Introduction

This project is about checking users post in **Hacker News**

In [2]:
```
from csv import reader

opened_file = open('hacker_news.csv')
read_file = reader(opened_file)
hn = list(read_file)
headers = hn[0]
hn = hn[1:]
print(headers)
print(hn[:5])
```

```
['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at']
[['12224879', 'Interactive Dynamic Video', 'http://www.interactivedynamicvideo.com/', '386', '52', 'ne0phyte', '8/4/2016 11:52'], ['10975351', 'How to Use Open Source and Shut the
Fuck Up at the Same Time', 'http://hueniverse.com/2016/01/26/how-to-use-open-source-and-shut-the-fuck-up-at-the-same-time/', '39', '10', 'josep2', '1/26/2016 19:30'], ['11964716',
"Florida DJs May Face Felony for April Fools' Water Joke", 'http://www.thewire.com/entertainment/2013/04/florida-djs-april-fools-water-joke/63708/', '2', '1', 'vezycash', '6/23/201
6 22:20'], ['11919867', 'Technology ventures: From Idea to Enterprise', 'https://www.amazon.com/Technology-Ventures-Enterprise-Thomas-Byers/dp/0073523429', '3', '1', 'hswarna', '6/
17/2016 0:01'], ['10301696', 'Note by Note: The Making of Steinway L1037 (2007)', 'http://www.nytimes.com/2007/11/07/movies/07stein.html?_r=0', '8', '2', 'walterbell', '9/30/2015
4:12']]
```

In [3]:
```
ask_posts = []
show_posts = []
other_posts = []
for row in hn:
    title = row[1]
    if title.lower().startswith('ask hn'):
        ask_posts.append(row)
    elif title.lower().startswith('show hn'):
        show_posts.append(row)
    else:
        other_posts.append(row)

print(len(ask_posts))
print(len(show_posts))
print(len(other_posts))
```

```
1744
1162
17194
```

In [4]:
```
total_ask_comments = 0
for row in ask_posts:
    num_comments = row[4]
    num_comments = int(num_comments)
    total_ask_comments += num_comments
    avg_ask_comments = total_ask_comments/ len(ask_posts[4])

print(avg_ask_comments)

total_show_comments = 0
for row in show_posts:
    num_comments =  row[4]
    num_comments = int(num_comments)
    total_show_comments += num_comments
    avg_show_comments = total_show_comments/ len(show_posts[4])

print(avg_show_comments)
```

```
3497.5714285714284
1712.5714285714287
```

Do show posts or ask post receive more comments?

According to the result above, ask post get more comments.

In [14]:
```
import datetime as dt

# Create a list that contains the creation times and number of comments (ask-posts only)

result_list = []
for post in ask_posts:
    created_at = post[6]
    num_comments = int(post[4])
    result_list.append([created_at, num_comments])
#print (result_list[:3])

# Build frequency tables for the number of posts and for the number of comments, per hour of the day
counts_by_hour = {}
comments_by_hour = {}

for row in result_list:
    created_at = dt.datetime.strptime(row[0], '%m/%d/%Y %H:%M')
    hour = created_at.hour
    if hour not in counts_by_hour:
        counts_by_hour[hour] = 1
        comments_by_hour[hour] = row[1]
    else:
        counts_by_hour[hour] += 1
        comments_by_hour[hour] += row[1]

# Create a table that contains the hours of day and the average number of comments per posts
avg_by_hour = []
for hour in counts_by_hour:
    num_posts = counts_by_hour[hour]
    num_comments = comments_by_hour[hour]
    average = num_comments / num_posts
    avg_by_hour.append([hour, average])

# Sort the list (on its first element, being the hour of day)
avg_by_hour.sort()

# Print the result
output = "For hour {:02d} the average number of comments per post is {:.2f}"
for row in avg_by_hour:
    print (output.format(row[0], row[1]))
```

```
For hour 00 the average number of comments per post is 8.13
For hour 01 the average number of comments per post is 11.38
For hour 02 the average number of comments per post is 23.81
For hour 03 the average number of comments per post is 7.80
For hour 04 the average number of comments per post is 7.17
For hour 05 the average number of comments per post is 10.09
For hour 06 the average number of comments per post is 9.02
For hour 07 the average number of comments per post is 7.85
For hour 08 the average number of comments per post is 10.25
For hour 09 the average number of comments per post is 5.58
For hour 10 the average number of comments per post is 13.44
For hour 11 the average number of comments per post is 11.05
For hour 12 the average number of comments per post is 9.41
For hour 13 the average number of comments per post is 14.74
For hour 14 the average number of comments per post is 13.23
For hour 15 the average number of comments per post is 38.59
For hour 16 the average number of comments per post is 16.80
For hour 17 the average number of comments per post is 11.46
For hour 18 the average number of comments per post is 13.20
For hour 19 the average number of comments per post is 10.80
For hour 20 the average number of comments per post is 21.52
For hour 21 the average number of comments per post is 16.01
For hour 22 the average number of comments per post is 6.75
For hour 23 the average number of comments per post is 7.99
```

In [6]:
```
avg_by_hour = []

for hour in comments_by_hour:
    avg_by_hour.append([hour,comments_by_hour[hour]/ counts_by_hour[hour]])

print(avg_by_hour)
```

```
[['07', 7.852941176470588], ['10', 13.440677966101696], ['19', 10.8], ['11', 11.051724137931034], ['05', 10.08695652173913], ['06', 9.022727272727273], ['13', 14.741176470588234],
['14', 13.233644859813085], ['20', 21.525], ['12', 9.41095890410959], ['16', 16.796296296296298], ['17', 11.46], ['21', 16.009174311926607], ['09', 5.577777777777775], ['03', 7.79
6296296296297], ['15', 38.5948275862069], ['00', 8.127272727272727], ['08', 10.25], ['18', 13.20183486238532], ['02', 23.810344827586206], ['01', 11.383333333333333], ['22', 6.7464
78873239437], ['23', 7.985294117647059], ['04', 7.170212765957447]]
```

In [7]:
```
swap_avg_by_hour = []
for hour in avg_by_hour:
    first_e = hour[0]
    second_e = hour[1]
    swap = [first_e,second_e]
    swap_avg_by_hour.append(swap)
print(swap_avg_by_hour)
```

```
[['07', 7.852941176470588], ['10', 13.440677966101696], ['19', 10.8], ['11', 11.051724137931034], ['05', 10.08695652173913], ['06', 9.022727272727273], ['13', 14.741176470588234],
['14', 13.233644859813085], ['20', 21.525], ['12', 9.41095890410959], ['16', 16.796296296296298], ['17', 11.46], ['21', 16.009174311926607], ['09', 5.577777777777775], ['03', 7.79
6296296296297], ['15', 38.5948275862069], ['00', 8.127272727272727], ['08', 10.25], ['18', 13.20183486238532], ['02', 23.810344827586206], ['01', 11.383333333333333], ['22', 6.7464
78873239437], ['23', 7.985294117647059], ['04', 7.170212765957447]]
```

In [13]:
```
swap_avg_by_hour = []
for row in avg_by_hour:
    swap_avg_by_hour.append([row[1], row[0]])

# Created a sorted version of this list
sorted_swap = sorted (swap_avg_by_hour, reverse = True)

# Display the results
print ('Top 5 Hours for Ask Posts Comments', '\n')
output = "{}: {:.2f} average comments per post"
for row in sorted_swap[:5]:
    thetime = dt.datetime.strptime(str(row[1]), '%H')
    thetime = thetime.strftime('%H:%M')
    print ( output.format(thetime,row[0] ))
```

```
Top 5 Hours for Ask Posts Comments

15:00: 38.59 average comments per post
02:00: 23.81 average comments per post
20:00: 21.52 average comments per post
16:00: 16.80 average comments per post
21:00: 16.01 average comments per post
```

conclution

Refering back to the goal of this study, let's summarize the conclusions.

Post title: when creating posts, adding Ask HN to your post title will do better for attracting comments than adding Show HN:

Ask HN: 14.04 average comments per post Show HN: 10.32 average comments per post (It has not been compared with posts for not adding a tag at all.)

Post timing: the time of day of posting appears to have significant impact on the number of comments that you will attract. Based on an analysis of the Ask HN posts, the top hours (in Central European Time) are:

21:00 - 22:00: 38.59 average comments per post 08:00 - 09:00: 23.81 average comments per post 02:00 - 03:00: 21.52 average comments per post

In [ ]:
```
```