HACKER NEWS

Hacker News is a social news website focusing on computer science and entrepreneurship. It is run by Paul Graham's investment fund and startup incubator, Y Combinator. In general, content that can be submitted is defined as "anything that gratifies one's intellectual curiosity.

We wil be comparing two types of posts:

1. Do Ask HN or Show HN receive more comments on average?
2. Do posts created at a certain time reveive more comments on average?

You can view the source here at dataset (https://www.kaggle.com/hacker-news/hacker-news-posts).

In [1]:
```
from csv import reader

open_file = open('hacker_news.csv',  encoding='utf8')
read_file = reader(open_file)
hn = list(read_file)
print(hn[0:5])
```

```
[['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at'], ['12579008', 'You have two days to comment if you want stem cells to be classified as your own', 'http://www.regulations.gov/document?D=FDA-2015-D-3719-0018', '1', '0', 'altstar', '9/26/2016 3:26'], ['12579005', 'SQLAR  the SQLite Archiver', 'https://www.sqlite.org/sqlar/doc/trunk/README.md', '1', '0', 'blacksqr', '9/26/2016 3:24'], ['12578997', 'What if we just printed a flatscreen television on the si de of our boxes?', 'https://medium.com/vanmoof/our-secrets-out-f21c1f03fdc8#.ietxmez43', '1', '0', 'pavel_lishin', '9/26/2016 3:19'], ['12578989', 'algorithmic music', 'http://cacm.acm.org/magazines/2011/7/109891-algorithmic-composition/fulltext', '1', '0', 'poindontcare', '9/26/2016 3:16']]
```

In [2]:
```
hn_headers = hn[0]
hn = hn[1:]
```

Seperating Posts

We used methods to seperate posts beginning with **Ask HN** and **Show HN** into two different lists.

We then appended all the other posts to a different list to create a more clean and readible lists to use later that will help us

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('Number of posts in Ask Posts List:', len(ask_posts))
print('Number of posts in Show Posts List:', len(show_posts))
print('Number of posts in Other Posts List',len(other_posts))
```

```
Number of posts in Ask Posts List: 9139
Number of posts in Show Posts List: 10158
Number of posts in Other Posts List 273822
```

In [4]:
```
print(hn_headers)
print(ask_posts[0])
```

```
['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at']
['12578908', 'Ask HN: What TLD do you use for local development?', '', '4', '7', 'Sevrene', '9/26/2016 2:53']
```

Average Number of Comments

We will be calculating the average number of comments on Ask HN and Show HN posts to be able to see who has more comments on average.

In [5]:
```
total_ask_comments = 0

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

print('Average comments for Ask HN',avg_ask_comments)

total_show_comments = 0

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

print('Average comments for Show HN',avg_show_comments)
```

```
Average comments for Ask HN 10.393478498741656
Average comments for Show HN 4.886099625910612
```

Calculating average on ask posts list

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

In [6]:
```
import datetime as dt

result_list = []

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


posts_by_hour = {}
comments_by_hour = {}

for row in result_list:
    date = row[0]
    n_comments = row[1]
    time = dt.datetime.strptime(date, "%m/%d/%Y %H:%M").strftime("%H") # Extracting using strftime
    if time not in posts_by_hour:
        posts_by_hour[time] = 1
        comments_by_hour[time] = n_comments
    else:
        posts_by_hour[time] += 1
        comments_by_hour[time] += n_comments
print("Hour: Number of comments")
print(comments_by_hour)
print('\n')
print('Hour: Counts')
print(posts_by_hour)
```

```
Hour: Number of comments
{'02': 2996, '01': 2089, '22': 3372, '21': 4500, '19': 3954, '17': 5547, '15': 18525, '14': 4972, '13': 7245, '11': 2797, '10': 3013, '09': 1477, '07': 1585, '03': 2154, '23': 2297, '20': 4462, '16': 4466, '08': 2362, '00': 2277, '18': 4877, '12': 4234, '04': 2360, '06': 1587, '05': 1838}


Hour: Counts
{'02': 269, '01': 282, '22': 383, '21': 518, '19': 552, '17': 587, '15': 646, '14': 513, '13': 444, '11': 312, '10': 282, '09': 222, '07': 226, '03': 271, '23': 343, '20': 510, '16': 579, '08': 257, '00': 301, '18': 614, '12': 342, '04': 243, '06': 234, '05': 209}
```

Average number of comments per post.

In [7]:
```
avg_by_hour = []


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

print(avg_by_hour)
```

```
[['02', 11.137546468401487], ['01', 7.407801418439717], ['22', 8.804177545691905], ['21', 8.687258687258687], ['19', 7.163043478260869], ['17', 9.449744463373083], ['15', 28.676470588235293], ['14', 9.692007797270955], ['13', 16.31756756756757], ['11', 8.96474358974359], ['10', 10.684397163120567], ['09', 6.653153153153153], ['07', 7.013274336283186], ['03', 7.948339483394834], ['23', 6.696793002915452], ['20', 8.749019607843136], ['16', 7.713298791018998], ['08', 9.190661478599221], ['00', 7.5647840531561465], ['18', 7.94299674267101], ['12', 12.380116959064328], ['04', 9.7119341563786], ['06', 6.782051282051282], ['05', 8.794258373205741]]
```

Swopping first and second index

Swop the indexs so that average comments are first and hour is second so that we can sort them in desending order and find the best time to post on Ask HN posts.

In [8]:
```
swap_avg_by_hour = []

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

sorted_swap = sorted(swap_avg_by_hour, reverse=True)
```

Final Conclusion

These are the Top 5 hours in which you should create a post in order to have a higher chance of receiving comments.

In [9]:
```
print("Top 5 Hours for 'Ask HN' Comments")
for avg, hr in sorted_swap[:5]:
    print("{}: {:.2f} average comments per post".format(
          dt.datetime.strptime(hr, "%H").strftime("%H:%M"),avg))
```

```
Top 5 Hours for 'Ask HN' Comments
15:00: 28.68 average comments per post
13:00: 16.32 average comments per post
12:00: 12.38 average comments per post
02:00: 11.14 average comments per post
10:00: 10.68 average comments per post
```