Analysis of BCM seminars

March 30, 2016

0.1 Analysis of BCM seminars (this is markdown !)

Figure 1: Timc Logo

First we need to use the right libraries to gather the data !

```
In [1]: from lxml import html
        import requests
```

```
In [3]: page = requests.get('http://membres-timc.imag.fr/Nicolas.Thierry-Mieg/jc_bcm.html')
        tree = html.fromstring(page.content)
```

tree now contains the whole *HTML* file in a nice tree structure which we can go over two different ways: *XPath*

1 Overview of the data

```
<tr>
<td>01/04/2016, <span style="color: rgb(255, 0, 0); font-weight: bold;">R32</span></td>
<td>Thomas Dias-Alves</td>
<td>Jupyter: an open source tool for data science, computing and pedagogy.</td>
</tr>
```

```
In [8]: from IPython.core.display import HTML
        HTML('''
        <tr>
        <td>01/04/2016, <span style="color: rgb(255, 0, 0); font-weight: bold;">R32</span></td>
        <td>Thomas Dias-Alves</td>
        <td>Jupyter: an open source tool for data science, computing and pedagogy.</td>
        </tr>
        ''')
```

```
Out[8]: <IPython.core.display.HTML object>
```

Now let's parse the data and get the titles !

```
In [19]: titles = tree.xpath('//table//td[3]/text()')

In [21]: titles[10:20]

Out[21]: ['Demographic inference under the coalescent in a spatial continuum.',
          'Reflections on stochastic spatial simulations.',
          'Team strategies and tools to improve hospital patient safety : contribution of the Experienc
          'Contribution of selected various methods of spatial analysis of incidence data : Application
          u"M\xe9thodes d'apprentissage statistique pour les tests d'association \xe9cologique\n",
          'MutaScript: a new tool for calculating a mutational score for each coding transcript.',
          'Functional genetic diversity of the yeast galactose network.',
          'Anomalous tracer diffusion in crowded medium.',
          '\n',
          'Study at the single molecule level of conformational changes of the DNA molecule:\n    impact

In [23]: print(titles[14])

Méthodes d'apprentissage statistique pour les tests d'association écologique

In [25]: dates = tree.xpath('//table//td[1]/text()')
         speaker = tree.xpath('//table//td[2]/text()')

In [31]: from sklearn.feature_extraction.text import CountVectorizer

         flatten_titles =  " ".join(titles)
         titles_words = CountVectorizer().build_tokenizer()(flatten_titles)

In [50]: from collections import Counter

         counts = Counter(titles_words)

In [51]: counts.most_common(20)

Out[51]: [(u'of', 67),
          (u'and', 34),
          (u'the', 32),
          (u'de', 32),
          (u'in', 29),
          (u'to', 20),
          (u'for', 19),
          (u'data', 13),
          (u'et', 13),
          (u'genetic', 12),
          (u'des', 11),
          (u'populations', 8),
          (u'pour', 8),
          (u'using', 8),
          (u'g\xe9n\xe9tique', 7),
          (u'la', 7),
          (u'population', 7),
          (u'cancer', 7),
          (u'analysis', 7),
          (u'on', 7)]
```

```
In [41]: import nltk
         nltk.download()
```

```
showing info http://www.nltk.org/nltk_data/
```

```
Out[41]: True
```

```
In [53]: from nltk.tokenize import word_tokenize

         words = word_tokenize(flatten_titles)

         from nltk.stem.snowball import EnglishStemmer

         stemmer = EnglishStemmer()
         words_stem = [stemmer.stem(w) for w in words if len(w) > 2]
```

```
In [54]: counts_stem = Counter(words_stem)
         counts_stem.most_common(20)
```

```
Out[54]: [(u'and', 35),
          (u'the', 34),
          (u'model', 21),
          (u'for', 19),
          (u'popul', 16),
          (u'genet', 14),
          (u'data', 13),
          (u'des', 11),
          (u'network', 11),
          (u'analysi', 10),
          (u'applic', 10),
          (u'use', 9),
          (u'g\xe9n\xe9tiqu', 9),
          (u'cancer', 9),
          (u'method', 8),
          (u'pour', 8),
          (u'genom', 8),
          (u'mod\xe8l', 8),
          (u'system', 7),
          (u'structur', 7)]
```

```
In [57]: %matplotlib inline
```

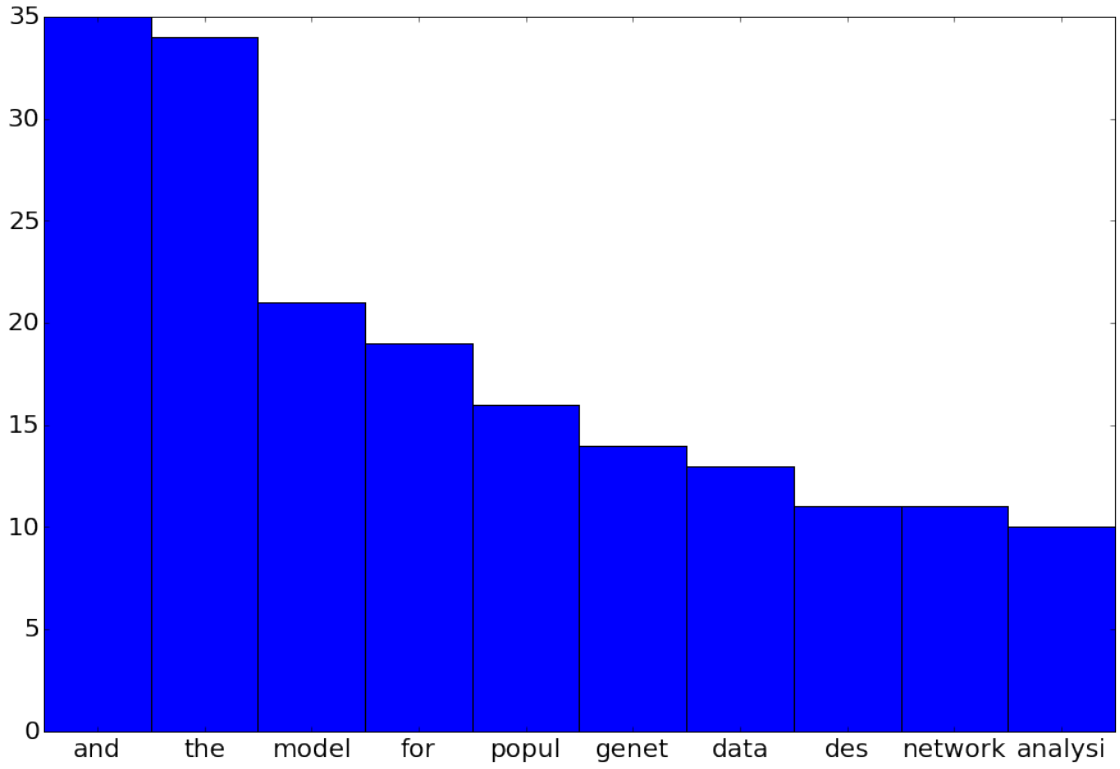
```
In [69]: import matplotlib
         import matplotlib.pyplot as plt
         import numpy as np

         width = 15
         height = 10
         plt.figure(figsize=(width, height))
         matplotlib.rcParams.update({'font.size': 20})

         labels, values = zip(*counts_stem.most_common(10))

         indexes = np.arange(len(labels))
         width = 1
```

```
plt.bar(indexes, values, width)
plt.xticks(indexes + width * 0.5, labels)
plt.show()
```



In [ ]:
```
from bokeh.io import output_notebook
output_notebook()
```

In [91]:
```
from bokeh.charts import Bar, output_file, show
from bokeh.sampledata.autompg import autompg as df

import pandas as pd

df = pd.DataFrame({
     'count' : values
      }, index=labels)
p = Bar(df)
output_notebook()
show(p)
```

In [ ]: