Project : **Exploring Hacker News Posts**

Aim : To analyse dataset of Hacker News posts which is a site started by the startup incubator Y Combinator.

Link to Dataset : Link

We'll compare two types of posts on the site, 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?

Let's start by importing the dataset-

In [34]:
```
from csv import reader
opened_file = open("hacker_news.csv")
read_file = reader(opened_file)
hn = list(read_file)
```

print(hn[:5])

Separating header column and data rows-

In [35]:
```
headers = hn[0]
hn = hn[1:]
print(headers)
```

```
['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created
_at']
```

In [36]:
```
print(hn[:5])
```

```
[['12224879', 'Interactive Dynamic Video', 'http://www.interactivedynam
```

```
icvideo.com/', '386', '52', 'ne0phyte', '8/4/2016 11:52'], ['10975351',
'How to Use Open Source and Shut the Fuck Up at the Same Time', 'htt
p://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'], ['1196
4716', "Florida DJs May Face Felony for April Fools' Water Joke", 'htt
p://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.co
m/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/07ste
in.html?_r=0', '8', '2', 'walterbell', '9/30/2015 4:12']]
```

Creating new lists of lists containing just the data for titles- Ask HN or Show HN

In [37]:
```
ask_posts = []
show_posts = []
other_posts = []

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

print("Number of ask posts:", len(ask_posts))
print("Number of show posts:", len(show_posts))
print("Number of other posts:", len(other_posts))
```

```
Number of ask posts: 1744
Number of show posts: 1162
Number of other posts: 17193
```

In [38]:
```
print(ask_posts[:5])
```

```
[['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', '2
9', 'tkfx', '11/22/2015 13:43'], ['11610310', 'Ask HN: Aby recent chang
es to CSS that broke mobile?', '', '1', '1', 'polskibus', '5/2/2016 10:
14'], ['12210105', 'Ask HN: Looking for Employee #3 How do I do it?',
'', '1', '3', 'sph130', '8/2/2016 14:20'], ['10394168', 'Ask HN: Someon
e offered to buy my browser extension from me. What now?', '', '28', '1
7', 'roykolak', '10/15/2015 16:38']]
```

In [39]:
```
print(show_posts[:5])
```

```
[['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 22:
46'], ['11590768', 'Show HN: Shanhu.io, a programming playground powere
d by e8vm', 'https://shanhu.io', '1', '1', 'h8liu', '4/28/2016 18:05'],
['12178806', 'Show HN: Webscope  Easy way for web developers to communi
cate with Clients', 'http://webscopeapp.com', '3', '3', 'fastbrick',
'7/28/2016 7:11'], ['10872799', 'Show HN: GeoScreenshot  Easily test Ge
o-IP based web pages', 'https://www.geoscreenshot.com/', '1', '9', 'kps
ychwave', '1/9/2016 20:45']]
```

To Find - Let's determine if ask posts or show posts receive more comments on average.

In [40]:
```
## Finding total number of comments in ask posts

total_ask_comments = 0
for row in ask_posts:
    num_comments = int(row[4])
    total_ask_comments += num_comments

print("No of ask_comments: ", total_ask_comments)
print("avg_ask_comments", total_ask_comments / len(ask_posts))
```

```
No of ask_comments:  24483
avg_ask_comments 14.038417431192661
```

In [41]:
```
## Finding total number of comments in ask posts

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

print("No of show_comments: ", total_show_comments)
print("avg_show_comments", total_show_comments / len(show_posts))
```

```
No of show_comments:  11988
avg_show_comments 10.31669535283993
```

Finding - ask posts received more comments on average than show posts The average comments for the title Ask HN is around 14 and the average comments for the title Show HN is around 10.

To Find - if ask posts created at a certain time are more likely to attract comments We'll use the following steps to perform this analysis:

- Calculate the amount of ask posts created in each hour of the day, along with the number of comments received.
- Calculate the average number of comments ask posts receive by hour created.

In [42]:
```
import datetime as dt
result_list = []
for row in ask_posts:
    created_at = row[6]
    num_comments = int(row[4])
    result_list.append([created_at, num_comments])
print(result_list[2])
```

```
['5/2/2016 10:14', 1]
```

In [43]:
```
counts_by_hour = {}
comments_by_hour = {}
for row in result_list:
    date = row[0]
    comments = row[1]
    time = dt.datetime.strptime(date, "%m/%d/%Y %H:%M").strftime("%H")
    if time in counts_by_hour:
        counts_by_hour[time] += 1
        comments_by_hour[time] += comments
    else:
        counts_by_hour[time] = 1
        comments_by_hour[time] = comments
```

In [44]:
```
## No of ask posts created in each hour of the day

print(counts_by_hour)
```

```
{'19': 110, '06': 44, '22': 71, '21': 109, '17': 100, '05': 46, '13': 8
5, '03': 54, '18': 109, '09': 45, '23': 68, '07': 34, '12': 73, '02': 5
8, '20': 80, '08': 48, '04': 47, '10': 59, '15': 116, '00': 55, '16': 1
08, '11': 58, '01': 60, '14': 107}
```

In [45]:
```
## No of comments received on ask posts created in each hour of the day

print(comments_by_hour)
```

```
{'19': 1188, '06': 397, '22': 479, '21': 1745, '17': 1146, '05': 464,
'13': 1253, '03': 421, '18': 1439, '09': 251, '23': 543, '07': 267, '1
2': 687, '02': 1381, '20': 1722, '08': 492, '04': 337, '10': 793, '15':
4477, '00': 447, '16': 1814, '11': 641, '01': 683, '14': 1416}
```

Calculating the average number of comments

for posts created during each hour of the day.

In [46]:
```
avg_by_hour = []
for hour in comments_by_hour:
    avg_by_hour.append([hour,comments_by_hour[hour] / counts_by_hour[ho
ur]])

print(avg_by_hour)
```

```
[['19', 10.8], ['06', 9.022727272727273], ['22', 6.746478873239437],
['21', 16.009174311926607], ['17', 11.46], ['05', 10.08695652173913],
['13', 14.741176470588234], ['03', 7.796296296296297], ['18', 13.201834
86238532], ['09', 5.5777777777777775], ['23', 7.985294117647059], ['0
7', 7.852941176470588], ['12', 9.41095890410959], ['02', 23.81034482758
6206], ['20', 21.525], ['08', 10.25], ['04', 7.170212765957447], ['10',
13.440677966101696], ['15', 38.5948275862069], ['00', 8.12727272727272
7], ['16', 16.796296296296298], ['11', 11.051724137931034], ['01', 11.3
83333333333333], ['14', 13.233644859813085]]
```

Let's finish by sorting the list of lists and printing the five highest values in a format that's easier to read.

In [53]:
```
##Sorting and Printing Values from a List of Lists


swap_average_by_hour = [[h[1],h[0]] for h in avg_by_hour]
print(swap_average_by_hour)
```

```
[[10.8, '19'], [9.022727272727273, '06'], [6.746478873239437, '22'], [1
6.009174311926607, '21'], [11.46, '17'], [10.08695652173913, '05'], [1
4.741176470588234, '13'], [7.796296296296297, '03'], [13.2018348623853
2, '18'], [5.5777777777777775, '09'], [7.985294117647059, '23'], [7.852
941176470588, '07'], [9.41095890410959, '12'], [23.810344827586206, '0
2'], [21.525, '20'], [10.25, '08'], [7.170212765957447, '04'], [13.4406
77966101696, '10'], [38.5948275862069, '15'], [8.127272727272727, '0
```

```
0'], [16.796296296296298, '16'], [11.051724137931034, '11'], [11.383333
333333333, '01'], [13.233644859813085, '14']]
```

In [55]:
```
sorted_swap = sorted(swap_average_by_hour, reverse = True)
print(sorted_swap)
```

```
[[38.5948275862069, '15'], [23.810344827586206, '02'], [21.525, '20'],
[16.796296296296298, '16'], [16.009174311926607, '21'], [14.74117647058
8234, '13'], [13.440677966101696, '10'], [13.233644859813085, '14'], [1
3.20183486238532, '18'], [11.46, '17'], [11.383333333333333, '01'], [1
1.051724137931034, '11'], [10.8, '19'], [10.25, '08'], [10.086956521739
13, '05'], [9.41095890410959, '12'], [9.022727272727273, '06'], [8.1272
72727272727, '00'], [7.985294117647059, '23'], [7.852941176470588, '0
7'], [7.796296296296297, '03'], [7.170212765957447, '04'], [6.746478873
239437, '22'], [5.5777777777777775, '09']]
```

Top 5 Hours for Ask Posts Comments

In [66]:
```
print("Top 5 Hours for Ask Posts Comments")
for avg,h in sorted_swap[:5]:
    time = dt.datetime.strptime(h, "%H").strftime("%H:%M")
    print("{}: {:.2f} average comments per post".format(time,avg))
```

```
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
```

Conclusion

We find that hours in which one should create a post in order to recieve maximum number of comments is 3pm. And the rest of top hours following the 3pm time are 2am, 8pm, 4pm, 9pm respectively. As per the documentation(link provided in beginning of project), The time zone is Eastern Time in the US.