Hacker News Posts Analysis

In this project, we are going to explore two types of posts from the 'Hacker News' data set. 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 [1]:

```
from csv import reader

opened_file = open('hacker_news.csv')
read_file = reader(opened_file)
hn = list(read_file)

print(hn[:5])
```

```
[['id', 'title', 'url', 'num_points', 'num_comments', 'author', 'created_at'], ['12579008', 'You h
ave 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 side 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-al
gorithmic-composition/fulltext', '1', '0', 'poindontcare', '9/26/2016 3:16']]
```

Extracting Header

In [2]:

```
hn_header = hn[0]
hn = hn[1:]

print(hn_header)
print(hn[: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 side 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-al
gorithmic-composition/fulltext', '1', '0', 'poindontcare', '9/26/2016 3:16'], ['12578979', 'How th
e Data Vault Enables the Next-Gen Data Warehouse and Data Lake',
'https://www.talend.com/blog/2016/05/12/talend-and-Â\x93the-data-vaultÂ\x94', '1', '0',
'markgainor1', '9/26/2016 3:14']]
```

Cleaning the data

We are going to extract Ask HN or Show HN from the 'hn' list of lists. We will iterate through hn and find any title starts with Ask HN or Show HN using the string method 'startswith'.

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

```
Number of Ask HN posts:  9139
Number of Show HN posts:  10158
Number of Other posts:  273822
```

Below are the first five rows in the ask_posts list of lists.

In [4]:

```
ask_posts[:5]
```

Out[4]:

```
[['12578908',
  'Ask HN: What TLD do you use for local development?',
  '',
  '4',
  '7',
  'Sevrene',
  '9/26/2016 2:53'],
 ['12578522',
  'Ask HN: How do you pass on your work when you die?',
  '',
  '6',
  '3',
  'PascLeRasc',
  '9/26/2016 1:17'],
 ['12577908',
  'Ask HN: How a DNS problem can be limited to a geographic region?',
  '',
  '1',
  '0',
  'kuon',
  '9/25/2016 22:57'],
 ['12577870',
  'Ask HN: Why join a fund when you can be an angel?',
  '',
  '1',
  '3',
  'anthony_james',
  '9/25/2016 22:48'],
 ['12577647',
  'Ask HN: Someone uses stock trading as passive income?',
  '',
  '5',
  '2',
  '00taffe',
  '9/25/2016 21:50']]
```

Below are the first five rows in the show_posts list of lists.

In [5]:

```
show_posts[:5]
```

Out[5]:

```
[['12578335',
```

```
  'Show HN: Finding puns computationally',
  'http://puns.samueltaylor.org/',
  '2',
  '0',
  'saamm',
  '9/26/2016 0:36'],
 ['12578182',
  'Show HN: A simple library for complicated animations',
  'https://christinecha.github.io/choreographer-js/',
  '1',
  '0',
  'christinecha',
  '9/26/2016 0:01'],
 ['12578098',
  'Show HN: WebGL visualization of DNA sequences',
  'http://grondilu.github.io/dna.html',
  '1',
  '0',
  'grondilu',
  '9/25/2016 23:44'],
 ['12577991',
  'Show HN: Pomodoro-centric, heirarchical project management with ES6 modules',
  'https://github.com/jakebian/zeal',
  '2',
  '0',
  'dbranes',
  '9/25/2016 23:17'],
 ['12577142',
  'Show HN: Jumble  Essays on the go #PaulInYourPocket',
  'https://itunes.apple.com/us/app/jumble-find-startup-essay/id1150939197?ls=1&mt=8',
  '1',
  '1',
  'ryderj',
  '9/25/2016 20:06']]
```

Calculate Number of Comments

Since we are using the number of comments as our determinant, we need to calculate average number of comments for each situations.

In [6]:

```
total_ask_comments = 0

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

avg_ask_comments = total_ask_comments / len(ask_posts)
print('Average number of Ask HN post comments: ', avg_ask_comments)

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

avg_show_comments = total_show_comments / len(show_posts)
print('Average number of Show HN post comments: ',avg_show_comments)
```

```
Average number of Ask HN post comments:  10.393478498741656
Average number of Show HN post comments:  4.886099625910612
```

On average, Ask HN posts get about 6 more comments compare to the Show HN posts. Since Ask HN post receive more comments on average, we are going to focus on Ask HN posts for the remaining analysis.

Finding the Amount of Ask Posts and Comments by Hour Created

In [7]:

```
import datetime as dt

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

counts_by_hour = {}
comments_by_hour = {}
for i in result_list:
    date_time = i[0]
    time = dt.datetime.strptime(date_time, '%m/%d/%Y %H:%M')
    hour = time.strftime('%H')
    if hour not in counts_by_hour:
        counts_by_hour[hour] = 1
        comments_by_hour[hour] = i[1]
    else:
        counts_by_hour[hour] += 1
        comments_by_hour[hour] += i[1]

sorted_comments_by_hour = sorted(comments_by_hour.items(), key = lambda x:x[1], reverse = True)
sorted_comments_by_hour
```

Out[7]:

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

Calculating the Average Number of Comments for Ask HN Posts by Hour

In [8]:

```
avg_comt_by_hour = []

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

avg_comt_by_hour
```

Out[8]:

```
[['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]]
```

Sorting and Printing Values from a List of Lists

In [16]:

```
for i in avg_comt_by_hour:
    hour = i[0]
    avg_comment = round(i[1],2)
    i[1] = avg_comment

sorted_avg_comt_by_hour = sorted(avg_comt_by_hour, key=lambda x:x[1], reverse=True)
sorted_avg_comt_by_hour[:5]
```

Out[16]:

```
[['15', 28.68], ['13', 16.32], ['12', 12.38], ['02', 11.14], ['10', 10.68]]
```

As we can see, the hour that receives the most comments per post on average is 15:00 with about 28.68 comments per post. There is a significant increase in the number of comments between 15:00 and 13:00.

It would be feasible to say writing a post on 15:00 (3:00 pm est.) is highly recommended for anyone who want their question answered.

Conclusion

We analyzed two types of posts and determined which type of post and time received the most comments on average. Based on our obsevation, we recommend the Ask HN category and create the post between 15:00 and 16:00 (3:00 - 4:00 pm).