

# Exploring the Post Engagement per Hour at HackerNews

**Author : Kenneth Lucas Kusima**  
**Contributor: DataQuest**

## Project Origin:

This project was made per instructions from:

**The DataQuest Course: Python for Data Science Intermediate**

## Objectives:

We're specifically interested in posts whose titles begin with either Ask HN or Show HN. Users submit Ask HN posts to ask the Hacker News community a specific question. Likewise, users submit Show HN posts to show the Hacker News community a project, product, or just generally something interesting

We'll compare these two types of posts to determine the following:

- Do Ask HN or Show HN receive more comments on average?
- Do posts created at a certain time receive more comments on average?

In [78]:

```
from csv import reader
opened_file = open('hacker_news.csv')
read_file = reader(opened_file)
hn = list(read_file)
print('The first 5 rows of the dataset being used: \n')
print(hn[0:5])
```

The first 5 rows of the dataset being used:

```
[['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/63798/', '2', '1', 'vezycash', '6/23/2016 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']]
```

In [79]:

```
#Extracting the headers of the dataset
headers = hn[0]
hn = hn[1:]
print('The header : \n')
print(headers)
print('\n The first 5 rows of the dataset being used - excluding the header: \n')
print(hn[0:5])
```

The header :

```
['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at']
```

The first 5 rows of the dataset being used - excluding the header:

```
[['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/63798/', '2', '1', 'vezycash', '6/23/2016 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']]
```

## Filtering the data

**Filtering posts according to Ask HN, Show HN and other posts:**

In [80]:

```

#initializing the separate lists corresponding to the different types of posts
ask_posts = []
show_posts = []
other_posts = []

for row in hn:
    title = row[1]
    if title.lower().startswith('ask hn'): #filtering out only the lower versions of
        ask_posts.append(row)
    elif title.lower().startswith('show hn'):
        show_posts.append(row)
    else:
        other_posts.append(row)

print('Numnber of "Ask HN" posts:', len(ask_posts))
print('Numnber of "Show HN" posts:', len(show_posts))
print('Numnber of other types of posts:', len(other_posts))

print('\n The first five rows of the ask_posts list: \n')
print(ask_posts[0:5])

print('\n The first five rows of the show_posts list: \n')
print(show_posts[0:5])

```

```

Numnber of "Ask HN" posts: 1744
Numnber of "Show HN" posts: 1162
Numnber of other types of posts: 17194

```

The first five rows of the ask\_posts list:

```

[['12296411', 'Ask HN: How to improve my personal website?', '', '2',
'6', 'ahmedbaracat', '8/16/2016 9:55'], ['10610020', 'Ask HN: Am I the
only one outraged by Twitter shutting down share counts?', '', '28',
'29', 'tkfx', '11/22/2015 13:43'], ['11610310', 'Ask HN: Aby recent ch
anges to CSS that broke mobile?', '', '1', '1', 'polskibus', '5/2/2016
10:14'], ['12210105', 'Ask HN: Looking for Employee #3 How do I do i
t?', '', '1', '3', 'sph130', '8/2/2016 14:20'], ['10394168', 'Ask HN:
Someone offered to buy my browser extension from me. What now?', '',
'28', '17', 'roykolak', '10/15/2015 16:38']]

```

The first five rows of the show\_posts list:

```

[['10627194', 'Show HN: Wio Link ESP8266 Based Web of Things Hardware
Development Platform', 'https://iot.seeed.cc', '26', '22', 'kfihihc',
'11/25/2015 14:03'], ['10646440', 'Show HN: Something pointless I mad
e', 'http://dn.ht/picklecat/', '747', '102', 'dhotson', '11/29/2015 2
2:46'], ['11590768', 'Show HN: Shanhu.io, a programming playground pow
ered by e8vm', 'https://shanhu.io', '1', '1', 'h8liu', '4/28/2016 18:0
5'], ['12178806', 'Show HN: Webscope Easy way for web developers to c
ommunicate with Clients', 'http://webscopeapp.com', '3', '3', 'fastbri
ck', '7/28/2016 7:11'], ['10872799', 'Show HN: GeoScreenshot Easily t
est Geo-IP based web pages', 'https://www.geoscreenshot.com/', '1',
'9', 'kpsychwave', '1/9/2016 20:45']]

```

## Analysing the number of comments

In [81]:

```
total_ask_comments = 0
for row in ask_posts:
    num_comments = int(row[4])
    total_ask_comments = total_ask_comments + num_comments
avg_ask_comments = total_ask_comments / len(ask_posts)
print('Average number of comments for the "Ask HN" posts:', avg_ask_comments)

total_show_comments = 0
for rows in show_posts:
    num_comments_ = int(rows[4])
    total_show_comments = total_show_comments + num_comments_
avg_show_comments = total_show_comments / len(show_posts)
print('Average number of comments for the "Show HN" posts:', avg_show_comments)
```

Average number of comments for the "Ask HN" posts: 14.038417431192661

Average number of comments for the "Show HN" posts: 10.31669535283993

From the results above we can see that the "Ask HN" posts have on average, more comments per post compared to the "Show HN" posts

## Analysing which posting times attract the most comments

Since asks posts produce the most comments on average, analysis will be focused for these kinds of posts

### Isolating the number of posts created in each hour and the corresponding comments generated

In [86]:

```
#Calculating the amount of ask posts created per hour, along with the total amount of

import datetime as dt
result_list = []

for row in ask_posts:
    created_at = row[6]
    num_coments = int(row[4])
    result_list.append([created_at , num_coments])

counts_by_hour = {}
comments_by_hour = {}

for row in result_list:
    ob = row[0]
    comm = row[1]
    date_time = dt.datetime.strptime(ob, "%m/%d/%Y %H:%M")
    hour_ = date_time.strftime('%H')
    if hour_ not in counts_by_hour:
        counts_by_hour[hour_] = 1
        comments_by_hour[hour_] = comm
    else:
        counts_by_hour[hour_] += 1
        comments_by_hour[hour_] = comments_by_hour[hour_] + comm

print('Number of "Ask HN" posts created for each hour: \n', counts_by_hour)
print('\n Corresponding total number of comments "Ask HN" posts created at each hour: \n', comments_by_hour)
```

Number of "Ask HN" posts created for each hour:

```
{'12': 73, '16': 108, '20': 80, '07': 34, '00': 55, '19': 110, '15': 116, '03': 54, '17': 100, '01': 60, '22': 71, '18': 109, '21': 109, '13': 85, '06': 44, '05': 46, '04': 47, '08': 48, '02': 58, '11': 58, '09': 45, '14': 107, '23': 68, '10': 59}
```

Corresponding total number of comments "Ask HN" posts created at each hour received:

```
{'12': 687, '16': 1814, '20': 1722, '07': 267, '00': 447, '19': 1188, '15': 4477, '03': 421, '17': 1146, '01': 683, '22': 479, '18': 1439, '21': 1745, '13': 1253, '06': 397, '05': 464, '04': 337, '08': 492, '02': 1381, '11': 641, '09': 251, '14': 1416, '23': 543, '10': 793}
```

## Calculating the average number comments per posts for each hour

In [87]:

```

avg_by_hour = []

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

print(avg_by_hour)

[['12', 9.41095890410959], ['16', 16.796296296296298], ['20', 21.525],
 ['07', 7.852941176470588], ['00', 8.127272727272727], ['19', 10.8],
 ['15', 38.5948275862069], ['03', 7.796296296296297], ['17', 11.46],
 ['01', 11.383333333333333], ['22', 6.746478873239437], ['18', 13.20183
 486238532], ['21', 16.009174311926607], ['13', 14.741176470588234],
 ['06', 9.022727272727273], ['05', 10.08695652173913], ['04', 7.1702127
 65957447], ['08', 10.25], ['02', 23.810344827586206], ['11', 11.051724
 137931034], ['09', 5.5777777777777775], ['14', 13.233644859813085],
 ['23', 7.985294117647059], ['10', 13.440677966101696]]

```

Ordering the above list based on highest number of comments per post per hour:

In [100]:

```

swap_avg_by_hour = []

for row in avg_by_hour:
    element_1 = row[1]
    element_2 = row[0]
    swap_avg_by_hour.append([element_1, element_2])

#Sorting by the average number of comments
sorted_swap = sorted(swap_avg_by_hour, reverse = True)
#print(sorted_swap)

print(' \n Top 5 Hours for "Ask HN" Posts Comments: \n')
for val in sorted_swap[0:5]:
    template = "{hour}:00: {num:.2f} average comments per post"
    result = template.format(hour = val[1], num = val[0])
    print(result)

```

Top 5 Hours for "Ask HN" 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

```