	-	-	66 ***	-	-	-

<sup>\*</sup>An a consider third contributed grown in the policy, substance and province extension, and see 20.

### NAME OF THE WORLSON

Child Malesse — Proceedings of dichilers 12–15 pass, dell'emission in child inhone air the time of the survey, child its consciolated in the invasional child linkes and the following conditions; (a) dichilers 12–11 pass, bill ofter, diching the orderenear asset, dichi di houd one hour of assessment and rings or all bases 22 linears of manufacili dichines, or (a) children 12–12 pass, and other, during the orderenear asset, dichi di houri 14 hours of moneta malitici e al houri 22 linears and in manufacilions.

Orbit markings — Percentage of common 20-21 parts old who cares first marked or in order before they cares 10 parts old and presentage of common 20-21 parts old who cares first marked or in order before the common 20-21 parts old who cares first marked or in order before

Both registration: - Proceedings of children less than 5 pass, all also were registered at the time of the savery. The community of this includes includes abbilious advantability and facility was some by the inter-

Francis postiled multilation/swifing PEMIC 1—(i) Warmer processing or described TV-Glyces of sinks have undergone (TSUC, (b) gide processings of gide 3—13 years obtained have undergone (TSUC) (as equal-title) that meltion(), (b) suggest for the possition processings of account TV-Glyces of stabilizer.

And a state of well become "Providings of content and their 12-10 gives an attraction of state of the best position of contents of their and best or an of the specified responsible states, i.e., This subfoliance for local, a gaps and his in, gaps an attack of the subfoliance for local gaps and his in, and an attack of the subfoliance for local gaps and his interpretation of subfoliance for local gaps and the subfoliance of subfoliance for local gaps and the subfoliance of the subfoli

### MANAGEMENT AND REAL

Odd bloor - Maligin Indiator States Surveys (MCS), Servey optioned Health Surveys (SHS) and other national surveys

Oild markeys - MCI, Diffusionise rational surerys.

Bell registration - MCI, DKI, other national household surveys, sensors and stad registration system

Jacilland and wife leading - MEX DHE and other national surveys.

ent Building - MEX. DIS. and other sectional suppose.

### WOTE

- Date red accident
- Dalmatin of 100% were assumed place that stall registration options in those countries are complete and all stall recents (including pinting) are requirement. Transmit Dalmatin, Department of Transmits and Trains, Stall stalls, Papulation and Trial Trainsies Superior, Department of Transmits and Trainsies Superior, Department of Transmits and Trainsies Superior, Department of Transmits and Trainsies Superior, Department of Trainsies, Department of
- Sale rate to pray or periods ofter than those specified in the salaron healing. Sold data are not included in the salarities of regional and global energies.
- y Zuka differ from the standard definition or order to only part of accountry. If they full within the roted reference partial, such data are included in the substation of regional analysis of accountry.
- A more detailed explanation of the methodology, and the changes invadualising these extinsions can be found in the Control Nations the Date, pa
- Data refer to the most reservi year auxiliable sturing the period specified in the column has
- "Terlain D

### DISTRICT NEW YORK STREET, STRE

-	-	-	96	-	-		-	16	E.E.	4	46	10	96	energy.
-	-	-	-	-	-	-	-	* 88	-	-	28 PK	* 90	F 4E	Semantical Communication of the Communication of th
-	-	-	-	-	-		-	48	23		Ag	4.55	A EL	Estimated
-	-	-	-	-	-	-	-	-	-	-	-	-	-	PHINO.
-	-		-	-	-	-	-	A 000	-	-	2.5	* E	F E	490
78	100	18	E39	-	HE.	Am	11	PL.	10	60	HE	96	90	PPP(C)
68	28	28	OIE	A ge	14		10	10	10	60	30	400	60	Central African Republic
-	-	-	-	-	-	-	-	A 001	-		-	-	-	Section 2
110	28	28	4.9	339	£.	A 1	L	19	HE	Di-	07	137		Camming
-	-		A pp	A 22	-	-	-	E3	111		A se	A DE	A DE	Cambridge
			41.	A ML	-			18	11.	E	Alte E	Alway	Fig.	Calcul Vendo
-	-	-	EE	22	-	-	-	160	30	E	AE.	38	90	(purel)
EN	111	100	117	310		Di.	96	44	21	96	HE.		600	many eventual
-	-		-		-	-	-	4.00%			-	-		epeling
-	-		-	-	-	-	-	-	-	-	-	-		malessured innud
-	-	-	-	-	-	-	-	ATE	38	14	As	A 15	Ag	District
-			-	-	-		-	11			A L	A 15	Ag	newageg
OB	08	98			-	-	-	001	7	0		£		anivogeoet line sheed
-	-	-	BL.	-	-	-	-	ABC	23	E	A se	A se	Asc	innertendrating strated
-	-	-	100	-	-	-	-	001	38		E	E	E	wearing.
-	-	-	437	71.	L.	A g	113	OIE	HE		17	429	99	NOTICE THE PERSON
30	14	14		-	-	-	-	166	100	E		£		4098
-	-	-	-	-	-	-	-	4.000	-	-	-	-	-	unding
A ES	A 49	A 100	7	7	-	-	-	A-001	E	0	E	E.	L	snapag
-	-	-	-	-	-	-	-	-	-	-	-	-	-	and and self-real
-	-		A EE	-	-	-	-	33	10	60	E	86	E1.	Azelalged
-	-	-	-	-	-	-	-	-	-	-	2 E		2.0	NAME OF TAXABLE PARTY.
-	-	-	-	-	-	-	-	-	-	-	-	-	-	severyeg
1.6	86	96.	67	89	-		-	76	EL.	L.	As	A 20	A 4	nejedwich
-	-	-	-	-	-	-	-	4 000	-	-	-	-	-	eueny
-			-	-	-		-	4 000	-	-	-	-		equipmy
439	22	00	-	30	-	-	-	001	4	0	E			ROWNING
-	-	-	-	-	-	-	-	A gg	-	-	As	A 20	A 4	neguality
-	-	-	-	-	-	-	-	-	-	-	-	-	-	shorted int augint.
-	-	-	-	-	-	-	-	* 90	-	-	N 60	× 65	9.00	spelley
-	-	-	-	-	-	-	-	4 000	-	-	-	-	-	ensionA
48	68	888	10	-	-	-	-	66	E		Ay	A p	Ag	manify

```
#Merging PDF files
```

```
#A function to merge 2 pdfs together
def PDFmerge(pdfs, output):
  # creating pdf file merger object
   pdfMerger = PyPDF2.PdfFileMerger()
  #append file object of each pdf to pdf merger object
  for pdf in pdfs:
     with open(pdf, 'rb') as f:
        pdfMerger.append(f)
  #writing combined pdf to output pdf file
  with open(output, Wb') as f:
     pdfMerger.write(f)
#pdf files to merge
```

```
# pdf files to merge
pdfs = ['EN-FINAL Table 9.pdf', 'rotated_table_9.pdf']
# output pdf file name
output = 'combined_table.pdf'
# calling pdf merge function
PDFmerge(pdfs = pdfs, output = output)
```

-	 12
	 ~

Algeria	5 y	8 y	4.9		2	99	-		_	-	68	88	89	87
Andorra	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Angola	24 ×	22 x	25 x	-	-	26 x	-	_	-	-	-	-		-
Artigue and Barbuda	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Argentina	2 y	8 y	5 y	-	-	99 y	-	-		-	-	-	_	_
Armenia	4		2		7	100	-	-	-	20	9	70	72	62
Australia	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Austria	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Azerbaijan	2 y	8 y	5 y	1	12	94	-	-	-	58	49	75	29	71
Rahamas	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bahrain	5 ×	0 ×	2 ×	-	-	-	-	-	-	-	-	-	-	-
Rangladesh	13	18	8	29	45	21	-	-	-	-	22 y	-	-	-
Barbados	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Belarus	1	1	2	0	2	100 y	-	-	-	4	4	65 y	67 y	62 y
Belgium	-	-	-	-	-	100 v	_	-	-	-	-	-	_	-
Relize		7	6	2	26	95	-	-	-	-	9	71	71	70
Renin	48	67	46		34	80	13	2 y	1	16	47	-	-	-
Bhutan	3	2	2		26	100	-	-	-	-	68	-	-	-
Bolivia (Plurinational State of)	28 y	28 y	26 y	2	22	76 y	_	_	_	-	16	-	_	_
Bosnia and Herzegovina	6	7	4	0	4	100	-	-	-		6	55	60	60
Botowana	8 y	11 y	7 y	-	-	72	-	-	-	-	-	-	-	-
Brazil	3 y	11 y	6 y	11	26	93 y	-	-	-	-	-	-	-	-
Brunei Darussalam	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bulgaria	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Burbina Faso	29	62	36	10	62	77	76	13		36	66	83	84	82
Burundi	26	26	27	2	20	76	-	-	-	44	72	-	-	-
Cabo Verde	2 x,y	4 xy	2 жу	2	18	91	-	-	-	18 y	17	-	-	-
Cambodia	38 y	38 y	38 y	2	18	62	-	-	-	22 y	48 y	-	-	-
Cameroon	42	63	60	13	200	61	1	1 y	7	29	47	93	93	90
Canada	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Central African Republic	29	27	30	29	68	61	24	1	11	80 y	80	92	92	90
Ched	26	26	28	29	68	16	64	18 y	28	-	62	84	85	84.
Chile	3 ×	2 ×	2 ×	-	-	100 y	-	-	-	-	-	-	-	-
China	-	-	-	-	-	-	-	-	-					
Colombia	13 y	17 y	9 y		23	97	-	-	-					
Comoros	27 ×	28 x	28 ×	-	-	88 x	-	-	-			The state of		
Conge	25	24	25	7	22	91	_	-	-				⊢ l'	<b>∖,</b>  _F

78 THE STATE OF THE WORLD'S CHILDREN 2014 IN NUMBER

TABLE 9 CHILD PROTECTION

Child below (N)*					riage (%)	Biolic registration	Femalege	olal mullation 2000-2012*	(AF) geither)		-			
Countries		2009-2012*		300-	one.	(14)"	971	desce	attitudes		_	_		
and areas	total	male	Innale	married by 15	married by 18	2005-Date*	manner'	444	support for the proofine			P <sub>N</sub>		N-F
Cook Islands	-	-	-	-	-	-	-	-	-					`\ - -
Costa Rica	5 x	6 x	2 ×	-	-	-	-	-	-		_			
Cáte d'Ivoire	26	25	28	10	22	65	28	10	16	4				
Croatia	-	-	-	-	-	-	-	-	-					
Cuba	-	-	-	9	40	100 y	-	-	-	-	-	-	-	-
Cyprus	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Czech Republic	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Democratic People's Republic of Korea	-	-		-	-	100	-	_	-	-	_	-	_	-
Democratic Republic of the Congo	15	13	17		29	28	-	_	-	-	76	92	90	91
Denmark	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Djibouti		8	8	2	6	92	93	49 y	37	-	-	72	72	71
Dominica	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Dominican Republic	13	18	8	12	61	82	-	-	-	-	4	62	68	46
Equador		7	8	4 ×	22 ×	90	-	-	-	-	-	-	-	-
Egypt	8 y	16 y	4 y	2	17	99 y	91	17	64	-	29 y	91	92	90
El Salvador	10 y	-	-	6	26	99	-	-	-	-	-	-	-	-
Equatorial Guinea	28 ×	28 x	28 ×	-	-	27 x	-	-	-	-	-	-	-	-
Editor	-	-	-	20 x	47 x	-	89	63 y	49	-	71 ×	-	-	-
Esponia	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Ethiopia	27	21	26	16	61	7	74	24	31	45	66	-	-	-
Fiji	-	-	-	-	-	-	-	-	-	-	-	72 y	-	-
Finland	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
France	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Gabon	13	16	12		22	90	-	-	-	60	50	-	-	-
Gambia	19	21	18	7	26	52	76	54	66	-	76	90	90	91
Georgia	18	20	17	1	14	99	-	-	-	-	7	62	70	63
Germany	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Shara	34	34	36	6	21	62	4	1	2	28 y	64	94	94	94
Greece	-	-	-	-	-	100 v	-	-	-	-	-	-	-	-
Grenada	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Guatemala	28 y	25 y	18 y	7	30	97	-	-	-	-	-	-	-	-

EN-FINAL Table 90.pdf

EN-FINAL Table 91.pdf

EN-FINAL Table 92.pdf

```
#Splitting PDF file
#A function to split the pdf file
def PDFsplit(pdf, splits):
  # creating input pdf file object
   pdfFileObj = open(pdf, 'rb')
  # creating pdf reader object
   pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
   # starting index of first slice
   start = 0
   #starting index of last slice
   end = splits[0]
   for it in range(len(splits)+1):
            iting pdf writer object for (i+1)th split
            iter = PyPDF2.PdfFileWriter()
            out pdf file name
            str(i) is the xth number of split we are currently on
            pdf = pdf.split('.pdf')[0] + str(i) + '.pdf'
            ng pages to pdf writer object
            ge in range(start,end):
            Writer.addPage(pdfReader.getPage(page))
      #writing split pdf pages to pdf file
      with open(outputpdf, "wb") as f:
         pdfVVriter.write(f)
      #interchanging page split start position for next split
      start = end
        # setting split end positon for next split
         end = splits[i+1]
      except IndexError:
        #setting split end position for last split
         end = pdfReader.numPages
```

# SOURCES/ SHAREABLE LINK

- Shareable Link of my Tutorial:
  - https://colab.research.google.com/drive/1Qm7 N-UzV9E7Qucywjfl2A fpKWYXm3v
- Sources:
  - Jacqueline Kazil, Katherine Jarmul Data Wrangling with Python (Ch.5)
  - https://www.geeksforgeeks.org/working-with-pdf-files-in-python/
  - <a href="https://stackoverflow.com/questions/2365411/convert-unicode-to-ascii-without-errors-in-python/355362">https://stackoverflow.com/questions/2365411/convert-unicode-to-ascii-without-errors-in-python/355362</a>