Python for applied practitioners

F. Curci, F. Masera, T. Rodríguez Martínez

February 24th, 2017

Sketch presentation

How Python can be useful to applied practitioners

- Fede C
  - **What Python is and scraping**
- Fede M
  - GIS (geography and maps) application
- Tomás
  - Machine learning tools

What Python is

- Python is a widely used programming languare for general-purpose programming. It has a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java
- Free
- Users contribution: libraries
- Many platforms use Python as communicating way (becoming universal language)
- Very powerful in dealing with strings

Good way to learn Python

- Coursera course: *Python for everybody specialization. University of Michigan*
- Book: *Python for informatics*

Use

How to obtain it and language

- *Download*
- Languages currently used: Python 2 and Python 3
- *Anaconda*

Two methods

- Notepad+command prompt
  - Need to save file as .py
  - Type file.py in command prompt
- Compiler: *Pycharm*, *Atom*, *Spyder*

Web browsing

Python allows you to write a web browser

- Library to open resources by URL: urllib
- Example: we want to Python to go to web to obtain FIPS code of Texas counties
    - *Counties*

Web browsing

www2.census.gov/geo/docs/reference/codes/files/st48_tx_cou.txt

```
TX,48,001,Anderson County,H1
TX,48,003,Andrews County,H1
TX,48,005,Angelina County,H1
TX,48,007,Aransas County,H1
TX,48,009,Archer County,H1
TX,48,011,Armstrong County,H1
TX,48,013,Atascosa County,H1
TX,48,015,Austin County,H1
TX,48,017,Bailey County,H1
TX,48,019,Bandera County,H1
TX,48,021,Bastrop County,H1
TX,48,023,Baylor County,H1
TX,48,025,Bee County,H1
TX,48,027,Bell County,H1
TX,48,029,Bexar County,H1
TX,48,031,Blanco County,H1
TX,48,033,Borden County,H1
TX,48,035,Bosque County,H1
TX,48,037,Bowie County,H1
TX,48,039,Brazoria County,H1
TX,48,041,Brazos County,H1
TX,48,043,Brewster County,H1
TX,48,045,Briscoe County,H1
TX,48,047,Brooks County,H1
TX,48,049,Brown County,H1
TX,48,051,Burleson County,H1
TX,48,053,Burnet County,H1
TX,48,055,Caldwell County,H1
TX,48,057,Calhoun County,H1
TX,48,059,Callahan County,H1
TX,48,061,Cameron County,H1
TX,48,063,Camp County,H1
TX,48,065,Carson County,H1
TX,48,067,Cass County,H1
TX,48,069,Castro County,H1
TX,48,071,Chambers County,H1
TX,48,073,Cherokee County,H1
TX,48,075,Childress County,H1
TX,48,077,Clay County,H1
TX,48,079,Cochran County,H1
TX,48,081,Coke County,H1
TX,48,083,Coleman County,H1
TX,48,085,Collin County,H1
TX,48,087,Collingsworth County,H1
```

Web browsing

urlopen_county.py - [C:\Users\fcurci\AppData\Local\Temp\urlopen_county.py] - C:\Users\fcurci\Documents\UC3M\AppliedReading\Python\Codes\urlopen_county.py - PyCharm Community Edition 2016.3.2

File Edit View Navigate Code Refactor Run Tools VCS Window Help

urlopen_county.py

Project

urlopen_county.py

External Libraries

urlopen_county.py

No Python interpreter configured for the project

```
import urllib
counties=urllib.urlopen('http://www2.census.gov/geo/docs/reference/codes/files/st48_tx_cou.txt')
for line in counties:
    print line.strip()
```

Run urlopen_county

```
TX,48,421,Sherman County,H1
TX,48,423,Smith County,H1
TX,48,425,Somervell County,H1
TX,48,427,Starr County,H1
TX,48,429,Stephens County,H1
TX,48,431,Sterling County,H1
TX,48,433,Stonewall County,H1
TX,48,435,Sutton County,H1
TX,48,437,Swisher County,H1
TX,48,439,Tarrant County,H1
TX,48,441,Taylor County,H1
TX,48,443,Terrell County,H1
TX,48,445,Terry County,H1
TX,48,447,Throckmorton County,H1
TX,48,449,Titus County,H1
TX,48,451,Tom Green County,H1
TX,48,453,Travis County,H1
TX,48,455,Trinity County,H1
TX,48,457,Tyler County,H1
TX,48,459,Upshur County,H1
TX,48,461,Upton County,H1
TX,48,463,Uvalde County,H1
TX,48,465,Val Verde County,H1
TX,48,467,Van Zandt County,H1
TX,48,469,Victoria County,H1
TX,48,471,Walker County,H1
TX,48,473,Waller County,H1
TX,48,475,Ward County,H1
TX,48,477,Washington County,H1
TX,48,479,Webb County,H1
TX,48,481,Wharton County,H1
```

Web browsing

urlopen_county.py - [C:\Users\fcurci\AppData\Local\Temp\urlopen_county.py] - C:\Users\fcurci\Documents\UC3M\AppliedReading\Python\Codes\urlopen_county.py - PyCharm Community Edition 2016.3.2

File Edit View Navigate Code Refactor Run Tools VCS Window Help

urlopen_county.py

Project

urlopen_county.py

External Libraries

urlopen_county.py

No Python interpreter configured for the project

```
import urllib
counties=urllib.urlopen('http://www2.census.gov/geo/docs/reference/codes/files/st48_tx_cou.txt')
for line in counties:
    print line.strip()
```

Run urlopen_county

```
TX,48,421,Sherman County,H1
TX,48,423,Smith County,H1
TX,48,425,Somervell County,H1
TX,48,427,Starr County,H1
TX,48,429,Stephens County,H1
TX,48,431,Sterling County,H1
TX,48,433,Stonewall County,H1
TX,48,435,Sutton County,H1
TX,48,437,Swisher County,H1
TX,48,439,Tarrant County,H1
TX,48,441,Taylor County,H1
TX,48,443,Terrell County,H1
TX,48,445,Terry County,H1
TX,48,447,Throckmorton County,H1
TX,48,449,Titus County,H1
TX,48,451,Tom Green County,H1
TX,48,453,Travis County,H1
TX,48,455,Trinity County,H1
TX,48,457,Tyler County,H1
TX,48,459,Upshur County,H1
TX,48,461,Upton County,H1
TX,48,463,Uvalde County,H1
TX,48,465,Val Verde County,H1
TX,48,467,Van Zandt County,H1
TX,48,469,Victoria County,H1
TX,48,471,Walker County,H1
TX,48,473,Waller County,H1
TX,48,475,Ward County,H1
TX,48,477,Washington County,H1
TX,48,479,Webb County,H1
TX,48,481,Wharton County,H1
```

Ways to extract information from the web

API (Application programming interface)

- Structured way to access data from websites
- Usually limited number of queries
- Not all website provide it
- Twitter, Google maps, etc.
- Some Stata commands

Scraping

- Transformation of unstructured data (HTML format) on the web into structured data (database)
- Stata command: copy

API

Talk to web services: URLs that are designed explicitly to hand data back cleaned for your application

- Need to agree on wire protocol for python and java

Two wire formats to exchange data between applications

- XML
- JSON

Libraries

- xml.etree.ElementTree
- json

Google map API

- Possible to retrieve address, latitude and longitude of places
- *Documentation*
- Limits
  - 2,500 requests per day
  - 50 requests per second

Google map API

geojson_loop.py - [C:\Users\fcurci\AppData\Local\Temp\geojson_loop.py] - C:\Users\fcurci\Documents\UC3M\AppliedReading\Python\Codes\geojson_loop.py - PyCharm Community Edition 2016.3.2

File Edit View Navigate Code Refactor Run Tools VCS Window Help

geojson_loop.py

Project

geojson_loop.py

External Libraries

geojson_loop.py

No Python interpreter configured for the project

```python
import urllib
import json

universidades=["Universidad de Alcala","Universidad Autonoma de Madrid","Universidad Carlos III de Madrid",

serviceurl = 'http://maps.googleapis.com/maps/api/geocode/json?'
# serviceurl = 'http://python-data.dr-chuck.net/geojson?'

for universidad in universidades:
    address = universidad
    url = serviceurl + urllib.urlencode({'sensor':'false', 'address': address})
    print 'Retrieving', url
    uh = urllib.urlopen(url)
    data = uh.read()
    try: js = json.loads(str(data))
    except: js = None
    if 'status' not in js or js['status'] != 'OK':
        print '==== Failure To Retrieve ====='
        continue
    lat = js["results"][0]["geometry"]["location"]["lat"]
    lng = js["results"][0]["geometry"]["location"]["lng"]
    print 'lat',lat,'lng',lng
    location = js['results'][0]['formatted_address']
    print universidad +','+ location
```

Run geojson_loop

```
Retrieving http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Universidad+de+Alcala
lat 40.4824722 lng -3.3628674
Universidad de Alcala,Plaza de San Diego, s/n, 28801 Alcalá de Henares, Madrid, Spain
Retrieving http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Universidad+Autonoma+de+Madrid
lat 40.5466983 lng -3.6943619
Universidad Autonoma de Madrid,Ciudad Universitaria de Cantoblanco, 28049 Madrid, Spain
Retrieving http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Universidad+Carlos+III+de+Madrid
lat 40.316966 lng -3.7270795
Universidad Carlos III de Madrid,Calle Madrid, 126, 28903 Getafe, Madrid, Spain
Retrieving http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Universidad+Complutense+de+Madrid
lat 40.4478246 lng -3.7285872
```

Google map API

geojson_loop.py - [C:\Users\fcurci\AppData\Local\Temp\geojson_loop.py] - C:\Users\fcurci\Documents\UC3M\AppliedReading\Python\Codes\geojson_loop.py - PyCharm Community Edition 2016.3.2

File Edit View Navigate Code Refactor Run Tools VCS Window Help

geojson_loop.py

Project

geojson_loop.py

External Libraries

geojson_loop.py

No Python interpreter configured for the project

```python
import urllib
import json

universidades=["Universidad de Alcala","Universidad Autonoma de Madrid","Universidad Carlos III de Madrid"]

serviceurl = 'http://maps.googleapis.com/maps/api/geocode/json?'
# serviceurl = 'http://python-data.dr-chuck.net/geojson?'

for universidad in universidades:
    address = universidad
    url = serviceurl + urllib.urlencode({'sensor':'false', 'address': address})
    print 'Retrieving', url
    uh = urllib.urlopen(url)
    data = uh.read()
    try: js = json.loads(str(data))
    except: js = None
    if 'status' not in js or js['status'] != 'OK':
        print '==== Failure To Retrieve ====='
        continue
    lat = js["results"][0]["geometry"]["location"]["lat"]
    lng = js["results"][0]["geometry"]["location"]["lng"]
    print 'lat',lat,'lng',lng
    location = js['results'][0]['formatted_address']
    print universidad +','+ location
```

Run geojson_loop

```
Retrieving http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Universidad+de+Alcala
lat 40.4824722 lng -3.3628674
Universidad de Alcala,Plaza de San Diego, s/n, 28801 Alcalá de Henares, Madrid, Spain
Retrieving http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Universidad+Autonoma+de+Madrid
lat 40.5466983 lng -3.6943619
Universidad Autonoma de Madrid,Ciudad Universitaria de Cantoblanco, 28049 Madrid, Spain
Retrieving http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Universidad+Carlos+III+de+Madrid
lat 40.316966 lng -3.7270795
Universidad Carlos III de Madrid,Calle Madrid, 126, 28903 Getafe, Madrid, Spain
Retrieving http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Universidad+Complutense+de+Madrid
lat 40.4478246 lng -3.7285872
```

Web scraping

HTML

- Standard markup language for creating web pages and web applications. Messy

```
<html>
    <head>
        <title>This is the title</title>
    </head>
    <body>
        <a href ="www.wikipedia.es" > Wikipedia link</a>
    </body>
</html>
```

Web scraping

HTML

- Standard markup language for creating web pages and web applications. Messy

```
<html>
    <head>
        <title>This is the title</title>
    </head>
    <body>
        <a href="www.wikipedia.es">Wikipedia link</a>
    </body>
</html>
```

Start tag

Element

End tag

Attribute name

Attribute value

Content

Web scraping

HTML

- Standard markup language for creating web pages and web applications. Messy

```
<html>
    <head>
        <title>This is the title</title>
    </head>
    <body>
        <a href="www.wikipedia.es">Wikipedia link</a>
    </body>
</html>
```

Start tag: `<head>`

Element: `<title>This is the title</title>`

End tag: `</head>`

Attribute name: `href`

Attribute value: `"www.wikipedia.es"`

Content: `Wikipedia link`

- Possible to retrieve all html with python
- Libraries: *BeautifulSoup*, *HTMLParser*

Looking for house prices

https://www.idealista.com/venta-viviendas/madrid-madrid/mapa

idealista

Pon tu an

idealista > Madrid provincia > Madrid

64.692 27.053

Inmobiliarias en Madrid

Mapa de Madrid, ver las 27.054 viviendas »

Dibujar tu propia zona

Mapa de Madrid, ver las 27.054 viviendas »

Y además 245 promos de obra nueva

Áreas de Madrid

Arganzuela (763)

Looking for house prices

https://www.idealista.com/venta-viviendas/madrid-madrid/mapa

Mapa de Madrid, ver las 27.054 viviendas »

Y además 245 promos de obra nueva

Áreas de Madrid

Arganzuela (763)
Barajas (399)
Carabanchel (1.7
Centro (1.331)
Chamartín (2.10
Chamberí (1.338
Ciudad Lineal (1
Fuencarral (1.494)
Hortaleza (1.798)
Latina (1.188)
Moncloa (2.041)

Abrir enlace en una pestaña nueva
Abrir enlace en una ventana nueva
Abrir enlace en una nueva ventana privada
Añadir este enlace a marcadores
Guardar enlace como...
Guardar enlace en Pocket
Copiar la ruta del enlace
Buscar "Arganzuela (763..." en Google
Inspeccionar elemento

Inspector | Consola | Depurador | Editor de es... | Rendimiento | Memoria | Red

2 de 27 area

```
<div id="map-title-mobile" class="d-none"></div>
<div id="map-template" class="d-none"></div>
<div id="grey-map-wrapper">
  <map id="map-mapping" name="map-mapping">
    <area shape="poly" coords="126,284,132,284,132,294,136,300,142,305,155,308,169,308,175,…,334,162,331,153,328,146,323,140,317,127,317,126,312,126,284"
    href="/venta-viviendas/madrid/arganzuela/mapa" data="Arganzuela, 763 viviendas">
    <area shape="poly" coords="385,29,428,29,437,25,442,18,448,16,448,96,448,176,382,175,38…,103,333,97,337,92,346,86,356,75,365,56,383,38,385,31,385,29"
    href="/venta-viviendas/madrid/barajas/mapa" data="Barajas, 399 viviendas">
```

Looking for house prices

https://www.idealista.com/venta-viviendas/madrid-madrid/mapa

Mapa de Madrid, ver las 27.054 viviendas »

Y además 245 promos de obra nueva

Áreas de Madrid

Arganzuela (763)
Barajas (399)
Carabanchel (1.7
Centro (1.331)
Chamartín (2.10
Chamberí (1.338
Ciudad Lineal (1
Fuencarral (1.494)
Hortaleza (1.798)
Latina (1.188)
Moncloa (2.041)

Abrir enlace en una pestaña nueva
Abrir enlace en una ventana nueva
Abrir enlace en una nueva ventana privada
Añadir este enlace a marcadores
Guardar enlace como...
Guardar enlace en Pocket
Copiar la ruta del enlace
Buscar "Arganzuela (763..." en Google
Inspeccionar elemento

Inspector | Consola | Depurador | Editor de es... | Rendimiento | Memoria | Red

2 de 27 area

```
<div id="map-title-mobile" class="d-none"></div>
<div id="map-template" class="d-none"></div>
<div id="grey-map-wrapper">
<map id="map-mapping" name="map-mapping">
<area shape="poly" coords="126,284,132,284,132,294,136,300,142,305,155,308,169,308,175,…,334,162,331,153,328,146,323,140,317,127,317,126,312,126,284"
href="/venta-viviendas/madrid/arganzuela/mapa" data="Arganzuela, 763 viviendas">
<area shape="poly" coords="385,29,428,29,437,25,442,18,448,16,448,96,448,176,382,175,38…,103,333,97,337,92,346,86,356,75,365,56,383,38,385,31,385,29"
href="/venta-viviendas/madrid/barajas/mapa" data="Barajas, 399 viviendas">
```

Looking for house prices

```
<div id="map-title-mobile" class="d-none"></div>
<div id="map-template" class="d-none"></div>
<div id="grey-map-wrapper">
  <map id="map-mapping" name="map-mapping">
    <area shape="poly" coords="126,284,132,284,132,294,136,300,142,305,155,308,169,308,175,…,334,162,331,153,328,146,323,140,317,127
    href="/venta-viviendas/madrid/arganzuela/mapa" data="Arganzuela, 763 viviendas">
    <area shape="poly" coords="385,29,428,29,437,25,442,18,448,16,448,96,448,176,382,175,38…,103,333,97,337,92,346,86,356,75,365,56,
    href="/venta-viviendas/madrid/barajas/mapa" data="Barajas, 399 viviendas">
    <area shape="poly" coords="23,433,39,414,44,397,52,379,43,368,41,365,46,364,57,370,62,3…419,112,419,97,425,86,426,73,426,52,433,
    href="/venta-viviendas/madrid/carabanchel/mapa" data="Carabanchel, 1.732 viviendas">
```

Looking for house prices

File Edit View Navigate Code Refactor Run Tools VCS Window Help

2_idealista_barrio.py

Project

2_idealista_barrio.py

External Libraries

2_idealista_barrio.py

No Python interpreter configured for the project

```python
import urllib
import time
from BeautifulSoup import *

url='https://www.idealista.com/venta-viviendas/madrid-madrid/mapa'
html = urllib.urlopen(url).read()
soup = BeautifulSoup(html)
tags = soup('area')
barrios=list()
href_barrios=list()
list_barrio_href=list()
for tag in tags:
    colindancias=tag.get('h4', None)
    barrio=tag.get('data', None)
    firstpos=barrio.find(',')
    secondpos=barrio.find('viviendas')
    #Avoid to take neighbouring cities (colindancias)
    if firstpos==len(barrio)-1:
        continue
    else:
        barrios.append(barrio[0:firstpos])
        href_barrio=tag.get('href',None)
        href_barrios.append(href_barrio)
        list_barrio_href.append([barrio[0:firstpos],barrio[firstpos+1:secondpos],href_barrio])

for barrio in list_barrio_href:
    for minibarrio in barrio:
        print minibarrio
```

Run 2_idealista_barrio

```
C:\Python27\python.exe C:/Users/fcurci/Documents/UC3M/AppliedReading/Python/Codes/2_idealista_barrio.py
Arganzuela
 763
/venta-viviendas/madrid/arganzuela/mapa
Barajas
 399
/venta-viviendas/madrid/barajas/mapa
Carabanchel
 1.732
/venta-viviendas/madrid/carabanchel/mapa
```

Looking for house prices

File Edit View Navigate Code Refactor Run Tools VCS Window Help

2_idealista_barrio.py

Project

2_idealista_barrio.py

External Libraries

2_idealista_barrio.py

No Python interpreter configured for the project

```python
import urllib
import time
from BeautifulSoup import *

url='https://www.idealista.com/venta-viviendas/madrid-madrid/mapa'
html = urllib.urlopen(url).read()
soup = BeautifulSoup(html)
tags = soup('area')
barrios=list()
href_barrios=list()
list_barrio_href=list()
for tag in tags:
    colindancias=tag.get('h4', None)
    barrio=tag.get('data', None)
    firstpos=barrio.find(',')
    secondpos=barrio.find('viviendas')
    #Avoid to take neighbouring cities (colindancias)
    if firstpos==len(barrio)-1:
        continue
    else:
        barrios.append(barrio[0:firstpos])
        href_barrio=tag.get('href',None)
        href_barrios.append(href_barrio)
        list_barrio_href.append([barrio[0:firstpos],barrio[firstpos+1:secondpos],href_barrio])

for barrio in list_barrio_href:
    for minibarrio in barrio:
        print minibarrio
```

Run 2_idealista_barrio

```
C:\Python27\python.exe C:/Users/fcurci/Documents/UC3M/AppliedReading/Python/Codes/2_idealista_barrio.py
Arganzuela
 763
/venta-viviendas/madrid/arganzuela/mapa
Barajas
 399
/venta-viviendas/madrid/barajas/mapa
Carabanchel
 1.732
/venta-viviendas/madrid/carabanchel/mapa
```

Looking for house prices

https://www.idealista.com/venta-viviendas/madrid/arganzuela/acacias/

two formulas pause latex

Nuevos anuncios en tu email:

Guardar búsqueda

Viviendas

Precio

Mín Máx

Tamaño

Mín Máx

Tipo de vivienda

- Pisos
- Casas o chalets
- Casas rústicas
- Dúplex
- Áticos

Habitaciones

- 0 habitaciones (estudios)

Listado Mapa

Ordenar: Relevancia Baratos Recientes Más

Destacado

Piso en calle de Martín de Vargas, Acacias, Madrid

315.000 €

3 hab. 135 m² 4ª planta exterior con ascensor

"Inmobiliarias Encuentro vende piso todo exterior de 135 metros cuadrados, consta de 3 dormitorios amplios, 1 cuartos de baño comple...

912 664 063

Destacado

Piso en calle Majuelo, 3, Acacias, Madrid

429.000 € Garaje incluido

2 hab. 85 m² 2ª planta exterior con ascensor

De las mejores urbanizaciones de Pasillo Verde. Piso espectacular con calidades de primera. La cocina de diseño super equipada con tended...

916 350 423

GILMAR CONSULTING INMOBILIARIO ARGANZUELA | PASILLO VERDE

Destacado

Dúplex en paseo de las acacias, 6, Acacias, Madrid

358.500 €

2 hab. 103 m² 5ª planta exterior con ascensor

Inspector Consola Depurador Editor de es... Rendimiento Memoria Red

Buscar en HTML

```
<div class="logo-branding"></div>
<a class="item-link " href="/inmueble/35062114/" title="Piso en calle de Martín de Vargas, Acacias, Madrid" data-xiti-click="listado::enlace">
Piso en calle de Martín de Vargas, Acacias, Madrid</a>
<div class="row price-row clearfix">
  ::before
  <span class="item-price">
    315.000
    <span>€</span>
  </span>
  ::after
</div>
```

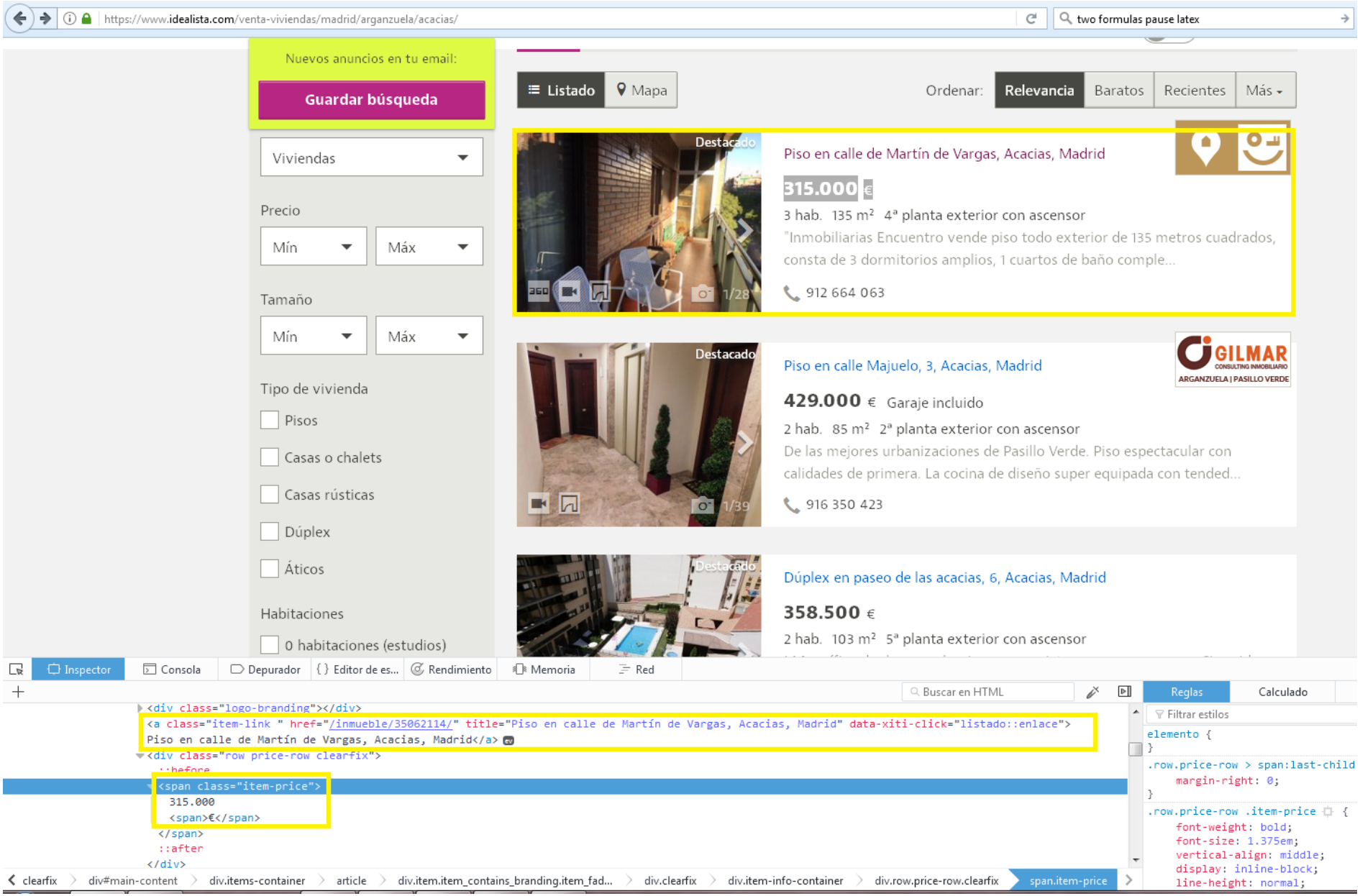
Reglas Calculado

Filtrar estilos

```
elemento {
}
.row.price-row > span:last-child
    margin-right: 0;
}
.row.price-row .item-price {
    font-weight: bold;
    font-size: 1.375em;
    vertical-align: middle;
    display: inline-block;
    line-height: normal;
```

clearfix > div#main-content > div.items-container > article > div.item.item_contains_branding.item_fad... > div.clearfix > div.item-info-container > div.row.price-row.clearfix > span.item-price

Looking for house prices



Looking for house prices

```
<div class="logo-branding"></div>
<a class="item-link " href="/inmueble/35062114/" title="Piso en calle de Martín de Vargas, Acacias, Madrid" data-xiti-click="li
Piso en calle de Martín de Vargas, Acacias, Madrid</a> ev
<div class="row price-row clearfix">
  ::before
  <span class="item-price">
    315.000
    <span>€</span>
  </span>
  ::after
</div>
```

Looking for house prices

File Edit View Navigate Code Refactor Run Tools VCS Window Help

idealista_flat.py

Project

idealista_flat.py

External Libraries

idealista_flat.py

No Python interpreter configured for the project

```
ascensor=list()
for tag in tags:
    #Obtain link of house
    class_link=tag.get('class',None)
    class_link2=str(class_link)
    if 'item-price-down icon-pricedown' in class_link2:
        continue
    elif 'item-price' in class_link2:
        #Obtain title price
        parser = myhtmlparser()
        tag2=str(tag)
        parser.feed(tag2)
            # Extract data from parser
        tags  = parser.NEWTAGS
        attrs = parser.NEWATTRS
        precio  = parser.HTMLDATA
            # Clean the parser
        parser.clean()
            # Print out our data
        precios.append(int(precio[0].replace('.','')))
    elif 'item-detail' in class_link2:
        #Obtain info
        parser = myhtmlparser()
        tag2=str(tag)
        parser.feed(tag2)
            # Extract data from parser
        tags  = parser.NEWTAGS
        attrs = parser.NEWATTRS
        detail  = parser.HTMLDATA
            # Clean the parser
        parser.clean()
            # Print out our data
```

Run idealista_flat

```
C:\Python27\python.exe C:/Users/fcurci/Documents/UC3M/AppliedReading/Python/Codes/idealista_flat.py
[35062114, 'Piso', 'calle de Mart\xc3\xadn de Vargas', -999999, 'Arganzuela', 'Acacias', 315000, 135, '4', 'exterior', 'con ascensor', '22/02/2017']
[35841803, 'Piso', 'calle Majuelo', 3, 'Arganzuela', 'Acacias', 429000, 85, '2', 'exterior', 'con ascensor', '22/02/2017']
[34128182, 'D\xc3\xbaplex', 'paseo de las acacias', 6, 'Arganzuela', 'Acacias', 358500, 103, '5', 'exterior', 'con ascensor', '22/02/2017']
[34808794, 'Piso', 'ronda DE TOLEDO', -999999, 'Arganzuela', 'Acacias', 340000, 86, '4', 'exterior', 'con ascensor', '22/02/2017']
```

Looking for house prices

File Edit View Navigate Code Refactor Run Tools VCS Window Help

idealista_flat.py

Project

idealista_flat.py

External Libraries

idealista_flat.py

No Python interpreter configured for the project

```python
ascensor=list()
for tag in tags:
    #Obtain link of house
    class_link=tag.get('class',None)
    class_link2=str(class_link)
    if 'item-price-down icon-pricedown' in class_link2:
        continue
    elif 'item-price' in class_link2:
        #Obtain title price
        parser = myhtmlparser()
        tag2=str(tag)
        parser.feed(tag2)
            # Extract data from parser
        tags  = parser.NEWTAGS
        attrs = parser.NEWATTRS
        precio  = parser.HTMLDATA
            # Clean the parser
        parser.clean()
            # Print out our data
        precios.append(int(precio[0].replace('.','')))
    elif 'item-detail' in class_link2:
        #Obtain info
        parser = myhtmlparser()
        tag2=str(tag)
        parser.feed(tag2)
            # Extract data from parser
        tags  = parser.NEWTAGS
        attrs = parser.NEWATTRS
        detail  = parser.HTMLDATA
            # Clean the parser
        parser.clean()
            # Print out our data
```

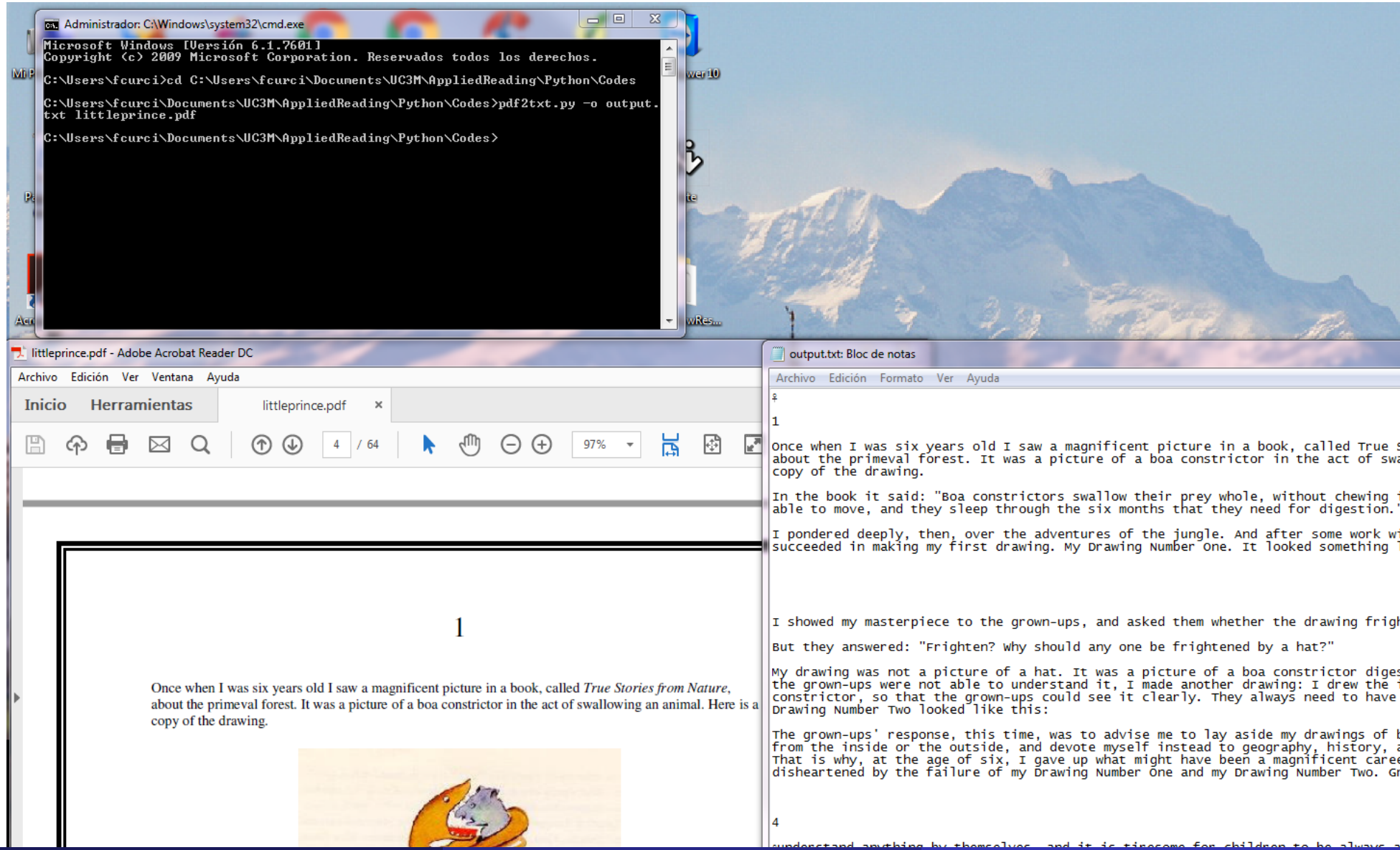
Run idealista_flat

```
C:\Python27\python.exe C:/Users/fcurci/Documents/UC3M/AppliedReading/Python/Codes/idealista_flat.py
[35062114, 'Piso', 'calle de Mart\xc3\xadn de Vargas', -999999, 'Arganzuela', 'Acacias', 315000, 135, '4', 'exterior', 'con ascensor', '22/02/2017']
[35841803, 'Piso', 'calle Majuelo', 3, 'Arganzuela', 'Acacias', 429000, 85, '2', 'exterior', 'con ascensor', '22/02/2017']
[34128182, 'D\xc3\xbaplex', 'paseo de las acacias', 6, 'Arganzuela', 'Acacias', 358500, 103, '5', 'exterior', 'con ascensor', '22/02/2017']
[34808794, 'Piso', 'ronda DE TOLEDO', -999999, 'Arganzuela', 'Acacias', 340000, 86, '4', 'exterior', 'con ascensor', '22/02/2017']
```

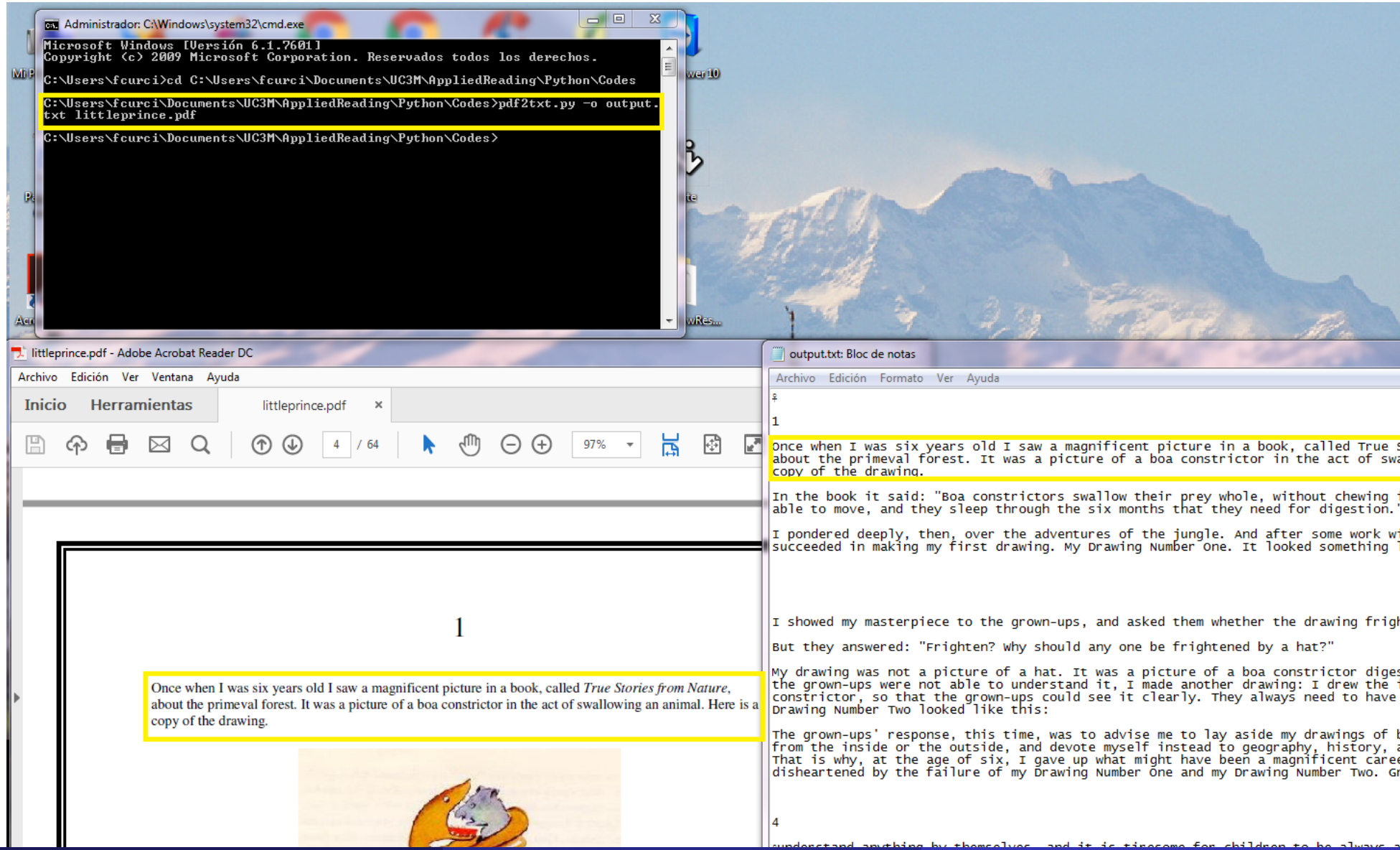
PDF Scraping

- Easier when PDF is not an image
- Possible to convert entire PDF to txt file
- Possible to dump the internal contents of a PDF file in pseudo-XML format
- Library: *PDFMiner*

Scraping the little prince



Scraping the little prince



Sketch presentation

How Python can be useful to applied practitioners

- Fede C
  - What Python is and scraping
- Fede M
  - **GIS (geography and maps) application**
- Tomás
  - Machine learning tools

Introduction

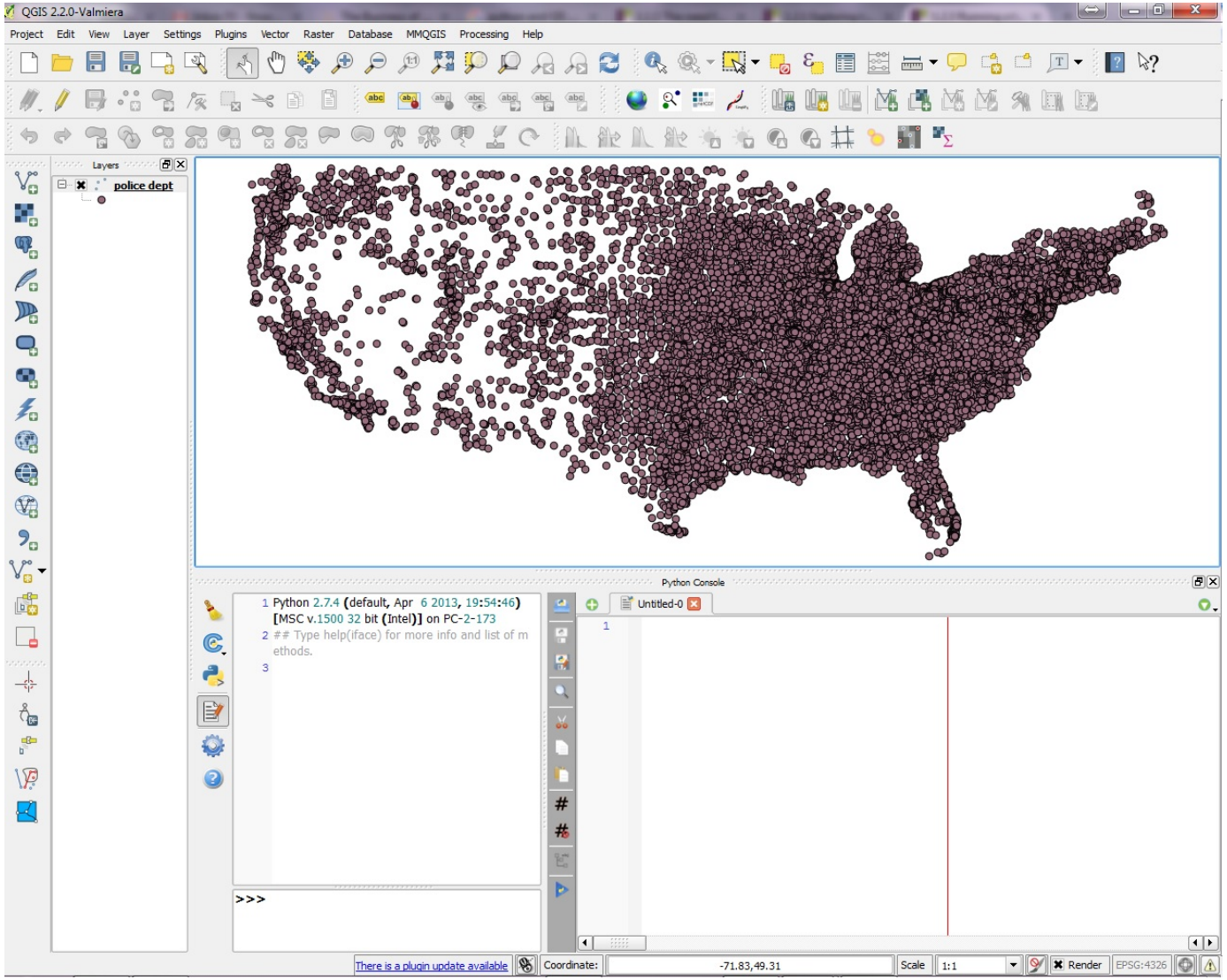
Python can be used both in QGIS and ArcGIS as a language to interact with these programs.

These will be useful to:

- Automated processes that you can already do by clicking in your preferred GIS program. (Very important for using data from STATA or creating data for STATA use)
- STATA can automate many of this operations already (not all of them) but GIS programs much more efficient
- Perform some new function only available if using Python

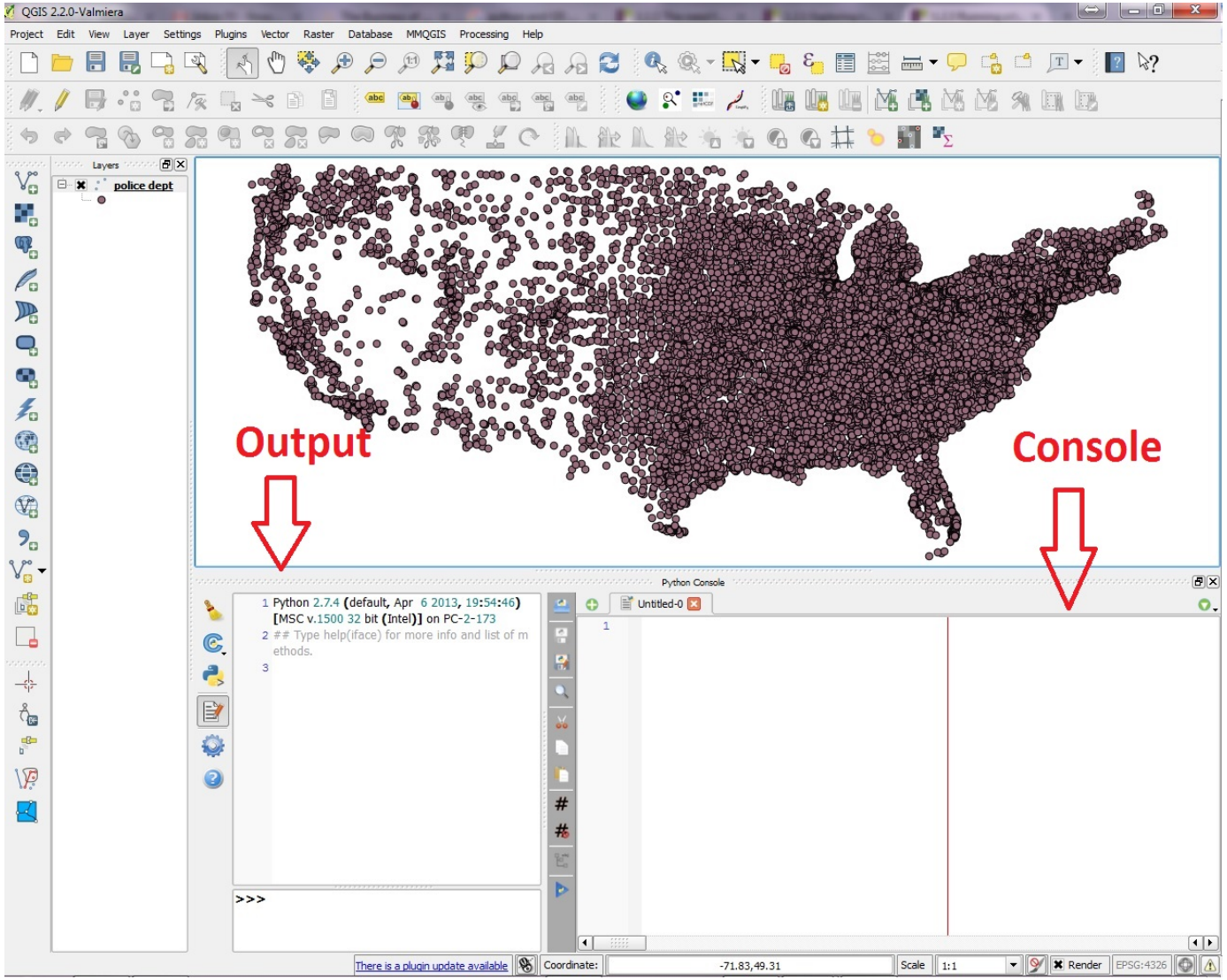
Where to Find Python? (QGIS)

Plugins ⇒ Python Console



Where to Find Python? (QGIS)

Plugins ⇒ Python Console



Learn to Write GIS Python Code

1. Follow the course found in *qgistutorials.com*
2. Search in *stackexchange.com*
3. Processing ⇒ “Graphical modeler” (a drag and drop tool that generates a Python Code)
4. Write your own .py file

Example 1 - Automation of Presentation of Results

You have shapefiles of all the crimes in counties in the US 1960-2014. For exploring the data you want to create a map for each year. Change the color of places with a high violent crime rate.

You can write your own .py code:

1. For loop for each file (for yy in year_available)
2. open .shp file - .addMapLayer( )
3. select high crime places - .getFeatures( )
4. Change color - .setSelectionColor( )
5. save map

If you have multiple maps you can even created videos of maps

Example 2 - Automation Getting Inputs from STATA

Example: Data on number of police officers for all police departments and want to perform some operation with QGIS using this data.
For QGIS to read data from STATA needs a .csv file with an identifier (ID of police department) and the variable of interest (number of police officers)

Python can easily automate all this in case you make some change to any variable you do not need to manually redo everything:

1. Open .shp file and .csv files - .addMapLayer( )
2. Join the 2 files using the common identifier - .addJoin
3. Perform whatever operation you wish

Example 3 - Summary Statistics by Distance

Calculate number of police officers in a radius of 20km around a police department.
This could be done in STATA by calculating the distance of all police dept. from all police dept. and summing up the number of police officers if the distance is less than 20km. This is computationally unfeasible (15000 police dept.)

QGIS is very efficient when dealing with GIS data and distances. We can write a code on Python:

1. Open the file we created in example 2 - .addMapLayer( )
2. Create a buffer of 20km around all police departments - .buffer()
3. Select only police departments that intersect with the buffer - .intersects()
4. Sum up police officers if it intersects and save in a new variable

Example 3 - Summary Statistics by Distance

Why is it useful?

1. Much faster than STATA (sometimes is just unfeasible on STATA)
2. Once you have the Python code is easy to change buffers or variables
3. You can then save the data and use this newly created variable back in STATA - .writeAsVectorFormat()

Example 4 - Neighbors of Polygons

All the previous examples the GIS data was points (police department). Many times data is in polygons (Counties, States, ...)

Let's say that we have population level data at the county level. We want to calculate the total population of the neighbors of all counties (where neighbors means to have one border in common). Python can do that.

Is a long code but the main tricky thing is how to determine neighbors. You do with the following function - .intersects(geom.boundingBox())

Again this new variable created in QGIS can be then exported in STATA

A Taste of More Complex Things you Can Do in Python

Excellent Guide: “Night Lights and ArcGIS: A Brief Guide” by Matt Lowe

Let’s say you want to create a 20kmX20Km pixel-year panel of luminosity.

You can download a raster NASA NOOA satellite data of luminosity at night of the earth (data for many years and many types of measurements).
You can do it in Python with .CreateFishnet_management()

Given that the data is already on Python you can add a lot of information to this pixels dataset (Country, Distance to the coast, ...) and then save it in a readable file for STATA

Sketch presentation

How Python can be useful to applied practitioners

- Fede C
  - What Python is and scraping
- Fede M
  - GIS (geography and maps) application
- Tomás
  - **Machine learning tools**

Machine Learning with Scikit-Learn

- **Idea:** How to use a couple of methods of Imbens/Athey NBER summer lectures with the library scikit-learn
- **Machine Learning...**
  - ...is more about fit/prediction.
  - ..cares a lot about scalability of the algorithm (big data, many covariates).
  - ...is neither about causality or inference (Susan Athey disagrees).
  - ...nor the formal asymptotic properties.
- **Focus on:** Out of sample performance, cross-validation, regularization.
- Why Python and Scikit-learn?

How to learn?

- Machine Learning:
  - **Coursera:** Stanford course with Andrew NG (matlab), Washington University specialization (python).
  - **edX:** Columbia course from the MicroMasters Program.
  - **Book:** *The Elements of Statistical Learning. Data mining, Inference and Prediction.* Trevor Hastie, Robert Tibshirani and Jerome Friedman.
- Basic packages to handle data on Python:
  - **Panda (Most popular library for data manipulation):** Resources.
  - **Numpy (Efficient python matrix handler):** Tutorial, lecture notes.
- Scikit-Learn:
  - **Links:** Official Tutorial, video lectures, other Tutorial .
  - **Book:** Learning scikit-learn: Machine Learning in Python.

Ex.1 Classification + Regularization

- **Built in scikit learn:** Lasso, Ridge regression, Least angle regression, Lars, Elastic Nets.
- **Example:** Reviews from an Amazon product. Which word has the highest predictive power of a positive review?
- **Method:** Regularized logistic regression.
- **Scikit implementation:** First, get your matrix of words using the *CountVectorizer*, then use *LogisticRegressionCV* with a Lasso like penalty lasso.
- *LogisticRegressionCV* has a built in cross validation to find the "optimal" penalty parameter α.

Ex.1 Classification + Regularization

- 166752 data points with 141224 covariates! “Only” around 3300 are nonzero.

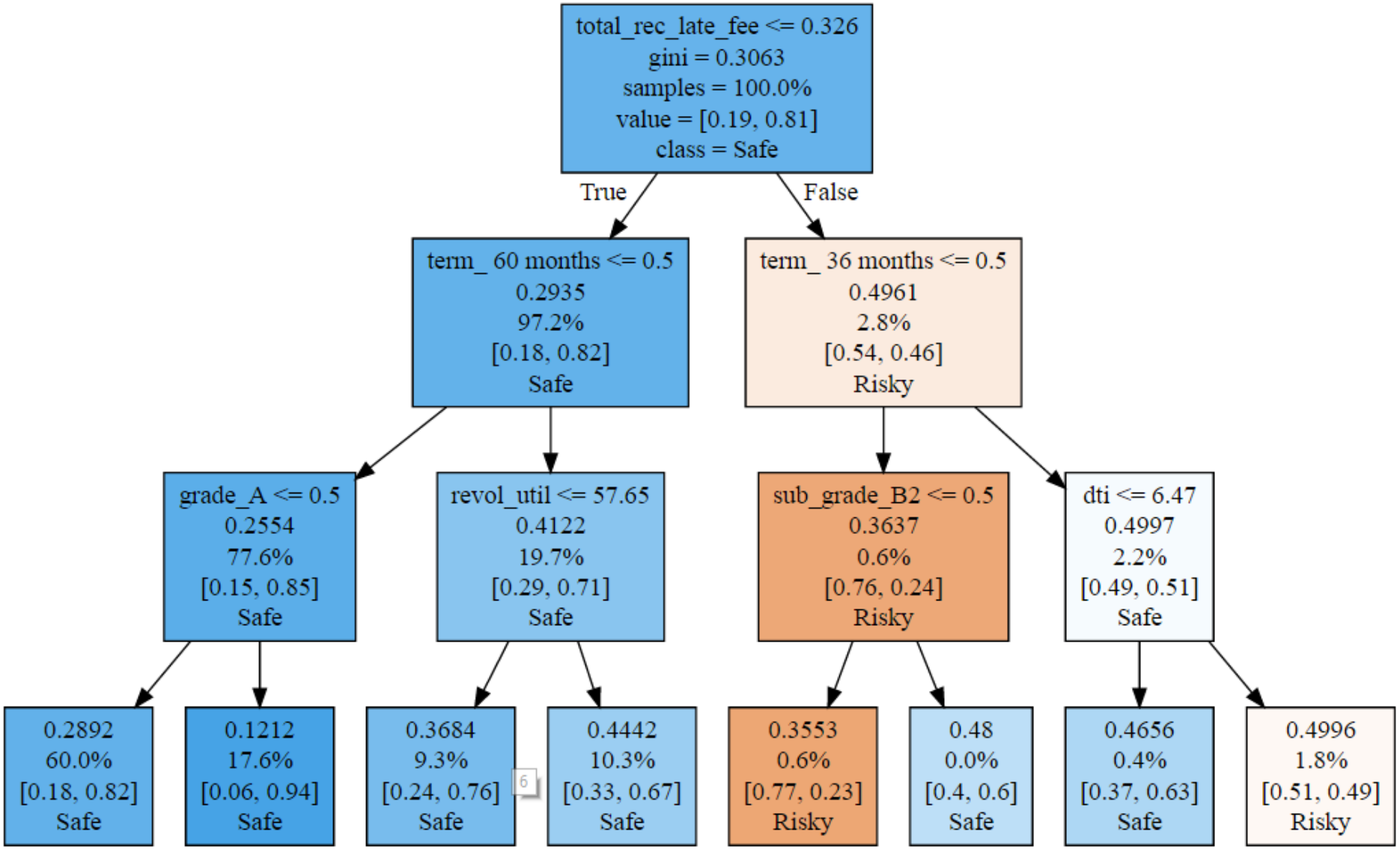
	Positive Words		Negative Words	
1	oustanding	2.12	dissapointed	-2.58
2	saves	2.03	worst	-2.47
3	lifesaver	1.90	worthless	-2.40
4	skeptical	1.80	theory	-2.38
5	adores	1.79	unusable	-2.33
6	south	1.69	disappointing	-2.26
7	con	1.67	ripoff	-2.25
8	penny	1.65	poorly	-2.22
9	awesome	1.63	ineffective	-2.20
10	pleasantly	1.63	useless	-2.02

Ex.2 Regression Trees

- **Built in scikit learn:** Random Forests, Bagging, Boosting (Adaboost).
- **Example:** When to give a loan? Data on good/bad loans + covariates of the individual (employment, debt to income ratio...).
- **Method:** Random Forests: Fit different regression trees in random subsamples with replacement and apply a weighted average between them.
- **Scikit implementation:** *RandomForestClassifier*

```
forest =RandomForestClassifier(max_depth=4).fit(X,Y)
forest.predict_proba(X[1,:])
print forest.feature_importances_
```

Ex.2 Regression Trees

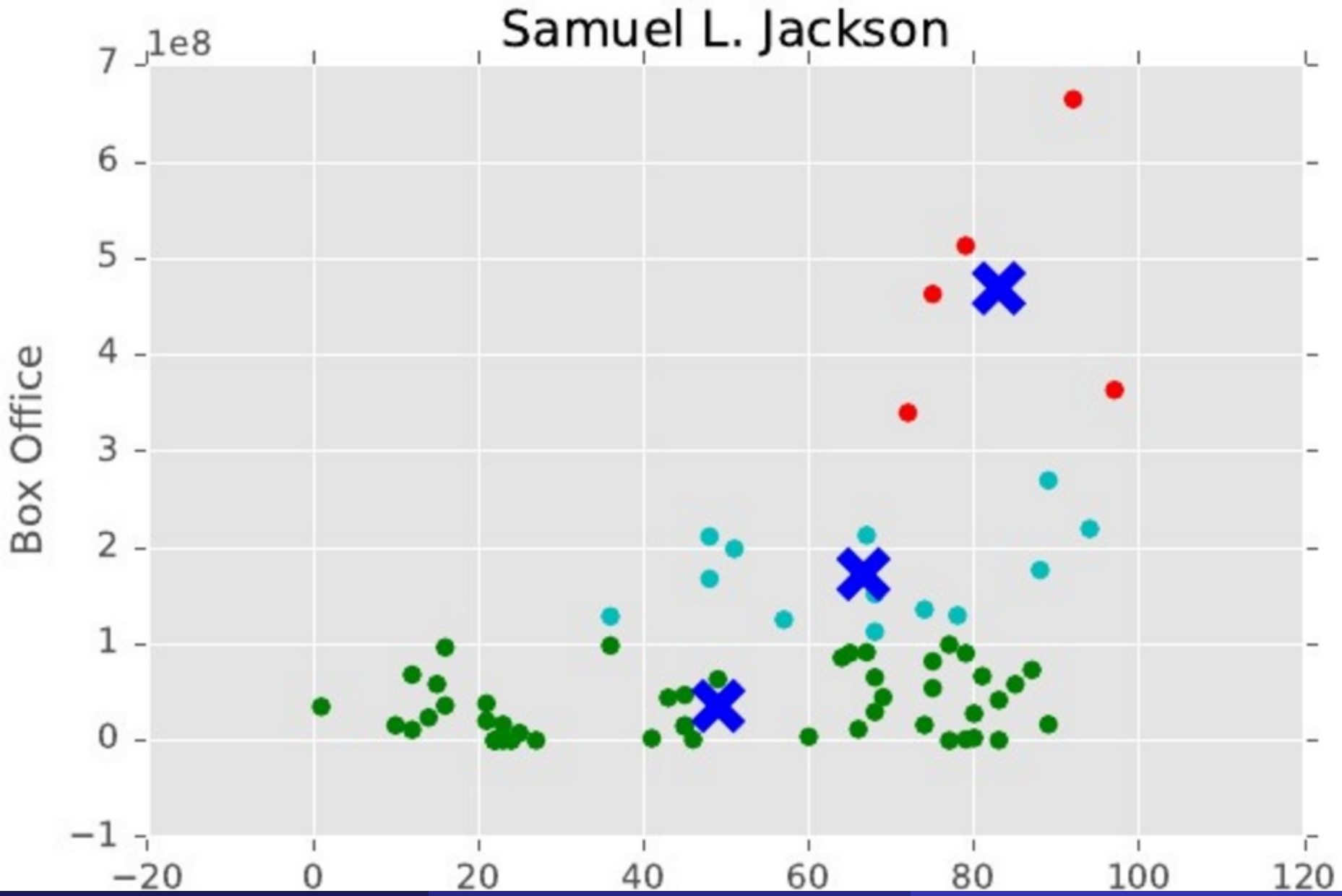


Ex.3 Clustering

- **Built in scikit learn:** K-means, Gaussian Mixtures.
- **Example:** Clustering Samuel L. Jackson movies by quality and revenue box office.
- **Method:** K-Means.
- **Scikit implementation:** *KMeans*

```
kmeans = KMeans(n_clusters=3,random_state=10).fit(data)
centroids = kmeans.cluster_centers_
labels = kmeans.labels_
```

Ex.3 Clustering



Code Appendix

Classifier + Regularization:

```
# Word Matrix
vectorizer = CountVectorizer(token_pattern=r'\b\w+\b')
words_matrix = vectorizer.fit_transform(products['review_clean
name = vectorizer.get_feature_names()

# Model

sentiment_model = linear_model.LogisticRegressionCV(penalty='l1
                                ...,solver='liblinear')
sentiment_model.fit(words_matrix, dependent_var)
coefficients = sentiment_model.coef_
```

Appendix 1

Lasso

$$\min_{\beta} \sum_{i}^{N} (Y_i - X_i \beta)^2 + \alpha ||\beta||_1 \tag{1}$$