Literate programming with Python, R, Julia and Stata**

Miguel Portela

Minho University

16 December, 2019

Abstract

In this presentation I will discuss how we can enhance the workflow by using literate programming to combine key features of different statistical packages, namely Stata, R, Julia and Python, on the one hand, and Latex as the typesetting system on the other. The goal is to demonstrate and share a template aiming at producing a highly automated report, or research paper, within the same framework. The tasks will run from exploratory data analysis to regression analysis, where the output, from summary to regression tables and figures, is seamlessly included in the final document. Furthermore, important elements of Latex editing, such as automatic referencing, will be highlighted. We aim at freeing the researcher form repetitive tasks to focus on critical and creative writing. Efficiency and replicability will be at the core of the discussion. RStudio will be used to edit and compile R Markdown. The focus will be on producing PDF outputs. In the presentation I will make use of packages such as bookdown, knitr, stargazer, dlookr, ggplot2, plotly, Statamarkdown, reticulate, JuliaCall, pandas, numpy, matplotlib or FixedEffectModels. The current code is an adaptation of the Rmd by Paul C. Bauer, Mannheim Centre for European Social Research, mail@paulcbauer.eu..

**Corresponding address: miguel.portela@eeg.uminho.pt.

1 Exploratory data analysis

I start by exploring the data **NLSWORK** (National Longitudinal Survey. Young Women 14-26 years of age in 1968).

```
## ExPanDaR: Explore Panel Data Interactively

  library(ExPanDaR)

    ## type ExPanD() in the Console

setwd("/Users/miguelportela/Documents/GitHub/prjs/logs")

library(haven)
library(ggplot2)

nlswork <- read_dta("/Users/miguelportela/Documents/GitHub/prjs/data/nlswork.dta")

nls<-data.frame(nlswork)

attach(nlswork)

head(nlswork)
```

2 A tibble: 6 x 21

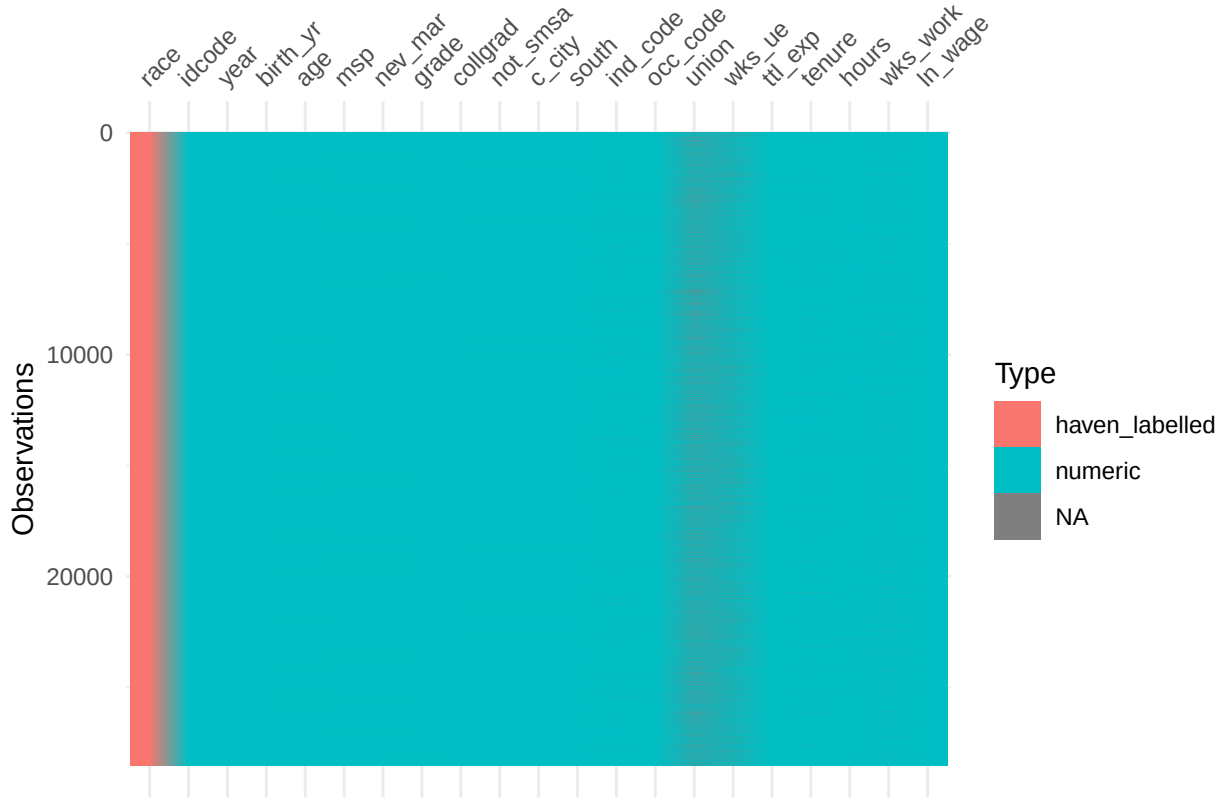
idcode year birth_yr age race msp nev_mar grade collgrad not_smsa <dbl+l> 1 1 70 51 18 2 [bla~ 0 1 12 0 0 2 1 71 51 19 2 [bla~ 1 0 12 0 0 3 1 72 51 20 2 [bla~ 1 0 12 0 0 4 1 73 51 21 2 [bla~ 1 0 12 0 0 5 1 75 51 23 2 [bla~ 1 0 12 0 0 6 1 77 51 25 2 [bla~ 0 0 12 0 0 # ... with 11 more variables: c_city , south , ind_code , # occ_code , union , wks_ue , ttl_exp , tenure , # hours , wks_work , ln_wage

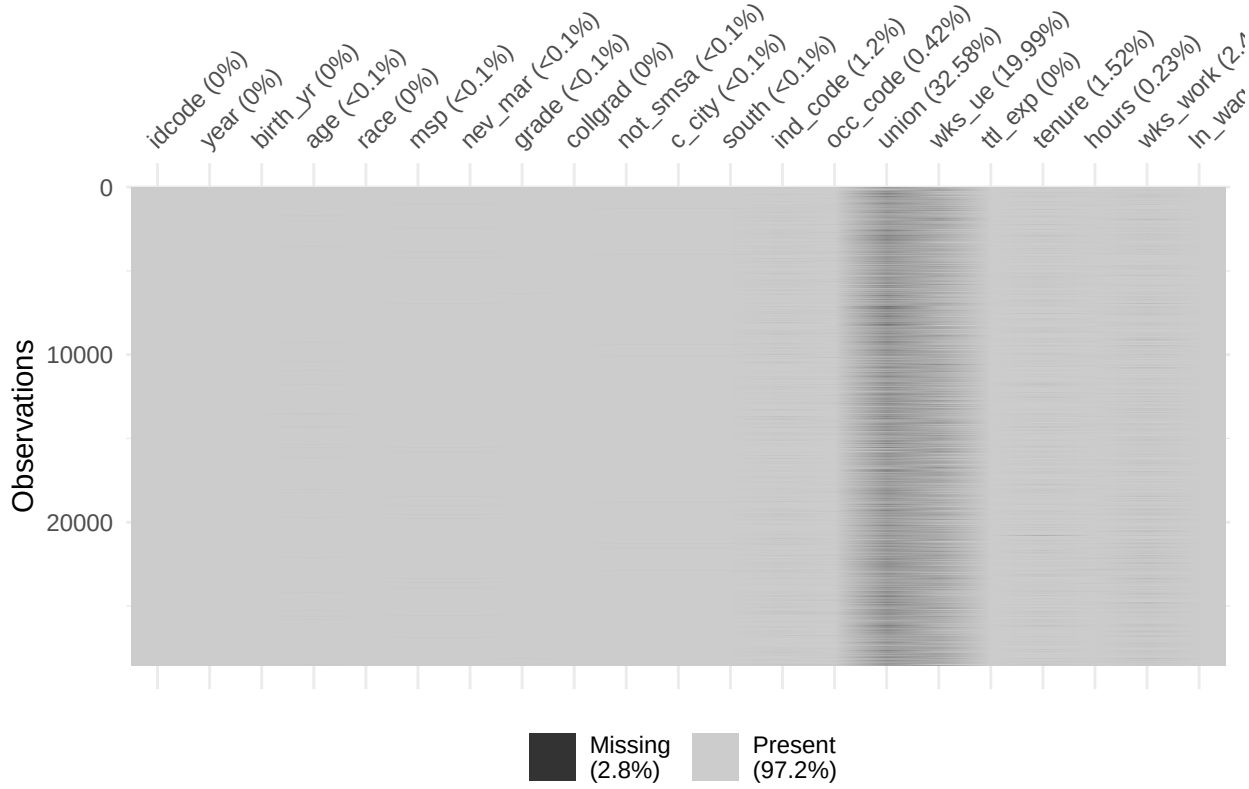
```
library(stargazer)
stargazer(nls,
          title = "Summary statistics",
          label="tab1",
          table.placement = "ht",
          header=FALSE)
```

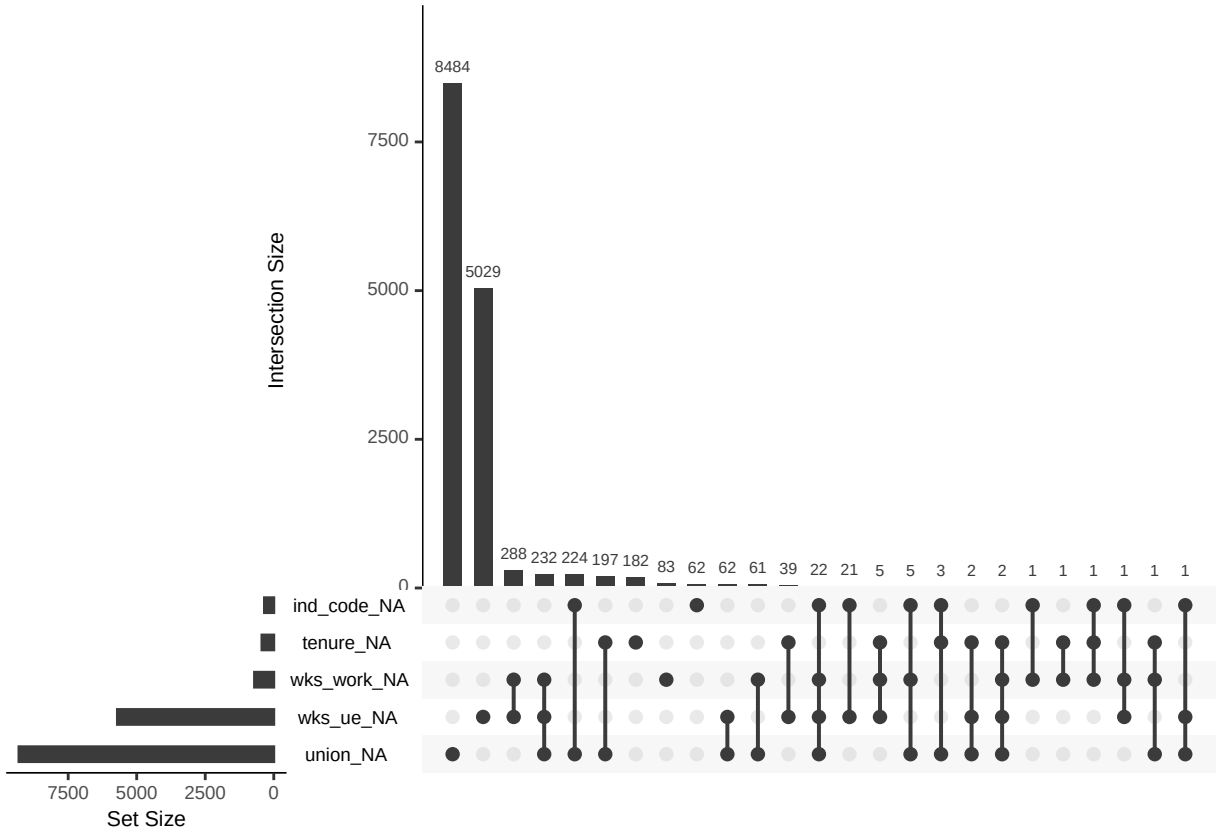
Table 1: Summary statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
idcode	28,534	2,601.284	1,487.359	1	1,327	3,881	5,159
year	28,534	77.959	6.384	68	72	83	88
birth_yr	28,534	48.085	3.013	41	46	51	54
age	28,510	29.045	6.701	14.000	23.000	34.000	46.000
race	28,534	1.303	0.482	1	1	2	3
msp	28,518	0.603	0.489	0.000	0.000	1.000	1.000
nev_mar	28,518	0.230	0.421	0.000	0.000	0.000	1.000
grade	28,532	12.533	2.324	0.000	12.000	14.000	18.000
collgrad	28,534	0.168	0.374	0	0	0	1
not_smsa	28,526	0.282	0.450	0.000	0.000	1.000	1.000
c_city	28,526	0.357	0.479	0.000	0.000	1.000	1.000
south	28,526	0.410	0.492	0.000	0.000	1.000	1.000
ind_code	28,193	7.693	2.994	1.000	5.000	11.000	12.000
occ_code	28,413	4.778	3.065	1.000	3.000	6.000	13.000
union	19,238	0.234	0.424	0.000	0.000	0.000	1.000
wks_ue	22,830	2.548	7.294	0.000	0.000	0.000	76.000
ttl_exp	28,534	6.215	4.652	0.000	2.462	9.128	28.885
tenure	28,101	3.124	3.751	0.000	0.500	4.167	25.917
hours	28,467	36.560	9.870	1.000	35.000	40.000	168.000
wks_work	27,831	53.989	29.032	0.000	36.000	72.000	104.000
ln_wage	28,534	1.675	0.478	0.000	1.361	1.964	5.264

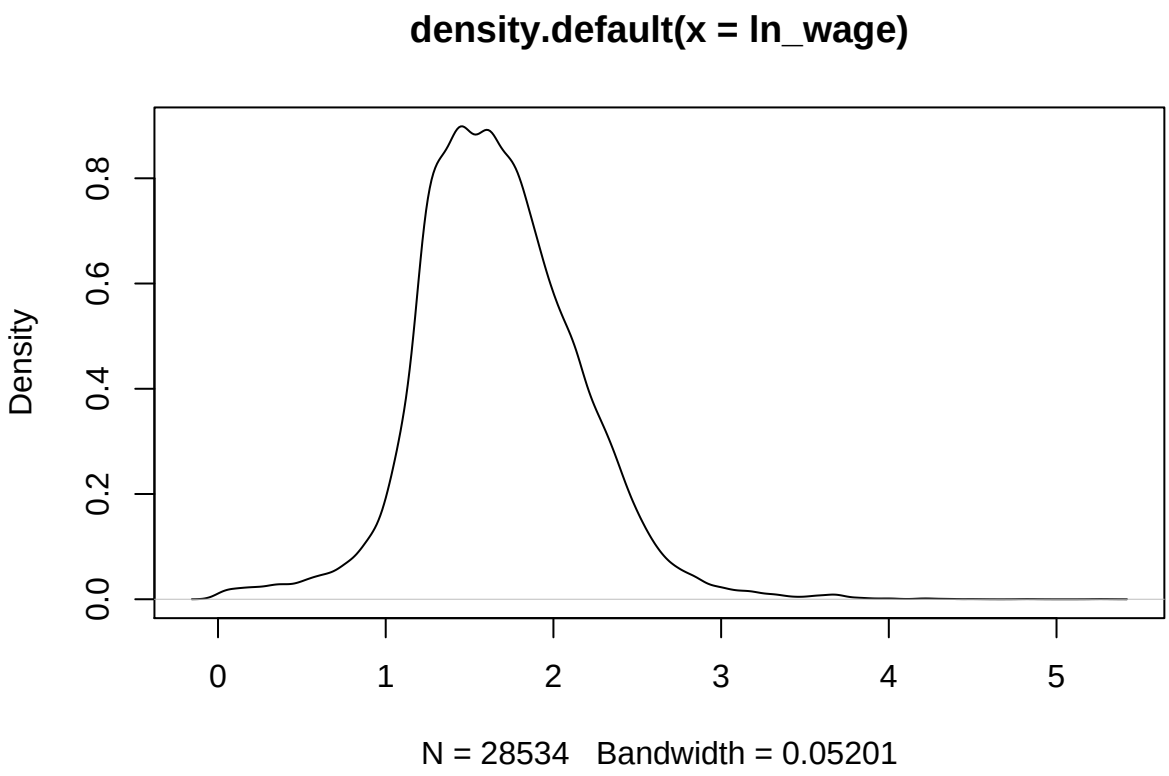
```
 [1] "idcode"   "year"     "birth_yr" "age"      "race"     "msp"
 [7] "nev_mar"  "grade"    "collgrad" "not_smsa" "c_city"   "south"
[13] "ind_code" "occ_code" "union"    "wks_ue"   "ttl_exp"  "tenure"
[19] "hours"    "wks_work" "ln_wage"
```

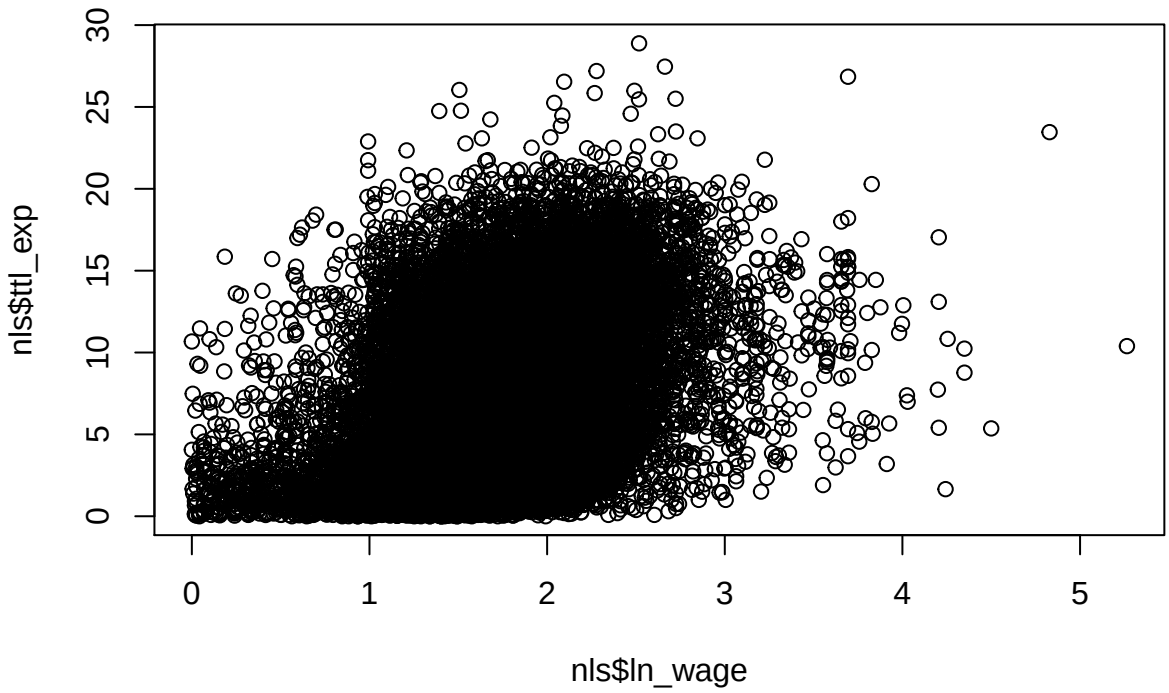

race
idcode
year
birth_yr
age
msp
nev_mar
grade
collgrad
not_smsa
c_city
south
ind_code
occ_code
union
wks_ue
ttl_exp
tenure
hours
wks_work
ln_wage
0
10000
20000
Observations
Type
haven_labelled
numeric
NA

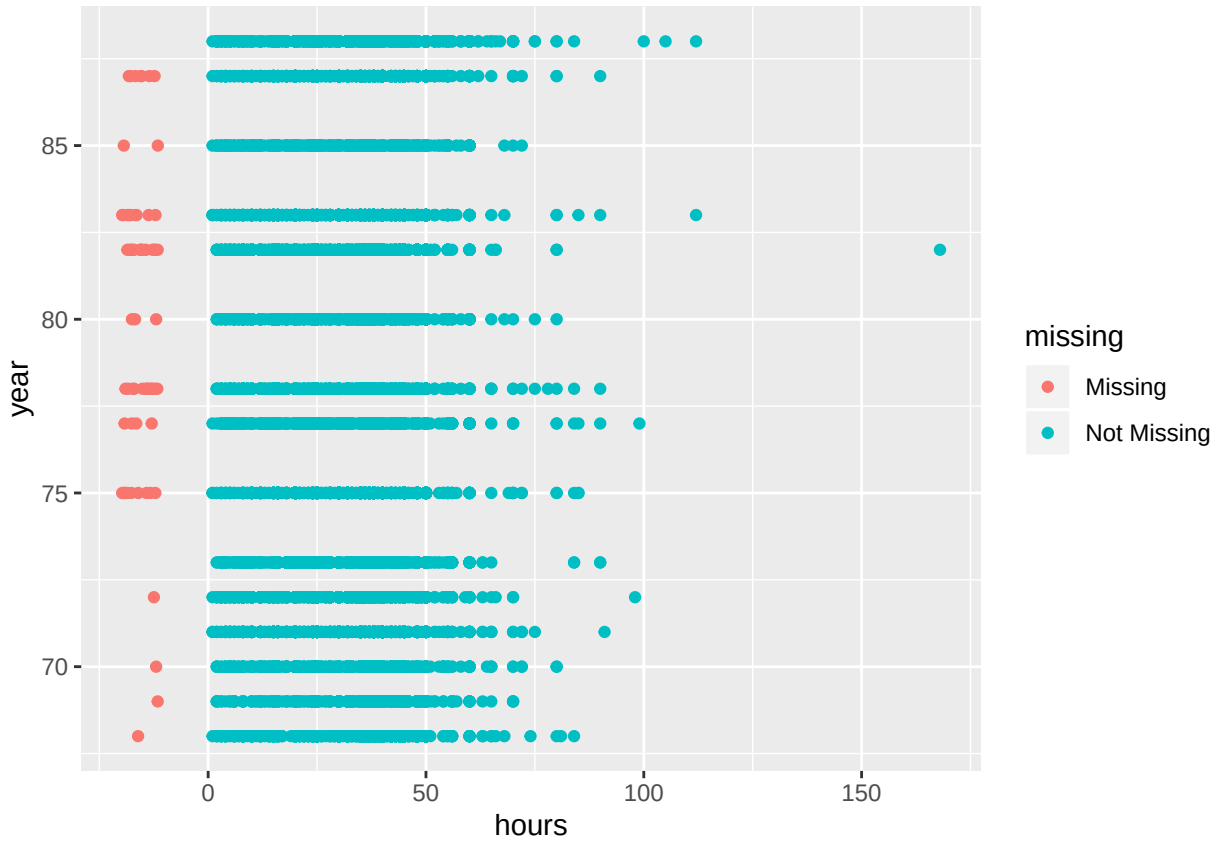

idcode (0%)
year (0%)
birth_yr (0%)
age (<0.1%)
race (0%)
msp (<0.1%)
nev_mar (<0.1%)
grade (<0.1%)
collgrad (0%)
not_smsa (<0.1%)
c_city (<0.1%)
south (<0.1%)
ind_code (1.2%)
occ_code (0.42%)
union (32.58%)
wks_ue (19.99%)
ttl_exp (0%)
tenure (1.52%)
hours (0.23%)
wks_work (2.
ln_wa
Observations
0
10000
20000
Missing
(2.8%)
Present
(97.2%)

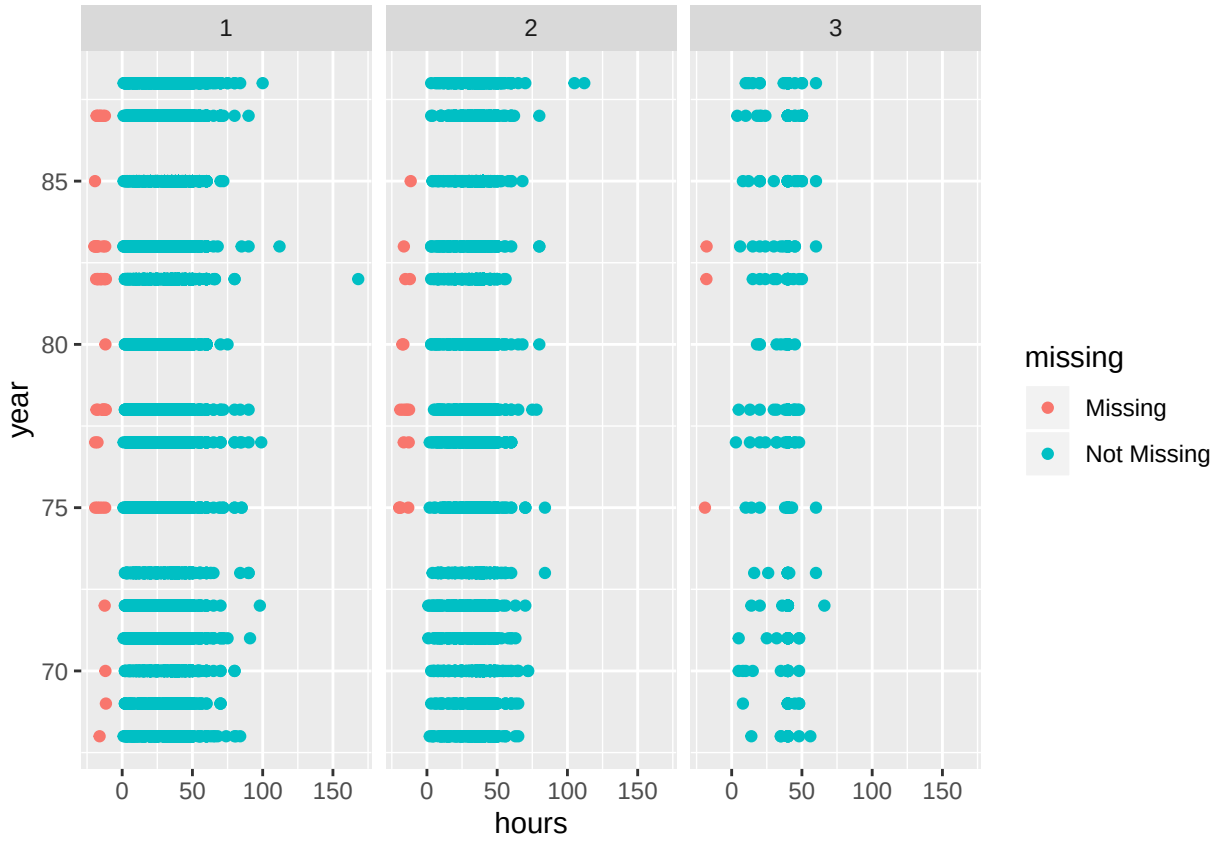



```
num [1:28534] 1.45 1.03 1.59 1.78 1.78 ...
- attr(*, "label")= chr "ln(wage/GNP deflator)"
- attr(*, "format.stata")= chr "%9.0g"
```


density.default(x = ln_wage)
Density
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
N = 28534 Bandwidth = 0.05201


nls$ttl_exp
0
5
10
15
20
25
30
0
1
2
3
4
5
nls$ln_wage


year
85
80
75
70
0
50
100
150
hours
missing
Missing
Not Missing



The average age in our data is 29.

3 Tables

Producing good tables and referencing these tables within a R Markdown PDF has been a hassle but got much better. Examples that you may use are shown below. The way you reference tables is slightly different, e.g., for `stargazer` the label is contained in the function, for `kable` it's contained in the chunk name.

3.1 stargazer(): Summary and regression tables

Table 1 shows summary stats of your data.[1] I normally use `stargazer()` (Hlavac 2013) which offers extreme flexibility regarding table output (see `?stargazer`).

[1] To reference the table where you set the identifier in the stargazer function you only need to use the actual label, i.e., Â´tab1Â´.

```
library(stargazer)
stargazer(cars,
          title = "Summary table with stargazer",
          label="tab1cars",
          table.placement = "H",
          header=FALSE)
```

Table 2: Summary table with stargazer

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
speed	50	15.400	5.288	4	12	19	25
dist	50	42.980	25.769	2	26	56	120

Table 3 shows the output for a regression table. Make sure you name all your models and explicitly refer to model names (M1, M2 etc.) in the text.

```
library(stargazer)
model1 <- lm(speed ~ dist, data = cars)
model2 <- lm(speed ~ dist, data = cars)
model3 <- lm(dist ~ speed, data = cars)
stargazer(model1, model2, model3,
          title = "Regression table with stargazer",
          label="tab2",
          table.placement = "H",
          column.labels = c("M1", "M2", "M3"),
          model.numbers = FALSE,
          header=FALSE)
```

Table 3: Regression table with stargazer

	Dependent variable:		
	speed		dist
	M1	M2	M3
dist	0.166***	0.166***	
	(0.017)	(0.017)	
speed			3.932***
			(0.416)
Constant	8.284***	8.284***	−17.579**
	(0.874)	(0.874)	(6.758)
Observations	50	50	50
R^2	0.651	0.651	0.651
Adjusted R^2	0.644	0.644	0.644
Residual Std. Error (df = 48)	3.156	3.156	15.380
F Statistic (df = 1; 48)	89.567***	89.567***	89.567***
Note:			*p<0.1; **p<0.05; ***p<0.01

4 Figures

4.1 R base graphs

Inserting figures can be slightly more complicated. Ideally, we would produce and insert them directly in the `.rmd` file. It's relatively simple to insert R base graphs as you can see in Figure 1.

```
plot(cars$speed, cars$dist)
```

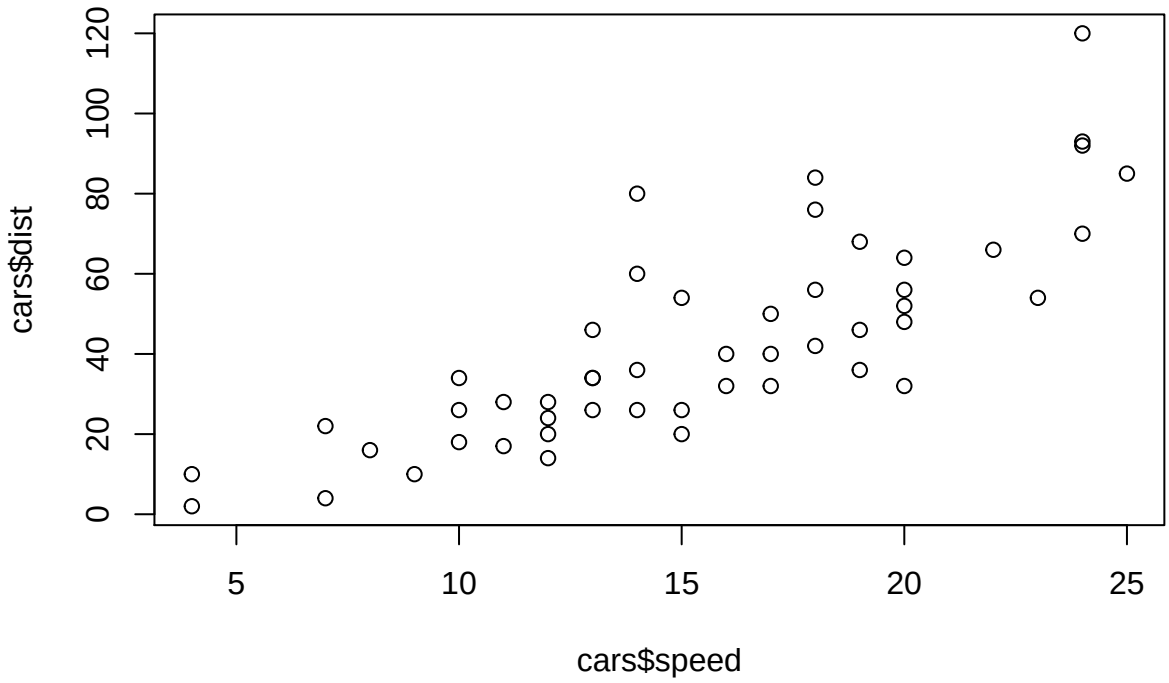


Figure 1: Scatterplot of Speed and Distance

But it turns out that it doesn't always work so well.

4.2 ggplot2 graphs

Same is true for ggplot2 as you can see in Figure 2.

```
mtcars$cyl <- as.factor(mtcars$cyl) # Convert cyl to factor
library(ggplot2)
ggplot(mtcars, aes(x=wt, y=mpg, shape=cyl)) + geom_point() +
  labs(x="Weight (lb/1000)", y = "Miles/(US) gallon",
       shape="Number of \n Cylinders") + theme_classic()
```

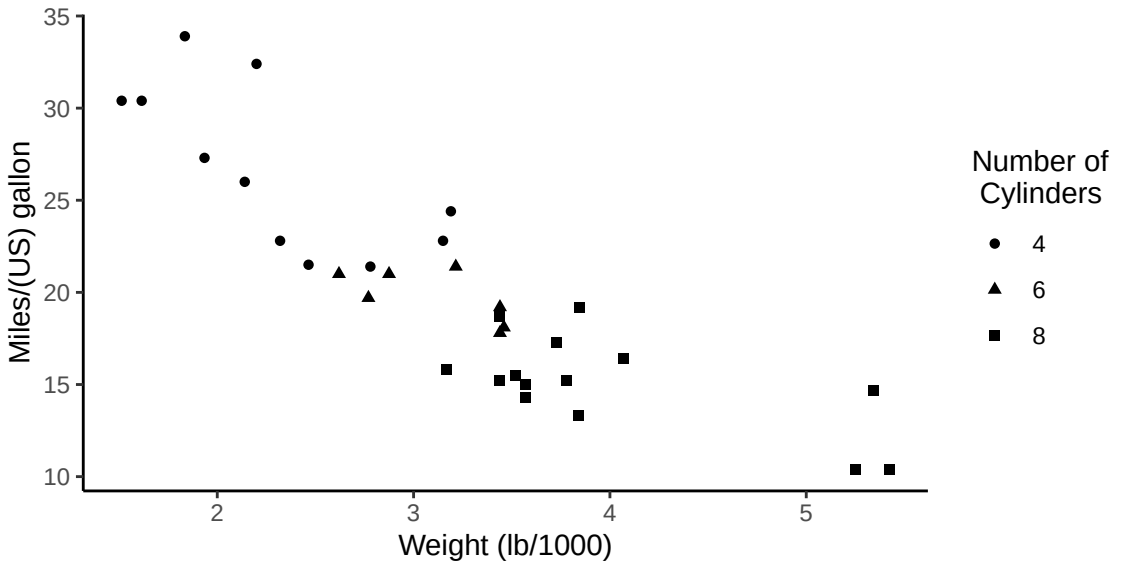


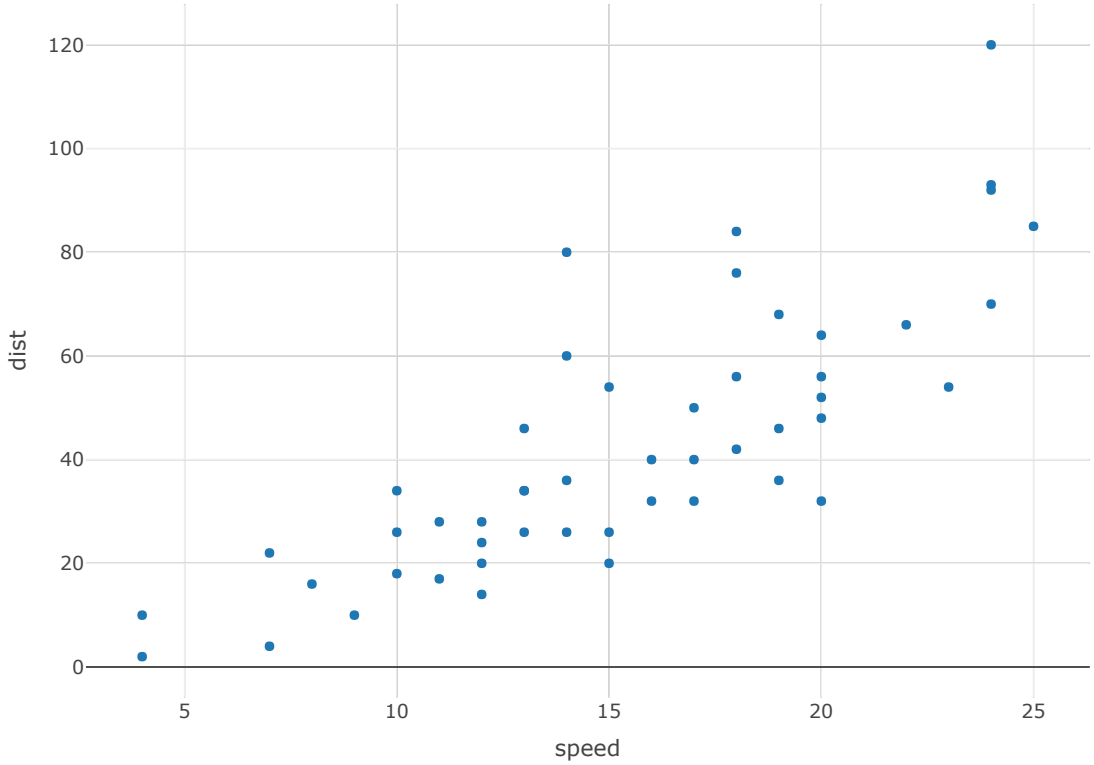
Figure 2: Miles per gallon according to the weight

4.3 Plotly graphs

Plotly is a popular graph engine that let's you also produce interactive graphs that you can embed in html webpages or documents (e.g., see here). I am a big fan. For some time there was no easy, automatic way to insert high resolution Plotly graphs into your R Markdown PDF. However, this changed since Plotly provided Orca, a command line application for generating static images from Plotly graphs. The installation is a bit tricky (see here: https://github.com/plotly/orca#installation) but once you get it running you can produce beautiful graphs and include them in your RMarkdown PDF using some simple latex as shown below in Figure 3. Potentially, in case you did not install the command line application this part may fail. If so simply exclude the chunk and the latex code.

```
library(plotly)
p <- plot_ly(cars, type = "scatter", mode="markers",
        x=~speed,
        y=~dist)
Sys.setenv('MAPBOX_TOKEN' = '12423423') # set arbitrary token
orca(p, "plotly-plot.pdf")
```

Figure 3: An plotly plot that was exported as PDF with orca before



5 Python

5.1 API data download using Python

```
import sys
print(sys.version)
```

```
3.8.0 (v3.8.0:fa919fdf25, Oct 14 2019, 10:23:27)
[Clang 6.0 (clang-600.0.57)]
```

```
import json
##from json.decoder import JSONDecodeError
import requests
import numpy as np
import pandas as pd
```

```
## INE: https://www.ine.pt/ine/json_indicador/pindica.jsp?
## op=2&varcd=0008074&Dim1=S7A2015&Dim2=200&Dim3=3&lang=PT

# api-endpoint

URL = "https://www.ine.pt/ine/json_indicador/pindica.jsp"

# define parameters

OP="2"
VARCD="0008074"
DIM1="S7A2015"
DIM2="200"
DIM3="3"
LANG="PT"


# defining a params dict for the parameters to be sent to the API
PARAMS = {'op':OP,'varcd':VARCD,'Dim1':DIM1,'Dim2':DIM2,'Dim3':DIM3,'lang':LANG}

# sending get request and saving the response as response object
r = requests.get(url = URL,params=PARAMS)

# extracting data in json format
data = r.json()

valor = data[0]['Dados']['2015'][0]['valor']

valor
```

```
'1.8'
```

The value is 1.8.

This works fine and as expected.

```
x = 42 * 2
print(x)
```

```
84
```

The value of `x` in the Python session is 84.

5.2 Import data from PDF files

```
  cd /Users/miguelportela/Documents/GitHub/prjs/pdfs
    find . -name '*.pdf' -print0 | xargs -0 -n1 pdfsandwich -gray
    find . -name '*ocr.pdf' -print0 | xargs -0 -n1 pdftotext
```

```
['', 'PORTARIAS 111111111 DE REGULAMENTAGAO DO TRABALHO', 'PORTARIAs de EXTENSAO 4444444

FILE:  sample_text_v4
match 1
match 4
match 1
match 4
match 1
match 4
match 3
match 1
match 4
['zzzz', 'PE dasalteragoes do, CCTentre a Assoc. Nacional  dos, Opticos e a FETESE -- Fe

FILE:  sample_text_v5
-> match 5
PE dasalteragoes do, CCTentre a Assoc. Nacional  dos, Opticos e a FETESE -- Feder. dos S
99999

   linha  ...          source
0      1  ...  sample_text_v4
1      2  ...  sample_text_v4
2      3  ...  sample_text_v4
3      6  ...  sample_text_v4
4      9  ...  sample_text_v4
5      1  ...  sample_text_v5

[6 rows x 4 columns]
```

And now we use Stata to explore the data.

```
quiet cd "/Users/miguelportela/Documents/GitHub/prjs/logs"
quiet import delimited "/Users/miguelportela/Documents/GitHub/prjs/data/PE.csv", encodin
tab source
```

```
 command window is unrecognized
r(199);
```

```
        source |      Freq.     Percent        Cum.
---------------+-----------------------------------
sample_text_v4 |          5       83.33       83.33
sample_text_v5 |          1       16.67      100.00
---------------+-----------------------------------
         Total |          6      100.00
```

```
quietly{
cd /Users/miguelportela/Documents/GitHub/prjs/chunks

use nipcs, clear
compress
contract nipc
drop _freq
drop if nipc == .
format %12.0f nipc
}

codebook nipc

tab nipc
```

```
 command window is unrecognized
r(199);
```

```
-------------------------------------------------------------------------------
nipc                                                            (unlabeled)
-------------------------------------------------------------------------------

                  type:  numeric (long)

                 range:  [5.106e+08,5.155e+08]        units:  1
         unique values:  23                       missing .:  0/23
```

```
                    mean:   5.1e+08
                 std. dev:   1.9e+06

              percentiles:        10%       25%       50%       75%       90%
```

nipc	Freq.	Percent	Cum.
510649068	1	4.35	4.35
510779174	1	4.35	8.70
511056737	1	4.35	13.04
511117060	1	4.35	17.39
511124899	1	4.35	21.74
511240619	1	4.35	26.09
511247478	1	4.35	30.43
513208348	1	4.35	34.78
513587128	1	4.35	39.13
514118890	1	4.35	43.48
514525657	1	4.35	47.83
514532718	1	4.35	52.17
514591889	1	4.35	56.52
515002666	1	4.35	60.87
515080985	1	4.35	65.22
515092550	1	4.35	69.57
515092649	1	4.35	73.91
515464236	1	4.35	78.26
515478377	1	4.35	82.61
515484920	1	4.35	86.96
515517135	1	4.35	91.30
515518565	1	4.35	95.65
515522988	1	4.35	100.00
Total	23	100.00	

6 Julia experiments

6.1 Computations

```
1.4142135623730951
```

6.2 Grab results in R

```
[1] 1.414214
```

```
Julia Object of type FixedEffectModel.
                           Fixed Effect Model
==========================================================================
Number of obs:               147715   Degrees of freedom:           67180
R2:                           0.978   R2 Adjusted:                  0.960
F Statistic:                 23.362   p-value:                      0.000
R2 within:                    0.001   Iterations:                     419
Converged:                     true
==========================================================================
             Estimate   Std.Error t value Pr(>|t|)   Lower 95%  Upper 95%
--------------------------------------------------------------------------
education  0.00155631 0.000597587 2.60432    0.009 0.000385043 0.00272758
lnsales    0.00622989 0.000987569 6.30831    0.000  0.00429426 0.00816552
==========================================================================
```

```
use /Users/miguelportela/Documents/GitHub/prjs/data/data_short, clear

timer on 1

    reghdfe lnrealwage education lnsales,absorb(workerid firmid year)

timer off 1
timer list 1
timer clear 1
```

```
 command window is unrecognized
r(199);


( )


(MWFE estimator converged in 236 iterations)

HDFE Linear regression                            Number of obs   =    147,715
Absorbing 3 HDFE groups                           F(   2,  99667) =      28.91
```

```
                                                Prob > F        =     0.0000
                                                R-squared       =     0.9782
                                                Adj R-squared   =     0.9677
                                                Within R-sq.    =     0.0006
                                                Root MSE        =     0.0943

------------------------------------------------------------------------------
  lnrealwage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   education |   .0015563   .0005372     2.90   0.004     .0005034    .0026092
     lnsales |   .0062299   .0008877     7.02   0.000     .0044899    .0079698
       _cons |   1.577908   .0148587   106.19   0.000     1.548785     1.60703
------------------------------------------------------------------------------

Absorbed degrees of freedom:
-----------------------------------------------------+
 Absorbed FE | Categories  - Redundant  = Num. Coefs |
-------------+---------------------------------------|
    workerid |     44047           0       44047     |
      firmid |     23127       19131        3996     |
        year |         4           1           3    ?|
-----------------------------------------------------+
? = number of redundant parameters may be higher


   1:     21.26 /        1 =      21.2590
```

```r
library(lfe)
data_short <- read_dta("/Users/miguelportela/Documents/GitHub/prjs/data/data_short.dta")

system.time(est_hdfe <- felm(data_short$lnrealwage ~ data_short$education + data_short$l

summary(est_hdfe)
```

The estimated return to education is 0.2%. The model has an R^2 of 0.9782.

	lnrealwage	
	(1)	(2)
education	0.006***	0.002**
	(0.000)	(0.001)
lnsales	0.013***	0.006***
	(0.001)	(0.001)
workerid	Yes	Yes
year	Yes	Yes
firmid		Yes
Estimator	OLS	OLS
N	147,715	147,715
R^2	0.970	0.978

6.3 Output Julia's table for HDFE

7 Miguel's tests

7.1 Tasks

Produzir um relatório com base no NLSWORK, desde estatística descritiva, com os valores inseridos automaticamente no texto, gráficos e regressões. Com o Python corremos o EDA, Julia o REGHDFE for speed, com R o RMarkdown + functions & Stata ??? functions???

WORKSHOP: fazer uma acta do evento no formato de um 'package' com a replicabilidade, markdown, ...

Python: explorar o Pandas e o Numpy

7.2 R

Table 5 ... See Section 7.3

Example of an equation

$$\int_0^{2\pi} \sin x \, dx$$

Example of a matrix

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$f\left(k\right) = \binom{n}{k} p^k \left(1-p\right)^{n-k} \tag{1}$$

$$

See Equation (1).

$$y_{ijt} = \beta x_{ijt} + \eta_i + \gamma_j + \lambda_t + \varepsilon_{ijt} \tag{2}$$

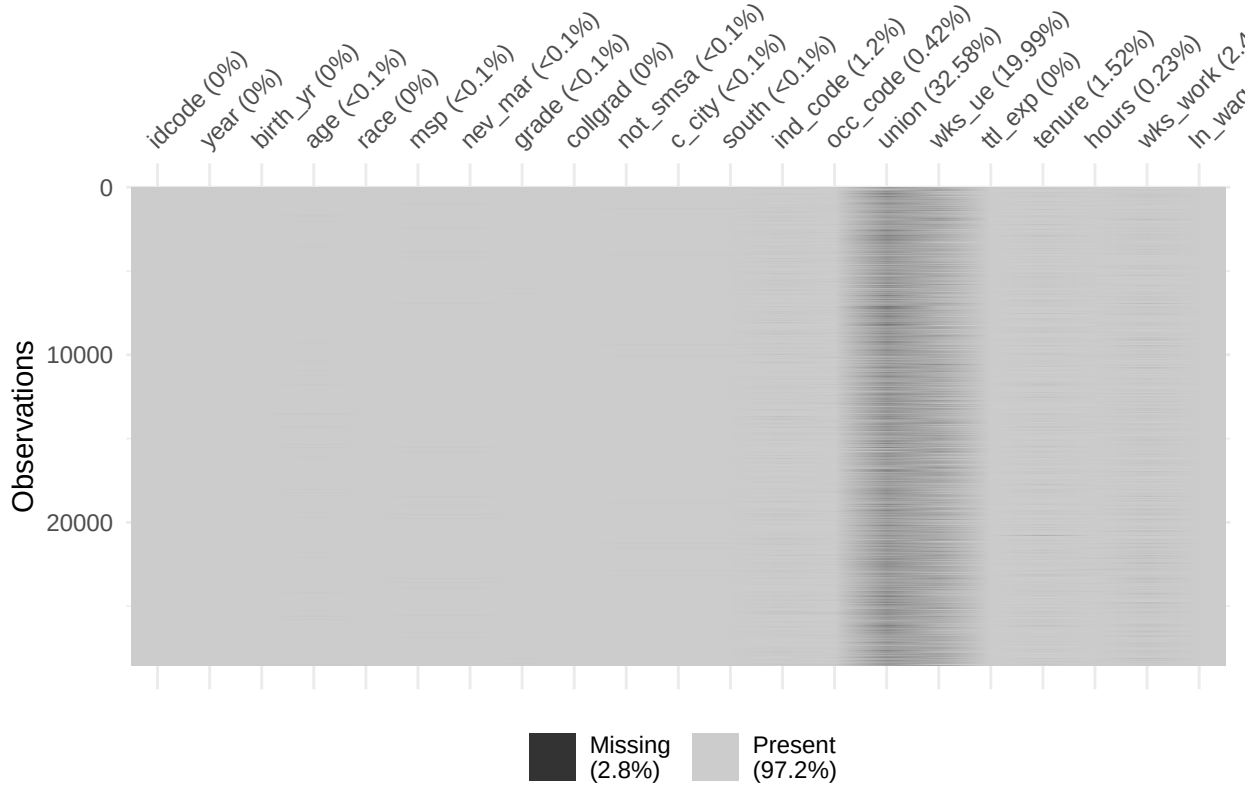
Table 4: Summary table

Statistic	N	Pctl(75)	St. Dev.
idcode	28,534	3,881	1,487.359
year	28,534	83	6.384
birth_yr	28,534	51	3.013
age	28,510	34.000	6.701
race	28,534	2	0.482
msp	28,518	1.000	0.489
nev_mar	28,518	0.000	0.421
grade	28,532	14.000	2.324
collgrad	28,534	0	0.374
not_smsa	28,526	1.000	0.450
c_city	28,526	1.000	0.479
south	28,526	1.000	0.492
ind_code	28,193	11.000	2.994
occ_code	28,413	6.000	3.065
union	19,238	0.000	0.424
wks_ue	22,830	0.000	7.294
ttl_exp	28,534	9.128	4.652
tenure	28,101	4.167	3.751
hours	28,467	40.000	9.870
wks_work	27,831	72.000	29.032
ln_wage	28,534	1.964	0.478

Table 5: Regression table with stargazer

	Dependent variable:		
	price		
	M1	M2	M3
mpg	−49.512	−52.217	−63.210
	(86.156)	(83.740)	(84.218)
weight	1.747***	2.111***	2.442***
	(0.641)	(0.619)	(0.688)
rep78			
Observations	74	69	69
R^2	0.293	0.365	0.376
Adjusted R^2	0.273	0.335	0.337
Residual Std. Error	2,514.029 (df = 71)	2,374.370 (df = 65)	2,370.832 (df = 64)
F Statistic	14.740*** (df = 2; 71)	12.437*** (df = 3; 65)	9.654*** (df = 4; 64)

Note: *p<0.1; **p<0.05; ***p<0.01



```r
library(stargazer)
stargazer(cars,
          title = "Summary 24",
          label="tab24",
          table.placement = "ht",
          header=FALSE)
```

Table 6: Summary 24

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
speed	50	15.400	5.288	4	12	19	25
dist	50	42.980	25.769	2	26	56	120

7.3 Stata

This a Stata example, Arellano (2003). See also Arellano and Bond (1991) and Blundell and Bond (1998). While ... (check Arellano and Bover 1995).

```
 command window is unrecognized
r(199);

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
       price |         74    6165.257    2949.496       3291      15906

     Repair |
Record 1978 |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          2        2.90        2.90
          2 |          8       11.59       14.49
          3 |         30       43.48       57.97
          4 |         18       26.09       84.06
          5 |         11       15.94      100.00
------------+-----------------------------------
      Total |         69      100.00
```

```
(file /Users/miguelportela/Documents/GitHub/prjs/logs/density.pdf written in PD
> F format)

      Source |       SS           df       MS      Number of obs   =       234
-------------+----------------------------------   F(7, 226)       =     46.99
       Model |  145.879747         7  20.8399639   Prob > F        =    0.0000
    Residual |  100.230749       226  .443498888   R-squared       =    0.5927
-------------+----------------------------------   Adj R-squared   =    0.5801
       Total |  246.110496       233  1.05626822   Root MSE        =    .66596

------------------------------------------------------------------------------
       lngdp |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   education |   .2136664   .0193553    11.04   0.000     .1755265    .2518063
         lnk |   .1978085   .0308039     6.42   0.000     .1371089    .2585082
       openk |   .0062439   .0011852     5.27   0.000     .0039085    .0085794
             |
        year |
       1975  |  -.0694608   .1387178    -0.50   0.617    -.3428064    .2038849
       1980  |   -.177992   .1401702    -1.27   0.205    -.4541996    .0982156
       1985  |  -.2226975   .1400607    -1.59   0.113    -.4986894    .0532943
       1990  |    -.34965   .1425169    -2.45   0.015    -.6304819   -.0688182
             |
       _cons |    3.38917   .7508785     4.51   0.000     1.909552    4.868789
------------------------------------------------------------------------------
```

The mean is `s $xx` ...

We now export a set of statistics to an Excel file.

```
 command window is unrecognized
r(199);


version 15.1

/Users/miguelportela/Library/Application Support/Stata/ado/plus/x/xtabond2.ado

Checksum for /Users/miguelportela/Library/Application Support/Stata/ado/plus/x/
```

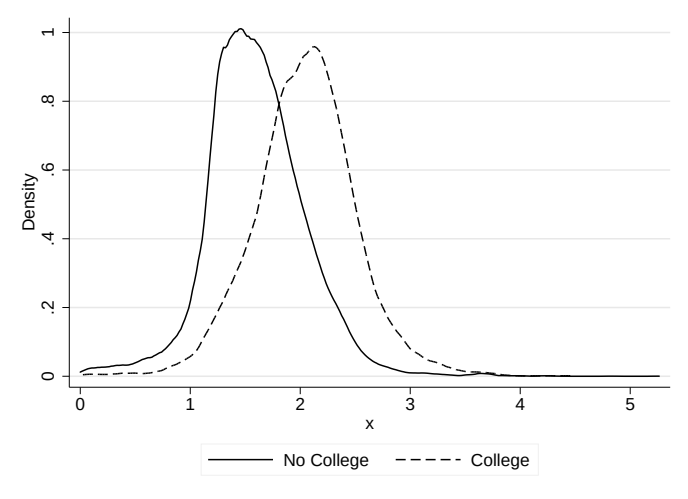


Figure 4: Wage density

```
> xtabond2.ado = 616966544, size = 39434

/Users/miguelportela/Documents/GitHub/prjs/logs


Variable   Obs Unique      Mean       Min       Max  Label
-------------------------------------------------------------------------------
country    839    106         .         .         .  Country name
year       839      9  1980.906      1960      2000  Year of observation
education  839    574  4.794076       .04     12.25  Education
lngdp      839    838  9.308131  5.983335  12.51058  Log Real GDP per Worker
open       839      2  .4982122         0         1  1 = high degree of open...
gdp        839    838  20100.66  396.7612  271192.2  GDP level
-------------------------------------------------------------------------------


Note: file will be replaced when the first putexcel command is issued


`"a"' `"b"' `"c"' `"d"' `"e"' `"f"' `"g"' `"h"' `"i"' `"j"' `"k"' `"l"' `"m"' `
> "n"' `"p"' `"r"' `"s"' `"t"' `"u"' `"v"' `"z"'
```

Table 7: Regression analysis

	Simple model	Include capital	Full model
Education	0.3169***	0.212***	0.2***
	(0.0093)	(0.020)	(0.0)
Capital		0.125***	0.2***
		(0.029)	(0.0)
Openness degree			0.0***
			(0.0)
R^2	0.58	0.54	0.59
RMSE	0.78	0.70	0.67
N	857	234	234

* $p < 0.1$; ** $p < 0.05$; *** $p < 0.01$

```
Country's first letter:     a

    Insufficient number of countries; n countries = 5



Country's first letter:     b

    Number of countries:    11



Country's first letter:     c

    Number of countries:    9



Country's first letter:     d

    Insufficient number of countries; n countries = 2
```

```
Country's first letter:     e

    Insufficient number of countries; n countries = 5



Country's first letter:     f

    Insufficient number of countries; n countries = 3



Country's first letter:     g

    Insufficient number of countries; n countries = 4



Country's first letter:     h

    Insufficient number of countries; n countries = 4



Country's first letter:     i

    Number of countries:    7



Country's first letter:     j

    Insufficient number of countries; n countries = 3
```

```
Country's first letter:     k

    Insufficient number of countries; n countries = 2



Country's first letter:     l

    Insufficient number of countries; n countries = 2



Country's first letter:     m

    Number of countries:    8



Country's first letter:     n

    Number of countries:    6



Country's first letter:     p

    Number of countries:    7



Country's first letter:     r
```

```
    Insufficient number of countries; n countries = 2



Country's first letter:     s

    Number of countries:    14



Country's first letter:     t

    Insufficient number of countries; n countries = 5



Country's first letter:     u

    Insufficient number of countries; n countries = 4



Country's first letter:     v

    Insufficient number of countries; n countries = 1



Country's first letter:     z

    Insufficient number of countries; n countries = 2
```

```
x = 5  # radius of a circle
```

For a circle with the radius 5, its area is 78.5398163.

See Figure 5.

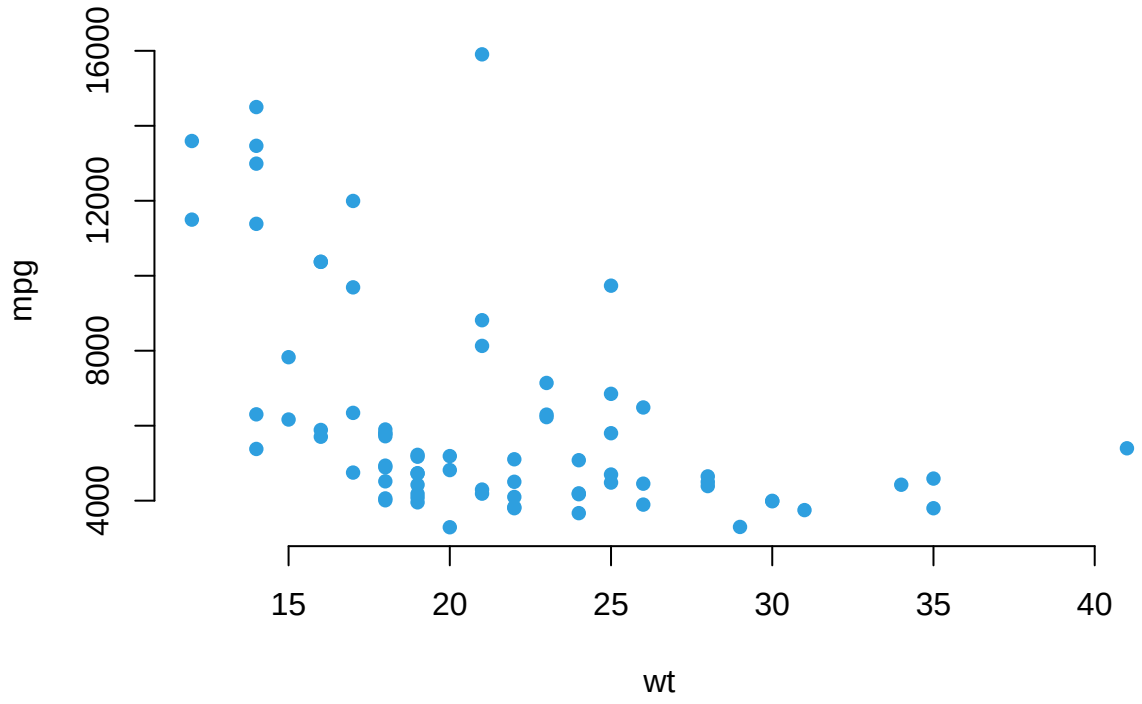


Figure 5: Scatterplot test MP

8 Final remarks

Check: https://github.com/tlamadon/blm-replicate

9 Appendix

9.1 Software versioning

```
cat(paste("#", capture.output(sessionInfo()), "\n", collapse =""))
```

```
# R version 3.6.1 (2019-07-05)
# Platform: x86_64-apple-darwin15.6.0 (64-bit)
# Running under: macOS Catalina 10.15.2
#
# Matrix products: default
# BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
# LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
#
# locale:
# [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#
# attached base packages:
# [1] stats     graphics  grDevices utils     datasets  methods   base
#
# other attached packages:
#  [1] JuliaCall_0.17.1    plotly_4.9.1        naniar_0.4.2
#  [4] visdat_0.5.3        dlookr_0.3.12       mice_3.6.0
#  [7] lattice_0.20-38     dplyr_0.8.3         ggplot2_3.2.1
# [10] haven_2.1.1         ExPanDaR_0.4.0      Statamarkdown_0.3.9
# [13] stargazer_5.2.2     reticulate_1.13
#
# loaded via a namespace (and not attached):
#   [1] readxl_1.3.1          backports_1.1.5       Hmisc_4.2-0
#   [4] corrplot_0.84         plyr_1.8.4            lazyeval_0.2.2
#   [7] splines_3.6.1         crosstalk_1.0.0       digest_0.6.20
#  [10] htmltools_0.4.0       gdata_2.18.0          fansi_0.4.0
#  [13] magrittr_1.5          checkmate_1.9.4       memoise_1.1.0
#  [16] cluster_2.1.0         ROCR_1.0-7            openxlsx_4.1.0.1
#  [19] readr_1.3.1           xts_0.11-2            sandwich_2.5-1
#  [22] askpass_1.1           colorspace_1.4-1      blob_1.2.0
#  [25] rvest_0.3.5           pan_1.6               xfun_0.11
#  [28] tcltk_3.6.1           libcoin_1.0-5         crayon_1.3.4
#  [31] jsonlite_1.6          lme4_1.1-21           zeallot_0.1.0
```

```
#  [34] survival_2.44-1.1     zoo_1.8-6             glue_1.3.1
#  [37] kableExtra_1.1.0      smbinning_0.9         gtable_0.3.0
#  [40] UpSetR_1.4.0          webshot_0.5.1         car_3.0-4
#  [43] quantmod_0.4-15       jomo_2.6-9            abind_1.4-5
#  [46] scales_1.0.0          mvtnorm_1.0-11        DBI_1.0.0
#  [49] Rcpp_1.0.3            viridisLite_0.3.0     xtable_1.8-4
#  [52] htmlTable_1.13.2      foreign_0.8-72        bit_1.1-14
#  [55] Formula_1.2-3         sqldf_0.4-11          DT_0.9
#  [58] htmlwidgets_1.5.1     httr_1.4.1            gplots_3.0.1.1
#  [61] RColorBrewer_1.1-2    acepack_1.4.1         ellipsis_0.3.0
#  [64] pkgconfig_2.0.3       nnet_7.3-12           utf8_1.1.4
#  [67] labeling_0.3          tidyselect_0.2.5      rlang_0.4.0
#  [70] later_1.0.0           munsell_0.5.0         cellranger_1.1.0
#  [73] tools_3.6.1           cli_1.1.0             gsubfn_0.7
#  [76] generics_0.0.2        moments_0.14          RSQLite_2.1.2
#  [79] broom_0.5.2           evaluate_0.14         stringr_1.4.0
#  [82] fastmap_1.0.1         yaml_2.2.0            processx_3.4.1
#  [85] knitr_1.26            bit64_0.9-7           shinycssloaders_0.2.0
#  [88] zip_2.0.4             caTools_1.17.1.2      purrr_0.3.3
#  [91] mitml_0.3-7           nlme_3.1-141          mime_0.7
#  [94] tictoc_1.0            xml2_1.2.2            compiler_3.6.1
#  [97] rstudioapi_0.10       curl_4.2              e1071_1.7-2
# [100] tibble_2.1.3          stringi_1.4.3         ps_1.3.0
# [103] forcats_0.4.0         Matrix_1.2-17         classInt_0.4-2
# [106] nloptr_1.2.1          vctrs_0.2.0           RcmdrMisc_2.5-1
# [109] pillar_1.4.2          lifecycle_0.1.0       data.table_1.12.6
# [112] bitops_1.0-6          httpuv_1.5.2          R6_2.4.0
# [115] latticeExtra_0.6-28   bookdown_0.16         promises_1.1.0
# [118] KernSmooth_2.23-16    gridExtra_2.3         rio_0.5.16
# [121] boot_1.3-23           MASS_7.3-51.4         gtools_3.8.1
# [124] assertthat_0.2.1      chron_2.3-54          proto_1.0.0
# [127] openssl_1.4.1         withr_2.1.2           nortest_1.0-4
# [130] DMwR_0.4.1            parallel_3.6.1        hms_0.5.1
# [133] grid_3.6.1            prettydoc_0.3.0       rpart_4.1-15
# [136] tidyr_1.0.0           class_7.3-15          minqa_1.2.4
# [139] inum_1.0-1            rmarkdown_2.0         carData_3.0-2
# [142] TTR_0.23-5            partykit_1.2-5        shiny_1.4.0
# [145] base64enc_0.1-3       tinytex_0.18
```

```
  # or use message() instead of cat()
```

9.2 All the code in the paper

To simply attach all the code you used in the PDF file in the appendix see the R chunk in the underlying `.rmd` file:

```
knitr::opts_chunk$set(cache = FALSE)
# Use chache = TRUE if you want to speed up compilation

# A function to allow for showing some of the inline code
rinline <- function(code){
  html <- '<code  class="r">``` `r CODE` ```</code>'
  sub("CODE", code, html)

  ##https://opensource.com/article/19/5/python-3-default-mac

  Sys.setenv(RETICULATE_PYTHON = "/usr/local/bin/python3")

##install.packages("reticulate")
library(reticulate)
##use_python("/Library/Frameworks/Python.framework/Versions/3.8/bin/python3")

use_virtualenv("/Users/miguelportela/.pyenv/version")

##knitr::opts_chunk$set(python.reticulate=FALSE)

library(JuliaCall)

library(Statamarkdown)
stataexe <- "/Applications/Stata15/StataMP.app/Contents/MacOS//stata-mp"
knitr::opts_chunk$set(engine.path=list(stata=stataexe))

}
Sys.setenv(RETICULATE_PYTHON = "/usr/local/bin/python3")
library(reticulate)
use_virtualenv("/Users/miguelportela/.pyenv/version")
library(stargazer)
library(Statamarkdown)
```

```
stataexe <- "/Applications/Stata15/StataMP.app/Contents/MacOS//stata-mp"
knitr::opts_chunk$set(engine.path=list(stata=stataexe))


## ExPanDaR: Explore Panel Data Interactively

  library(ExPanDaR)

    ## type ExPanD() in the Console

setwd("/Users/miguelportela/Documents/GitHub/prjs/logs")

library(haven)
library(ggplot2)

nlswork <- read_dta("/Users/miguelportela/Documents/GitHub/prjs/data/nlswork.dta")

nls<-data.frame(nlswork)

attach(nlswork)

head(nlswork)

library(stargazer)
stargazer(nls,
          title = "Summary statistics",
          label="tab1",
          table.placement = "ht",
          header=FALSE)

library(dplyr)
library(dlookr)
library(ggplot2)

##eda_report(nlswork,output_dir = "/Users/miguelportela/Documents/GitHub/prjs/reports/

## The data

names(nlswork)
##summary(nlswork)
```

```r
## Missing values

library("visdat")

  vis_dat(nlswork)

## https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html

library(naniar)

  vis_miss(nlswork)

  gg_miss_upset(nlswork)

## GRAPHS
dplyr::glimpse(nlswork$ln_wage)
d <- density(ln_wage)
plot(d)

plot(nls$ln_wage,nls$ttl_exp)

ggplot(nlswork,
       aes(x = hours,
           y = year)) +
geom_miss_point()

ggplot(nlswork,
       aes(x = hours,
           y = year)) +
geom_miss_point() +
facet_wrap(race)

stats <- summary(nlswork$age)

library(stargazer)
stargazer(cars,
          title = "Summary table with stargazer",
          label="tab1cars",
          table.placement = "H",
          header=FALSE)
```

```
library(stargazer)
model1 <- lm(speed ~ dist, data = cars)
model2 <- lm(speed ~ dist, data = cars)
model3 <- lm(dist ~ speed, data = cars)
stargazer(model1, model2, model3,
          title = "Regression table with stargazer",
          label="tab2",
          table.placement = "H",
          column.labels = c("M1", "M2", "M3"),
          model.numbers = FALSE,
          header=FALSE)
plot(cars$speed, cars$dist)
mtcars$cyl <- as.factor(mtcars$cyl) # Convert cyl to factor
library(ggplot2)
ggplot(mtcars, aes(x=wt, y=mpg, shape=cyl)) + geom_point() +
  labs(x="Weight (lb/1000)", y = "Miles/(US) gallon",
       shape="Number of \n Cylinders") + theme_classic()
library(plotly)
p <- plot_ly(cars, type = "scatter", mode="markers",
        x=~speed,
        y=~dist)
Sys.setenv('MAPBOX_TOKEN' = '12423423') # set arbitrary token
orca(p, "plotly-plot.pdf")
import sys
print(sys.version)


import json
##from json.decoder import JSONDecodeError
import requests
import numpy as np
import pandas as pd

## INE: https://www.ine.pt/ine/json_indicador/pindica.jsp?
## op=2&varcd=0008074&Dim1=S7A2015&Dim2=200&Dim3=3&lang=PT

# api-endpoint

URL = "https://www.ine.pt/ine/json_indicador/pindica.jsp"
```

```
# define parameters

OP="2"
VARCD="0008074"
DIM1="S7A2015"
DIM2="200"
DIM3="3"
LANG="PT"


# defining a params dict for the parameters to be sent to the API
PARAMS = {'op':OP,'varcd':VARCD,'Dim1':DIM1,'Dim2':DIM2,'Dim3':DIM3,'lang':LANG}

# sending get request and saving the response as response object
r = requests.get(url = URL,params=PARAMS)

# extracting data in json format
data = r.json()

valor = data[0]['Dados']['2015'][0]['valor']

valor

x = 42 * 2
print(x)
  cd /Users/miguelportela/Documents/GitHub/prjs/pdfs
    find . -name '*.pdf' -print0 | xargs -0 -n1 pdfsandwich -gray
    find . -name '*ocr.pdf' -print0 | xargs -0 -n1 pdftotext
import os
import numpy as np
import pandas as pd
import re

## CHECK PyPDF2

## wget -A pdf -m -p -E -k -K -np https://joram.madeira.gov.pt/joram/4serie/
## find . -name '*.pdf' -print0 | xargs -0 -n1 pdfsandwich -gray
## find . -name '*ocr.pdf' -print0 | xargs -0 -n1 pdftotext

# Create list with .txt files for the specified folder
```

```
files_list = list()
for (dirpath, dirnames, filenames) in os.walk('/Users/miguelportela/Documents/bte/pdfs_t
    files_list += [os.path.join(dirpath, file)
                   for file in filenames if file.endswith('.txt')]


##print("START:FILES -- list")

##print(files_list)

##print("END:FILES -- list")

p1 = r'PORTARIA'
p2 = r'EXTENSAO'
p3 = r'Materiais'
p5 = r'PE das'

linha = []
output = []
other = []
palavra = []
source = []

for file in files_list:

    f = open(file, "r", encoding='latin8')
    data = f.read()
    f.close()

    line = []
    nh = 0

    tmp1 = str(data)
    #print(tmp1)
    tmp2 = tmp1.splitlines()
    #print(tmp2)
    for n,tmp3 in enumerate(tmp2):
        #print(tmp3)
        if (tmp3.find("PE das") == 0):
            tmp4 = tmp3 + tmp2[2]
```

```
            line.append(tmp4)
            #print(n)
            nh = 1
        elif (nh == 1):
            nh = 0
            continue
        elif (nh == 0):
            line.append(tmp3)

    print(line)

    print("   ")

    print("FILE: ", file[46:-4])

    for num, word in enumerate(line):
            if num == 0:
                continue
            else:
                match1 = re.search(p1, word)
                match2 = re.search(p2, word)
                match3 = re.search(p3, word)
                match4 = re.search(r'\d{9}', word)
                match5 = re.search(p5, word)
                ##print("   ")
                ##print("START: ",num)

                if match1:
                        ##print("   ")
                        print("match 1")
                        if match4:
                            ##print("   ")
                            print("match 4")
                            linha.append(num)
                            output.append(re.search(r'\d{9}', word).group())
                            other.append("vazio")
                            palavra.append(p1)
                            source.append(file[46:-4])
                elif match2:
                            ##print("   ")
```

```
                        print("match 2")
                        linha.append(num)
                        output.append(re.search(r'\d{9}', word).group())
                        other.append("vazio")
                        palavra.append(p2)
                        source.append(file[46:-4])
                elif match3:
                        ##print("   ")
                        print("match 3")
                        linha.append(num)
                        output.append(re.search(r'\d{9}', word).group())
                        other.append("vazio")
                        palavra.append(p3)
                        source.append(file[46:-4])
                elif match5:
                        ##print("   ")
                        print("-> match 5")
                        ##word.sub(" e o ", " e a ",1)
                        print(word)
                        linha.append(num)

                        if (word.find(" e o ") > 0):
                            print("11111")
                            output.append((word.split("re a", 1)[1]).split(" e o ",
                            other.append((word.split("re a", 1)[1]).split(" e o ",
                        elif (word.find(" e a ") > 0):
                            print("99999")
                            output.append((word.split("re a", 1)[1]).split(" e a ",
                            other.append((word.split("re a", 1)[1]).split(" e a ",

                        palavra.append(p5)
                        source.append(file[46:-4])
## o parágrafo tem de estar na mesma linha e temos de ter 'e a' em vez de 'e o'
df = pd.DataFrame({'linha': linha, 'output': output,
                   'outra': other, 'source': source})
print(df)

df.to_csv('data/PE.csv', index=False)
df.to_stata('data/PE.dta', write_index = False)
```

```
import os
import pandas as pd
import re

# Create list with .txt files for the specified folder
files_list = list()
for (dirpath, dirnames, filenames) in os.walk('/Users/miguelportela/Documents/GitHub/prj
    files_list += [os.path.join(dirpath, file) for file in filenames if file.endswith('


nipc = []



quiet cd "/Users/miguelportela/Documents/GitHub/prjs/logs"
quiet import delimited "/Users/miguelportela/Documents/GitHub/prjs/data/PE.csv", encodin
tab source



quietly{
cd /Users/miguelportela/Documents/GitHub/prjs/chunks

use nipcs, clear
compress
contract nipc
drop _freq
drop if nipc == .
format %12.0f nipc
}

codebook nipc

tab nipc



## This is a julia language chunk.
## In julia, the command without ending semicolon will trigger the display
## so is JuliaCall package.
## The julia display will follow immediately after the corresponding command
## just as the R code in R Markdown.
```

```
a = sqrt(2);
a = sqrt(2)

using ReadStat
using StatFiles
using StatsBase
using DataFrames
using FixedEffectModels

@time results_hdfe1 = reg(DataFrame(load("/Users/miguelportela/Documents/GitHub/prjs/dat

@time results_hdfe2 = reg(DataFrame(load("/Users/miguelportela/Documents/GitHub/prjs/dat



using RegressionTables
regtable(results_hdfe1,results_hdfe2; renderSettings = latexOutput("logs/hdfe_output.tex

library(JuliaCall)

julia_eval("a")

  julia_eval("results_hdfe2")

betas <- julia_eval("coef(results_hdfe2)")
r2 <- julia_eval("r2(results_hdfe2)")
use /Users/miguelportela/Documents/GitHub/prjs/data/data_short, clear

timer on 1

    reghdfe lnrealwage education lnsales,absorb(workerid firmid year)

timer off 1
timer list 1
timer clear 1
library(lfe)
data_short <- read_dta("/Users/miguelportela/Documents/GitHub/prjs/data/data_short.dta")

system.time(est_hdfe <- felm(data_short$lnrealwage ~ data_short$education + data_short$l
```

```r
summary(est_hdfe)
library(stargazer)
library(Statamarkdown)
stataexe <- "/Applications/Stata15/StataMP.app/Contents/MacOS//stata-mp"
knitr::opts_chunk$set(engine.path=list(stata=stataexe))

setwd("/Users/miguelportela/Documents/GitHub/prjs/logs")
rm(list = ls())
library(haven)
nlswork <- read_dta("../data/nlswork.dta")

auto <- read_dta("../data/auto.dta")

attach(nlswork)

regs1 <- lm(auto$price ~ auto$mpg + auto$weight)
regs2 <- lm(auto$price ~ auto$mpg + auto$weight + auto$rep78)
regs3 <- lm(auto$price ~ auto$mpg + auto$weight + auto$rep78 + auto$trunk)

regs4 <- lm(ln_wage ~ union)
regs5 <- lm(ln_wage ~ union + collgrad)
regs6 <- lm(ln_wage ~ union + collgrad + age)

##summary(auto)
##summary(regs1)

## https://www.jakeruss.com/cheatsheets/stargazer/

nls<-data.frame(nlswork)

stargazer(nls, summary.stat = c("n", "p75", "sd"), summary.logical = FALSE,
          title = "Summary table",
          label="tab23",
          table.placement = "ht",
          header=FALSE)


stargazer(regs1, regs2, regs3,
          title = "Regression table with stargazer",
          label="tab3",
```

```
          table.placement = "ht",
          column.labels = c("M1", "M2", "M3"),
          model.numbers = FALSE,
          header=FALSE,keep=c(0,1,2,3))

attach(auto)


library(naniar)
vis_miss(nlswork)

# plot(y=price,x=mpg)

library(stargazer)
stargazer(cars,
          title = "Summary 24",
          label="tab24",
          table.placement = "ht",
          header=FALSE)
quiet sysuse auto
sum price

tab rep78

global xx = r(N)

quiet cd "/Users/miguelportela/Documents/GitHub/prjs/logs"

quiet use ../data/nlswork, clear

twoway (kdensity ln_wage if collgrad == 0) || (kdensity ln_wage if collgrad == 1), scher

graph export "/Users/miguelportela/Documents/GitHub/prjs/logs/density.pdf", replace

use ../data/data_full, clear
        quiet generate lngdp = ln(rgdpwok)
        quiet ge lnk = ln(capital)

        label var rgdpwok "Real GDP per worker"
        label var education "Education (in years)"
```

```
        label var capital "Capital"
        label var open "Degree of openness"

// # regression analysis

    quiet reg lngdp education
        estimates store r1

    quiet reg lngdp education lnk
        est store r2

    reg lngdp education lnk openk i.year
        est store r3

outreg, clear
    quiet estimates restore r1
        outreg using growth_analysis_frag, tex fragment replace rtitles("Education" \ ""
                */ drop(_cons) /*
                */ ctitle("","Simple model") /*
                */ nodisplay varlabels bdec(4) se starlevels(10 5 1) starloc(1) summsta

    quiet estimates restore r2
        outreg using growth_analysis_frag, tex fragment merge rtitles("Education" \ "" \
                */ drop(_cons) /*
                */ ctitle("","Include capital") /*
                */ nodisplay varlabels bdec(3) se starlevels(10 5 1) starloc(1) summsta

    quiet estimates restore r3
        outreg using growth_analysis_frag, tex fragment merge rtitles("Education" \ "" \
                */ drop(_cons 1975.year 1980.year 1985.year 1990.year) /*
                */ ctitle("","Full model") /*
                */ nodisplay varlabels bdec(1) se starlevels(10 5 1) starloc(1) summsta


version
//ado describe

findfile xtabond2.ado
checksum "/Users/miguelportela/Library/Application Support/Stata/ado/plus/x/xtabond2.ado
```

```
cd "/Users/miguelportela/Documents/GitHub/prjs/logs"

quiet use ../data/graph_data, clear
    codebook, compact

            putexcel clear
            putexcel set descriptives.xlsx, sheet("Avg. Educ. & desc.") replace


gen first = substr(country,1,1)

    levelsof first,local(ff)

    foreach vv of local ff {

        di _new(3) "Country's first letter: `vv'"

        preserve
        quiet keep if first == "`vv'"

        quiet unique country

            if r(unique) > 5 {
            di _new(2) "    Number of countries:    " r(unique) _new(1)
            quietly {
                collapse (mean) lngdp education,by(country)
                    putexcel set descriptives.xlsx, sheet("FIRST LETTER `vv'") modify

                    regress lngdp education

                            matrix list r(table)

                        matrix results = r(table)
                            mat l results

                        mat b = results[1,1...]'
                        mat t = results[3,1...]'

                        putexcel C2="Coef." F2="t"
                        putexcel B3 = matrix(b), rownames nformat(number_d2) right
```

```
                        putexcel D3 = matrix(t),nformat("0.00")
                }
            }

            if r(unique) <= 5 {
                di _new(2) "    Insufficient number of countries; n countries = " r(uniq
            }

        restore

    }

x = 5  # radius of a circle
plot(x = mpg, y = price,
     pch = 16, frame = FALSE,
     xlab = "wt", ylab = "mpg", col = "#2E9FDF")
cat(paste("#", capture.output(sessionInfo()), "\n", collapse =""))
  # or use message() instead of cat()
```

9.3 Exploratory data analysis report

References

Arellano, Manuel. 2003. *Panel Data Econometrics*. Oxford University Press.

Arellano, Manuel and Stephen Bond. 1991. "Some Tests of Specification for Panel Data: Monte Carlo Evidence and an Application to Employment Equations." *The Review of Economic Studies* 58(2):277–97.

Arellano, Manuel and Olympia Bover. 1995. "Another Look at the Instrumental Variable Estimation of Error-Components Models." *Journal of Econometrics* 68(1):29–51.

Blundell, Richard and Stephen Bond. 1998. "Initial Conditions and Moment Restrictions in Dynamic Panel Data Models." *Journal of Econometrics* 87(1):115–43.

Hlavac, Marek. 2013. "Stargazer: LaTeX Code and Ascii Text for Well-Formatted Regression and Summary Statistics Tables." *URL: Http://CRAN. R-Project. Org/Package= Stargazer*.