Data Mining The Computer Structures Class

Terry Yin

November 18, 2014

> What is essential is invisible to the eye. de Saint Exup6ry (1971)

1 What is Data Mining

Data mining is the process to discover knowledge from the database Wikipedia (2014*a*). It's a rapidly expanding subject across computer science, statistics, and many, many other areas (Brookshear 2011, p.414). I think data mining is the technology that helps people understand **what** has happened. Sometimes, understanding the **what** will help us discover the knowledge of **why**. We might not be able to understand **why** in some situations, but it can still help us to predict the future. The forms of data mining includes:

Class Description
: finds the common character among given data collection.

Cluster analysis
: divides the given data collection into group with unique characters.

Class discrimination
: studies the properties that distinguishing the data groups.

Association analysis
: studies the link between the data groups.

Outlier analysis
: find the information that is not related to the norm, or the noisy data.

sequential pattern analysis
: is a common scene of data mining, which is applied on sequential data over a timeline to indentify patterns.

2 An Example

Our Computer Structures course has been running for 11 weeks. There have been many posts by the colleagues in my class in the discussion. I will try to mine this database and see what I can find.

I will use IPython Notebook Pérez & Granger (2007) and Pandas McKinney (n.d.) as tools to show this example. Below is some preparing code that imports the Python libraries that will be used later.

```
%matplotlib inline
import pandas
import numpy
import re
from datetime import datetime
```

2.1 Data pre-processing

Data mining is not looking at the magical crystal ball to seek for a clue. It comes from the data we prepared. "Garbage in, garbage out" is particularly true to data mining Wikipedia (2014*b*). This is the most trivial and time-consuming step, but it's very important.

I've put all our posts from the previous 10 weeks in the a database now. The database is ... just a csv file that store all the information in plain text format. The data table include these columns: author, date, minutes_of_the_day, post, thread, weekday, content, and content length. And then I cannot wait to see what we have been talking about.

The above *tag cloud* Wikipedia (2014*d*) is from all our post content. I don't know why but it seems we like to use the word "use" a lot. Other than that, we use a lot of "compute", "system", "data", "algorithm". If you look harder, you can see "Craig" and "Terry".

2.2 Class Description

First, I load the data from the database (the csv file) into a Pandas DataFrame. Pandas DataFrame is a two dimensional data structure that can do many data mining and plotting things. After loading the data, I display the first 5 posts.

```
posts = pandas.read_csv("threads/allpost.csv").drop("Unnamed: 0", 1)
posts[:5][["author","date", "post", "minutes_of_the_day"]]
```

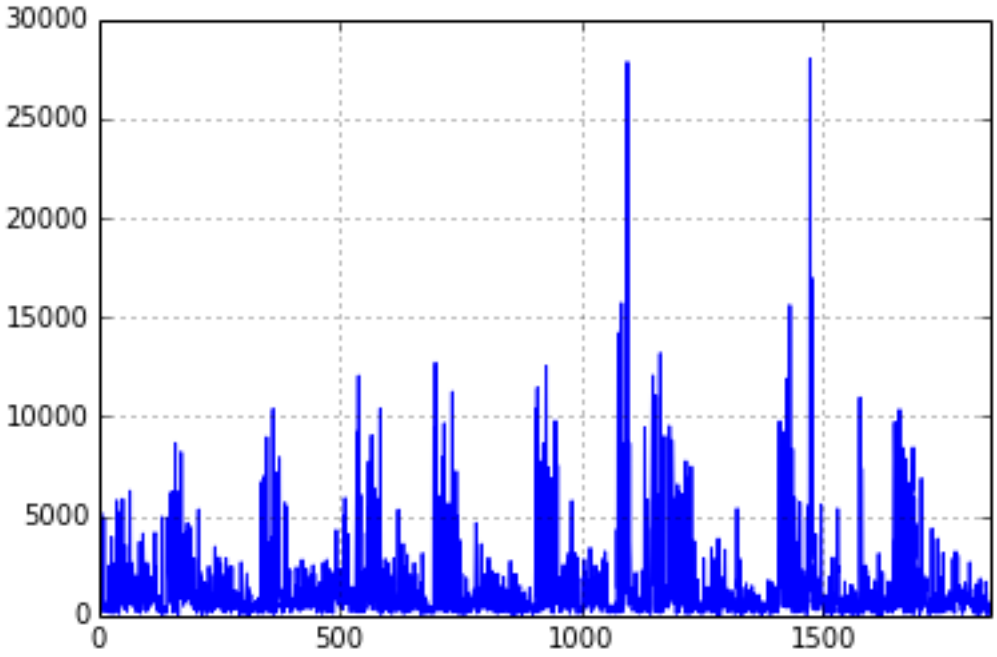
Out[111]:

	author	date	post	minutes_of_the_day
0	Anthony Ayoola	2014-08-29 17:55:00	Your history of computing	1075
1	Terry Yin	2014-09-04 04:34:00	RE: My first Computing Experience { A (long) s...	274
2	Christopher Burns	2014-09-04 06:53:00	RE: My first Computing Experience { A (long) s...	413
3	BABATUNDE KOLAWOLE	2014-09-04 08:17:00	RE: My first Computing Experience { A (long) s...	497
4	Terry Yin	2014-09-04 06:41:00	RE: Your history of computing	401

Then, let's see if we can find anything by plotting the post length on a timeline (the record is already sorted by date of post).

```
posts.length.plot()
```

Out[200]: <matplotlib.axes.AxesSubplot at 0x10f9392d0>



It seems that our posts are getting longer and longer since the first week. And there seems to be a pattern. So what if we group our post by weekdays?

```
d = posts.groupby("weekday").aggregate({"length":[numpy.sum, numpy.mean, numpy.count_nonzero]})
d
```

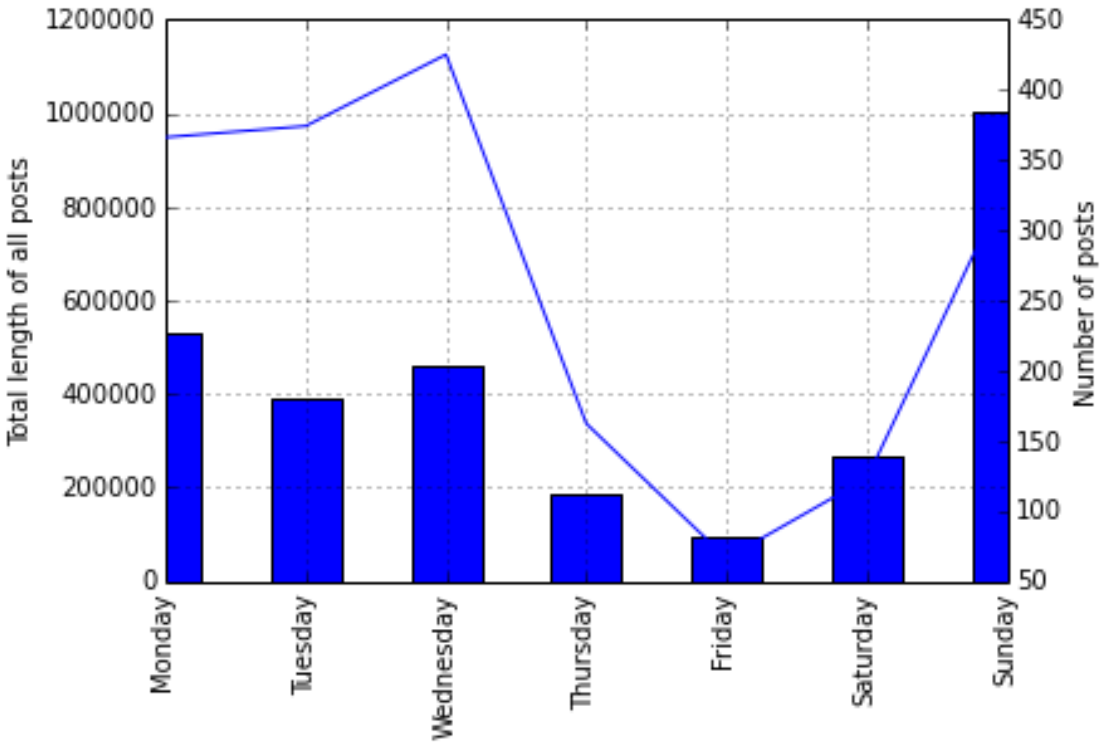
Out[198]:

	length		
	sum	mean	count_nonzero
weekday			
0	528551	1444.128415	366
1	390041	1042.890374	374
2	458212	1078.145882	425
3	183239	1131.104938	162
4	93266	1413.121212	66
5	268022	2179.040650	123
6	1000720	3005.165165	333

Let's plot the data and see what we can see.

```
d.index = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
d.length["sum"].plot(kind='bar').set_ylabel('Total length of all posts')
d.length["count_nonzero"].plot(secondary_y=True).set_ylabel('Number of posts')
```

```
Out[188]: <matplotlib.text.Text at 0x10f399850>
```



This is probably obvious even without the data. But now we can clearly see that we write long posts on Sunday, but we write more posts on Wednesday.

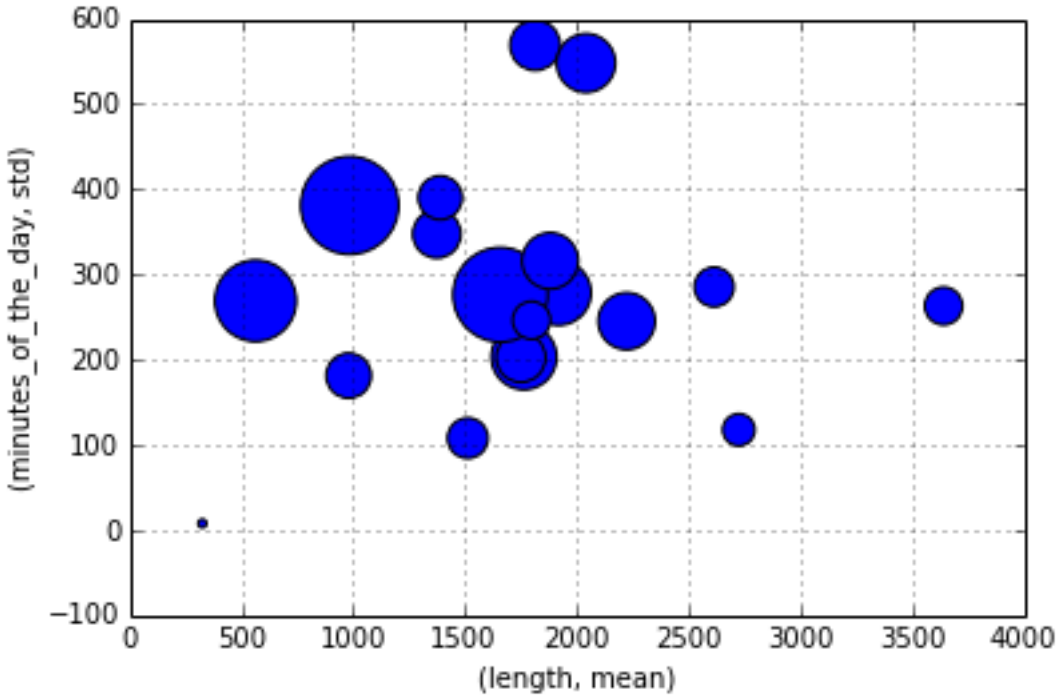
2.3 Cluster analysis

There must be some different behavioural types in all the colleague (and Dr. Ayoola). Let's what we can get by grouping by the author of the posts.

In the data I prepared, there is one column named `minutes_of_the_day`. It is the minutes passed since the beginning of that day. We are going to get the **standard deviation** (Wikipedia 2014*c*, std) of the `minutes_of_the_day` of each author. Let's assume this standard deviation will show whether a colleague follows a fixed time schedule for study or follows a more flexible style.

```
d=posts.groupby("author").aggregate(
        {"length":[numpy.mean, numpy.count_nonzero],
         "minutes_of_the_day":numpy.std})
d.plot(kind='scatter', x=('length', 'mean'), y=('minutes_of_the_day', 'std'), s=d.length['count_nonzero']*5)
```

Out[199]: <matplotlib.axes.AxesSubplot at 0x10f6d2910>



In the above picture, each bubble represents one colleague. The size of the bubble represents the number of posts he every posted. X-axis is the average post length. Y-axis is the standard deviation of time. From the picture we can see, most colleagues study in around 5 hours (300minutes) time frame. There are a couple of colleagues who are very disciplined. They study in a 2 hours frame (the two near the bottom, excluding the one on the bottom left corner).

3 Conclusion

The quality of the input is crucial for data mining. Many people do data mining spend most of their time preparing their data. It's the same for the easy example I just showed above. I spend most of my time collecting all the posts, parsing the raw data (HTML), improving data quality and putting the data into database. Once the data is ready, tools like Pandas are pretty handy to further explore the data and visualize them.

```
d=posts.groupby("author").aggregate(
        {"length":[numpy.sum, numpy.count_nonzero],
         "minutes_of_the_day":numpy.std,
         "content":lambda x:''.join(x)})
d.columns = ["a",'b','c','t']
d.sort('c')
```

Out[82]:

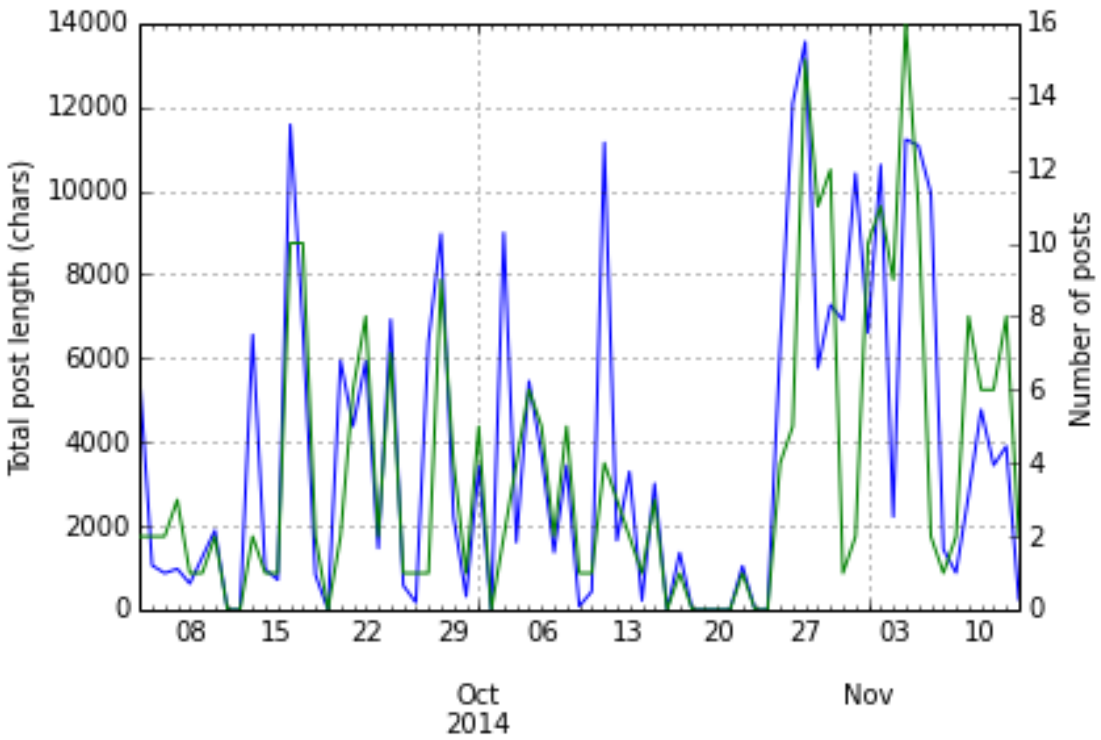
author	a	\
Sohail Iqbal	\n\t	\nMy parent planned to take ...
Mohamed Ali	\n\t	The answers:\n1. Based on th...
CHIKA ACHA	\n\t	I would beginning by asking ...
Kharavela Jain	\n\t	Back in 1997, when I was in ...
King Tan Yu	\n\t	IntroductionToday, the compu...
Tresor Lungu	\n\t	Computer, this term is reaso...
Numan Arshad	\n\t	Let me try to answer the que...
Ramin Cyrus	\n\t	Hi Craig Wasn't expect...
martins Ebam	\n\t	MY HISTORY OF COMPUTER \nInt...
Belinda Ndlovu	\n\t	Hi Anthony,\n\nWhat a relief...
Bram Tullemans	\n\t	From single use to multi-use...
Ricardo Biggs	\n\t	Dr. Anthony,\nThank you for ...
Joseph Warero	\n\t	Hi.\nInteresting account and...
Augusto Schoonewolff	\n\t	My first computer experience...
tanisha gittens	\n\t	\nYour first experiences usi...
BABATUNDE KOLAWOLE	\n\t	\nInteresting story....let me ...
Christopher Burns	\n\t	Hi ,\n\nLoved the story espe...
Bo Mogensen	\n\t	Hi Anthony\nGreat story. Wel...
Anthony Ayoola	\n\t	Do you remember the first ti...
Craig Thomas	\n\t	I'm going to try the quiz fi...
Terry Yin	\n\t	Interesting geeky story. I d...

author	b	c	t
Sohail Iqbal	3887	1	NaN
Mohamed Ali	649	2	7.778175
CHIKA ACHA	79025	29	117.738397
Kharavela Jain	71903	40	246.298286
King Tan Yu	145661	40	263.080957
Tresor Lungu	115043	44	285.367976
Numan Arshad	69560	46	107.942096
Ramin Cyrus	75000	54	390.164939
martins Ebam	55817	57	181.309636
Belinda Ndlovu	89297	65	347.548392
Bram Tullemans	119002	68	202.479490
Ricardo Biggs	128780	71	569.139901
Joseph Warero	167415	89	316.058351
Augusto Schoonewolff	202311	91	245.262561
tanisha gittens	198015	97	548.258880
BABATUNDE KOLAWOLE	177172	111	272.130333
Christopher Burns	226431	118	278.572342
Bo Mogensen	211634	120	203.252682
Anthony Ayoola	104657	186	269.159951

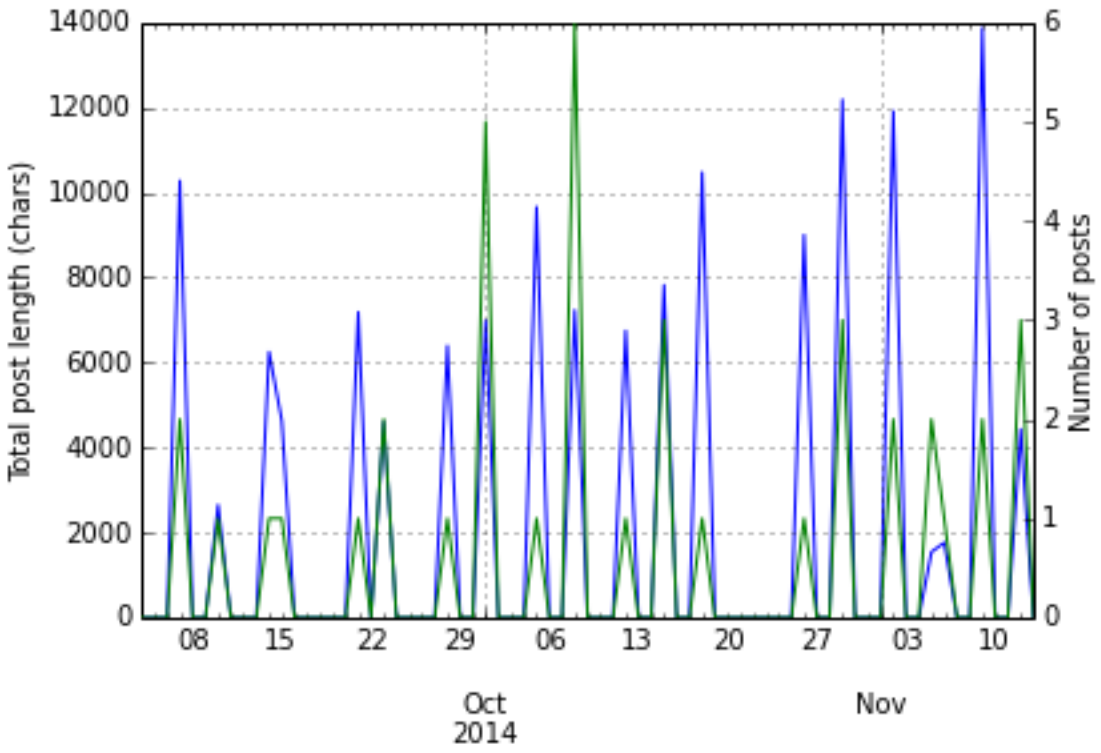
```
Craig Thomas          416118  251  276.246933
Terry Yin             264674  269  381.113642
```

```
def plot_user(name):
    au = posts[(posts.author.apply(lambda x: name in x))]
    g = au.groupby(au.date.apply(lambda x: datetime.strptime(x, '%Y-%m-%d %H:%M:%S').date())).aggregate(
            {"length":[numpy.sum, numpy.count_nonzero]})
    idx = pandas.date_range('09-04-2014', '11-13-2014')
    g.index = pandas.DatetimeIndex(g.index)
    g = g.reindex(idx, fill_value=0)
    g.length["sum"].plot().set_ylabel('Total post length (chars)')
    g.length["count_nonzero"].plot(secondary_y=True).set_ylabel('Number of posts')
```

```
plot_user("Terry")
```




```
plot_user("King Tan Yu")
```




```
from collections import Counter
words = d['a']['BABATUNDE KOLAWOLE'].lower().split()
phrases = [' '.join(x) for x in zip(words, words[1:], words[2:])]
cnt = Counter(phrases)
import operator
sorted_x = sorted(cnt.items(), key=operator.itemgetter(1), reverse=True)
```

```
a="abcdefg"
zip(a, a[1:], a[2:])
```

```
Out[34]: [('a', 'b', 'c'),
          ('b', 'c', 'd'),
          ('c', 'd', 'e'),
          ('d', 'e', 'f'),
          ('e', 'f', 'g')]
```

```
? d['a']['BABATUNDE KOLAWOLE'].count('Babatunde')
```

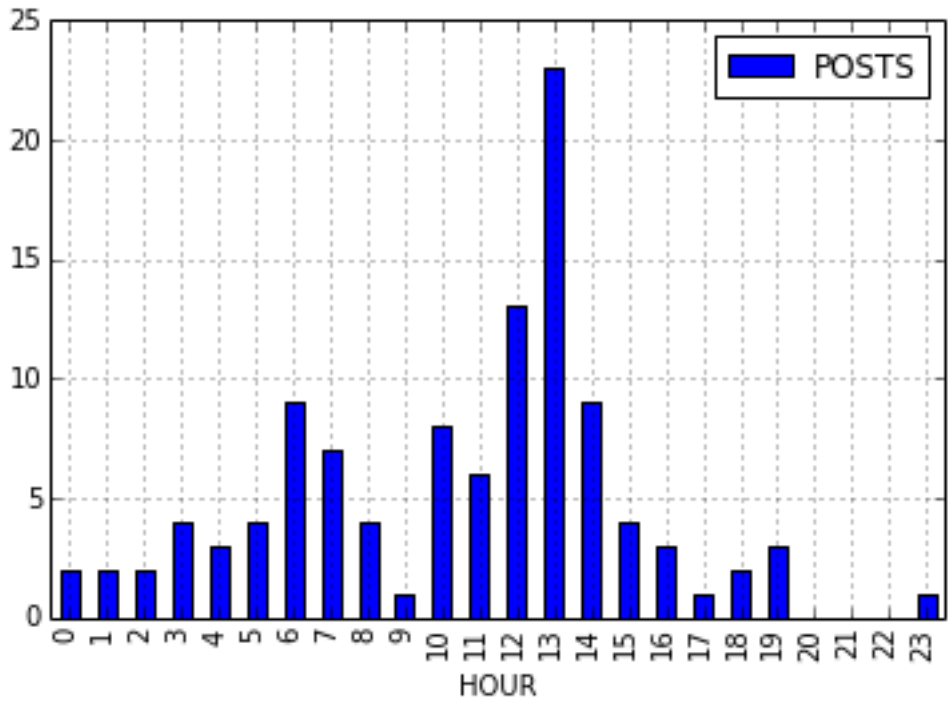
Out[75]: 90

```
" ".join(d['a']).count('Babatunde')
```

Out[70]: 151

```
hours = posts[posts.author == 'BABATUNDE KOLAWOLE'][["minutes_of_the_day"]]//60
hours.columns = ["HOUR"]
hours = hours.groupby("HOUR").aggregate(
            {"HOUR":len})
hours.columns = ["POSTS"]
idx = range(24)
#g.index = pandas.DatetimeIndex(g.index)
hours = hours.reindex(idx, fill_value=0)
hours.plot(kind="bar")
```

Out[146]: <matplotlib.axes.AxesSubplot at 0x11680a890>




```
2//3
```

Out[118]: 0

'

References

Brookshear, J. G. (2011), *Computer science: an overview*, Paul Muljadi.

de Saint Exup6ry, A. (1971), 'The little prince (1943)'.

McKinney, W. (n.d.), 'pandas: a python data analysis library', *see http://pandas. pydata. org* .

Pérez, F. & Granger, B. E. (2007), 'IPython: a System for Interactive Scientific Computing', *Computing in Science & Engineering* **9**(3), 21–29. URL: http://ipython.org.

Wikipedia (2014*a*), 'Data mining — wikipedia, the free encyclopedia'. [Online; accessed 15-November-2014].
URL: *http://en.wikipedia.org/w/index.php?title=Data_mining&oldid=633853279*

Wikipedia (2014*b*), 'Data pre-processing — wikipedia, the free encyclopedia'. [Online; accessed 15-November-2014].
URL: *http://en.wikipedia.org/w/index.php?title=Data_pre-processing&oldid=630904340*

Wikipedia (2014*c*), 'Standard deviation — wikipedia, the free encyclopedia'. [Online; accessed 15-November-2014].
URL: *http://en.wikipedia.org/w/index.php?title=Standard_deviation&oldid=632331612*

Wikipedia (2014*d*), 'Tag cloud — wikipedia, the free encyclopedia'. [Online; accessed 15-November-2014].
URL: *http://en.wikipedia.org/w/index.php?title=Tag_cloud&oldid=611212945*